

وابستگی و نرمال سازی



5/1- مقدمه را با یک سوال آغاز می کنیم : اگر همه اطلاعات مربوط به درس (شامل درس ها و گروه های آنها) را در یک جدول بیاوریم چه می شود ؟ به عبارت دیگر اگر ستون های جداول مربوط به درس ها را مانند شکل 5/1 کنار هم بچینیم چه اتفاقی می افتد ؟ (البته ستون #C نباید تکرار شود)

C#	Clg#	Unit	Cname	Pname	Term	S#	Sec#	Score
10172	10	3	شبیه سازی	قربانی	761	7113384 8	1724	14.5
10172	10	3	شبیه سازی	قربانی	761	7220330 5	1724	16.25
10172	10	3	شبیه سازی	قربانی	761	7213182 5	1747	15.75
10172	10	3	شبیه سازی	قربانی	761	7213050 2	1747	12.5
10172	10	3	شبیه سازی	قربانی	761	7418253 2	1747	17

شکل 5/1- جدول فرضی شامل اطلاعات مربوط به جداول CRS و SEC

با بررسی شکل 5/1 سه اشکال عمده ملاحظه می شود :

1- افزونگی داده ها (*data redundancy*)

قبلا گفته شد که افزونگی یعنی تکرار بی رویه داده ها . در بانک اطلاعات رابطه ای ، تکرار داده ها تنها راه برقراری ارتباط بین جداول است و از آن به عنوان کلید خارجی یاد می شود . تکرار بیش از این ، بی رویه است و افزونگی نام دارد . در برقراری ارتباط بین جداول CRS و SEC تکرار کلیدی مانند C# در هر جدول ضرورت دارد اما بیش از آن بی مورد است .

جدول فرضی بالا تکرار بی رویه را نمایش می دهد . به ازای همه گروه های درسی و همه دانشجویان تکرار کرده ایم که نام درس چیست ، تعداد واحد آن چند است و غیره . باید توجه نمود که این تکرار بی رویه در جدول بزرگ SEC اتفاق افتاده نه در جدول کوچک

CFS (این تکرار در شکل مشخص شده است .) وقتی جداول را ادغام می کنیم همواره افزونگی به میزان بزرگترین جدول رخ می دهد . این رخداد ، دو زیان بزرگ به همراه دارد . اول هدر دادن فضا که آشکار است . دومین زیان بزرگ که به ظاهر دیده نمی شود و پایین آمدن سرعت است . خیلی طبیعی است که وقتی حجم داده ها چند برابر می شود زمان کار کردن با آنها هم افزایش می یابد .

آیا ادغام جداول و بروز پدیده افزونگی امتیازی هم به همراه دارد ؟ بله ، البته ؟ یک لحظه تصور کنید که هر بانک اطلاعات فقط یک جدول بود . آنوقت کاربرها نفس راحتی می کشیدند . دیگر نه نیازی به پیوند طبیعی بود ، نه ضرب دکارتی ، نه پرتو ، و نه ترکیب جداول به هر شکل . به عبارت دیگر استخراج اطلاعات از داده ها ساده می شد ، یعنی پیچیدگی های نوشتن پرس و جو ها از میان می رفت . خوانندگان گرامی حتما تا بحال متوجه شده اند که “ بانک اطلاعات دانشگاه ” هم که فصل های گذشته مبنای کار ما قرار گرفت به میزان اندکی دارای افزونگی ! این تصمیم برای ساده تر شدن بحث ها و مثال های پرس و جو (جبر رابطه ای ، حساب رابطه ای و SQL) اتخاذ گردیده است . این پدیده در بین کتابهای بانک اطلاعات هم جای خود را باز کرده و به “ نرمال سازی ”

(*denormalization*) موسوم است . منظور این است که نرمال سازی را هم نباید مطلق کرد و از افتادن در چاه نرمال سازی بی رویه هم باید پرهیز نمود .

2. بی نظمی (*anomaly*)

وجود افزونگی در شکل 5/1 باعث بی نظمی در تغییر داده ها می شود . مثلثال فرض کنید در واحد آموزش دانشکده ای بخواهند نمرات دروسی را وارد کامپیوتر کنند . با وارد کردن نمره هر دانشجو باید مشخصات کامل درس مربوطه هم وارد شود ، در صورتیکه درس ممکن است چند صد دانشجو داشته باشد . این در حالی است که ما فقط دو جدول را ادغام کرده ایم . اگر جداول دیگر را هم با آنها ادغام کنیم آنگاه با وارد کردن نمره هر دانشجو باید مشخصات کامل درس و استاد و دانشکده هم وارد شود . همین اتفاق در حذف کردن و به روز در آوردن داده ها هم می افتد و کار کنترل مخصوصا کنترل جامعیت ارتباطی به معنای عام ، بسیار سخت می شود و مرتبا برنامه ها پرس و جو ها که در تار عنکبوتی طراحی غلط دست و پا می زنند ، مرتکب خطا می شوند و ساقط می گردند و مجددا شروع می کنند و در هم می پیچند . این اتفاق به این دلیل می افتد که داده ها به هم وابستگی دارند یعنی مثلا هر جای سیستم که درس DB1 دیده شود باید نام بانک اطلاعات و اساتید مربوطه و دانشکده ارائه دهنده و ... به طور کاملا دقیق وارد شود . نباید فراموش کرد که با ادغام جداول ، حجم داده ها چند برابر می شود و همه این وارد کردن ها و کنترل ها هم به همین نسبت افزایش می یابد .

3- مقادیر تهی (*NULL values*)

با ادغام جداول ، از نشان دادن بعضی از اقلام اطلاعاتی بدون استفاده از مقادیر تهی نا توانیم . مثلثال دانشکده معمولا همه درس های خود را در همه ترم ها ارائه نمی دهند . دروسی که گروهی برای آنها اعلام نشده یا باید در جدول نیابند (که خلاف واقعیت است

(و یا برای گروه و دانشجو و ترم و سایر صفات آنها مقادیر تهی در نظر گرفته شود .
مقادیر تهی علاوه بر اینکه جای زیادی را اشغال می کنند مشکلات دیگری را نیز باعث
می گردند که در فصل قبل به بعضی از آنها پرداختیم .

5/2- وابستگی تابعی (*functional dependency*)

افزونگی و وابستگی ارتباط تنگاتنگ دارند . تئوری وابستگی از مفاهیم عمده بانک
اطلاعات است تا جائیکه بعضی از نویسندگان ، بانک اطلاعات را به صورت “ داده ها و
وابستگی ” تعریف کرده اند . وابستگی انواع مختلف دارد ، مانند وابستگی تابعی
(*functional dependency*) ، وابستگی پیوندی (*join*) و وابستگی چند
مقداری (*multi – value dependency*) . در این کتاب به وابستگی تابعی توجه
ویژه ای شده است و انواع دیگر وابستگی که کمیابند به اختصار بیان شده اند .
مفاهیم و تعاریف زیر از همان آغاز پیدایش مدل رابطه ای توسط آقای کاد ارائه شده اند .

▪ تعریف :

اگر A و B دو مجموعه صفت در شمای R باشند آنگاه وابستگی تابعی $A \rightarrow B$
برقرار است اگر برای تمام رابطه ها در R ، به ازای هر مقدار A فقط یک مقدار B
داشته باشیم.

در این تعریف نکات زیر قابل توجه است :

1. وابستگی تابعی باید برای تمام رابطه ها درست باشد ، یعنی از معنی و ذات و آن صفتها سرچشمه بگیرد نه از موارد خاص در یک یا چند رابطه . برای روشن شدن مطلب به جدول زیر توجه فرمایید :

کلاس	استاد	درس	ترم
301	فاضل	بانک اطلاعات	771
302	تبریزی	ریاضی گسسته	771
303	قربانی	هوش مصنوعی	771
304	میر شمسی	مدار منطقی	772

در این جدول ، با این داده ها ، وابستگی های زیادی دیده می شود که در واقع صحیح نیست ، مثل استاد → کلاس و کلاس → درس و استاد → درس و درس → استاد .

