عنوان: **نرمال سازی (قسمت دوم: Second Normal Form)**
نویسنده: محمد سلیم ابادی
تاریخ: ۱۳۹۱/۱۱/۱۳ ۱۷:۳۵
آدرس: www.dotnettips.info
گروه‌ها: second normal form ,نرمالسازی ,database design, normalization

وابستگی تابعی

برای وارد شدن به بحث نظری نرمالسازی نیاز هست با مفهوم وابستگی تابعی آشنا شویم.
وابستگی تابعی یک مبحث نسبتا مفصل و تئوری هست که زمان زیادی برای شرح جزئیات آن نیاز هست در نتیجه در حد آشنایی و نیازمان به آن توجه خواهیم داشت.

به جدول زیر نگاه کنید:

primary key		primary key	
S#	City	P#	Qty
S1	London	P1	100
S1	London	P2	100
S2	Paris	P1	200
S2	Paris	P2	200
S3	Paris	P2	300
S4	London	P2	400
S4	London	P4	400
S4	London	P5	400

این جدول نشان می‌دهد هر عرضه کننده(S#) چه قطعه (P#) را به چه تعداد (Qty) تولید کرده است. City هم شهریست که عرضه کننده در آن سکونت دارد.

از داده‌های فعلی جدول می‌شود برداشت‌های مختلفی داشت که چندتای آن به قرار زیر:
عرضه کنندگان یکسان دارای شهرهای یکسان هستند
هر عرضه کننده و قطعه تنها با یک مقدار از qty در تنظار است.

تعریف وابستگی تابعی یا functional dependency

تعریف رسمی:
اگر r یک رابطه و X و Y زیر مجموعه‌های دلخواهی از مجموعه خصیصه‌های r باشند آنگاه می‌گوییم Y به صورت تابعی وابسته به X است و آن را به صورت زیر می‌نویسیم:

X-->Y

اگر و تنها اگر در هر مقدار مجاز و ممکن از r، هر مقدار X متناظر با دقیقا یک مقدار از Y باشد. یعنی به ازای هر X تنها یک Y داشته باشیم. به بیان دیگر هرگاه دو چندتایی از r مقدار مقدار X یکسانی داشته باشند آنگاه مقدار Y آنها یکسان باشد .

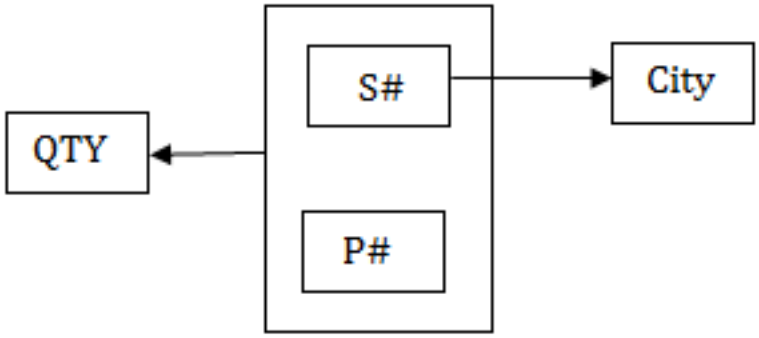
گفته شد که هر عرضه کنند تنها با یک شهر تناظر دارد. مثلا عرضه کنده ای با مقدار S1 تنها با شهر London در تناظر است. و به ازای هر عرضه کننده قطعه تنها یک QTY خواهیم داشت مثلا به ازای عرضه کننده با مقدار S4 و قطعه با مقدار P2 تنها یک سطر (در نتیجه یک Qty) وجود دارد (این دو خصیصه کلید هستند)

اما P# به S# وابستگی تابعی ندارد. مثلا به ازای S4 ما چند عرضه کننده خواهیم داشت.

وابستگی تابعی را می‌توان بشکل نمودار در آورد. در زیر نمودار وابستگی همراه با وابستگی‌های تابعی جدول مورد نظر آمده است:

$S\# \rightarrow City$

$\{S\#, P\#\} \rightarrow Qty$



تعریف شکل نرمال دوم

یک متغیر رابطه ای به شکل دوم نرمال است اگر و فقط اگر به شکل اول نرمال بوده و هر خصیصه غیر کلیدی وابسته به کلید اولیه باشد.

