

موسسه بابان

انتشارات بابان و انتشارات راهیان ارشد

پایگاه داده‌ها

فصل سوم: مدل رابطه‌ای

(ویژه کنکور ارشد ۱۴۰۵)

ویژه‌ی داوطلبان کنکور کارشناسی ارشد مهندسی کامپیوتر و IT

براساس کتب مرجع

آبراهام سیلبرشاتز، راما کریشنن، رامز المصری، سی جی دیت، توماس کانلی و کارولین بگ

ارسطو خلیلی فر

مدل رابطه‌ای

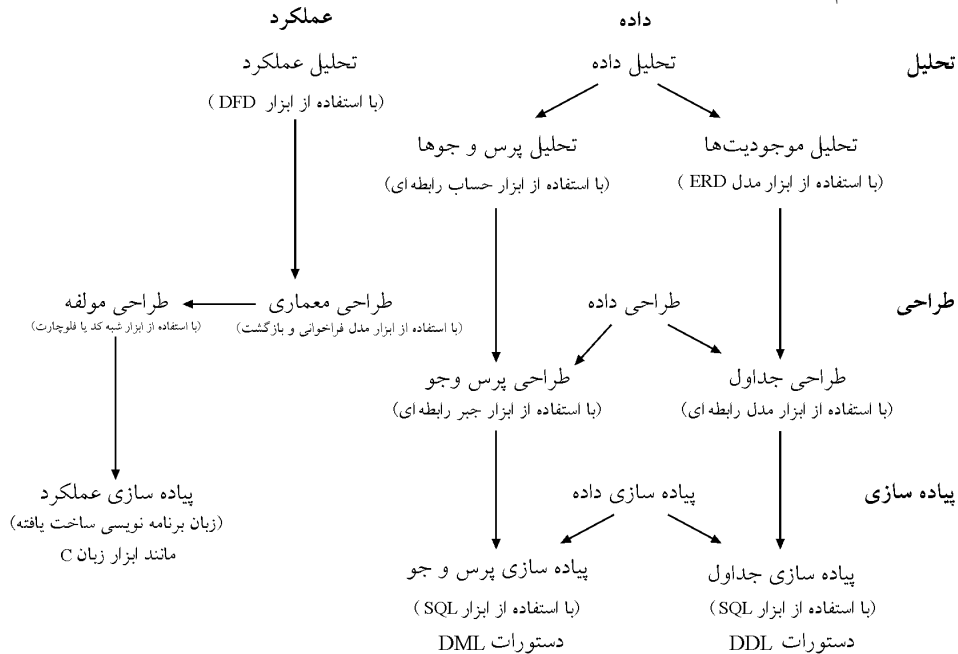
۳

مقدمه

همانطور که در فصل اول (مفاهیم پایه) گفتیم، یک محصول نرم‌افزاری به واسطه‌ی فرآیند تولید نرم‌افزار که شامل فعالیت‌های مدل تحلیل، مدل طراحی، پیاده‌سازی و تست می‌باشد، ایجاد می‌گردد.

مدل تحلیل

در فصل دوم (مدل تحلیل یا مدل ER) بررسی شد. شکل زیر گویای این مطلب می‌باشد:



مدل طراحی

پس از مدل تحلیل، نوبت به مدل طراحی می‌رسد، پس از مدل تحلیل، باید برای پیاده‌سازی نرم‌افزار آماده شد. واضح است که گذر مستقیم به مرحله تولید کد و پیاده‌سازی عاقلانه نیست. طراحی داده بر دو بخش **طراحی جدول** و **طراحی پرس‌وجو** می‌باشد. طراحی جدول از بخش طراحی داده، تحلیل موجودیت (ERD) از مدل تحلیل را به عنوان ورودی دریافت کرده و توسط مدل رابطه‌ای، طراحی جدول را انجام می‌دهد. طراحی پرس‌وجو از بخش طراحی داده، تحلیل پرس‌وجو (حساب رابطه‌ای) از مدل تحلیل را به عنوان ورودی دریافت کرده و توسط جبر رابطه‌ای، طراحی پرس‌وجو را انجام می‌دهد.

توجه: جهت ذخیره‌سازی اطلاعات نیاز به جدول داریم و برای بازیابی اطلاعات بر اساس پرس‌وجوهای مورد نظر نیاز به پرس‌وجونویسی داریم. در پایگاه داده رابطه‌ای که بر پایه مجموعه‌ها است، جدول توسط مدل رابطه‌ای و پرس‌وجو توسط جبر رابطه‌ای ایجاد می‌شود.

مدل رابطه‌ای (Relational Model)

برای شناسایی هر مدل یا ساختاری ابتدا باید به بررسی بخش‌های مختلف آن مدل یا ساختار پردازیم. مدل رابطه‌ای از دو بخش زیر تشکیل شده است:

۱- رابطه (جدول)

۲- قوانین مدل رابطه‌ای

توجه: در مدل تحلیل منظور از رابطه، ارتباط بین موجودیت‌هاست، اما در مدل طراحی منظور از رابطه، جداول مدل رابطه‌ای است.

جبر رابطه‌ای (Relational Algebra)

جبر رابطه‌ای از دو بخش زیر تشکیل شده است:

۱- عملوندها (جداول)

۲- عملگرها

توجه: با استفاده از عملگرهای موجود در پرس‌وجوها، داده‌ها از عملوندها (جداول) استخراج می‌شوند. **عملوندها** در ادامه همین فصل مدل رابطه‌ای و **عملگرها** در فصل جبر رابطه‌ای به طور مفصل تشریح می‌شود.

رابطه (Relation)

در ریاضیات به هر زیرمجموعه دلخواه از حاصلضرب دکارتی دنباله‌ای از دامنه‌ها یک رابطه گفته می‌شود. به مجموعه مقادیر ممکن یک صفت، **دامنه** یا **میدان** (Domain) گفته می‌شود.

مثال: دامنه‌های D_1 و D_2 را به صورت زیر در نظر بگیرید:

$$D_1 = \{a, b\}$$

$$D_2 = \{x, y\}$$

حاصلضرب دکارتی دو دامنه D_1 و D_2 به صورت زیر است:

$$D_1 \times D_2 = \{(a, x), (a, y), (b, x), (b, y)\}$$

رابطه R به عنوان یک زیر مجموعه دلخواه از حاصلضرب دکارتی دو دامنه D_1 و D_2 به صورت زیر است:

$$R \subseteq D_1 \times D_2$$

$$R = \{(a, x), (a, y), (b, y)\}$$

توجه: به هر یک از اعضای رابطه، **تاپل (Tuple)** یا یک **چندتایی مرتب** گفته می‌شود. به یک **دوتایی مرتب**، **زوج مرتب** یا **جفت مرتب** نیز گفته می‌شود. برای مثال (a, x) یک تاپل از رابطه R است.

جدول (Table)

بهترین راه نمایش و پیاده‌سازی رابطه استفاده از جدول است، پس جدول یک شیوه نمایش تاپل‌های رابطه است. جدول از تعدادی سطر (رکورد یا تاپل (Tuple)) و ستون تشکیل شده است. رابطه T به عنوان یک زیر مجموعه دلخواه از حاصلضرب دکارتی دو دامنه D_1 و D_2 به صورت زیر است:

D_1	D_2
a	x
a	y
b	y

جدول T

رابطه R به عنوان یک زیر مجموعه تهی از حاصلضرب دکارتی دو دامنه D_1 و D_2 به صورت زیر است:

$$R = \{\}$$

جدول T به صورت زیر است:

D_1	D_2
NULL	NULL

جدول T

مثال: دامنه‌های D_2 و D_3 را به صورت زیر در نظر بگیرید:

$$D_2 = \{x, y\}$$

$$D_3 = \{1, 2, 3\}$$

حاصلضرب دکارتی دو دامنه D_2 و D_3 به صورت زیر است:

$$D_2 \times D_3 = \{(x, 1), (x, 2), (x, 3), (y, 1), (y, 2), (y, 3)\}$$

رابطه R به عنوان یک زیر مجموعه دلخواه از حاصلضرب دکارتی دو دامنه D_2 و D_3 به صورت زیر است:

$$R \subseteq D_2 \times D_3$$

$$R = \{(x, 1), (x, 2), (y, 1), (y, 2)\}$$

جدول T به صورت زیر است:

D_2	D_3
x	1
x	2
y	1
y	2

جدول T

رابطه R به عنوان یک زیر مجموعه تهی از حاصلضرب دکارتی دو دامنه D_2 و D_3 به صورت زیر است:

$$R = \{\}$$

جدول T به صورت زیر است:

D_2	D_3
NULL	NULL

جدول T

مثال: دامنه‌های D_1 ، D_2 و D_3 را به صورت زیر در نظر بگیرید:

$$D_1 = \{a, b\}$$

$$D_2 = \{x, y\}$$

$$D_3 = \{1, 2, 3\}$$

حاصلضرب دکارتی سه دامنه D_1 ، D_2 و D_3 به صورت زیر است:

$$D_1 \times D_2 \times D_3 = \{(a, x, 1), (a, x, 2), (a, x, 3), (a, y, 1), (a, y, 2), (a, y, 3), \\ (b, x, 1), (b, x, 2), (b, x, 3), (b, y, 1), (b, y, 2), (b, y, 3)\}$$

رابطه R به عنوان یک زیر مجموعه دلخواه از حاصلضرب دکارتی سه دامنه D_1 ، D_2 و D_3 به صورت زیر است:

$$R \subseteq D_1 \times D_2 \times D_3$$

$$R = \{(a, x, 1), (a, x, 2), (b, x, 1), (b, x, 2)\}$$

جدول T به صورت زیر است:

D_1	D_2	D_3
a	x	1
a	x	2
b	x	1
b	x	2

جدول T

رابطه R به عنوان یک زیر مجموعه تهی از حاصلضرب دکارتی سه دامنه D_1 ، D_2 و D_3 به صورت زیر است:

$$R = \{\}$$

جدول T به صورت زیر است:

D_1	D_2	D_3
NULL	NULL	NULL

جدول T

ویژگی‌های رابطه در مدل رابطه‌ای

۱- در یک رابطه، تاپل تکراری وجود ندارد. زیرا رابطه یک مجموعه است و مجموعه در ریاضیات طبق تعریف، تاپل تکراری ندارد. بنابراین در جدول نیز، رکورد تکراری وجود ندارد، جدای از این، رابطه یک زیر مجموعه دلخواه از حاصلضرب دکارتی چند دامنه است و ضرب دکارتی به طور ذاتی هیچگاه تاپل تکراری تولید نمی‌کند.

مثال: دامنه‌های D_1 و D_2 را به صورت زیر در نظر بگیرید:

$$D_1 = \{a, b\}$$

$$D_2 = \{x, y\}$$

حاصلضرب دکارتی دو دامنه D_1 و D_2 به صورت زیر است:

$$D_1 \times D_2 = \{(a, x), (a, y), (b, x), (b, y)\}$$

در واقع ضرب دکارتی فقط و فقط یکبار یک چندتایی مرتب به هم چسبیده را تولید می‌کند، بنابراین به طور ذاتی هیچگاه دو تاپل کاملاً یکسان مثل $\{(a, x), (a, x)\}$ زیر مجموعه دلخواه از حاصلضرب دکارتی چند دامنه نخواهد بود. پس در رابطه R امکان ثبت تاپل تکراری (a, x) وجود ندارد.

$$R = \{(a, x), (a, y), (b, y)\}$$

D_1	D_2
a	x
a	y
b	y

جدول T

۲- در یک رابطه، جایگاه و ترتیب تاپل‌ها هیچ اهمیتی ندارد. زیرا در یک مجموعه، جایگاه و ترتیب اعضا هیچ اهمیتی ندارد $\{1, 2, 3\} = \{2, 1, 3\}$. بنابراین در یک جدول نیز، جایگاه و ترتیب رکوردها هیچ اهمیتی ندارد.

مثال: دامنه‌های D_1 و D_2 را به صورت زیر در نظر بگیرید:

$$D_1 = \{a, b\}$$

$$D_2 = \{x, y\}$$

حاصلضرب دکارتی دو دامنه D_1 و D_2 به صورت زیر است:

$$D_1 \times D_2 = \{(a, x), (a, y), (b, x), (b, y)\}$$

$$R = \{(a, x), (a, y), (b, y)\} = \{(a, y), (b, y), (a, x)\}$$

D_1	D_2	=	D_1	D_1
a	x		a	y
a	y		b	y
b	y		a	x

جدول T

جدول T

۳- در یک رابطه، جایگاه و ترتیب مولفه‌ها (خصیصه‌ها) داخل تاپل‌ها هیچ اهمیتی ندارد. یعنی مولفه‌های تاپل‌ها را می‌توان جابه‌جا کرد. البته به شرطی که نسبت به رابطه درست اولیه، مولفه‌ها در تمام تاپل‌ها به طور یکسان جابه‌جا شود. بنابراین در یک جدول نیز، جایگاه و ترتیب ستون‌ها داخل رکوردها هیچ اهمیتی ندارد. یعنی ستون‌های رکوردها را می‌توان جابه‌جا کرد. البته به شرطی که نسبت به جدول درست اولیه، ستون‌ها در تمام رکوردها به طور یکسان جابه‌جا شود.

مثال: دامنه‌های D_1 ، D_2 و D_3 را به صورت زیر در نظر بگیرید:

$$D_1 = \{a, b\}$$

$$D_2 = \{x, y\}$$

$$D_3 = \{1, 2, 3\}$$

حاصلضرب دکارتی سه دامنه D_1 ، D_2 و D_3 به صورت زیر است:

$$D_1 \times D_2 \times D_3 = \{(a, x, 1), (a, x, 2), (a, x, 3), (a, y, 1), (a, y, 2), (a, y, 3), \\ (b, x, 1), (b, x, 2), (b, x, 3), (b, y, 1), (b, y, 2), (b, y, 3)\}$$

$$R = \{(a, x, 1), (a, x, 2), (b, x, 1), (b, x, 2)\}$$

اگر ترتیب مولفه‌های تاپل‌های رابطه R' نسبت به R به صورت زیر باشد:

$$R' = \{(x, 1, a), (x, 2, a), (x, 1, b), (x, 2, b)\}$$

آنگاه رابطه R' معادل رابطه R است ($R = R'$)، زیرا نسبت به رابطه درست اولیه R ، مولفه‌ها در تمام تاپل‌ها به طور یکسان جابه‌جا شده‌اند.

D_1	D_2	D_3	=	D_2	D_3	D_1
a	x	1		x	1	a
a	x	2		x	2	a
b	x	1		x	1	b
b	x	2		x	2	b

جدول T

جدول T'

اگر ترتیب مولفه‌های تاپل‌های R'' نسبت به R به صورت زیر باشد:

$$R'' = \{(x, 1, a), (2, x, a), (1, b, x), (x, 2, b)\}$$

آنگاه R معادل رابطه R نیست ($R \neq R''$)، زیرا نسبت به رابطه درست اولیه R ، مولفه‌ها در تمام تاپل‌ها به طور یکسان جابه‌جا نشده‌است.

۴- هر مجموعه‌ای از تاپل‌ها الزاماً یک رابطه نیست، بلکه اگر مجموعه‌ای از تاپل‌ها بخواهد یک رابطه را تشکیل دهد، باید **دو شرط** در مورد تاپل‌های آن صدق کند:

شرط اول: مولفه‌ها در تمام تاپل‌های یک رابطه باید باهم **هم‌دامنه** باشند.

مثال: تاپل‌های مجموعه $\{(a, x), (x, l)\}$ **هم‌دامنه** نیست، زیرا تاپل (a, x) از حاصلضرب دکارتی دو دامنه D_1 و D_2 و تاپل (x, l) از حاصلضرب دکارتی دو دامنه D_2 و D_3 انتخاب شده است.

شرط دوم: مولفه‌ها در تمام تاپل‌های یک رابطه باید باهم **هم‌درجه** باشند.

مثال: تاپل‌های مجموعه $\{(a, x), (a, x, l)\}$ **هم‌درجه** نیست، زیرا تاپل (a, x) درجه 2 و تاپل (a, x, l) درجه 3 است.

خلاصه دو شرط: تمام تاپل‌های یک مجموعه باید باهم **هم‌تیر** (**هم‌دامنه** و **هم‌درجه**) باشند، تا به آن رابطه گفته شود.

نتیجه: اگر همه اعضای یک مجموعه، فقط و فقط از یک حاصلضرب دکارتی انتخاب شود، آنگاه آن مجموعه قطعاً رابطه است و به طور ذاتی همه اعضای رابطه باهم **هم‌تیر** (**هم‌دامنه** و **هم‌درجه**) هستند.

مثال: دامنه‌های D_1 و D_2 را به صورت زیر در نظر بگیرید:

$$D_1 = \{a, b\}$$

$$D_2 = \{x, y\}$$

حاصلضرب دکارتی دو دامنه D_1 و D_2 به صورت زیر است:

$$D_1 \times D_2 = \{(a, x), (a, y), (b, x), (b, y)\}$$

رابطه R به عنوان یک زیر مجموعه دلخواه از حاصلضرب دکارتی دو دامنه D_1 و D_2 به صورت زیر است:

$$R = \{(a, x), (a, y), (b, y)\}$$

نتیجه: همه اعضای مجموعه R ، فقط و فقط از حاصلضرب دکارتی $D_1 \times D_2$ انتخاب شده‌است، بنابراین به طور ذاتی همه اعضای رابطه R باهم **هم‌تیر** (**هم‌دامنه** و **هم‌درجه**) هستند.

۵- در یک رابطه، تمام مولفه‌ها در تاپل‌ها تجزیه‌ناپذیرند. به عبارت دیگر در مدل رابطه‌ای نمی‌توان فیلد مرکبی که از فیلدهای ساده تشکیل شده‌است، تعریف نمود. یعنی فیلدهای موجود در جدول مدل رابطه‌ای **نباید مرکب** مثل تاریخ تولد و آدرس باشد و **نباید چندمقداری** مثل مدرک تحصیلی باشد. به عبارت دیگر هیچ رابطه‌ای نمی‌تواند شامل تاپل‌های تودرتو، تاپل تو تاپل یا جدول تو جدول باشد، به بیان دیگر مولفه‌های هریک از تاپل‌های یک رابطه نباید مرکب یا چندمقداری باشد، به عبارت ساده‌تر هر مولفه یک تاپل، نمی‌تواند خودش یک تاپل باشد.

مثال: مجموعه R را در نظر بگیرید:

$$R = \{((a, x), 1), (a, x, 2), (b, x, 1), (b, x, 2)\}$$

مجموعه R یک رابطه نیست، زیرا مولفه نخست تاپل اول یعنی (a, x) از $((a, x), 1)$ خودش یک تاپل است که مطابق قوانین مدل رابطه‌ای، هر مولفه یک تاپل، نمی‌تواند خودش یک تاپل باشد. اما همانطور که گفتیم رابطه R به عنوان یک زیر مجموعه دلخواه از حاصلضرب دکارتی سه دامنه D_1, D_2 و D_3 یک رابطه است.

$$R = \{(a, x, 1), (a, x, 2), (b, x, 1), (b, x, 2)\}$$

مثال: طراحی نادرست صفت مرکب تاریخ تولد

$$R = \{(S1, (1373, 7, 5)), (S2, (1381, 6, 4))\}$$

SID	Date of birth		
S1	1373	7	5
S2	1381	6	4

جدول T

مثال: طراحی درست صفت مرکب تاریخ تولد

$$R = \{(S1, 1373, 7, 5), (S2, 1381, 6, 4)\}$$

SID	Year	Month	Day
S1	1373	7	5
S2	1381	6	4

جدول T

خواص رابطه در مدل رابطه‌ای

بطور کلی خواص رابطه به دو شکل زیر بررسی می‌گردد:

الف) درجه رابطه

ب) کاردینالیته رابطه

توجه: همه روابط دو خصیصه فوق را به طور هم‌زمان دارند، اما مقادیر این خصیصه‌ها در روابط مختلف، متفاوت است.

الف) درجه رابطه

به تعداد ستون‌های یک رابطه، **درجه رابطه** گفته می‌شود. تعداد ستون‌های یک رابطه همواره بزرگتر از صفر $\deg(R) > 0$ می‌باشد، یعنی جدول بدون ستون نداریم.

مثال: جدول S را به صورت زیر در نظر بگیرید:

<u>S#</u>	Sname	City
S1	Sn1	C1
S2	Sn2	C2
S3	Sn3	C3
S4	Sn4	C4

درجه رابطه فوق برابر مقدار 3 است، به صورت زیر:

$$\deg(S) = 3$$

ب) کاردینالیتی رابطه

به تعداد سطرهای یک رابطه، **کاردینالیتی رابطه** گفته می‌شود. تعداد سطرهای یک رابطه برابر صفر یا بزرگتر از صفر $\text{card}(R) \geq 0$ می‌باشد، یعنی جدول بدون سطر داریم که جدول تهی است.

مثال: جدول S را به صورت زیر در نظر بگیرید:

<u>S#</u>	Sname	City
S1	Sn1	C1
S2	Sn2	C2
S3	Sn3	C3
S4	Sn4	C4

کاردینالیتی رابطه فوق برابر مقدار 4 است، به صورت زیر:

$$\text{card}(S) = 4$$

<u>S#</u>	Sname	City
NULL	NULL	NULL

کاردینالیتی رابطه فوق برابر مقدار 0 است، به صورت زیر:

$$\text{card}(S) = 0$$

مثال: سیستم تولیدکنندگان و قطعات شامل جداول S، SP و P را در نظر بگیرید.

<u>S#</u>	Sname	City	<u>S#</u>	<u>P#</u>	QTY	<u>P#</u>	Pname	Color
S1	Sn1	C1	S1	P1	10	P1	Pn1	Red
S2	Sn2	C2	S1	P2	20	P2	Pn2	Blue
S3	Sn3	C3	S2	P1	30	جدول P (قطعات)		
جدول S (تولیدکنندگان)			جدول SP (تولید)					

دامنه‌های D_1 برای ستون $S\#$ ، D_2 برای ستون Sname و D_3 برای ستون City را برای جدول S به صورت زیر در نظر بگیرید:

$$D_1 = \{S1, S2, S3\}$$

$$D_2 = \{Sn1, Sn2, Sn3\}$$

$$D_3 = \{C1, C2, C3\}$$

حاصلضرب دکارتی سه دامنه D_1 ، D_2 و D_3 به صورت زیر است:

$$D_1 \times D_2 \times D_3 = \{(S1, Sn1, C1), (S1, Sn1, C2), (S1, Sn1, C3),$$

$$(S1, Sn2, C1), (S1, Sn2, C2), (S1, Sn2, C3),$$

$$(S1, Sn3, C1), (S1, Sn3, C2), (S1, Sn3, C3),$$

$$(S2, Sn1, C1), (S2, Sn1, C2), (S2, Sn1, C3),$$

$$(S2, Sn2, C1), (S2, Sn2, C2), (S2, Sn2, C3),$$

$$(S2, Sn3, C1), (S2, Sn3, C2), (S2, Sn3, C3)$$

$$(S3, Sn1, C1), (S3, Sn1, C2), (S3, Sn1, C3),$$

$$(S3, Sn2, C1), (S3, Sn2, C2), (S3, Sn2, C3),$$

$$(S3, Sn3, C1), (S3, Sn3, C2), (S3, Sn3, C3)\}$$

رابطه S به عنوان یک زیر مجموعه دلخواه از حاصلضرب دکارتی سه دامنه D_1 ، D_2 و D_3 به صورت زیر است:

$$S = \{(S1, Sn1, C1),$$

$$(S2, Sn2, C2),$$

$$(S3, Sn3, C3)\}$$

جدول S به صورت زیر است:

S#	Sname	City
S1	Sn1	C1
S2	Sn2	C2
S3	Sn3	C3

جدول S

توجه: رابطه یا جدول S یک **تصویر لحظه‌ای (Snapshot)** از حاصلضرب دکارتی فوق است.

انواع دامنه‌های استاندارد در زبان SQL

انواع دامنه‌هایی که در زبان SQL پشتیبانی می‌شود، به صورت زیر است:

Integer ، Smallint ، Float ، Real ، Double ، Numeric(p,q) ، Char(n) ، Varchar(n) ،
Date ، Time ، Timestamp ، Logical و Blob

دستور Create Domain

این دستور جهت ایجاد یک دامنه جدید علاوه بر دامنه‌های استاندارد مورد استفاده قرار می‌گیرد. ساختار کلی این دستور به صورت زیر است:

```
Create Domain domain_name AS DataType
Default ...
Constraint constraint_name
Check (Value...)
```

مثال:

```
Create Domain mynumbers AS Integer
Default 1
Constraint myn
Check (Value >= 1 and Value <=1000 )
```

پس از تعریف دامنه جدید می‌توان از mynumbers برای تعریف دامنه یک ستون از یک جدول استفاده کرد، دامنه جدید کییدامنه، mynumbers فقط اعداد بازه 1 تا 1000 را به عنوان مقدار می‌پذیرد. Constraint myn نام محدودیت جامعیتی است که بعد آن توسط Check تعریف شده است.

مثال:

```
Create Table SP
(
S# char (5),
P# char (5),
QTY mynumbers,
Primary key (S#, P#),
...
)
```

توجه: در جدول SP برای ستون QTY دامنه جدید mynumbers مورد استفاده قرار گرفته است.

مفهوم کلید در پایگاه داده

کلید در پایگاه داده مفهومی نیست که از دنیای کامپیوتر به دنیای واقعی (محیط عملیاتی) رفته باشد؛ بلکه به عکس، مفهوم کلید از دنیای واقعی وارد کامپیوتر شده است. مفهوم کلید در دنیای واقعی وجود دارد، پس در دنیای کامپیوتر نیز باید پیاده‌سازی شود، چون انسان نیاز به جستجو دارد. در محیط واقعی:

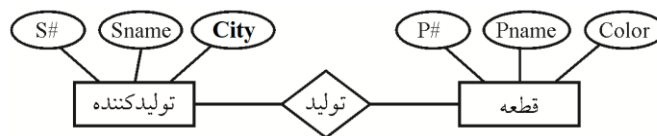
- هر فرد، کد ملی دارد.
- هر دانشجو، شماره دانشجویی دارد.
- هر تولیدکننده، شماره تولید کننده دارد.
- هر قطعه، شماره قطعه دارد.

توجه: کلید در دنیای واقعی (محیط عملیاتی) از جنس صفات است، بنابراین در مفاهیم پایگاه داده نیز از جنس صفات (ستون‌ها) است که در نهایت باید پیاده‌سازی شود.

توجه: کلید در دنیای واقعی از جنس **معنا** است و نه مقدار، بنابراین در پایگاه داده نیز بر اساس همان معنا انتخاب می‌شود و نه مقدار.

مثال: مفهوم کلید در سیستم تولیدکنندگان و قطعات

تحلیل موجودیت (توسط مدل ER)



طراحی جدول (توسط مدل رابطه‌ای)

S#	Sname	City	S#	P#	QTY	P#	Pname	Color
S1	Sn1	C1	S1	P1	10	P1	Pn1	Red
S2	Sn2	C2	S1	P2	20	P2	Pn2	Blue
S3	Sn3	C2	S2	P1	30	جدول P (قطعات)		
جدول S (تولیدکنندگان)			جدول SP (تولید)					

پیاده‌سازی جدول (توسط SQL:DDL)

Create Table S (S# char (5), Sname char (20), City char (15), Primary key (S#))	Create Table SP (S# char (5), P# char (5), QTY numeric (10), Primary key (S#, P#), Foreign key (S#) References S(S#) on delete cascade on update cascade, Foreign key (P#) References P(P#) on delete cascade on update cascade, Check (QTY>1 AND QTY<1000))	Create Table P (P# char (5), Pname char (20), Color char (10), Primary key (P#))
--	--	---

پیاده‌سازی پرس‌وجو (توسط SQL : DML)

پرس‌وجو روی ستون کلید S#: نام تولیدکننده شماره S1:

Select Sname
From S
Where S#='S1'

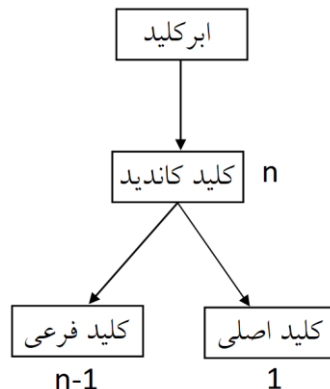
خروجی $\Rightarrow$

Sname
Sn1

خاصیت کلیدی: اگر بر اساس آن مورد، جستجو انجام شود، فقط و فقط یک مورد را به صورت یکتا برگرداند. در واقع قدرت تفکیک و تمایز یکتا داشته باشد. یعنی اگر براساس آن جستجو انجام شود، فقط و فقط یک نفر را می‌گیرد و بازمی‌گرداند. در اینجا S# خاصیت کلیدی دارد چون دقیقاً مطابق دنیای واقعی اگر براساس آن جستجو انجام شود، فقط و فقط یک سطر را بازمی‌گرداند.

انواع کلید در پایگاه داده‌ها

در مورد انواع کلید، جدول به جدول نظر می‌دهیم.



ابرکلید (Supper key)

هر ترکیبی از صفات (ستون‌ها) در **دنیای واقعی** که **خاصیت کلیدی** داشته باشد، یک **ابرکلید** است.

توجه: خاصیت کلیدی بر اساس دنیای واقعی در تمام لحظات ممکن (Historical Data) مشخص می‌گردد و نه محتوا و مقادیر جدول در یک لحظه خاص (Snapshot).

مثال: جدول S سیستم تولیدکنندگان و قطعات

S#	Sname	City
S ₁	Sn ₁	C ₁
S ₂	Sn ₂	C ₂
S ₃	Sn ₂	C ₃
S ₄	Sn ₄	C ₂

بررسی ابرکلید در جدول S:

- (S#): ابرکلید است. ✓ ؛ (Sname): ابرکلید نیست. X ؛ (City): ابرکلید نیست. X
 (Sname, City): ابرکلید نیست. X ؛ (S#, Sname): ابرکلید است. ✓
 (S#, City): ابرکلید است. ✓ ؛ (S#, Sname, City): ابرکلید است. ✓

(کارشناسی ارشد- دولتی ۷۳)

تست - کدام یک از چهار گزینه زیر کلید جدول رابطه‌ای زیر می‌باشد؟

C ₁	C ₂	C ₃
S	B	D ₁
T	J	D ₁
M	B	D ₁
T	C	D ₂
S	K	D ₂

{C₁} (۱)

{C₂, C₃} (۲)

{C₁, C₃} (۳)

{C₂} (۴)

پاسخ: گزینه (۳) صحیح است.

کلید کاندید (Candidate key)

هر ابرکلیدی که عضو زائد (چیز میز اضافه) نداشته باشد، کلید کاندید است، به عبارت دیگر به ابرکلید کمینه (minimal)، کلید کاندید گفته می‌شود. منظور از ابرکلید کمینه، ابرکلیدی نیست که کمترین تعداد صفت را داشته باشد، بلکه منظور، ابرکلیدی است که صفت زائد (چیز میز اضافه) نداشته باشد. نتیجه اینکه شروط کلید کاندید بودن بدین صورت است که **شرط لازم**: ابرکلید باشد و **شرط کافی**: عضو زائد نداشته باشد.

مثال:

S#	Sname	City	S#	P#	QTY	P#	Pname	Color
S ₁	Sn ₁	C ₁	S ₁	P ₁	10	P ₁	Pn ₁	Red
S ₂	Sn ₂	C ₂	S ₁	P ₂	20	P ₂	Pn ₂	Blue
S ₃	Sn ₃	C ₂	S ₂	P ₁	30	P ₃	Pn ₃	Blue

جدول S

جدول SP

جدول P

بررسی ابرکلید و کلید کاندید در جدول S :

- (S#): ابرکلید است. ✓ ، کلید کاندید هم هست. ✓
- (Sname): ابرکلید نیست. X ، کلید کاندید هم نیست. X
- (City): ابرکلید نیست. X ، کلید کاندید هم نیست. X
- (Sname, City): ابرکلید نیست. X ، کلید کاندید هم نیست. X
- (S#, Sname): ابرکلید است. ✓ ، کلید کاندید نیست. X ، چون عضو زائد Sname دارد. در واقع صفت S# ، به تنهایی خاصیت کلیدی دارد و به کمک کسی هم نیاز ندارد.
- (S#, City): ابرکلید است. ✓ ، کلید کاندید نیست. X ، چون عضو زائد City دارد.
- (S#, Sname, City): ابرکلید است. ✓ ، کلید کاندید نیست. X ، عضو زائد Sname, City دارد.

بررسی ابرکلید و کلید کاندید در جدول SP :

- (S#) : ابرکلید نیست. X ، کلید کاندید هم نیست. X
- (P#) : ابرکلید نیست. X ، کلید کاندید هم نیست. X
- (QTY) : ابرکلید نیست. X ، کلید کاندید هم نیست. X
- (S#, QTY) : ابرکلید نیست. X ، کلید کاندید هم نیست. X
- (P#, QTY) : ابرکلید نیست. X ، کلید کاندید هم نیست. X
- (S#, P#) : ابرکلید است. ✓ ، کلید کاندید هم هست. ✓
- (S#, P#, QTY) : ابرکلید است. ✓ ، کلید کاندید نیست. X ؛ چون عضو زائد QTY دارد.

توجه: کلید کاندید خودش به تنهایی خاصیت یکتا دارد و به کمک کسی هم نیاز ندارد.

توجه: یک کلید کاندید اگر در کنار خودش، هم نشین زائد و چیز میز اضافه داشته باشد، به ابرکلید تبدیل می شود. به قول سعدی شیرازی که در بوستان گفته است:

پسر نوح (کلید کاندید) با بدان (عضو زائد) بنشست خاندان نبوتش گم شد

کاربرد ابرکلید چیست؟

کشف کلید کاندید از دو طریق (۱) ابرکلید و (۲) وابستگی های تابعی امکان پذیر است. بنابراین کاربرد ابرکلید در کشف کلید کاندید است.

مراحل کشف کلید کاندید از طریق ابرکلید

مثال: جدول دانشجو

SID	NID	Name	Major	Term
-----	-----	------	-------	------

جدول Student

۱ - شناسایی ستون‌های جدول

SID(Student ID): شماره دانشجویی

NID(National ID): کد ملی

Name: نام

Major: رشته

Term: ترم

۲ - بررسی مجموعه‌های یکتا (ابرقیدها)

هر ترکیبی از صفات که بتواند با جستجو یک رکورد یکتا بازگرداند یک ابرکلید است، برای مثال:

(SID): ابرکلید است.

(NID): ابرکلید است.

(SID, Name): ابرکلید است.

(NID, Name): ابرکلید است.

(SID, NID): ابرکلید است.

۳ - حذف صفات زائد

از بین ابرکلیدها، ابرکلیدی که صفت زائد ندارد می‌شود کلید کاندید.

(SID): ابرکلید است. ✓ ، کلید کاندید هم هست. ✓

(NID): ابرکلید است. ✓ ، کلید کاندید هم هست. ✓

(SID, Name): ابرکلید است. ✓ ، کلید کاندید نیست. X ، چون عضو زائد Name دارد.

(NID, Name): ابرکلید است. ✓ ، کلید کاندید نیست. X ، چون عضو زائد Name دارد.

(SID, NID): ابرکلید است. ✓ ، کلید کاندید نیست. X ، **مهم:** چون ترکیب دو کلید کاندید باهم

ابرکلید می‌شود زیرا هر یک برای دیگری عضو زائد محسوب می‌شود.

۴ - تشخیص کلیدهای کاندید

(SID): کلید کاندید است. ✓

(NID): کلید کاندید است. ✓

توجه: کشف کلید کاندید از روی ابرکلید مثل هرس درخت است، انقد هرس می‌کنیم تا عضوهای

زائد حذف شود تا به کلید کاندید خالص برسیم.

رابطه دوطرفه ابرکلید و کلید کاندید

رابطه ابرکلید و کلید کاندید یک رابطه دوطرفه است، یعنی هم از ابرکلید می‌توان به کلید کاندید رسید و هم از کلید کاندید می‌توان به ابرکلید رسید.



ساخت ابرکلید روی کلید کاندید

جهت ساخت ابرکلید روی کلید کاندید، کافیهست به کلید کاندید، عضو زائد (پوشال و چیز میز اضافه) اضافه کنیم. بنابراین همواره هر ابرکلید در دل خودش حتما حداقل یک کلید کاندید دارد، بنابراین ابرکلید یک فوق مجموعه از کلید کاندید است. ابرکلید هر فوق مجموعه‌ای روی یک کلید کاندید است.

جدول تمام کلید

در یک رابطه، تاپل تکراری وجود ندارد. زیرا رابطه یک مجموعه است و مجموعه در ریاضیات طبق تعریف، تاپل تکراری ندارد. بنابراین در جدول نیز، رکورد تکراری وجود ندارد. بنابراین یک جدول در مدل رابطه‌ای همواره حداقل یک کلید کاندید دارد. زیرا در بدترین حالت، تمام صفات با هم یک کلید کاندید می‌شود، که به این رابطه، تمام کلید (All key) گفته می‌شود.

مثال: جدول زیر یک جدول تمام کلید است.

A	B	C
1	2	3
1	6	3
1	2	7
8	2	3

توجه: یک رابطه در مدل رابطه‌ای می‌تواند، بیش از یک کلید کاندید نیز داشته باشد.

مثال:

نام خانوادگی	کد ملی	شماره دانشجویی	نام
	کلید کاندید	کلید کاندید	

بررسی مقدار NULL در کلید کاندید

از آنجا که وظیفه کلید کاندید (اصلی و فرعی) ایجاد تمایز میان سطرهاى مختلف یک جدول است، هیچگاه کلید کاندید (اصلی و فرعی) نمی‌تواند مقدار NULL داشته باشد، چون امید ما برای جستجوی یکتا، کلید کاندید است و امید ما نباید ناامید شود. NULL مقدار نیست بلکه هیچ مقدار است، ستونی که هنگام عملیات درج، مقداری برای آن لحاظ نشده است هیچ مقدار است، NULL یک نشانه برای هیچ مقدار است.

قانون جامعیت موجودیت (Entity Integrity Rule)

مطابق قانون جامعیت موجودیت، هیچگاه نباید تمام یا بخشی از کلید کاندید (اصلی و فرعی) مقدار NULL داشته باشد. برای مثال درج مقدار NULL در ستون S# از جدول S باعث نقض قانون جامعیت موجودیت می‌شود.

قانون حداقل تعداد ابرکلید

حداقل تعداد ابرکلیدهای یک جدول برابر تعداد کلیدهای کاندید همان جدول است. چون هر کلید کاندید خودش یک ابرکلید است.

تعداد کلیدهای کاندید = حداقل تعداد ابرکلیدها

قانون حداکثر تعداد ابرکلید

حداکثر تعداد ابرکلیدهای یک جدول برابر تمام فوق مجموعه‌هایی است که شامل حداقل یک کلید کاندید باشد.

تمام فوق مجموعه‌هایی که شامل حداقل یک کلید کاندید باشد = حداکثر تعداد ابرکلیدها

شماریش تعداد ابرکلید در حالت‌های خاص

حالت اول: در یک رابطه با n خصیصه، اگر فقط و فقط یک کلید کاندید با k ستون داشته باشد، آنگاه رابطه حداقل 1 و حداکثر دارای 2^{n-k} ابرکلید است. در واقع کلید کاندید بعلاوه هر فوق مجموعه‌ای از $n-k$ ستون باقی مانده، ابرکلید می‌سازد. در واقع $n-k$ صفت آزاد دارد.

مثال: در رابطه S با 3 خصیصه، اگر فقط و فقط ستون S# کلید کاندید باشد، آنگاه این رابطه حداقل و حداکثر چند ابرکلید دارد؟

پاسخ:

تعداد کلیدهای کاندید = حداقل تعداد ابرکلیدها

یک کلید کاندید S# دارد، پس حداقل ابرکلیدها = 1

تمام فوق مجموعه‌هایی که شامل حداقل یک کلید کاندید باشد = حداکثر تعداد ابرکلیدها

$$2^{n-k} = 2^{3-1} = 2^2 = 4$$

S#	+	Sname	City	ابر کلید
کلید کاندید	0	0	0 →	(S#)
	0	1	1 →	(S#, City)
	1	0	0 →	(S#, Sname)
	1	1	1 →	(S#, Sname, City)

مثال: در رابطه R با 5 خصیصه، اگر فقط و فقط ستون‌های AB باهم کلید کاندید باشد، آنگاه این رابطه حداقل و حداکثر چند ابرکلید دارد؟

A	B	C	D	E

پاسخ:

تعداد کلیدهای کاندید = حداقل تعداد ابرکلیدها

یک کلید کاندید AB دارد، پس حداقل ابرکلیدها = 1

تمام فوق مجموعه‌هایی که شامل حداقل یک کلید کاندید باشد = حداکثر تعداد ابرکلیدها

$$2^{n-k} = 2^{5-2} = 2^3 = 8$$

تست - فرض کنید R دارای هشت (8) صفت است که ترکیب سه صفت، کلید اصلی (اولیه) را تشکیل می‌دهند. تعداد سوپرکلیدهای رابطه کدام است؟
(مهندسی کامپیوتر - دولتی ۱۴۰۴)

256 (۱) 40 (۲) 32 (۳) 8 (۴)

پاسخ: گزینه (۳) صحیح است.

تعداد کلیدهای کاندید = حداقل تعداد ابرکلیدها

یک کلید کاندید ABC دارد، پس حداقل ابرکلیدها = 1

تمام فوق مجموعه‌هایی که شامل حداقل یک کلید کاندید باشد = حداکثر تعداد ابرکلیدها

$$2^{n-k} = 2^{8-3} = 2^5 = 32$$

حالت دوم: جدول تمام کلید، فقط و فقط یک کلید کاندید دارد، بنابراین **حداقل** و **حداکثر** هم فقط و فقط یک ابرکلید دارد.

مثال: در رابطه R با 5 خصیصه، اگر همه ستون‌های جدول باهم کلید کاندید باشد یعنی جدول تمام کلید باشد، آنگاه این رابطه حداقل و حداکثر چند ابرکلید دارد؟

A	B	C	D	E

پاسخ:

تعداد کلیدهای کاندید = حداقل تعداد ابرکلیدها

یک کلید کاندید ABCDE دارد، پس حداقل ابرکلیدها = 1

تمام فوق مجموعه‌هایی که شامل حداقل یک کلید کاندید باشد = حداکثر تعداد ابرکلیدها

$$2^{n-k} = 2^{5-5} = 2^0 = 1$$

حالت سوم: در یک رابطه با n خصیصه، اگر چندین کلید کاندید مختلف داشته باشد، حداقل و حداکثر تعداد ابرکلیدها مطابق **قاعده شمول و عدم شمول** به صورت زیر محاسبه می‌شود.