2. وابستگی تابعی برای تعریف محدودیت های بانک اطلاعات نیز به کار می رود . یک وابستگی ممکن است در یک بانک درست باشد و در بانک مشابه دیگر غلط . مثلا در جدول بالا ، اگر وابستگی استاد → (درس ، ترم) برقرار باشد ، معنای آن این است که در هر ترم دو استاد نمی توانند مدرس یک درس باشند (همه گروه های آن درس توسط یک استاد ارائه می شود) . پس طراح می تواند قواعد بانک اطلاعات خود را با وابستگی تابعی نیز بیان کند .

▪ تعریف :

$A \rightarrow B$ را وابستگی تابعی کامل (*full functional dependency* یا *FFD*) می گوئیم اگر B به هیچ زیر مجموعه محض (*proper subset*) از A وابسته نباشد .

▪ تعریف :

اگر برای تمام صفت‌های B در R داشته باشیم $A \rightarrow B$ آنگاه A را ابر کلید R می نامند و به صورت $A \rightarrow R$ نمایش می دهند .
اگر این وابستگی از نوع *FFD* باشد آنگاه A کلید کاندید R است .

▪ تعریف :

اگر B زیر مجموعه ای از A باشد آنگاه همواره $A \rightarrow B$. این وابستگی تابعی را بدیهی (*trivial FD*) می نامیم .

▪ مثال :

در جدول *SEC* وابستگی زیر مشاهده می شود :

$$(\text{sec\#,c\#,s\#,term}) \rightarrow \text{pname}$$

این وابستگی از نوع *FFD* نیست زیرا داریم :

$$(\text{sec\#,c\#,term}) \rightarrow \text{pname}$$

وابستگی اخیر از نوع FFD است زیرا :

در دوترم مختلف ، گروه i از درس X بادو استاد متفاوت قابل ارائه است و نیز در یک دانشاه بزرگ معمولا اساتید مختلف برای گروه های مختلف یک درس تعیین می شود ، پس هیچکدام از زیر مجموعه های $(sec\#,c\#,term)$ نام استاد را نمی دهد .

یعنی : $(sec\#,c\#) \rightarrow pname$

$(sec\#,term) \rightarrow pname$

$(c\#,term) \rightarrow pname$

با توجه ه اینکه $(s\#,score) \rightarrow (sec\#,c\#,term)$ پس این مجموعه صفت ، کلید sec نیست . به عبارت دیگر مجموعه صفت $(sec\#,c\#,term)$ کلید نسیت چون دانشجو ونمره او را مشخص نمی کند . توجه به معنی صفتها ، خاصیت کلیدی آنها را مشخص می کند . مجموعه صفت $(s\#,c\#,term)$ کلید کاندید است زیرا یک دانشجو در یک ترم نمی تواند در دو گروه از یک درس ثبت نام کند .

در یک بانک اطلاعات باید از همه وابستگی های آن مطلع باشیم . تعدادی از وابستگی های بین صفت ها قابل فهم و شناسائی هستند ولی باید سایر وابستگی ها را هم به دست آورد . هر گاه تعدادی وابستگی داشته باشیم ممکن است بتونیم وابستگی های دیگری نیز از آنها به دست آوریم و یا بعضی از آنها را که حرف تازه ای نمی زنند حذف کنیم .

مثال : فرض کنید داریم :

$$R = (S, T, U, V, W)$$

$$F = \{S \rightarrow T, V \rightarrow SW, T \rightarrow U\}$$

آنگاه منطقاً می توان وابستگی های تابعی زیر را نیز به دست آورد و نتیجه گرفت که V کلید کاندید R است :

$$S \rightarrow U \quad (S \rightarrow T, T \rightarrow U \quad)$$

$$V \rightarrow S$$

$$V \rightarrow W$$

$$V \rightarrow T \quad (V \rightarrow S, S \rightarrow T \quad \text{زیرا} \quad)$$

$$V \rightarrow U \quad (V \rightarrow S, S \rightarrow U \quad \text{زیرا} \quad)$$

1-2-5- مجموعه پوششی وابستگی

از یک مجموعه وابستگی ، می توان وابستگی های دیگری استخراج نمود و به آن افزود . در این بخش روشهای نسبتاً ساده ای برای دستیابی به همه وابستگی ها و نیز کلید های کاندید ارائه می شود .

▪ تعریف

اگر F یک مجموعه از وابستگی های تابعی باشد آنگاه مجموعه تمام وابستگی های تابعی که از آن منتج می شود را مجموعه پوششی F می نامیم و با F^+ نمایش می دهیم.

آقای آرمسترانگ در سال 1974 روشی برای استخراج پوششی ارائه داد او ثابت کرد که سه قاعده زیر برای استخراج مجموعه پوششی کافی است ، یعنی با اعمال مکرر این سه قاعده می توان به تمام وابستگی های منتج دست یافت و هیچ وابستگی اضافی نیز تولید نمی شود :

1. **بازتاب (reflexivity)**: اگر B زیر مجموعه A باشد آنگاه $A \rightarrow B$

2. **افزایش (augmentation)**: اگر $A \rightarrow B$ و C صفت باشد آنگاه $AC \rightarrow BC$

3. **انتقال (transitivity)**: اگر $A \rightarrow B$ و $B \rightarrow C$ آنگاه $A \rightarrow C$

هر چند ، قاعده های فوق برای استخراج F^+ کفایت می کرد ولی اعمال آنها مشکل بود . بعد ها دیگران قواعد دیگری را بیان کردند که کار را سهولت بخشید . مهمترین آنها عبارتند از :

4. **اجتماع (union)**: اگر $A \rightarrow B$ و $A \rightarrow BC$ آنگاه $A \rightarrow BC$

5. **تجزیه (decomposition)**: اگر $A \rightarrow BC$ آنگاه $A \rightarrow B$ و $A \rightarrow C$

6. **ترکیب (composition)**: اگر $A \rightarrow B$ و $C \rightarrow D$ آنگاه $AC \rightarrow BD$

مجموعه وابستگی بهینه

با اعمال قواعد فوق وابستگی زیادی بدست می آید که بعضا تکراری و اضافی هستند . در اینجا روشی برای حذف اینگونه وابستگی ها و رسیدن به مجموعه وابستگی بهینه ارائه می دهیم .

▪ تعریف :

دو مجموعه وابستگی تابعی $F1$ و $F2$ را معادل (*equivalent*) می نامیم اگر

$$F1^+ = F2^+ \text{ یعنی آنها برابر باشد ،}$$

با استفاده از قواعد سه گانه زیر می توانیم مجموعه وابستگی را به مجموعه بهینه

معادل آن تبدیل کرد :

1. سمت راست هر وابستگی فقط یک صفت باشد

2. هر صفتی که F^+ را تغییر نیم دهد از سمت چپ حذف شود

3. وابستگی های تکراری و اضافی حذف شود .

بطور خلاصه باید گفت که برای یافتن وابستگی های تابعی در یک بانک اطلاعات می

بایست ابتدا مجموعه پوششی وابستگی ها را تعیین کرد و سپس آنرا بهینه نمود .