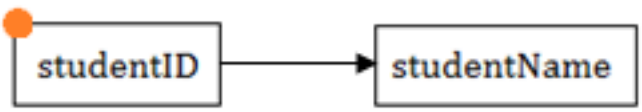
بر می‌گردیم به آخرین جدول مطلب گذشته یعنی:

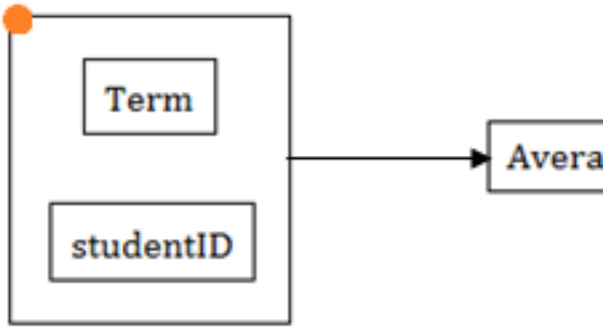
PK			PK
معدل	ترم	نام دانشجو	کد دانشجو
۱۴	۱	نام۱	کد۱
۱۵	۲	نام۱	کد۱
۱۸	۱	نام۲	کد۲
۱۳	۲	نام۲	کد۲
۱۵	۳	نام۲	کد۲
۱۷	۴	نام۲	کد۲
۱۴	۱	نام۳	کد۳
۱۷	۲	نام۳	کد۳
۱۲	۳	نام۳	کد۳
		نام۴	کد۴

کلید اولیه این جدول از ترکیب دو ستون کد دانشجو و ترم تشکیل شده است.
معدل را کلید اولیه تعیین می‌کند یعنی معدل وابسته به مقدار کلید اولیه است، اما نام دانشجو وابستگی به کلید اولیه ندارد و به جای آن وابسته به ستون کد دانشجو است. در نتیجه طبق تعریفی که داشتیم این جدول به شکل دوم نرمال نیست.
این جدول دقیقا مشابه به جدول عرضه کننده - قطعات است (که در ابتدا مطلب آمده است) پس نمودار FD آن نیز با FD این جدول برابر است.

برای تبدیل از فرم 1 به فرم 2 نرمال باید جدول را تجزیه کنیم به دو جدول:
جدول دانشجو (کد دانشجو - نام دانشجو)
جدول معدل (کد دانشجو - ترم - معدل)

به نمودار FD جدول فوق بعد از تجزیه شدن دقت بفرمایید:





همانطور که مشاهده می‌شود فلش‌ها تنها از خصیصه‌های کلید اولیه خارج شده اند در حالی که قبل از تجزیه شدن فلش ای وجو داشت که از کلید اولیه خارج نشده بود. کلیدهای اولیه توسط نقطه نارنجی رنگ علامت گذاری شده اند.

و بالاخره فرم دوم نرمال جدول سابق:

کد دانشجو	ترم	معدل
کد۱	۱	۱۴
کد۱	۲	۱۵
کد۲	۱	۱۸
کد۲	۲	۱۳
کد۲	۳	۱۵
کد۲	۴	۱۷
کد۳	۱	۱۴
کد۳	۲	۱۷
کد۳	۳	۱۲

کد دانشجو	نام دانشجو
کد۱	نام۱
کد۲	نام۲
کد۳	نام۳
کد۴	نام۴

کلیدهای اولیه با نقطه بنفش علامت گذاری شده است.

در اینجا با تجزیه جدول، به شکل سوم نرمال رسیدیم. در پست بعدی مثالی از یک جدول نرمال دوم خواهم آورد و همزمان با بررسی معایب آن شکل سوم نرمال را نیز معرفی خواهم نمود.

مرجع

کتاب پایگاه داده‌ی C.J. Date

نظرات خوانندگان

نویسنده: senaps
تاریخ: ۱۳۹۱/۱۱/۱۳ ۱۸:۴۷

خوب من خیلی خوشحالم....
من همیشه دیتابیسم رو به همین شکل طراحی میکنم! (یعنی حداقل جداولم حد نرمال دوم رو دارن! :))....
حالا تا ببینم در اینده چی‌میشه ماجرا که ببینم بر این اساس، ایا من کلا جداولم رو نرمال طراحی میکنم یا چی؟!
اخه من هیچوقت نرمال سازی رو یاد نگرفتم(البته تو دانشگاه هم درس نداد این مسئله رو استاد مربوطه...!) ولی خوب طراحی دیتابیس رو دوتایی با هم اینجوری کار کردیم که من معمولا مثل جدول‌های اخر این پست کار میکنم....