قاعده شمول و عدم شمول (Inclusion – Exclusion)

برای شمارش اجتماع مجموعه‌ها به صورت زیر است:

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$$

مثال: فرض کنید سه مجموعه زیر وجود دارد:

$$A = \{x, y\}$$

$$B = \{y, z\}$$

$$C = \{y, w\}$$

می‌خواهیم حساب کنیم $|A \cup B \cup C|$ چقدر است؟

پاسخ: دقت کنید مسئله به صورت مجموعه‌ای (نمودار ون) حل می‌شود.

گام ۱: محاسبه اعضای هر مجموعه مطابق نمودار ون

$$|A| + |B| + |C| = 2 + 2 + 2 = 6$$

گام ۲: کم کردن اشتراک‌های دوتایی مطابق نمودار ون

$$A \cap B = \{x, y\} \cap \{y, z\} = \{y\}$$

$$A \cap C = \{x, y\} \cap \{y, w\} = \{y\}$$

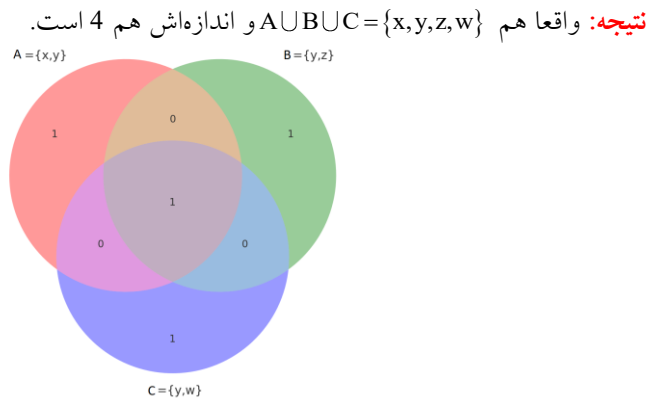
$$B \cap C = \{y, z\} \cap \{y, w\} = \{y\}$$

$$|A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| = 6 - 1 - 1 - 1 = 3$$

گام ۳: اضافه کردن اشتراک سه‌تایی مطابق نمودار ون

$$A \cap B \cap C = \{x, y\} \cap \{y, z\} \cap \{y, w\} = \{y\}$$

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C| = 6 - 3 + 1 = 4$$



توجه: اینجا عنصر y مشکل ساز بود:

در سه مجموعه وجود داشت.

بار اول سه بار شمرده شد.

در مرحله دوتایی سه بار کم شد (بیش از حد). دوبار حذف کافی بود.

مرحله سه تایی یک بار دوباره برگردونده شد و در نهایت آخرش فقط یک بار باقی موند.

مثال: رابطه $R(A, B, C, D, E)$ را در نظر بگیرید، اگر در این رابطه تنها کلیدهای کاندید به صورت زیر باشد:

$$CK1 = \{A, B, C\}$$

$$CK2 = \{D, C\}$$

آنگاه این رابطه حداقل و حداکثر چند ابرکلید دارد؟

پاسخ:

تعداد کلیدهای کاندید = حداقل تعداد ابرکلیدها

دو کلید کاندید ABC و DC دارد، پس حداقل ابرکلیدها = 2

تمام فوق مجموعه هایی که شامل حداقل یک کلید کاندید باشد = حداکثر تعداد ابرکلیدها

گام ۱: محاسبه تعداد ابرکلیدهای هر کلید کاندید $CK1$ و $CK2$ به ترتیب از رابطه 2^{n-k} :

$$|SK1| + |SK2| = 2^{(5-3)} + 2^{(5-2)} = 4 + 8 = 12$$

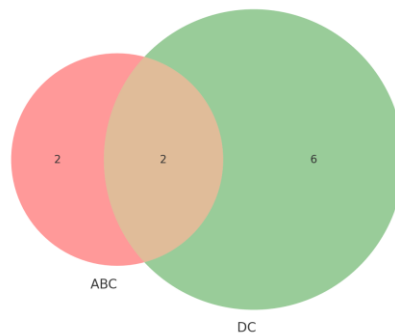
توجه: همه ابرکلیدهای مشتق از هر کلید کاندید شمرده شد.

توجه: مسیر $CK1$ ، 2 صفت آزاد و مسیر $CK2$ ، 3 صفت آزاد دارد.

گام ۲: اجتماع ستون های 2 تا کلید کاندید باهم کم شود، جهت حذف 2 بار تکرار شده ها، از رابطه 2^{n-k} :

توجه: بعضی مجموعه‌ها هم‌زمان شامل هر دو کلید کاندید ABC و DC هستند یعنی شامل A,B,C,D هستند.

- مثلاً ABCD هم شامل ABC هست، هم شامل DC
 - یا ABCDE هم شامل ABC هست، هم شامل DC
 - یعنی این دو مجموعه هم در گروه اول (۴تایی) افتادن، هم در گروه دوم (۸تایی). به همین خاطر دوبار حساب شدن.
- تکرار از آنجا به وجود می‌آید که ABCD و ABCDE هم‌زمان شامل هر دو کلید کاندید هستند و اگر کم نشوند، دوبار شمرده می‌شوند.



توجه: مجموعه مشترک، یکبار با مجموعه سمت چپ، یکبار با مجموعه سمت راست، شمرده شده است. کافی است تعداد مجموعه مشترک یکبار کسر شود تا شمارش به تعادل درست برسد.

توجه: ترکیب کلیدهای کاندید باهم، ابرکلید می‌سازد.

$$CK1 \cup CK2 = \{A,B,C,D\} \rightarrow 2^{(5-4)} = 2$$

توجه: اجتماع صفات کلید کاندیدهای بالا را با **اشتراک** قاعده شمول و عدم شمول اشتباه نگیرید، جهت جلوگیری از خطای ذهنی، مجموعه‌ای و روی نمودار ون فکر کنید. رابطه فوق تعداد اعضا در مجموعه مشترک بین دو مجموعه را محاسبه می‌کند.

$$|SK1 \cap SK2| = 2$$

$$\rightarrow (12) - (2) = 10$$

نتیجه نهایی:

راه حل عمومی: برای به دست آوردن ابرکلیدها، باید به هر یک از دو کلید کاندید فوق، صفر یا بیش‌تر از صفت‌های دیگر را اضافه کرد. بنابراین باید به کلید کاندید ABC، صفر، یک یا دو صفت از بین صفت‌های دیگر (D,E) اضافه کرد. برای این کار 4 حالت وجود دارد:

$$\binom{2}{0} + \binom{2}{1} + \binom{2}{2} = 2^2 = 4$$

ترکیب اول: هیچ یا ترکیبی از صفات باقیمانده + کلید کاندید ABC = ابرکلید

A B C +	E D	ترکیبات صفات باقی مانده	ابرکلیدها
	0 0	تهی	A B C
	0 1	D	A B C D
	1 0	E	A B C E
	1 1	E D	A B C E D

توجه: واضح است که 4 ابرکلید، ایجاد می‌گردد. (2^2)

به طور مشابه، باید به کلید کاندید DC، صفر، یک، دو یا سه مورد از صفتهای دیگر (A,B,E) اضافه کرد. برای این کار 8 حالت، وجود دارد:

$$\binom{3}{0} + \binom{3}{1} + \binom{3}{2} + \binom{3}{3} = 2^3 = 8$$

ترکیب دوم: هیچ یا ترکیبی از صفات باقی مانده + کلید کاندید DC = ابرکلید

D C +	A B E	ترکیبات صفات باقی مانده	ابرکلیدها
	0 0 0	تهی	D C
	0 0 1	E	D C E
	0 1 0	B	D C B
	0 1 1	B E	D C B E
	1 0 0	A	D C A
	1 0 1	A E	D C A E
	1 1 0	A B	D C A B
	1 1 1	A B E	D C A B E

توجه: واضح است که 8 ابرکلید، ایجاد می‌گردد. (2^3)

بنابراین حاصل جمع ابرکلیدها 12 عدد خواهد بود، که از این مجموعه حاصل، ابرکلیدهای ABCDE و ABCD در دو مجموعه تکراری هستند، بنابراین با کنار گذاشتن ابرکلیدهای تکراری، در نهایت 10 ابرکلید خواهیم داشت.

مثال: رابطه $R(A,B,C,D,E,F)$ را در نظر بگیرید، اگر در این رابطه تنها کلیدهای کاندید به صورت زیر باشد:

$$CK1 = \{A\}$$

$$CK2 = \{B,C\}$$

آنگاه این رابطه حداقل و حداکثر چند ابرکلید دارد؟

پاسخ:

تعداد کلیدهای کاندید = حداقل تعداد ابرکلیدها

سه کلید کاندید A و BC دارد، پس حداقل ابرکلیدها = 2

تمام فوق مجموعه‌هایی که شامل حداقل یک کلید کاندید باشد = حداکثر تعداد ابرکلیدها

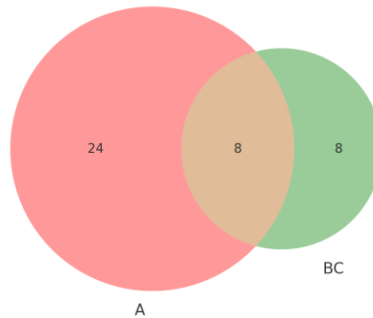
گام ۱: محاسبه تعداد ابرکلیدهای هر کلید کاندید CK1 و CK2 به ترتیب از رابطه 2^{n-k} :

$$|SK1| + |SK2| = 2^{(6-1)} + 2^{(6-2)} = 32 + 16 = 48$$

توجه: همه ابرکلیدهای مشتق از هر کلید کاندید شمرده شد.

توجه: مسیر CK1، 5 صفت آزاد و مسیر CK2، 4 صفت آزاد دارد.

گام ۲: اجتماع ستون‌های 2 تا کلید کاندید باهم کم شود، جهت حذف 2 بار تکرار شده‌ها، از رابطه 2^{n-k} :



توجه: ترکیب کلیدهای کاندید باهم، ابرکلید می‌سازد.

$$CK1 \cup CK2 = \{A, B, C\} \rightarrow 2^{(6-3)} = 8$$

$$|SK1 \cap SK2| = 8$$

$$\rightarrow (48) - (8) = 40$$

نتیجه نهایی:

مثال: رابطه $R(A, B, C, D, E)$ را در نظر بگیرید، اگر در این رابطه تنها کلیدهای کاندید به صورت زیر باشد:

$$CK1 = \{A, D, E\}$$

$$CK2 = \{A, D, C\}$$

$$CK3 = \{A, D, B\}$$

آنگاه این رابطه حداقل و حداکثر چند ابرکلید دارد؟

پاسخ:

تعداد کلیدهای کاندید = حداقل تعداد ابرکلیدها

سه کلید کاندید ADE، ADC و ADB دارد، پس حداقل ابرکلیدها = 3

تمام فوق مجموعه‌هایی که شامل حداقل یک کلید کاندید باشد = حداکثر تعداد ابرکلیدها

گام ۱: محاسبه تعداد ابرکلیدهای هر کلید کاندید CK1، CK2 و CK3 به ترتیب از رابطه 2^{n-k} :
 $|SK1| + |SK2| + |SK3| = 2^{(5-3)} + 2^{(5-3)} + 2^{(5-3)} = 4 + 4 + 4 = 12$

توجه: همه ابرکلیدهای مشتق از هر کلید کاندید شمرده شد.

توجه: مسیر CK1، 2 صفت آزاد، مسیر CK2، 2 صفت آزاد و مسیر CK3، 2 صفت آزاد دارد.

گام ۲: اجتماع ستون‌های 2 تا کلید کاندید باهم کم شود، جهت حذف 2 بار تکرار شده‌ها، از رابطه 2^{n-k} :

توجه: ترکیب کلیدهای کاندید باهم، ابرکلید می‌سازد.

$$CK1 \cup CK2 = \{A, D, E, C\} \rightarrow 2^{(5-4)} = 2$$

$$|SK1 \cap SK2| = 2$$

$$CK1 \cup CK3 = \{A, D, E, B\} \rightarrow 2^{(5-4)} = 2$$

$$|SK1 \cap SK3| = 2$$

$$CK2 \cup CK3 = \{A, D, C, B\} \rightarrow 2^{(5-4)} = 2$$

$$|SK2 \cap SK3| = 2$$

$$\rightarrow (2 + 2 + 2) = 6$$

گام ۳: اجتماع ستون‌های 3 تا کلید کاندید باهم اضافه شود، جهت درج مجدد 3 بار حذف شده، از رابطه 2^{n-k} :



$$CK1 \cup CK2 \cup CK3 = \{A, B, C, D, E\} \rightarrow 2^{(5-5)} = 1$$

$$|SK1 \cap SK2 \cap SK3| = 1$$

$$\rightarrow (12) - (6) + (1) = 7$$

نتیجه نهایی:

$$\begin{aligned} |SK1 \cup SK2 \cup SK3| &= \\ |SK1| + |SK2| + |SK3| - |SK1 \cap SK2| - |SK1 \cap SK3| - |SK2 \cap SK3| + |SK1 \cap SK2 \cap SK3| &= \\ 4 + 4 + 4 - 2 - 2 - 2 + 1 &= 7 \end{aligned}$$

اشتراک سه تایی:

فقط یک مجموعه مشترک بین هر سه وجود دارد (دقیقا در وسط نمودار): $\{A, B, C, D, E\}$ ابرکلید $\{A, B, C, D, E\}$ هم در اشتراک اول بود، هم در اشتراک دوم، هم در اشتراک سوم! یعنی 3 بار شمرده شده بود، اما 3 بار هم حذف شد! درحالیکه باید 2 بار حذف می‌شد، پس 1 عدد دوباره اضافه شد. برای جبران، یکبار دوباره برش می‌گردونیم. اشتراک‌های دوتایی دوبار تکرار شده بودن یکبار هم حذف شدن که تعادل دارن.

مراحل محاسبه دستی قاعده شمول و عدم شمول

$$\begin{aligned} &+ \binom{1}{n} \text{ ترکیب انتخاب یک کلیدکاندید از کل کلیدهای کاندید} \\ &- \binom{2}{n} \text{ ترکیب انتخاب 2 کلیدکاندید از کل کلیدهای کاندید} \\ &\text{مثال: } \{(CK1, CK2), (CK1, CK3), (CK2, CK3)\} \\ &+ \binom{3}{n} \text{ ترکیب انتخاب 3 کلیدکاندید از کل کلیدهای کاندید} \\ &- \binom{4}{n} \text{ ترکیب انتخاب 4 کلیدکاندید از کل کلیدهای کاندید} \\ &\dots \end{aligned}$$

علامت‌ها به همین ترتیب به صورت یکی در میان +، -، +، - ... تا بی‌نهایت

تست: جدولی با n ستون دارای یک کلید اصلی و یک کلید کاندید دیگر هر کدام شامل یک صفت می‌باشد. این جدول چند ابرکلید (Super Key) دارد؟

(مهندسی کامپیوتر- دولتی ۸۴)

$$(1) \ 3 \times 2^{n-2} \quad (2) \ 2^{n-2} \quad (3) \ 3 \times 2^{n-3} \quad (4) \ 2^{n-1}$$

پاسخ: گزینه (۱) صحیح است.

مطابق صورت سوال، کلیدهای کاندید 2 تا و هر کدام تک خصیصه یعنی 1 صفت دارد.

تعداد کلیدهای کاندید = حداقل تعداد ابرکلیدها

دو کلید کاندید یکی اصلی و کلید کاندید دیگری دارد، پس حداقل ابرکلیدها = 2

تمام فوق مجموعه‌هایی که شامل حداقل یک کلید کاندید باشد = حداکثر تعداد ابرکلیدها

گام ۱: محاسبه تعداد ابرکلیدهای هر کلید کاندید CK1 و CK2 به ترتیب از رابطه 2^{n-k} :

$$2^{(n-1)} + 2^{(n-1)} = 2 \times 2^{(n-1)} = 2^1 \times 2^{(n-1)} = 2^n$$

گام ۲: اجتماع ستون‌های 2 تا کلید کاندید باهم کم شود، جهت **حذف** تکراری‌ها، از رابطه 2^{n-k} :
توجه: ترکیب کلیدهای کاندید باهم، ابرکلید می‌سازد.

$$CK1 \cup CK2 = \{X, Y\} \rightarrow 2^{(n-2)}$$

نتیجه نهایی:

$$\rightarrow [2^n] - [2^{(n-2)}] = [2^2 \times 2^{n-2}] - [2^{(n-2)}] = (4-1) \times 2^{(n-2)} = 3 \times 2^{(n-2)}$$

پادزنجیر یکنواخت (Uniform Antichain)

مجموعه‌ای از زیرمجموعه‌ها که همگی **هم‌اندازه‌اند** (کتابی) و هیچ عضوی **داخل دیگری** نیست.

مثال: روی $\{1, 2, 3, 4\}$ مجموعه $\{\{1\}, \{2\}, \{3\}, \{4\}\}$ یک پادزنجیر یکنواخت 1 تایی است.

مثال: روی $\{1, 2, 3, 4\}$ مجموعه $\{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}\}$ یک پادزنجیر یکنواخت 2 تایی است.

مثال: روی $\{1, 2, 3, 4\}$ مجموعه $\{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}\}$ یک پادزنجیر یکنواخت 3 تایی است.

حداقل و حداکثر تعداد ابرکلید پادزنجیر یکنواخت 1 تایی

در یک رابطه با n خصیصه، اگر همه 1 تایی‌ها (تک تک خصیصه‌ها) **کلید کاندید** باشد، آنگاه هر فوق مجموعه‌ای با اندازه‌ی $1 \leq$ حتماً **ابرکلید** است، چون حتماً حداقل یک 1 تایی را به عنوان زیرمجموعه در خود دارد. در این شرایط، تک تک خصیصه‌ها به تنهایی **کلید کاندید** هستند، پس رابطه دارای n **کلید کاندید** است و هر فوق مجموعه از تک خصیصه‌های این رابطه هم یک **ابرکلید** است. که در این حالت تعداد ابرکلیدهای یک رابطه با n خصیصه **حداقل** n و **حداکثر** برابر با $2^n - \binom{n}{0}$ یا $2^n - 1$ است به صورت زیر:

$$2^n - \binom{n}{0} = 2^n - 1 = \binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{n}$$

تست: در یک رابطه دارای 5 خصیصه، اگر همه 1 تایی‌ها به صورت پادزنجیر یکنواخت کلید کاندید باشد، آنگاه این رابطه **حداقل** و **حداکثر** چند ابرکلید دارد؟

(۲) 5 و 31

(۱) 1 و 32

(۴) 5 و 27

(۳) 1 و 25

پاسخ: گزینه (۲) صحیح است.

تعداد کلیدهای کاندید = حداقل تعداد ابرکلیدها

تعداد ۵ یا $\binom{5}{1}$ کلید کاندید دارد، پس حداقل ابرکلیدها = ۵

تمام فوق مجموعه‌هایی که شامل حداقل یک کلید کاندید باشد = حداکثر تعداد ابرکلیدها

$$\binom{5}{1} + \binom{5}{2} + \binom{5}{3} + \binom{5}{4} + \binom{5}{5} = 5 + 10 + 10 + 5 + 1 = 31$$

$$2^5 - \binom{5}{0} = 32 - 1 = 31$$

حداقل و حداکثر تعداد ابرکلید پادزنجیر یکنواخت ۲تایی

در یک رابطه با n خصیصه، اگر همه ۲تایی‌ها (جفت خصیصه‌ها) **کلید کاندید** باشد، آنگاه هر فوق مجموعه‌ای با اندازه‌ی $2 \leq$ حتماً **ابرکلید** است، چون حتماً حداقل یک ۲تایی (جفت) را به‌عنوان زیرمجموعه در خود دارد. در این شرایط، جفت خصیصه‌ها به تنهایی **کلید کاندید** هستند، پس رابطه دارای $\binom{n}{2}$ **کلید کاندید** است و هر فوق مجموعه از جفت خصیصه‌های این رابطه هم یک **ابرکلید** است. که در این حالت تعداد ابرکلیدهای یک رابطه با n خصیصه **حداقل** $\binom{n}{2}$ و **حداکثر** برابر با $2^n - \binom{n}{0} - \binom{n}{1}$ یا $2^n - 1 - n$ است به صورت زیر:

$$2^n - \binom{n}{0} - \binom{n}{1} = 2^n - 1 - n = \binom{n}{2} + \binom{n}{3} + \dots + \binom{n}{n}$$

تست: در یک رابطه دارای ۵ خصیصه، اگر همه ۲تایی‌ها به صورت پادزنجیر یکنواخت **کلید کاندید** باشد، آنگاه این رابطه **حداقل** و **حداکثر** چند ابرکلید دارد؟

(۲) ۱ و ۲۶

(۱) ۱۰ و ۲۶

(۴) ۵ و ۲۷

(۳) ۱ و ۲۵

پاسخ: گزینه (۱) صحیح است.

تعداد کلیدهای کاندید = حداقل تعداد ابرکلیدها

تعداد ۱۰ یا $\binom{5}{2}$ کلید کاندید دارد، پس حداقل ابرکلیدها = ۱۰

تمام فوق مجموعه‌هایی که شامل حداقل یک کلید کاندید باشد = حداکثر تعداد ابرکلیدها

$$\binom{5}{2} + \binom{5}{3} + \binom{5}{4} + \binom{5}{5} = 10 + 10 + 5 + 1 = 26$$

$$2^5 - \binom{5}{0} - \binom{5}{1} = 32 - 1 - 5 = 32 - 6 = 26$$

حداقل و حداکثر تعداد ابرکلید پادزنجیر یکنواخت ktایی

حداقل و حداکثر تعداد ابرکلید پادزنجیر یکنواخت ktایی به همین ترتیب ادامه دارد.

تست: در یک رابطه دارای 5 خصیصه، اگر همه 3تایی‌ها به صورت پادزنجیر یکنواخت کلید کاندید باشد، آنگاه این رابطه حداقل و حداکثر چند ابرکلید دارد؟

- (۱) 10 و 16 (۲) 10 و 17 (۳) 1 و 15 (۴) 5 و 11

پاسخ: گزینه (۱) صحیح است.

تعداد کلیدهای کاندید = حداقل تعداد ابرکلیدها

تعداد 10 یا $\binom{5}{3}$ کلید کاندید دارد، پس حداقل ابرکلیدها = 10

تمام فوق مجموعه‌هایی که شامل حداقل یک کلید کاندید باشد = حداکثر تعداد ابرکلیدها

$$\binom{5}{3} + \binom{5}{4} + \binom{5}{5} = 10 + 5 + 1 = 16$$

$$2^5 - \binom{5}{0} - \binom{5}{1} - \binom{5}{2} = 32 - 1 - 5 - 10 = 32 - 16 = 16$$

نکات ابرکلید و کلید کاندید

توجه: اگر CK1 و CK2 کلیدهای کاندید رابطه R باشند، آنگاه $CK1 \cup CK2$ دیگر کلید کاندید نیست و فقط یک ابرکلید است، چون هر کلید کاندید به تنهایی کلید کاندید است و به کمک کسی نیاز ندارد.

توجه: اگر SK1 و SK2 ابرکلیدهای رابطه R باشند، آنگاه $SK1 \cap SK2$ گاهی کلید کاندید و گاهی هم تهی است.

مثال: جدول (stud (nano, stno, avg) را در نظر بگیرید. nano (کد ملی) یک کلید کاندید و به تبع یک ابرکلید است. همچنین stno (شماره دانشجویی) یک کلید کاندید و به تبع یک ابرکلید دیگر است. اجتماع این دو می‌شود: (nano, stno) که هر چند ابرکلید است ولی کلید کاندید نیست. توجه کنید که کلید کاندید باید کمینه باشد و عضو زائد نداشته باشد، هم‌چنین اشتراک این دو برابر مجموعه تهی است که نمی‌تواند نه ابرکلید و نه کلید کاندید باشد. به صورت زیر:

$$\{nano\} \cap \{stno\} = \{\}$$

اما اگر ابرکلیدهای SK1 و SK2 به ترتیب nano, stno و nano, avg باشد، آنگاه $SK1 \cap SK2$ یک کلید کاندید به صورت زیر خواهد بود:

$$\{nano, stno\} \cap \{nano, avg\} = \{nano\}$$

تست: با توجه به دو گزاره زیر کدام یک از گزینه‌های زیر صحیح است؟
(مهندسی IT - دولتی ۹۶)

(۱) اگر $K1$ و $K2$ کلیدهای کاندید در رابطه r باشند، آن‌گاه $K1 \cup K2$ نیز یک کلید کاندید در رابطه r است.

(۲) اگر $K1$ و $K2$ ابرکلیدهای (super key) در رابطه r باشند، آن‌گاه $K1 \cap K2$ نیز یک کلید کاندید در رابطه r است.

(۱) هر دو گزاره صحیح هستند.

(۲) هر دو گزاره غلط هستند.

(۳) تنها گزاره دوم صحیح است.

(۴) تنها گزاره اول صحیح است.

پاسخ: گزینه (۲) صحیح است.

جدول معلوم

در جدول معلوم، کلیدهای کاندید جدول، مشخص و تعیین شده‌است، در جدول معلوم بحث حداقل و حداکثر در مورد ابرکلید صادق است، اما بحث حداقل و حداکثر در مورد کلید کاندید مطرح نیست چون یک مقدار مشخص دارد. جداول معلوم در مثال‌های قبلی بررسی شد.

جدول مجهول

در جدول مجهول، کلیدهای کاندید جدول، مشخص و تعیین نشده‌است. در جدول مجهول بحث حداقل و حداکثر هم در مورد ابرکلید و هم در مورد کلید کاندید صادق است.

توجه: در جدول مجهول، کمترین مقدار ممکن در تعداد ابرکلیدهای یک رابطه با n خصیصه زمانی رخ می‌دهد که همه خصیصه‌ها باهم کلید کاندید باشد، جدول تمام کلید باشد.

توجه: در جدول مجهول، بیشترین مقدار ممکن در تعداد ابرکلیدهای یک رابطه با n خصیصه زمانی رخ می‌دهد که همه 1 تایی‌ها (تک تک خصیصه‌ها) کلید کاندید باشد.

تست: در یک رابطه دارای n خصیصه، حداقل و حداکثر تعداد ابرکلیدها به ترتیب کدام است؟

$$(۱) \quad n \text{ و } 2^{n-1} \quad (۲) \quad n \text{ و } 2^n - 1$$

$$(۳) \quad 1 \text{ و } 2^n - 1 \quad (۴) \quad n \text{ و } 2^{n-1}$$

پاسخ: گزینه (۳) صحیح است. به صورت سوال دقت کنید، در این سوال صحبتی از شرایط و تعداد کلید کاندید نشده‌است. بنابراین جدول مجهول است. حال اگر n خصیصه داشته باشیم در حداقل ممکن چند ابرکلید داریم؟ پاسخ می‌شود زمانی که جدول تمام کلید باشد. و در حداکثر ممکن چند ابرکلید داریم؟ پاسخ می‌شود زمانی که همه 1 تایی‌ها (تک تک خصیصه‌ها) کلید کاندید باشد.

تعداد کلیدهای کاندید = حداقل تعداد ابرکلیدها

یک کلید کاندید تمام کلید دارد، پس حداقل ابرکلیدها = 1

تمام فوق مجموعه‌هایی که شامل حداقل یک کلید کاندید باشد = حداکثر تعداد ابرکلیدها

$$2^n - \binom{n}{0} = 2^n - 1 = \binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{n}$$

حداقل و حداکثر تعداد کلید کاندید پادزنجر یکنواخت

همانطور که گفتیم، به مجموعه‌ای از زیرمجموعه‌ها که همگی **هم‌اندازه‌اند** (k تایی) و هیچ عضوی **داخل دیگری** نیست، پادزنجر یکنواخت گفته می‌شود.

قضیه اسپرنر (Sperner's Theorem)

در یک مجموعه‌ی n-عضوی (برای مثال $\{1, 2, \dots, n\}$)، اگر بخواهیم مطابق قاعده پادزنجر یکنواخت، **حداکثر** تعداد زیرمجموعه را برداریم طوری که هیچ عضوی **داخل دیگری** نباشد، **حداکثر (بیشترین)** تعداد ممکن برابر است با:

$$\max\left(\binom{n}{0}, \binom{n}{1}, \binom{n}{2}, \dots, \binom{n}{n}\right) = \binom{n}{\lfloor n/2 \rfloor}$$

توجه: در **جدول مجهول** مطابق قاعده پادزنجر یکنواخت و قضیه اسپرنر، **حداکثر** مقدار ممکن در **تعداد کلیدهای کاندید یک رابطه** با n خصیصه زمانی رخ می‌دهد که $\binom{n}{\lfloor n/2 \rfloor}$ باشد.

مثال: اگر $n = 2$

$$\max\left(\binom{2}{0}, \binom{2}{1}, \binom{2}{2}\right) = \binom{2}{1}$$

مثال: اگر $n = 5$

$$\max\left(\binom{5}{0}, \binom{5}{1}, \binom{5}{2}, \binom{5}{3}, \binom{5}{4}, \binom{5}{5}\right) = \binom{5}{2}$$

مثال: اگر $n = 6$

$$\max\left(\binom{6}{0}, \binom{6}{1}, \binom{6}{2}, \binom{6}{3}, \binom{6}{4}, \binom{6}{5}, \binom{6}{6}\right) = \binom{6}{3}$$

هر کدام از ترکیب‌های فوق مثل $\binom{6}{1}$ یا $\binom{6}{2}$ یا $\binom{6}{3}$ تا $\binom{6}{6}$ می‌تواند تعداد کلیدهای کاندید باشد، اما **حداکثر** تعداد آن ترکیب $\binom{6}{3}$ یعنی **بخش میانه** است. دقت کنید که نباید ترکیب‌ها را جمع کنید، حاصل جمع ترکیب‌ها، تعداد ابرکلیدها را می‌شمارد.

سوال مهم: چرا قاعده **پادزنجر یکنواخت** و **قضیه اسپرنر** در شمارش **حداکثر** تعداد **کلید کاندید** استفاده می‌شود؟ چون در ساخت و سپس شمارش کلید کاندید، کلیدهای کاندید نباید شامل هم باشند، به عبارت دیگر نباید هیچ عضوی **داخل دیگری** باشد.

توجه بسیار مهم: کلیدهای کاندید، یک پادزنجیرند؛ هیچ‌کدام زیرمجموعه دیگری نیست.

تست: یک رابطه دارای 5 خصیصه، حداقل و حداکثر چند کلید کاندید دارد؟

(۲) 1 و 10

(۱) 1 و 9

(۴) 5 و 11

(۳) 1 و 8

پاسخ: گزینه (۲) صحیح است. به صورت سوال دقت کنید، در این سوال صحبتی از شرایط و تعداد کلید کاندید نشده است. بنابراین جدول مجهول است. حال اگر n خصیصه داشته باشیم در حداقل ممکن چند کلید کاندید داریم؟ پاسخ می‌شود زمانی که جدول تمام کلید باشد. و در

حداکثر ممکن چند کلید کاندید داریم؟ پاسخ می‌شود $\binom{n}{\lfloor n/2 \rfloor}$

$1 = \text{حداقل تعداد کلید کاندید}$

یک کلید کاندید تمام کلید دارد، پس حداقل کلید کاندید $= 1$

$\text{حداکثر تعداد کلید کاندید} = \binom{n}{\lfloor n/2 \rfloor}$

$$\max\left(\binom{n}{0}, \binom{n}{1}, \binom{n}{2}, \dots, \binom{n}{n}\right) = \binom{n}{\lfloor n/2 \rfloor} = \binom{5}{\lfloor 5/2 \rfloor} = \binom{5}{2} = 10$$

تست - با توجه به گزاره‌های زیر، کدام مورد درست است؟ **(مهندسی کامپیوتر-دولتی ۹۷)**

(a) اگر رابطه R ، دارای $n=2k$ خصیصه باشد، آنگاه تعداد کلیدهای کاندید آن حداقل یک و حداکثر $\binom{n}{k}$ است.

(b) اگر رابطه R ، دارای n خصیصه باشد، آنگاه تعداد ابرکلیدهای این رابطه حداکثر $\binom{n}{0} + \binom{n}{1} + \dots + \binom{n}{n}$ است.

(c) تعداد ابرکلیدهای یک رابطه همواره بیشتر از تعداد کلیدهای کاندید آن رابطه است.

(d) هر صفت مرکب، لزوماً تک مقداری است.

(۱) یک گزاره نادرست است. (۲) دو گزاره نادرست است.

(۳) سه گزاره نادرست است. (۴) چهار گزاره نادرست است.

پاسخ: گزینه (۳) صحیح است. فقط گزاره a درست است. در گزاره a ، صحبتی از شرایط و تعداد کلید کاندید نشده است. بنابراین جدول مجهول است. حال اگر n خصیصه داشته باشیم در

حداقل ممکن چند کلید کاندید داریم؟ پاسخ می شود زمانی که جدول تمام کلید باشد. و در حداکثر ممکن چند کلید کاندید داریم؟ پاسخ می شود $\binom{n}{\lfloor n/2 \rfloor}$

کلید اصلی (Primary key)

کلید اصلی، یکی از کلیدهای کاندید است که بر اساس **تشخیص** طراح پایگاه داده و مناسب محیط عملیاتی، انتخاب می شود.

مثال: کدملی و شماره دانشجویی هر دو کلید کاندید است، اما در محیط عملیاتی دانشگاه، بهتر است شماره دانشجویی به عنوان کلید اصلی انتخاب می شود، هر چند کدملی هم می تواند به عنوان کلید اصلی انتخاب شود.

توجه: اگر چندین کلید کاندید در یک جدول وجود داشته باشد، در انتخاب کلید اصلی اولویت با کلید کاندیدی است که کمترین تعداد صفت را دارد، البته رعایت این اولویت الزام ندارد. **توجه:** مطابق قوانین مدل رابطه ای، هر جدول همواره فقط و فقط یک کلید اصلی دارد.

$1 = \text{تعداد کلید اصلی}$

بررسی مقدار NULL در کلید اصلی

از آنجا که وظیفه کلید کاندید (اصلی و فرعی) ایجاد تمایز میان سطرهاى مختلف یک جدول است، هیچگاه کلید کاندید (اصلی و فرعی) نمی تواند مقدار NULL داشته باشد، چون امید ما برای جستجوی یکتا کلید کاندید است و امید ما نباید ناامید شود. NULL مقدار نیست بلکه هیچ مقدار است، ستونی که هنگام درج، مقداری برای آن لحاظ نشده است هیچ مقدار NULL است، NULL یک **نشانه** برای هیچ مقدار است.

قانون جامعیت موجودیت (Entity Integrity Rule)

مطابق قانون جامعیت موجودیت، هیچگاه نباید تمام یا بخشی از کلید کاندید (اصلی و فرعی) مقدار NULL داشته باشد. برای مثال درج مقدار NULL در ستون S# از جدول S باعث نقض قانون جامعیت موجودیت می شود.

تست - برای تعیین کلید اصلی یک رابطه توجه به کدام یک از موارد زیر ضروری است؟

(مهندسی کامپیوتر- دولتی ۹۰)

- (۱) معنی خصیصه های رابطه در دنیای خارج
 - (۲) عملیات مورد نیاز روی رابطه
 - (۳) مقادیر ابرکلیدهای رابطه
 - (۴) مقادیر خصیصه های رابطه در زمان بار کردن رابطه
- پاسخ:** گزینه (۱) صحیح است. کلید در دنیای واقعی از جنس **معنا** است و نه مقدار، بنابراین در پایگاه داده نیز بر اساس همان معنا انتخاب می شود و نه مقدار.

کلید فرعی یا بدیل یا جایگزین یا ثانویه (Alternative key)

هر کلید کاندیدی که به عنوان کلید اصلی انتخاب نشده است، یک کلید فرعی است. کاربرد کلید فرعی در مباحث شاخص گذاری اطلاعات است که در فصل مربوطه به طور مفصل تشریح خواهد شد.

1 - تعداد کلید کاندید = تعداد کلید فرعی

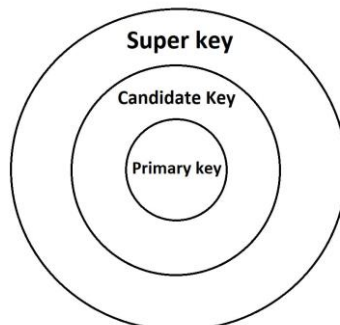
توجه: منظور از مقدار 1 در فرمول فوق، تعداد کلید اصلی هست که هر جدول همواره فقط و فقط یک کلید اصلی دارد.

مثال:

نام	نام خانوادگی	کد ملی	شماره دانشجویی
		کلید کاندید (کلید فرعی)	کلید کاندید (کلید اصلی)

توجه: جدول فوق، 2 کلید کاندید، 1 کلید اصلی و 1 کلید فرعی دارد.

در جمع بندی نهایی انواع کلید، شکل زیر گویای مطلب است:



توجه: هر کلید کاندید قطعا یک ابرکلید است. اما هر ابرکلید الزاما کلید کاندید نیست مگر اینکه **کمینه** باشد، یعنی عضو زائد (چیز میز اضافه) نداشته باشد.

توجه: هر جدول همواره فقط و فقط یک کلید اصلی دارد، هر کلید اصلی قطعا یک کلید کاندید و ابرکلید است.

توجه: ابرکلید، یا برابر یک کلید کاندید است یا یک فوق مجموعه کلید کاندید است.

تست - کلید بدیل (Alternate Key) در یک پایگاه داده، چیست؟

(مهندسی IT - دولتی ۱۴۰۳)

۱) یک کلید اصلی، که در چند جدول مختلف تکرار می شود.

- ۲) یک کلید خارجی (Foreign Key)، که برای ایجاد ارتباط بین دو جدول استفاده می‌شود.
- ۳) کلیدی که برای شناسایی یکتای رکوردها در یک جدول استفاده می‌شود و جایگزین کلید اصلی (Primary Key) است.
- ۴) هر کلید که می‌تواند رکوردها را به‌صورت یکتا شناسایی کند، اما به‌عنوان کلید اصلی انتخاب نشده است.

پاسخ: گزینه (۳) صحیح است.

سازمان سنجش آموزش کشور، ابتدا در کلید اولیه خود گزینه چهارم را به عنوان پاسخ اعلام نمود، اما در کلید نهایی گزینه‌ی سوم را به عنوان پاسخ نهایی اعلام نمود، که کار درستی بوده است.

گزینه چهارم پاسخ سوال نیست. زیرا، عبارت «هر کلید که می‌تواند رکوردها را به‌صورت یکتا شناسایی کند، اما به‌عنوان کلید اصلی انتخاب نشده است» شامل **ابرکلید** هم می‌شود، چون هم یکتا است و هم به عنوان کلید اصلی انتخاب نمی‌شود. درحالی که **کلید بدیل** باید یکی از کلیدهای کاندید باشد.

گزینه سوم پاسخ سوال است. زیرا، عبارت «کلیدی که برای شناسایی یکتای رکوردها در یک جدول استفاده می‌شود و جایگزین کلید اصلی (Primary Key) است.» مطابق تعریف **کلید بدیل** است. عبارت جایگزین کلید اصلی یعنی هم تراز آن خاصیت **کلید کاندید** بودن را دارد.

کلید خارجی (Foreign key)

کلید خارجی ارتباط میان ستون‌های دو جدول را برقرار می‌کند. کلید خارجی از نظر واژه **کلید**، واژه مشترک مثل ابرکلید، کلید کاندید، کلید اصلی و کلید فرعی دارد، اما کاربرد آن مثل آنها نیست، بنابراین می‌تواند مقدار تکراری داشته باشد، در واقع کلید خارجی کاربرد جستجوی یکتا و ایجاد تمایز یکتا میان سطرها را ندارد. بلکه کاربرد آن صرفاً برقراری ارتباط، میان ستون‌های دو جدول مرتبط است. کلید خارجی، الزام خاصیت کلیدی ندارد چون کاربردش جستجوی یکتا نیست بلکه کاربردش صرفاً برقراری ارتباط، میان دو جدول مرتبط است.

توجه: ابرکلید، کلید کاندید، کلید اصلی و کلید فرعی الزام خاصیت کلیدی داشتن، یکتا بودن و مقدار تکراری نداشتن را دارد، اما کلید خارجی این الزام را ندارد، یعنی می‌تواند مقدار تکراری داشته باشد، چون کاربرد آن موضوع دیگری است.

توجه: کاربرد کلید خارجی در مباحث نرمال‌سازی است که در فصل مربوطه به طور مفصل تشریح می‌شود.

قانون جامعیت ارجاعی (Referential Integrity Rule)

مطابق قانون جامعیت ارجاعی، هیچگاه نباید کلید خارجی، دچار ارجاع NULL گردد. برای مثال درج مقدار S4 در ستون S# از جدول SP بدون وجود مقدار S4 در ستون S# از جدول S باعث نقض قانون جامعیت ارجاعی می‌شود. کلید خارجی در دو دیگه‌گاه **ساختاری** و **محتوایی** مورد بررسی قرار می‌گیرد:

دیدگاه ساختاری کلید خارجی

تعریف: اگر صفت(هایی) در یک جدول به عنوان کلید خارجی تعریف شود،
اول اینک: این صفت(ها) در جدول خودش شرط خاصی ندارد یعنی الزام خاصیت کلیدی ندارد.
دوم اینک: این صفت(ها) در همان جدول یا جدول دیگری، باید کلید کاندید (اصلی یا فرعی) باشد.

مثال:

کلید کاندید			کلید خارجی			کلید خارجی			کلید کاندید		
↓			↓			↓			↓		
S#	Sname	City	S#	P#	QTY	P#	Pname	Color			
S ₁	Sn ₁	C ₁	S ₁	P ₁	10	P ₁	Pn ₁	Red			
S ₂	Sn ₂	C ₂	S ₁	P ₂	20	P ₂	Pn ₂	Blue			
S ₃	Sn ₃	C ₂	S ₂	P ₁	30	P ₃	Pn ₃	Blue			
جدول S			جدول SP			جدول P					

توجه: S# در جدول اصلی S، کلید کاندید است، پس عناصرش یکتاست.
توجه: S# در جدول فرعی SP، به عنوان کلید خارجی، الزام خاصیت کلیدی ندارد، پس ممکن است، مقادیر تکراری داشته باشد.

