

درس ۲ :

Database Normalization

نرمال سازی پایگاه داده

تعریف نرمال سازی

نرمال سازی داده ها : عبارتست از فرآیند پالایش ساختار پایگاه داده به منظور افزایش سرعت دسترسی به داده ها و یکپارچگی پایگاه داده.

ایجاد یک پایگاه داده مستلزم شناخت ساختار مورد نیاز و ایجاد جدول ها و فیلدهای لازم است. ساختار مناسب میتواند منجر به افزایش سرعت گردد؛ مثلا اگر همه اطلاعات مرتبط با هم در یک جدول قرار داده شوند برنامه میتواند با مراجعه به یک جدول مشخص ، همه اطلاعات مورد نیازش را پیدا کند.

ساختار یک پایگاه داده می تواند با هدف تامین یکپارچگی داده ها نیز طراحی شود. پایگاه داده یک فروشگاه را در نظر بگیرید ، مثلا اگر تمام فاکتورها را در یک جدول و مشخصات مشتریان در جدول دیگر قرار داده شود، در این حالت سرعت دسترسی به داده ها کاهش می یابد، زیرا برای یافتن مجموعه اطلاعات مربوط به یک فاکتور پایگاه داده باید اطلاعات را از دو جدول استخراج کرد و این فرآیند به مراتب کندتر از وضعیت قبلی است.

طراحی یک پایگاه داده نرمال مستلزم برقراری موازنه بین سرعت و یکپارچگی است.

نرمال سازی روشی برای طراحی جداول پایگاه داده است و داده ها را به روشی ساماندهی می کند که باعث کاهش افزونگی داده و رفع مشکلات ساختاری شود. هدف از نرمال سازی حذف افزونگی داده و باقی نگهداشتن وابستگی بین داده های مرتبط است.

سه مورد از رایج ترین شکل نرمال سازی، فرم نرمال اول (1NF) ، فرم نرمال دوم (2NF)، و فرم نرمال سوم (3NF) در جدول زیر توضیح داده شده است.

سطح	قاعده یا قانون
1NF	یک جدول در فرم اول نرمال (1NF) است اگر و فقط اگر فاقد گروه داده تکرار شونده باشد.
2NF	یک جدول در فرم دوم نرمال (2NF) است اگر اولاً 1NF باشد و ثانياً کلیه ستون های غیرکلید با کلید اصلی وابستگی تابعی کامل داشته باشند. زمانی که تمام ویژگی های غیرکلیدی آن کاملاً به داده اصلی وابسته هستند.

3NF	یک جدول در فرم سوم نرمال (3NF) است وقتی 3NF باشد و تمام ویژگی‌های آن مستقیماً به داده اصلی بستگی داشته باشد. با حذف فیلدهائی که وابستگی مستقیم با کلید ندارند به فرم سوم نرمال می‌رسیم.
-----	---

قواعد نرمال سازی (مثال عملی)

قاعده اول: هر دسته از اطلاعات مرتبط باید در یک جدول جداگانه قرار گیرند (2FN)

قاعده دوم: حذف گروه‌های تکراری (1FN)

قاعده سوم: انتخاب کلید

قاعده چهارم: حذف ستون‌هایی که به کلید اولیه وابسته نیستند. (3FN)

قاعده پنجم: تعداد فیلدهای جدول را بسته به نیاز برنامه کاربردی کاهش می‌دهیم

قاعده ششم: افزودن فیلدهای لازم

قاعده هفتم: بررسی روابط بین جداول

مثال: نرمال سازی اطلاعات پرسنل شرکت (سیستم حقوق و دستمزد)

اطلاعات پرسنل ممکن است شامل فیلدهای زیر باشد.