نویسنده: محمد سلم ابادی
تاریخ: ۱۳۹۱/۱۱/۱۳ ۲۰:۵۸

این دو جدول آخر به شکل سوم نرمال هستند. یعنی شرط نرمال سوم را نیز محقق کرده اند. در مطلب بعدی یک مثال از جدولی خواهم آورد که به شکل دوم نرمال بوده ولی به شکل سوم نرمال نباشد.

نویسنده: حسینی
تاریخ: ۱۳۹۲/۰۵/۲۶ ۱۷:۴

با سلام؛ شما ترکیب کد دانشجو و ترم رو کلید اصلی در نظر گرفتید؛ به نظرتون بهتره که کلید اصلی رو یه ستون جدا در نظر بگیریم یا همین کاری که شما انجام دادید؟ لطفا مزایا و معایب هر کدام را بفرمائید.
با سپاس فراوان

نویسنده: محمد سلیم ابادی
تاریخ: ۱۳۹۲/۰۵/۲۶ ۲۲:۳۲

این امکان هم وجود داره که یک ستون دیگه به جدول اضافه کنید و آن را به عنوان PK در نظر بگیرید. اما باید به این نکته بسیار مهم نیز توجه داشته باشید که نمیشه آن دو ستون (کددانشجو و ترم) را همینطور به حال خود رها کرد. با این فرض که با اضافه شدن این ستون دیگه هیچ دو سطر تکراری به خاطر uniqueness بودن PK نخواهیم داشت.
شما لازمه که یک قید منحصربفرد تکریبی در کنار PK برای آن دو ستون در جدول ایجاد کنید. تا به ازای یک ترم معین و یک دانشجو معین تنها یک معدل ثبت بشه.
پس با لحاظ توضیحات فوق جدول به این شکل در می‌آید:

```
create table Avgs
(
   identifier int not null identity(1,1) primary key,
   student_id varchar(10) not null
        references Students
   term_id tinyint not null
        references Terms
   average tinyiny,
   check (averge between 0 and 20),
   unique (student_id, term_id)
)
```

ستونی به نام identity وظیفه PK را به عهده می‌گیره. و از نوع identity هم هست.
دو ستون کد دانشجو و کد معدل کلیدهای خارجی هستند. و ترکیب این دو ستون برای حفظ یکپارچگی و جامعیت داده‌ها منحصربفرد نظر گرفته شدن.
یک قید هم برای معدل گذاشته شده که معدل غیر متعارف در آن درج نشه.

به سناریوی زیر توجه کنید:

فرض کنید میخواهید بر اساس کد دانشجو و یک ترم معین در جدول برای بدست آوردم معدل جستجو داشته باشید. خب لازمه که بر اساس آن دو ستون جستجو داشته باشید نه آن ستونی که به عنوان PK در نظر گرفته شده. پس محتویات ستون identity کاملا مصنوعی و غیر طبیعی بوده و بطور مستقیم قابل استفاده نیست.

البته لازم به ذکر که عموما کلید اولیه همزمان unique clustered index نیز در نظر گرفته میشه. اگر داده‌های این ستون بطور متوالی و پشت سر هم در جدول درج نشن باعث ایجاد fragmentation میشه. و لازمه که ایندکس rebuild بشه.
و اگر کلید اولیه ترکیبی باشه کار در ارتباطات کمی دشوار میشه چون نیاز به کلیدهای خارجی ترکیبی نیز هست.
در joinها نیز چون پیوند بر اساس کلید اولیه و کلیدخارجی هست، هر چه کلید اولیه سبک‌تر باشه (حجم کمتری داشته باشه و از نوعی باشه که سریع‌تر توسط پردازنده پردازش بشه) سرعت پردازش نیز طبیعتا افزایش پیدا میکنه.

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۵/۲۷ ۰:۴۳

به این نوع کلیدها surrogate key هم می‌گن.

نویسنده: محمد سلیم ابادی
تاریخ: ۱۳۹۲/۰۵/۲۷ ۲:۲۱

بله همینطوره.
سایت ویکی توضیحات خوبی راجب معایب و مزایای استفاده از surrogate key داده.
یه مرور بسیار جزئی که به معایب و مزایا داشتم متوجه شدم که در پست قبلیم از معایبش به normalization و از مزایاش به performance اش اشاره ای داشتم.