مثال:

نام خانوادگی	نام	شماره ملی	شماره ملی	شماره دانشجویی
		کلید خارجی	کلید فرعی	کلید اصلی
جدول R ₂		جدول R ₁		

توجه: در مثال فوق، از آنجا که شماره ملی در جدول فرعی R₂، به طور ذاتی خاصیت کلیدی دارد، بنابراین کلید خارجی، خاصیت کلیدی دارد، اما الزامی بر این شرط نبوده است.

دیدگاه محتوایی کلید خارجی

به ازای هر مقدار موجود در یک کلید خارجی (جدول فرعی)، باید دقیقاً یک مقدار متناظر در کلید کاندید (جدول اصلی) متناظر آن وجود داشته باشد، در غیر این صورت می‌گوییم، کلید خارجی دارای ارجاع NULL است. به بیان دیگر، مقادیر کلید خارجی همواره باید زیرمجموعه مقادیر کلید کاندید باشد.

توجه: یک کلید خارجی در یک رابطه هیچگاه نباید ارجاع NULL داشته باشد، این مسأله را به عنوان یک قانون جامعیتی در مدل رابطه‌ای، می‌شناسیم و آن را **قانون جامعیت ارجاعی** می‌نامیم.

مثال:

S#	Sname	City	S#	P#	QTY	P#	Pname	Color
S ₁	Sn ₁	C ₁	S ₁	P ₁	10	P ₁	Pn ₁	Red
S ₂	Sn ₂	C ₂	S ₁	P ₂	20	P ₂	Pn ₂	Blue
S ₃	Sn ₃	C ₃	S ₂	P ₁	30	P ₃	Pn ₃	Blue
			S ₄	P ₃	40			

جدول اصلی S جدول فرعی SP جدول اصلی P

در جدول فرعی SP، سطر (S₄, P₃, 40) وجود دارد، اما در جدول اصلی P، سطر (P₃) وجود ندارد. این سطر در جدول فرعی SP با علامت × و عبارت "ارجاع NULL" مشخص شده است.

درج رکورد (S₄, P₃, 40) در جدول فرعی SP، غیر مجاز و غیر ممکن است، زیرا سبب ارجاع NULL می‌گردد، در واقع درج رکورد (S₄, P₃, 40) در جدول فرعی SP بدون وجود مقدار متناظر در جدول اصلی S و جدول اصلی P، باعث می‌شود رکود داخل جدول فرعی بی‌صاحب بماند. مقادیر ستون کلید خارجی در جدول فرعی نباید بی‌صاحب باشند.

توجه: هر مقداری که در کلید خارجی (جدول فرعی) وجود دارد، باید دارای مقدار متناظر در کلید کاندید (جدول اصلی) باشد ولی عکس آن صادق نیست.

مثال: مانند S₃ که در جدول اصلی S قرار دارد ولی الزامی ندارد در جدول فرعی SP هم باشد. اما باید همه مقادیر کلید خارجی در کلید کاندید باشد، از بالا به پایین می‌بارد، از پایین به بالا که نمی‌بارد!

توجه: در مفهوم کلید خارجی، ارتباطات به ما قدرت می‌دهد، در جهت کسب اطلاعات بیشتر جدول فرعی از جدول اصلی، برای مثال مقدار S₁ از جدول فرعی SP را در دست‌تان نگه دارید و مُشت کنید، حالا ببرید روی جدول S و دست‌تان را باز کنید، می‌بینید که روی ستون S# و S₁ می‌افتد، بنابراین مشاهده می‌کنید که Sname و City شماره تولیدکننده S₁ به ترتیب برابر Sn₁ و C₁ است.

(مهندسی کامپیوتر - دولتی ۹۰)

تست - رابطه‌های r و s را در نظر بگیرید.

$r(R_1 : D_1, R_2 : D_2, R_3 : D_3)$

$s(R_2 : D_2, R_3 : D_3, R_4 : D_4)$

کدام زیر مجموعه از صفات r می‌تواند کلید خارجی باشد؟

R_2, R_3 (۴)

R_2 (۳)

R_3 (۲)

R_1, R_2 (۱)

پاسخ: گزینه (۳) صحیح است.

ستون R_2 در جدول r یک کلید خارجی است به جدول s .

برای این که ستونی از یک جدول بتواند کلید خارجی به جدول دیگر باشد باید نوع آن با نوع کلید کاندید (کلید اصلی یا فرعی) جدول دیگر برابر باشد. همچنین کلید خارجی یک رابطه باید در رابطه دیگر کلید کاندید باشد. کلید کاندید رابطه s خصیصه R_2 است بنابراین خصیصه R_2 در رابطه r کلید خارجی آن است.

توجه: اگرچه کلید خارجی هیچگاه نباید، ارجاع NULL داشته باشد، اما می‌تواند مقدار NULL داشته باشد، البته به شرطی که کلید خارجی در شرایط **قانون بالا دستی** قرار نگیرد. به تفاوت **ارجاع NULL** و **مقدار NULL** دقت نمایید.

مثال:

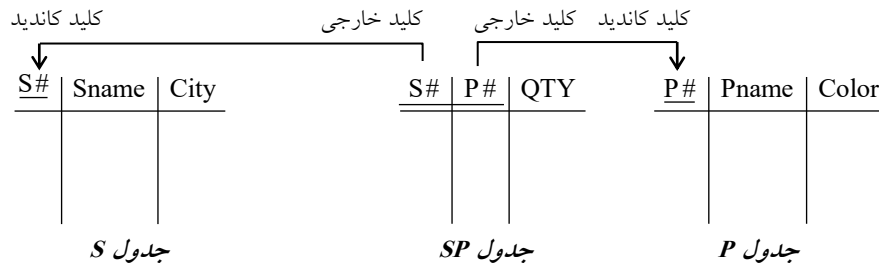
کلید کاندید			کلید خارجی	
<u>a</u>	b	c	<u>a</u>	<u>d</u>
1	4	7	2	1
2	6	5	3	2
3	4	5	2	3
			1	4

R_1

R_2

توجه: می‌توان مقادیر ستون a در جدول R_2 را NULL قرار داد. کلید کاندید (کلید اصلی) جدول R_2 ستون d است.

مثال:

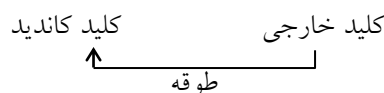


توجه: می‌توان مقادیر کلید خارجی را برابر NULL قرار داد، البته به شرطی که **قانون بالا دستی** بر کلید خارجی حاکم نباشد. در مثال فوق S# و P# در جدول فرعی SP به عنوان کلید خارجی نمی‌توانند NULL باشند، زیرا (S#, P#) با هم قبل از اینکه به تنهایی کلید خارجی باشند، در جدول SP کلید اصلی جدول SP بوده‌اند. بنابراین **قانون بالا دستی جامعیت موجودیت**، بدین معنی که هیچگاه نباید تمام یا بخشی از کلید اصلی NULL باشد، جلوی شدن S# و P# به عنوان کلید خارجی را در جدول SP می‌گیرد.

توجه: مطابق تعریف کلید خارجی، کلید خارجی باید به کلید کاندید همان رابطه یا رابطه دیگر، ارجاع کند، در واقع الزامی ندارد که روابط فرضی R₁ و R₂ از هم مجزا باشند تا کلید خارجی در یک رابطه و کلید کاندید در رابطه‌ای دیگر باشد. زیرا هم کلید کاندید و هم کلید خارجی می‌توانند در یک رابطه باشند، یعنی R₁ و R₂ می‌توانند یک رابطه باشند، بنابراین کلید خارجی یک رابطه می‌تواند متناظر با کلید کاندید همان رابطه باشد. در واقع ممکن است، دیاگرام ارجاعی دارای **طوقه** (کلید خارجی خودارجاع) باشد، به این معنی که مبداء و مقصد کلید خارجی با هم یکی باشد. این مورد زمانی رخ می‌دهد که درجه ارتباط موجودیت‌ها در مدل تحلیل (مدل ER) برابر مقدار 1 یعنی درجه 1 باشد.

مثال: فرض کنید در یک جامعه، هر شخص به صورت **اجباری**، دقیقاً باید با یک شخص دیگر در رابطه همسری باشد. یعنی هر شخص باید **متاهل** باشد و دقیقاً یک همسر داشته باشد. حال بگویید نام همسر آقای 1 چیست؟

PersonID	Name	SpouseID
1	Ali	2
2	Sara	1
3	Re za	4
4	Maryam	3



پاسخ: همسر آقای 1 خانم 2 به نام Sara است.

توجه: ستون SpouseID به عنوان کلید خارجی به ستون PersonID ارجاع می‌کند، در واقع دیاگرام ارجاعی دارای طوقه (کلید خارجی خودارجاع) است به این معنی که مبدأ و مقصد کلید خارجی یک جدول است.

کلید خارجی ترکیبی: مجموعه‌ای از صفات رابطه R_2 را کلید خارجی این رابطه می‌گویند، هرگاه در همان رابطه یا رابطه دیگر مثل R_1 کلید کاندید (اصلی یا فرعی) باشد، بنابراین هیچ الزامی برای اینکه کلید خارجی ترکیبی نباشد و ساده باشد، وجود ندارد، بنابراین کلید خارجی می‌تواند یک کلید ترکیبی باشد، یعنی بیش از یک صفت داشته باشد، لذا اگر کلید خارجی ترکیبی باشد حتماً کلید کاندید متناظر نیز ترکیبی است.

مثال:

کلید خارجی			کلید کاندید		
			↓		
a	b	c	b	c	d
5	1	2	1	2	8
6	3	4	3	4	9
4	NULL	NULL	5	6	3
			7	2	4
			1	8	1

توجه: اگر کلید خارجی ترکیبی باشد (یعنی بیش از یک صفت داشته باشد) یا تماماً NULL است یا تماماً غیر NULL، به عبارت دیگر هیچگاه بخشی از یک کلید خارجی ترکیبی نمی‌تواند مقدار NULL بگیرد.

توجه: ابرکلید و کلیدکاندید در مباحث تئوری و کلید اصلی و کلید خارجی در پیاده‌سازی مورد استفاده قرار می‌گیرند. کاربرد ابرکلید در کشف کلید کاندید است.

(مهندس کامپیوتر - دهانی ۸۸)

تست - کدام عبارت، در مورد کلید خارجی در مدل رابطه‌ای صحیح است؟

- (۱) کلید خارجی یکی از کلیدهای کاندید همان رابطه است.
- (۲) کلید خارجی بایستی کلید اصلی رابطه دیگری باشد. (نباید کلید فرعی باشد)
- (۳) کلید خارجی بایستی خصیصه ساده باشد.
- (۴) کلید خارجی یک رابطه می‌تواند متناظر با مقادیر کلید کاندید همان رابطه باشد.

پاسخ: گزینه (۴) صحیح است.

تست - اگر FK کلید خارجی رابطه R1 متناظر با نامزد کلیدی CK از رابطه R2 باشد، کدام عبارت صحیح است؟

- (۱) FK نمی تواند NULL باشد.
(۲) روابط R1 و R2 متفاوت هستند.
(۳) هر مقدار غیر NULL برای FK برابر یک مقدار CK است.
(۴) هر مقدار غیر NULL برای CK برابر یک مقدار FK است.

پاسخ: گزینه (۳) صحیح است.

نگاشت مدل تحلیل (نمودار نهاد و رابطه) به مدل طراحی (مدل رابطه‌ای)

به طور کلی در مدل رابطه‌ای، هر **موجودیت** شناسایی شده در مدل تحلیل (نمودار ER) هنگام نگاشت به مدل طراحی (مدل رابطه‌ای) به یک **جدول** تبدیل می‌شود. همچنین **صفت‌های** موجودیت پس از نگاشت آن در مدل رابطه‌ای به صورت **ستون‌های** جدول بیان می‌شوند. همچنین **ارتباط** بین موجودیت‌ها از طریق **کلید خارجی** در جداول برقرار می‌گردد.
توجه: در بررسی سربار جداول وجود **مقادیر NULL** غیرقابل تحمل‌ترین نوع سربار، سپس وجود **سربار افزونگی طبیعی** و نه افزونگی تکنیکی در جایگاه بعدی و در نهایت **سربار الحاق طبیعی** در جایگاه آخر به عنوان قابل تحمل‌ترین نوع سربار است.
توجه: به طور کلی، هر طراحی که باعث ورود حجم زیادی **مقدار NULL** به پایگاه داده شود طراحی غیربهبوده و نادرست است، چون منجر به هدررفتن حافظه می‌شود.

فرآیند نگاشت مدل تحلیل به مدل طراحی

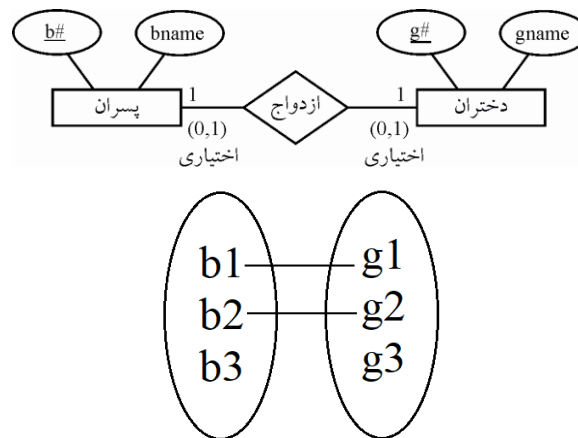
- ۱ - **طراحی توسط یک جدول (مدل یک جدولی):** اگر با یک جدول مقادیر NULL تولید نشود، طراحی بهینه و درست است. اما اگر مقادیر NULL تولید شود، آنگاه باید سراغ تعریف کلید خارجی برویم.
- ۲ - **طراحی توسط کلید خارجی (مدل دو جدولی):** اگر با تعریف **کلید خارجی** مقادیر NULL تولید نشود، طراحی بهینه و درست است. اما اگر مقادیر NULL تولید شود، آنگاه باید سراغ تعریف جدول واسط برویم.
- ۳ - **طراحی توسط جدول واسط (مدل سه جدولی):** تعریف جدول واسط

نگاشت رابطه یک به یک بین دو موجودیت (درجه دو) به مدل رابطه‌ای

حالت اول (اختیاری - اختیاری):

طراحی توسط یک جدول (مدل یک جدولی): اگر با یک جدول مقادیر NULL تولید نشود، طراحی بهینه و درست است. اما اگر مقادیر NULL تولید شود، آنگاه باید سراغ تعریف کلید خارجی برویم.

مثال مدل تحلیل (اختیاری - اختیاری):



درجه ارتباط: درجه 2

کاردینالیتی ارتباط: 1:1

اختیاری اجباری: اختیاری - اختیاری

توجه: با توجه به اختیاری - اختیاری بودن ارتباط، هر پسر حداقل صفر و حداکثر یک دختر برای ازدواج دارد و هر دختر حداقل صفر و حداکثر یک پسر برای ازدواج دارد.

مدل طراحی نادرست (مدل یک جدولی):

b#	bname	g#	gname	Marriage
b1	bn1	g1	gn1	1360
b2	bn2	g2	gn2	1370
b3	bn3	NULL	NULL	NULL
NULL	NULL	g3	gn3	NULL

توجه: در مدل یک جدولی فوق، طرف‌های اختیاری، باعث تولید مقادیر NULL شده‌اند، بنابراین طراحی غیربهینه و نادرست است و باید سراغ تعریف کلید خارجی برویم.

طراحی توسط کلید خارجی (مدل دو جدولی): اگر با تعریف **کلید خارجی** مقادیر NULL تولید نشود، طراحی بهینه و درست است. اما اگر مقادیر NULL تولید شود، آنگاه باید سراغ تعریف جدول واسط برویم.

مدل طراحی نادرست (مدل دو جدولی):

<u>b#</u>	bname	<u>g#</u>	gname	Marriage	b#
b1	bn1	g1	gn1	1360	b1
b2	bn2	g2	gn2	1370	b2
b3	bn3	g3	gn3	NULL	NULL

توجه: در جدول دختران بالا، b# کلید خارجی است.

مدل طراحی نادرست (مدل دو جدولی):

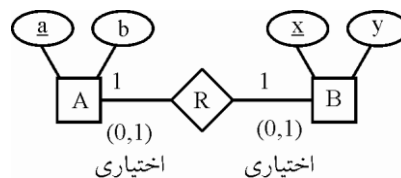
<u>g#</u>	gname	<u>b#</u>	bname	Marriage	g#
g1	gn1	b1	bn1	1360	g1
g2	gn2	b2	bn2	1370	g2
g3	gn3	b3	bn3	NULL	NULL

توجه: در جدول پسران بالا، g# کلید خارجی است.

توجه: در مدل دو جدولی فوق، تعریف **کلید خارجی**، داخل طرف‌های **اختیاری**، باعث **تولید مقادیر NULL** شده‌است، چون در طرف **اختیاری الزام** ارتباط هر نمونه با یک نمونه دیگر وجود ندارد. بنابراین طراحی غیربهینه و نادرست است و باید سراغ تعریف جدول واسط برویم.

طراحی توسط جدول واسط (مدل سه جدولی): در این شرایط، هر موجودیت به یک جدول تبدیل می‌گردد. و یک جدول واسط نیز به عنوان ارتباط دهنده دو جدول مورد استفاده قرار می‌گیرد. همچنین کلید کاندید جدول واسط **یکی از** کلیدهای دو جدول دیگر می‌باشد. **صفات متصل به رابطه**، درون جدول واسط **مستتر** می‌شود.

مدل تحلیل:



مدل طراحی:

<u>a</u> b	a x ...	<u>x</u> y
جدول A	جدول AB	جدول B

مدل طراحی درست (مدل سه جدولی):

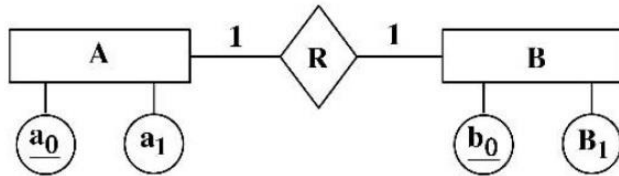
<u>b#</u>	bname	<u>b#</u>	<u>g#</u>	Marriage	<u>g#</u>	gname
b1	bn1	b1	g1	1360	g1	gn1
b2	bn2	b2	g2	1370	g2	gn2
b3	bn3				g3	gn3
جدول پسران		جدول ازدواج		جدول دختران		

توجه: کلید کاندید جدول ازدواج، ستون $b\#$ یا ستون $g\#$ می‌باشد. زیرا هر یک به تنهایی قدرت تفکیک سطرها را دارند. ترکیب ستون‌های $b\#$ و $g\#$ باهم به عنوان $b\#g\#$ ، ابرکلید ایجاد می‌کند.

توجه: ستون $b\#$ در جدول ازدواج به عنوان کلید خارجی تعریف می‌گردد که به جدول پسران ارجاع می‌کند. همچنین ستون $g\#$ در جدول ازدواج به عنوان کلید خارجی تعریف می‌گردد که به جدول دختران ارجاع می‌کند.

تست - کدام مورد در تبدیل نمودار موجودیت رابطه مطرح شده به جدول، درست است؟ (لازم به ذکر است تعداد نمونه موجودیت‌های A و B بسیار زیاد و نرخ شرکت کردن آنها در رابطه R بسیار اندک است).

(مهندسی کامپیوتر-دولتی ۱۴۰۱)



جدول ABR

<u>a₀</u>	a ₁	b ₀	b ₁
----------------------	----------------	----------------	----------------

(۱)

جدول A

<u>a₀</u>	a ₁	b ₀
----------------------	----------------	----------------

جدول B

<u>b₀</u>	b ₁
----------------------	----------------

(۲)

جدول A

<u>a₀</u>	a ₁
----------------------	----------------

جدول B

<u>b₀</u>	b ₁
----------------------	----------------

جدول R

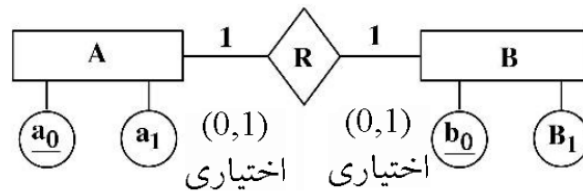
<u>a₀</u>	<u>b₀</u>
----------------------	----------------------

(۳)

(۴) همه موارد درست هستند.

پاسخ - گزینه (۱) صحیح است.

مدل تحلیل:



مدل طراحی:

<u>a₀</u>	a ₁
----------------------	----------------

جدول A

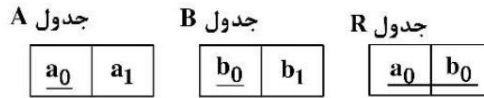
a ₀	b ₀	...
----------------	----------------	-----

جدول AB

<u>b₀</u>	b ₁
----------------------	----------------

جدول B

توجه: سازمان سنجش آموزش کشور، در **کلید اولیه** خود، گزینه سوم را به عنوان پاسخ اعلام کرده بود. اما در **کلید نهایی** این سوال حذف گردید، که کار درستی بوده است. زیرا مطابق گزینه سوم به صورت زیر:

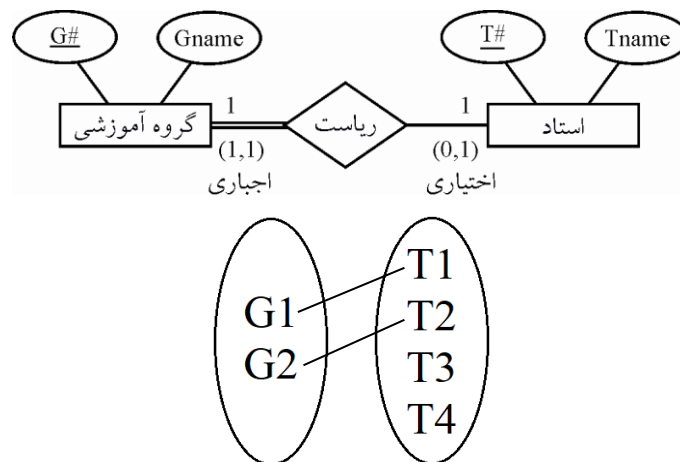


ترکیب ستون‌های a_0 و b_0 با هم به عنوان a_0b_0 در جدول R، ابرکلید ایجاد می‌کند، که پاسخ درستی به صورت سوال نیست.

حالت دوم (اختیاری - اجباری)

طراحی توسط یک جدول (مدل یک جدولی): اگر با یک جدول مقادیر NULL تولید نشود، طراحی بهینه و درست است. اما اگر مقادیر NULL تولید شود، آنگاه باید سراغ تعریف کلید خارجی برویم.

مثال مدل تحلیل (اختیاری - اجباری):



توجه: با توجه به اختیاری - اجباری بودن ارتباط، هر گروه آموزشی حداقل یک و حداکثر یک استاد برای ریاست دارد و هر استاد حداقل صفر و حداکثر یک گروه آموزشی را ریاست می‌کند.

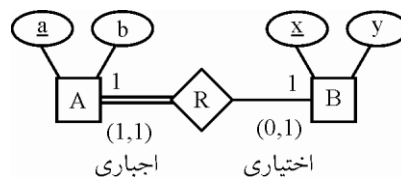
مدل طراحی نادرست: (مدل یک جدولی)

G#	Gname	T#	Tname
G1	Gn1	T1	Tn1
G2	Gn2	T2	Tn2
NULL	NULL	T3	Tn3
NULL	NULL	T4	Tn4

توجه: در مدل یک جدولی فوق، طرف **اختیاری**، باعث **تولید مقادیر NULL** شده‌است، بنابراین طراحی غیربهینه و نادرست است و باید سراغ تعریف کلید خارجی برویم.

طراحی توسط کلید خارجی (مدل دو جدولی): اگر با تعریف **کلید خارجی** مقادیر NULL تولید نشود، طراحی بهینه و درست است. در این شرایط، هر موجودیت به یک جدول تبدیل می‌گردد. و کلید کاندید جدول موجودیت اختیاری در جدول موجودیت اجباری به عنوان کلید خارجی تعریف می‌گردد. **صفات متصل به رابطه**، درون جدول اجباری **مستتر** می‌شود.

مدل تحلیل:



مدل طراحی:

<u>a</u>	b	x

جدول A

<u>x</u>	y

جدول B

مدل طراحی درست: (مدل دو جدولی)

G#	Gname	T#
G1	Gn1	T1
G2	Gn2	T2

T#	Tname
T1	Tn1
T2	Tn2
T3	Tn3
T4	Tn4

جدول استاد (اختیاری) جدول گروه آموزشی (اجباری)

توجه: در مدل دو جدولی فوق، تعریف **کلید خارجی**، داخل طرف **اجباری**، باعث **تولید مقادیر NULL** نمی‌شود، چون در طرف **اجباری الزام** ارتباط هر نمونه با یک نمونه دیگر وجود دارد. بنابراین طراحی بهینه و درست است و سراغ تعریف جدول واسط نمی‌رویم.

توجه: اگر بالعکس نگاشت فوق عمل کنیم و کلید کاندید جدول موجودیت اجباری را در جدول موجودیت اختیاری به عنوان کلید خارجی تعریف کنیم، آنگاه جدول موجودیت اختیاری با مقادیر زیاد NULL مواجه می‌گردد.

مدل طراحی نادرست: (مدل دو جدولی)

<u>G #</u>	Gname	<u>T #</u>	Tname	G #
G1	Gn1	T1	Tn1	G1
G2	Gn2	T2	Tn2	G2
		T3	Tn3	NULL
		T4	Tn4	NULL

جدول استاد (اختیاری) جدول گروه آموزشی (اجباری)

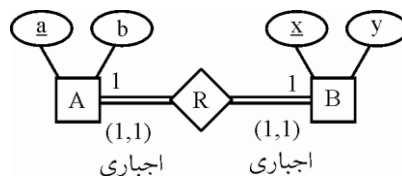
توجه: مقادیر NULL در ستون G# قابل مشاهده است، زیرا بسیاری از اساتید، رئیس هیچ گروه آموزشی نیستند!

توجه: در مدل دو جدولی فوق، تعریف **کلید خارجی**، داخل طرف **اختیاری**، باعث **تولید مقادیر NULL** شده است، چون در طرف **اختیاری الزام** ارتباط هر نمونه با یک نمونه دیگر وجود ندارد. بنابراین طراحی غیربهبوده و نادرست است.

حالت سوم (اجباری = اجباری)

طراحی توسط یک جدول (مدل یک جدولی): اگر با یک جدول مقادیر NULL تولید نشود، طراحی بهبود یافته و درست است. در این شرایط، هر دو موجودیت در قالب یک جدول قرار می‌گیرند. و کلید کاندید جدول حاصل، کلید کاندید هر یک از موجودیت‌ها می‌تواند باشد. **صفات متصل به رابطه**، درون جدول **مستتر** می‌شود.

مدل تحلیل:



مدل طراحی:

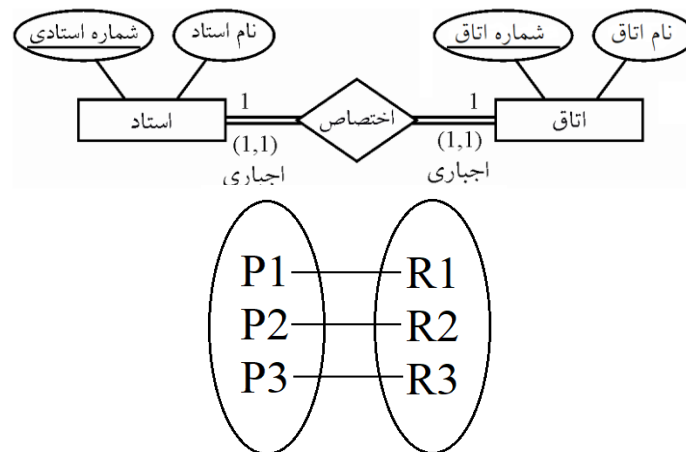
<u>a</u>	b	x	y

یا

a	b	<u>x</u>	y

جدول AB جدول AB

مثال مدل تحلیل (اجباری - اجباری):



توجه: با توجه به اجباری - اجباری بودن ارتباط، هر استاد حداقل و حداکثر یک اتاق دارد و هر اتاق حداکثر و حداقل یک استاد دارد.

مدل طراحی درست: (مدل یک جدولی)

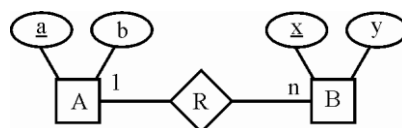
Prof#	Pname	Room#	Rname	یا	Prof#	Pname	Room#	Rname
P1	Pn1	R1	Rn1		P1	Pn1	R1	Rn1
P2	Pn2	R2	Rn2		P2	Pn2	R2	Rn2
P3	Pn3	R3	Rn3		P3	Pn3	R3	Rn3

نتیجه گیری: به طور کلی، در نگاشت رابطه یک به یک بین دو موجودیت (درجه دو) به مدل رابطه‌ای، هرچه عرصه توسط اجباری بودن تنگ‌تر شود، طراحی از مدل سه جدولی به مدل دو جدولی و در نهایت به مدل یک جدولی سوق پیدا می‌کند.

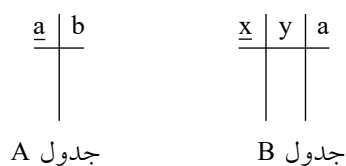
نگاشت رابطه یک به چند بین دو موجودیت (درجه دو) به مدل رابطه‌ای

مستقل از اختیاری یا اجباری بودن موجودیت‌ها، هر موجودیت به یک جدول تبدیل می‌گردد. و کلید کاندید جدول یک در جدول چند به عنوان کلید خارجی تعریف می‌گردد، برای حفظ **خاصیت تابعی**، جهت اشاره، از چند به یک است. همچنین **صفات متصل به رابطه**، درون جدول چند **مستتر** می‌شود.

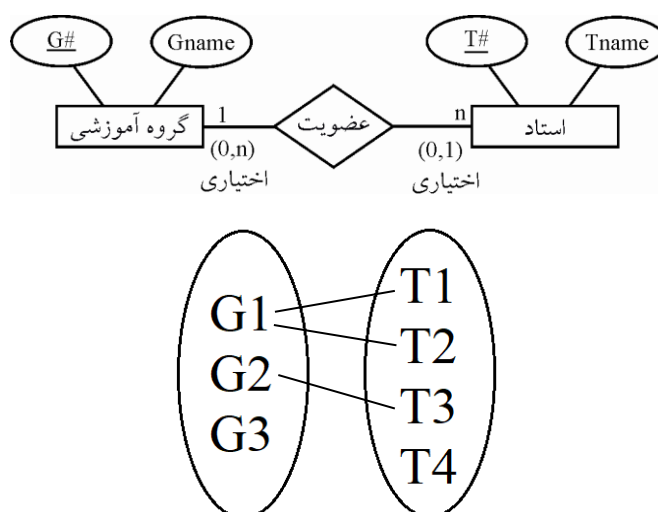
مدل تحلیل:



مدل طراحی درست: (مدل دو جدولی)



مثال مدل تحلیل (اختیاری - اختیاری):



توجه: با توجه به اختیاری - اختیاری بودن ارتباط، هر گروه آموزشی حداقل صفر استاد و حداکثر چند استاد دارد و هر استاد حداقل صفر و حداکثر در یک گروه آموزشی عضویت دارد.

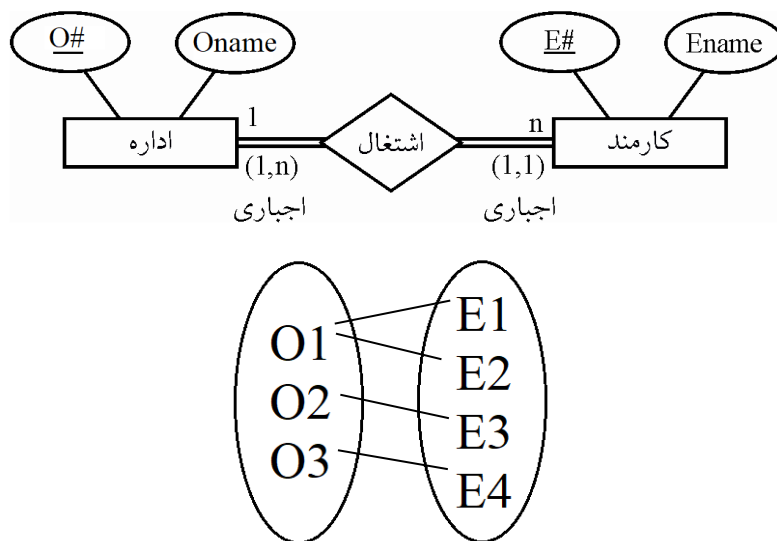
مدل طراحی درست: (مدل دو جدولی)

G #	Tname	T #	Tname	G #
G1	Gn1	T1	Tn1	G1
G2	Gn2	T2	Tn2	G1
G3	Gn3	T3	Tn3	G2
		T4	Tn4	NULL

جدول استاد (اختیاری) جدول گروه آموزشی (اختیاری)

توجه: اگر استادی مثل T4 در جدول استاد، عضو گروه آموزشی نباشد، ستون G# برای آن مقدار NULL می‌گیرد، اشکالی هم ندارد چون مدل طراحی بهینه‌تری ندارد. اصولاً در یک به چند مدل دو جدولی کافی هست و نیاز به مدل سه جدولی و تحمیل سربار الحاق بیش از حد نیست.

مثال مدل تحلیل (اجباری - اجباری):



توجه: با توجه به اجباری - اجباری بودن ارتباط، هر اداره حداقل یک کارمند و حداکثر چند کارمند دارد و هر کارمند حداقل و حداکثر در یک اداره اشتغال دارد.

مدل طراحی درست: (مدل دو جدولی)

O#	Oname	E#	Ename	O#
O1	On1	E1	En1	O1
O2	On2	E2	En2	O1
O3	On3	E3	En3	O2
		E4	En4	O3

جدول کارمند (اجباری) جدول اداره (اجباری)

توجه: با توجه به اجباری بودن موجودیت کارمند، هنگام پیاده‌سازی، ستون O# در جدول کارمند می‌بایست NOT NULL تعریف گردد.

توجه: در 1:N هر سطر در سمت «چند» فقط به یک سطر در سمت «یک» وصل است. FK همیشه داخل طرف N قرار می‌گیرد چون تنها یک مقدار واحد برای لینک در طرف یک برای آن وجود دارد.

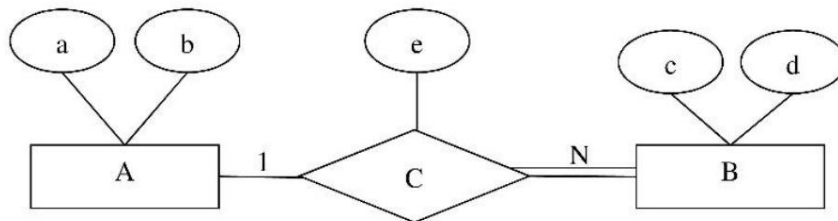
مدل طراحی نادرست: (مدل یک جدولی)

O#	Oname	E#	Ename
O1	On1	E1	En1
O1	On1	E2	En2
O2	On2	E3	En3
O3	On3	E4	En4

توجه: در مدل تک جدولی فوق صفت E# کلید کاندید است. مقادیر ستون Oname به ازای O# تکرار شده‌است که افزونگی طبیعی (تکرار اطلاعات محتوایی) محسوب می‌شود، علت هم وابستگی انتقالی ستون Oname به ستون O# است که در مباحث نرمال‌سازی به طور مفصل به آن خواهیم پرداخت. در مدل تک جدولی، اگر بخواهیم به جای کارمند، اداره را هم مستقل شناسایی کنیم، دچار مشکل می‌شویم، چون هر اداره به تنهایی در جدول کلید ندارد. به همین دلایل در طراحی نرمال، رابطه را به دو جدول می‌شکنیم (مدل دو جدولی) تا هم کارمند کلید داشته باشد، هم اداره و هم افزونگی طبیعی رخ ندهد.

تست - با توجه به نمودار ER زیر، کدام مورد، درست است؟

(مهندسی IT - دولتی ۱۴۰۱)

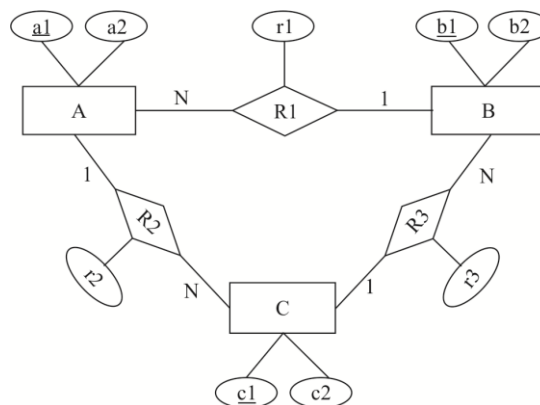


- (۱) صفت e را می‌توان به موجودیت A بدون از دست دادن اطلاعات منتقل کرد.
- (۲) صفت e را می‌توان به هر کدام از دو موجودیت A یا B بدون از دست دادن اطلاعات منتقل کرد.
- (۳) صفت e را می‌توان به موجودیت B بدون از دست دادن اطلاعات منتقل کرد.
- (۴) صفت e را نمی‌توان بدون از دست دادن اطلاعات به هیچ کدام از موجودیت‌ها منتقل کرد.

پاسخ - گزینه (۳) صحیح است.

تست - مدل رابطه‌ای متناظر با نمودار ER زیر کدام است؟

(مهندسی کامپیوتر - دولتی ۹۹)



- (۱) $A(a_1, a_2, b_1, r_1) B(b_1, b_2, c_1, r_3) C(c_1, c_2, a_1, r_2)$
- (۲) $A(a_1, a_2, b_1, r_1) B(b_1, b_2, c_1, r_3) C(c_1, c_2, a_1, r_2)$
- (۳) $A(a_1, a_2) B(b_1, b_2, a_1, r_1) C(c_1, c_2, b_1, r_3) R_1(a_1, b_1, r_1) R_3(b_1, c_1, r_3)$
- (۴) $A(a_1, a_2) B(b_1, b_2) C(c_1, c_2) R_1(a_1, b_1, r_1) R_2(a_1, c_1, r_2) R_3(b_1, c_1, r_3)$

گزینه (۱) صحیح است.

<u>c1</u>	c2

جدول C

<u>b1</u>	b2	c1	r3

جدول B

<u>a1</u>	a2	b1	r1

جدول A

<u>b1</u>	b2	c1	r3

جدول B

<u>a1</u>	a2	b1	r1

جدول A

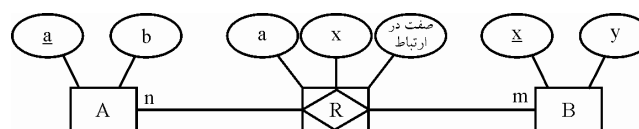
<u>c1</u>	c2	a1	r2

جدول C

نگاشت رابطه چند به چند بین دو موجودیت (درجه دو) به مدل رابطه‌ای

مستقل از اختیاری یا اجباری بودن موجودیت‌ها، هر موجودیت به یک جدول تبدیل می‌گردد و یک **جدول پُل (Bridge)** نیز به عنوان ارتباط دهنده دو جدول مورد استفاده قرار می‌گیرد. همچنین کلید کاندید جدول پُل از ترکیب کلید کاندید دو جدول دیگر ایجاد می‌گردد. همچنین **صفات متصل به رابطه**، درون جدول پُل، **مستتر** می‌شود.

مدل تحلیل:



مدل طراحی درست: (مدل سه جدولی)

<u>a</u>	b	<u>a</u>	x	...	<u>x</u>	y

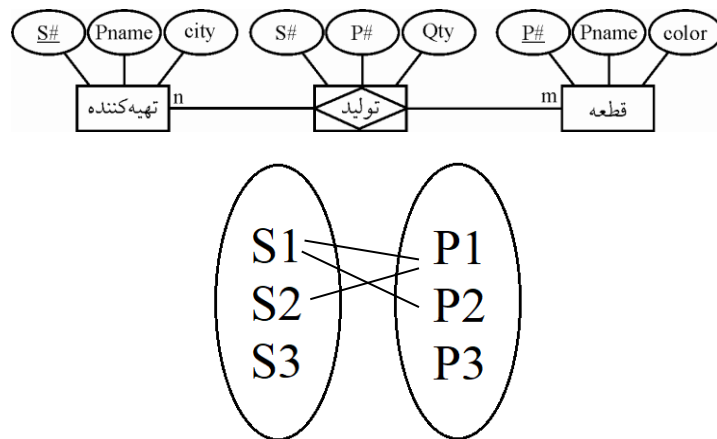
جدول A

جدول AB

جدول B

توجه: ستون a در جدول AB به عنوان کلید خارجی تعریف می گردد که به جدول A ارجاع می کند. همچنین ستون x در جدول AB به عنوان کلید خارجی تعریف می گردد که به جدول B ارجاع می کند.

مثال مدل تحلیل:



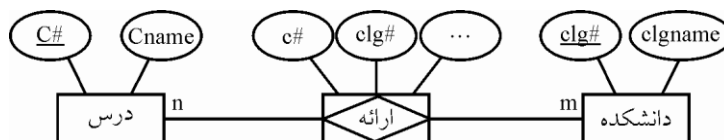
توجه: صفت QTY، به عنوان **صفت در ارتباط**، تعداد قطعات را مشخص می کند.

مدل طراحی درست: (مدل سه جدولی)

S#	Sname	City	S#	P#	QTY	P#	Pname	Color
S1	Sn1	C1	S1	P1	10	P1	Pn1	Red
S2	Sn2	C2	S1	P2	20	P2	Pn2	Blue
S3	Sn3	C3	S2	P1	30	P3	Pn3	Green

جدول S (تهیه کننده) جدول SP (تولید) جدول P (قطعه)

مثال مدل تحلیل:



مدل طراحی درست: (مدل سه جدولی)

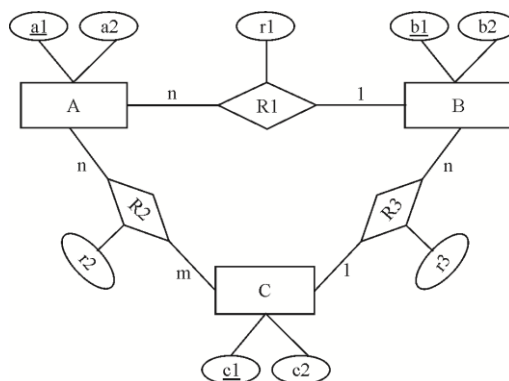
<u>C#</u>	Cname	<u>C#</u>	Clg#	...	<u>Clg#</u>	C lg name

جدول درس جدول ارائه جدول دانشکده

نتیجه‌گیری: حداکثر تعداد جداول نگاشت بین دو موجودیت (درجه دو) به مدل رابطه‌ای برابر $n+1$ یعنی $2+1=3$ است. (n درجه رابطه است که بین دو موجودیت درجه 2 است).

