

مقدمه

نرمال سازی یا normalization باعث جلوگیری از تکرار و افزونگی اطلاعات می‌شود. و همچنین مانع از یکسری ناهنجاری‌ها در عملیات درج، بروز رسانی، حذف و انتخاب خواهد شد. شکل‌های نرمال متعددی تعریف شده اند که به شرح زیر است:

شکل نرمال اول (1NF)

شکل نرمال دوم (2ND)

شکل نرمال سوم (3NF)

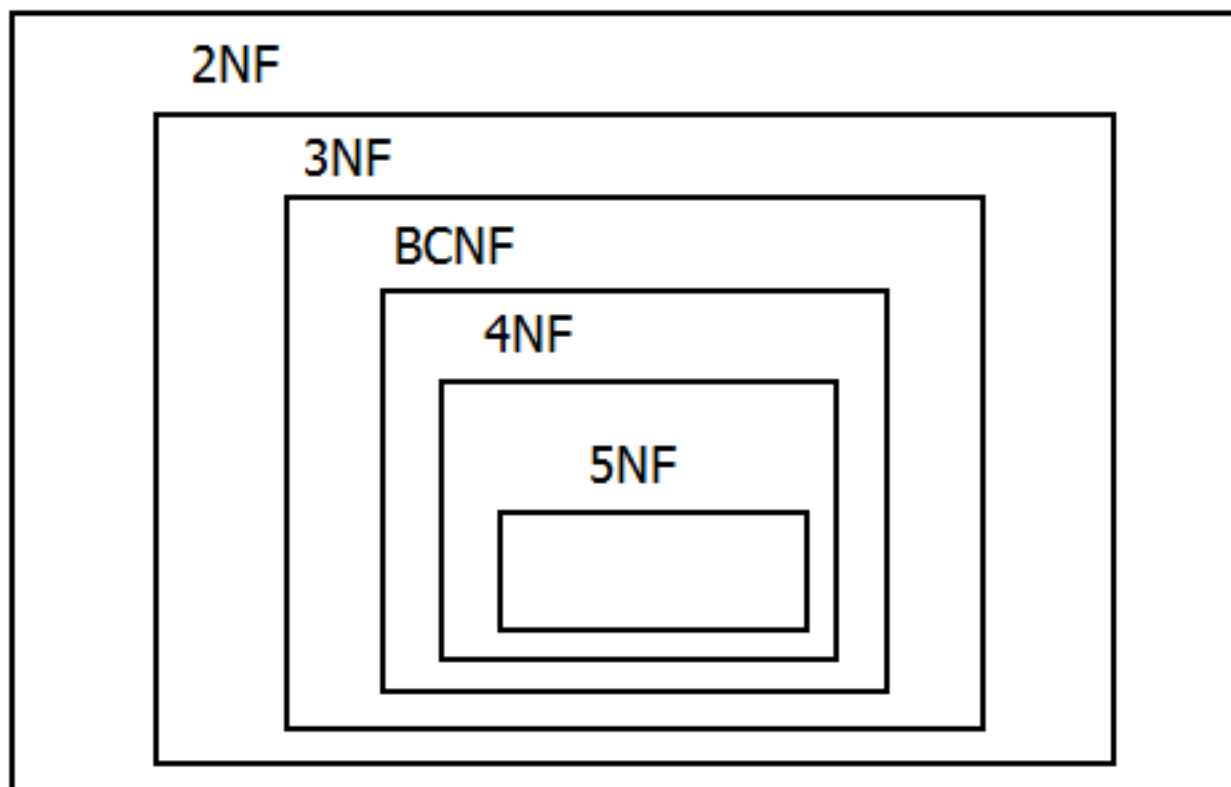
شکل نرمال بویس کاد (BCNF)

شکل نرمال چهارم (4NF)

شکل نرمال پنجم (5NF)

سه شکل اول نرمال یعنی 1NF ، 2NF و 3NF توسط دکتر Codd تعریف شده اند. شکل نرمال بویس کاد نیز که یک تعریف اصلاح شده و قوی‌تر از 3NF به Boyce و Codd منسوب است . بعد از آن Fagin شکل چهارم نرمال (4NF) را تعریف کرد (چرا که در آن زمان BCNF شکل سوم نرمال خوانده می‌شد).

1NF



تصویر فوق می‌گوید اگر جدولی در شکل سوم نرمال باشد حتما دارای شکل دوم نرمال و شکل اول نرمال هم خواهد بود.

شکل اول نرمال (First Normal Form)

تعریف رسمی:

یک متغیر رابطه ای به شکل اول نرمال است اگر و فقط اگر در هر مقدار مجاز آن متغیر رابطه ای، هر چندتایی فقط یک مقدار برای هر خصیصه داشته باشد.

منظور از اصطلاحات متغیر رابطه ای، چندتایی و خصیصه به طور غیر رسمی به ترتیب برابر است با جدول، سطر و ستون.

قسمت کلیدی تعریف، این جمله است: "فقط یک مقدار برای هر خصیصه داشته باشد"

به دو جدول زیر توجه کنید، این جداول به شکل اول نرمال نمی‌باشد چرا که به ازای هر مشتری برای خصیصه شماره تلفن چند مقدار خواهیم داشت:

Primary Key				
کد مشتری	نام مشتری	شماره تلفن	شماره تلفن	شماره تلفن
کد ۱	مشتری ۱	مقدار ۱	مقدار ۲	
کد ۲	مشتری ۲			
کد ۳	مشتری ۳	مقدار ۱		
کد ۴	مشتری ۴	مقدار ۱	مقدار ۲	مقدار ۳

Primary Key		
کد مشتری	نام مشتری	شماره تلفن ها
کد ۱	مشتری ۱	مقدار ۱ - مقدار ۲
کد ۲	مشتری ۲	
کد ۳	مشتری ۳	مقدار ۱
کد ۴	مشتری ۴	مقدار ۱ - مقدار ۲ - مقدار ۳

در جدول اول ستون شماره تلفن چند بار تکرار شده است. یعنی برای یک مشتری چند مقدار برای خصیصه شماره تلفن خواهیم داشت که این مغایر با تعریف شکل اول نرمال است. همین اتفاق نیز در جدول دوم افتاده است با این فرق که مقادیر خصیصه شماره تلفن در یک ستون درج شده اند.