• مثال :

در بانک اطلاعات زیر ، مجموعه وابستگی پوششی را بیابید :

$$R = (U, V, W, X, Y, Z)$$

$$F = \{U \rightarrow XY, X \rightarrow Y, XY \rightarrow ZV, U \rightarrow ZV\}$$

حل : ابتدا F^+ را می یابیم و سپس آن را بهینه می کنیم .

$$F^+ = \{U \rightarrow XY, X \rightarrow Y, XY \rightarrow ZV, U \rightarrow ZV\}$$

قاعده 1 روی وابستگی اول :

$$F^+ = \{U \rightarrow X, U \rightarrow Y, X \rightarrow Y, XY \rightarrow ZV, U \rightarrow ZV\}$$

قاعده 2 روی $XY \rightarrow ZV$ با توجه به $X \rightarrow Y$:

$$F^+ = \{U \rightarrow X, U \rightarrow Y, X \rightarrow Y, XY \rightarrow ZV, U \rightarrow ZV\}$$

حال قاعده 1 روی سایر وابستگی ها :

$$F_{opt} = \{U \rightarrow X, U \rightarrow Y, U \rightarrow Z, U \rightarrow V, X \rightarrow Y, X \rightarrow Z, X \rightarrow V\}$$

در دنباله همین فصل ، مثال های دیگری ارائه خواهد شد .

2-5/2- کلید های کاندید

اگر مجموعه ای از صفت ها را ATTR و مجموعه وابستگی تابعی آنها و تعدادی صفت دیگر را F بنامیم آنگاه الگوریتم زیر ، مجموعه تمام صفت های وابسته به ATTR را می دهد (بهتر است ابتدا F را بهینه کنیم) .

$$ATTR^+ = ATTR$$

تکرار کن

برای هر $X \rightarrow Y$ در F

اگر X زیر مجموعه $ATTR^+$ باشد آنگاه

$$ATTR^+ = ATTR^+ \cup Y$$

تا زمانی که $ATTR^+$ دیگر تغییر نکند .

با استفاده از این الگوریتم می توان کلید های کاندید را به دست آورد . بطور کلی باید به نکات زیر توجه کرد .

1. هر کلید کاندید شامل مجموعه ای از صفت هائی است که در سمت چپ پیکانها می آیند .

2. کلید کاندید باید کمینه باشد ، یعنی زیر مجموعه ای از آن خاصیت کلیدی نداشته باشد .

3. بانک اطلاعات ممکن است چند کلید کاندید داشته باشد .

4. کلید کاندید ممکن است در یک یا چند صفت مشترک باشند .

• مثال : اگر

$$R = (S, TU, V, W)$$

$$F = \{S \rightarrow T, V \rightarrow SW, T \rightarrow U\}$$

آنگاه

$$\{S, V\}^+ = \{S, V\} \Rightarrow \{S, V, T, W\} \Rightarrow \{S, V, T, W, U\}$$

می توان این الگوریتم را برای مجموعه صفت های دیگر سمت چپ پیکان مرتبا به کاربرد و وابستگیهای مختلف و نیز کلید ها را به دست آورد .

اگر در مثال فوق V^+ را محاسبه کنیم خواهیم دید که $\{S, V\}$ کاهش پذیر است :

$$V^+ = \{V\} \Rightarrow \{V, S, W\} \Rightarrow \{V, S, W, T\} \Rightarrow \{V, S, W, T, U\}$$

در نتیجه $\{S, V\}$ کلید کاندید نیست ولی V هست .

3-5/2- هر گاه تعداد صفت ها در بانک زیاد و وابستگی آنها گسترده باشد ، می توان با رسم نمودار وابستگی به فهم بهتر آن بانک اطلاعات کمک کرد . در این نمودار ، صفتها در مستطیل قرار می گیرند و پیکانی از آنها به هر یک از صفتهای وابسته به آن رسم می شود . برای مجموعه صفتها و وابستگان آنها نیز مستطیل های دیگر و

پیکانه‌های دیگری رسم می‌گردد. مهمولا پیکانهایی که وابستگی به کلید اصلی را نشان می‌دهند در بالای صفت‌ها و سایر پیکانها زیر آنها کشیده می‌شوند. ابتدا باید مجموعه بهینه وابستگی‌ها را به دست آوریم و آنگاه به رسم نمودار وابستگی مبادرت کنیم.

مثال: در بانک اطلاعات زیر ابتدا همه کلید‌های کاندید را بیابید و سپس نمودار وابستگی را رسم کنید.

$$R = (U, V, W, X, Y, Z, O, P, Q)$$

$$F = \{U \rightarrow VXQ, UVP \rightarrow O, OQ \rightarrow YZ, UP \rightarrow XY\}$$

حل:

ابتدا مجموعه بهینه معادل F را پیدا می‌کنیم (تمرین):

$$F_{Opt} = \{U \rightarrow V, U \rightarrow X, U \rightarrow Q, OQ \rightarrow Y, OQ \rightarrow Z, UP \rightarrow Y, UP \rightarrow O, UP \rightarrow Z\}$$

سپس مجموعه صفت‌های وابسته به تمام مجموعه صفت‌های چپ پیکانها را می‌یابیم:

$$U^+ = \{U\} \Rightarrow \{U, V, X, Q\}$$

$$UP^+ = \{U, P\} \Rightarrow \{U, P, V, X, Q, O, Y\} \Rightarrow \{U, P, V, X, O, Q, Y, Z\}$$

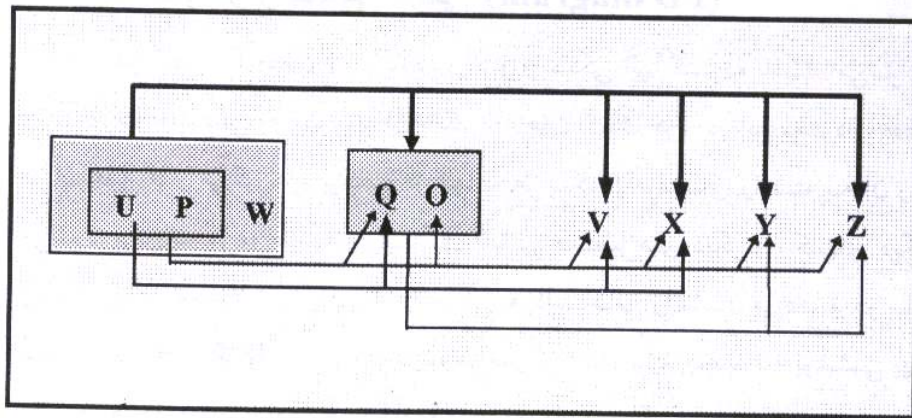
$$OQ^+ = \{O, Q\} \Rightarrow \{O, Q, Y, Z\}$$

نتیجه می‌گیریم که بیشترین صفت‌ها را UP^+ می‌دهد. تنها صفتی که از UP^+ خارج

است W است. پس، UPW کلید کاندیدی می‌باشد، یعنی $UPW \rightarrow R$

کلید کاندید دیگری به دست نمی‌آید، زیرا از O و Q نمی‌توان به U و P رسید (یادآوری می‌شود که کلید کاندید باید کمینه باشد).

نمودار وابستگی این بانک اطلاعات در شکل 5/2 مشاهده می شود .



شکل 5/2- مثالی از نمودار وابستگی تابعی

تمرین : وابستگی $P \rightarrow OQ$ را به مثال فوق اضافه کنید . آیا کلید کاندید دیگری خواهیم داشت ؟

5/3- نرمال سازی (normalization)

همواره این سوال در ذهن طراحان بانک اطلاعات مطرح می شود که آیا آنچه ما ارائه داده ایم بهترین است ؟ در مدل رابطه ای ، روشی کاملاً شناخته شده برای پاسخگوئی به این سوال وجود دارد . این روش را نرمالس ازی می نامند . با استفاده از این متدولوژی می توان جداول خوبی طرای کرد و یا جداول موجود را بهتر نمود . در نرمال سازی قدم های زیر برداشته می شود :

1. از جداول موجود یا لیستهای خروجی یا مشخصات سیستم شروع کن .
2. داده ها و ارتباط ها وابستگی ها را شناسائی کن .
3. ترجیحا نمودار وابستگی را رسم کن .