(مهندسی IT - دولتی ۹۸)

تست - مدل رابطه‌ای متناظر با نمودار ER زیر کدام است؟



(۱) $A(\underline{a1}, a2) B(\underline{b1}, b2) C(\underline{c1}, c2) R1(\underline{a1}, \underline{b1}, r1) R2(\underline{a1}, \underline{c1}, r2) R3(\underline{b1}, \underline{c1}, r3)$

(۲) $A(\underline{a1}, a2) B(\underline{b1}, b2, a1, r1) C(\underline{c1}, c2, b1, r3) R1(a1, b1, r1) R3(b1, c1, r3)$

(۳) $A(\underline{a1}, a2, b1) B(\underline{b1}, b2, c1, r1) C(\underline{c1}, c2, r3) R2(\underline{a1}, \underline{c1}, r2)$

(۴) $A(\underline{a1}, a2, b1, r1) B(\underline{b1}, b2, c1, r3) C(\underline{c1}, c2) R2(\underline{a1}, \underline{c1}, r2)$

پاسخ - گزینه (۴) صحیح است.

<u>c1</u>	c2	<u>b1</u>	b2	c1	r3

جدول C جدول B

a1	a2	b1	r1

جدول A

b1	b2	c1	r3

جدول B

a1	a2	b1	r1

جدول A

a1	c1	r2

جدول r2

c1	c2

جدول C

جمع‌بندی نگاشت رابطه درجه دو به مدل رابطه‌ای

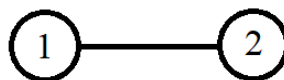
درجه ارتباط ۲			
ابزار	تعداد جدول	اختیاری - اختیاری	کاردینالیتی ارتباط
جدول واسط	سه جدولی	اختیاری - اختیاری	1:1
کلید خارجی	دو جدولی	اختیاری - اجباری	
یک جدول	یک جدولی	اجباری - اجباری	
کلید خارجی	دو جدولی	مستقل	1:n
جدول واسط	سه جدولی	مستقل	n:m

نگاشت رابطه یک به یک در یک موجودیت (درجه یک) به مدل رابطه‌ای

حالت اول (اختیاری): اگر موجودیت **اختیاری** باشد. موجودیت به یک جدول اصلی تبدیل می‌گردد و یک جدول فرعی برای نمایش روابط مورد استفاده قرار می‌گیرد.

مثال: فرض کنید در یک جامعه، هر شخص به صورت **اختیاری**، حداکثر با یک شخص دیگر در رابطه همسری باشد. یعنی هر شخص می‌تواند **مجرد** باشد یا فقط یک همسر داشته باشد.

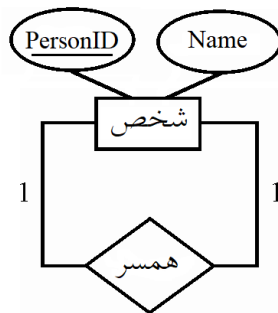
توجه: مدل ریاضی رابطه یک به یک درجه یک، **جفت** است.



از سمت **شخص**: هر شخص **حداکثر** یک همسر دارد. (هر شخص متاهل فقط به یک نفر وصل است.)

از سمت **همسر**: هر همسر **حداکثر** یک شخص دارد. (هر همسر فقط به یک نفر وصل است.)

مثال مدل تحلیل (اختیاری):



مدل طراحی درست: (مدل دو جدولی)

PersonID	Name	PersonID	SpouseID
1	Ali	1	2
2	Sara	2	1
3	Re za		
4	Maryam		

جدول شخص جدول همسر

توجه: Ali و Sara جفت (متاهل) هستند. Reza و Maryam مجرد هستند.

توجه: ستون SpouseID در جدول همسر به عنوان **کلید خارجی** به ستون PersonID از جدول شخص ارجاع می‌کند.

توجه: انجام فعالیت طراحی در حالت **اختیاری**، توسط **مدل یک جدولی**، سبب ایجاد مقادیر NULL زیاد می‌گردد. بنابراین همان **مدل دو جدولی** مدل بهینه و درست است.

مدل طراحی نادرست: (مدل یک جدولی)

PersonID	Name	SpouseID
1	Ali	2
2	Sara	1
3	Re za	NULL
4	Maryam	NULL

کلید خارجی کلید کاندید

طوقه

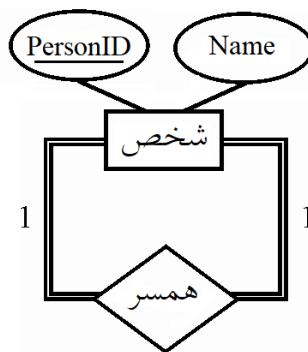
حالت دوم (اجباری): اگر موجودیت اجباری باشد. موجودیت به یک جدول تبدیل می‌گردد. رابطه نیز در همان جدول با **کلید خارجی خودارجاع** نمایش داده می‌شود.

مثال: فرض کنید در یک جامعه، هر شخص به صورت **اجباری**، دقیقاً باید با یک شخص دیگر در رابطه همسری باشد. یعنی هر شخص باید **متاهل** باشد و دقیقاً یک همسر داشته باشد.

از سمت شخص: هر شخص **دقیقاً** یک همسر دارد. (هر شخص دقیقاً به یک نفر وصل است).

از سمت همسر: هر همسر **دقیقاً** یک شخص دارد. (هر همسر دقیقاً به یک نفر وصل است).

مدل تحلیل (اجباری):



مدل طراحی درست: (مدل یک جدولی)

PersonID	Name	SpouseID
1	Ali	2
2	Sara	1
3	Reza	4
4	Maryam	3

کلید خارجی کلید کاندید
طوقه

توجه: (Ali و Sara) و (Maryam و Reza) **جفت (متاهل)** هستند.

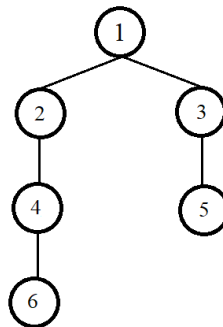
توجه: مطابق تعریف کلید خارجی، کلید خارجی باید به کلید کاندید همان رابطه یا رابطه دیگر، ارجاع کند، در واقع لزومی ندارد که روابط فرضی R1 و R2 از هم مجزا باشند تا کلید خارجی در یک رابطه باشد و کلید کاندید در رابطه‌ای دیگر. زیرا هم کلید کاندید و هم کلید خارجی می‌توانند در یک رابطه باشند، یعنی R1 و R2 می‌توانند یک رابطه باشند، بنابراین کلید خارجی یک رابطه می‌تواند متناظر با کلید کاندید همان رابطه باشد.

توجه: ستون SpouseID به عنوان کلید خارجی به ستون PersonID ارجاع می‌کند، در واقع دیاگرام ارجاعی دارای طوقه (کلید خارجی خودارجاع) است به این معنی که مبدأ و مقصد کلید خارجی یک جدول است.

نگاشت رابطه یک به چند در یک موجودیت (درجه یک) به مدل رابطه‌ای

مستقل از اختیاری یا اجباری بودن موجودیت، موجودیت به یک جدول تبدیل می‌گردد.

مثال: فرض کنید در یک سازمان به صورت درختی، هر مدیر چندین کارمند دارد و هر کارمند یک مدیر بالاسری دارد، به جز مدیر کل که مدیر بالاسری ندارد.
توجه: مدل ریاضی رابطه یک به چند درجه یک، درخت است.

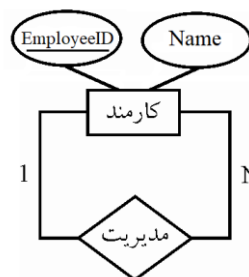


توجه: 1 مدیر 2 و 3 است، 2 مدیر 4 است، 3 مدیر 5 است و 4 مدیر 6 است.

از سمت کارمند: هر کارمند یک مدیر دارد. (هر کارمند به یک مدیر بالادستش وصل است).

از سمت مدیر: هر مدیر چندین کارمند دارد. (هر مدیر به چند کارمند پایین‌دستش وصل است).

مثال مدل تحلیل:



مدل طراحی درست: (مدل یک جدولی)

EmployeeID	Name	ManagerID
1	Amir	NULL
2	Maryam	1
3	Reza	1
4	Sara	2
5	Behzad	3
6	Shirin	4

کلید خارجی (شماره مدیر) کلید کاندید (شماره کارمندی)
↑
طوقه

توجه: سطر مربوط به مدیر کل، که مدیر بالاتری ندارد، مقدار NULL در فیلد ManagerID خود می گیرد.

توجه: شماره مدیر، برای شخصی با شماره کارمندی 4 برابر 2 است و نام 2 برابر Maryam است.
توجه: ستون ManagerID به عنوان کلید خارجی به ستون EmployeeID ارجاع می کند، در واقع دیاگرام ارجاعی دارای طوقه (کلید خارجی خودارجاع) است به این معنی که مبدأ و مقصد کلید خارجی یک جدول است.

تست - در مدل رابطه ای پایگاه داده یک سیستم آموزشی، ارتباط یک به چند میان درس و پیش نیازهای آن، ارتباط چند به چند میان درس و دانشکده، و ارتباط یک به یک ریاست میان گروه آموزشی و استاد وجود دارد. کدام عبارت نمایش مناسبتر هر رابطه را به ترتیب بیان می کند؟

(مهندسی IT - دولتی ۹۴)

۱) درج کلید اصلی درس در موجودیت درس به عنوان کلید خارجی، درج کلیدهای اصلی دانشکده و درس در یک موجودیت جدید به عنوان کلیدهای خارجی، درج کلید اصلی استاد در موجودیت گروه آموزشی به عنوان کلید خارجی.

۲) درج کلیدهای اصلی درس و پیش نیازهای آن در یک موجودیت جدید به عنوان کلیدهای خارجی، درج کلید اصلی دانشکده در موجودیت درس به عنوان کلید خارجی، درج کلیدهای اصلی گروه آموزشی و استاد در یک موجودیت جدید به عنوان کلیدهای خارجی.

۳) درج کلیدهای اصلی درس و پیش نیازهای آن در یک موجودیت جدید به عنوان کلیدهای خارجی، درج کلیدهای اصلی دانشکده و درس در یک موجودیت جدید به عنوان کلیدهای خارجی، و درج کلیدهای اصلی استاد و گروه آموزشی در یک موجودیت جدید به عنوان کلیدهای خارجی.

۴) درج کلید اصلی درس در موجودیت درس به عنوان کلید خارجی، درج کلیدهای اصلی دانشکده و درس در یک موجودیت جدید به عنوان کلیدهای خارجی، درج کلید اصلی گروه آموزشی در موجودیت استاد به عنوان کلید خارجی.

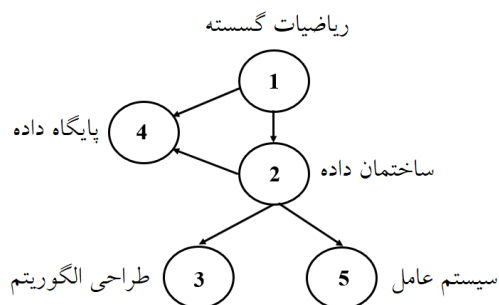
پاسخ - گزینه (۱) صحیح است.

نگاشت رابطه چند به چند در یک موجودیت (درجه یک) به مدل رابطه‌ای

مستقل از اختیاری یا اجباری بودن موجودیت، موجودیت به یک **جدول اصلی** تبدیل می‌گردد. و یک **جدول فرعی** برای نمایش روابط موجود مورد استفاده قرار می‌گیرد. همچنین کلید کاندید جدول اصلی، در جدول ارتباط به عنوان کلید خارجی تعریف می‌گردد.

مثال: فرض کنید در یک دانشگاه به صورت **گراف**، هر درس وابسته می‌تواند چندین درس پیش‌نیاز داشته‌باشد. و هر درس پیش‌نیاز می‌تواند چندین درس وابسته داشته‌باشد.

توجه: مدل ریاضی رابطه چند به چند درجه یک، **گراف** است.

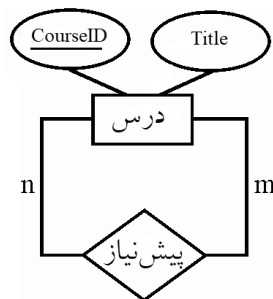


توجه: ساختمان داده پیش‌نیازش ریاضیات گسسته است، طراحی الگوریتم پیش‌نیازش ساختمان داده است، پایگاه داده پیش‌نیازش ریاضیات گسسته و ساختمان داده است و سیستم‌عامل پیش‌نیازش ساختمان داده است.

از سمت درس وابسته: هر درس وابسته می‌تواند چندین درس پیش‌نیاز داشته‌باشد. (هر درس وابسته به چند درس پیش‌نیاز می‌تواند وصل باشد). مثل پایگاه داده

از سمت درس پیش‌نیاز: هر درس پیش‌نیاز می‌تواند چندین درس وابسته داشته‌باشد. (هر درس پیش‌نیاز به چند درس وابسته می‌تواند وصل باشد). مثل ساختمان داده

مثال مدل تحلیل:



مدل طراحی درست: (مدل دو جدولی)

CourseID	Title	CourseID	PrereqID
1	Discrete	2	1
2	DS	3	2
3	Algorithm	4	1
4	DB	4	2
5	OS	5	2

جدول درس (Course)

جدول پیش نیاز (CoursePrereq)

توجه: ستون PrereqID در جدول پیش نیاز (CoursePrereq) به عنوان کلید خارجی به ستون CourseID از جدول درس (Course) ارجاع می‌کند.

توجه: کلید کاندید جدول درس، CourseID است و کلید کاندید جدول پیش نیاز (PrereqID) و CourseID باهم است.

نتیجه گیری: حداکثر تعداد جداول نگاشت بین یک موجودیت (درجه یک) به مدل رابطه‌ای برابر $n+1$ یعنی $1+1=2$ است. (n درجه رابطه است که بین یک موجودیت درجه 1 است).

جمع بندی نگاشت رابطه درجه یک و دو به مدل رابطه‌ای

درجه ارتباط ۲			
ابزار	تعداد جدول	اختیاری - اختیاری	کاردینالیتی ارتباط
جدول واسط	سه جدولی	اختیاری - اختیاری	1:1

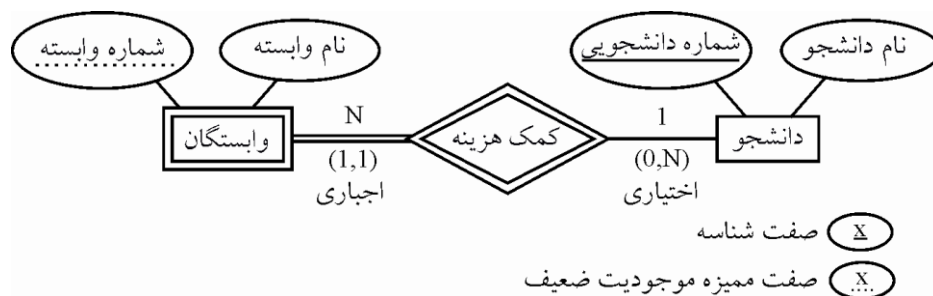
کلید خارجی	دو جدولی	اختیاری - اجباری	
یک جدول	یک جدولی	اجباری - اجباری	
کلید خارجی	دو جدولی	مستقل	1 : n
جدول واسط	سه جدولی	مستقل	n:m
درجه ارتباط ۱			
کلید خارجی	دو جدولی	اختیاری	1:1
یک جدول	یک جدولی	اجباری	
یک جدول	یک جدولی	مستقل	1 : n
کلید خارجی	دو جدولی	مستقل	n:m

نگاشت وابستگی وجودی موجودیت ضعیف و قوی به مدل رابطه‌ای

اگر در یک بانک اطلاعاتی، وجود یک موجودیت، وابسته به موجودیت دیگری باشد که در صورت حذف و تغییر موجودیت اصلی (موجودیت قوی) این موجودیت نیز تغییر کند، این نوع وابستگی را **وابستگی وجودی** گفته و به پدیده وابسته، **موجودیت ضعیف (Weak Entity)** گویند. همچنین موجودیت ضعیف کلید موجودیت قوی را در بر دارد تا هرگونه تغییر یا حذف در موجودیت قوی به موجودیت ضعیف اعمال شود. کاردینالیتی ارتباط موجودیت قوی و ضعیف به صورت 1:N است.

توجه: موجودیت ضعیف با دو مستطیل تو در تو نمایش داده می‌شود.

مثال مدل تحلیل:



توجه: نماد خط مضاعف افقی نشانه اجباری بودن موجودیت چسبیده به آن است.

توجه: نماد خط افقی نشانه اختیاری بودن موجودیت چسبیده به آن است.

توجه: قید (0,N) نشان می‌دهد که هر نمونه موجودیت از دانشجو حداقل با صفر و حداکثر با N نمونه موجودیت از وابستگان ارتباط دارد و قید (1,1) نشان می‌دهد که هر نمونه موجودیت از

وابستگیان حداقل با یک و حداکثر با یک نمونه موجودیت از دانشجو ارتباط دارد.

مدل طراحی:

ST#	STD#	SDname	ST#	Sname
1	101	SD1	1	Sn1
1	102	SD2	2	Sn2
2	101	SD3	3	Sn3
2	103	SD4		
3	101	SD5		

جدول دانشجو

جدول وابستگیان

توجه: کلید کاندید جدول یک یعنی **موجودیت قوی** در جدول چند یعنی **موجودیت ضعیف** به عنوان کلید خارجی تعریف می گردد.

توجه: کلید کاندید جدول چند یعنی **موجودیت ضعیف** برابر ترکیب **کلید خارجی** و **صفت ممیزه** در جدول موجودیت ضعیف است. یعنی کلید کاندید جدول وابستگیان برابر (ST#,STD#) است.

توجه: **صفت ممیزه** یا **کلید جزئی** به طور سراسری در یک موجودیت ضعیف یکتا نیست، بلکه فقط در بین نمونه‌ها یا دسته‌هایی که با موجودیت قوی ارتباط دارند، یکتا است.

توجه: یک موجودیت ضعیف **همیشه** در ارتباطش با موجودیت قوی رابطه **اجباری** دارد، بنابراین کلید خارجی جدول موجودیت ضعیف (جدول وابستگیان) یعنی ستون ST# هیچگاه نمی تواند NULL باشد.

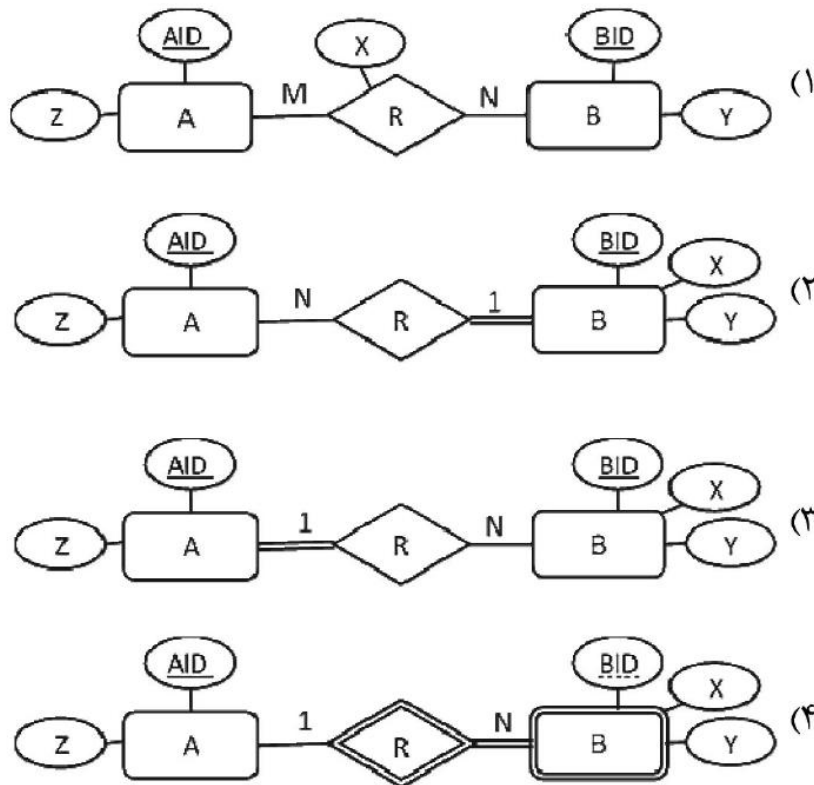
توجه: موجودیت قوی **دانشجو** به طور مستقل برای خود کلید کاندید ST# دارد، اما موجودیت ضعیف **وابستگیان** به طور **وابسته**، کلید کاندید ترکیبی ST#,STD# دارد.

توجه: اگر یک رکورد از جدول **دانشجو** حذف شود، چون رابطه **وابستگی وجودی** موجودیت ضعیف است، بنابراین جدول **وابستگیان** نسبت به **تغییرات حذف** در جدول **دانشجو** به صورت **Cascade** رفتار می کند و تمام رکوردهای مربوطه به طور کامل از جدول **وابستگیان** نیز حذف می شود. این به همیشه وجودم به وجودت وابسته است.

تست - جداول زیر، طراحی منطقی نمودار EER کدام است؟

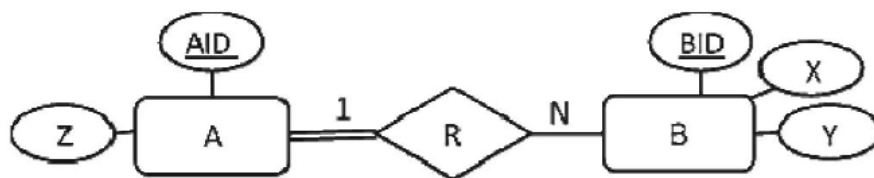
T1 (AID, Z)
T2 (BID, AID, X, Y)

توجه: در طراحی جداول، از خط ممتد در زیر صفت (ها) برای نمایش کلید اصلی و از خط چین برای نمایش کلید خارجی استفاده شده است. در نمودارهای EER، از خط ممتد در زیر صفت برای نمایش کلید اصلی و از خط چین برای نمایش صفت ممیزه یا کلید جزئی موجودیت ضعیف استفاده شده است. (مهندسی IT - دولتی ۱۴۰۲)



پاسخ - گزینه (۳) صحیح است.

بهترین روش حل این نوع سوالات این است که خود EER را به مدل رابطه‌ای تبدیل نمایید.

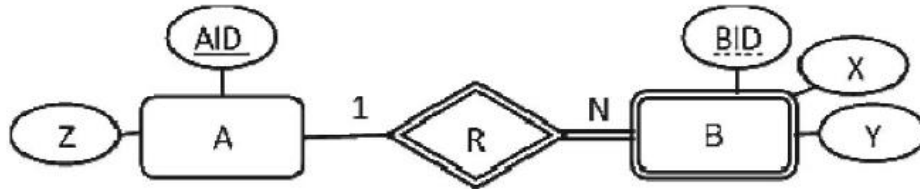


در رابطه یک به چند، مستقل از اختیاری یا اجباری بودن موجودیت‌ها، هر موجودیت به یک جدول تبدیل می‌گردد. و کلید کاندید جدول یک در جدول چند به عنوان کلید خارجی تعریف می‌گردد.

$A = T1(\underline{AID}, Z)$

$B = T2 (\underline{BID}, \underline{AID}, X, Y)$

بررسی گزینه چهارم:



کلید کاندید جدول یک یعنی موجودیت قوی در جدول چند یعنی موجودیت ضعیف به عنوان کلید خارجی تعریف می‌گردد. کلید کاندید جدول چند برابر ترکیب کلید خارجی و صفت ممیزه در جدول موجودیت ضعیف است. یعنی کلید کاندید جدول B برابر (AID, BID) است.

$A = T1 (\underline{AID}, Z)$

$B = T2 (\underline{BID}, \underline{AID}, X, Y)$

تست - در خصوص موجودیت ضعیف (Weak Entity)، کدام مورد درست است؟

(مهندسی IT - دولتی ۱۴۰۴)

(۱) فقط می‌تواند در ارتباط یک به یک مشارکت کند.

(۲) در مدل گسترش یافته استفاده نمی‌شود.

(۳) به تنهایی کلید اصلی ندارد.

(۴) مشارکت آن در ارتباط، اختیاری است.

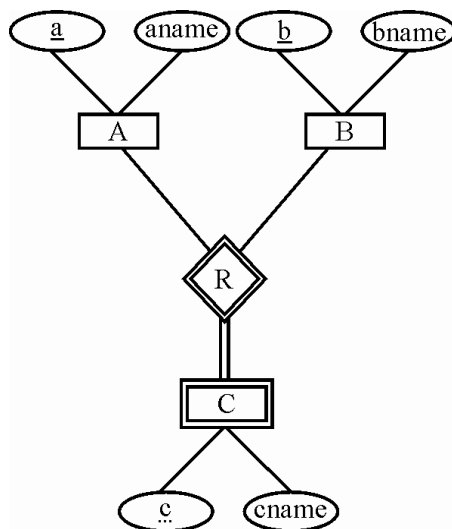
پاسخ - گزینه (۳) صحیح است.

موجودیت قوی چندگانه

موجودیت ضعیف وابسته به یک موجودیت قوی (مانند E1) می‌تواند خود با موجودیت قوی دیگری مانند (مانند E2) رابطه داشته باشد.

مثال: فرض کنید که موجودیت ضعیف C، با دو موجودیت قوی A و B رابطه داشته باشد، یعنی مدل تحلیل (مدل ER) و مدل طراحی (مدل رابطه‌ای) آن به چه شکل خواهد بود؟

مدل تحلیل:



توجه: در شکل فوق صفت a کلید موجودیت قوی A و صفت b کلید موجودیت قوی B است، همچنین صفت c، صفت ممیزه موجودیت ضعیف C است.

مدل طراحی:

<u>a</u>	aname	...

جدول A

<u>a</u>	<u>b</u>	<u>c</u>	cname

جدول C

<u>b</u>	bname	...

جدول B

توجه: کلیدهای کاندید دو جدول موجودیت قوی در جدول موجودیت ضعیف به عنوان کلید خارجی تعریف می‌گردند.

توجه: ستون a در جدول C به عنوان کلید خارجی تعریف می‌گردد که به جدول A ارجاع می‌کند، همچنین ستون b در جدول C به عنوان کلید خارجی تعریف می‌گردد که به جدول B ارجاع می‌کند.

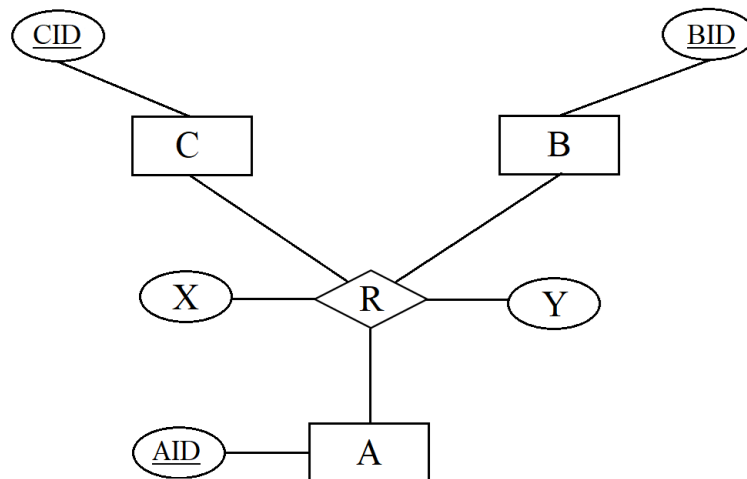
توجه: کلید کاندید جدول موجودیت ضعیف برابر ترکیب کلیدهای خارجی و صفت ممیزه در جدول موجودیت ضعیف است. بنابراین کلید کاندید جدول C برابر (a,b,c) است.

نگاشت رابطه درجه 3 به مدل رابطه‌ای به روش عمومی

اگر درجه ارتباط بیش از 2 باشد، یک روش طراحی عمومی این است که چندی ارتباط $N : M : P$ فرض شود و طراحی مستقل از اختیاری یا اجباری بودن، مطابق نگاشت معمول $N : M$ درجه 2 انجام شود و در صورت نیاز، جداول حاصل جهت بهینه‌سازی بیشتر نرمال‌سازی شود.

توجه: هرگاه n موجودیت در یک ارتباط شرکت داشته باشند، در حالت کلی، حداقل $n+1$ جدول طراحی می‌شود.

مثال مدل تحلیل:



مدل طراحی:

A (AID , ...)

B (BID , ...)

C (CID , ...)

ABC (AID, BID, CID , X, Y)

توجه: ستون AID ، BID و CID در جدول ABC به عنوان کلید خارجی تعریف می‌گردد که به ترتیب به جدول A ، B و C ارجاع می‌کند.

توجه: کلید کاندید جدول ABC برابر (AID, BID, CID , X, Y) است.

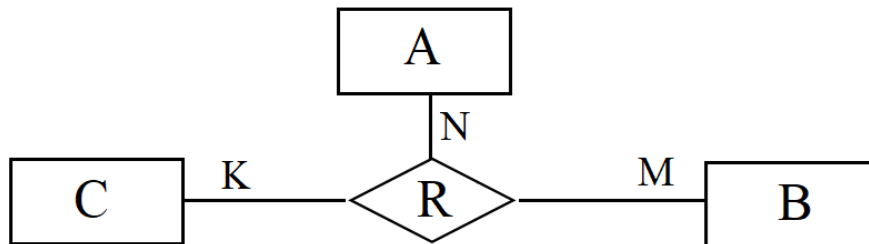
توجه: ستون‌های X و Y به عنوان **صفات در ارتباط** در جدول ABC به صورت **مستتر** قرار گرفته‌است.

تبدیل رابطه درجه 3 (ارتباط سه‌گانی) به سه ارتباط درجه 2 (ارتباط دوگانی)

اگر حداقل در 2 ارتباط چندی n باشد، آنگاه مستقل از اختیاری اجباری بودن روابط، **ارتباط**

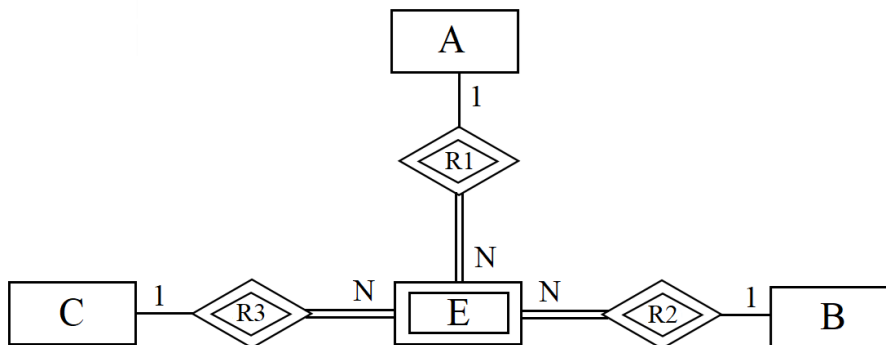
سه‌گانی (یا n گانی) به ارتباط دوگانی تبدیل می‌شود. به صورت زیر:

مدل تحلیل:



توجه: حداقل در 2 ارتباط چندی n است. N سمت موجودیت A ، M سمت موجودیت B و K سمت موجودیت C.

توجه: برای پیاده‌سازی این رابطه درجه 3 ، یک موجودیت ضعیف وابسته به هر سه موجودیت مشارکت‌کننده در ارتباط به نام E تعریف می‌شود، به صورت زیر:



توجه: یک موجودیت ضعیف همیشه در ارتباطش با موجودیت قوی رابطه اجباری دارد.

مدل طراحی:

کلید کاندید جدول یک یعنی موجودیت قوی در جدول چند یعنی موجودیت ضعیف به عنوان کلید خارجی تعریف می‌گردد. همچنین صفات متصل به رابطه، درون جدول چند مستتر می‌شود.

توجه: دقت کنید روابط فوق درجه 3 نیست، چون در وسط ارتباطات به جای لوزی یک موجودیت ضعیف قرار گرفته‌است. بنابراین رابطه درجه 3 اولیه به 3 رابطه درجه 2 ثانویه تبدیل شده‌است.

A (AID , ...)

B (BID, ...)

C (CID, ...)

ABC (AID, BID, CID, X, Y)

توجه: ستون‌های AID, BID و CID در جدول ABC به عنوان کلید خارجی تعریف می‌گردند که به ترتیب به جدول A, B و C ارجاع می‌کنند.

توجه: کلید کاندید جدول ABC برابر (AID, BID, CID, X, Y) است.

توجه: ستون‌های X و Y به عنوان صفات در ارتباط (در اینجا صفت ممیزه) در جدول ABC قرار گرفته‌است.

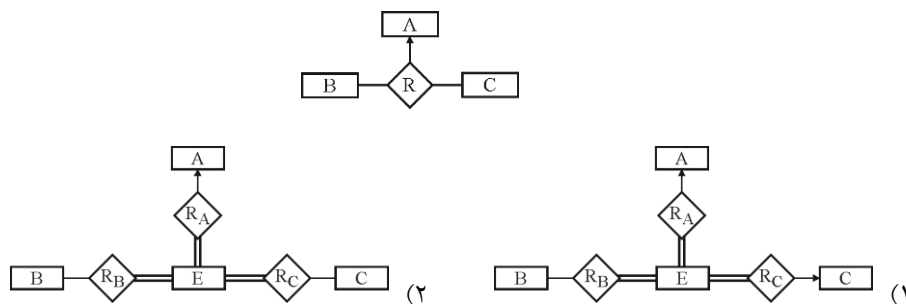
توجه مهم: به ازای ستون‌های X و Y به عنوان صفات در ارتباط (صفات ممیزه) به طور اجباری باید مقادیر ستون‌های AID, BID و CID در جدول ABC مشخص باشد، بنابراین یک موجودیت ضعیف تشکیل شده‌است. زیرا اگر مجموعه‌ای از صفات مثل X و Y به طور مستقل قابل شناسایی نباشند (صفت ممیزه)، آنگاه به صورت **موجودیت ضعیف** و وابسته به موجودیت قوی در نظر گرفته می‌شود.

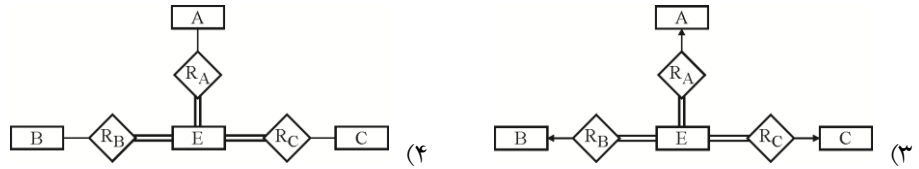
توجه: نتیجه اینکه از یک ارتباط سه‌گانی (رابطه درجه سه)، می‌توان سه ارتباط دوگانی (رابطه درجه دو) را نتیجه گرفت، اما عکس آن صادق نیست.

توجه: واضح است که هر دو روش نگاشت رابطه درجه 3 چه روش عمومی و چه روش تبدیل به درجه 2، مدل طراحی یکسان دارد.

تست - فرض کنید که رابطه سه‌تایی بین موجودیت‌های A, B, C وجود دارد، حال اگر بخواهیم این رابطه سه‌تایی را با رابطه‌های دودویی نمایش دهیم، کدام یک از نمودارهای موجودیت و رابطه (ERD) زیر دقیقاً معادل با این رابطه سه‌تایی می‌باشد؟

(مهندسی IT - دولتی ۸۸)





پاسخ - گزینه (۳) صحیح است.

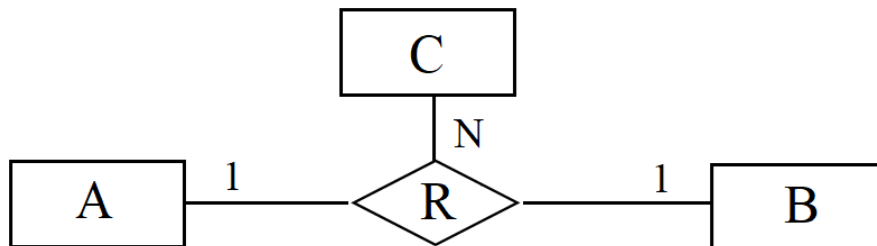
<u>a</u>	...	<u>b</u>	...	<u>a</u>	<u>b</u>	<u>c</u>	<u>x</u>	<u>c</u>	...

جدول A جدول B جدول ABC جدول C

تبدیل رابطه درجه 3 (ارتباط سه‌گانی) به دو رابطه درجه 2 (ارتباط دوگانی)

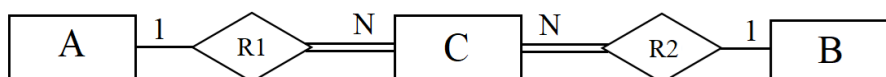
اگر حداقل در 2 ارتباط چندی 1 باشد، آنگاه مستقل از اختیاری اجباری بودن روابط، ارتباط سه‌گانی (یا n‌گانی) به ارتباط دوگانی تبدیل می‌شود. به صورت زیر:

مدل تحلیل:



توجه: حداقل در 2 ارتباط چندی 1 است. 1 سمت موجودیت A، 1 سمت موجودیت B و N سمت موجودیت C.

توجه: در تبدیل ارتباط سه‌گانی به دوگانی فوق، به ازای هر مقدار از موجودیت سمت چند یعنی C، به طور اجباری باید مقدار آن در صفت AID در موجودیت A و صفت BID در موجودیت B مشخص باشد، به صورت زیر:



توجه: برای هر رکورد C، حتماً باید مشخص شود که مربوط به کدام رکورد AID و کدام رکورد BID است. بنابراین شرکت موجودیت C در رابطه با A و B باید اجباری باشد

توجه: در رابطه دوگانی، اگر برای مثال بگوییم دانشجو با استاد، ممکن است وجود دانشجو بدون استاد معنی دار باشد، چون دانشجو و استاد خود موجودیتی مستقل هستند. اما در رابطه سه گانی، هدف فقط توصیف رابطه بین سه چیز مثل استاد، درس و دانشجو باهم است، نه وجود جداگانه‌ی آنها. یعنی هر سه باید با هم رخ بدهند تا رابطه معنا داشته باشد. در رابطه سه گانه حضور فقط یکی از موجودیت‌ها بدون دو تای دیگر، هیچ معنایی ندارد. بنابراین این اجبار (الزام NOT NULL بودن AID و BID) برای حفظ درستی و معنای منطقی رابطه سه گانه است. اگر یکی از آنها NULL باشد، دیگر آن رابطه سه گانه اولیه قابل بازسازی نیست.

مدل طراحی:

کلید کاندید جداول یک در جدول چند به عنوان کلید خارجی تعریف می گردد. همچنین **صفات متصل به رابطه**، درون جدول چند **مستتر** می شود.

A (AID , ...)

B (BID , ...)

C (AID, BID, CID , X , Y)

توجه: ستون‌های AID و BID در جدول C به عنوان کلید خارجی تعریف می گردند که به ترتیب به جدول A و B ارجاع می کنند. همچنین هر دو باید NOT NULL تعریف شود به دلیل الزام اجباری بودن.

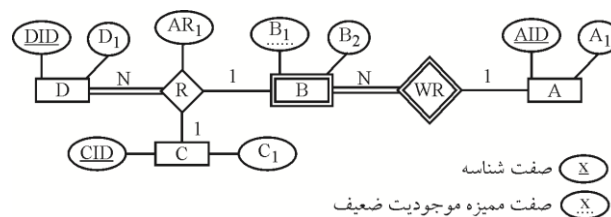
توجه: کلید کاندید جدول C برابر CID است.

توجه: ستون‌های X و Y به عنوان صفات در ارتباط در جدول C قرار گرفته است.

توجه: نتیجه اینکه از یک ارتباط سه گانی (رابطه درجه سه)، می توان دو ارتباط دوگانی (رابطه درجه دو) را نتیجه گرفت، اما عکس آن صادق نیست.

تست - کدام مورد، شامل طراحی منطقی درست ارتباط R در نمودار ER زیر است؟

(مهندسی IT - دولتی ۹۶)



(۱) $R(DID, AID, B1, CID, AR1)$ به صورت یک جدول مستقل
P.K.

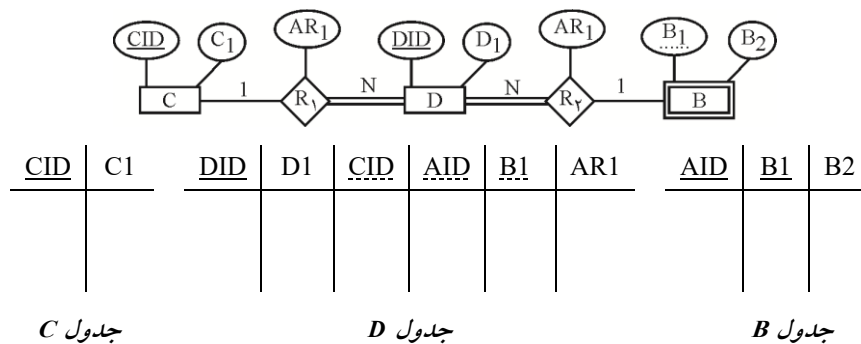
(۲) $B(B1, B2, DID, CID, AR1)$ مستتر در جدول موجودیت B
P.K. F.K. F.K.

(۳) $C(CID, C1, DID, AID, B1, AR1)$ مستتر در جدول موجودیت C

P.K. F.K. F.K.
D(DID, D₁, CID, AID, B₁, AR₁) (۴)
P.K. F.K. F.K.

پاسخ - گزینه (۴) صحیح است.

مدل طراحی:



توجه: ستون CID در جدول D به عنوان کلید خارجی تعریف می‌گردد که به جدول C ارجاع می‌کند. همچنین ستون‌های (AID, B1) در جدول D به عنوان کلید خارجی تعریف می‌گردد که به جدول B ارجاع می‌کند.