برای تبدیل جدول غیر نرمال فوق به یک جدول نرمال اول، بایستی کاری کنیم که خصیصه شماره تلفن فقط یک مقدار را بگیرد. یعنی: در جدول فوق می بینید که برای خصیصه شماره تلفن به ازای هر سطر فقط یک مقدار داریم.

Primary Key		Primary Key	
شماره تلفن	نام مشتری	کد مشتری	شماره تلفن
مقدار ۱	مشتری ۱	کد ۱	
مقدار ۲	مشتری ۱	کد ۱	
	مشتری ۲	کد ۲	
مقدار ۱	مشتری ۳	کد ۳	
مقدار ۱	مشتری ۴	کد ۴	
مقدار ۲	مشتری ۴	کد ۴	
مقدار ۳	مشتری ۴	کد ۴	

در جدول غیر نرمال مثال پیشین چند مقدار برای یک خصیصه داشتیم. حال به مثالی می پردازیم که یک مجموعه از خصیصه ها چند

بار تکرار می‌شوند.

به جدول غیر نرمال زیر توجه کنید. دو خصیصه ترم و معدل چند بار در جدول تکرار می‌شوند. اصلاحا به این‌ها گروه‌های تکرار شونده می‌گویند.



کد دانشجو	نام دانشجو	ترم	معدل	ترم	معدل	ترم	معدل	ترم	معدل
کد ۱	نام ۱	۱	۱۴	۲	۱۵				
کد ۲	نام ۲	۱	۱۸	۲	۱۳	۳	۱۵	۴	۱۷
کد ۳	نام ۳	۱	۱۴	۲	۱۷	۳	۱۲		
کد ۴	نام ۴								

گروه‌های تکرار شونده را با آکولاد ({}) مشکل کرده ام. این گونه جداول (که حتی در شکل نرمال اول هم قرار ندارند) مشکلات فراوانی دارند که در زیر به مواردی اشاره خواهیم داشت:

چگونه معدل ترم ۵ را در جدول درج کنیم؟ پس برای اینکه بتوانیم تمام معدل‌ها را در جدول داشته باشیم باید به تعداد حداکثر ترم تحصیلی گروه‌های تکرار شونده در جدول داشته باشیم.

برای دانشجویی که فقط یک ترم تحصیل کرده است تمام گروه‌های تکرار شونده به غیر از یکی خالی خواهد ماند. فضای بسیاری به هدر خواهد رفت.

گزارش گیری بسیار سخت خواهد شد. بطور نمونه، چطور می‌خواهید بالاتری معدل دانشجویان را بدست بیاورین؟

پس با تبدیل جدول غیر نرمال به شکل نرمال اول، به مشکلات فوق غلبه خواهیم کرد:

PK		PK	
معدل	ترم	نام دانشجو	کد دانشجو
۱۴	۱	نام ۱	کد ۱
۱۵	۲	نام ۱	کد ۱
۱۸	۱	نام ۲	کد ۲
۱۳	۲	نام ۲	کد ۲
۱۵	۳	نام ۲	کد ۲
۱۷	۴	نام ۲	کد ۲
۱۴	۱	نام ۳	کد ۳
۱۷	۲	نام ۳	کد ۳
۱۲	۳	نام ۳	کد ۳
		نام ۴	کد ۴

اما یک متغیر رابطه ای که فقط به صورت شکل اول نرمال است ساختاری دارد که به دلایل متعدد، نامطلوب است.

در جدول فوق اطلاعاتی وجود دارد که به دفعات تکرار شده است. مثلاً نام دانشجو به تعداد ترم‌ها تکرار شده است. در صورتی که باید نام دانشجو یکبار ذخیره شده باشد. پس یک جدولی که به فرم نرمال اول هست می‌تواند افزونگی اطلاعات داشته باشد.

در بخش بعدی ابتدا وابستگی تابعی مورد بررسی قرار خواهد گرفت سپس به فرم دوم نرمال پرداخته خواهد شد.

نظرات خوانندگان

نویسنده: سعید

تاریخ: ۱۳۹۱/۱۱/۱۳ ۱۶:۳۲

با تشکر از مطلب خوبتان.

این نوع مباحث رو با orm ها و کلاس های دات نتى بهتر ميشه توضيح داد. مثلا يك مشتري داريم با چندتا تلفن. يك دانشجو داريم با چندتا ترم و درس و نمره. اين چندتا رو ميشه به صورت يك icollection تعريف كرد در يك كلاس بجاي اينكه پشت سر هم خاصيت اضافه كنيم. يا حتى زمانيكه مشتري سه تا تلفن داره مشكلي نداره تمامش در همان جدول اصلى قرار بگيره. در ef به اين نوع ها، [complextypes](#) گفته ميشه (يك خاصيت تو در تو در كلاس، حاليكه خاصيت كلاس خودش از نوع يك كلاس هست اما اين كلاس تبديل به يك جدول جدا نميشه) يا مثلا در nhibernate به اون [component mapping](#) هم مي گن.

نویسنده: سمیرا

تاریخ: ۱۳۹۲/۱۰/۱۰ ۲۱:۲۰

با سلام

میدونید چرا رابطه های دو صفته BCNF هستن؟