

درس : آزمایشگاه پایگاه داده

آموزش کاربردی SQL server

استاد : هادی امیر احمدی

مقدمه ای بر سیستم پایگاه داده رابطه‌ای :

یک سیستم پایگاه داده مجموعه ای از اجزای نرم افزاری و پایگاه داده های مختلف است.

پایگاه داده از بخش های زیر تشکیل شده است :

۱- **برنامه های کاربردی:** برنامه های کاربردی نرم افزار های خاصی هستند که توسط شرکت های کامپیوتری یا کاربران تولید شده است .

۲- **سرویس گیرنده:** نرم افزار های خاصی هستند که توسط شرکت های متخصص پایگاه داده طراحی شده و کاربران به وسیله اجزای سرویس گیرنده به داده های پایگاه داده دسترسی پیدا می کنند .

۳- **Server:** برای مدیریت داده های ذخیره شده روی پایگاه داده کاربرد دارد .

پایگاه داده از دو دید بررسی می شوند :

۱- **از دید کاربران**

۲- **از دید سیستم پایگاه داده**

از دید کاربران پایگاه داده مجموعه ای از داده های منطقی مرتبط به هم هستند که با هم ارتباط برقرار می کنند .

از دید سیستم پایگاه داده پایگاه داده دنباله ای از بایت ها است که روی دیسک ذخیره می شوند .

هر سیستم پایگاه داده دارای یک سری ویژگی ها هستند :

۱- **واسط کاربری:** که شامل فرم ها و منو ها و.... است.

۲- **استقلال فیزیکی داده ها:** یعنی برنامه های کاربردی به ساختار فیزیکی پایگاه داده سیستم وابسته نیستند .

۳- **استقلال منطقی داده ها:** با هر تغییر در ساختار فایل برنامه های کاربردی دچار تغییر می شود .

۴- **بهینه سازی:** یعنی هر پایگاه داده باید دارای یک سری بهینه ساز باشند تا از طریق پرس و جو های موجود و ممکن بهترین را انتخاب کند .

۵- **جامعیت داده ها:** یعنی داده های ناسازگار را تشخیص داده و از ذخیره آن ها جلوگیری کند .

۶- **کنترل هم رندی:** از پروسس هایی که به طور هم زمان اجرا می شوند جلوگیری کند (یعنی برنامه های کاربردی مختلف داده های یکسانی را بروز کنند ، تغییرات باید به صورت کنترل شده باشند) .

مدل داده ای : مجموعه ایی از مفاهیم روابط بین آن ها و محدودیت هایی است که برای نمایش داده های یک مسئله واقعی استفاده می شود .

مثال : شرکت را در نظر بگیرید که دارای یک سری کارمند است و هر کارمند متعلق به یک اداره است و هر کارمند می تواند در چند پروژه شرکت کند .

جدول دپارتمنت :

Dept-num	Dept-name	Dept-adress
----------	-----------	-------------

۱۰۰	اداره اموال	گوهردشت
۱۰۱	حسابداری	حصارک
۱۰۲	واحد تشخیص	باغستان
۱۰۳	تولید منیزیم	گوهردشت
۱۰۴	تولید مس	گوهردشت
۱۰۵	تولید آهن	مهرشهر

جدول کارمندها

Emp-num	Emp-firstn	Emp-lastn	Dept-num
۳۲۵۱۴	علی	محمدی	100
۳۲۵۱۵	حسین	مرادی	100
۳۲۵۱۶	رضا	حسینی	12

یک زبان مجموعه گرا است یعنی تعداد زیادی سطر و ستون در یک جدول وجود دارند

Sql دو تا زبان دارد :

Dml (data manipulating language) : زبان دستکاری داده ها است که با آن می توان داده ها را

دستکاری کرد شامل دستورات select-insert-update-delete.

DDL(Data Definition language): برای ساخت یک جدول Create و ارتباط جداول با هم aiter

و... به کار می رود .