توجه: کلید کاندید جدول D فقط و فقط کلید کاندید موجودیت D یعنی صفت DID است.

توجه: اگر یک نوع موجودیت (مانند E₁) تنها یک صفت داشته باشد و تنها با یک نوع موجودیت دیگر (مانند E₂) در ارتباط باشد، می‌توانیم آن را حذف و به عنوان صفت موجودیت دوم (یعنی E₂) در نظر بگیریم. زیرا بهترین راه شناسایی موجودیت‌های یک محیط عملیاتی، بررسی مقادیری است که قرار است ذخیره شود، ذخیره‌سازی مقادیر داخل صفات (ستون‌ها) انجام می‌شود و این صفات قطعاً مرتبط با یک موجودیت هستند. بنابراین هر موجودیت باید دارای چندین صفت باشد. پس موجودیت فاقد صفت معنا ندارد. حتی در اغلب مواقع، موجودیت تک خصیصه‌ای نیز قابل قبول نمی‌باشد و به احتمال فراوان آن صفت مرتبط به یک موجودیت دیگر است که به خطا به عنوان یک موجودیت تک خصیصه‌ای شناخته شده است.

(مهندسی IT - دولتی ۹۶)

تست - کدام مورد، درست نیست؟

(۱) می‌توان از سه ارتباط دوگانی یک ارتباط سه‌گانی را نتیجه گرفت.

(۲) موجودیت ضعیف وابسته به یک موجودیت قوی (مانند E_1) می‌تواند خود با موجودیت قوی دیگری (مانند E_2) رابطه داشته باشد.
(۳) برای حذف یک ارتباط سه‌گانی می‌توان به جای آن یک موجودیت ضعیف وابسته به هر سه موجودیت مشارکت‌کننده در ارتباط، در نظر گرفت.
(۴) اگر یک نوع موجودیت (مانند E_1) تنها یک صفت داشته باشد و تنها با یک نوع موجودیت دیگر (مانند E_2) در ارتباط باشد، می‌توانیم آن را حذف و به عنوان صفت موجودیت دوم (یعنی E_2) در نظر بگیریم.

پاسخ - گزینه (۱) صحیح است.

مدل‌سازی انواع روابط بین موجودیت‌ها یا کلاس‌ها در EER

روابط ساده مدل ER میان موجودیت‌ها با کاردینالیتی $1:1$ ، $1:n$ و $n:m$ قبل‌تر بررسی شد. اما در مدل EER روابط بیشتری مطرح شده‌است که در ادامه به تشریح آن می‌پردازیم.

۱- مدل‌سازی رابطه بخشی یا جزئی (is a part of)

هرگاه رابطه‌ای مابین یک واحد کل و تعدادی واحد جزء مطرح باشد، یعنی یک موجودیت کل از همکاری تعدادی موجودیت جزء تشکیل شده باشد، آنگاه در این شرایط رابطه بخشی خواهیم داشت. این نوع ارتباط، بین موجودیت کل و موجودیت جزء است. رابطه بخشی به دو رابطه تجمع و ترکیب تقسیم می‌گردد.

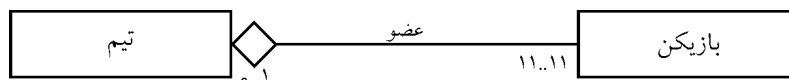
الف) رابطه تجمع (Aggregation) یا رابطه حیات مستقل میان موجودیت کل و جزء

رابطه تجمع، رابطه‌ای مابین یک واحد کل و تعدادی واحد جزء است، یعنی یک موجودیت کل از همکاری تعدادی موجودیت جزء تشکیل شده باشد. برای مثال یک شیء کل «تیم» را در نظر بگیرید، که خود از چندین شیء جزء دیگر مانند اشیاء «بازیکن» تشکیل شده است. به طرف جزء، بخش (Part) و به طرف کل (Whole)، تجمع گفته می‌شود که با لوزی توخالی در انتهایی از خط رابطه که موجودیت کل قرار دارد، نشان داده می‌شود.

مثال: EER برای رابطه (Aggregation) is a part of

مدل تحلیل: (Player is a part of Team)

نمودار EER زیر یک رابطه تجمع میان شیء کل «تیم» و شیء جزء «بازیکن» را نشان می‌دهد.



توجه: یک بازیکن، ممکن است عضو یک تیم باشد یا نباشد، بدین معنی که یک بازیکن می‌تواند مستقل از یک تیم وجود داشته باشد، بنابراین از نماد $1 \dots 0$ استفاده شده است.

در این رابطه، حیات شیء جزء به حیات شیء کل وابسته نیست. به بیان دیگر حیات شیء جزء و شیء کل به هم تقدم و تاخر دارند. یعنی ممکن است شیء جزء موجود باشد، بدون اینکه شیء کل ایجاد گردد و موجود باشد. در این رابطه اگر شیء کل از بین برود، اشیاء جزء، همچنان به حیات خود ادامه می‌دهند. در بازی کامپوتری فوتبال، اشیاء جزء یا همه بازیکن‌های تیم ملی توسط برنامه کامپیوتری ایجاد می‌شوند، اما شما بر حسب سلیقه 11 نفر از آن‌ها را انتخاب می‌کنید و تیم خود یا شیء کل را می‌سازید. اگر هم تیم را منحل کنید، اشیاء بازیکن که توسط برنامه کامپیوتری از همان ابتدای اجرای برنامه ایجاد شده‌اند، همچنان هستند، و کافی است در یک دسته‌بندی دیگر تیم جدیدی را ایجاد نمایید.

نگاشت رابطه (Aggregation) IS A PART OF به مدل رابطه‌ای

موجودیت کل به یک جدول و موجودیت جزء نیز به یک جدول تبدیل می‌گردد و کلید کاندید موجودیت کل در موجودیت جزء به عنوان کلید خارجی تعریف می‌گردد.

مدل طراحی:

TeamID	TeamName	PlayerID	Name	Position	TeamID
1	Tn1	101	N1	Forward	1
2	Tn2	102	N2	Goalkeeper	2
3	Tn3	103	N3	Midfielder	3
جدول Team		104	N4	Forward	1
		105	N5	Midfielder	2
		106	N6	Goalkeeper	3
		107	N7	Forward	NULL

جدول Player

Team (TeamID, TeamName)

Player (PlayerID, Name, Position, TeamID)

توجه: موجودیت کل Team به طور مستقل برای خود کلید کاندید TeamID دارد و همچنین موجودیت جزء Player نیز به طور مستقل برای خود کلید کاندید PlayerID دارد.

توجه: ستون TeamID در جدول Player به عنوان کلید خارجی تعریف می‌گردد که به جدول Team ارجاع می‌کند.

توجه: اگر یک فیلد مثل تیم 1 از جدول Team حذف شود، چون رابطه Aggregation است بنابراین جدول Player نسبت به **تغییرات حذف (Delete)** در جدول Team به صورت **Set** NULL رفتار می‌کند و فیلدهای مربوطه در ستون TeamID از جدول Player مقدار **NULL**

می‌گیرد. همچنین اگر یک فیلد مثل تیم 1 از جدول Team به مقدار 11 **آپدیت** شود، جدول Player نسبت به **تغییرات آپدیت (Update)** در جدول Team به صورت **Cascade** رفتار می‌کند و فیلدهای مربوطه در ستون TeamID از جدول Player مقدار 11 می‌گیرد، قطعه کد مربوطه در داخل جدول فرعی Player جهت اشاره به جدول اصلی Team به صورت زیر است:

Foreign key (TeamID) References Team (TeamID)
on delete Set NULL
on update Cascade

توجه: یک بازیکن به عنوان یک موجودیت مستقل می‌تواند مدتی در یک تیم بازی کند و مدتی بعد در تیم دیگری بازی کند. بازیکن 107 یک بازیکن آزاد است و فعلاً عضو تیمی نیست.

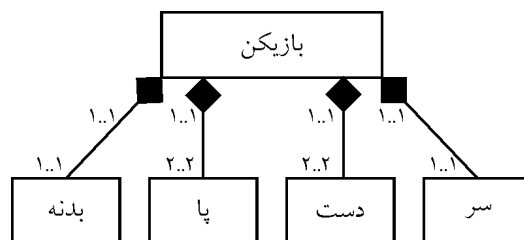
ب) رابطه ترکیب (Composition) یا رابطه حیات وابسته میان موجودیت کل و جزء

رابطه **ترکیب**، رابطه‌ای مابین یک واحد کل و تعدادی واحد جزء است، یعنی یک موجودیت کل از همکاری تعدادی موجودیت جزء تشکیل شده باشد. به نحوی که شیء کل و جزء با هم ایجاد شوند و با هم از بین بروند. برای مثال یک شیء کل «بازیکن» را در نظر بگیرید، که خود از چندین شیء جزء دیگر مانند **سر**، **دست**، **پا** و **بدنه** تشکیل شده است. به نحوی که شیء کل و جزء با هم ایجاد شوند و با هم از بین بروند. به طرف جزء، **بخش** (Part) و به طرف کل (Whole)، **ترکیب** گفته می‌شود که با **لوزی توپر** در انتهای از خط رابطه که **موجودیت کل** قرار دارد، نشان داده می‌شود.

مثال: EER برای رابطه (Composition) is a part of

مدل تحلیل: (BodyPart is a part of Player)

نمودار EER زیر یک رابطه **ترکیب** میان شیء کل «بازیکن» و شیء جزء «سر، دست، پا و بدنه» را نشان می‌دهد.



توجه: حیات اجزای «سر، دست، پا و بدنه» کاملاً به بازیکن، وابسته است، بدین معنی که اجزای «سر، دست، پا و بدنه» هرگز نمی‌توانند مستقل از یک بازیکن وجود داشته باشد، بنابراین از نماد 1..1 در سمت بازیکن استفاده شده است.

در این رابطه، حیات شیء جزء به حیات شیء کل وابسته است. به بیان دیگر حیات شیء جزء و

شیء کل به هم تقدم و تاخر ندارند. یعنی امکان ندارد شیء جزء موجود باشد، بدون اینکه شیء کل ایجاد گردد و موجود باشد. در این رابطه اگر شیء کل از بین برود، اشیاء جزء نیز به طور همزمان با شیء کل از بین می‌روند. در بازی کامپوتری فوتبال، شیء کل بازیکن از اشیاء جزء سر، دست، پا و بدنه تشکیل شده است. و پس از پایان فعالیت، شیء کل و جزء به طور همزمان باهم از بین می‌روند.

نگاشت رابطه IS A PART OF (Composition) به مدل رابطه‌ای

موجودیت کل به یک جدول و موجودیت جزء نیز به یک جدول تبدیل می‌گردد و کلید کاندید موجودیت کل در موجودیت جزء به عنوان کلید خارجی تعریف می‌گردد.

مدل طراحی:

PlayerID	Name	PlayerID	PartID	PartType	HealthStatus	Power
1	N1	1	101	Head	Healthy	90
2	N2	1	102	Hand	Injured	70
جدول Player		1	103	Leg	Injured	50
		1	104	Body	Healthy	98
		2	101	Head	Injured	60
		2	102	Hand	Healthy	88
		2	103	Leg	Healthy	95
		2	104	Body	Injured	70

جدول BodyPart

Player (PlayerID, Name)

BodyPart (PlayerID, PartID, PartType, HealthStatus, Power)

توجه: موجودیت کل Player به طور مستقل برای خود کلید کاندید PlayerID دارد، اما موجودیت جزء BodyPart به طور وابسته، کلید کاندید ترکیبی **PlayerID, PartID** دارد و مانند **Aggregation** کلید کاندید مستقل ندارد.

توجه: ستون PlayerID در جدول BodyPart به عنوان کلید خارجی تعریف می‌گردد که به جدول Player ارجاع می‌کند.

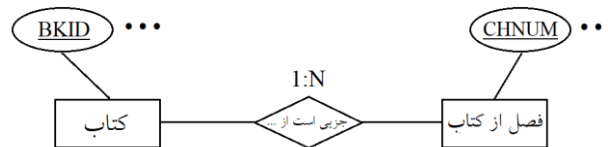
توجه: ستون PlayerID در جدول BodyPart علاوه بر اینکه کلید خارجی است، **جزء کلید کاندید** جدول BodyPart نیز هست.

توجه: اگر یک فیلد مثل بازیکن 1 از جدول Player حذف شود، چون رابطه **Composition** است، بنابراین جدول BodyPart نسبت به **تغییرات حذف (Delete)** در جدول Player به صورت

Cascade رفتار می‌کند و تمام رکوردهای مربوط به بازیکن 1 به طور کامل از جدول BodyPart نیز **حذف** می‌شود. چون بازیکن و اجزاء آن یا همه باهم هستن یا همه باهم نیستن. همچنین اگر یک فیلد مثل بازیکن 1 از جدول Player به مقدار 11 **آپدیت** شود، جدول BodyPart نسبت به **تغییرات آپدیت (Update)** در جدول Player به صورت **Cascade** رفتار می‌کند و فیلدهای مربوطه در ستون PlayerID از جدول BodyPart مقدار 11 می‌گیرد، قطعه کد مربوطه در داخل جدول فرعی BodyPart جهت اشاره به جدول اصلی Player به صورت زیر است:

Foreign key (PlayerID) References Player (PlayerID)
on delete Cascade
on update Cascade

مثال: هر فصل کتاب، جزیی از کتاب است.



Book (BKID , BKTITLE)
C.K.

Chapter (BKID, CHNUM , CHAPTITLE , NO – OF – PAGE)
C.K.

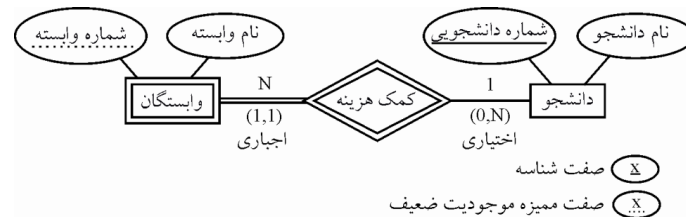
توجه: ستون BKID در جدول Chapter علاوه بر اینکه کلید خارجی است، **جزء کلید کاندید** جدول Chapter نیز هست. بقیه موارد، همانند مثال قبلی از Composition است. کتاب و فصل‌های آن یا همه باهم هستن یا همه باهم نیستن.

توجه: در عالم مادی نیز انسان کل با اشیاء جزء خود به طور هم‌زمان متولد می‌شود و سپس در انتهای آن به حیات مادی خود به همراه اشیاء جزء خود به طور هم‌زمان پایان می‌دهد، که این **ترکیب (حیات وابسته)** است. اما این انسان در طول حیات خود در گروه‌ها، تیم‌ها و کلاس‌های مختلفی شرکت می‌کند، که این **تجمع (حیات مستقل)** است. اصولاً بعد از هر **تجمع** و راهپیمایی هرکس به طور **مستقل** میره سر خونه زندگی خودش.

توجه: رابطه بخشی به عنوان رابطه «جزیی است از» یا **IS-A PART-OF** یا «دارد» یا **HAS** یا «شامل است» یا **CONTAINS** شناخته می‌شود.

تفاوت موجودیت جزء با موجودیت ضعیف

اول اینک: موجودیت جزء **می تواند** از خود شناسه مستقل (کلید کاندید مستقل) داشته باشد مثل رابطه **Aggregation**، ولی موجودیت ضعیف هیچگاه و تحت هیچ شرایطی از خود شناسه مستقل ندارد و فقط صفت ممیزه (کلید جزئی) دارد.



دوم اینک: با حذف موجودیت کل، الزاماً موجودیت جزء حذف نمی‌شود مثل رابطه **Aggregation**، به بیان دیگر، موجودیت جزء، لزوماً وابستگی وجودی به موجودیت کل ندارد و بستگی به **حیات وابسته و مستقل** دارد.

توجه مهم: به طور کلی در حالت‌های زیر کلید خارجی، جزء تشکیل دهنده کلید کاندید است:
۱- ارتباط چند به چند N:M: برای مثال در جدول SP ستون S# و P# به عنوان کلید خارجی، هر کدام به تنهایی جزء تشکیل دهنده کلید کاندید نیز هستند.

S#	Sname	City	S#	P#	QTY	P#	Pname	Color
S1	Sn1	C1	S1	P1	10	P1	Pn1	Red
S2	Sn2	C2	S1	P2	20	P2	Pn2	Blue
S3	Sn3	C3	S2	P1	30	P3	Pn3	Green

جدول S (تهیه کننده) جدول SP (تولید) جدول P (قطعه)

۲- نوع موجودیت ضعیف (ارتباط شناسا)

۳- ارتباط جزئی است از (IS A PART OF) در حالت **ترکیب** (Composition)

توجه: به عمل بررسی اجزا تشکیل دهنده یک موجودیت، **تجزیه (جداسازی)** گفته می‌شود. **موجودیت کل**، صفات و ساختار خود را دارد و **موجودیت جزء** نیز صفات و ساختار خاص خود را دارد. موجودیت کل، شامل موجودیت‌های جزء است. **تجزیه** به معنی حرکت از موجودیت کل به سمت موجودیت‌های جزء و در نهایت شناخت موجودیت‌های جزء است.

همانطور که گفتیم، در **مدل EER** روابط بیشتری مطرح شده است که «**مدل‌سازی رابطه بخشی (Aggregation و Composition)**» بررسی شد، در ادامه به تشریح «**مدل‌سازی رابطه**» می‌پردازیم.

وراثت» می‌پردازیم.

۲- مدل‌سازی رابطه وراثت

وراثت (Inheritance) در ساخت یافته (مدل رابطه‌ای)

وراثت در ساخت یافته (مدل رابطه‌ای) در ادامه توسط **تعمیم** و **تخصیص** مورد تحلیل و بررسی کامل قرار می‌گیرد.

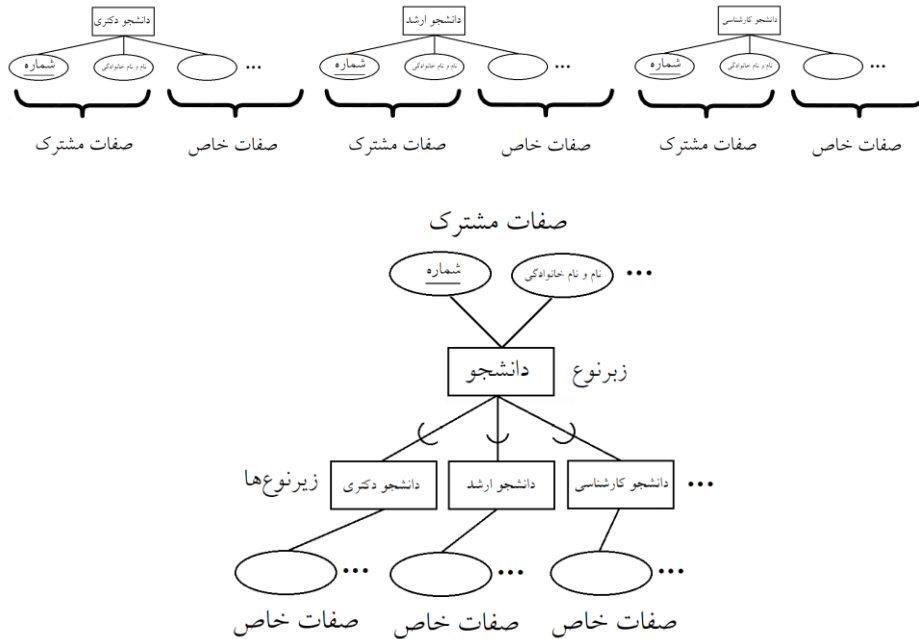
تعمیم (Generalization) و تخصیص (Specialization)

تعمیم و تخصیص به ارتباط بین نوع موجودیت خاص و نوع موجودیت عام گفته می‌شود. به فرآیند انتقال **صفات مشترک** و **شناسه مشترک** بین موجودیت‌های هم خانواده به سمت بالا **تعمیم** گفته می‌شود و به فرآیند انتقال **صفات متفاوت** بین موجودیت‌های هم خانواده به سمت پایین **تخصیص** گفته می‌شود. این مدل‌سازی به این صورت تفسیر معنایی می‌شود که، اطلاعات یک رکورد اطلاعاتی به دو تکه **اطلاعات عمومی** و **اطلاعات اختصاصی** تقسیم می‌شود. اطلاعات عمومی در **بخش تعمیم** (پدر) نگهداری می‌شود و اطلاعات اختصاصی در **بخش تخصیص** (فرزند) نگهداری می‌شود. در نهایت دو سطر درج شده در دو جدول پدر و فرزند بهم مرتبط می‌شوند. تعمیم و تخصیص را با یک **درخت** به نام درخت تعمیم و تخصیص نمایش می‌دهند. اگر d تعداد سطوح این درخت باشد، داریم: $d \geq 2$. **تعمیم** و **تخصیص** از نظر اختیاری و اجباری رابطه بالا به پایین «**پدر با فرزندان**» به دو نوع **کامل** و **ناقص** و از نظر کاردینالیته رابطه بالا به پایین «**پدر با فرزندان**» به دو نوع **رابطه غیرپوشا** (Disjoint) و **رابطه پوشا** (Overlap) است که در ادامه مورد بررسی قرار می‌گیرد.

توجه: لازمه تعمیم و تخصیص این است که حداقل دو نوع موجودیت داشته باشیم که **حداقل در صفت شناسه (کلید کاندید)**، **اشتراک** دارند $N \geq 1$ ، بدون برقراری این حداقل، تعمیم و تخصیص امکان‌پذیر نیست. N صفات مشترک بین n موجودیت است.

مثال: نوع موجودیت‌های **دانشجوی کارشناسی**، **دانشجوی ارشد** و **دانشجوی دکتری** را در نظر بگیرید. این نوع موجودیت‌ها، مجموعه‌ای از **صفات مشترک** دارند، مثل شماره دانشجویی، نام و نام خانوادگی و سال ورود و در عین حال، هریک از این انواع می‌توانند **صفات خاص** خود را هم داشته باشند. یک زیرنوع (ابرنوع) به نام نوع موجودیت دانشجوی در نظر می‌گیریم و مجموعه صفات مشترک بین آن نوع موجودیت‌ها را به این زیرنوع می‌دهیم که به آن **تعمیم** گفته می‌شود. آن نوع

موجودیت‌ها اینک هریک زیرنوع این زبرنوع می‌شوند، با صفات خاص خود، که به آن **تخصیص** گفته می‌شود.



توجه: به سلسله مراتب فوق **رابطه وراثت (ارث‌بری)** گفته می‌شود. رابطه وراثت به عنوان رابطه «هست یک یا is-a» یا «نوعی از is a kind of» یا «is a type of» نیز شناخته می‌شود.

تست - عبارت «is a type of» در مدل‌سازی داده‌ها، به چه منظوری استفاده می‌شود؟

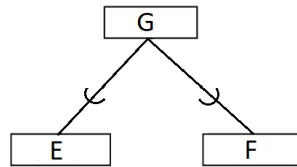
(مهندسی کامپیوتر-دولتی ۱۴۰۳)

- ۵) نشان‌دادن رابطه یک به یک بین دو جدول در پایگاه داده
- ۶) نشان‌دادن روابط چند به چند بین جداول در پایگاه داده
- ۷) مشخص کردن انواع داده‌هایی که در یک ستون خاص می‌توانند ذخیره شوند.
- ۸) تعریف یک سلسله مراتب وراثت بین کلاس‌ها یا جداول، جایی که یک کلاس یا جدول خاص، زیرمجموعه‌ای از کلاس یا جدول دیگری است.

پاسخ - گزینه (۴) صحیح است.

روش عمومی نگاشت رابطه ISA یا وراثت به مدل رابطه‌ای

اگر در مسائل وراثت، **Disjoint** و **Overlap** روی مدل لحاظ نشود، نگاشت به مدل رابطه‌ای از روش عمومی انجام می‌شود.



در این حالت به $n+1$ جدول نیاز داریم. اگر پدر (ابرنوع یا زیرنوع) n فرزند (زیرنوع) داشته باشد یک جدول برای پدر در نظر می‌گیریم و برای هر یک از فرزندان هم یک جدول با صفتهای خاص هر فرزند در نظر گرفته می‌شود. همچنین کلید کاندید جدول پدر در جداول فرزند به عنوان **کلید خارجی** تعریف می‌گردد و هم‌زمان در جداول فرزند **کلید کاندید** هم می‌باشد. و یک طراحی به شکل مدل $n+1$ جدولی اما با تحمل سربار حاصل از عمل الحاق طبیعی میان جداول پدر و فرزند ایجاد می‌گردد.

توجه: اگر در مسائل وراثت، **Disjoint** و **Overlap** روی مدل لحاظ نشود، پیش فرض، مدل آزادتر یعنی **Overlap** اختیاری مدل سه جدولی است، روش عمومی هم معادل **Overlap** اختیاری مدل سه جدولی است.

مثال: نوع موجودیت دانشجو و گونه‌های خاص آن: دانشجوی «کارشناسی»، دانشجوی «ارشد» و دانشجوی «دکتری» را در نظر بگیرید. چهار رابطه طراحی به صورت زیر است:

STUD (STID , STNAME , ...)
C.K.

صفات مشترک در همه نمونه‌های نوع موجودیت دانشجو در عنوان رابطه STUD می‌آیند.

BSSTUD (STID , A , B , ...)
C.K. , F.K.

MSSTUD (STID , M , N , ...)
C.K. , F.K.

DOCSTUD (STID , X , Y , Z , ...)
C.K. , F.K.

می‌بینیم که در این روش هرگاه پدر E دارای n فرزند E_i باشد، $n+1$ جدول طراحی می‌شود. در هریک از سه جدول آخر، **صفات خاص** دانشجویان دوره **کارشناسی**، **ارشد** و **دکتری** در نظر گرفته می‌شوند و دیگر لزومی ندارد که صفات مشترک بین این گونه‌ها، که در همان جدول STUD آورده می‌شوند، در **جداول پایین‌دست** تکرار شوند، چون **گونه‌های دانشجو** این صفات را

از نوع موجودیت دانشجو به ارث می‌برند. البته کلید کاندید STUD باید در سه جدول دیگر آورده شود.

این مثال، روش عمومی نمایش مفاهیم زبرنوع (Supertype) و زیرنوع (Subtype) در مدل رابطه‌ای را نشان می‌دهد.

توجه: در جداول نشان‌دهنده هر زیرنوع، کلید خارجی جدول، کلید کاندید آن هم هست.

سوال: نوع موجودیت‌های F و G مفروضند. می‌خواهیم با استفاده از تکنیک تعمیم، نوع موجودیت E را بر اساس دو نوع موجودیت F و G، در مدل‌سازی منظور کنیم. در اینصورت نوع موجودیت‌های F و G باید:

(الف) حداقل یک صفت هیچ‌مقدار ناپذیر مشترک داشته باشند.

(ب) حداقل شناسه مشترک داشته باشند.

(ج) در دو صفت، که یکی از آنها شناسه باشد، مشترک باشند.

(د) در صفت شناسه و یک صفت چندمقداری، مشترک باشند.

پاسخ: گزینه «ب» صحیح است.

سوال: اگر N تعداد صفات مشترک بین n نوع موجودیت باشد، در تعمیم این n نوع موجودیت، یک شرط لازم کدام است؟

$$(1) N \geq 2$$

$$(2) N \geq 1$$

$$(3) N > 1$$

$$(4) N > 2$$

پاسخ: گزینه «۲» صحیح است.

تست - کدام مورد یک طراحی منطقی درست برای نمودار EER زیر است؟

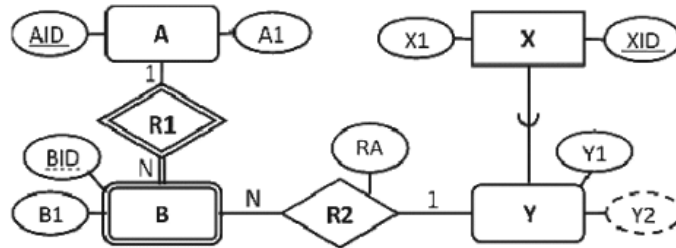
[توجه: در نمودار EER، از خط ممتد در زیر صفت برای نمایش کلید کاندید و از خط‌چین زیر

صفت برای نمایش صفت ممیزه یا کلید جزئی موجودیت ضعیف استفاده شده است. در طراحی

جداول، از خط ممتد در زیر صفت(ها) برای نمایش کلید اصلی و از خط‌چین برای نمایش کلید

خارجی استفاده شده است.]

(مهندسی کامپیوتر-دولتی-۱۴۰۲)



$A(\underline{AID}, A1)$ $X(\underline{XID}, X1)$
 $B(\underline{AID}, \underline{BID}, B1, \underline{XID}, RA)$ $Y(\underline{XID}, Y1)$ (۱)

$A(\underline{AID}, A1)$ $X(\underline{XID}, X1)$
 $B(\underline{AID}, \underline{BID}, B1)$ $Y(\underline{XID}, \underline{AID}, \underline{BID}, Y1, Y2, RA)$ (۲)

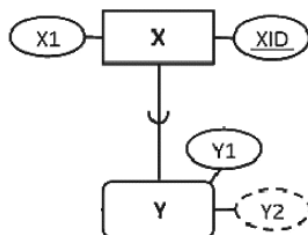
$A(\underline{AID}, A1)$ $X(\underline{XID}, X1)$
 $B(\underline{BID}, B1, \underline{XID}, RA)$ $Y(\underline{XID}, Y1)$ (۳)

$A(\underline{AID}, A1)$ $X(\underline{XID}, X1)$
 $B(\underline{AID}, \underline{BID}, B1)$ $Y(\underline{XID}, Y1)$ $R2(\underline{AID}, \underline{BID}, \underline{XID}, RA)$ (۴)

پاسخ: گزینه «۱» صحیح است.

توجه مهم: جهت تبدیل مدل ER به مدل مدل رابطه‌ای از سمت و موجودیتی شروع کنید که ورودی صفت از موجودیت دیگری نداشته باشد.

مدل تحلیل (وراثت عمومی)



توجه: در شکل فوق و بر اساس نوع نمایش و نشانه‌گذاری‌ها، صفت Y2 مشتق (پویا) است.
مدل طراحی (مدل رابطه‌ای)



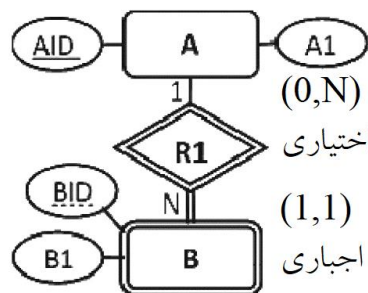
توجه: در وراثت عمومی، **کلید کاندید** جدول پدر (جدول X) در جدول فرزند (جدول Y) به عنوان **کلید خارجی** تعریف می‌گردد و هم‌زمان در جدول فرزند **کلید کاندید** هم می‌باشد.

توجه: کلید کاندید جدول X برابر (XID) است.

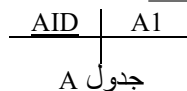
توجه: کلید کاندید جدول Y برابر (XID) است.

توجه: توصیه می‌گردد صفت مشتق (پویا) در جداول بانک اطلاعات مورد استفاده قرار نگیرد، بنابراین صفت مشتق Y2 در مدل EER در جدول Y به عنوان ستون نیامده است.

مدل تحلیل:

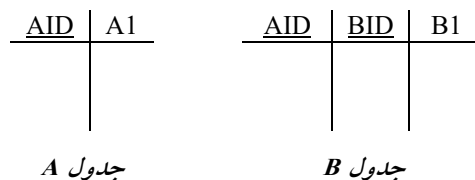


مدل طراحی جدول A به صورت A(AID, A1) است.



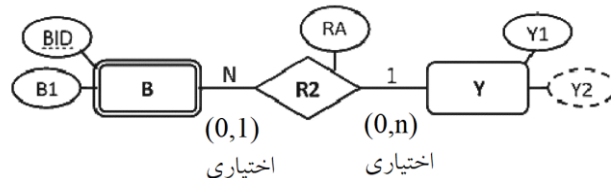
توجه: صفت AID کلید **موجودیت قوی** A و صفت BID، **صفت ممیزه موجودیت ضعیف** B است. صفت ممیزه موجودیت ضعیف توسط نقطه چین نشان داده می‌شود.

مدل طراحی



توجه: **کلید کاندید** جدول یک یعنی **موجودیت قوی** در جدول چند یعنی **موجودیت ضعیف** به عنوان **کلید خارجی** تعریف می‌گردد. کلید کاندید جدول A برابر (AID) است.

توجه: کلید کاندید جدول چند یعنی **موجودیت ضعیف** برابر ترکیب **کلید خارجی** و **صفت ممیزه** در جدول موجودیت ضعیف است. کلید کاندید جدول B برابر (AID,BID) است.
مدل تحلیل



توجه: همانطور که قبل تر گفتیم کلید کاندید جدول موجودیت Y صفت (XID) است.

مدل طراحی

همانطور که گفتیم، جدول Y به صورت $(Y1, XID)$ و جدول B به صورت $(B1, BID, AID)$ است.

<u>AID</u>	<u>BID</u>	B1
جدول B		

<u>XID</u>	Y1
جدول Y	

توجه: کلید کاندید (XID) جدول یک یعنی موجودیت Y در جدول چند یعنی موجودیت B به عنوان کلید خارجی (XID) تعریف می گردد، همچنین صفت RA در جدول B مستتر می شود:

<u>AID</u>	<u>BID</u>	B1	<u>XID</u>	RA
جدول B				

توجه: کلید کاندید جدول چند یعنی موجودیت B برابر همان کلید کاندید سابق در جدول موجودیت B است. کلید کاندید جدول B برابر (AID,BID) است.

نگاشت رابطه ISA یا وراثت به مدل رابطه‌ای

اجباری و اختیاری رابطه بالا به پایین «پدر با فرزندان» به دو صورت زیر است:

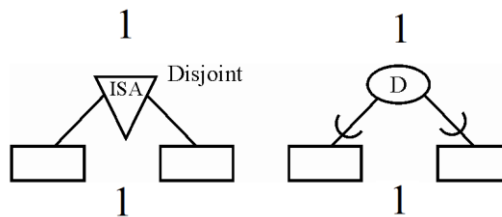
- رابطه **اجباری** یا **کامل** یا **کلی** (Total) با نماد خط مضاعف عمودی

توجه: بدین معنی که هر پدر **باید** حداقل یک فرزند داشته باشد. به عبارت دیگر تمام نمونه‌های پدر **باید** به طور اجباری در رابطه با فرزند شرکت کنند.

- رابطه **اختیاری** یا **ناقص** یا **جزئی** (Partial) با نماد خط عمودی

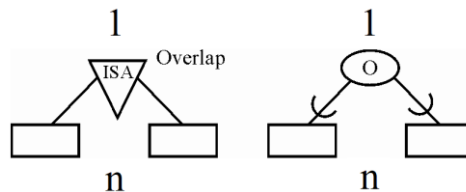
توجه: بدین معنی که هر پدر می‌تواند حداقل صفر فرزند داشته باشد. به عبارت دیگر نمونه‌های پدر می‌توانند به طور اختیاری در رابطه با فرزند شرکت نکنند.

کاردینالیتی رابطه بالا به پایین «پدر با فرزندان» به دو صورت زیر است:
- رابطه غیرپوشا یا تخصیص مجزا (Disjoint یا 1:1 یا یک به یک)



توجه: بدین معنی که هر پدر حداکثر یک فرزند دارد.

- رابطه پوشا یا تخصیص غیرمجزا (Overlap یا 1:n یا یک به چند)



توجه: بدین معنی که هر پدر حداکثر چند فرزند دارد.

توجه: رابطه پایین به بالا «فرزندان با پدر» همواره و تحت هر شرایطی، فقط و فقط صورت رابطه اجباری یا کامل یا کلی (Total) با نماد خط مضاعف است که اغلب روی شکل با همان نماد تک خط ساده نشان داده می‌شود. اجباری است چون در وراثت، فرزند بدون پدر نداریم. همچنین کاردینالیتی رابطه پایین به بالا «فرزندان با پدر» همواره فقط و فقط به صورت رابطه 1:1 (یک به یک) است. بدین معنی که هر فرزند حداقل و حداکثر یک پدر دارد.

توجه: نگاشت رابطه ISA یا وراثت به مدل رابطه‌ای به **چهار مدل** زیر انجام می‌شود:

الف) مدل تحلیل کامل (اجباری) و Disjoint	ب) مدل تحلیل ناقص (اختیاری) و Disjoint	ج) مدل تحلیل کامل (اجباری) و Overlap	د) مدل تحلیل ناقص (اختیاری) و Overlap

توجه: در ادامه، ابتدا انواع مدل‌های **تک جدولی**، **دو جدولی** و **سه جدولی** معرفی می‌شود؛ سپس چهار مدل فوق مورد بررسی قرار می‌گیرد.

مدل تک جدولی

در این مدل همه صفات موجودیت پدر (زیر نوع) و فرزند (زیر نوع) در یک جدول قرار می‌گیرد.

مدل دو جدولی

در این مدل همه صفات موجودیت پدر در جداول فرزند قرار داده می‌شود، همچنین کلید کاندید هر جدول فرزند همان کلید کاندید پدر است.

مدل سه جدولی

در این مدل به $n+1$ جدول نیاز داریم. اگر پدر (زیر نوع) n فرزند (زیر نوع) داشته باشد، یک جدول برای پدر در نظر می‌گیریم و برای هر یک از فرزندان هم یک جدول با صفات خاص هر فرزند در نظر گرفته می‌شود. همچنین **کلید کاندید** جدول پدر در جداول فرزند به عنوان **کلید خارجی** تعریف می‌گردد و هم‌زمان در جداول فرزند **کلید کاندید** هم می‌باشد.

توجه: در بررسی سربار جداول، وجود **مقادیر NULL افزایشی** (مصرف حافظه) **غیرقابل تحمل‌ترین نوع سربار**، سپس وجود **سربار افزونگی طبیعی** (مصرف حافظه) در جایگاه بعدی و در نهایت **سربار الحاق طبیعی** (مصرف CPU به دلیل عملیات ضرب) در جایگاه آخر به عنوان **قابل تحمل‌ترین نوع سربار** است.

توجه: جهت به خاطر سپاری مدل‌های مختلف نگاشت، موارد زیر را در نظر بگیرید:

مورد اول: سربار مقادیر NULL افزایشی از پدر بدون فرزند حاصل می‌شود و این حالت فقط و فقط زمانی ایجاد می‌شود که مدل **تک جدولی اختیاری** یا **دو جدولی اختیاری** باشد.

مورد دوم: سربار افزونگی طبیعی از **تکرار رکورد پدر** در جداول طرفین حاصل می‌شود و این حالت فقط و فقط زمانی ایجاد می‌شود که **Overlap مدل دو جدولی** باشد.

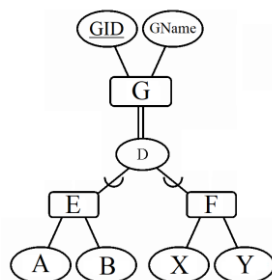
مورد سوم: سربرار الحاق طبیعی فقط و فقط از **مدل سه جدولی** حاصل می‌شود. مدل سه جدولی علاوه بر سربرار الحاق طبیعی، به دلیل تعریف **کلید خارجی**، افزودن تکنیکی (فنی) دارد که سربرار محسوب نمی‌شود.

جمع‌بندی: در مدل وراثت، **اختیاری بودن**، عامل تولید مقادیر **NULL افزایشی**، همچنین **Overlap بودن دو جدولی**، عامل **افزودن طبیعی** (تکرار رکوردها در جداول طرفین) و در نهایت **مدل سه جدولی** عامل ایجاد **سربرار الحاق طبیعی** است.

توجه مهم: مقادیر **NULL غیرافزایشی (بی‌عارضه)** با مقادیر **NULL افزایشی (باعارضه)** تفاوت دارد که در ادامه بررسی می‌شود، دقت کنید که مقادیر **NULL افزایشی غیرقابل تحمل‌ترین نوع سربرار** محسوب می‌شود.

الف) مدل تحلیل کامل (اجباری) و Disjoint

مدل تحلیل (نمودار ISA)



مدل طراحی (مدل رابطه‌ای)

مدل تک جدولی بهینه در شرایط خاص: دقت کنید که در شرایط خاص یعنی زمانی که (۱) تعداد صفات زبرنوع زیاد، (۲) تعداد زیرنوع‌ها کم و (۳) تعداد صفات زیرنوع‌ها کم باشد، امکان پیاده‌سازی تک جدولی دارد و بهینه هم هست.

سربرار مقادیر NULL افزایشی ندارد: چون **اختیاری بودن** عامل تولید مقادیر **NULL افزایشی** است.

سربرار افزودن طبیعی ندارد: چون **Overlap بودن دو جدولی** عامل **افزودن طبیعی** (تکرار رکورد) است.

سربرار الحاق طبیعی ندارد: چون مدل سه جدولی عامل ایجاد سربرار الحاق طبیعی میان جداول زبرنوع و زیرنوع در وراثت است.

صفات مشترک		صفات خاص E		صفات خاص F	
GID	GName	A	B	X	Y
GID1	GN1	A1	B1	NULL	NULL
GID2	GN2	A2	B2	NULL	NULL
GID3	GN3	NULL	NULL	X3	Y3

جدول GEF

توجه: مقادیر NULL در جدول فوق از نوع NULL غیرافزایشی است و NULL افزایشی نیست، چون **اختیاری بودن** عامل تولید مقادیر **NULL افزایشی** است. این اختیاری بودن است که بر طبل تولید NULL افزایشی می‌کوبد.

مدل دو جدولی بهینه: امکان پیاده‌سازی دو جدولی دارد و بهینه هم هست.
سربار مقادیر NULL افزایشی ندارد: چون **اختیاری بودن** عامل تولید مقادیر **NULL افزایشی** است.
سربار افزونگی طبیعی ندارد: چون **Overlap بودن دو جدولی** عامل **افزونگی طبیعی** (تکرار رکورد) است.
سربار الحاق طبیعی ندارد: چون مدل سه جدولی عامل ایجاد سربار الحاق طبیعی میان جداول زیرنوع و زیرنوع در وراثت است.