4. جداول را قدم به قدم به فرم های نرمال تا سطح لازم تبدیل کن .

1-3/5- فرم های نرمال (normal forms)

نرمال سازی در عمل یعنی پیروی از یک سری فرم های نرمال که منجر به تجزیه جداول می شوند . آقای کاد در همان مقاله اولیه خود سه فرم نرمال معرفی کرد که به 1NF و 2NF و 3NF معروفند . چیزی نگذشت که شخص دیگری به نام بویس همراه با ایشان فرم نرمال دیگری به نام BCNE تعریف کرد . این چهار فرم نرمال بر بمنای وابستگی تابعی تعریف شده اند . بعد ها دیگران وابستگی های دیگر و فرمهای نرمال دیگر مانند 4NF و 5NF معرفی کردند که نادر هستند و در موارد ویژه ای کاربرد دارند . در این فصل نرمال سازی تا سطح 5NF بررسی می شود . فرم های نرمال هرکدام مشکلی را که در فرم قبلی وجود دارد حل می کنند . فرم 1NF در واقع ملزومات مدل رابطه ای را بیان می کند . در این بخش فرم های اصلی نرمال را در قالب یک مثال که خصوصیات آن از مشخصات معنایی یک بانک وقاعی استخراج می شود ارائه می دهیم . همچنین چند انتزاعی (بدون توجه به معنای صفت ها) ارائه میگردد .

▪ تعریف : جدولی در 1NF است که :

1. همه کلید های آن تعریف شده باشند .
2. تمام صفت های آن به کلید اصلی وابستگی تابعی داشته باشند .
3. صفت های آن از دامنه تو در تو (*nested domain*) نباشد .

منظور از شرط سوم این است که صفات ترکیبی نداشته باشیم ، یعنی مثلا یا صفتی به نام آدرس بدون توجه به بخشهای آن (شهر و ...) داشته باشیم یا بخشهای آن را بدون توجه به کل .

• مثال :

مشخصات کلی بانک اطلاعات ثبت نام یک دانشگاه در زیر آمده است . این بانک را طراحی و نرمال سازی کنید :

1. کد درس ها غیر تکراری است و نام و تعداد واحد دروس نیز مشخص است .
 2. هر دانشکده درس های مشخصی ارائه میکنند و مسئول دانشکده و شهر آن مشخص است . دروس همنام در دانشکده های مختلف واحد متفاوتی دارند .
 3. هر درس در چند گروه در زمان و مکان مشخص توسط یک استاد ارائه می شود و شماره گروه در دروسها و نیمسال های مختلف تکرار می گردد .
- دانشجویان در گروه های مختلف ثبت نام می کنند و نمره می گیرند .

حل :

الف : جداول مقدماتی

با استفاده از مشخصات کلی این بانک ابتدا جداول مقدماتی آن را با توجه به آنچه تا کنون از طراحی بانکاطلاعات آموخته ایم تعیین می کنیم . همچنین با در نظر گرفتن معنای صفات ، وابستگی های تابعی آنها را تعریف می کنیم . جدول دانشجویان و اساتید تغییر نیم کند و به شناسه آنها اکتفا می نمائیم .

$$crs = \{c\#, cname, unit, clg\#\}$$

$$f1 = \{c\# \rightarrow cname, unit, clg\#, (cname, unit) \rightarrow clg\#\}$$

$$clg = (clg\#, clg\ name, city, pname)$$

$$f2 = \{clg\# \rightarrow clg\ name, city, pname\}$$

$$sec = (c\#, sec\#, s\#, pname, term, score, time, place)$$

$$f3 = \{(s\#, c\#, term) \rightarrow sec\#, score, (sec\#, c\#, term) \rightarrow pname, time, place\}$$

ب: کلید های کاندید

- جداول crs و clg هر کدام دارای یک کلید کاندید هستند که همان کلید اصلی است .

- کلید جدول sec را پیدا می کنیم :

کلید اول :

$$\begin{aligned} (s\#, c\#, term)^+ &= (s\#, c\#, term, sec\#, score) \\ &= (s\#, c\#, term, sec\#, score, pname, time, place) \end{aligned}$$

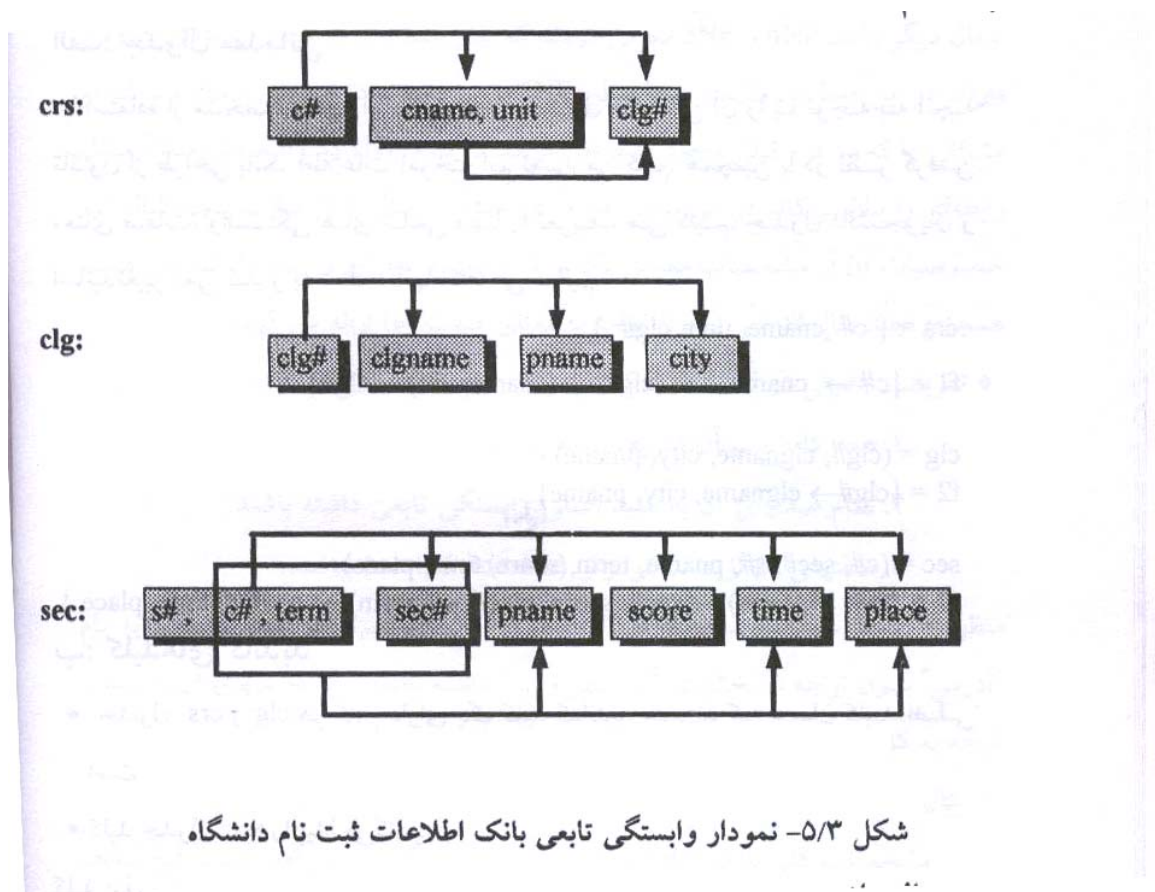
در نتیجه $(s\#, c\#, term)$ کلید کاندید اول جدول sec است .

