موسسه بابان

انتشارات بابان و انتشارات راهیان ارشد

پایگاه داده‌ها

فصل ششم: نرمال‌سازی(درسنامه)

(ویژه کنکور ارشد ۱۴۰۵)

ویژه‌ی داوطلبان کنکور کارشناسی ارشد مهندسی کامپیوتر و IT

براساس کتب مرجع

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

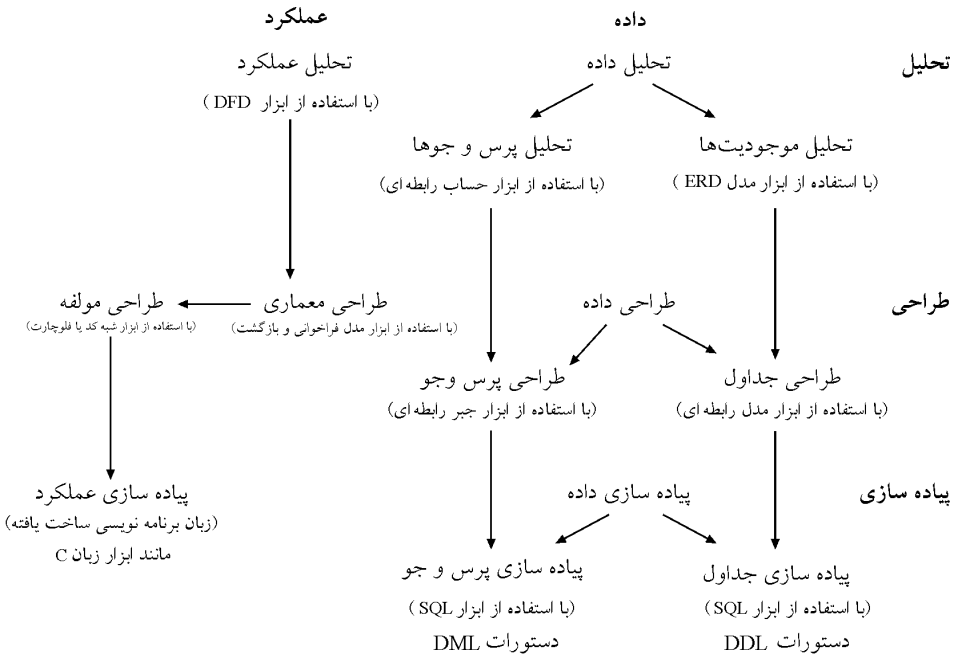
ارسطو خلیلی‌فر

کلیه‌ی حقوق مادی و معنوی این اثر در سازمان اسناد و کتابخانه‌ی ملی ایران به ثبت رسیده است.

۶ نرمال‌سازی

مقدمه

همانطور که در فصل وابستگی تابعی گفتیم، از نظر جانمایی مبحث **نرمال‌سازی** مربوط به همان فعالیت طراحی و مدل رابطه‌ای و پیش از شروع پیاده‌سازی است. چون جداول اگر غیرنرمال باشند، اول باید طراحی نرمال شوند، سپس پیاده‌سازی شوند تا از پیاده‌سازی غیرنرمال جلوگیری شود. شکل زیر گویای مطلب است:



جداول **غیرنرمال** به مشکل **افزونگی محتوایی (طبیعی یا داده‌ای)** مبتلا هستند و از آن **رنج**

می‌برند که خود این افزونگی ریشه در **تکرار اطلاعات** دارد و تکرار اطلاعات ریشه در وجود **وابستگی‌های بد** (افزونگی‌ساز و خرابکار) بین ستون‌های جدول دارد. راه **درمان** جداول **غیرنرمال (کثیف)** و تبدیل آن به جداول **نرمال (تمیز)**، فرآیند **نرمال‌سازی** است.

به طور کلی سه نوع وابستگی بین ستون‌های جداول وجود دارد:

- **وابستگی تابعی (FD: Functional Dependency)**
- **وابستگی چند مقداری (MVD: Multi-Valued Dependency)**
- **وابستگی الحاقی (JD: Join Dependency)**

توجه: وابستگی چندمقداری (MVD) و وابستگی الحاقی (JD) نوع دیگری از وابستگی‌ها هستند که **تابعی نیستند**. شرح **خوب** و **بد** در ادامه همین بررسی می‌شود.

به طور کلی دو نوع **وابستگی تابعی** بین ستون‌های جداول وجود دارد:

الف) وابستگی‌های تابعی خوب:

۱- وابستگی تابعی ساده یا ابرکلید (FD: Functional Dependency)

۲- وابستگی تابعی کامل (تام) یا کلید کاندید (FFD: Fully Functional Dependency)

ب) وابستگی‌های تابعی بد:

۱- وابستگی تابعی بخشی (غیرکامل) (PFD: Partial Functional Dependency)

۲- وابستگی تابعی انتقالی (تراگذری) (TFD: Transitive Functional Dependency)

۳- وابستگی تابعی معکوس(RFD: Reverse Functional Dependency)

در فصل سوم (مدل رابطه‌ای) ، فرآیند تبدیل **مدل تحلیل** (نمودار ER) به **مدل طراحی** (جداول) را دیدیم و گفتیم که اگر این فرآیند **به‌درستی** انجام شود، نتیجه آن جداول **نرمال** (خوش طرح) است. اما اگر این فرآیند به درستی انجام نشود، نتیجه آن جداول **غیرنرمال (کثیف)** است که دچار **تکرار اطلاعات** و به تبع دچار **افزونگی داده‌ای (محتوایی یا طبیعی)** است و از آن **رنج** می‌برد که با عملیات **نرمال‌سازی** باید **درمان** شود. به مثال زیر دقت کنید:

مثال: جدول غیرنرمال studclg با مقادیر زیر را در نظر بگیرید:

وابستگی تابعی بد (از نوع وابستگی تابعی انتقالی)

S#	Sname	Clg#	Clgname
8621	Ali	12	Computer
8442	Reza	12	Computer
8731	Abbas	NULL	NULL
9215	Hassan	13	Bargh

افزونگی طبیعی

اثرات وابستگی تابعی بد $Clg\# \rightarrow Clg\,name$ **در جدول studclg :**

۱- **افزونگی داده‌ای:** این اطلاعات که «کد دانشکده‌ی کامپیوتر برابر 12 است.» در سطرهای اول و دوم تکرار شده است، پس **افزونگی داده‌ای (محتوایی یا طبیعی)** وجود دارد که باعث هدر رفتن حافظه شده‌است.

۲ – **آنومالی:** اگر شخصی بخواهد کد دانشکده‌ی کامپیوتر را به 22 تغییر دهد و به اشتباه، فقط مقدار Clg# در سطر اول جدول studclg را عوض کند، از این به بعد دانشکده کامپیوتر دو کد خواهد داشت: 12 و 22 ، یعنی بالاخره کد دانشکده کامپیوتر 12 هست یا 22 ؟ این موضوع یعنی **آنومالی (بی‌نظمی، ناسازگاری، ناهنجاری)** در **آپدیت**. به این مدل ناسازگاری، **ناسازگاری ارجاعی** هم گفته می‌شود چون یک مقدار است که همه جا باید یکسان باشد ولی در اثر تغییر در جدول غیرنرمال همه جا یکسان نیست. پدیده ناسازگاری ارجاعی با نرمال‌سازی و تعریف کلید خارجی مرتفع می‌شود. **ناسازگاری داده‌ای** انواع مختلف دارد که در فصل مفاهیم پایه بررسی شد.

۳ – **مقادیر NULL:** در سطر 3 از جدول studclg، مقادیر **NULL** وجود دارد. به طور کلی، به ازای هر سطر از جدول غیرنرمال studclg که Clg# آن برابر NULL باشد یک سطر در جدول studclg خواهیم داشت که مقادیر دو ستون آخر آن NULL خواهد بود. پس مشکل مقادیر NULL نیز وجود دارد.

توجه: برقراری وابستگی تابعی بد (از نوع وابستگی تابعی انتقالی) $Clg\# \rightarrow Clg\,name$ از جدول studclg به عنوان **محدودیت جامعیتی** به طور **صریح** توسط دستور **Assertion** و **Trigger** کنترل می‌شود.