آشنایی با برخی مفاهیم پایگاه داده :

۱- **نرمال سازی**: یافتن وابستگی بین ستون های یک جدول در صورت پیدا شدن جدول به دو جدول دیگر

تقسیم می شود و این پروسه تا زمانی ادامه دارد که وابستگی بین دو جدول وجود نداشته باشد.

۲- وابستگی تابعی: به میزان ازای هر مقدار از ستون فقط یک مقدار منحصر به فرد در ستون دیگر وجود داشته باشد ، مثلا در ازای شماره ۵ فقط یکی پیدا شود.

۳- وابستگی چند مقداری: به ازای یک مقدار از یک ستون مجموعه ای از مقادیر داشته باشیم.

۴- خصوصیت مرکب: چند مقدار اتمیک یا چند مقداری است .

۵- خصوصیت مشتق: یعنی یک مقدار با محاسبه مقادیر ستون های دیگر به دست می آید ، مثل معدل دانشجویان کلاس .

۶- نرمال سازی سطح یک: در هر ستون جدول یک مقدار فقط می تواند ذخیره شود .

Emp no	Project no	job	Date
2530	103	آنالیزور	۹۸/۸/۲۵

Emp no	Project no	Job	Date
2530	۱	آنالیزور	۹۸/۸/۲۵
2530	۳	آنالیزور	۹۸/۸/۲۵

این جدول نرمال سازی سطح یک است .

نرمال سازی سطح دو: جدول نرمال سازی سطح دو است که اولاً نرمال سازی سطح یک باشد و ثانياً هیچ ستون غیرکلیدی به کلید اصلی وابسته نباشد.

نکته : هر جدولی کلید اصلی آن یک ستون باشد همیشه در سطح نرمال سازی سطح دوم هستش.

نرمال سازی سطح سوم: جدولی سطح سوم است که اولاً سطح دوم باشد دوماً هیچ وابستگی تابعی بین ستون های غیرکلیدی وجود نداشته باشد

Emp-no	Emp -fname	emp-lname	Dept-num	Dept-name
--------	------------	-----------	----------	-----------

2530	علی	احمدی	۱۰۰	مختصات
------	-----	-------	-----	--------

اجزای Sql

زبان Database engine دارای یک سری ویژگی ها هست مانند زبان های برنامه نویسی دیگه هستند مثل مقادیر حرفی شناسه ها جداکننده ها و کلمات کلیدی . در این مرحله به یک نمونه از sql وصل می شویم .

ایجاد پایگاه داده و جدول

Sql یک پایگاه داده رابطه ای که دارای چندین نمونه است ، هر نمونه می تواند شامل چندین پایگاه داده رابطه ای باشد و هر پایگاه داده می تواند شامل چندین جدول باشد ، هر جدول نیز می تواند یک رابطه منطقی با جدول های دیگر داشته باشد و هر جدول شامل * یا چندین رکورد است با استفاده از نرم افزار SQL Server Management Studio می توانید که به پایگاه داده های موجود در برنامه دسترسی پیدا کنید . هر چیزی که داخل پایگاه داده وجود دارد مانند جدول view یا index ذخیره شده و.... اشیا پایگاه داده هستند.

مثال : برای آن که مفهوم پایگاه داده جدول و رکورد با توجه به شبیه سازی یک مثال می زنیم

یک پایگاه داده به نام انتشارات گوهر داریم که در پایگاه داده انتشارات گوهر سه فیلد داریم :

کتاب	مشتری	سفارش
------	-------	-------

کد کتاب	نام کتاب	موجودی	قیمت
۲۵	ویندوز	۵	۲۵۰۰۰
۲۶	پاسکال	۱۰	۳۵۰۰۰

کد مشتری	نام مشتری	آدرس	تلفن
۴	علی	کرج	۲۲۲۲
۵	حمید	تهران	۱۱۱۱

X	کد کتاب	کد مشتری	تعداد
	۱	۴	۶
	۲	۵	۷

یک رابطه منطقی بین جدول مشتری و جدول سفارش و یک رابطه منطقی دیگر بین جدول کتاب و جدول سفارش وجود دارد ، یعنی مثلا کد ۴ که علی است کتاب ویندوز را ۵ بار سفارش داده است .

