

بسمه تعالی

جزوهء آموزشی

پایگاه داده ها

مدرس :

مهندس علی محمدزاده

دانشگاه علمی کاربردی

سازمان مدیریت صنعتی

فصل اول

تعریف پایگاه داده :

یک سیستم یکپارچه کامپیوتری که داده ای کاربران و فراداده ها را در آن ذخیره می کند و کاربران می توانند به صورت اشتراکی از آن استفاده کنند.

فرا داده :

اطلاعات اضافی در مورد داده ها می باشد .

جنسیت	تلفن	آدرس	کد ملی	ش شناسنامه	شماره دانشجویی	نام خانوادگی	نام
مرد	۰۹۱۴	ارومیه	۲۸۰۱۳۴۰۰۰	۱۲۳	۸۹۳۶۰۶۰	قدیم خانی	حیدر

رکورد : ردیف ، سطر

فیلد : ستون

تاریفچه پایگاه داده :

قبل از ورود کامپیوتر به عرصه ذخیره سازی و نگهداری داده ، اطلاعات به صورت پرونده در مکانهای مشخص ذخیره می شدند که مشکلاتی از قبیل نگهداری ، جستجو و ویرایش داشتند همچنین امنیت پایینی در مقابل بلایای طبیعی از قبیل آتش سوزی و زلزله و ... داشتند از این رو با ورود کامپیوتر اطلاعات به صورت سیستم فایل در کامپیوتر ذخیره شدند سیستم فایل نیز دارای مشکلاتی بود که بدلیل وجود این مشکلات ، جای خود را به پایگاه داده دادند .

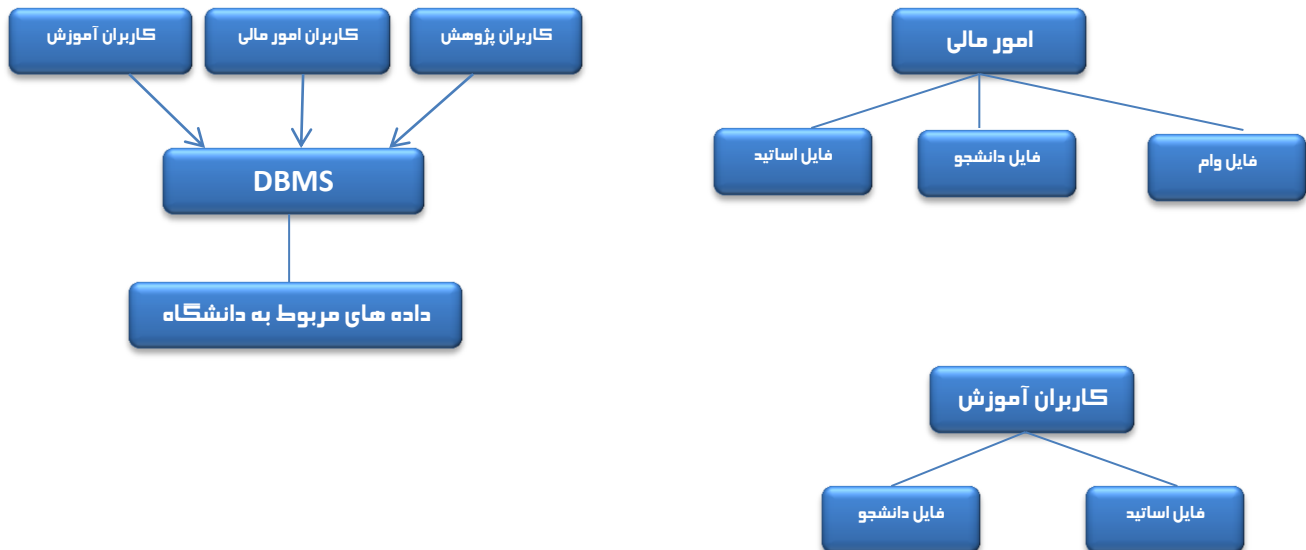
مشکلات سیستم فایل :