انواع آنومالی در عملیات ذخیره‌سازی (درج، حذف و آپدیت)

۱ – عدم امکان انجام یک عمل (که منطقا باید قابل انجام باشد)

۲ – بروز پیامد بد (عارضه جانبی) پس از انجام یک عمل

۳ – بروز فزونکاری در سیستم در انجام یک عمل

آنومالی در درج: اگر رکورد (Computer , 22 , sajad , 5261) درج شود. از این به بعد دانشکده کامپیوتر دو کد خواهد داشت: 12 و 22 یعنی بالاخره کد دانشکده کامپیوتر 12 هست یا 22 ؟ این آنومالی از نوع **«بروز پیامد بد (عارضه جانبی) پس از انجام یک عمل»** است. برای مثالی دیگر اگر بخواهیم فقط رکورد (mathematic , 14) را برای واقعیت مشخصات دانشکده ریاضی درج کنیم، تا زمانی‌که S# برای آن مقدار ندارد، امکان درج رکورد (mathematic , 14) نیست چون S# کلید کاندید جدول غیرنرمال stuclg است و محدودیت جامعیت موجودیت جلوی درج آنرا

می‌گیرد. این آنومالی از نوع **«عدم امکان انجام یک عمل (که منطقا باید قابل انجام باشد)»** است.

آنومالی در حذف: اگر قرار باشد فقط اطلاعات یک دانشکده (Clg# , Clgname) حذف شود در کنار آن، اطلاعات دانشجویی (S# , Sname) هم ناخواسته از دست می‌رود. این آنومالی از نوع **«بروز پیامد بد (عارضه جانبی) پس از انجام یک عمل»** است.

آنومالی در آپدیت: اگر کد دانشکده‌ی کامپیوتر به 22 تغییر کند و به اشتباه، فقط مقدار Clg# در سطر اول جدول studclg تغییر کند، از این به بعد دانشکده کامپیوتر دو کد خواهد داشت. بنابراین به طور دستی باید کد دانشکده‌ی کامپیوتر در همه سطرها به 22 تغییر کند. این آنومالی از نوع **«بروز فزونکاری در سیستم در انجام یک عمل»** است.

نتیجه: علت آنومالی‌های فوق، وجود **وابستگی‌های تابعی بد** در جدول **studclg** است، که با حذف وابستگی‌های بد، آنومالی در عملیات ذخیره‌سازی (درج، حذف و آپدیت) هم ایجاد نمی‌شود.

نتیجه: اگر فرآیند تبدیل **مدل تحلیل** (نمودار ER) به **مدل طراحی** (جداول) به‌درستی انجام نشود، نتیجه آن جداول **غیرنرمال** است که دچار مشکلات زیر است:

۱- افزونگی داده‌ای یا طبیعی یا محتوایی (Data Redundancy)

۲- آنومالی (مشکل در درج، حذف و آپدیت)

۳- مقادیر تهی (NULL Values)

نتیجه: برای مقدار **NULL** ، فضا برای آن تخصیص می‌یابد اما برای هیچ و پوچ که باعث هدر رفتن حافظه می‌شود.

با **تجزیه جدول** studclg به دو جدول clg و stud ، وابستگی تابعی بد $Clg\# \rightarrow Clg\,name$ از جدول studclg **حذف** می‌شود و مشکل **درمان** می‌شود.

حال اگر جدول studclg را مطابق فرآیند **نرمال‌سازی** به دو جدول clg و stud **تجزیه** کنیم:

Clg#	Clgname
12	Computer
13	Bargh

جدول clg

S#	Sname	Clg#
8621	Ali	12
8442	Reza	12
8731	Abbas	NULL
9215	Hassan	13

جدول Stud

مشکلات فوق مرتفع و نتایج زیر حاصل خواهد شد:

۱ – **کاهش افزونگی داده‌ای** (طبیعی یا محتوایی) مانند تکرار بی‌دلیل سطر (12,Computer)

۲ – **افزایش افزونگی تکنیکی** (فنی یا ساختاری) به دلیل تعریف کلید خارجی

۳ – **کاهش مقادیر NULL**

۴ – **عدم آنومالی** (عدم مشکل در درج، حذف و آپدیت)

توجه: کاهش **بعضی** از انواع افزونگی در فرآیند نرمال‌سازی، یعنی افزونگی طبیعی کاهش می‌یابد، اما به واسطه عمل تجزیه و تعریف کلید خارجی (در قالب ستون‌های مشترک)، افزونگی تکنیکی (ساختاری) افزایش می‌یابد.

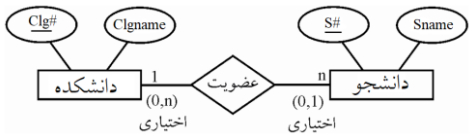
توجه: هرجا جدول دیدیم به آن نمی‌گوییم پایگاه داده نرمال، بلکه هرجا جدول نرمال دیدیم به آن می‌گوییم پایگاه داده نرمال.

توجه: دقت کنید نرمال‌سازی زمانی کاربرد دارد که خود مدل رابطه‌ای در سطح مدل طراحی به شکل بهینه انجام نشود، برای مثال رابطه یک به چند بین دو موجودیت در یک ERD یک مدل رابطه‌ای **بهینه** دارد. که اگر عمل نگاشت مدل تحلیل به مدل طراحی بهینه انجام شود جداول حاصل خود به خود نرمال هستن و نیاز به نرمال‌سازی ندارد.

نگاشت رابطه یک به چند بین دو موجودیت 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.

مدل تحلیل:



مدل طراحی درست:

Clg#	Clgname
12	Computer
13	Bargh

جدول clg

S#	Sname	Clg#
8621	Ali	12
8442	Reza	12
8731	Abbas	NULL
9215	Hassan	13

جدول Stud

توجه: اگر دانشجویی عضو دانشکده‌ای نباشد، ستون #Clg برای آن مقدار NULL می‌گیرد، مانند شماره دانشجویی 8731 در جدول Stud. همچنین #Clg در جدول Stud **کلید خارجی** است.

توجه: همانطور که مشاهده می‌کنید اگر اصول تحلیل (مدل ER) و طراحی (مدل رابطه‌ای) رعایت شود. جداول حاصل، خود به خود نرمال هستن و نیاز به نرمال‌سازی ندارد.

بررسی سربار سیستم (Overhead) در عمل بازیابی قبل و بعد از نرمال‌سازی

جدول **غیرنرمال** studclg به صورت زیر است:

S#	Sname	Clg#	Clgname
8621	Ali	12	Computer
8442	Reza	12	Computer
8731	Abbas	NULL	NULL
9215	Hassan	13	Bargh

جدول *studclg*

جداول **نرمال** stud و clg نیز به صورت زیر است:

S#	Sname	Clg#
8621	Ali	12
8442	Reza	12
8731	Abbas	NULL
9215	Hassan	13

جدول Stud

Clg#	Clgname
12	Computer
13	Bargh

جدول clg

پرس‌وجو ۱: (بعد از نرمال‌سازی)

```
select S#
from stud
```

ستون #S از جدول نرمال stud استخراج می‌شود و نیازی به الحاق دو جدول stud و clg ندارد.

پرس‌وجو ۲: (قبل از نرمال‌سازی)

```
select S#
from studclg
```

ستون #S از جدول غیرنرمال studclg استخراج می‌شود، اما با زمان واکشی **بیشتر** نسبت به جدول

نرمال stud، چون طول رکوردها در جدول غیرنرمال studclg بیشتر از جدول نرمال stud است.

نتیجه: اگر پرس‌وجو فقط وابسته به یکی از جداول نرمال باشد مثل ستون S# از جدول stud، در این شرایط واکشی S# از جدول stud زمان کمتری نسبت به واکشی S# از جدول studclg دارد، چون در نرخ انتقال برابر، طول رکورد جدول stud از جدول studclg کمتر است.

$$\text{زمان انتقال رکورد } (T_f) = \frac{\text{طول رکورد } (L)}{\text{نرخ انتقال } (R)}$$

پرس‌وجو ۳: (بعد از نرمال‌سازی)

```
select *
from stud, clg
where stud.clg# = clg.clg# AND clgname = 'computer'
```

ستون‌های مشخصات دانشجویان دانشکده کامپیوتر از دو جدول نرمال Stud و clg استخراج می‌شود و نیاز به الحاق دو جدول stud و clg دارد، بنابراین سربار حاصل از الحاق دو جدول را هم دارد.