کلید دوم :

$$(sec\#, c\#, term)^+ = (sec\#, c\#, term, pname, time, place)$$

با توجه به کلید اول ، باید $s\#$ را به این مجموعه افزود که $(s\#, sec\#, c\#, term)$ به دست می آید . این کلید دیگر کمینه نیست و با حذف $sec\#$ همان کلید اول به دست می آید که تنها کلید کاندید و کلید اصلی جدول است .

پ : رسم نمودار وابستگی



شکل 5/3- نمودار وابستگی تابعی بانک اطلاعات ثبت نام دانشگاه

شکل 5/3- نمودار وابستگی تابعی بانک اطلاعات ثبت نام دانشگاه

ت : نرمال سازی

این سه جدول در 1NF هستند زیرا کلید اصلی دارند و همه صفت‌های هر کدام به کلید اصلی وابسته است .

این مثال را با تعریف فرم های دیگر نرمال ادامه می دهیم .

• تعریف : جدولی در 2NF است که :

1. در 1NF باشد

2. صفت‌های آن به زیر مجموعه های کلید اصلی وابستگی نداشته باشند (وابستگی

جزئی یا *partial dependency*)

مشاهده می شود که از سه جدول مربوط به نمودار 5/3 فقط جدول sec در 2NF نیست زیرا سه صفت Place و time و pname به بخشی از کلید اصلی وابسته اند

الگوریتم زیر ، جدول 1NF را به جدول 2NF تجزیه می کند :

1. هر بخش از کلید اصلی را که ایجاد وابستگی جزئی کرده است ، با همه وابسته

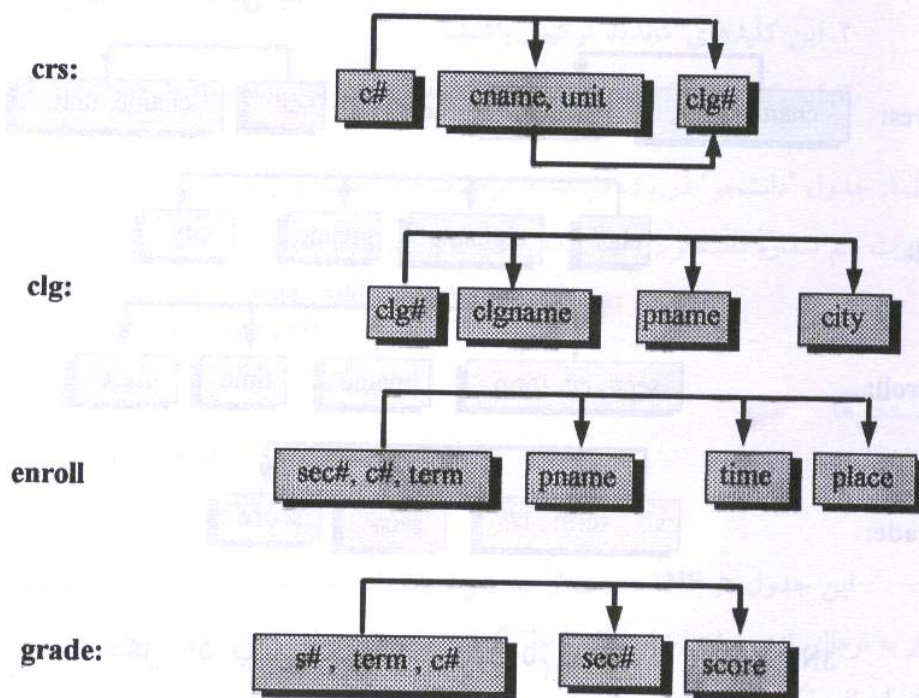
های آن کنار ه بگذار

2. کل کلید اصلی را با صفت های باقیمانده کنار هم بگذار

3. صفت های کلیدی را به عنوان کلید خارجی به 2 اضافه کن .

از سه جدول شکل 5/3 فقط جدول sec در 2NF نیست . با پیروی از الگوریتم بالا

جدول sec به دو جدول enroll و grade تجزیه می شود (شکل 5/4) .



شکل 5/4- مراحل تبدیل بانک اطلاعات ثبت نام دانشگاه از 1NF به 2NF

شکل 5/4- مراحل تبدیل بانک اطلاعات ثبت نام دانشگاه از 1NF به 2NF

تعریف : جدولی در 3NF است که :

1. در 2NF باشد

2. وابستگی انتقالی (وابستگی صفت های غیر کلیدی) نداشته باشد .

الگوریتم زیر ، گذار از 2NF به 3NF را بیان می کند :

1. صفتی را که وابستگی انتقالی ایجاد کرده است ، با همه وابسته های آن کنار هم

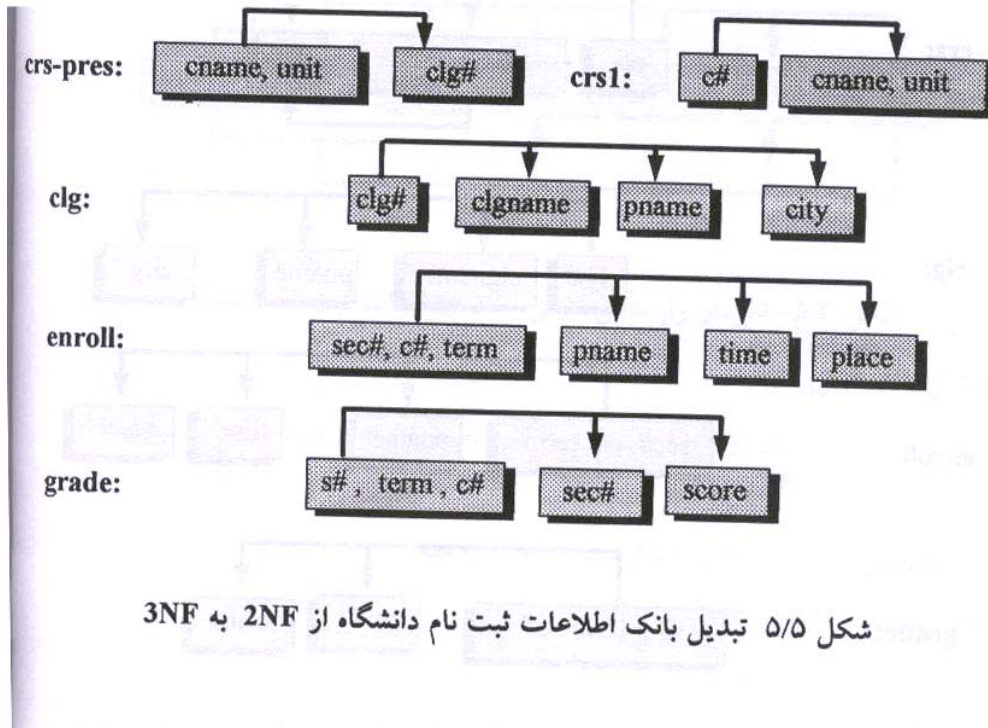
بگذار

2. کلید اصلی را با صفت های باقیمانده کنار هم بگذار

3. صفت های کلیدی را به عنوان کلید خارجی به 2 اضافه کن

در شکل 5/4 فقط جدول crs دارای وابستگی انتقالی است با پیروی از الگوریتم فوق ،

شکل 5/5 به دست می آید .



شکل 5/5 تبدیل بانک اطلاعات ثبت نام دانشگاه از 2NF به 3NF .

در دیدگاه اولیه آقای کاد ، اینجا پایان راه است . آقای بویس روی یک نکته انگشت

گذاشت و فرم نرمال خود (**BCNF** یا **BOYCE_CODD NORMAL**)

را تعریف کرد. این نکته در زیر بررسی می شود .