شماره پرسنلی، نام، نام خانوادگی، شماره شناسنامه، محل صدور شناسنامه، تاریخ تولد، کد ملی، وضعیت تاهل، نام همسر، فامیلی همسر، تعداد فرزندان، نام فرزند ۱، تاریخ تولد فرزند ۱، نام فرزند ۲، تاریخ تولد فرزند ۲، نام فرزند ۳، تاریخ تولد فرزند ۳، ...، میزان تحصیلات، رشته تحصیلی، آدرس منزل، شماره تلفن منزل، شروع کار، پایان کار، شماره بیمه، گروه شغلی، حقوق پایه، شغل، محل کار، تلفن، آدرس محل کار، نام پروژه، موضوع پروژه، نرخ پروژه، نرخ اضافه کار، نرخ غیبت، مالیات، بیمه، شماره حساب و ...)

- Employee: (Emp_no, Fname, Lname, C-id, birthdate, melli code, marry, wife name, wife lname, child_num, chname, chbirthdate, tahsilat, course, address, phon, begindate, enddate, bime-no, base_salary, job, dept_name, dept_phon, dept_address, projname, projsubject, proj_price, Overtime, absent, tax, bime_no, account)

قاعده اول: هر دسته از اطلاعات مرتبط باید در یک جدول جداگانه قرار گیرند.

ما میتوانیم برای اعمال قاعده اول، داده‌های پراکنده را حذف کنیم و درجداول مستقل قرار می‌دهیم.

- Project : (pname, subject, price)
- Department : (dept-name, dept-address)
- Employee : (Emp_no, Fname, Lname, C-id, birthdate, melli code, marry, wife name, wife lname, child_num, chname, chbirthdate, Course, grade, phon, address, begindate, enddate, bime-no, base_salary, job, Overtime, absent, tax, bime, account)

قاعده دوم : حذف گروه های تکراری

ممکن است کارکنان شرکت بیش از یک فرزند داشته باشند و تمام اطلاعات پرسنل به ازای هر فرزند تکرار می شود، بهتر است این فیلدهای تکراری از جدول اولیه حذف شوند و در یک جدول جداگانه قرار داده شود.

این کار سبب می شود فضای ذخیره سازی هدر نرود، ذخیره سازی و بازیابی اطلاعات ردیف های فرزندان ساده تر شود.

در این صورت فیلد emp_no برای برقراری ارتباط بین این دو جدول استفاده می شود* و جدول Child به صورت زیر خواهد بود:

- Child (Emp_no, chname, chbirthdate)
- Employee: (Emp_no, Fname, Lname, C-id, mellicode, birthdate, marry, wife name, wife lname, course, grade, phon, address, begindate, enddate, bime-no, job, Overtime, absent, tax, bime, account)

قاعده سوم : انتخاب کلید

حال برای جداول کلیدهای اولی و ثانویه و خارجی تعریف می کنیم

کلید اولیه (Primery Key): روی فیلد منحصر به فرد. مثل شماره پرسنلی

یکی از فیلدهای جدول که یکتا می باشد و هیچگاه طول صفر ندارند را بعنوان کلید اولیه انتخاب می کنیم. مثل شماره پرسنلی

سعی می کنیم این کلید روی یکی از اطلاعات طبیعی باشد. مثلاً روی شماره شناسنامه در جدول مشخصات. در صورتی که در جدول فیلد منحصر به فرد نیست یک فیلد اضافه می کنیم و آن را بعنوان کلید اصلی انتخاب می کنیم.

کلید ثانویه (Secondary Key): معمولاً جستجو و مرتب کردن اطلاعات روی فیلدی غیر از کلید اصلی هم می باشد که احتمالاً منحصر به فرد نیست. مثل نام خانوادگی

کلید خارجی (Foreign key): برای ارتباط جداول باید از این کلید استفاده نمود.

- Project : (**proj no**, pname, subject, price)
- Department(**dept no**, dept-name, dept-address)
- Child : (**child no**, Emp-no, chname, chbirthdate)
- Employee: (**Emp no** , Fname ,Lname, C-id ,mellicode ,birthdate, marry , wife name, wife lname, course, grade, phon , address, begindate, enddate, bime-no, job , base_salary, Overtime, absent, tax, bime, account)
- Datald:(**Emp no** , **proj no**, **dept no**)

نکته: در این مثال کلید های اصلی زیر خط دار و bold نمایش داده شده اند و کلید های خارجی رنگی نمایش داده شده اند.