در قسمت Database File مشخصات آن پایگاه داده را نمایش می دهد که با پسوند mdf ایجاد شده است و در قسمت insialsize اندازه فایل که قسمت بالایی ۲ و پایینی ۱ است را نشان می دهد .

مقادیر پیش فرض در زمان تعریف پایگاه داده از تنظیمات مربوط به پایگاه داده مدل واقع در

قسمت `system database` دیده می شود ، برای تغییرات این خصوصیات می توانید روی پایگاه داده کلیک راست کرده و گزینه `Properties` را انتخاب می کنیم

در پایگاه داده رابطه ای داده در جدول ذخیره می شوند ، هر جدول در پایگاه داده به صورت پیش فرض $(2*2)$ که شامل سطر و ستون هستند ، هر سطر یک رکورد و هر ستون یک فیلد نامیده می شود ، داخل ستون ها تعریف می شود و داده در رکورد ها و ردیف ها ذخیره می شود برای ساختن جدول پایه اول اسم آن را درست کنیم .

تذکره ۱: در صورتی که نام جدول را عوض کنیم ، بعد از ذخیره سازی نام جدول از شما پرسیده می شود . **تذکره ۲:** هر نوع داده ای مقداری از فضای ذخیره سازی را اشغال می کند و هرچه فضای ذخیره سازی کم تر باشد حجم فایل هم کمتر می شود ، و سرعت پردازش (عملیات ها) بیشتر و سریع تر می شود **تذکره ۳:** `nvarchar` برای رشته های متغیر استفاده می شود .

Allow nulls: یعنی به صورت پیش فرض می توان چیزی را وارد نکنیم (فیلد را خالی بگذاریم). عدد n تا ۸۰۰ اگر داخل پرانتز باشد هر کاراکتر ۱ بایت را اشغال می کند (فکرکنم منظورش `nvarchar` است) . `nchar` : میزان `nchar` دو بایت را اشغال می کند و به ازای هر کاراکتر بین ۱ تا ۴۰۰ تعیین بشه .

`Tiny` : ۱ , `big int` : ۸ و `int` : $2^{31} - 1$ تا 2^n

نکته : ابتدا باید جدول را بسازیم و بعد اطلاعات را وارد کنیم .

اگر بخواهیم تمامی اطلاعات را در یک جدول داشته باشیم افزونگی ایجاد می شود .

نکته : پایگاه داده رابطه ای توسط رابطه ای که بین جدول تعیین می شود رابطه پیدا می کند این رابطه کمک می کند تا از افزونگی داده و تکرار شدن آن ها جلوگیری شود . برای تعریف رابطه از کلید های اصلی (`primary`) یا خارجی (`foreign`) استفاده می شود . کلید اصلی یک شاخص

یکتا که یک ردیف را مشخص می کند برای مثال کد کتاب در جدول کتاب یک کلید اصلی است، یعنی یک کد مخصوص دارد که دیگر کتاب ها آن را ندارد .

نکته : در بعضی از حالات ممکن است که یک جدول دارای دو کلید اصلی باشد در مثال ها کد کتاب از جدول Book و کد مشتری از جدول مشتری یکتا هستند . در این مثال یک رابطه بین فیلد کد کتاب از جدول کتاب با فیلد کد کتاب در جدول سفارش وجود دارد ، بنابراین فیلد BID از جدول سفارش یک کلید خارجی است ، پس یک فیلد که در یک جدول کلید خارجی است در جدول دیگر کلید اصلی است .

قواعد جامعیت داده (Data Intergity rules) : در یک پایگاه داده برای درستی ، دقت و

سازگاری داده یعنی جامعیت داده می توان یک یا چند قانون تعیین کرد .

۱- Null : زمانی که این محدودیت اعمال شود یک فیلد حتما باید داده داشته باشد .

۲- Check : توسط این محدودیت می توانید تعیین کنید که مقدار یک فیلد معتبر است یا نیست
مثال : سن کارمندان یک شرکت بررسی شود ۲۲ تا ۶۰ سال باشد .