از 3NF تا BCNF

3NF با مداولی که هر سه شرط زیر را دارند ممکن است مشکل داشته باشد . در این صورت باید این نوع جداول را تا سطح بیشتری نرمال سازی نمود . این سه شرط عبارتند از :

1. جدول دارای حداقل دو کلید کاندید باشد

2. این کلید های کاندید ترکیبی باشند

3. این کلید های ترکیبی ، صفت های مشترکی داشته باشند

مثال 1 : جدول دانشجو در یک موسسه کوچک که دانشجوی همنام ندارد (در این صورت هم شماره دانشجویی کلید کاندید است و هم نام دانشجو) :

stud(s#,sname,address,avg)

candidate key(s#)

candidate key(sname)

وابستگی :

s# → stud

sname → stud

این جدول در 3NF هست . از سه شرط بالا فقط شرط 1 را داراست . بنابراین نیاز به نرمال سازی بیشتر ندارد . با در نظر گرفتن دادهای مناسب ، می توان دید که این جدول ، افزونگی ندارد .

مثال 2 : جدول “ درس – دانشجو ” در همین موسسه :

crs_stud(s#,sname,c#,score)

candidate key(s#,c#)

candidate key(sname,c#)

وابستگی ها :

$(s\#,c\#) \rightarrow crs_stud$

$(sname,c\#) \rightarrow crs_stud$

$s\# \rightarrow sname$

$sname \rightarrow s\#$

این جدول هر سه شرط بالا را داراست . بنابراین ممکن است نیاز به نرمال سازی بیشتر داشته باشد . داده های زیر نشان می دهد که این جدول دارای افزونگی است و برای برطرف نمودن آن نیاز به نرمال سازی بیشتر دارد . می توان این جداول را به دو جدول $grade(s\#,c\#,score),st(s\#,sname)$ شکست (در جدول *grade* می توان به جای $s\#$ از *sname* نیز استفاده کرد .)

crs_stud

S#	sname	C#	Score
S1	علی	C1	17
S1	علی	C2	12
S1	علی	C3	19
S1	علی	C4	20
S1	علی	C5	10

افزونگی

مثال 3: جدول امتحان شامل شماره دانشجویی، شماره درس، رتبه دانشجو در درس (غیر تکراری):

$Exam(s\#, subj\#, rank)$
 $candidate \quad key(s\#, subj\#)$
 $candidat \quad key(subj\#, rank)$

وابستگی ها:

$(s\#, subj\#) \rightarrow Exam$
 $(subj\#, rank) \rightarrow Exam$

این جدول در 3NF هست و هر سه شرط را نیز داراست، ولی نیاز به نرمال سازی بیشتر ندارد (با در نظر گرفتن داده هایی می توان دید که این جدول افزونگی ندارد).

چگونه باید فهمید که چنین جداولی نیاز به نرمال سازی دارند یا خیر؟ پاسخ این سوال را باید در تعریف BCNF جستجو کرد. اگر جدولی در فرم BCNF باشد، نیاز به شکستن ندارد.

▪ تعریف: جدولی BCNF است که ستون های آن فقط به کلید های کاندیدش وابستگی تابعی داشته باشند.

در مثال های 1 و 3، وابستگی به غیر کلید های کاندید وجود ندارد ولی در مثال 2 این نوع وابستگی وجود دارد ($sname \rightarrow s\#, s\# \rightarrow sname$).

توجه به نکات زیر در مورد BCNF ضروری است:

1. بر خلاف فرم های نرمال دیگر ، BCNF بدون استفاده از 3NF و فرم های قبلی نرمال تعریف می شود . غالباً می توان بانک اطلاعات را با استفاده از تعریف BCNF در یک قدم نرمال کرد و نیازی به تعریف وابستگی هایی از قبیل وابستگی انتقالی نیست .

بنابراین ، BCNF جامعترین تعریف نرمال سازی بر مبنای وابستگی تابعی را به طور مستقل ارائه می دهد .

2. در مواردی ، نرمال سازی تا BCNF لازم نیست و بهتر است از تبدیل جدول فرم 3NF به BCNF خودداری کرد .

به عنوان مثال ، برای ثبت آدرس فعلی و آدرس خانواده دانشجویان می توان جدول زیر را تعریف کرد و برای هر دانشجو بیش از یک آدرس در نظر گرفت :

$s_addr(s\#,city,no,zip)$

$candidate\ key(s\#,city\#)$

$candidat\ key(s\#,zip)$

کد پستی برای هر شهر منحصر به فرد است و می تواند به جای نام شهر قرار گیرد . این جدول در فرم 3NF هست ولی در فرم BCNF نیست ، زیرا داریم :

$zip \rightarrow city$

با اینهمه نمی توان با اطمینان خاطر این جدول را شکست زیرا کد پستی ، بخش جدایی ناپذیر آدرس است و جدا کردن آن باعث پیچیده شدن پرس و جوی مربوط به آدرس می شود .

چون وابستگی $city \rightarrow zip$ نیز برقرار است ، همین مطلب در مورد 2NF هم صادق است . بطور خلاصه باید گفت که پرس و جو نیز می تواند در طراحی جداول نقش داشته باشد . اگر طراح تشخیص دهد که تجزیه یک جدول ، هر چند افزونگی هم داشته باشد ، باعث پایین آمدن سرعت اکثر پرس و جو ها می شود مجاز است از نرمالتر سازی آن صرفنظر کند .

مثال : در بانک اطلاعات زیر ابتدا کلید های کاندید را بیابید و سپس آنرا به طور کامل نرمال سازی نمایید :

$$R = \{A, B, C, D, E, F, G\}$$

$$F = \{AF \rightarrow BE, FC \rightarrow DE, F \rightarrow CD, D \rightarrow E, C \rightarrow A\}$$

حل :

سمت راست وابستگی ها را به یک صفت تبدیل می کنیم .

$$1) AF \rightarrow B$$

$$2) AF \rightarrow E$$

$$3) FC \rightarrow D$$

$$4) FC \rightarrow E$$

$$5) F \rightarrow C$$

$$6) F \rightarrow D$$

$$7) D \rightarrow E$$

$$8) C \rightarrow A$$

از شماره 5 می توانیم برای ساده کردن سمت چپ شماره های 3 و 4 استفاده کنیم .

در نتیجه : $3(F \rightarrow D, 4)F \rightarrow E$

از شماره های 5 و 8 می توان نتیجه گرفت که $F \rightarrow A$

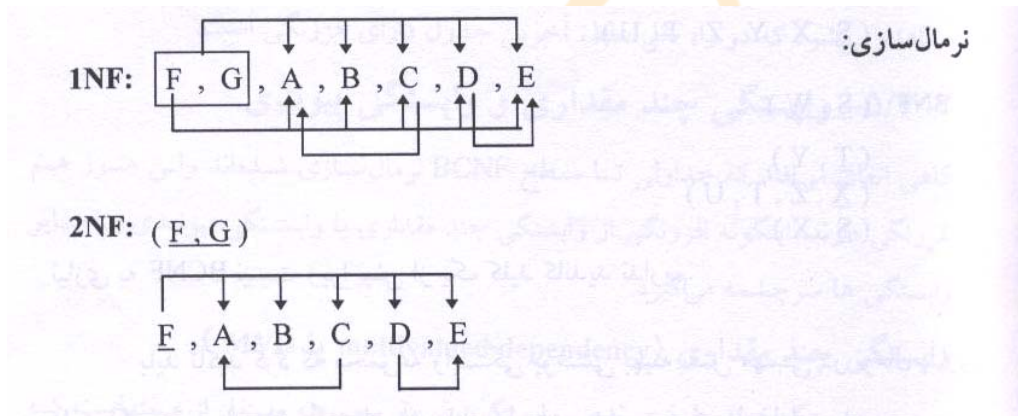
این نتیجه را روی شماره های 1 و 2 اعمال می کنیم :

$$1) F \rightarrow B, 2) F \rightarrow E$$

$$F_{opt} = \{F \rightarrow A, F \rightarrow B, F \rightarrow C, F \rightarrow D, F \rightarrow E, D \rightarrow E, C \rightarrow A\} \quad \text{پس}$$

در نتیجه صفت F همه صفت های دیگر ، بجز G ، را می دهد . پس (F و G) کلید کاندید است . این کلید کاندید منحصر به فرد است زیرا هیچ صفتی F و G را نمی دهد (یعنی در هر کلید کاندید ، این دو صفت لازم هستند) .