پرس‌وجو ۴: (قبل از نرمال‌سازی)

```
select *
from studclg
where clgname = 'computer'
```

ستون‌های مشخصات دانشجویان دانشکده کامپیوتر از جدول غیرنرمال studclg استخراج می‌شود، اما با زمان واکشی **کمتر** نسبت به الحاق جداول نرمال stud, clg ، هرچندکه طول رکوردها در جدول studclg **برابر** الحاق جداول stud, clg است. اما در حالت جداول نرمال، برای داشتن تمام ستون‌ها نیاز به یک عمل اضافه و دارای سربار به نام **الحاق** وجود دارد.

نتیجه: نرمال‌سازی باعث تجزیه جداول می‌شود، اگر پرس‌وجو روی جداول نرمال نیاز به الحاق نداشته باشد، سربار حاصل از الحاق ایجاد نمی‌شود، که منجر به **کاهش** زمان پاسخگویی نسبت به جدول غیرنرمال می‌شود. اما اگر پرس‌وجو روی جداول نرمال نیاز به الحاق داشته باشد، سربار حاصل از الحاق ایجاد می‌شود، که منجر به **افزایش** زمان پاسخگویی نسبت به جدول غیرنرمال می‌شود.

فرآیند نرمال‌سازی

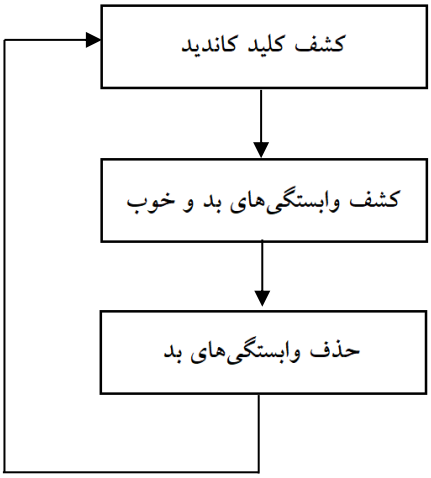
فرآیند نرمال‌سازی یعنی **تبدیل** یک جدول پایه غیرنرمال به چند جدول نرمال از طریق **تجزیه عمودی یا پارسازی عمودی (Vertical DE composition)**. در این فرآیند، انواع **وابستگی‌های بد** به ترتیب، سطح به سطح و به طور سلسله مراتبی (و نه یکباره) **حذف** می‌شوند که به تبع منجر به کاهش تکرار اطلاعات و کاهش افزونگی داده‌ای به طور سلسله مراتبی می‌شود. فرآیند

نرمال‌سازی از **سه قدم** زیر تشکیل شده است:

۱- کشف کلید کاندید

۲- کشف وابستگی‌های بد و خوب

۳- حذف وابستگی‌های بد

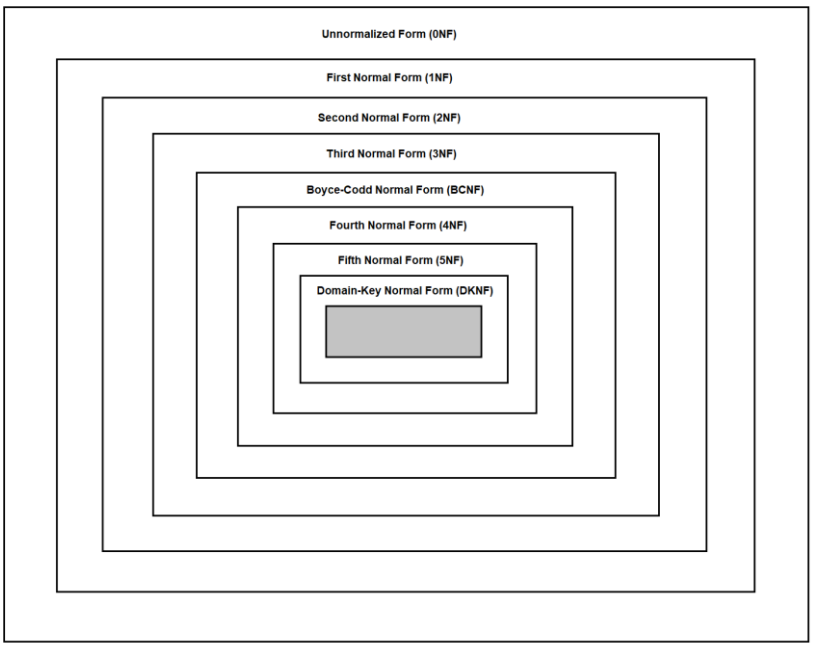


توجه: در دیاگرام فوق، چرخه به این دلیل است که فرآیند نرمال‌سازی، عملی بـه یکبـاره نیسـت، بلکه وابستگی‌های بد، سطح به سطح و به طور سلسله مراتبـی در یـک **رویـه کـاهش** از جـداول حذف می‌شوند، در نتیجه افزونگی داده‌ای نیز سطح به سطح کاهش می‌یابد و نه یکباره.

توجه: فرآیند نرمال‌سازی یک **رویه بازسـازنده (بازگشـت‌پذیر)** اسـت، یعنـی از جـداول نرمـال می‌توان جدول پایه (اولیه) غیرنرمال را به طـور **بی‌کاسـت** (بـدون کـم شـدن سـطر و سـتون) و **بی‌حشو** (بدون اضافه شدن سطر و ستون) مجدد به‌دست آورد. اگر از همان راه که آمدیم، از همان راه هم برگردیم، مجدد به همان نقطه مبدا می‌رسیم.

سطوح نرمال‌سازی

سطوح نرمال‌سازی دارای خواص سلسله مراتبی است، بدین معنی که چنانچـه جـدولی در سـطح kُام نرمال قرار داشته باشد، حتما در سطح k-1 نرمال نیز قرار دارد. بـدیهی اسـت کـه بـا افـزایش سطح نرمال، احراز شرایط آن سطح نیز مشکل‌تر خواهد بود. همچین هر سطح از سطح قبلی خـود نرمال‌تر است. شکل زیر گویای مطلب است:



توجه: شکل فوق نشان می‌دهد، هر سطح داخلی زیر مجموعه سطح خارجی است:

$$5NF \subset 4NF \subset BCNF \subset 3NF \subset 2NF \subset 1NF \subset 0NF$$

مثال: 2NFها از مجموعه 1NFها هستند که به 2NF هم رسیدن.

توجه: راه در جلوی نرمال‌سازی باز است، یعنی همیشه می‌توان، جدولی که در یک سطح از نرمال قرار دارد، در سطح بالاتر نرمال آنرا قرار داد.

توجه: فرآیند نرمال‌سازی همانند فرآیند ادامه تحصیل (مقدماتی (0NF)، دیپلم (1NF)، کاردانی (2NF)، کارشناسی (3NF)، کارشناسی ارشد (BCNF) ، دکتری (4NF) و پست دکتری (5NF)) است و همچنین دارای خواص سلسله مراتبی است، بدین معنی که چنانچه فردی در مقطع دکتری باشد، حتما کارشناسی ارشد و به قبل هم دارد، اما عکس آن صادق نیست، یعنی اگر شخصی کارشناسی ارشد داشته باشد، دلیل نمی‌شود که حتما دکتری هم دارد. همچنین، شخصی که کارشناسی دارد، همیشه راه برای ادامه تحصیل آن تا سطوح بالاتر وجود دارد.

توجه: به سطوح نرمال، **صورت نرمال** و **دیس بهنجار** هم گفته می‌شود.

شروط قرار داشتن یک جدول در یک سطح از نرمال

شرط لازم: در نرمال سطح قبلی قرار داشته باشد.

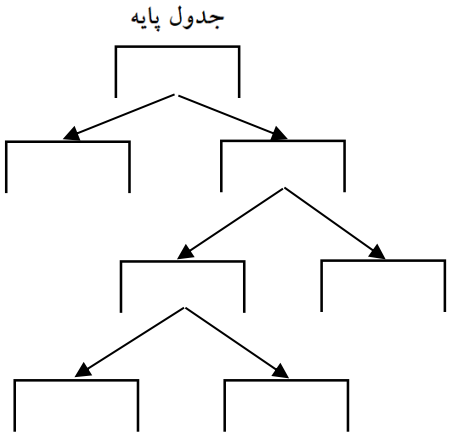
شرط کافی: قواعد تعریف شده در آن سطح از نرمال را پاس کرده باشد.