<u>GID</u>	GName	A	B	<u>GID</u>	GName	X	Y
GID1	GN1	A1	B1	GID3	GN3	X3	Y3
GID2	GN2	A2	B2				

جدول F

جدول E

توجه: در مدل دوجداولی، کل صفات پدر، در جداول فرزند قرار داده می‌شود، **کل صفات باباشو** بیارید جلوی چشم بچه‌هاش.
مدل سه جدولی بهینه با تحمل سربار الحاق طبیعی: امکان پیاده‌سازی سه جدولی دارد و بهینه هم هست.
سربار مقادیر NULL افزایشی ندارد: چون در مدل سه جدولی جداول زیرنوع و زیرنوع ازهم جدا هستند.
سربار افزونگی طبیعی ندارد: چون در مدل سه جدولی جداول زیرنوع و زیرنوع ازهم جدا هستند.
سربار الحاق طبیعی دارد: چون مدل سه جدولی عامل ایجاد سربار الحاق طبیعی میان جداول زیرنوع و زیرنوع در وراثت است.

GID	GName
GID1	GN1
GID2	GN2
GID3	GN3

جدول G

GID	A	B
GID1	A1	B1
GID2	A2	B2

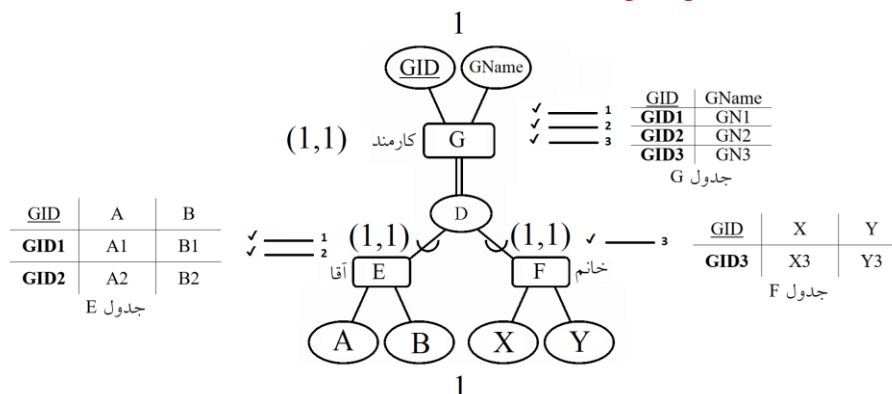
جدول E

GID	X	Y
GID3	X3	Y3

جدول F

توجه: بهینه‌ترین مدل طراحی برای مدل تحلیل کامل (اجباری) و Disjoint مدل دو جدولی است.

توجه: دقت کنید که کلید خارجی جدول زیرنوع، کلید کاندید آن هم هست.
مدل شماتیک مدل تحلیل کامل (اجباری) و Disjoint به صورت زیر است:



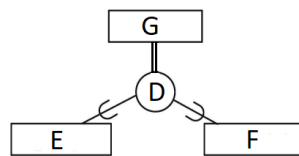
جمع‌بندی: چون رابطه بالا به پایین پدر با فرزندان از نوع اجباری است، پس هر رکورد از پدر باید حداقل یک فرزند داشته باشد. برای مثال رکورد GID1,GN1 حداقل یک فرزند مثل A1,B1 از موجودیت E دارد. (در مدل یک، دو و سه جدولی صادق است).

جمع‌بندی: چون کاردینالیتی رابطه بالا به پایین پدر با فرزندان از نوع Disjoint یا 1:1 است، پس هر رکورد از پدر فقط و فقط می‌تواند یک فرزند داشته باشد. برای مثال رکورد GID1,GN1 فقط و فقط می‌تواند یک فرزند مثل A1,B1 از موجودیت E یا F داشته باشد. (در مدل یک، دو و سه جدولی صادق است).

نتیجه: قید محدودیت رابطه بالا به پایین برابر (1,1) است، بدین معنی که هر پدر حداقل و حداکثر یک فرزند دارد، همچنین قید محدودیت رابطه پایین به بالا برابر (1,1) است، بدین معنی که هر فرزند حداقل و حداکثر یک پدر دارد.

توجه: اگر e مجموعه نمونه‌های E ، f مجموعه نمونه‌های F و g مجموعه نمونه‌های G باشد، آنگاه $\text{card}(g) = \text{card}(e) + \text{card}(f)$ است. یعنی $3 = 2 + 1$ چون هر رکوردی که در پدر هست، **باید** فقط و فقط یک نمونه از آن در یکی از فرزندان هم باشد.

سوال: با توجه به نمودار زیر :



اگر e مجموعه نمونه‌های E ، f مجموعه نمونه‌های F و g مجموعه نمونه‌های G باشد:

۱) $\text{card}(g) \geq \text{card}(e) + \text{card}(f)$

۲) $\text{card}(g) = \text{card}(e) + \text{card}(f)$

۳) $\text{card}(g) \neq \text{card}(e) + \text{card}(f)$

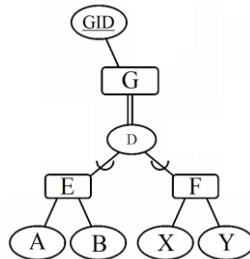
۴) $\text{card}(g) \leq \text{card}(e) + \text{card}(f)$

پاسخ: گزینه «۲» صحیح است.

مثال: در دنیای واقعی، رابطه بالا به پایین شماره شاسی یک اتومبیل (پدر) با نوع اتومبیل (فرزندان) از نوع **اجباری** است، بنابراین یک رکورد شماره شاسی، **باید** حداقل یک فرزند مثل دومحور (2WD) یا چهارمحور (4WD) داشته باشد.

مثال: کاردینالیتی رابطه بالا به پایین شماره شاسی یک اتومبیل (پدر) با نوع اتومبیل (فرزندان) به صورت Disjoint یا 1:1 است، بنابراین یک رکورد شماره شاسی، فقط و فقط می‌تواند یک فرزند از دومحور (2WD) یا چهارمحور (4WD) داشته باشد. نمی‌شود یک شماره شاسی مربوط به دو اتومبیل مختلف باشد.

سوال: در نمودار زیر، نوع موجودیت G فقط یک صفت شناسه دارد. طراحی با چند رابطه (جدول) بهتر است انجام شود تا کمترین هیچ‌مقدار (NULL) و کمترین افزونگی پدید آید؟



(۱) یک رابطه

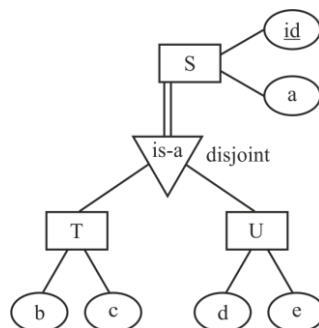
(۲) دو رابطه

(۳) سه رابطه

(۴) طراحی با یک یا دو رابطه باهم فرقی ندارند ولی از سه رابطه بهتر است.

پاسخ: گزینه «۲» صحیح است. شکل صورت سوال، مدل تحلیل کامل (اجباری) و Disjoint است که در مدل سه جدولی افزونگی طبیعی ندارد، اما به واسطه تعریف کلید خارجی بین جداول افزونگی تکنیکی دارد، خواسته سوال کمترین افزونگی (چه طبیعی و چه تکنیکی) است. همچنین در مدل تک جدولی NULL افزایشی ندارد، اما NULL غیرافزایشی دارد که خواسته سوال کمترین NULL (چه افزایشی و چه غیرافزایشی) است.

تست - قطعه زیر از یک نمودار موجودیت- رابطه را در نظر بگیرید: (مهندسی IT- دولتی ۹۶)



کدام یک از مجموعه رابطه‌های زیر برای تبدیل این قطعه به مدل رابطه‌ای مناسب‌ترین است؟

(۱) $T(id,a,b,c)$, $U(id,a,d,e)$

(۲) $S(id,a)$, $T(id,b,c)$, $U(id,d,e)$

(۳) مقدار Type یکی از سه مقدار S,T یا U می‌باشد. $R(id,a,b,c,d,e,Type)$

(۴) $S(id,a,b,c,d,e,isT,isU)$ دو مقدار isT و isU مقادیر True یا False را می‌پذیرند.

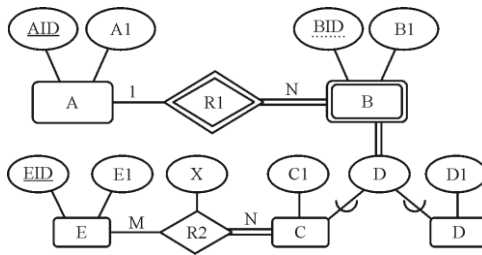
پاسخ - گزینه (۱) صحیح است. بهینه‌ترین مدل طراحی برای مدل تحلیل کامل (اجباری) و Disjoint مدل دو جدولی است. کل صفات باباشو بیارید جلوی چشم بچه‌هاش.

تست - با فرض وجود نمودار EER زیر کدام یک از رابطه‌های زیر به درستی طراحی نشده است؟

(مهندسی فناوری اطلاعات-دولتی ۹۵)

A (AID, A1) B (BID, B1) E (EID, E1, AID, BID)

C (AID, BID, B1, C1) D (AID, BID, B1, D1) R2 (AID, BID, EID, X)



A, B (۱)

B, E (۲)

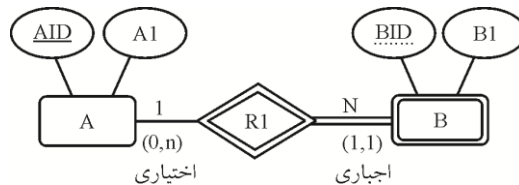
C, D (۳)

E, C (۴)

پاسخ - گزینه (۲) صحیح است.

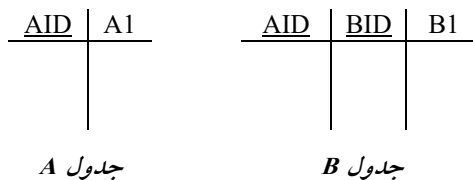
توجه مهم: جهت تبدیل مدل EER به مدل مدل رابطه‌ای از سمت و موجودیتی شروع کنید که ورودی صفت از موجودیت دیگری نداشته باشد.

مدل تحلیل:



توجه: در شکل فوق صفت AID کلید موجودیت قوی A و صفت BID، صفت ممیزه موجودیت ضعیف B است.

مدل طراحی:

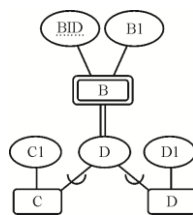


توجه: کلید کاندید جدول یک یعنی **موجودیت قوی** در جدول چند یعنی **موجودیت ضعیف** به عنوان کلید خارجی تعریف می‌گردد.

توجه: کلید کاندید جدول چند یعنی **موجودیت ضعیف** برابر ترکیب **کلید خارجی** و **صفت متمیزه** در جدول موجودیت ضعیف است. یعنی کلید کاندید جدول B برابر (AID, BID) است.

توجه: طراحی نادرست جدول B در صورت سوال به صورت (BID, B1) است، که طراحی درست آن (AID, BID, B1) است. همچنین طراحی جدول A در صورت سوال درست است، بنابراین گزینه دوم پاسخ سوال است.

مدل تحلیل (نمودار (ISA))



مدل طراحی

طراحی جدول B به صورت (AID, BID, B1) است.

توجه: بهینه‌ترین مدل طراحی برای مدل تحلیل **کامل (اجباری)** و **Disjoint** مدل دو جدولی است. **کل صفات باباشو بیارید جلوی چشم بچه‌هاش.**

<u>AID</u>	<u>BID</u>	B1

جدول B

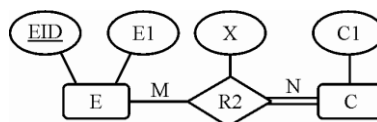
<u>AID</u>	<u>BID</u>	B1	C1

جدول C

<u>AID</u>	<u>BID</u>	B1	D1

جدول D

مدل تحلیل:



مدل طراحی:

توجه: در رابطه چند به چند، کلید کاندید طرفین در جدول پُل به عنوان کلید خارجی تعریف می‌شود. و صفت در ارتباط در جدول پُل مستتر می‌شود.

<u>EID</u>	E1	<u>AID</u>	<u>BID</u>	<u>EID</u>	X	<u>AID</u>	<u>BID</u>	B1	C1
جدول E		جدول R2				جدول C			

توجه: طراحی نادرست جدول E در صورت سوال به صورت (EID,E1,AID,BID) است، که طراحی درست آن (EID,E1) است.

تست - در نمودار E-R، اگر رابطه IS-A از نوع منفصل و کامل باشد، کدام مورد برای تبدیل نمودار به مدل رابطه‌ای مناسب‌تر است؟

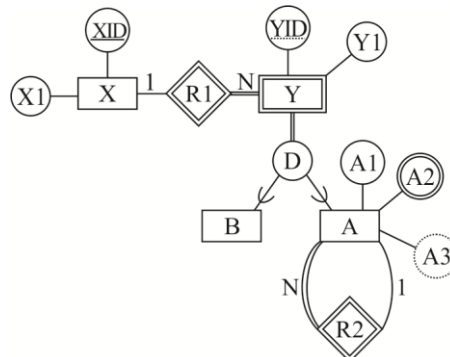
(مهندسی IT - دولتی ۹۸)

- کلید اصلی موجودیت پدر را به عنوان کلید خارجی به موجودیت‌های فرزند اضافه می‌کنیم.
- کلید اصلی موجودیت‌های فرزند را به عنوان کلید خارجی به موجودیت پدر اضافه می‌کنیم.
- برای موجودیت پدر، رابطه جداگانه ایجاد نمی‌کنیم و ویژگی‌های رابطه پدر را به موجودیت‌های فرزند اضافه می‌کنیم.
- موجودیت جداگانه‌ای ایجاد می‌شود که شامل کلید اصلی موجودیت پدر و کلید اصلی موجودیت‌های فرزند است.

پاسخ - گزینه (۳) صحیح است. بهینه‌ترین مدل طراحی برای مدل تحلیل کامل (اجباری) و Disjoint مدل دو جدولی است. کل صفات باباشو بیارید جلوی چشم بچه‌هاش.

تست - در نمودار EER زیر، طراحی منطقی رابطه یا رابطه‌های مربوط به موجودیت A، کدام مورد خواهد بود؟ (در گزینه‌های داده شده، زیر خط ممتد به معنای کلید اصلی و خط چین به معنای کلید خارجی است.)

(مهندسی کامپیوتر - دولتی ۹۹)



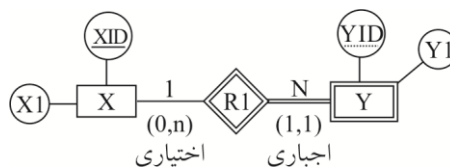
(۱) $A(\underline{YID}, \underline{YIDR2}, A1, A2, A3)$

(۲) $A(\underline{YID}, \underline{YIDR2}, A1, A2)$

(۳) $AA2(\underline{YID}, A2) \quad A(\underline{YID}, Y1, \underline{YIDR2}, A1)$

(۴) $AA2(\underline{XID}, \underline{YID}, A2) \quad A(\underline{XID}, \underline{YID}, Y1, \underline{XIDR}, \underline{YIDR2}, A1)$

گزینه (۴) صحیح است.



توجه: در مدل تحلیل فوق صفت XID کلید موجودیت قوی X و صفت YID، صفت ممیزه موجودیت ضعیف Y است. صفت ممیزه موجودیت ضعیف توسط نقطه چین نشان داده می‌شود.

مدل طراحی:

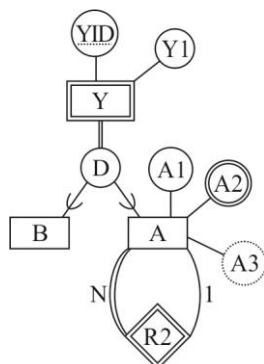
توجه: کلید کاندید جدول یک یعنی **موجودیت قوی** در جدول چند یعنی **موجودیت ضعیف** به عنوان **کلید خارجی** تعریف می‌گردد.

<u>XID</u>	X1	<u>XID</u>	<u>YID</u>	Y1

جدول X جدول Y

توجه: کلید کاندید جدول چند یعنی موجودیت ضعیف برابر ترکیب **کلید خارجی** و **صفت ممیزه** در جدول **موجودیت ضعیف** است. کلید کاندید جدول Y برابر (XID, YID) است.

مدل تحلیل (نمودار (ISA))



مدل طراحی (مدل رابطه‌ای)

طراحی جدول Y به صورت (XID, YID, Y1) است.

توجه: بهترین مدل طراحی برای مدل تحلیل کامل (اجباری) و Disjoint مدل دو جدولی است. کل صفات باباشو بیارید جلوی چشم بچه‌هاش.

XID	YID	Y1

جدول Y

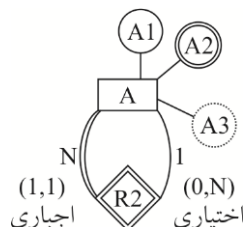
XID	YID	Y1	...

جدول B

XID	YID	Y1	A1	A2

جدول اولیه غیر نرمال A

مدل تحلیل:



توجه: در شکل فوق و بر اساس نوع نمایش و نشانه‌گذاری‌ها صفت A2 چند مقداری و صفت A3 مشتق (پویا) است.

مدل طراحی غیرنرمال:

توجه: در نگاشت رابطه یک به چند در یک موجودیت به مدل رابطه‌ای مستقل از اختیاری یا اجباری بودن موجودیت، موجودیت به یک جدول تبدیل می‌گردد، بنابراین در جدول A صفات (XIDR, YIDR2) به عنوان کلید خارجی به کلید کاندید (XID, YID) از همان جدول A ارجاع می‌کند.

<u>XID</u>	<u>YID</u>	Y1	A1	A2	<u>XIDR</u>	<u>YIDR2</u>
------------	------------	----	----	----	-------------	--------------

جدول غیرنرمال A

توجه: کلید کاندید جدول A برابر (XID, YID) است.

توجه: شاید این سوال در ذهن شما شکل بگیرد که در کلید خارجی، XIDR چرا XIDR2 نوشته نشده است، پاسخ این است که خب بهتر بود همین XIDR2 نوشته می‌شد که احتمالاً خطای حروفچینی سوال بوده است ولی مهم هم نیست چون مهم است کلید خارجی تعریف شود که شده است.

توجه: صفت مشتق A3 در مدل EER در جدول A به عنوان ستون نیامده است. در جدول A در گزینه چهارم هم صفت مشتق A3 نیامده است.

توجه: صفت A2 در مدل EER مطرح شده در صورت سوال بر اساس نوع نمایش و نشانه‌گذاری صفت چند مقداری است و همین عامل باعث شده است که بگوییم جدول A غیرنرمال است. که برای حل این مساله و حذف صفت چند مقداری A2 از جدول A، این جدول باید در سطح اول نرمال قرار بگیرد.

مدل طراحی غیرنرمال:

<u>XID</u>	<u>YID</u>	Y1	A1	A2	<u>XIDR</u>	<u>YIDR2</u>
------------	------------	----	----	----	-------------	--------------

جدول غیرنرمال A

توجه: جدول غیرنرمال A به دو جدول نرمال تجزیه می‌شود. کلید کاندید جدول غیرنرمال A و صفت چند مقداری در جدول AA2 درج می‌شود. همچنین کلید کاندید جدول AA2 به عنوان کلید خارجی، جهت ارتباط با جدول A تعریف می‌شود، به صورت زیر:

مدل طراحی نرمال:

XID	YID	A2
-----	-----	----

جدول نرمال AA2

XID	YID	Y1	A1	XIDR	YIDR2
-----	-----	----	----	------	-------

جدول نرمال A

توجه: کلید کاندید جدول A صفات (XID,YID) است.

توجه: کلید کاندید جدول AA2 صفات (XID,YID,A2) یعنی تمام کلید است.

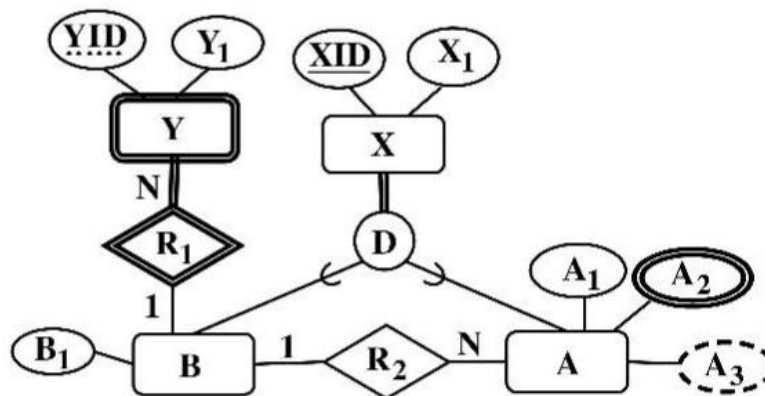
توجه: طراحی جدول A و AA2 در گزینه چهارم به ترتیب به صورت $A(XID, YID, Y1, XIDR, YIDR2, A1)$ و $AA2(XID, YID, A2)$ است.

توجه: نقطه چین بودن صفت XID در جدول A در گزینه چهارم، به این دلیل است که در نگاشت بالاتر یعنی یک به چند بین دو موجودیت X و Y کلید کاندید جدول X یعنی XID به عنوان کلید خارجی در جدول Y تعریف شده بود. در جداول کلید خارجی توسط نقطه چین یا خط چین نشان داده می شود.

تست - با توجه به نمودار EER زیر، کدام گزینه، طراحی منطقی صحیح موجودیت های A و Y را نمایش می دهد؟

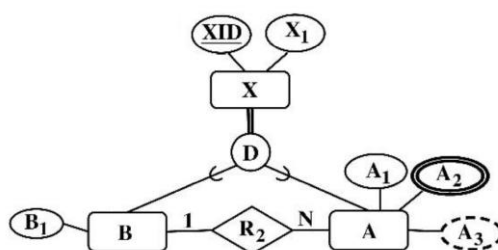
[در نمودار EER، از خط ممتد در زیر صفت برای نمایش شناسه و از خط چین برای نمایش صفت ممیزه یا کلید جزئی استفاده شده است. در طراحی جداول در گزینه ها، از خط ممتد در زیر صفت (ها) برای نمایش کلید اصلی و از خط چین برای نمایش کلید خارجی استفاده شده است.]

(مهندسی IT-دولتی ۱۴۰۱)



- (۱) $A(\underline{XID}, \underline{BXID}, A1, A2, A3)$ $Y(\underline{XID}, \underline{YID}, Y1)$
- (۲) $A(\underline{XID}, A1, A3, \underline{BXID})$ $AA2(\underline{XID}, A2)$ $Y(\underline{YID}, Y1, \underline{XID})$
- (۳) $A(\underline{XID}, A1, A2, A3)$ $Y(\underline{YID}, Y1)$
- (۴) $A(\underline{XID}, A1, \underline{BXID})$ $AA2(\underline{XID}, A2)$ $Y(\underline{XID}, \underline{YID}, Y1)$

گزینه (۴) صحیح است.
مدل تحلیل



مدل طراحی

طراحی جدول X به صورت (XID, X1) است.

توجه: بهترین مدل طراحی برای مدل تحلیل کامل (اجباری) و Disjoint مدل دو جدولی است. کل صفات باباشو بیارید جلوی چشم بچه‌هاش.

<u>XID</u>	X1

جدول X

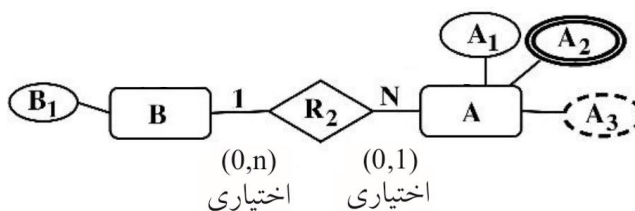
<u>XID</u>	X1	B1	...

جدول B

<u>XID</u>	X1	A1	A2

جدول غیرنرمال A

مدل تحلیل



توجه: در شکل فوق و بر اساس نوع نمایش و نشانه‌گذاری‌ها صفت A2 چند مقداری و صفت A3 مشتق (پویا) است. کلید کاندید جدول موجودیت A صفت (XID) است.

مدل طراحی

توجه: از قبل طراحی جدول B داریم، کلید کاندید (XID) جدول یک یعنی موجودیت B در جدول چند یعنی موجودیت A به عنوان کلید خارجی (BXID) تعریف می‌گردد.

XID	X ₁	B ₁
جدول B		

XID	X ₁	A ₁	A ₂	BXID
جدول غیرنرمال A				

توجه: صفت مشتق A3 در جدول A به عنوان ستون نمی‌آید.

توجه: کلید کاندید جدول B برابر (XID) است.

توجه: کلید کاندید جدول A برابر (XID) است.

مدل طراحی غیرنرمال:

توجه: صفت A2 در مدل EER مطرح شده در صورت سوال بر اساس نوع نمایش و نشانه‌گذاری صفت چند مقداری است و همین عامل باعث شده است که بگوییم جدول A غیرنرمال است. که برای حل این مساله و حذف صفت چند مقداری A2 از جدول A، این جدول باید در **سطح اول نرمال** قرار بگیرد.

XID	X ₁	A ₁	A ₂	BXID
جدول غیرنرمال A				

توجه: جدول غیرنرمال A به دو جدول نرمال تجزیه می‌شود. کلید کاندید جدول غیرنرمال A و صفت چند مقداری در جدول AA2 درج می‌شود. همچنین کلید کاندید جدول AA2 به عنوان کلید خارجی، جهت ارتباط با جدول A تعریف می‌شود، به صورت زیر:

مدل طراحی نرمال:

XID	A ₂
جدول نرمال AA2	

XID	X ₁	A ₁	BXID
جدول نرمال A			

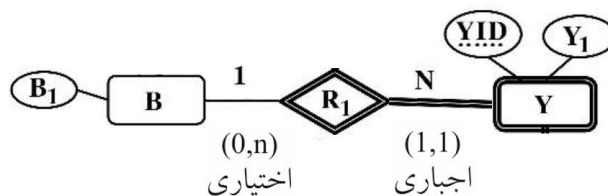
توجه: کلید کاندید جدول A صفت (XID) است.

توجه: کلید کاندید جدول AA2 صفات (XID, A2) یعنی **تمام کلید** است.

توجه: تجزیه در نرمال‌سازی یعنی دوری و دوستی، دور هستن یعنی دو جدول شدن، اما دوست هم هستن یعنی توسط تعریف کلید خارجی الحاق پذیر هم هستن و توسط عمل الحاق طبیعی همان جدول پایه ایجاد می‌گردد.

توجه: شاید این سوال در ذهن شما شکل بگیرد که چرا در گزینه‌ی چهارم، جدول A به صورت $A(\underline{XID}, A_1, BXID)$ آمده است و ستون X_1 جا افتاده است! پاسخ این است که گزینه چهارم در صورت سوال اصلی سنجش خطای حروفچینی داشته است و صحیح آن همان $A(\underline{XID}, X_1, A_1, BXID)$ است.

مدل تحلیل:



همانطور که گفتیم، مدل طراحی جدول B به صورت $B(\underline{XID}, X_1, B_1)$ است.

<u>XID</u>	X_1	B_1
جدول B		

توجه: صفت XID **کلید موجودیت قوی** B و صفت YID ، **صفت ممیزه موجودیت ضعیف** Y است. صفت ممیزه موجودیت ضعیف توسط نقطه چین نشان داده می‌شود.

مدل طراحی

توجه: کلید کاندید جدول یک یعنی **موجودیت قوی** در جدول چند یعنی **موجودیت ضعیف** به عنوان **کلید خارجی** تعریف می‌گردد.

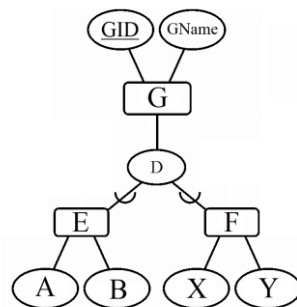
<u>XID</u>	X_1	B_1
جدول B		

<u>XID</u>	<u>YID</u>	Y_1
جدول Y		

توجه: کلید کاندید جدول چند یعنی **موجودیت ضعیف** برابر ترکیب **کلید خارجی** و **صفت** ممیزه در جدول **موجودیت ضعیف** است. کلید کاندید جدول Y برابر (XID, YID) است.

ب) مدل تحلیل ناقص (اختیاری) و Disjoint

مدل تحلیل (نمودار ISA)



مدل طراحی (مدل رابطه‌ای)

مدل تک جدولی غیربهمینه در شرایط خاص به دلیل افزایش مقدار NULL: دقت کنید که در شرایط خاص یعنی زمانی که (۱) تعداد صفات زیرنوع زیاد، (۲) تعداد زیرنوع‌ها کم و (۳) تعداد صفات زیرنوع‌ها کم باشد، امکان پیاده‌سازی تک جدولی دارد، اما بهمینه نیست.

سربار مقادیر NULL افزایشی دارد: چون **اختیاری بودن** عامل تولید مقادیر NULL افزایشی است، ارتباط زیرنوع با زیرنوع اختیاری است، پس ممکن است رکوردهای زیرنوع، فرزند نداشته باشد و به تبع مقادیر NULL افزایشی ایجاد می‌شود. مانند سطر 2. هرچند تعداد صفات زیرنوع‌ها هم جهت NULL سازی کم است.

سربار افزونگی طبیعی ندارد: چون **Overlap بودن دو جدولی** عامل **افزونگی طبیعی** (تکرار رکورد) است.

سربار الحاق طبیعی ندارد: چون مدل سه جدولی عامل ایجاد سربار الحاق طبیعی میان جداول زیرنوع و زیرنوع در وراثت است.

صفات مشترک		صفات خاص E		صفات خاص F	
GID	GName	A	B	X	Y
GID1	GN1	A1	A1	NULL	NULL
GID2	GN2	NULL	NULL	NULL	NULL
GID3	GN3	NULL	NULL	X3	Y3

جدول GEF

توجه: مقادیر NULL در جدول فوق از نوع NULL غیرافزایشی (سطر 1 و 3) و NULL افزایشی (سطر 2) است، چون **اختیاری بودن** عامل تولید مقادیر **NULL افزایشی** است. اختیاری بودن بر طبل تولید NULL افزایشی می‌کوبد.

مدل دو جدولی غیربهمینه به دلیل افزایش مقادیر NULL: امکان پیاده‌سازی دو جدولی دارد، اما بهمینه نیست.

سربار مقادیر NULL افزایشی دارد: چون **اختیاری بودن** عامل تولید مقادیر **NULL افزایشی** است، ارتباط زیرنوع با زیرنوع، ناقص **اختیاری** است، پس ممکن است زیرنوع، فرزند نداشته باشد و به تبع مقادیر NULL افزایشی ایجاد می‌شود. مانند سطر 2 جدول E.

سربار افزونگی طبیعی ندارد: چون **Overlap بودن دوجداولی** عامل **افزونگی طبیعی** (تکرار رکورد) است.

سربار الحاق طبیعی ندارد: چون مدل سه جدولی عامل ایجاد سربار الحاق طبیعی میان جداول زیرنوع و زیرنوع در وراثت است.

<u>GID</u>	GName	A	B
GID1	GN1	A1	B1
GID2	GN2	NULL	NULL

جدول E

<u>GID</u>	GName	X	Y
GID3	GN3	X3	Y3

جدول F

توجه: در مدل دوجداولی، کل صفات پدر، در جداول فرزند قرار داده می‌شود.

توجه: مقادیر NULL در جدول فوق از نوع NULL افزایشی است.

مدل سه جدولی بهمینه با تحمل سربار الحاق طبیعی: امکان پیاده‌سازی سه جدولی دارد و بهمینه هم هست.

سربار مقادیر NULL افزایشی ندارد: چون در مدل سه جدولی جداول زیرنوع و زیرنوع ازهم جدا هستند.

سربار افزونگی طبیعی ندارد: چون در مدل سه جدولی جداول زیرنوع و زیرنوع ازهم جدا هستند.

سربار الحاق طبیعی دارد: چون مدل سه جدولی عامل ایجاد سربار الحاق طبیعی میان جداول زیرنوع و زیرنوع در وراثت است.

<u>GID</u>	GName
GID1	GN1
GID2	GN2
GID3	GN3

جدول G

<u>GID</u>	A	B
GID1	A1	B1

جدول E

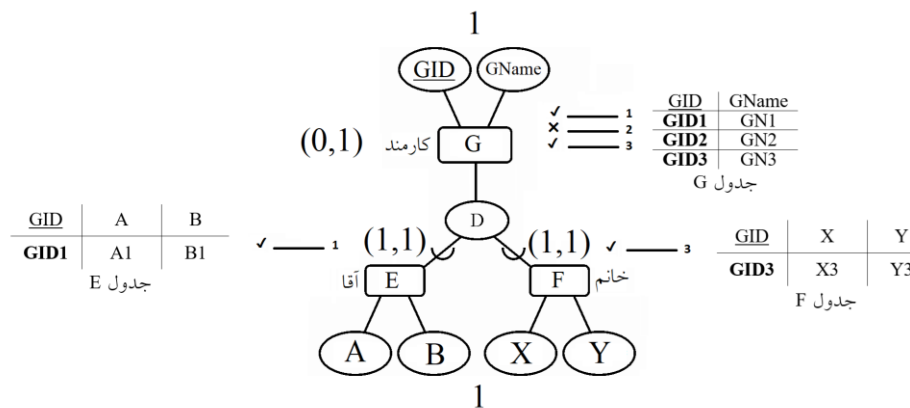
<u>GID</u>	X	Y
GID3	X3	Y3

جدول F

توجه: دقت کنید که کلید خارجی جدول زیرنوع، کلید کاندید آن هم هست.

توجه: دقت کنید که مقدار GID2 در جدول E یا F درج نشده است چون فرض شده است مقادیر صفات خاص آن برابر مقدار NULL است. بنابراین مدل سه جدولی سربار مقادیر NULL افزایشی را ندارد.

مدل شماتیک مدل تحلیل ناقص (اختیاری) و Disjoint به صورت زیر است:



جمع‌بندی: چون رابطه بالا به پایین پدر با فرزندان از نوع **اختیاری** است، پس ممکن است رکوردهای پدر، فرزند نداشته باشد. برای مثال رکورد GID2,GN2 هیچ فرزندی ندارد. (در مدل یک، دو و سه جدولی صادق است.)

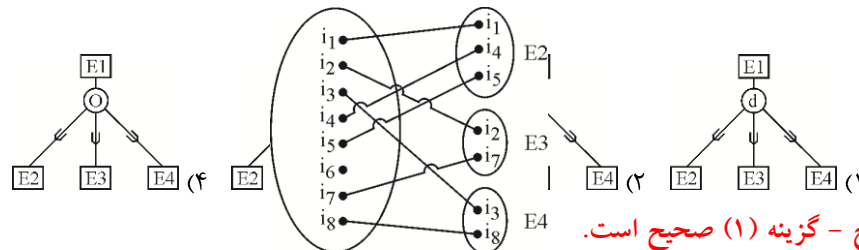
جمع‌بندی: چون کاردینالیتی رابطه بالا به پایین پدر با فرزندان از نوع **(Disjoint یا 1:1)** است، پس هر رکورد از پدر فقط و فقط می‌تواند یک فرزند داشته باشد. برای مثال رکورد GID1,GN1 فقط و فقط می‌تواند یک فرزند مثل A1,B1 از موجودیت E یا از موجودیت F داشته باشد. (در مدل یک، دو و سه جدولی صادق است.)

نتیجه: قید محدودیت رابطه بالا به پایین برابر (0,1) است، بدین معنی که هر پدر حداقل صفر و حداکثر یک فرزند دارد، همچنین قید محدودیت رابطه پایین به بالا برابر (1,1) است، بدین معنی که هر فرزند حداقل و حداکثر یک پدر دارد.

توجه: اگر مجموعه نمونه‌های E، f مجموعه نمونه‌های F و g مجموعه نمونه‌های G باشد، آنگاه $\text{card}(g) \geq \text{card}(e) + \text{card}(f)$ است. یعنی $3 \geq 1+1$ چون هر رکوردی که در پدر هست، ممکن است نمونه‌ای از آن در فرزندان نباشد.

تست - نمونه‌هایی از چهار نوع موجودیت E1، E2، E3 و E4 در شکل زیر نمایش داده شده است. کدام مورد بهترین نمودار EER معرف محیط است؟

(مهندسی کامپیوتر - دولتی ۹۸)



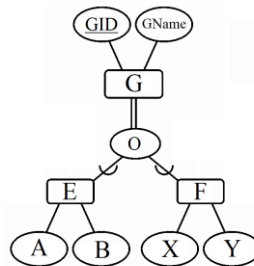
پاسخ - گزینه (۱) صحیح است.

نمونه موجودیت ۱ در موجودیت ۱. یعنی E1. رنه موجودیت‌های فرزند یعنی موجودیت‌های E2، E3 و E4 وارد رابطه نشده است. و از آنجا که اگر حداقل یکی از نمونه‌های موجودیت پدر در رابطه با فرزندان شرکت نکرده باشد، آن رابطه **اختیاری** است، بنابراین نمونه‌های موجودیت E1 یعنی پدر به صورت **اختیاری** وارد رابطه با فرزندان یعنی موجودیت‌های E2، E3 و E4 شده‌اند. بنابراین گزینه‌های دوم و سوم پاسخ سوال نیستند.

توجه: از آنجا که کاردینالیتی رابطه پدر با فرزندان یک به یک است، بنابراین رابطه پدر با فرزندان حالت **Disjoint** است و گزینه اول پاسخ سوال است.

ج) مدل تحلیل کامل (اجباری) و Overlap

مدل تحلیل (نمودار ISA)



مدل طراحی (مدل رابطه‌ای)

مدل تک جدولی بهینه در شرایط خاص: دقت کنید که در شرایط خاص یعنی زمانی که (۱) تعداد صفات زیرنوع زیاد، (۲) تعداد زیرنوع‌ها کم و (۳) تعداد صفات زیرنوع‌ها کم باشد، امکان پیاده‌سازی تک جدولی دارد و بهینه هم هست.

سربار مقادیر NULL افزایشی ندارد: چون **اختیاری بودن** عامل تولید مقادیر NULL افزایشی است.

سربار افزونگی طبیعی ندارد: چون **Overlap بودن دو جدولی** عامل **افزونگی طبیعی** (تکرار رکورد) است.

سربار الحاق طبیعی ندارد: چون مدل سه جدولی عامل ایجاد سربار الحاق طبیعی میان جداول زیرنوع و زیرنوع در وراثت است.

صفات مشترک		صفات خاص E		صفات خاص F	
GID	GName	A	B	X	Y
GID1	GN1	A1	B1	X1	Y1
GID2	GN2	A2	B2	NULL	NULL
GID3	GN3	A3	B3	X3	Y3

جدول GEF

توجه: مقادیر NULL در جدول فوق از نوع NULL غیرافزایشی است و NULL افزایشی نیست، چون **اختیاری بودن** عامل تولید مقادیر NULL افزایشی است.

مدل دو جدولی غیربهمینه به دلیل افزایش افزونگی: امکان پیاده‌سازی دو جدولی دارد اما بهمینه نیست.

سربار مقادیر NULL افزایشی ندارد: چون **اختیاری بودن** عامل تولید مقادیر NULL افزایشی است.

سربار افزونگی طبیعی دارد: چون **Overlap بودن دو جدولی** عامل **افزونگی طبیعی** (تکرار رکورد) است. ارتباط Overlap است، پس ممکن است یک رکورد خاص زیرنوع G در دو جدول E و F تکرار شود.

سربار الحاق طبیعی ندارد: چون مدل سه جدولی عامل ایجاد سربار الحاق طبیعی میان جداول زیرنوع و زیرنوع در وراثت است.

GID	GName	A	B	GID	GName	X	Y
GID1	GN1	A1	B1	GID1	GN1	X1	Y1
GID2	GN2	A2	B2	GID3	GN3	X3	Y3
GID3	GN3	A3	B3				

جدول E

جدول F

توجه: در مدل دوجداولی، کل صفات پدر، در جداول فرزند قرار داده می‌شود.

توجه: دقت کنید که تکرار GID1 یا GID3 همراه نام در دو جدول E و F به صورت مدل دو جدولی، **سربار افزونگی طبیعی** محسوب می‌شود.

مدل سه جدولی بهینه با تحمل سربار الحاق طبیعی: امکان پیاده‌سازی سه جدولی دارد و بهینه هم هست.

سربار مقادیر NULL افزایشی ندارد: چون در مدل سه جدولی جداول زیرنوع و زیرنوع ازهم جدا هستند.

سربار افزونگی طبیعی ندارد: چون در مدل سه جدولی جداول زیرنوع و زیرنوع ازهم جدا هستند.

سربار الحاق طبیعی دارد: چون مدل سه جدولی عامل ایجاد سربار الحاق طبیعی میان جداول زیرنوع و زیرنوع در وراثت است.

GID	GName
GID1	GN1
GID2	GN2
GID3	GN3

جدول G

<u>GID</u>	A	B
GID1	A1	B1
GID2	A2	B2
GID3	A3	B3

جدول E

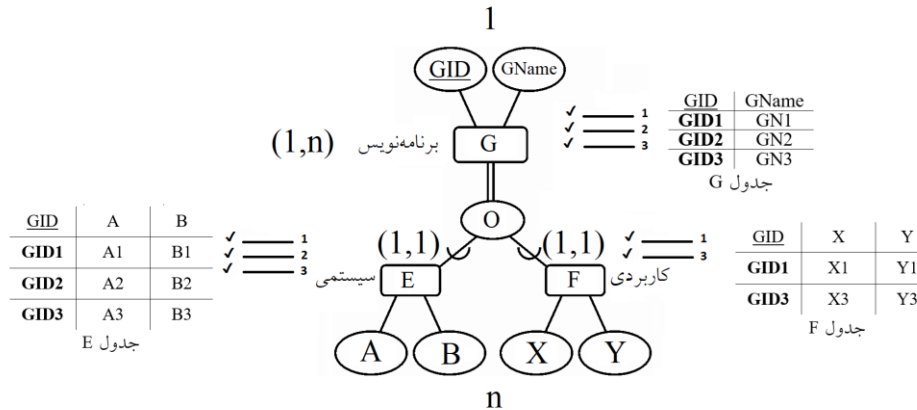
<u>GID</u>	X	Y
GID1	X1	Y1
GID3	X3	Y3

جدول F

توجه: دقت کنید که کلید خارجی جدول زیرنوع، کلید کاندید آن هم هست.

توجه: دقت کنید که تکرار GID1 یا GID3 در دو جدول E و F به صورت مدل سه جدولی، افزونگی طبیعی محسوب نمی‌شود بلکه افزونگی تکنیکی (فنی) به دلیل تعریف کلید خارجی محسوب می‌شود.