۳- Unique : این شرط تعیین می کند که مقدار یک فیلد در هیچ کدام از فیلد های دیگر در یک ستون قابل تکرار نیست .

۴- Primary key : این محدودیت باعث می شود که ، تا مقدار یک فیلد در فیلد های دیگر تکرار نشود . برای مثال : کتابی با کد ۱ دیگر نباید در هیچ کدام از ردیف ها تکرار شود .

۵- Foreign key : این قید به یک مقدار یکتا در یک جدول دیگر در همان پایگاه داده اشاره دارد
مانند فیلد کد کتاب یا مشتری در یک جدول سفارش

Book > ID

Customer > ID

Order (BID,CID)

سعی کنید قواعد جامعیت داده را در زمان طراحی جدول تعیین کنید .

می خواهیم با check یک محدودیت برای جدول سفارش درست کنیم تا مشتری نتواند ۱۰۰ کتاب بیشتر انتخاب کند. ($Qty \leq 100$) .

بعد از این که ارتباط جداول Customer با Order و Book با Order وارد کردید اگر یکی از جداول Full text مانند Customer اطلاعات وارد باشد خطای جامعیت داده Customer به نام fk-order رخ می دهد که همان خطای کلید خارجی است که آماده نیست در ادامه توضیح داده شده که این خطا در پایگاه داده گوهر جدول Customer ID رخ داده است .

تذکر : با استفاده از قواعد جامعیت داده توانستیم محدودیت هایی در مورد داده ها وارد کنیم تا درستی ، دقت و سازگاری در اطلاعات پایگاه داده حفظ شود .

نمودار رابطه پایگاه داده (base diagram) :

در کنسول object explorer به کمک شاخه Data می توانید چندین نمودار پایگاه داده ایجاد نمایید که به کمک این نمودار ها رابطه بین جداول بیشتر مشخص می شود .

روی شاخه Data base diagram بر روی گزینه New data base diagram کلیک کنید در این جدول لیست پایگاه داده ایجاد شده است ، می خواهیم سه جدول Book و Customer و Order ایجاد کنیم با استفاده از این نمودار ساختار پایگاه داده و رابطه بین جدول های آن واضح می شود . با راست کلیک بر روی نمودار و گزینه Add table می توانیم یک جدول را به آن نمودار اضافه می کنیم . با کلیک بر روی هر جدول و گزینه Delete هر جدول را حذف می کنیم .

ایجاد فایل پرس و جو

روش ایجاد پرس و جو در پایگاه داده

مواردی که در پایگاه داده کاربرد زیادی دارد انجام پرس و جو یا Query است .

در نوار ابزار انواع پرس و جو گزینه **new query** که شما می توانید با دستورات **Tsql** دستورات را وارد کنید .

با **Use gohar** تعیین می کنیم که با پایگاه داده گوهر کار می کنیم .

Select * from book یعنی تمامی رکورد های **book** را می آورد .

برای اجرای دستورات روی دکمه **execute** کلیک کنید یا **f5** را فشار می دهیم .

باید دقت کرد پایگاه داده گوهر انتخاب شود .

نکته : در صورتی که جواب جست و جوی شما زیاد باشد و بخواهید تعداد محدودی از ابتدای جست و جوی را انتخاب کنید از عبارت **top** استفاده می کنیم .

مثال :

Select Top (5) * from Book

با استفاده از دستور بالا فقط ۵ رکورد اول نشان داده می شود .

Select Top (20) Percent * from book

با استفاده از دستور بالا ۲۰ درصد رکورد ها را نشان می دهد .

اگر از جدولی دیگر بخواهیم استفاده کنیم نام جدول را به جای **book** به نام دیگری تغییر می دهیم

اگر بخواهیم از کلمات شبیه کلمات استفاده شده (رزرز شده استفاده نکنیم) استفاده کنیم باید آن را داخل کوتیشن بگذاریم .

تذکر : بعد از دستور **select** می خواهیم تمامی فیلد های جدول نشان داده شود .