مثال: یک شخص فارغ التحصیل مقطع دکتری است:

شرط لازم: کارشناسی ارشد داشته باشد.

شرط کافی: شرایط مقطع دکتری را به طور کامل پاس کرده باشد.

توجه: فرآیند نرمال‌سازی، مبتنی بر **تجزیه** جداول است، و حاصل آن یک **درخت تجزیه** است که **برگ‌های** آن **جداول نرمال** (که همگی در سطح kُام نرمال قرار دارند) و **ریشه** آن **جدول پایه** غیرنرمال است.



توجه: از نظر تئوری، تعداد سطوح نرمال، می‌تواند تا هر سطحی ادامه یابد، اما از نظر عملی در اغلب موارد رسیدن به سطح سوم نرمال (3NF) برای یک جدول لازم و کافی است.

مشکلات پایگاه داده غیرنرمال

هر جدول ممکن است مشکلات مختلفی داشته باشد. در جدول زیر مشکلات پایگاه داده آمده است:

شماره	مشکل	راه حل
1	وجود صفت چندمقداری و مرکب	نرمال فرم اول (1NF)
2	وجود وابستگی تابعی بخشی	نرمال فرم دوم (2NF) برای حذف وابستگی بخشی
3	وجود وابستگی تابعی انتقالی	نرمال فرم سوم (3NF) برای حذف وابستگی انتقالی
4	وجود وابستگی تابعی معکوس	نرمال فرم (BCNF) برای حذف وابستگی معکوس
5	وجود وابستگی چندمقداری	نرمال فرم چهارم (4NF) برای حذف وابستگی چندمقداری
6	وجود وابستگی الحاقی	نرمال فرم پنجم (5NF) برای حذف وابستگی الحاقی

نرمال فرم اول (First Normal Form-1NF)

فرآیند نرمال‌سازی یک روند سطح به سطح است. سطح **نخست** نرمال‌سازی، تبدیل جداول به **نرمال فرم اول** است.

شروط قرار داشتن یک جدول در نرمال فرم اول به صورت زیر است:

شرط لازم: در نرمال سطح قبلی قرار داشته باشد، که قرار دارد چون نقطه شروع 0NF است.

شرط کافی: قواعد تعریف شده در آن سطح از نرمال را پاس کرده باشد، که به صورت زیر است:

۱ – دارای حداقل یک کلید کاندید باشد.

۲ – همه ستون‌های آن غیرقابل تجزیه و ساده باشند (جدول باید فاقد ستون مرکب باشد)

۳ – همه ستون‌های آن تک‌مقداری باشند (جدول باید فاقد ستون چندمقداری باشد)

توجه: شرط کافی، همان قانون **جامعیت درون رابطه‌ای** (Intra-Relational Integrity Rule) است، که مطابق این قانون، هر رابطه (جدول) باید به تنهایی درست باشد. یعنی صفات رابطه اتومیک باشد (چند مقداری و مرکب نباشد) و همچنین هر رابطه باید حتماً حداقل دارای یک کلید کاندید باشد.

توجه: 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 جداول مدل رابطه‌ای **نباید مرکب** مثل تاریخ تولد و آدرس باشد و **نباید چندمقداری** مثل مدرک تحصیلی و شماره تلفن 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 در تاپل اول یعنی (a,x) از $((a,x),1)$ خودش یک تاپل است که مطابق قوانین مدل رابطه‌ای، هر مولفه در یک تاپل، نمی‌تواند خودش یک تاپل باشد.

مثال ۱ (نرمال‌سازی صفت مرکب تاریخ تولید):

طراحی نادرست (غیرنرمال)

$$R = \{(Sl,Sn1,(1373,7,5)),(S2,Sn2,(1381,6,4))\}$$

SID	Sname	Date of birth
S1	Sn1	1373,7,5
S2	Sn2	1381,6,4

جدول غیرنرمال R

طراحی درست (نرمال) (بدون تجزیه جدول پایه)

$$R = \{(Sl,Sn1,1373,7,5),(S2,Sn2,1381,6,4)\}$$

SID	Sname	Year	Month	Day
S1	Sn1	1373	7	5
S2	Sn2	1381	6	4

جدول نرمال R

نتیجه: جدول نرمال R در **نرمال فرم اول** قرار دارد، چون توسط نرمال‌سازی سه ستون Year ، Month و Day برای آن تعریف شده‌است و از فرم تاپل تو تاپل خارج شده است. در روند فوق، نرمال‌سازی **بدون تجزیه** جدول پایه، و صرفا فقط توسط **درج ستون** انجام شده است. کلید

کاندید جدول نرمال R ستون SID است.

مثال ۲ (نرمال‌سازی صفت چندمقداری شماره تلفن):

طراحی نادرست (غیرنرمال)

$$R = \{(Sl, Sn1, (22445566, 88774411)), (S2, Sn2, (44332211, NULL))\}$$

وابستگی بد (از نوع وابستگی چندمقداری)

SID	Sname	Tel
S1	Sn1	22445566,88774411
S2	Sn2	44332211,NULL

جدول غیرنرمال R

طراحی درست (نرمال) (بدون تجزیه جدول پایه)

$$R = \{(Sl, Sn1, 22445566, 88774411), (S2, Sn2, 44332211, NULL)\}$$

SID	Sname	Tel1	Tel2
S1	Sn1	22445566	88774411
S2	Sn2	44332211	NULL

جدول نرمال R

نتیجه: جدول نرمال R در **نرمال فرم اول** قرار دارد، چون توسط نرمال‌سازی دو ستون Tel1 و Tel2 برای آن تعریف شده‌است و از فرم تاپل تو تاپل خارج شده است. در روند فوق، نرمال‌سازی **بدون تجزیه** جدول پایه، و صرفا فقط توسط **درج ستون** انجام شده است. کلید کاندید جدول نرمال R ستون SID است.

طراحی درست (نرمال) (با تجزیه جدول پایه)

با **تجزیه جدول**، وابستگی بد (از نوع وابستگی چندمقداری) $SID \rightarrow\rightarrow Tel$ از جدول غیرنرمال R **حذف** می‌شود و مشکل **درمان** می‌شود. به صورت زیر:

وابستگی کامل

SID	Sname
S1	Sn1
S2	Sn2

R1

SID	Tel
S1	22445566
S1	88774411
S2	44332211

R2

نتیجه: هر دو جدول R1 و R2 در **نرمال فرم اول** قرار دارند، چون جدول غیرنرمال R توسط نرمال‌سازی به دو جدول R1 و R2 **تجزیه** شد و از فرم تاپل تو تاپل خارج شده است. کلید کاندید

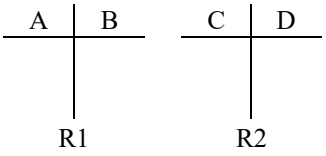
جدول R1 ستون SID و کلید کاندید جدول R2 تمام کلید (ستون‌های SID,Tel باهم) است.

شروط تجزیه مطلوب در نرمال‌سازی

سوال: آیا روال تجزیه دلبخواهی است؟ هرجوری دلم بخواهد است؟

پاسخ: خیر، روال تجزیه دلبخواهی نیست، بلکه برای رسیدن به تجزیه مطلوب، باید **شروط تجزیه مطلوب** محقق شود، وگرنه **تجزیه نامطلوب** می‌شود، که کاربردی هم ندارد.

مثال: رابطه $R(A,B,C,D)$ را با چهار ستون در نظر بگیرید، تجزیه زیر برای آن **تجزیه نامطلوب** است، چون بین دو جدول R1 و R2 **ستون مشترک** وجود ندارد و به تبع ارتباط بین دو جدول برای بازسازی **جدول اولیه** قطع می‌شود.



نتیجه: تجزیه مطلوب باید **ازدواج‌پذیر (پیوندپذیر)** باشد، یعنی بین دو جدول، **ستون مشترک** وجود داشته‌باشد.

در فرآیند نرمال‌سازی شروط **تجزیه مطلوب** به صورت زیر است:

تجزیه مطلوب
- **شرط لازم:** حافظ جدول اولیه باشد. (بی‌کاست و بی‌حشو)
- **شرط کافی:** حافظ FDها باشد.