مدل شماتیک مدل تحلیل کامل (اجباری) و Overlap به صورت زیر است:



جمع‌بندی: چون رابطه بالا به پایین پدر با فرزندان از نوع **اجباری** است، پس هر رکورد از پدر **باید** حداقل یک فرزند داشته باشد. برای مثال رکورد GID1,GN1 حداقل یک فرزند مثل A1,B1 از موجودیت E دارد. (در مدل یک، دو و سه جدولی صادق است.)

جمع‌بندی: چون کاردینالیتی رابطه بالا به پایین پدر با فرزندان از نوع **(Overlap یا 1:n)** است، پس هر رکورد از پدر ممکن است بیش از یک فرزند داشته باشد. برای مثال رکورد GID1,GN1 می‌تواند دو فرزند A1,B1 از موجودیت E و X1,Y1 از موجودیت F داشته باشد. (در مدل یک، دو و سه جدولی صادق است.)

نتیجه: قید محدودیت رابطه بالا به پایین برابر (1,n) است، بدین معنی که هر پدر حداقل یک و حداکثر چند فرزند دارد، همچنین قید محدودیت رابطه پایین به بالا برابر (1,1) است، بدین معنی که هر فرزند حداقل و حداکثر یک پدر دارد.

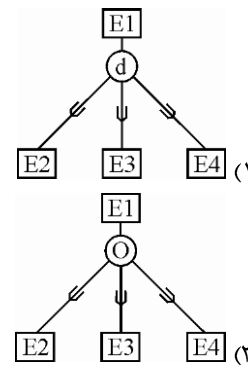
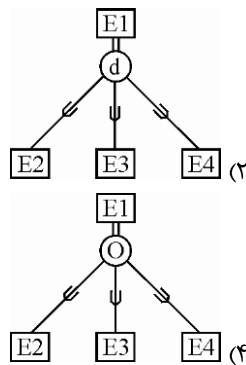
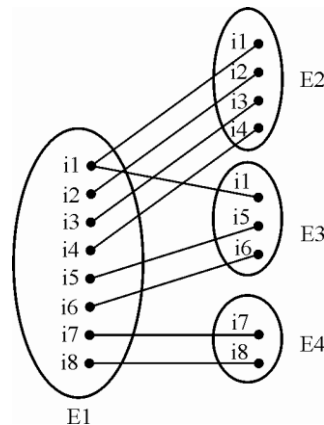
توجه: اگر e مجموعه نمونه‌های E، f مجموعه نمونه‌های F و g مجموعه نمونه‌های G باشد، آنگاه $\text{card}(g) \leq \text{card}(e) + \text{card}(f)$ است. یعنی $3 \leq 3 + 2$ چون هر رکوردی که در پدر هست، ممکن است چند نمونه از آن در فرزندان باشد.

سوال مهم: بر اساس مدل تحلیل کامل (اجباری) و Overlap، طراحی پایگاه داده رابطه‌ای با چند رابطه (جدول) باید انجام شود تا هیچ مقدار (NULL) پدید نیاید؟

- (۱) یک
- (۲) دو
- (۳) سه
- (۴) دو و سه

پاسخ: گزینه «۴» صحیح است. در مدل تحلیل کامل (اجباری) و Overlap مقادیر NULL در مدل یک جدولی از نوع NULL غیرافزایشی است و NULL افزایشی نیست، چون **اختیاری بودن** عامل تولید مقادیر **NULL افزایشی** است.

تست - نمونه‌هایی از چهار موجودیت E1، E2، E3 و E4 در شکل مقابل نمایش داده شده است. کدام گزینه بهترین نمودار EER معرف محیط است؟
(مهندسی IT - دولتی ۹۹)



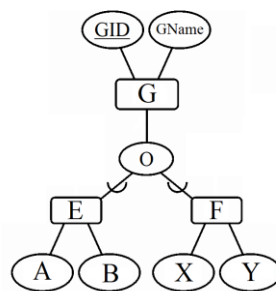
پاسخ - گزینه «۴» صحیح است.

همه نمونه موجودیت‌های i1 تا i8 در موجودیت E1 یعنی پدر با نمونه موجودیت‌های فرزند یعنی موجودیت‌های E2، E3 و E4 وارد رابطه شده‌اند. و از آنجاکه اگر **تمام** نمونه‌های پدر در رابطه شرکت کرده باشند، آن رابطه **اجباری** است، بنابراین نمونه‌های موجودیت E1 یعنی پدر به صورت اجباری وارد رابطه با فرزندان یعنی موجودیت‌های E2، E3 و E4 شده‌اند. بنابراین گزینه‌های اول و سوم پاسخ سوال نیستند.

توجه: از آنجا که کاردینالیتی رابطه پدر با فرزندان، **یک به چند** است. (نمونه موجودیت i1 در E1 یعنی پدر با نمونه موجودیت i1 در E2 یعنی فرزند و نمونه موجودیت i1 در E3 یعنی فرزند به طور یک به چند رابطه دارد)، بنابراین رابطه پدر با فرزندان حالت **Overlap** است و گزینه چهارم پاسخ سوال است.

د) مدل تحلیل ناقص (اختیاری) و Overlap

مدل تحلیل (نمودار ISA)



مدل طراحی (مدل رابطه‌ای)

مدل تک جدولی غیربهمینه به دلیل افزایش مقدار NULL در شرایط خاص: دقت کنید که در شرایط خاص یعنی زمانی که (۱) تعداد صفات زیرنوع زیاد، (۲) تعداد زیرنوع‌ها کم و (۳) تعداد صفات زیرنوع‌ها کم باشد، امکان پیاده‌سازی تک جدولی دارد اما بهمینه نیست.

سربار مقادیر NULL افزایشی دارد: چون **اختیاری بودن** عامل تولید مقادیر **NULL افزایشی** است، ارتباط زیرنوع با زیرنوع، اختیاری است، پس ممکن است رکوردهای زیرنوع، فرزند نداشته باشد و به تبع مقادیر NULL افزایشی ایجاد می‌شود. مانند سطر 2. هرچند تعداد صفات زیرنوع‌ها هم جهت NULL سازی کم است.

سربار افزونگی طبیعی ندارد: چون **Overlap بودن دو جدولی** عامل اصلی **افزونگی طبیعی** (تکرار رکورد) است.

سربار الحاق طبیعی ندارد: چون مدل سه جدولی عامل ایجاد سربار الحاق طبیعی میان جداول زیرنوع و زیرنوع در وراثت است.

صفات مشترک		صفات خاص E		صفات خاص F	
GID	GName	A	B	X	Y
GID1	GN1	A1	B1	X1	Y1
GID2	GN2	NULL	NULL	NULL	NULL
GID3	GN3	A3	B3	NULL	NULL

جدول GEF

توجه: مقادیر NULL در جدول فوق از نوع NULL غیرافزایشی (سطر 3) و NULL افزایشی (سطر 2) است، چون **اختیاری بودن** عامل تولید مقادیر NULL **افزایشی** است.

مدل دو جدولی غیربهمینه به دلیل افزایش افزونگی و افزایش مقادیر NULL: امکان پیاده‌سازی دو جدولی دارد اما بهمینه نیست.

سربار مقادیر NULL افزایشی دارد: چون **اختیاری بودن** عامل اصلی تولید مقادیر NULL **افزایشی** است، ارتباط زیرنوع با زیرنوع **اختیاری** است، پس ممکن است زیرنوع، فرزند نداشته باشد و به تبع مقادیر NULL افزایشی ایجاد می‌شود. مانند سطر 2 در جدول E و F.

سربار افزونگی طبیعی دارد: چون **Overlap بودن دو جدولی** عامل اصلی **افزونگی طبیعی** (تکرار رکورد) است. ارتباط **Overlap** است، پس ممکن است یک رکورد خاص زیرنوع G در دو جدول E و F تکرار شود.

سربار الحاق طبیعی ندارد: چون مدل سه جدولی عامل ایجاد سربار الحاق طبیعی میان جداول زیرنوع و زیرنوع در وراثت است.

GID	GName	A	B
GID1	GN1	A1	B1
GID2	GN2	NULL	NULL
GID3	GN3	A3	B3

جدول E

GID	GName	X	Y
GID1	GN1	X1	Y1

جدول F

توجه: در مدل دوجداولی، کل صفات پدر، در جداول فرزند قرار داده می‌شود.

توجه: مقادیر NULL در جدول فوق از نوع NULL **افزایشی** است.

توجه: دقت کنید که تکرار GID1 همراه نام در دو جدول E و F به صورت مدل دو جدولی، **سربار افزونگی طبیعی** محسوب می‌شود.

مدل سه جدولی بهمینه با تحمل سربار الحاق طبیعی: امکان پیاده‌سازی سه جدولی دارد و بهمینه هم هست.

سربار مقادیر NULL افزایشی ندارد: چون در مدل سه جدولی جداول زیرنوع و زیرنوع ازهم جدا هستند.

سربار افزونگی طبیعی ندارد: چون در مدل سه جدولی جداول زیرنوع و زیرنوع ازهم جدا هستند.

سربار الحاق طبیعی دارد: چون مدل سه جدولی عامل ایجاد سربار الحاق طبیعی میان جداول زیرنوع و زیرنوع در وراثت است.

GID	GName
GID1	GN1
GID2	GN2
GID3	GN3

جدول G

GID	A	B
GID1	A1	B1
GID3	A3	B3

جدول E

GID	X	Y
GID1	X1	Y1

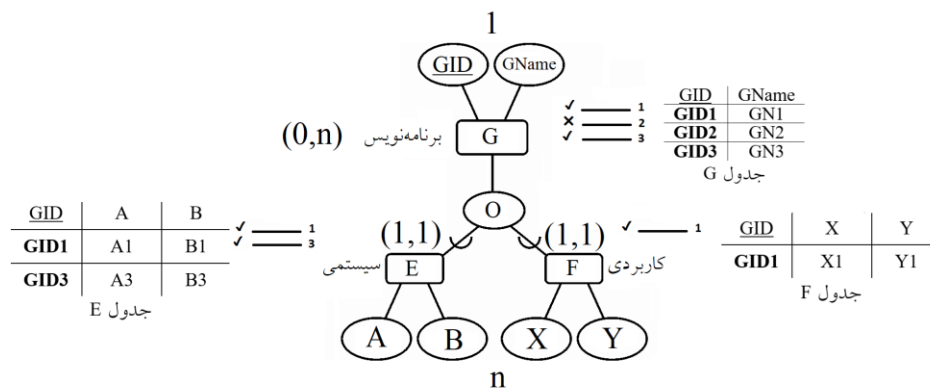
جدول F

توجه: دقت کنید که کلید خارجی جدول زیرنوع، کلید کاندید آن هم هست.

توجه: دقت کنید که مقدار GID2 در جدول E یا F درج نشده است چون فرض شده است مقادیر صفات خاص آن برابر مقدار NULL است. بنابراین مدل سه جدولی سربار مقادیر **NULL** افزایشی را ندارد.

توجه: دقت کنید که تکرار GID1 در دو جدول E و F به صورت مدل سه جدولی، افزونگی طبیعی محسوب نمی‌شود، بلکه افزونگی تکنیکی (فنی) به دلیل تعریف کلید خارجی محسوب می‌شود.

مدل شماتیک مدل تحلیل ناقص (اختیاری) و Overlap به صورت زیر است:



جمع‌بندی: چون رابطه بالا به پایین پدر با فرزندان از نوع **اختیاری** است، پس ممکن است رکوردهای پدر، فرزند نداشته باشد. برای مثال رکورد GID2,GN2 هیچ فرزندی ندارد. (در مدل یک، دو و سه جدولی صادق است).

جمع‌بندی: چون کاردینالیتی رابطه بالا به پایین پدر با فرزندان از نوع **(Overlap یا 1:n)** است، پس هر رکورد از پدر ممکن است بیش از یک فرزند داشته باشد. برای مثال رکورد GID1,GN1

می‌تواند دو فرزند $A1, B1$ از موجودیت E و $X1, Y1$ از موجودیت F داشته باشد. (در مدل یک، دو و سه جدولی صادق است).

نتیجه: قید محدودیت رابطه بالا به پایین برابر $(0, n)$ است، بدین معنی که هر پدر حداقل صفر و حداکثر چند فرزند دارد، همچنین قید محدودیت رابطه پایین به بالا برابر $(1, 1)$ است، بدین معنی که هر فرزند حداقل و حداکثر یک پدر دارد.

توجه: اگر e مجموعه نمونه‌های E ، f مجموعه نمونه‌های F و g مجموعه نمونه‌های G باشد، آنگاه $\text{card}(g) \leq \text{card}(e) + \text{card}(f)$ است. یعنی $3 \leq 2 + 1$ چون هر رکوردی که در پدر هست، ممکن است چند نمونه از آن در فرزندان باشد. یا $\text{card}(g) \geq \text{card}(e) + \text{card}(f)$ است، زمانی که تعداد رکوردهای پدر زیاد باشد و چون اختیاری است به شکل محدود در رابطه با فرزندان شرکت کند.

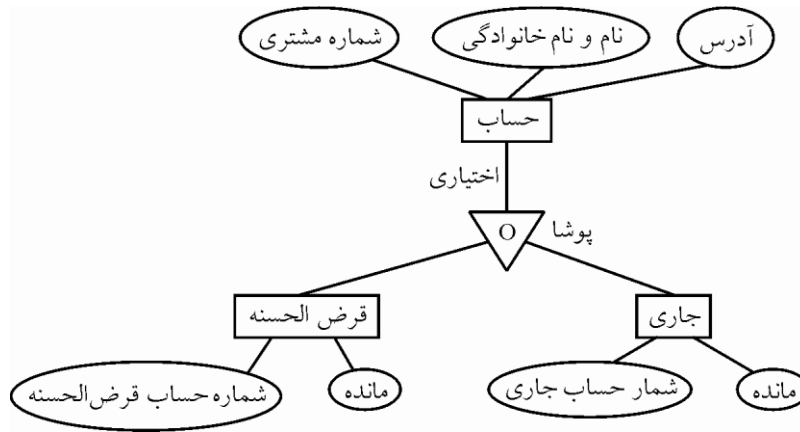
جمع‌بندی نگاشت رابطه ISA یا وراثت به مدل رابطه‌ای به صورت زیر است:

الف) مدل تحلیل کامل (اجباری) و Disjoint			ب) مدل تحلیل ناقص (اختیاری) و Disjoint			ج) مدل تحلیل کامل (اجباری) و Overlap			د) مدل تحلیل ناقص (اختیاری) و Overlap		
سه جدولی	دو جدولی	تک جدولی	سه جدولی	دو جدولی	تک جدولی	سه جدولی	دو جدولی	تک جدولی	سه جدولی	دو جدولی	تک جدولی
بهمه	مقادیر NULL افزایشی و افزونگی طبیعی	غیربهمه به دلیل مقادیر NULL افزایشی	بهمه	مقادیر NULL افزایشی	غیربهمه به دلیل مقادیر NULL افزایشی	بهمه	مقادیر NULL افزایشی	غیربهمه به دلیل مقادیر NULL افزایشی	بهمه	مقادیر NULL افزایشی	غیربهمه به دلیل مقادیر NULL افزایشی

جمع‌بندی: هر جا اختیاری بود در مدل تک‌جدولی و دوجداولی همیشه غیربهمه به دلیل سربرار مقادیر NULL افزایشی، هر جا Overlap دو جدولی بود همیشه غیربهمه به دلیل سربرار افزونگی طبیعی و بقیه موارد بهمه است و تمام.

جمع‌بندی: بدون در نظر گرفتن مدل تک جدولی در شرایط خاص، بهمینه‌ترین مدل طراحی برای مدل تحلیل کامل (اجباری) و Disjoint مدل دو جدولی است. و بهمینه‌ترین مدل طراحی برای مدل‌های دیگر مدل سه جدولی است.

مثال: نمودار نهاد و رابطه زیر را در نظر بگیرید: (مدل اختیاری و Overlap)

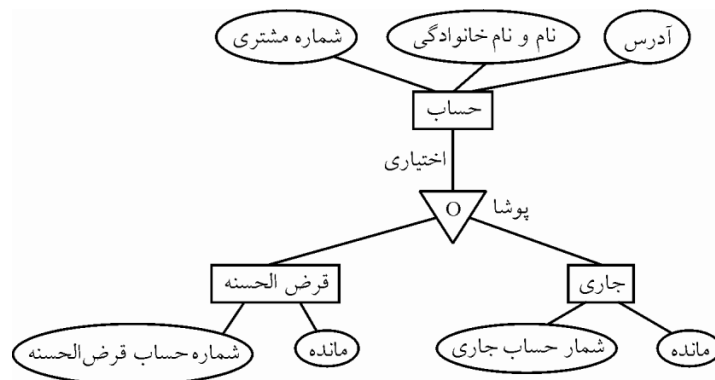


توجه: در واقعیت، رابطه بالا به پایین موجودیت حساب (پدر) با موجودیت حساب جاری (فرزند) و موجودیت حساب قرض الحسنه (فرزند) از نوع **اختیاری** است، پس ممکن است نمونه رکوردهای پدر، فرزند داشته باشد یا نداشته باشد. برای مثال ممکن است رکورد یک شماره مشتری خاص از موجودیت حساب، حساب جاری یا قرض الحسنه داشته باشد یا نداشته باشد.

توجه: کاردینالیتی رابطه بالا به پایین موجودیت حساب (پدر) با موجودیت حساب جاری و موجودیت حساب قرض الحسنه از نوع (Overlap یا 1:n) است، بنابراین رکورد یک شماره مشتری خاص، می تواند به طور هم زمان با نمونه موجودیت های حساب جاری و حساب قرض الحسنه در ارتباط باشد یا نباشد.

توجه: بنابراین در این شرایط مدل سه جدولی اما با تحمل سربرار حاصل از عمل الحاق بهینه است.

مدل تحلیل (نمودار ISA)



مدل طراحی (مدل رابطه‌ای)

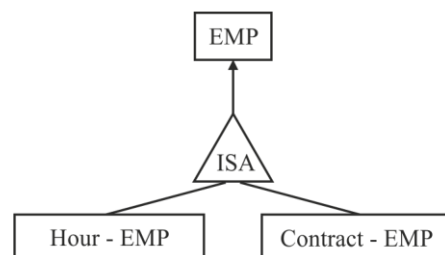
آدرس		نام و نام خانوادگی	شماره مشتری
جدول حساب			
مانده	شماره حساب جاری	شماره مشتری	مانده
جدول حساب جاری		شماره حساب قرض الحسنه	شماره مشتری
جدول حساب قرض الحسنه			

توجه: جداول فوق را می‌توان به این صورت تفسیر معنایی نمود که، هنگام ایجاد یک حساب جاری یا حساب قرض الحسنه در پایگاه داده، اطلاعات بسته به نوع حساب به دو بخش اطلاعات عمومی و اطلاعات اختصاصی تقسیم می‌شود. البته اطلاعات عمومی حساب در جدول حساب درج می‌شود، سپس اطلاعات اختصاصی در یکی از جداول حساب جاری یا حساب قرض الحسنه متناسب با نوع حساب درج می‌شود و در نهایت دو سطر درج شده در دو جدول پدر و فرزند از طریق شماره مشتری، که کلید اصلی جدول حساب و کلید خارجی جداول حساب جاری یا حساب قرض الحسنه است به یکدیگر مرتبط می‌شوند. به این ترتیب می‌توان گفت که اطلاعات عمومی در جدول حساب و اطلاعات اختصاصی در جداول حساب جاری و حساب قرض الحسنه قرار دارند.

تست - رابطه ISA (ارث‌بری) در نمودار ER زیر را چگونه می‌توان در مدل رابطه‌ای پیاده‌سازی نمود؟

(مهندسی کامپیوتر- دولتی ۸۴)

- ۱) رابطه ISA در مدل رابطه‌ای قابل پیاده‌سازی نیست.
- ۲) همه ویژگی‌های EMP را در Hour-EMP و Contract-EMP کپی می‌نماییم.
- ۳) کلید اصلی EMP را در Hour-EMP و Contract-EMP به عنوان کلید اصلی و کلید خارجی مجدداً تعریف می‌کنیم.
- ۴) موارد ۲ و ۳ هر دو صحیح می‌باشند.



پاسخ - گزینه «۴» صحیح است. گزینه ۲ مدل دو جدولی و گزینه ۳ مدل سه جدولی است.

تست - برای پیاده‌سازی رابطه IS-A در یک نمودار ER به مدل رابطه‌ای کدام یک از گزینه‌های زیر بهترین است؟

(مهندسی کامپیوتر - دولتی ۸۵)

(۱) کلید اصلی موجودیت پدر به عنوان کلید خارجی به موجودیت‌های فرزند آن موجودیت اضافه شود.

(۲) توسط موجودیت پلی (Bridge) ایجاد می‌شود که شامل کلید اصلی موجودیت پدر و موجودیت‌های فرزند آن می‌باشد.

(۳) بخشی از ویژگی‌های موجودیت پدر که در موجودیت‌های فرزند نیاز است را به موجودیت‌های فرزند اضافه می‌کند.

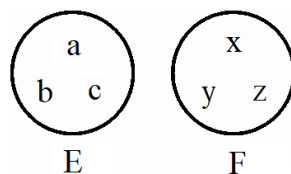
(۴) کلید اصلی موجودیت‌های فرزند را به عنوان کلید خارجی به موجودیت پدر اضافه می‌کند.

پاسخ - گزینه «۱» صحیح است.

گزینه ۲: موجودیت پل برای ارتباط چند به چند کاربرد دارد. گزینه ۳ و ۴: هیچکدام از مدل‌های پیاده‌سازی سه‌گانه نیست.

زیرنوع اجتماع (Union subtype: U-Type)

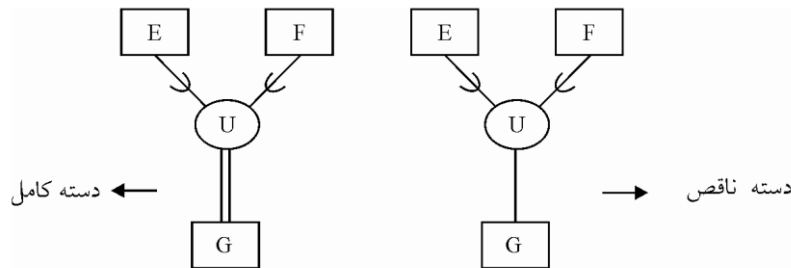
موضوع زیرنوع اجتماع (U-Type) روی اجتماع مجموعه‌های مستقل از هم هست و نه مجموعه‌های دارای اشتراک باهم. عملگر $\cup$ ، جهت تجمیع سطرهای مجموعه‌های مستقل از هم می‌تواند مورد استفاده قرار بگیرد. اگر $E \cap F = \emptyset$ باشد، نمودار نمایش اجتماع به صورت زیر است:



مطابق قوانین مجموعه در ریاضیات، هر مجموعه مثل E و F به طور ذاتی سطر تکراری ندارد، حال اگر دو مجموعه مستقل از هم باشند، آنگاه اجتماع آنها هم سطر تکراری ندارد، چون باهم اشتراک ندارند. بنابراین Union type ($E \cup F$) یعنی سطری که متعلق به E است یا F، در نگاه جدولی هم یعنی سطری که صفات E دارد یا صفات F، از این جهت به U-Type، **وراثت انتخابی** نیز گفته می‌شود.

مثال: اگر $E = \{a, b, c\}$ و $F = \{x, y, z\}$ دو مجموعه مستقل از هم باشند، آنگاه $E \cup F = \{a, b, c, x, y, z\}$ است. و هر عضو از $E \cup F$ ، از E است یا از F، مثل عضو a که فقط از E است، یا عضو x که فقط از F است.

توجه: به زیرنوع اجتماع، **دسته (Category)** یا **طبقه** نیز گفته می‌شود. زیرنوع اجتماع با حرف **U** (حرف اول کلمه **Union**) نشان داده می‌شود، به صورت زیر:



نتیجه: **G**، یک زیرنوع اجتماع (دسته) از موجودیت‌های **E** و **F** است.

توجه: یک سطر از **G** یک سطر از **E** یا یک سطر از **F** است.

توجه: یک سطر از زیرنوع اجتماع، بسته به اینکه از نوع کدام زیرنوع باشد، فقط و فقط صفات همان زیرنوع را به ارث می‌برد. بر این اساس، به زیرنوع اجتماع، **وراثت انتخابی** نیز گفته می‌شود. به وراثت **ISA** به شکل **Disjoint** و **Overlap** **وراثت ساختاری** نیز گفته می‌شود.

دسته کامل (اجباری): اگر همه سطرهای زیرنوع‌های **E** و **F** به طور اجباری در اجتماع شرکت کنند، آنگاه سطرهای زیر نوع اجتماع **G**، حاصل از اجتماع **(U)** **دسته کامل** است.

دسته ناقص (اختیاری): اگر بعضی سطرهای زیرنوع‌های **E** و **F** به طور اختیاری در اجتماع شرکت کنند، آنگاه سطرهای زیرنوع اجتماع **G**، حاصل از اجتماع **(U)** **دسته ناقص** است.

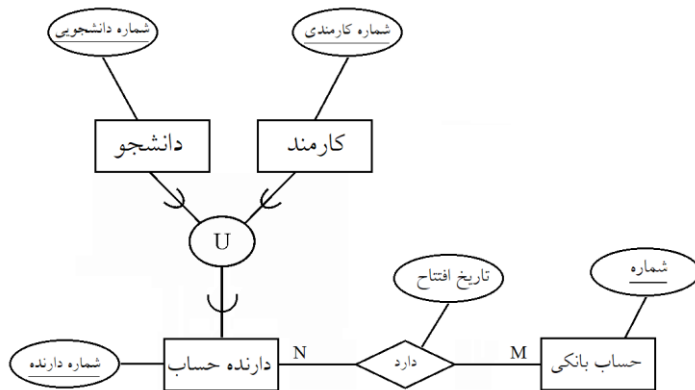
نتیجه: در **U-Type**، کامل یا ناقص بودن زیرنوع (دسته یا فرودست)، وابسته به اجباری یا اختیاری بودن زیرنوع‌ها (فرادست) است، بسته به اینکه زیرنوع‌ها اجباری یا اختیاری باشد، آنگاه زیرنوع نیز کامل یا ناقص می‌شود. در حالی که در مباحث **Disjoint** و **Overlap**، **مستقل** از اجباری یا اختیاری بودن پدر (زیرنوع)، فرزندان (زیرنوع‌ها) **همواره اجباری** هستند. دقت کنید که زیرنوع در **U-Type** به خودی خود مفهوم اجباری و اختیاری را ندارد، بلکه اجباری و اختیاری بودن زیرنوع در زیرنوع نمایان می‌شود. در **U-Type** اگر زیرنوع، **اجباری** باشد، زیرنوع هم «**دسته کامل**» است و اگر زیرنوع، **اختیاری** باشد، زیرنوع هم «**دسته ناقص**» است.

کلید کاندید (شناسه) زیرنوع اجتماع (U-Type)

اگر کلید کاندید زیرنوع‌ها (فرادست) از میدان یکسان باشد: آنگاه کلید کاندید زیرنوع هم همان کلید کاندید زیرنوع‌ها است.

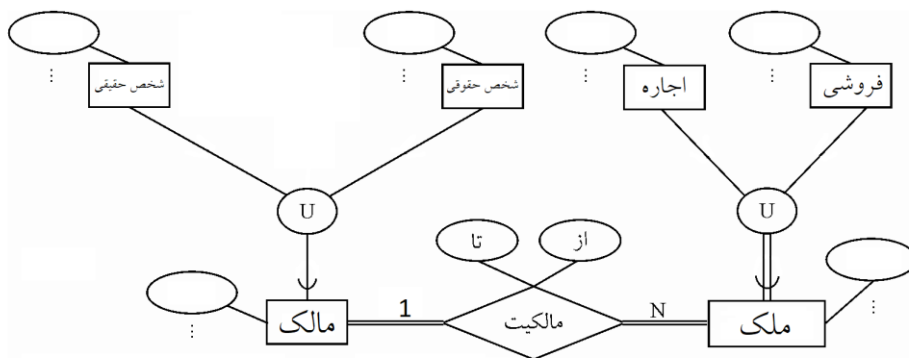
اگر کلید کاندید زیرنوع‌ها (فرادست) از میدان متفاوت باشد: آنگاه کلید کاندید زیرنوع توسط طراح بانک تعیین می‌شود.

مثال: موجودیت‌های کارمند و دانشجو را به صورت زیر در نظر بگیرید.



این دو موجودیت می‌توانند حساب بانکی داشته باشند. در این مثال رکوردهای موجودیت **دارنده حساب** می‌تواند دانشجو یا کارمند باشد. همانطور که گفتیم شناسه زیرنوع‌های یک **زیرنوع اجتماع** ممکن است از میدان یکسان باشد یا از میدان متفاوت. در این مثال اگر به جای صفت **شماره کارمندی** و **شماره دانشجویی**، صفت **کد ملی** لحاظ شود، شناسه دو زیرنوع دانشجو و کارمند، از میدان یکسان می‌شود و همان نیز برای زیرنوع اجتماع به عنوان شناسه منظور می‌شود. اما وقتی شناسه زیرنوع‌ها از میدان متفاوت باشد، طراح باید برای زیرنوع اجتماع، یک شناسه تعیین کند، مثل صفت **شماره دارنده حساب** در شکل فوق.

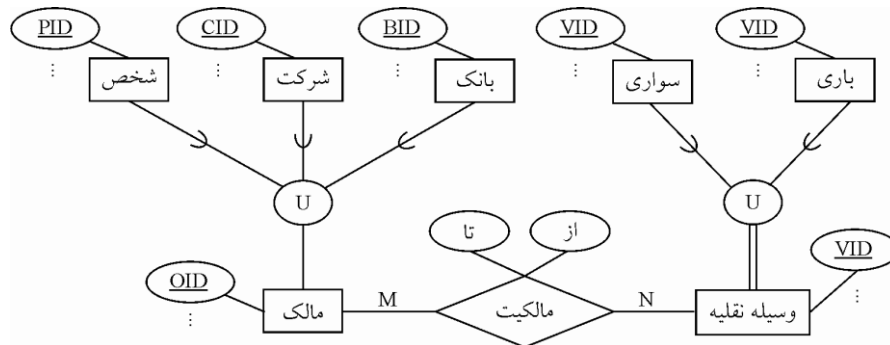
مثال: نمودار EER که دسته‌های املاک را نشان می‌دهد به صورت زیر است:



توجه: همه سطرهای **ملک فروشی** و **ملک اجاره‌ای** به طور **اجباری** باید در اجتماع شرکت کنند، چون چه ملک فروشی باشد و چه ملک اجاره‌ای باشد **الزاماً** ملک است. پس زیرنوع اجتماع **ملک**، **دسته کامل** است.

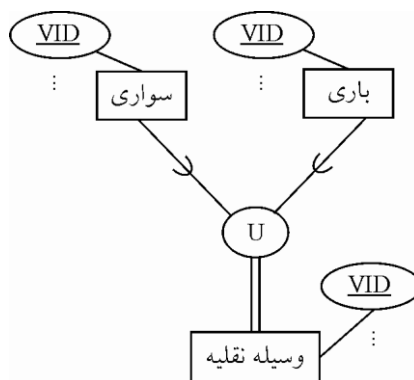
توجه: بعضی سطرهای **شخص حقوقی** و **شخص حقیقی** به طور **اختیاری** می‌توانند در اجتماع شرکت کنند، چون چه شخص حقوقی باشد و چه شخص حقیقی باشد **الزاما** مالک نیست. پس زیرنوع اجتماع مالک، **دسته ناقص** است.

مثال: مدل رابطه‌ای متناظر با نمودار EER زیر به چه صورتی است؟



نگاشت رابطه زیرنوع‌ها از میدان یکسان و زیرنوع اجتماع U-Type به مدل رابطه‌ای
در شکل زیر، **کلید کاندید** (شناسه) زیرنوع‌ها از **میدان یکسان** (VID) است، پس **کلید کاندید** موجودیت زیرنوع، همان کلید کاندید موجودیت زیرنوع‌ها است.

مدل تحلیل:



مدل طراحی:

<u>VID</u>	...	...
VID3		
VID4		

جدول سواری

<u>VID</u>	...	...
VID1		
VID2		

جدول باری

<u>VID</u>	...
VID1	
VID2	
VID3	
VID4	

جدول وسیله نقلیه

Cargo (VID,...)
Car (VID,...)
Vehicle (VID,...)

توجه: اگر N تعداد زیرنوع‌های زیرنوع اجتماع باشد، تعداد جداول لازم $N+1$ است.

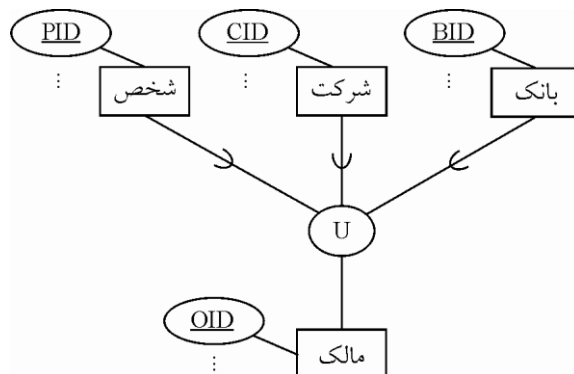
توجه: همه سطرهای وسیله نقلیه باری و وسیله نقلیه سواری به طور اجباری باید در اجتماع شرکت کنند، چون چه وسیله نقلیه باری باشد و چه وسیله نقلیه سواری باشد الزاماً وسیله نقلیه است. پس زیرنوع اجتماع وسیله نقلیه، دسته کامل است. همه سطرهای VID1 و VID2 از جدول وسیله نقلیه باری و همه سطرهای VID3 و VID4 از جدول وسیله نقلیه سواری در اجتماع شرکت کرده‌اند. بنابراین رابطه $\text{card}(R3) = \text{card}(R1) + \text{card}(R2)$ همواره برقرار است.

توجه مهم: همانطور که گفتیم، یک سطر از زیرنوع اجتماع، بسته به اینکه از نوع کدام زیرنوع باشد، فقط و فقط صفات همان زیرنوع را به ارث می‌برد. بر این اساس، به زیرنوع اجتماع، وراثت انتخابی نیز گفته می‌شود. برای مثال VID1 و VID2 از جدول وسیله نقلیه، توسط کلید خارجی فقط و فقط صفات جدول باری را انتخاب می‌کنند و VID3 و VID4 از جدول وسیله نقلیه، توسط کلید خارجی فقط و فقط صفات جدول سواری را انتخاب می‌کنند.

نگاشت رابطه زیرنوع‌ها از میدان متفاوت و زیرنوع اجتماع U-Type به مدل رابطه‌ای

در شکل زیر، کلید کاندید (شناسه) زیرنوع‌ها از میدان متفاوت BID و CID و PID است، پس کلید کاندید OID توسط طراح بانک برای زیرنوع تعیین می‌شود. همچنین کلید کاندید موجودیت زیرنوع (همان شناسه تعیین شده توسط طراح بانک) به عنوان کلید خارجی در زیرنوع‌ها تعریف می‌شود.

مدل تحلیل:



مدل طراحی:

<u>PID</u>	...	<u>OID</u>	<u>CID</u>	...	<u>OID</u>	<u>BID</u>	...	<u>OID</u>
PID1		OID3	CID1		OID2	BID1		OID1
PID2		OID4	CID2		NULL	BID2		NULL

جدول شخص جدول شرکت جدول بانک

<u>OID</u>	...
OID1	
OID2	
OID3	
OID4	

جدول مالک

توجه: کلید کاندید جدول مالک یعنی ستون OID در جدول‌های **بانک**، **شرکت** و **شخص** به عنوان **کلید خارجی** تعریف می‌گردد.

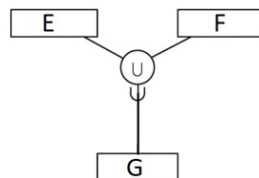
Bank (BID, ..., OID)
Company (CID, ..., OID)
Person (PID, ..., OID)
Owner (OID, ...)

توجه: بعضی سطرهای **بانک**، **شرکت** و **شخص** به طور **اختیاری** می‌توانند در اجتماع شرکت کنند، چون چه بانک باشد، چه شرکت باشد و چه شخص باشد **الزاماً** مالک نیست. پس زیرنوع

اجتماع مالک، دسته ناقص است. سطر BID2 از جدول بانک و سطر CID2 از جدول شرکت در اجتماع شرکت نکرده‌اند. بنابراین رابطه $\text{card}(R3) \leq \text{card}(R1) + \text{card}(R2)$ همواره برقرار است.

توجه مهم: همانطور که گفتیم، یک سطر از زیرنوع اجتماع، بسته به اینکه از نوع کدام زیرنوع باشد، فقط و فقط صفات همان زیرنوع را به ارث می‌برد. بر این اساس، به زیرنوع اجتماع، **وراثت انتخابی** نیز گفته می‌شود. برای مثال OID1 از **جدول مالک**، توسط **کلید خارجی** فقط و فقط صفات **جدول بانک** را انتخاب می‌کنند، OID2 از **جدول مالک**، توسط **کلید خارجی** فقط و فقط صفات **جدول شرکت** را انتخاب می‌کند و OID3 و OID4 از **جدول مالک**، توسط **کلید خارجی** فقط و فقط صفات **جدول شخص** را انتخاب می‌کنند.

سوال: با توجه به نمودار زیر :



اگر e مجموعه نمونه‌های E ، f مجموعه نمونه‌های F و g مجموعه نمونه‌های G باشد:

$$g = e \cup f \quad (1)$$

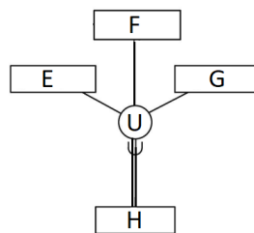
$$g \subset e \cup f \quad (2)$$

$$g \subseteq e \cup f \quad (3)$$

$$g \supseteq e \cup f \quad (4)$$

پاسخ: گزینه «۳» صحیح است. چون دسته ناقص است.

سوال: با توجه به نمودار زیر، زیرنوع موجودیت H ، چه صفاتی را به ارث می‌برد؟



(۱) همه صفات نوع موجودیت‌های E ، F و G را به ارث می‌برد.

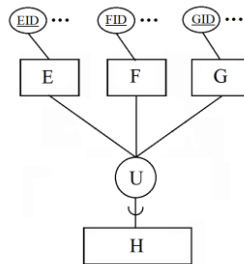
(۲) همه صفات حداقل یکی از سه نوع موجودیت E ، F و G را به ارث می‌برد.

(۳) حداقل شناسه سه نوع موجودیت E ، F و G را به ارث می‌برد.

(۴) صفات یکی از سه نوع موجودیت E یا F یا G را به ارث می‌برد.

پاسخ: گزینه «۴» صحیح است. بسته به کلید خارجی صفات یکی از سه نوع موجودیت E یا F یا G را به ارث می‌برد.

سوال: با توجه به نمودار، شناسه نوع موجودیت H چیست؟



فرض: $EID \neq FID \neq GID$

(۱) $HID \in \{EID, FID, GID\}$

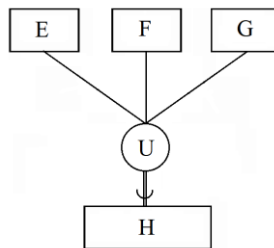
(۲) $HID \in \{(EID, GID), (EID, FID), (GID, FID)\}$

(۳) $HID \in \{EID, FID, GID\}$

(۴) $HID \neq I \mid I \in \{EID, FID, GID\}$

پاسخ: گزینه «۴» صحیح است. چون از میدان متفاوت است.

سوال: با توجه به نمودار، نوع موجودیت H چند شناسه (کلید کاندید) دارد؟



(۱) سه شناسه

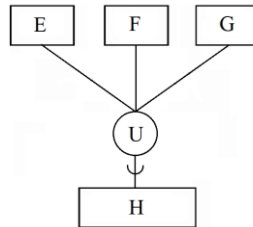
(۲) حداکثر سه شناسه

(۳) یک شناسه

(۴) حداقل یک شناسه

پاسخ: گزینه «۳» صحیح است. زیر نوع اجتماع، همواره یک کلید کاندید دارد.

سوال: با توجه به نمودار، نوع موجودیت H چند شناسه (کلید کاندید) دارد؟



(۱) یک شناسه

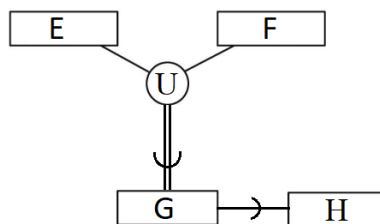
(۲) یک شناسه حاصل ترکیب شناسه‌های E و F و G

(۳) سه شناسه، هر یک حاصل ترکیب شناسه‌های هر بار دو نوع موجودیت

(۴) سه شناسه

پاسخ: گزینه «۱» صحیح است. زیر نوع اجتماع، یک کلید کاندید دارد.

سوال: در نمودار زیر، شناسه (کلید کاندید) نوع موجودیت H چه می‌تواند باشد؟



I: شناسه G، یکسان با شناسه E و شناسه F

II: شناسه G، یکسان با شناسه E یا شناسه F و نه هر دو

III: شناسه G، متفاوت با شناسه E و شناسه F

(۱) I یا III (۲) II

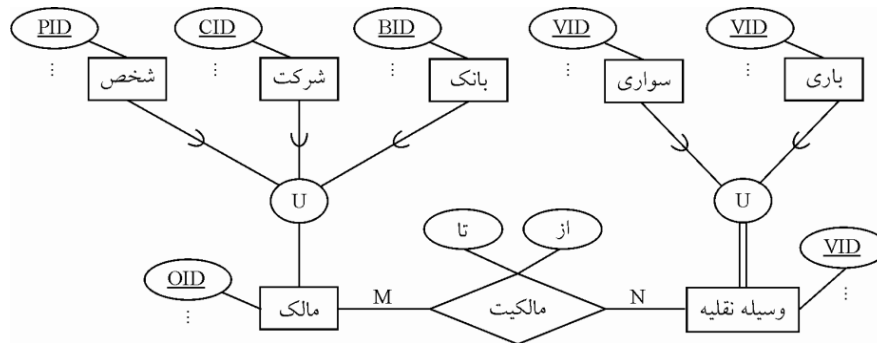
(۳) III (۴) I یا II

پاسخ: گزینه «۱» صحیح است. موجودیت H از G ارث‌بری عمومی دارد.