قاعده چهارم: حذف ستون هایی که به کلید اولیه وابسته نیستند.

هر فیلدی که کاملاً بیانگر یکی از ویژگیهای کلید اولیه نیست باید به یک جدول مستقل منتقل شود.

مثلاً میزان حقوق هیچگونه وابستگی به شخص کارمند ندارد و به گروه شغلی افراد بستگی دارد پس باید در جدول مستقل قرار گیرد.

- Datald:(**Emp no** , **group no**, **proj no**, **dept no**)
- Project : (**proj no**, pname, subject, price)
- Department(**dept no**, dept-name, dept-address)
- Child : (**child no**, Emp-no, chname, chbirthdate)
- Group: (**group no**, base_salary)
- Wife: (**Emp-no**, wife name, wife lname)
- Employee: (**Emp no** , Fname ,Lname, C-id ,mellicode ,birthdate, marry , course, grade, phon , address, begindate, enddate, bime-no , job, Overtime, absent, tax, bime, account)

قاعده پنجم: تعداد فیلدهای جدول را بسته به نیاز برنامه کاربردی کاهش می دهیم.

تا آنجا که ممکن است تعداد فیلدهای جدول را بسته به نیاز برنامه کاربردی کاهش می دهیم.

همانطور که می بیند بقیه اطلاعات مربوط به مشخصات پرسنل می باشد ولی از آنجاییکه شامل مشخصات اداری و مشخصات شخصی می باشد و ما همزمان به هر دو دسته اطلاعات نیاز نداریم بهتر است آن را به دو جدول Spec (مشخصات شخصی) و Works (مشخصات اداری) تبدیل نماییم، تا بسته به نیاز، گزارشات با سرعت بیشتری تهیه شود.

- Spec: (Emp_no, Fname, Lname, C-id, mellicode, birthdate, marry, course, grade, phon, address)
- Work: (Emp_no, begindate, enddate, bime-no, job, Overtime, absent, tax, bime, account)

قاعده ششم: افزودن فیلدهای لازم

ممکن است تمام اطلاعاتی که یک برنامه کاربردی به آن نیاز دارد، در مراحل اولیه موجود و یا لازم نباشد، بنابراین در مراحل بعدی کار اضافه می شود. مانند تلفن همراه شخص و

قاعده هفتم: بررسی روابط بین جداول

در انتها روابط بین جداول را بررسی نمایید. با شناسایی همه روابط، بین آنها اطمینان حاصل نمایید که تلفیق جداول امکان پذیر است.

بنابراین طراح باید به دنبال مناسب ترین جداول برای پایگاه داده باشد. جداول باید طوری تعریف شوند که انعطاف پذیری را برای نیازهای بعدی فراهم کنند و باید باهم مرتبط باشند تا امکان تلفیق آنها با هم امکان پذیر باشد.

پس از اعمال قواعد فوق جداول به صورت زیر خواهند بود:

- DataId: (Emp_no, group_no, proj_no, dept_no)

جدول مشخصات شخصی کارمند

- Spec: (Emp_no, Fname, Lname, C-id, mellicode, birthdate, marry, course, grade, phon, address)

جدول مشخصات اداری کارمند

- Salary : (Emp no ,begindate, enddate, bime-no, job, Overtime, absent, tax, bime, account)

جدول مشخصات همسر

- Wife: (Emp-no, wife name, wife lname)

جدول مشخصات فرزندان

- Child : (child no, Emp-no, chname, chbirthdate)

جدول پروژه های انجام شده در شرکت

- Project : (proj no, pname, subject, price)

جدول مشخصات محل کار

- Department(dept no, dept-name, dept-address)

جدول گروه شغلی

- Group: (group no, base_salary)

تمرین:

- نرمالایز و رسم دیاگرام سیستم کتابخانه

Diagram