توجه: همانطور که قبل‌تر گفتیم، **نرمال‌سازی** یک **رویه بازسازنده (بازگشت‌پذیر)** است، یعنی از جداول تجزیه‌شده نرمال، باید بتوان جدول اولیه (پایه) غیرنرمال را به‌دست آورد.

مثال: رابطه $R(A,B,C)$ با مجموعه وابستگی‌های تابعی (FD) زیر مفروض است:

$F = \{ A \rightarrow C \}$

وابستگی تابعی بخشی

عضو کلید کاندید | عضو کلید کاندید | غیرکلید

A	B	C
1	101	**201**
1	102	**201**
3	102	201
4	103	204

R

تجزیه مطلوب به صورت زیر است:

تمام کلید

A	B
1	101
1	102
3	102
4	103

R1

وابستگی تابعی کامل

A	C
1	201
3	201
4	204

R2

دو جدول R1 و R2 الحاق‌پذیر هستند، همچنین **صفت مشترک A** در جدول R2 **کلید کاندیـد** و در جدول R1 **کلید خارجی** است، بنابراین شرط لازم **بی‌کاست** و **بی‌حشو** به طور هم‌زمان محقق شده‌است. خروجی $R_1 \bowtie R_2$ به صورت زیر است:

کلید خارجی — کلید کاندید

A	B
1	101
1	102
3	102
4	103

R1

$\bowtie$

A	C
1	201
3	201
4	204

R2

=

A	B	C
1	101	**201**
1	102	**201**
3	102	201
4	103	204

R

واضح است که R = R1 JOIN R2 برقرار است و جدول اولیه **دقیقا** به‌دست می‌آید، یعنی الحاق (پیوند) بازسازنده (بازگشت‌پذیر) است. در ادامه به شرح شرط لازم و کافی **تجزیه مطلوب** می‌پردازیم:

۱- شرط لازم: (تجزیه، **حافظ تمام ستون‌ها (ساختار) و سطرهای (محتوا) جدول اولیه** باشد.)
Non-Loss (LossLess) DE composition باشد، یعنی **تجزیه بی‌کاست** (بی‌گمشدگی) باشد.
Non additive DE composition باشد، یعنی **تجزیه بی‌حشو** (بی‌اضافه‌شدگی) باشد.
توجه: تجزیه مطلوب، **باید** به طور هم‌زمان هردو شرط **بی‌کاست** و **بی‌حشو** را داشته باشد.
بی‌کاست (بدون کم شدن سطر و ستون نسبت به جدول اولیه): یعنی سطر و ستونی نسبت به جدول اولیه **کم نشود**، این مورد زمانی محقق می‌شود که **صفت مشترک** میان دو جدول وجود داشته باشد، اما کافی نیست چون فقط امکان بازسازی رکوردهای اولیه را دارد و ممکن است رکوردهای اولیه با رکوردهای اضافه‌تری بازسازی شود. بنابراین فقط بی‌کاست بودن از دقت کافی برخوردار نیست.
بی‌حشو (بدون اضافه شدن سطر و ستون جدید نسبت به جدول اولیه): یعنی سطر و ستونی نسبت به جدول اولیه **اضافه نشود**، این مورد زمانی محقق می‌شود که **صفت مشترک** در دو جدول، بر اساس بستار وابستگی‌های تابعی، **حداقل** در یکی از دو جدول **کلید کاندید** باشد. تجزیه بی‌حشو به این معناست که از پیوند دو جدول تجزیه شده تاپل‌های حشو (تاپل‌های اضافی یا تاپل‌های ساختگی که در رابطه اولیه وجود نداشته است) پدید نیاید. در واقع اگر از پیوند دو جدول تجزیه شده، تاپل‌های حشو پدید آید، دیگر اطمینان نداریم که چه تاپل‌هایی اصلی و چه تاپل‌هایی اضافی هستند، بنابراین اطلاعات درست در بازسازی از دست می‌رود. نتیجه اینکه بی‌حشو بودن از بی‌کاست بون مهم‌تر است، ضمن اینکه بی‌حشو بودن، بی‌کاست بودن هم پوشش می‌دهد. **پیوند بی‌حشو بودن**، از نوع **CK-CK** یا **CK-FK** است، که در هیچ‌یک از این دو نوع پیوند، تاپل حشو (اضافه) بروز نمی‌کند، یعنی بازسازنده است و تجزیه بی‌حشو می‌شود. در واقع تجزیه باید به گونه‌ای انجام شود که امکان **بازسازی ساختاری (ستونی)** و **بازسازی محتوایی (سطری) جدول اولیه** براساس جداول تجزیه‌شده امکان‌پذیر باشد، به این صورت که با الحاق جداول تجزیه‌شده باهم، جدول اولیه **دقیقا** به‌دست آید، به طوری که سطر و ستونی از جدول اولیه از دست نرود و همچنین سطر و ستونی به جدول اولیه اضافه نشود، **نه چیزی کم بشه نه چیزی اضافه بشه.** نتیجه اینکه در تجزیه مطلوب، هیچ اطلاعات سطری و ستونی نسبت به جدول اولیه کم و زیاد نمی‌شود، چون با پیوند مجدد جداول تجزیه‌شده، جدول اولیه به طور کامل در دسترس

است.

نتیجه: برای تحقق اینکه نه چیزی کم شود و نه چیزی اضافه شود، باید **صفت مشترک** در دو جدول، بر اساس بستار وابستگی‌های تابعی **حداقل** در یکی از دو جدول تجزیه‌شده **کلید کاندید** باشد. به عبارت دیگر اگر تجزیه به‌گونه‌ای باشد که **تعریف کلید خارجی** برقرار باشد، یعنی دو جدول تجزیه شده R1 و R2 **الحاق‌پذیر** باشند و حتما **صفت مشترک** در دو جدول، **حداقل** در یکی **کلید کاندید** و در دیگری **کلید خارجی** باشد، آنگاه شرط لازم **بی‌کاست** و **بی‌حشو** باهم محقق شده‌است.

نکته: اگر جدول اولیه R به دو جدول R1 و R2 تجزیه **بی‌کاست** و **بی‌حشو** شود، در این حالت R = R1 JOIN R2 برقرار است و جدول اولیه به‌دست می‌آید، یعنی الحاق (پیوند) بازسازنده (بازگشت‌پذیر) است.

چک ستونی تجزیه مطلوب

اینکه بعد از عمل تجزیه، **ستونی کم نشود** و **ستونی اضافه نشود** توسط عملگر اجتماع $\cup$ و به صورت زیر چک می‌شود:

$$R1 \cup R2 = R$$

مثال: رابطه $R(A, B, C)$ با مجموعه وابستگی‌های تابعی (FD) زیر مفروض است:

$$F = \{A \rightarrow C\}$$

پس از تجزیه به دو رابطه $R1(\mathbf{A}, B)$ و $R2(\mathbf{A}, C)$ ، ستون **A** ستون مشترک بین دو جدول است و $R1 \cup R2 = R$ **چک** می‌کند **ستونی کم نشود** و **ستونی اضافه نشود**، به صورت زیر:

$$R1 \cup R2 = \{A, B\} \cup \{A, C\} = \{A, B, C\}$$

چک سطری تجزیه مطلوب

اینکه بعد از عمل تجزیه، **سطری کم نشود** و **سطری اضافه نشود** باید چک شود که **صفت مشترک** یعنی $R1 \cap R2$، **حداقل** در یکی از دو جدول تجزیه شده R1 و R2، **کلید کاندید** است یا خیر؟، این کار توسط عملگر اشتراک $\cap$ و به صورت زیر انجام می‌شود:

$$R1(H) \cap R2(H) = \text{C.K. of } R1(H) \quad \textbf{OR} \quad R1(H) \cap R2(H) = \text{C.K. of } R2(H)$$

Ri(H) : یعنی صفات رابطه Ri

مثال: رابطه $R(A, B, C)$ با مجموعه وابستگی‌های تابعی (FD) زیر مفروض است:

$$F = \{A \rightarrow C\}$$

پس از تجزیه به دو رابطه $R1(\mathbf{A},B)$ و $R2(\mathbf{A},C)$ ، **ستون مشترک A** بین دو جدول در کدام، **کلید کاندید** است؟

چک جدول R1

$$1: R1(H) \cap R2(H) \to R1(H)$$
$$\{A,B\} \cap \{A,C\} \nrightarrow \{A,B\}$$
$$\{A\} \nrightarrow \{A,B\}$$

$$2: R1(H) \cap R2(H) \to R1(H) - R2(H)$$
$$\{A,B\} \cap \{A,C\} \nrightarrow \{A,B\} - \{A,C\}$$
$$\{A\} \nrightarrow \{B\}$$

$$\{A\}^+ = \{A, Stop\}$$

توضیح: مطابق وابستگی‌های تابعی، **صفت مشترک A** کلید کاندید جدول **R1** نیست، چون همه ستون‌های جدول R1 را تولید نمی‌کند، صفات AB کلید کاندید جدول R1 است.