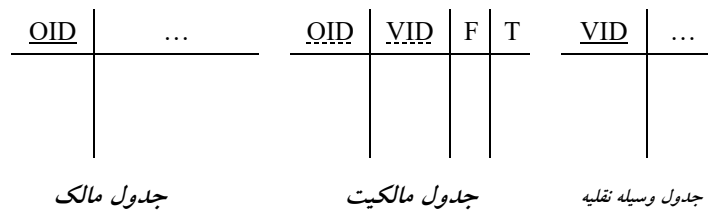
نگاشت رابطه چند به چند بین دو موجودیت با زیرنوع اجتماع U-Type به مدل رابطه‌ای

مستقل از اختیاری یا اجباری بودن موجودیت‌ها، هر موجودیت به یک جدول تبدیل می‌گردد و یک جدول پُل (Bridge) نیز به عنوان ارتباط دهنده دو جدول مورد استفاده قرار می‌گیرد. همچنین کلید کاندید جدول پُل از ترکیب کلید کاندید دو جدول دیگر ایجاد می‌گردد. همچنین صفات متصل به رابطه، درون جدول پُل مستتر می‌شود.

مدل تحلیل:



مدل طراحی:



توجه: ستون VID و OID در جدول **مالکیت** به عنوان **کلید خارجی** تعریف می‌گردند که به ترتیب به جدول **وسيله نقلیه** و **مالک** ارجاع می‌کنند، همچنین **صفات متصل** به رابطه مالکیت، درون جدول پل یعنی جدول مالکیت، **مستتر** می‌شود.

توجه: کلید کاندید جدول پل (جدول مالکیت) از ترکیب کلید کاندید دو جدول دیگر ایجاد می‌گردد، که برابر (OID,VID) است.

توجه: کلید کاندید جدول چند راست (جدول وسیله نقلیه) ستون VID است.

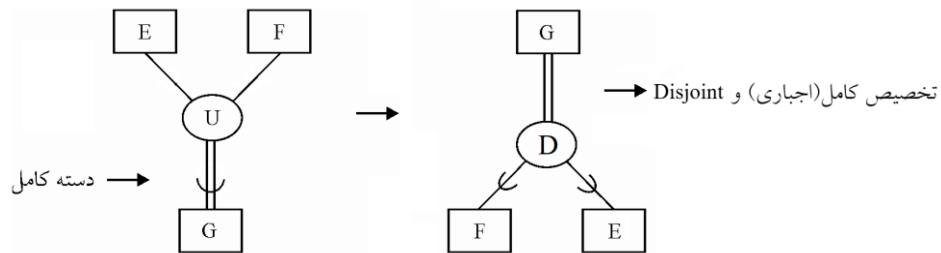
توجه: کلید کاندید جدول چند چپ (جدول مالک) ستون OID است.

توجه: سایر جداول باری، سواری، بانک، شرکت و شخص مطابق مثال قبل ایجاد می‌شود.

معادل‌سازی U-Type با مدل اجباری و Disjoint

همانطور که گفتیم، در U-Type، کامل یا ناقص بودن زیرنوع (دسته یا فرودست)، **وابسته** به اجباری یا اختیاری بودن زیرنوع‌ها (فرداست) است، بسته به اینکه زیرنوع‌ها اجباری یا اختیاری باشد، آنگاه زیرنوع نیز کامل یا ناقص می‌شود. در حالی که در مباحث Disjoint و Overlap، **مستقل** از اجباری یا اختیاری بودن پدر (زیرنوع)، فرزندان (زیرنوع‌ها) **همواره اجباری** هستند. دقت کنید که زیرنوع در U-Type به خودی خود مفهوم اجباری و اختیاری را ندارد، بلکه اجباری

و اختیاری بودن زیرنوع در زیرنوع نمایان می‌شود. در U-Type اگر زیرنوع، **اجباری** باشد، زیرنوع هم «**دسته کامل**» است و اگر زیرنوع، **اختیاری** باشد، زیرنوع هم «**دسته ناقص**» است. **توجه مهم:** اگر U-Type مدل «**دسته کامل**» باشد، آنگاه دقیقاً **معادل** مدل کامل (اجباری) و Disjoint است و بالعکس.



توجه: اگر **شرایط معادل‌سازی** برقرار باشد، آنگاه زیرنوع (فردست) در U-Type **معادل** زیرنوع (فرداست) در مدل کامل (اجباری) و Disjoint است و همچنین زیرنوع (فرداست) در U-Type **معادل** زیرنوع (فرداست) در مدل کامل (اجباری) و Disjoint است.

مدل طراحی به صورت U-Type

<u>GID</u>	..	<u>GID</u>	...
GID1		GID3	
GID2			

جدول E جدول F

<u>GID</u>	...
GID1	
GID2	
GID3	

جدول G

مدل طراحی به صورت مدل کامل (اجباری) و Disjoint

<u>GID</u>	...
GID1	
GID2	
GID3	

جدول G

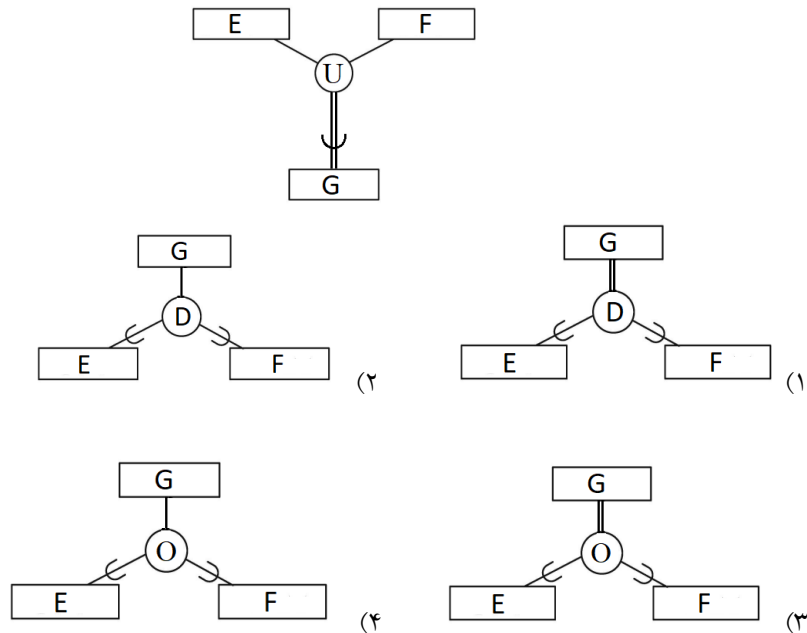
<u>GID</u>	..	<u>GID</u>	...
GID1		GID3	
GID2			

جدول E

جدول F

توجه بسیار مهم: به طور کلی، هدف نهایی جداول، نگهداری و ذخیره داده‌ها است. هر مدلی که امکان ذخیره‌سازی داده‌ها را دارد مناسب است، اما در شرایط مختلف باید بهینه بودن آن نیز بررسی شود. بنابراین اینکه کدام مدل‌سازی انتخاب شود، بسته به نظر طراح بانک اطلاعات است.

سوال: فرض کنید یک محیط را به صورت زیر مدل‌سازی کرده باشیم، به چه طرز دیگری می‌توان این محیط را مدل کرد؟

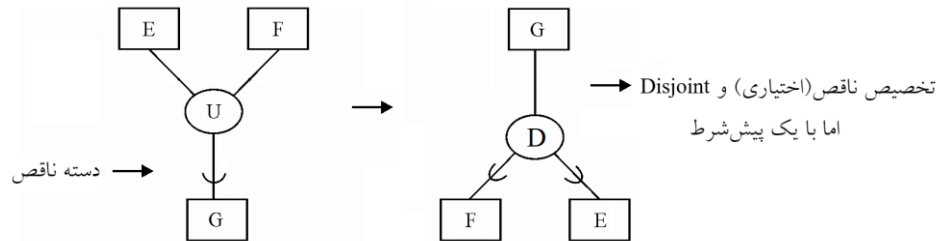


پاسخ: گزینه «۱» صحیح است.

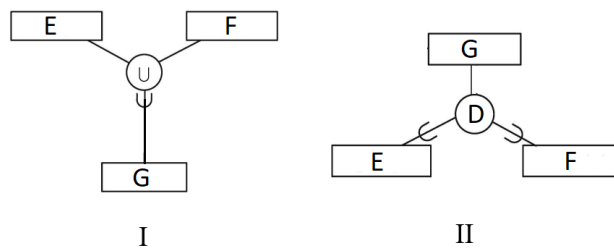
معادل‌سازی U-Type با مدل اختیاری و Disjoint

هرگاه U-Type مدل «دسته ناقص» باشد، می‌توان به جای آن، از مدل اختیاری و Disjoint اما با پیش‌شرط $C_G = C_E + C_F$ استفاده کرد. در واقع پیش‌شرط $C_G = C_E + C_F$ ، به شکل هوشمندانه، مدل «دسته ناقص» را در حالت مدل «دسته کامل» تثبیت و نگهداری می‌کند.

Cx یا $card(x)$: کاردینالیته نمونه رکوردهای موجودیت X است.



سوال: مدل سازی I را با چه شرطی می توان به صورت II هم انجام داد؟



$$C_G \leq C_E + C_F \quad (1)$$

$$C_G > C_E + C_F \quad (2)$$

$$C_G = C_E + C_F \quad (3)$$

$$C_G \geq C_E + C_F \quad (4)$$

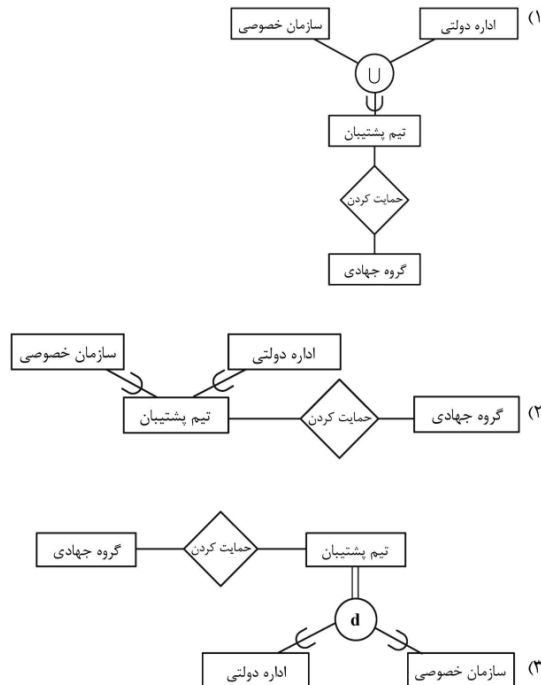
پاسخ: گزینه «۳» صحیح است.

توجه مهم: مدل U-Type تحت هیچ شرایطی امکان مدل سازی توسط Overlap را ندارد. چون ماهیت Overlap همپوشانی است که با ذات U-Type در تضاد است.

تست - می خواهیم برای عبارت زیر یک EER مناسب رسم کنیم. کدام مورد مناسب ترین گزینه است؟

«در هنگام بروز زلزله، تعدادی تیم پشتیبان تشکیل می شود که هر تیم پشتیبان از یک سازمان خصوصی یا یک اداره دولتی انتخاب می شود. این تیم کار حمایت از گروه های جهادی را برعهده دارد. بعضی سازمان های خصوصی علاقه ای به شرکت در این طرح ندارند.»

(مهندسی IT - دولتی ۱۴۰۲)



(۴) موارد ۱ و ۳ درست است.

پاسخ - گزینه (۱) صحیح است.

توجه: سازمان سنجش در **کلید اولیه گزینه چهارم** و در **کلید نهایی گزینه اول** را به عنوان پاسخ صحیح اعلام کرد که کار درستی بوده است.

«در هنگام بروز زلزله، تعدادی تیم پشتیبان تشکیل می‌شود که هر تیم پشتیبان از یک **سازمان خصوصی** «یا» یک **اداره دولتی** انتخاب می‌شود. بنابراین به دلیل ذکر «یا» تا اینجا مدل U یا D کاربرد دارد. دقت کنید که گزینه دوم یعنی مدل وراثت عمومی، مدل U و D را پشتیبانی نمی‌کند. زیرا در وراثت عمومی یک پدر خاص ممکن است همزمان یک فرزند از «اداره دولتی» و یک فرزند از «سازمان خصوصی» داشته‌باشد، که این موضوع با فرض سوال که تاکید روی «یا» دارد در **تضاد** است.

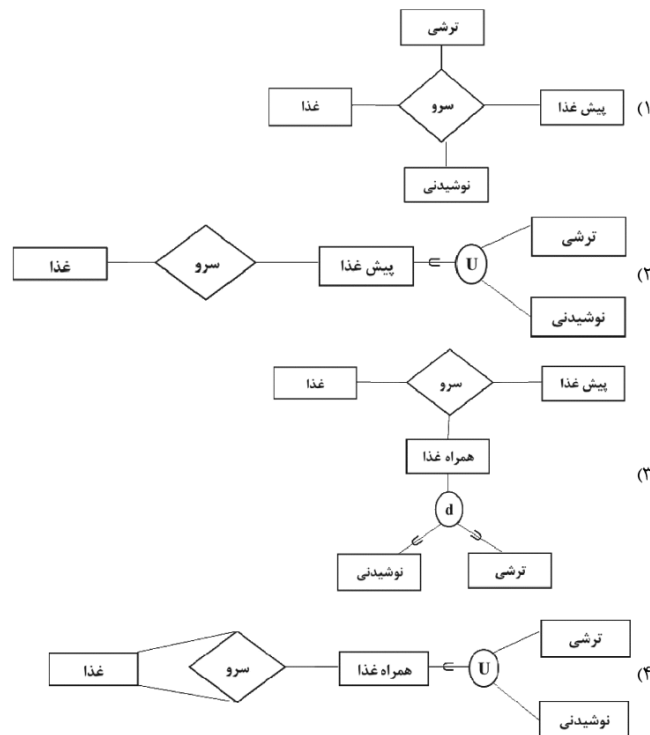
بعضی سازمان‌های خصوصی علاقه‌ای به شرکت در این طرح ندارند. **سازمان خصوصی یا اداره دولتی** در مدل U موجودیت زیرنوع (فراست) محسوب می‌شود که مطابق فرض سوال، شرکت **سازمان خصوصی** در رابطه **اختیاری** لحاظ شده است که نتیجه آن **دسته ناقص** است. همچنین سازمان خصوصی یا اداره دولتی در مدل D زیرنوع (فراست) محسوب می‌شود که در مدل D زیرنوع، مستقل از اجباری یا اختیاری بودن زیرنوع (فراست) همواره اجباری است، چون در مدل D فرزند بدون پدر معنی ندارد، در مدل D هر فرزندی به طور اجباری قطعاً یک پدر دارد. با توجه به فرض سوال شرکت سازمان

خصوصی در رابطه اختیاری است، درحالی که در گزینه سوم و مطابق الگوی D رابطه فرزندان با پدر همواره اجباری است. پس مدل D در حل این سوال کاربرد ندارد. بنابراین گزینه اول به صورت دسته ناقص (چون بعضی سازمان‌های خصوصی علاقه‌ای به شرکت در این طرح ندارند). پاسخ سوال است. **توجه:** اگر در صورت سوال بیان می‌شد که رابطه همه نمونه‌های تیم پشتیبان با سازمان خصوصی و اداره دولتی اجباری است و همچنین رابطه همه نمونه‌های سازمان خصوصی و اداره دولتی با تیم پشتیبان اجباری است، آنگاه مدل زیرنوع Disjoint و زیرنوع (فرداست) اجباری (کامل) یعنی **گزینه سوم** صحیح می‌بود.

تست - کدام نمودار بهترین توصیف برای عبارت زیر را نشان می‌دهد؟

«در اسناد تحلیل مربوط به امور تغذیه دانشجویان نوشته شده است: تمام غذاها این قابلیت را دارند که به عنوان پیش‌غذای دیگر نیز استفاده شوند. به عنوان مثال الویه که خود می‌تواند غذای مستقلی باشد، می‌تواند به عنوان پیش‌غذای چلوخورشت قیمه به دانشجو داده شود. همچنین همراه با پیش‌غذا یک نوشیدنی یا ترشی (و نه هر دو باهم) به دانشجو داده می‌شود.»

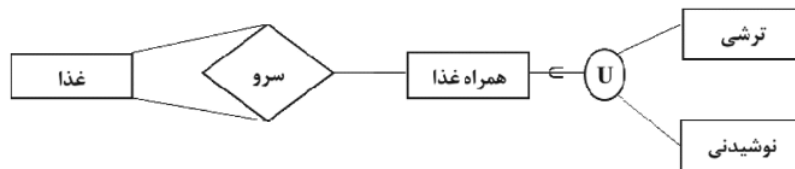
(مهندس کامپیوتر-دولتی ۱۴۰۲)



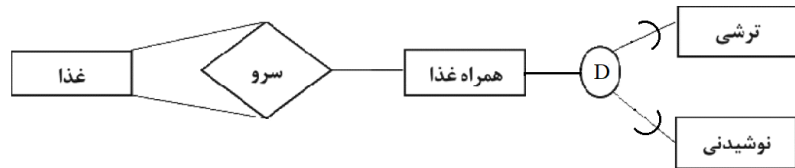
پاسخ - گزینه (۴) صحیح است.

در صورت سوال مطرح شده است که «تمام غذاها این قابلیت را دارند که به عنوان پیش‌غذای دیگر نیز استفاده شوند. به عنوان مثال الویه که خود می‌تواند غذای مستقلی باشد، می‌تواند به عنوان پیش‌غذای چلوخورشت قیمه به دانشجو داده شود.» **بنابراین بین غذا و پیش‌غذا رابطه درجه یک (خود ارجاعی طوقه) وجود دارد.** تا اینجا واضح است که به دلیل وجود رابطه درجه یک، گزینه چهارم پاسخ سوال است، مستقل از اینکه U-Type باشد یا Disjoint و بحثی در آن نیست.

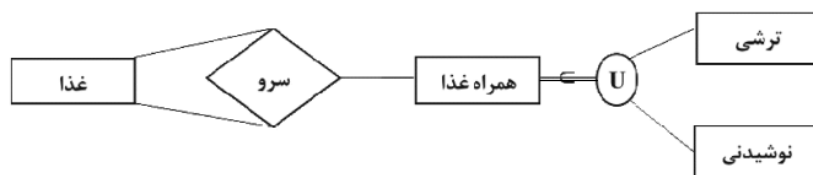
در صورت سوال **اجباری** یا **اختیاری** بودن شرکت نمونه‌های زیرنوع (فرداست) ترشی و نوشیدنی در مدل U-Type مشخص نشده‌است، بنابراین پیش فرض آن **اختیاری** لحاظ می‌شود که به تبع زیرنوع (فرداست) نیز **دسته ناقص** خواهد شد. که مطابق گزینه چهارم است، به صورت زیر:



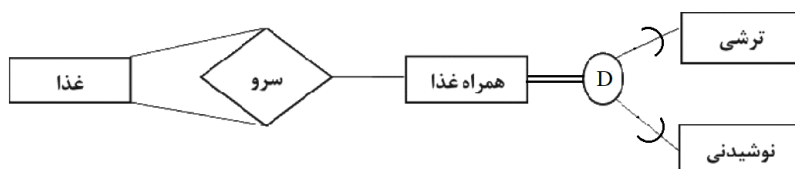
همچنین معادل مدل U-Type فوق با رعایت **پیش شرط** $CG = CE + CF$ به صورت زیر است:



اگر در صورت سوال بیان می‌شد که شرکت همه نمونه‌های زیرنوع (فرداست) ترشی و نوشیدنی **اجباری** است که به تبع زیرنوع (فرداست) نیز **دسته کامل** خواهد شد. آنگاه مدل U-Type آن به صورت زیر می‌بود:



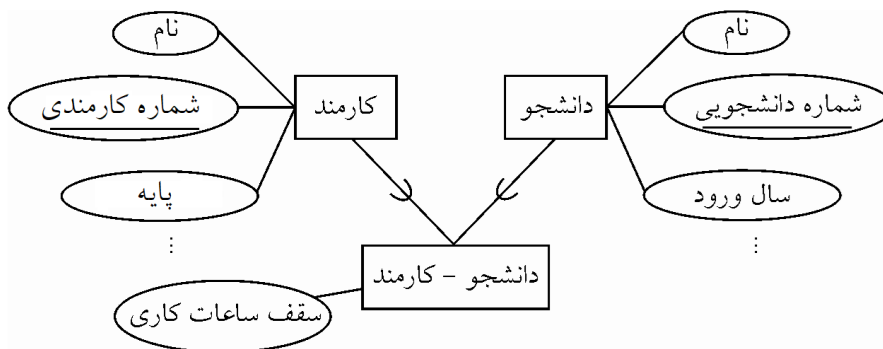
همچنین معادل مدل U-Type فوق به صورت زیر است:



نگاشت زیرنوع ارث‌بری چندگانه به مدل رابطه‌ای

هر موجودیت به یک جدول تبدیل می‌گردد و **کلید کاندید** موجودیت‌های زیرنوع در موجودیت زیرنوع به عنوان **کلید خارجی** تعریف می‌گردد. در جدول زیرنوع کلید خارجی همزمان کلید کاندید نیز می‌باشد.

مدل تحلیل:



مدل طراحی:

<u>EMID</u>	Name	...	<u>STID</u>	Name	..
10	E1		100	E1	
20	E2		200	E2	
30	E3		300	E4	

جدول کارمند

جدول دانشجو

<u>EMID</u>	<u>STID</u>	MAXW
10	100	8
20	200	12

جدول دانشجو کارمند

Employee (EMID , Name , Grade)

Student (STID , Name , Edate)

STEM (EMID , STID , MAXW)

توجه: شماره کارمندی 30 فقط کارمند و شماره دانشجویی 300 فقط دانشجو است.

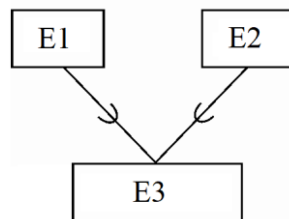
توجه: زیرنوع ارث‌بری چندگانه با اتصال به زیرنوع‌های خود توسط **کلید خارجی**، تمام صفات زیرنوع‌های خود را به ارث می‌برد. به عبارت دیگر مجموعه صفات «**قابل دسترسی**» جدول

زیرنوع STEM، **اجتماع** تمام مجموعه صفات جداول زیرنوع‌ها است. همچنین در اجتماع مجموعه صفات جداول زیرنوع‌ها، صفات تکراری مثل Name **فقط یکبار** شمارش می‌شود.

توجه: جدول STEM دو کلید کاندید مجزا و مستقل یعنی EMID و STID دارد که هر کدام به تنهایی خاصیت یکتا برای کل جدول STEM دارد، چون همه رکوردهای جدول کارمند-دانشجو، تک تک و به طور هم‌زمان هم کارمند است و هم دانشجو.

توجه: اگر زیرنوع، تعداد N زیرنوع داشته باشد، جدول زیرنوع **حداقل** N کلید کاندید مجزا و مستقل دارد. چرا حداقل؟ چون ممکن است هر زیرنوع خودش حداقل دو کلید کاندید مجزا و مستقل داشته باشد که به ازای تعریف هر کلید خارجی در جدول زیرنوع همان کلید کاندید هم می‌شود، هرچند که در عمل به ازای هر زیرنوع، تعریف فقط یک کلید خارجی در جدول زیرنوع کافی است.

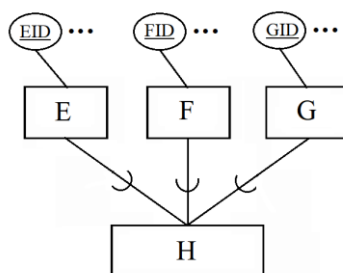
سوال: در نمودار زیر، نوع موجودیت E3:



- (۱) دو شناسه مستقل دارد.
- (۲) یک شناسه ترکیبی دارد.
- (۳) حداقل دو شناسه مستقل دارد.
- (۴) دو شناسه ترکیبی دارد.

پاسخ: گزینه «۳» صحیح است.

سوال: با توجه به نمودار EER زیر، کلید کاندید (شناسه) نوع موجودیت H چیست؟



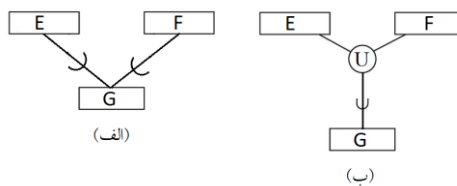
- (۱) $HID \in \{EID, FID, GID\}$
- (۲) $HID \in \{(EID, GID), (EID, FID), (GID, FID)\}$

$$HID = \{EID, FID, GID\} \quad (3)$$

(4) هیچکدام

پاسخ: گزینه «۱» صحیح است. چون EID یا FID یا GID هریک به تنهایی خاصیت یکتا برای کل جدول H دارد.

سوال: با توجه به نمودار (الف) و (ب) و گزاره‌های I و II، مورد درست کدام است؟



گزاره I: G یا صفات E را به ارث می‌برد یا صفات F را.

گزاره II: G هم صفات E را به ارث می‌برد و هم صفات F را.

(۱) گزاره II برای (ب) و گزاره I برای (الف)

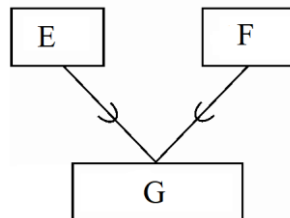
(۲) گزاره II برای (الف) و گزاره I برای (ب)

(۳) گزاره I برای (الف) و گزاره I برای (ب)

(۴) گزاره II برای (الف) و گزاره II برای (ب)

پاسخ: گزینه «۲» صحیح است. گزاره I به معنای U-Type، ارث‌بری انتخابی است.

سوال: با توجه به نمودار زیر، اگر A_X مجموعه صفات نوع موجودیت X باشد، در اینصورت:



آنگاه مجموعه صفات «قابل دسترسی» توسط G چگونه است؟

$$A_G = A_E \cup A_F \quad (1)$$

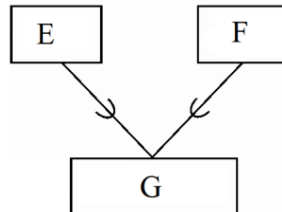
$$A_G \supseteq A_E \cup A_F \quad (2)$$

$$A_G = A_E \cap A_F \quad (3)$$

$$A_G = (A_E - A_F) \cup (A_F - A_E) \quad (4)$$

پاسخ: گزینه «۲» صحیح است. چون موجودیت G ممکن است صفات خاصه خودش را نیز اضافه کند.

سوال: با توجه به نمودار زیر، اگر $Card[A_X]$ کاردینالیته مجموعه صفات نوع موجودیت X باشد، در این صورت :



آنگاه شمارش عددی مجموعه صفات «قابل دسترسی» توسط G چگونه است؟

$$Card[A_G] = Card[A_E] + Card[A_F] \quad (1)$$

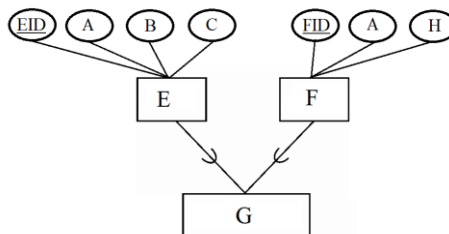
$$Card[A_G] \leq Card[A_E] + Card[A_F] \quad (2)$$

$$Card[A_G] < Card[A_E \cup A_F] \quad (3)$$

$$Card[A_G] \geq Card[A_E \cup A_F] \quad (4)$$

پاسخ: گزینه «۴» صحیح است. چون موجودیت G ممکن است صفات خاصه خودش را نیز اضافه کند.

سوال: با توجه به نمودار EER زیر، اگر برای نوع موجودیت G ، یک جدول طراحی کنیم، تعداد مجموعه صفات «قابل دسترسی» چند خواهد بود؟ فرض کنید جدول G ستون اضافه دیگری ندارد.



(۱) پنج (۲) حداقل شش (۳) هفت (۴) شش

پاسخ: گزینه «۴» صحیح است. مجموعه صفات «قابل دسترسی» جدول زیرنوع G ، اجتماع تمام مجموعه صفات جداول زیرنوع‌ها است. همچنین در اجتماع مجموعه صفات جداول زیرنوع‌ها، صفات تکراری مثل A ، فقط یکبار شمارش می‌شود.

$$\{EID, A, B, C\} \cup \{FID, A, H\} = \{EID, A, B, C, FID, H\}$$

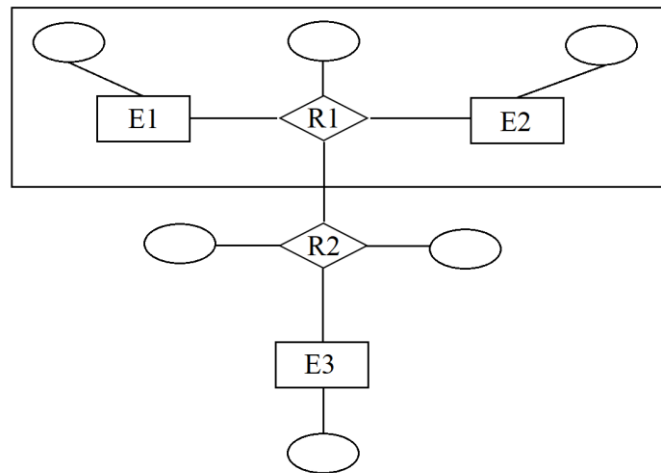
همانطور که گفتیم، در مدل EER روابط بیشتری مطرح شده است که «مدل سازی رابطه تجزیه و ترکیب (Composition و Decomposition)» و «مدل سازی رابطه وراثت» بررسی شد، در ادامه به تشریح «مدل سازی رابطه ارتباط با ارتباط» می پردازیم.

۳- مدل سازی رابطه ارتباط با ارتباط

این نوع ارتباط، بین ارتباط با ارتباط است.

تجمیع یا گروه سازی یا ارتباط با ارتباط (Aggregation)

تجمیع عبارتست از دیدن n ($n \geq 1$) نوع موجودیت که باهم در یک نوع ارتباط شرکت دارند، به صورت یک نوع موجودیت انتزاعی. در واقع مجموعه ای از نوع موجودیت های مرتبط را باهم مجتمع کرده و به عنوان یک نوع موجودیت دیگر، در نظر می گیریم. این نوع موجودیت خود می تواند با نوع موجودیت های دیگری ارتباط داشته باشد. به طور شهودی می توان گفت که زمانی از تکنیک تجمیع استفاده می کنیم که بخواهیم یک نوع ارتباط با یک نوع ارتباط دیگر را بیان کنیم، به ویژه اگر نوع ارتباط مورد نظر خود صفاتی هم داشته باشد. شکل زیر گویای مطلب است:

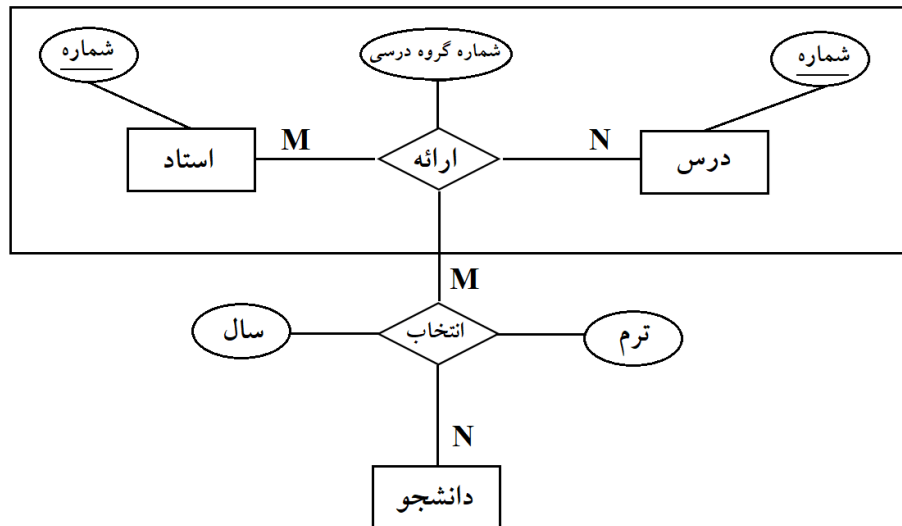


نگاشت رابطه (ارتباط با ارتباط) به مدل رابطه ای

در این حالت، ابتدا نمودار EER جزئی درون مستطیل، نشان دهنده نوع موجودیت انتزاعی حاصل از عمل تجمیع، مطابق مدل های نگاشت معمول، به جداول لازم تبدیل می شود. سپس با توجه به چندی و نحوه ارتباط بین موجودیت مرتبط با مستطیل انتزاعی، طراحی مطابق مدل های نگاشت معمول کامل می شود.

مثال: نمودار EER زیر را به مدل طراحی تبدیل نمایید.

مدل تحلیل:



مدل طراحی:

PROF (PRID , ...)

COUR (COID , ...)

STUD (STID , ...)

PRCO (PRID, COID , SECNUM)

ENRO (STID, PRID , COID, TR, YR)

توجه: جدول PRCO (مستطیل انتزاعی) مطابق مدل نگاشت معمول چند به چند M:N ایجاد شده است. و متشکل از کلید کاندید جدول PROF یعنی PRID و کلید کاندید جدول COUR یعنی COID است.

توجه: صفات PRID, COID, باهم، در جدول ENRO، کلید خارجی به جدول PRCO است.

توجه: جدول ENRO مطابق مدل نگاشت معمول چند به چند M:N ایجاد شده است، و متشکل از کلید کاندید جدول STUD یعنی STID و کلید کاندید جدول PRCO یعنی PRID, COID است.

توجه: در حالت کلی اگر درجه ارتباط داخل مستطیل انتزاعی n باشد، حداکثر تعداد جداول لازم در حالت همه چند به چند n+3 است.

یادآوری: همانطور که گفتیم مدل رابطه‌ای از دو بخش زیر تشکیل شده است:

۱- رابطه یا جدول

۲- قوانین مدل رابطه‌ای

رابطه یا جدول بررسی شد، در ادامه به بررسی قوانین مدل رابطه‌ای می‌پردازیم:

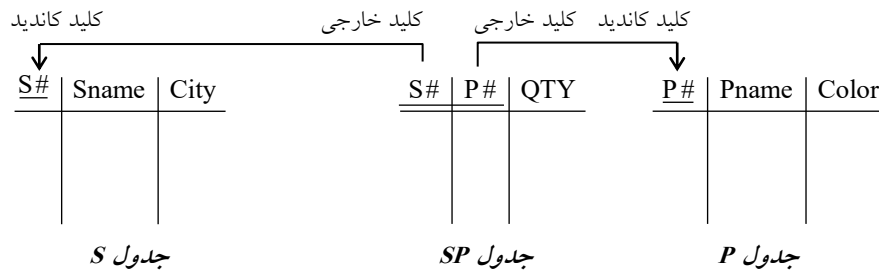
قوانین مدل رابطه‌ای

همانطور که گفتیم به طور کلی محدودیت‌های جامعیتی (Integrity Constraints) در سیستم‌های بانکی به دو طبقه‌ی **جامعیت داخلی** و **جامعیت خارجی** (قواعد کسب و کار یا Business rules) تقسیم می‌گردد. به حفظ قوانین مطرح شده از سوی مدل رابطه‌ای و DBMS در سطح پیاده‌سازی، **جامعیت داخلی** و به حفظ قوانین مطرح شده از سوی کسب‌وکار و اجرا شده توسط طراحان و برنامه‌نویسان بانک در سطح پیاده‌سازی، **جامعیت خارجی** گفته می‌شود. DBMS مسئول کنترل محدودیت‌های جامعیتی داخلی و خارجی در بانک است و هرگونه عاملی که باعث نقض محدودیت‌های جامعیتی داخلی و خارجی بانک گردد را رد می‌کند. قوانین داخلی بانک شامل **قانون جامعیت درون رابطه‌ای**، **قانون جامعیت موجودیت**، **قانون جامعیت ارجاعی** و **قانون جامعیت دامنه‌ای** است. قوانین خارجی بانک هم شامل هر قانونی است که مربوط به قواعد کسب‌وکار است و توسط طراحان و برنامه‌نویسان بانک پیاده‌سازی می‌شود، مثل تعریف بازه 0 تا 20 برای نمرات یک دانشگاه، تعریف بازه 0 تا بینهایت برای حساب‌های بانکی به معنی عدم وجود موجودی منفی در حساب‌های بانکی...

توجه: در سیستم‌های بانکی کنترل محدودیت‌های جامعیتی داخلی و خارجی به صورت خودکار توسط مکانیزم‌های موجود در DBMS انجام می‌گردد. اما طراحی و پیاده‌سازی محدودیت‌های جامعیتی داخلی و خارجی پایگاه داده برعهده برنامه‌نویسان است. حال یکبار دیگر کد تعریف جدول SP و روابط بین جداول را در نظر بگیرید:

Create Table SP

```
(
  S# char (5),
  P# char (5),
  QTY numeric (10),
  Primary key (S#, P#),
  Foreign key (S#) References S(S#)
    on delete cascade
    on update cascade,
  Foreign key (P#) References P(P#)
    on delete cascade
    on update cascade,
  Check (QTY>1 AND QTY<1000)
)
```



کارکرد قطعه کد زیر از کد تعریف جدول SP فوق به صورت زیر است:

Foreign key (S#) References S(S#)

on delete cascade

on update cascade

کاربرد دستور Cascade: اگر سطرهای جدول اصلی (S) حذف (on delete cascade) یا بروزرسانی (on update cascade) شود، آنگاه سطرهای جدول فرعی (SP) که توسط کلید خارجی به آن ارجاع کرده است نیز حذف یا بروزرسانی می‌شود. برای مثال اگر مقدار S1 در جدول S به S11 بروزرسانی شود و این مقدار در جدول SP نیز موجود باشد، آنگاه مطابق رابطه Cascade مقدار S1 در جدول SP به مقدار S11 نیز بروزرسانی می‌شود. برای مثالی دیگر اگر سطر S1 از جدول S حذف شود و این مقدار در جدول SP نیز موجود باشد، آنگاه مطابق رابطه Cascade سطر S1 از جدول SP نیز حذف می‌شود.

بنابراین قطعه کد فوق، سبب می‌گردد تا به طور خودکار هرگونه تغییری در ستون S# در جدول S به ستون S# در جدول SP نیز اعمال گردد. بنابراین **جامعیت داخلی** از نوع **جامعیت ارجاعی** نقض نمی‌گردد.

یا به طور مشابه، کارکرد قطعه کد زیر از کد تعریف جدول SP فوق به صورت زیر است:

Foreign key (P#) References P(P#)

on delete cascade

on update cascade

این قطعه کد، سبب می‌گردد تا به طور خودکار هرگونه تغییری در ستون P# در جدول P به ستون P# در جدول SP نیز اعمال گردد. بنابراین **جامعیت داخلی** از نوع **جامعیت ارجاعی** نقض نمی‌گردد.

کارکرد قطعه کد زیر از کد تعریف جدول SP فوق به صورت زیر است:

Check (QTY>1 AND QTY<1000)

این قطعه کد، سبب می‌گردد تا به طور خودکار هرگونه مقداردهی در ستون QTY از جدول SP در بازه 1 تا 1000 باشد، بنابراین **جامعیت خارجی** نقض نمی‌گردد.

در ادامه به بررسی قوانین جامعیت داخلی می‌پردازیم:

۱- قانون جامعیت درون رابطه‌ای (Intra-Relational Integrity Rule)

مطابق قانون جامعیت درون رابطه‌ای، هر رابطه (جدول) باید به تنهایی درست باشد. یعنی صفات رابطه اتمیک باشد (چند مقداری و مرکب نباشد) و همچنین هر رابطه باید حتماً حداقل دارای یک کلید کاندید باشد. برای مثال درج مقدار تکراری در ستون S# از جدول S باعث نقض قانون جامعیت درون رابطه‌ای می‌شود، چون ستون کلید کاندید نباید مقدار تکراری داشته باشد.

۲- قانون جامعیت موجودیت (Entity Integrity Rule)

مطابق قانون جامعیت موجودیت، هیچگاه نباید تمام یا بخشی از کلید کاندید مقدار NULL داشته باشد. برای مثال درج مقدار NULL در ستون S# از جدول S باعث نقض قانون جامعیت موجودیت می‌شود.

۳- قانون جامعیت دامنه‌ای (Domain Integrity Rule)

مطابق قانون جامعیت دامنه‌ای، صفات رابطه می‌بایست دارای نوع (دامنه) باشد و مقادیر صفات در محدوده همان دامنه باشد. برای مثال درج مقدار رشته‌ای در ستون عددی QTY از جدول SP باعث نقض قانون جامعیت دامنه‌ای می‌شود.

۴- قانون جامعیت ارجاعی (Referential Integrity Rule)

مطابق قانون جامعیت ارجاعی، هیچگاه نباید کلید خارجی، دچار ارجاع NULL گردد. برای مثال درج مقدار S4 در ستون S# از جدول SP بدون وجود مقدار S4 در ستون S# از جدول S باعث نقض قانون جامعیت ارجاعی می‌شود. برای این منظور باید تعریف ساختاری و محتوایی کلید خارجی برقرار باشد. همانطور که گفتیم کلید خارجی برای ارتباط میان جداول مورد استفاده قرار می‌گیرد. کلید خارجی در دو دیگاه ساختاری و محتوایی مورد بررسی قرار می‌گیرد:

دیدگاه ساختاری کلید خارجی

تعریف: اگر صفت(هایی) در یک جدول به عنوان کلید خارجی تعریف شود،

اول اینکه: این صفت(ها) در جدول خودش شرط خاصی ندارد یعنی الزاماً خاصیت کلیدی ندارد.
و دوم اینکه: این صفت(ها) در همان جدول یا جدول دیگری، باید کلید کاندید (اصلی یا فرعی) باشد.

دیدگاه محتوایی کلید خارجی

به ازای هر مقدار موجود در یک کلید خارجی، باید دقیقاً یک مقدار متناظر در کلید کاندید متناظر آن وجود داشته باشد، در غیر این صورت می‌گوییم، کلید خارجی دارای ارجاع NULL است. به بیان دیگر، مقادیر کلید خارجی همواره باید زیرمجموعه مقادیر کلید کاندید باشد و این مورد توسط DBMS حفظ شود.