نرمال سازی :



$$3NF : (\underline{F}, \underline{G})$$

$$(\underline{C}, A)$$

$$(\underline{D}, E)$$

$$(\underline{E}, B, C, D)$$

نیازی به BCNF نیست زیرا بیش از یک کلید کاندید نداریم .

مثال : رابطه $R = \{X, Y, Z, S, T, U, W\}$ را با FD های زیر در نظر بگیرید :

$$F = \{S \rightarrow X, T \rightarrow Y, X \rightarrow Y, XY \rightarrow TUZ\}$$

الف) کلید های کاندید را بیابید .

ب) به طور کامل نرمال سازی نمایید .

حل :

الف :

$$S \rightarrow X, X \rightarrow Y \Rightarrow S \rightarrow XY$$

$$S \rightarrow XY, XY \rightarrow TUZ \Rightarrow S \rightarrow T, S \rightarrow U, S \rightarrow Z$$

$$\Rightarrow (S, W)$$

$$F_{OPT} = \{S \rightarrow X, S \rightarrow Y, S \rightarrow Z, S \rightarrow T, S \rightarrow U, T \rightarrow Y, X \rightarrow Y, X \rightarrow T, X \rightarrow U, X \rightarrow Z\}$$

$$2NF : (\underline{S}, W)$$

$$(\underline{S}, X, Y, Z, T, U)$$

$$3NF : (\underline{S}, W)$$

$$(\underline{T}, Y)$$

$$(\underline{X}, Z, T, U)$$

$$(\underline{S}, X)$$

نیازی به BCNF نیست زیرا بیش از یک کلید کاندید نداریم .

باید تاکید کرد که مجموعه وابستگی پوششی بهینه نقش مهمی در نرمال سازی دارد .

اگر بعضی از وابستگی های آن را از نظر دو برداریم به نتیجه غلطی م ی‌رسیم . به مثال

زیر توجه کنید :

مثال : بانک اطلاعات زیر را نرمال سازی کنید .

$$R = (A, B, C, D, E, F, G)$$

$$F = \{(A, B) \rightarrow R, A \rightarrow G, B \rightarrow EF, G \rightarrow DF\}$$

راه حل غلط : بدون یافتن F^+

2NF (A, G)
(B, E, F)
(A,B, C,D,F,G)

3NF (A , G)
(B, E ,F)
(A,B,C)

3NF (G,D,F)
(A,G)
(B,E,F)
(A,B , C)

راه حل صحیح : با افزودن $A \rightarrow DF$ به وابستگی های تابعی

2NF (A , D ,F , G)
(B, E , F)
(A,B ,C)
3NF (G,D,F)
(A ,G)
(B , E , F)
(A,B , C)

مشاهده می شود که در راه حل غلط ، آخرین جدول دارای افزونگی است .

2-3/5- وابستگی چند مقداری و وابستگی پیوندی

گاهی اتفاق م یافتد که جداولی تا سطح BCNF نرمال سازی شده اند ولی هنوز هم افزونگی دارند . اینگونه افزونگی از وابستگی چند مقداری یا وابستگی پیوندی یا سایر وابستگی ها سرشچمه می گیرد .

وابستگی چند مقداری (*MVD* یا *multivalued dependency*)

وابستگی چند مقداری نوعی وابستگی بین دو مجموعه مستقل از صفتهاست که از وابستگی تابعی عام تر است . این حالت در مثال زیر مشاهده می شود .

مثال : جدول دانشجو شامل نام دانشجو ، اساتیدی که دانشجو با آنها درس دارد و وامهایی که دانشجو دریافت کرده و تاریخ آن وام ها (شکل صفحه بعد)

در این مثال ، نام اساتید دانشجو و وام های دریافت شده توسط دانشجو از یکدیگر مستقل هستند یعنی وابستگی تابعی ندارند . اگر یک دانشجو چند استاد داشته باشد و چند وام دریافت کرده باشد ، افزونگی بداده ایجاد می شود .

Sname	Prof	Loan	Date
حمید	حق جو	مسکن	1381
حمید	حق جو	ضروری	1383
حمید	جاهد	مسکن	1381
حمید	جاهد	ضروری	1383

حمید	حق جو	ضروری	1384
------	-------	-------	------

در جدول فوق مشاهده می شود که نام اساتید و وام های حمید تکرار شده اند (افزونگی) . این در حالی است که جدول فوق تا سطح BCNF نرمال سازی شده است . زیرا هیچگونه وابستگی معنی دار به جز وابستگی به کلید اصلی ، که شامل تمام صفت ها است ، وجود ندارد .

این نوع وابستگی و فرم نرمال مربوط به آن به صورت های زیر تعریف می شود :

تعریف : هر گاه دو ارتباط مستقل بین مجموعه صفت های یک رابطه R مثل $A:B$ و $A:C$ وجود داشته باشد ، وابستگی چند مقداری در رابطه R برقرار است که به دو صورت زیر نشان داده می شود :

$$A \twoheadrightarrow B, A \twoheadrightarrow C$$

$$A \twoheadrightarrow B/C$$

وابستگی چند مقداری به شکل های متفاوت در منابع مختلف تعریف شده است . یک تعریف دیگر هم در زیر آمده است :

تعریف : در رابطه $R(A,B,C)$ داریم $A \twoheadrightarrow B/C$ اگر هر مقدار A به طریقی به مقادیر B وابسته باشد که به C ارتباطی پیدا نکند .

مثال : در جدول فوق وابستگی های چند مقداری زیر وجود دارند :

$$sname \twoheadrightarrow prof, sname \twoheadrightarrow loan, data$$

$$sname \twoheadrightarrow prof | loan.date$$

تعریف: یک وابستگی چند مقداری $A \rightarrow B$ در رابطه R را جزئی (trivial) گویند اگر یکی از دو شرط زیر برقرار باشد:

$$1. B \subset A$$

$$2. A \cup B = R$$

اگر در یک وابستگی چند مقداری هیچ کدام از دو شرط فوق برقرار نباشد، به آن وابستگی چند مقداری غیر جزئی (*nontrivial*) گویند.

در جدول فوق هر دو وابستگی چند مقداری، غیر جزئی هستند.

تعریف: رابطه R در 4NF (*Fourth Normal Form*) است در صورتیکه اگر وابستگی چند مقداری غیر جزئی $A \rightarrow B$ در R وجود داشته باشد آنگاه A برابر کلید R باشد.

جداولی که دارای وابستگی چند مقداری هستند را می توان به صورت زیر تجزیه کرد و افزودگی را از بین برد.