چک جدول R2

$$1: R1(H) \cap R2(H) \to R2(H)$$
$$\{A,B\} \cap \{A,C\} \to \{A,C\}$$
$$\{A\} \to \{A,C\}$$

$$2: R1(H) \cap R2(H) \to R2(H) - R1(H)$$
$$\{A,B\} \cap \{A,C\} \to \{A,C\} - \{A,B\}$$
$$\{A\} \to \{C\}$$

$$\{A\}^+ = \{A\}$$
$$A \to C$$
$$\{A\}^+ = \{A,C\}$$

توضیح: مطابق وابستگی‌های تابعی، **صفت مشترک A** کلید کاندید جدول **R2** است، چون همه ستون‌های جدول R2 را تولید می‌کند.

توضیح: عبارت 1 و 2 در **متون کتب مرجع مختلف**، یک معنی و نتیجه یکسان دارند، در عبارت 1 **وابستگی‌های بدیهی** هم تولید می‌شود، درحالی‌که در عبارت 2 به واسطه تفاضل، وابستگی‌های بدیهی حذف می‌شود. $R1(H) \cap R2(H)$ یعنی **صفت مشترک**، عبارت $R1(H)$ یعنی همه ستون‌های R1، عبارت $R2(H)$ یعنی همه ستون‌های R2، عبارت $R1(H)-R2(H)$ یعنی همه ستون‌های R1 منهای صفات مشترک و عبارت $R2(H)-R1(H)$ یعنی همه ستون‌های R2 منهای صفات مشترک، هردو عبارت 1 و 2 بررسی می‌کند که **صفت مشترک** در جدول مورد بررسی **کلید کاندید** است یا خیر؟

مثال: رابطه $R(A, B, C, D, E)$ با مجموعه وابستگی‌های تابعی (FD) زیر مفروض است:

$F = \{(C, D) \rightarrow E\}$

پس از تجزیه به دو رابطه $R1(\mathbf{A}, B, \mathbf{C}, \mathbf{D})$ و $R2(\mathbf{A}, \mathbf{C}, \mathbf{D}, E)$، صفات مشترک **ACD** بین دو جدول در کدام، **کلید کاندید** است؟

چک جدول R1

$1: R1(H) \cap R2(H) \rightarrow R1(H)$

$\{A, B, C, D\} \cap \{A, C, D, E\} \nrightarrow \{A, B, C, D\}$

$\{A, C, D\} \nrightarrow \{A, B, C, D\}$

$2: R1(H) \cap R2(H) \rightarrow R1(H) - R2(H)$

$\{A, B, C, D\} \cap \{A, C, D, E\} \nrightarrow \{A, B, C, D\} - \{A, C, D, E\}$

$\{A, C, D\} \nrightarrow \{B\}$

$\{A, C, D\}^+ = \{A, C, D, Stop\}$

توضیح: مطابق وابستگی‌های تابعی، **صفات مشترک ACD** کلید کاندید جدول **R1** نیست، چون همه ستون‌های جدول R1 را تولید نمی‌کند.

چک جدول R2

$$1: R1(H) \cap R2(H) \rightarrow R2(H)$$
$$\{A, B, C, D\} \cap \{A, C, D, E\} \rightarrow \{A, C, D, E\}$$
$$\{A, C, D\} \rightarrow \{A, C, D, E\}$$

$$2: R1(H) \cap R2(H) \rightarrow R2(H) - R1(H)$$
$$\{A, B, C, D\} \cap \{A, C, D, E\} \rightarrow \{A, C, D, E\} - \{A, B, C, D\}$$
$$\{A, C, D\} \rightarrow \{E\}$$

$$\{A, C, D\}^+ = \{A, C, D\}$$
$$CD \rightarrow E$$
$$\{A, C, D\}^+ = \{A, C, D, E\}$$

توضیح: مطابق وابستگی‌های تابعی، **صفات مشترک ACD** کلید کاندید جدول **R2** است، چون همه ستون‌های جدول R2 را تولید می‌کند.

نتیجه: در جدول $R2(\mathbf{A}, \mathbf{C}, \mathbf{D}, E)$ صفات **ACD** به طور **بدیهی**، خودش خودش را می‌دهد و E را هم می‌دهد، بنابراین صفات **ACD** کلید کاندید جدول **R2** است.

نکته بسیار مهم: اگر بین دو جدول تجزیه‌شده فقط و فقط **صفات مشترک** باشد و صفات مشترک **حداقل** در یکی از دو جدول **کلید کاندید** نباشد، این حالت فقط تضمین می‌کند که نسبت به جدول اولیه **سطری کم نشود**، اما تضمین نمی‌کند که **سطری اضافه نشود**، بنابراین برای اینکه به طور همزمان **تضمین** شود که **سطری کم نشود** و **سطری اضافه نشود** باید **صفات مشترک** در **حداقل** یکی از دو جدول **کلید کاندید** باشد.

مثال: برای‌نمونه یک **تجزیه نامطلوب با ستون مشترک C** به صورت زیر است:

A	C
1	201
3	201
4	204

R1

⋈

B	C
101	201
102	201
103	204

R1

=

A	B	C
1	101	201
1	102	201
3	**101**	**201**
3	102	201
4	103	204

R

واضح است که R = R1 JOIN R2 برقرار نیست (سطر سوم **اضافه** است) و جدول اولیه **دقیقا** به‌دست نمی‌آید، یعنی الحاق (پیوند) بازسازنده (بازگشت‌پذیر) نیست. چون **ستون مشترک C** در **حداقل** یکی از دو جدول **کلید کاندید** نیست.

۲- شرط کافی: (تجزیه، **حافظِ تمام FDهای جدول اولیه** باشد.)

FDs preserving DE composition باشد، یعنی **تجزیه حافظ وابستگی‌های تابعی** باشد.

اگر تجزیه به‌گونه‌ای باشد که توسط وابستگی‌های تابعی دو جدول تجزیه شده R1 و R2 یعنی F' ، بتوانیم **همه** وابستگی‌های تابعی جدول اولیه یعنی F را استنتاج کنیم $F^+ = F'^+$ ، آنگاه شرط کافی محقق شده‌است. در **تجزیه مطلوب**، وابستگی‌های تابعی موجود در **جدول اولیه** نباید از دست بروند. به عبارت دیگر تمام FDهای موجود در جدول R یا باید در مجموعه FDهای R1 و R2 وجود داشته باشد، و یا از مجموعه FDهای موجود در R1 و R2 **مجدد** قابل استنتاج باشد.

$$F'^+ \equiv \left\{F_{\Pi_{<H1>}(R)} \cup F_{\Pi_{<H2>}(R)}\right\}^+$$

$$F^+ = F'^+$$

توجه مهم: در برخی متون کتب مرجع کلاسیک اصطلاح **بی‌حشو** مطرح نشده است و به همان **بی‌کاست (Non-Loss (LossLess))** بودن اکتفا شده است، در این مراجع، بی‌کاست بودن شامل بی‌حشو بودن هم هست، در واقع بی‌کاست و بی‌حشو هردو باهم در اصطلاح بی‌کاست خلاصه شده است. البته در برخی منابع مرجع دیگر، **بی‌کاست (Non-Loss (LossLess))** بودن به معنی **بی‌کاست**، **بی‌حشو** و **حافظ وابستگی‌های تابعی** هرسه باهم است. همین موضوع باعث شده‌است سوالات ناواضح طراحان کنکور در این مبحث در **کلید نهایی** سازمان سنجش، **حذف** شود.