Select name , tel from Customer

از جدول Customer فقط فیلدهای name, tel نشان داده می شود بنابراین بعد از دستور select فیلدهایی که می خواهید در پرس و جو (query) نمایش داده شوند را وارد کنید . بین فیلدها از (، یا کاما) استفاده می کنیم .

نام جدول Select * from

در این مثال : دستور Tsql را در دو خط وارد کردیم در صورتی که دستوری در یک خط یا چندین خط وارد کنید تفاوتی ندارد .

در دستور Tsql وارد شده Use gohar را وارد نکردیم چون قبلا انتخاب شده و نیازی به انتخاب ندارد .

تذکر : چون جدول را در جلسات قبلی طراحی کردیم درست تمامی فیلدهای جدول را به خوبی می دانیم ولی ممکن است بخواهیم با جدولی کار کنیم که از فیلدهای آن اطلاع نداریم برای مشاهده فیلدهای جدول، نام جدول select*from راه دیگر استفاده از پنجره object explorer است. در پنجره object explorer کلیک کنید و بعد در پایگاه داده فولدر table کلیک کنید و بعد در کلیک جمع table گزینه columns کلیک کنید .

تذکر: در صورتی که در جواب پرس و جو یا query رکورد های تکراری داشته باشید با استفاده از دستور distinct استفاده می کنیم بعد از عبارت select تعیین میکنید که هر رکورد فقط یکبار نمایش شود.

Select CID from orders

مثال : در صورتی که بخواهیم کل کتابهای سفارش داده شده را مشاهده کنیم از دستور بالا استفاده میکنیم.

اگر یک یا چند مشتری از یک کتاب چندین بار سفارش داده باشند باعث می شود در نتیجه جستجو چندین بار مشاهده شود ولی با استفاده از دستور زیر هر کد کتاب فقط یکبار نمایش داده شود

Select distinct CID from orders

با استفاده از دستور **where** میتوانیم شرطی برای فیلد ها بگذاریم تا تمامی فیلد ها نمایش داده شود بلکه فرایندهایی که خواستیم نشان داده شود.

توسط فیلد پرس و جوی داده شده فیلد های **name** و **qty** رکورد های نمایش داده میشود که مقدار **qty** آنها بیش از ۳۱۰۰ باشد.

Select name qty from book where qty > 3100

“	“	“	“	<=
“	“	“	“	>=
“	“	“	“	!= not (qty=3100)
“	“	“	“	Not
“	“	“	“	Qty between 10 and 20
“	“	“	“	In (2,3,5)
“	“	“	“	Like

Name not like n 'ش';

Name not like n 'ب';

با استفاده از دستور **like** میتوانیم شرطی تعیین کنیم که یک دستور حاوی یک شرط باشد x

Use gohar

Select name qty

From book

Where name like n 'ک';

Select name from book where name not like n 'ک';

Select name from book where name like 'ب';

Select name qty

from book

where qty between 2000 and x 5000

order by

با استفاده از دستور `order by` می توانید نتیجه پرس و جو از جدول بر حسب یک فیلد مرتب نمایید

Use gohar

Select from book order by name , qty

نتیجه این پرس و جو یک لیست مرتب شده از موجودی اولیه کتاب ها که شامل نام تمامی کتاب ها و موجودی اولیه می باشد .

Name	Qty
ویندوز	۳۰۱۰
شبکه	۳۲۰۰
دیتا بیس	۵۰۰۰

Use gohar select name qty from book desc

این پرس و جو لیست کتاب ها را بر اساس نزولی مرتب می کند

تذکر : پرس و جو های مطرح شده تا اینجا وابسته به یک جدول بودند ولی در یک مثال واقعی یا دیتا بیس های واقعی اکثرا نیاز داریم اطلاعات بک با چند جدول را استخراج کنیم.

مثال : می خواهیم لیست سفارشات را بر اساس نام کتاب مشاهده کنیم : بعد از عبارت from نام جدولی که می خواهیم جستجو روی آنها ایجاد شود می نویسیم سپس با استفاده از دستور where ارتباط دو جدول را می نویسیم

Use gohar

Select book name orders

from book orders

where book .id = orders bid and cid = 1