

گزارش پروژه پایگاه داده ها

رسول ابوالحسن زاده

95149004

مهرداد غلامحسینی

95149056

پاییز 1398

استاد : دکتر شکیبا

طراحی پایگاه داده ی سیستم مدیریت هتل بر اساس طراحی نمودار

Entity-Relationship و تبدیل آن به Data Base Schema

و تبدیل جدول های پایگاه داده به

فرم های نرمال

Normalization of Data Base

: Normalization

فرایندی است که با تفکیک کردن جداول باعث میشود داده ها در جداول پایگاه داده طوری سازمان دهی شوند که از به وجود آمدن داده های تکراری (data redundancy) و insert , update , delete anomaly جلوگیری میکند . به عنوان مثال با توجه به جدول زیر

rollno	name	branch	hod	office_tel
401	Akon	CSE	Mr. X	53337
402	Bkon	CSE	Mr. X	53337
403	Ckon	CSE	Mr. X	53337
404	Dkon	CSE	Mr. X	53337

اگر بخواهیم یک دانشجوی جدید را به جدول بالا اضافه کنیم تا زمانی که دانشجوی جدید شاخه تحصیلی خود را انتخاب نکرده باشد نمیتوانیم این دانشجو را به جدول بالا اضافه کنیم و یا باید دانشجو را اضافه کرده و برای اطلاعات مربوط به شاخه تحصیلی او مقدار NULL را اختصاص دهیم .

و یا اگر بخواهیم 100 دانشجو از یک شاخه تحصیلی را به جدول اضافه کنیم

باید اطلاعات تکراری مربوط به شاخه تحصیلی را برای هر کدام از آن 100 دانشجو تکرار کنیم. این سناریو ها insertion anomaly نام دارند.

و یا اگر MR.X از آن دانشکده برود و یا مدیر گروه یک رشته تحصیلی در این جدول تغییر کند ما باید برای همه دانشجویان آن رشته تحصیلی مقدار ستون مربوط به نام مدیر گروه را به روز رسانی کنیم که اگر تصادفاً برای یکی از ردیف ها این مقدار به روز رسانی نشود باعث به وجود آمدن مشکلاتی از قبیل data inconsistency خواهد شد. این سناریو update anomaly نام دارد.

قوانین مربوط به نرمال سازی در فرم های نرمال زیر دسته بندی میشوند :

1. First Normal Form
2. Second Normal Form
3. Third Normal Form
4. BCNF
5. Fourth Normal Form

فرم نرمال 1NF :

هر کدام از ستون ها (Attribute) های جدول برای هر ردیف باید تک مقداری یا مقدار Atomic داشته باشند. و نباید برای یک Attribute چندین مقدار ذخیره شود.

مقدار ذخیره شده برای هر Attribute در هر ردیف از جدول باید در محدوده ی مقادیر مجاز از نوع داده ی مربوط به همان Attribute باشد.

هر ستون از جدول باید دارای یک نام منحصر به فرد باشد.

فرم نرمال 2NF:

جدول باید در فرم 1NF باشد .

در جدول نباید هیچ وابستگی جزیی (partial dependency) وجود داشته باشد .

منظور از وابستگی در این جا وابستگی تابعی است . functional dependency

و وابستگی جزیی به این معناست که اگر ما در جدول خود چند ستون را به عنوان primary key در نظر بگیریم و بخواهیم که هر ستون دیگر به مجموعه ستون های primary key وابسته باشد . ولی ستون هایی وجود داشته باشند که فقط به بخشی از composite primary key ما وابسته باشند . در این حالت ما partial dependency خواهیم داشت .

فرم نرمال 3NF :

جدول باید در فرم 2NF باشد .

جدول نباید شامل Transitive Dependency باشد . به طوری که نباید یک non-prime Attribute به یک prime Attribute وابسته باشد .

و این به این معناست که جدول ما نباید شامل ستون هایی شود که به ستون هایی وابسته اند که ان ستون ها Primary key و یا بخشی از primary key نباشند .

فرم نرمال BCNF:

جدول مورد نظر باید در فرم 3NF باشد .

و برای هر وابستگی $A \rightarrow B$ باید یک super key باشد . به عبارت دیگر برای هر وابستگی $A \rightarrow B$ اگر B یک prime Attribute باشد (یعنی بخشی از کلید اصلی باشد) ان وقت A نمیتواند یک non-prime Attribute باشد .

فرم نرمال 4NF :

جدول باید در فرم BCNF باشد .

هیچ multi-value Dependency نباید در جدول وجود داشته باشد .

Multi-value dependency به معنای این است که برای هر وابستگی $A \rightarrow B$ برای هر مقدار از A چندین مقدار از B وجود داشته باشد . و برای اینکه در یک جدول multi-value dependency وجود داشته باشد آن جدول باید حد اقل دارای 3 ستون باشد .

و اگر در جدول به صورت $R(A,B,C)$ اگر بین A و B یک multi-value dependency وجود داشته باشد بین B و C نباید هیچ وابستگی وجود داشته باشد .

راه حل کلی تبدیل یک جدول به هر کدام از فرم های نرمال با توجه به قوانین گفته شده تفکیک آن جدول به جدول های دیگر است .

طراحی پایگاه داده مربوط به سیستم مدیریت هتل :

برای پایگاه داده ی سیستم مورد نظر در ابتدا 5 – Entity اصلی سیستم که عبارت اند از

پرداخت ها – اتاق ها – کارمندان – مشتری ها – نظرات را در نظر گرفته و رابطه (Relation) های میان آن ها را استخراج نموده و در نمودار ER رسم میکنیم . سپس برای هر کدام از Entity های مورد نظر Attribute های آن ها را رسم کرده و نوع هر کدام از آن ها را (composite- derived- multi-value) را مشخص میکنیم .

سپس موجودیت های ضعیف را مشخص کرده و همچنین primary key هر کدام از Entity ها را مشخص میکنیم .

از انجایی که در جداول ما هیچ جدولی تمام Attribute های آن به یک جدول دیگر وابسته نیست (تعریف Weak Entity) پس هیچ Weak Entity در طرح ما وجود ندارد. و در آخر محدوده ی Cardinality را بین Entity ها مشخص میکنیم .

حال نوبت به تبدیل نمودار ER به Data Base Schema میرسد . بعد از آماده شدن طرح اولیه Data Base Schema با توجه به قوانین گفته شده برای نرمال سازی جدول ها برای هر جدول شروع به نرمال سازی آن میکنیم . و پس از اتمام نرمال سازی و مشخص کردن Foreign key ها شروع به نوشتن کد های SQL برای سیستم پایگاه داده ی مورد نظر میکنیم .