نکته مهم: بین شرط **لازم** و **کافی** تجزیه مطلوب، شرط لازم از اهمیت و اولویت بالاتری برخوردار است، **چرا؟** چون شرط لازم، نیاز اولیه است و شرط کافی، نیاز ثانویه. به عبارت دیگر در **طراحی بانک اطلاعات** رعایت شرط لازم (Loss-Less Join) اولویت بالاتر دارد نسبت به شرط کافی (حفظ وابستگی‌ها تابعی). بعد از اینکه **طراحی بانک اطلاعات** تمام شد، حال نوبت به رعایت قوانین جامعیت داخلی (مثل جامعیت درون رابطه‌ای (داشتن حداقل یک کلید کاندید (اصلی))) و خارجی (قوانین کسب و کار) برای **جدول‌ها** می‌رسد. این آسیاب به نوبت است.

نرمال فرم دوم (Second Normal Form-2NF)

سطح **دوم** نرمال‌سازی، تبدیل جداول به **نرمال فرم دوم** است.

شروط قرار داشتن یک جدول در نرمال فرم دوم به صورت زیر است:

شرط لازم: جدول باید در **نرمال فرم اول** باشد.

شرط کافی: جدول باید فاقد **وابستگی تابعی بخشی (غیرکامل)** باشد.

وابستگی تابعی بخشی (PFD: Partial Functional Dependency) : وابستگی یک مولفه غیرکلیدی، به بخشی (جزئی یا عضوی) از کلید کاندید را **وابستگی تابعی بخشی** می‌نامند.

وابستگی تابعی بخشی $X \to Y$ ⇒ غیرکلید **(وابستگی بد)** عضو کلید کاندید

مولفه غیرکلیدی: هر صفتی که عضو هیچ کلید کاندیدی نباشد **مولفه غیرکلیدی** است.

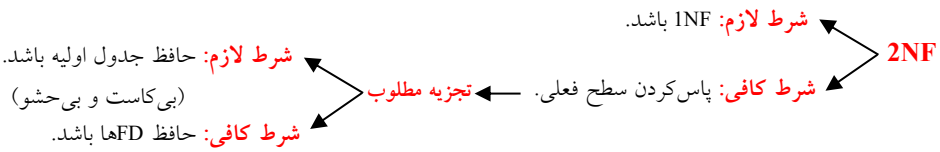
مولفه کلیدی (صفت عمده یا جزء یا عضو کلید کاندید): هر صفتی که عضو حداقل یک کلید کاندید باشد، مولفه کلیدی است. صفتی که بخشی از کلید کاندید است تحت هیچ شرایطی به آن، صفت غیرکلیدی گفته نمی‌شود، بلکه به آن همان **مولفه کلیدی** گفته می‌شود.

نکته مهم: اگر جدولی در **نرمال فرم اول** باشد و در آن **تمام** کلیدهای کاندید، **تک خصیصه‌ای** باشند، آن جدول به طور خودکار **فاقد وابستگی بخشی** است و بنابراین در **نرمال فرم دوم** قرار دارد. دقت کنید که شرط ایجاد وابستگی بخشی وجود کلید کاندید ترکیبی از چند خصیصه است، بنابراین کلید کاندید تک خصیصه‌ای شرط ایجاد وابستگی بخشی را ندارد.

حذف وابستگی تابعی بخشی با تئوری هیث (Heath)

جهت حذف وابستگی تابعی بخشی و به تبع قرارگیری جدول در نرمال فرم دوم، **تئوری هیث** مورد استفاده قرار می‌گیرد. در این تئوری، با **تجزیه جدول** به روش CUT کردن و برداشتن دُم (دترمینان) وابستگی تابعی بخشی (بعلاوه کل ملحقات چسبیده به وابستگی بخشی) از روی کلید کاندید و قرار دادن آن در یک **جدول مستقل** جدید، وابستگی تابعی بخشی، از جدول R **حذف** می‌شود و مشکل **درمان** می‌شود. با حذف وابستگی بخشی، از روی جدول اولیه R، **افزونگی محتوایی** ناشی از آن هم از بین می‌رود.

توجه مهم: تئوری هیث، **به‌طور خودکار** شروط **تجزیه مطلوب** (شرط لازم و کافی) را برقرار می‌کند، یعنی هم **حافظ تمام سطرها و ستون‌های جدول اولیه** و هم **حافظ تمام FDهای جدول اولیه** است.



فرآیند نرمال‌سازی از **سه قدم** زیر تشکیل شده است:

۱- کشف کلید کاندید

۲- کشف وابستگی‌های بد و خوب

۳- حذف وابستگی‌های بد

مثال: رابطه $R(A,B,C)$ با مجموعه وابستگی‌های تابعی (FD) زیر مفروض است:

$$F=\{A \rightarrow C\}$$

وابستگی تابعی بخشی

عضو کلید کاندید | عضو کلید کاندید | غیرکلید

A	B	C
1	101	**201**
1	102	**201**
3	102	201
4	103	204

R

حاصل تجزیه 2NF جدول R به چه صورت است؟

۱- کشف کلید کاندید

با توجه به وابستگی‌های مطرح شده داریم:

$$F=\{A \rightarrow C\}$$

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیربدیهی- تمام خصیصه‌های جدول= عضو کلید کاندید

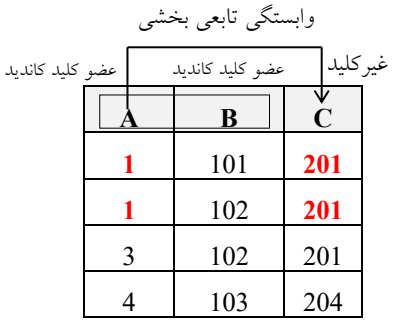
$$AB\not\subset -C=AB$$

بنابر رابطه فوق صفات AB حتماً باید **عضو کلید کاندید** باشد. بستار صفات AB به صورت زیر است:

$$\{AB\}^{+}=\{AB\} \Rightarrow$$
$$A \rightarrow C \Rightarrow \{A,B,C\}$$
$$\{AB\}^{+}=\{A,B,C\}$$

براساس بستار فوق، صفات AB، همه ستون‌های جدول R را تولید می‌کند، پس مطابق **قانون دوم** صفات AB **تنها کلید کاندید** جدول R است. و تحت هیچ شرایطی کلید کاندید دیگری ندارد.

۲ - کشف وابستگی‌های بد و خوب



A	B	C
1	101	201
1	102	201
3	102	201
4	103	204

R

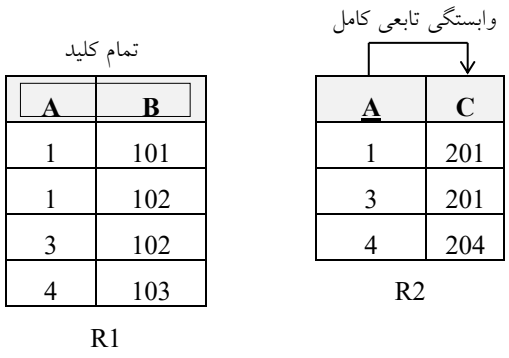
وابستگی تابعی بخشی (غیرکامل): **وابستگی بد** $\Rightarrow$ غیرکلید $A \rightarrow C$ عضو کلید کاندید

وابستگی تابعی کامل: **وابستگی خوب** $\Rightarrow$ غیرکلید $AB \rightarrow C$ کلید کاندید

توجه: وابستگی تابعی بد به طور ذاتی **مجوز** تکرار در سمت چپ دارد، اما وابستگی تابعی خوب به طور ذاتی **مجوز** تکرار در سمت چپ ندارد.

۳ - حذف وابستگی‌های بد (توسط تجزیه (سنتز))

با **تجزیه جدول** به روش CUT کردن، وابستگی بخشی $A \rightarrow C$ از جدول R، **حذف** می‌شود و مشکل **درمان** می‌شود. به صورت زیر:



A	B
1	101
1	102
3	102
4	103

R1

A	C
1	201
3	201
4	204

R2

توجه: وابستگی بخشی (بد) $A \rightarrow C$ از جدول R **حذف** شد و به وابستگی کامل (خوب) $A \rightarrow C$ در جدول R2 تبدیل شد.