جدول اول شامل مجموعه صفت های A, B

جدول دوم شامل مجموعه صفت های A, C

مثال:

جدول فوق به دو جدول زیر تجزیه می گردد و افزودگی از بین می رود.

sname	Prof
حمید	حق جو

حمید	جاهد
------	------

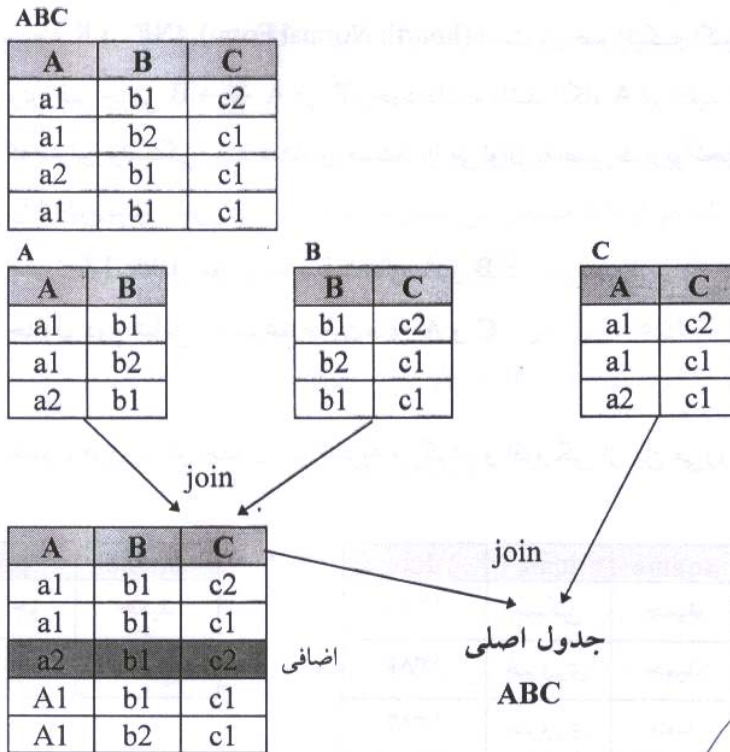
sname	Loan	Date
حمید	مسکن	1381
حمید	ضروری	1383
حمید	ضروری	1384

وابستگی پیوندی (join dependency یا JD)

در مواردی نمی‌توان جدولی را بطور صحیح به دو جدول تجزیه نمود ولی تجزیه آن به سه جدول یا بیشتر امکان پذیر است (شرایط صحت تجزیه در زیر آمده است). چنین جداول دارای وابستگی پیوندی هستند . این مفهوم ابتدا با مثال بیان می‌شود و سپس همراه با فرم نرمال مربوطه تعریف می‌گردد .

مثال :

جدول ABC زیر را که در آن هر سه ستون روی هم‌دیگر کلیدکandid را تشکیل می‌دهند می‌توان به سه جدول A و B و C تجزیه نمود ولی تجزیه آن بطور صحیح به دو جدول امکان ندارد .



تعریف: اگر R رابطه و ستون های هر یک از رابطه های A, B, \dots, P زیر مجموعه ستون های R باشند آنگاه R دارای وابستگی پیوندی روی A, B, \dots, P است اگر و تنها

$$R = A \bowtie B \bowtie \dots \bowtie P$$

اگر داشته باشیم

تعریف:

جدول R در $5NF$ (**Fifth Normal Form**) است اگر و تنها لگر فقط به کلید های

کاندیدش وابستگی پیوندی داشته باشد.

3-3-5- شرایط تجزیه جداول

هر بانک اطلاعات دورکن اصلی دارد : داده ها و ارتباط ها ، یا به عبارتی وابستگی ها . یک بانک اعم از اینکه یک جدول باشد یا چند جدول ، همواره باید این دو رکن آن حفظ شود . به عبارت دیگر ، در تجزیه جداول نباید اطلاعات و وابستگی های اصلی از دست برود .

• مثال :

جدول زیر و وابستگیهای آن را در نظر بگیرید .

$grade(s\#,c\#,sname,score)$

$(s\#,c\#) \rightarrow grade$

$s\# \rightarrow sname$

این جدول را می توان به صورت غلط زیر تجزیه کرد :

$(s\#,c\#,score)$

$(c\#,sname)$

این تجزیه غلط است زیرا $c\#$ و $sname$ به هم مرتبط نیستند . ما این واقعیت را از معنای صفتها تشخیص می دهیم . آیا راهی وجود دارد که بدون توجه به معنا صفتها ، و فقط با استفاده از وابستگی های موجود بتوانیم به درستی یا نادرستی یک تجزیه پی ببریم ؟

در پاسخ به این سوال ثابت شده است که علاوه بر رعایت ضوابط نرمال سازی ، باید مطمئن شد که دو مورد زیر رعایت شده است :

1. تجزیه یک شما (*schema*) به چند شمای کوچکتر باید “ پیوند پذیر ”

(*loss_less join*) باشد ، یعنی به ازای تمام جداول مربوطه ، از پیوند طبیعی

آن جداول دقیقاً جدول اصلی به دست آید .

■ تعریف :

تجزیه شمای R به $\{R_1, \dots, R_n\}$ در صورتی “ پیوند پذیر ” است که برای تمام جداول r از شمای R داشته باشیم :

$$r = \prod_{R_1}(r) \dots \prod_{R_n}(r)$$

این تعریف و همه تعاریف مشابه با توجه به معنی داده ها و صفت ها بیان می شود ، نه خود داده ها ، یعنی باید برای تمام جداول درست باشد .

واضح است که این تعریف قابل پیاده سازی نیست ، زیرا نمی توان تمام جداول ممکن روی یک شما را بررسی کرد . پژوهشگران راه حل ساده و عملی زیرا را یافته اند :

“ تجزیه R به $\{R_1, R_2\}$ در صورتی پیوند زیر است که کلید حداقل یکی از دو جدول ، ستونهای مشترک آن دو باشد ”

به عبارت دیگر :

$$R = \{R_1, R_2\} \Leftrightarrow R_1 \cap R_2 \rightarrow R_1 \quad R_1 \cap R_2 \rightarrow R_2$$

باید توجه نمود که این قاعده برای دو جدول بیان شده است . د مواردیکه جدولی به بیش از دو جدول تجزیه می شود ، باید صحت آن را قدم به قدم بررسی نمود .

مثال : بانک زیر را نرمال سازی نمایید .

$$R = (A, B, C, D, E, F)$$

$$F = \{(A, B) \rightarrow R, A \rightarrow CD, B \rightarrow E\}$$

حل : در این مثال فقط تبدیل به 2NF لازم است اما این عمل دوبار باید انجام شود ، یکبار برای A و بار دیگر برای B . اگر هر دو را با هم انجام دهیم ، نتیجه به صورت

زیر خواهد بود . در این حالت قاعده پیوند پذیری را نمی توان به وضوح دید و بررسی کرد .

$$R1(\underline{A}, C, D), R2(\underline{B}, E), R3(\underline{A}, \underline{B}, F)$$

اما اگر این بانک را در دو مرحله به 2NF تبدیل کنیم ، صحت قاعده پیوند پذیری به وضوح دیده می شود :

مرحله اول :

$$R1(\underline{A}, C, D) \quad , \quad R2(\underline{A}, \underline{B}, E, F)$$

$$R1(\underline{A}, C, D) \\ R21(\underline{B}, E) \quad , \quad R22(\underline{A}, \underline{B}, F)$$

2. تجزیه یک شما به چند شمای کوچکتر باید حافظ وابستگی (*dependency preserving*) باشد ، یعنی تمامی وابستگی های اصلی حفظ شود .

پژوهشگران راه عملی برای بررسی “ حافظ وابستگی ” بودن یک تجزیه را نیز پیدا کرده اند . این راه حل به صورت زیر است . یادآوری م یکنیم که یک بانک اطلاعات رابطه ای را م توان به صورت $\{ R, F \}$ نمایش داد ، یعنی شمای R با مجموعه وابستگی F .

تجزیه بانک اطلاعات $\{ R, F \}$ به $\{ \{R1, F1\}, \dots, \{Rn, Fn\} \}$ در صورتی حافظ وابستگی است که داشته باشیم :

$$F^+ = \{F1 \cup \dots \cup Fn\}^+$$