توجه: تجزیه در نرمال‌سازی یعنی **دوری و دوستی**، دور هستن یعنی دو جدول شدن، اما دوست

هم هستن، یعنی توسط تعریف **کلید خارجی**، الحاق‌پذیر هم هستن و توسط عمل الحاق طبیعی همان **جدول اولیه** ایجاد می‌گردد.

در فرآیند نرمال‌سازی شروط **تجزیه مطلوب** به صورت زیر است:

۱- شرط لازم: (تجزیه، **حافظ تمام ستون‌ها و سطرهای جدول اولیه** باشد.)

در مثال قبل، دو جدول R1 و R2 الحاق‌پذیر هستند، همچنین **صفت مشترک A** در جدول R2 **کلید کاندید** و در جدول R1 **کلید خارجی** است، بنابراین شرط لازم **بی‌کاست** و **بی‌حشو** به طور هم‌زمان محقق شده‌است. خروجی $R_1 \bowtie R_2$ به صورت زیر است:

کلید خارجی — کلید کاندید

A	B
1	101
1	102
3	102
4	103

R1

⋈

A	C
1	201
3	201
4	204

R2

=

A	B	C
1	101	**201**
1	102	**201**
3	102	201
4	103	204

R

واضح است که R = R1 JOIN R2 برقرار است و جدول اولیه **دقیقا** به‌دست می‌آید، یعنی الحاق (پیوند) بازسازنده (بازگشت‌پذیر) است.

نکته بسیار مهم: تجزیه جدول به روش **CUT کردن** باعث می‌شود، تجزیه به طور خودکار **بی‌کاست** و **بی‌حشو** شود.

ذهنیت جلسه کنکور: چیز بد (وابستگی بخشی) رو دیدی چیکارش باید کرد؟ هیچی دُمش رو بگیری بندازیش بیرون.

۲- شرط کافی: (تجزیه، **حافظ تمام FDهای جدول اولیه** باشد.)

همه وابستگی‌های موجود در جدول اولیه R توسط جداول تجزیه شده R1 و R2 قابل دستیابی است، بنابراین شرط کافی محقق شده‌است.

$$F'^{+} \equiv \left\{F_{\Pi_{<H1>}(R)} \cup F_{\Pi_{<H2>}(R)}\right\}^{+}$$

$$F^{+} = F'^{+}$$

$$F'^{+} = \{A \rightarrow C\}$$

وابستگی $A \rightarrow C$ در جدول R موجود است، از جدول تجزیه شده R2 هم در دسترس است.

نکته بسیار مهم: تجزیه جدول به روش **CUT کردن** باعث می‌شود، تجزیه به طور خودکار **حافظ FD** باشد.

شروط قرار داشتن یک جدول در نرمال فرم دوم به صورت زیر است:

شرط لازم: جدول باید در **نرمال فرم اول** باشد. دو جدول R1 و R2 در نرمال فرم اول قرار دارند.

شرط کافی: جدول باید فاقد **وابستگی تابعی بخشی (غیرکامل)** باشد. دو جدول R1 و R2 فاقد وابستگی بخشی هستند.

نتیجه: دو جدول R1 و R2 در **نرمال فرم دوم** قرار دارند.

نتیجه: مطابق **تئوری هیث**، اگر R(A , B , C) یک رابطه باشد و وابستگی تابعی $A \rightarrow C$ موجود باشد، آنگاه تجزیه R به دو رابطه R1 و R2 **تجزیه مطلوب** است، به صورت زیر:

$A \rightarrow C \quad : R1(A,C)$

$A \rightarrow B \quad : R1(A,B)$

تجزیه نامطلوب

یک جدول می‌تواند **چندین تجزیه** داشته باشد، بدین صورت که **ترکیبی** از ستون‌ها به طرق مختلف **ستون مشترک** باشد، که در بین آنها یکی **تجزیه مطلوب** و مابقی **تجزیه نامطلوب** خواهد بود، به صورت زیر:

R1	R2	
B A	A C	تجزیه مطلوب با ستون مشترک A
A B	B C	تجزیه نامطلوب با ستون مشترک B
A C	C B	تجزیه نامطلوب با ستون مشترک C

یا اینکه در تجزیه، اصلا ستون مشترک نباشد که آن هم **تجزیه نامطلوب** است، به صورت زیر:

R1	R2	
A B	C	تجزیه نامطلوب بدون ستون مشترک
A C	B	تجزیه نامطلوب بدون ستون مشترک
B C	A	تجزیه نامطلوب بدون ستون مشترک

برای‌نمونه یک **تجزیه نامطلوب با ستون مشترک C** به صورت زیر است:

A	C
1	201
3	201
4	204

R1

⋈

B	C
101	201
102	201
103	204

R1

=

A	B	C
1	101	201
1	102	201
3	**101**	**201**
3	102	201
4	103	204

R

واضح است که R = R1 JOIN R2 برقرار نیست (سطر سوم **اضافه** است) و جدول اولیه **دقیقا** به‌دست نمی‌آید، یعنی الحاق (پیوند) بازسازنده (بازگشت‌پذیر) نیست. چون ستون مشترک C در **حداقل** یکی از دو جدول **کلید کاندید** نیست.

نرمال فرم سوم (Third Normal Form-3NF)

سطح **سوم** نرمال‌سازی، تبدیل جداول به **نرمال فرم سوم** است.

شروط قرار داشتن یک جدول در نرمال فرم سوم به صورت زیر است:

شرط لازم: جدول باید در **نرمال فرم دوم** باشد.

شرط کافی: جدول باید فاقد **وابستگی تابعی انتقالی (تراگذری یا باواسطه)** باشد.

وابستگی تابعی انتقالی (TFD: Transitive Functional Dependency) : وابستگی یک مولفه غیرکلیدی، به یک مولفه غیرکلیدی دیگر را **وابستگی تابعی انتقالی** می‌نامند.

وابستگی تابعی انتقالی $X \rightarrow Y$ غیرکلید $\Rightarrow$ غیرکلید **(وابستگی بد)**

مولفه غیرکلیدی: هر صفتی که عضو هیچ کلید کاندیدی نباشد **مولفه غیرکلیدی** است.

حذف وابستگی تابعی انتقالی با تئوری ریسانن (Rissanen)

جهت حذف وابستگی تابعی انتقالی و به تبع قرارگیری جدول در نرمال فرم سوم، **تئوری ریسانن** مورد استفاده قرار می‌گیرد. در این تئوری، با **تجزیه جدول** به روش CUT کردن و برداشتن دُم (دترمینان) وابستگی تابعی انتقالی (بعلاوه ملحقات چسبیده به وابستگی انتقالی) از روی جدول پایه R و قرار دادن آن در یک جدول مستقل جدید، وابستگی تابعی انتقالی از جدول R **حذف** می‌شود و مشکل **درمان** می‌شود. با حذف وابستگی انتقالی از روی جدول پایه R، افزونگی‌های ناشی از آن هم از بین می‌رود.

توجه مهم: تئوری ریسانن، **به‌طور خودکار** شروط **تجزیه مطلوب** (شرط لازم و کافی) را برقرار

می‌کند، یعنی هم حافظ جدول پایه و هم حافظ FD ها است.

توجه مهم: جداولی که توسط تئوری ریسانن، تجزیه می‌شوند، به جداول مستقل از هم موسوم هستند.

نکته مهم: دقت کنید، مدل ریاضی تجزیه، یک **درخت تجزیه دودویی** است، یعنی در هر مرحله، تجزیه دودویی است. برای مثال اگر یک جدول شامل چند وابستگی تابعی انتقالی باشد، در هر مرحله از درخت تجزیه، یکی از وابستگی‌های تابعی انتقالی حذف می‌شود و همه باهم در یک مرحله حذف نمی‌شود.

فرآیند نرمال‌سازی از **سه قدم** زیر تشکیل شده است:

۱- کشف کلید کاندید

۲- کشف وابستگی‌های بد و خوب

۳- حذف وابستگی‌های بد

جهت دانلود ادامه جزوه به لینک زیر در سایت موسسه بابان مراجعه نمایید:

BabanAcademy.com/database

مثال: رابطه $R(A,B,C)$ با مجموعه وابستگی‌های تابعی (FD) زیر مفروض است:

$F=\{A\rightarrow B\ ,\ B\rightarrow C\}$

وابستگی کامل وابستگی انتقالی

A	B	C
1	101	202
2	102	203
3	102	203
4	103	202

R