۱. برنامه نویسی زیاد و پیچیده بود .
۲. وابستگی داده ای یعنی اگر می خواستیم طول فیلدها را تغییر دهیم کل برنامه کاربردی باید از ابتدا نوشته می شد .
۳. وابستگی ساختاری یعنی اگر می خواستیم تعداد فیلدهایمان را تغییر دهیم کل برنامه کاربردی را باید اول ایجاد می کردیم.
۴. عدم تامین امنیت کاربران یعنی اینکه هر کاربری هر تغییری را در داده ها می توانست بدهد .
۵. افزونگی داده ها یعنی اینکه یک داده در چند محل می توانست ذخیره شود افزونگی داده خود مشکلاتی داشت از قبیل

الف) هدر رفتن فضا ب) ناسازگاری داده ها ج) ناهنجاری حذف ، درج و ویرایش

سیستم های پایگاه داده :

در سیستم های پایگاه داده بر خلاف سیستم فایل ، کاربران به صورت مستقیم به داده ها دسترسی ندارند و باید درخواست خود را به قسمتی از سیستم به نام DBMS بدهند ، این درخواست را بر روی داده ها اجرا می کند و نتیجه را به کاربر می گرداند در حقیقت DBMS سیستم اصلی مدیریت داده ها و کاربران است و مسائلی از قبیل اشتراک کاربران و امنیت داده ها را فراهم می کند .



اجراء سیستم یک پایگاه داده :

(۱) **سخت افزار** : تمامی سخت افزار هایی که به عنوان ورودی و خروجی مورد استفاده قرار می گیرند مثل هارد دیسکی که پایگاه داده روی آن قرار می گیرد و یا پردازنده ای که پایگاه داده را پردازش می کند و...

(۲) **نرم افزار** : سیستم عامل ، نرم افزار DBMS و نرم افزار کاربردی که کاربر با آنها کار می کند .

(۳) **داده** : که شامل اطلاعات کاربران و فراداده می باشد .

(۴) **کاربر**:

الف) کاربران نهایی **end user** : اشخاصی هستند که هیچ تخصصی در مورد کامپیوتر ندارند و فقط از نرافزار کاربردی استفاده می کنند .

ب) مهندسین نرم افزار : افراد متخصصی هستند که وظیفه ایجاد برنامه کاربردی را بر عهده دارند .

ج) مدیر پایگاه داده **DBA** : فرد متخصصی می باشد که وظیفه مدیریت و پشتیبانی پایگاه داده را بر عهده دارد .

د) مدیر داده **DA** : فرد غیر متخصصی می باشد که مالک اطلاعات است .

وظایف DBMS :

- (۱) تامین استقلال داده که به دو صورت استقلال فیزیکی و استقلال منطقی ممکن است .
 - استقلال فیزیکی داده یعنی اینکه تغییر در رسانه فیزیکی داده ها باعث ایجاد تغییر در برنامه کامپیوتر نشود و استقلال منطقی یعنی اینکه تغییر در شمای مفهومی باعث ایجاد تغییر در برنامه کاربردی نشود .
 - (۲) مدیریت ذخیره سازی داده ها : یعنی اینکه مهندسین نرم افزار نیازی به دانستن جزئیات ذخیره سازی نداشته باشند .
 - (۳) مدیریت امنیتی داده ها
 - (۴) تامین امکان دسترسی مشترک چندین کاربر در آن واحد به پایگاه داده
 - (۵) تامین امکان دسترسی از روشهای مختلف به پایگاه داده
 - (۶) پشتیبانی از زبان پرس و جو
 - (۷) امکان ایجاد نسخه پشتیبان
 - (۸) جامعیت داده ای : که خود از دو قسمت تشکیل می شود اعتبار داده و سازگاری داده
- اعتبار داده یعنی اینکه هیچ داده نامعتبری نمی تواند وارد شود و سازگاری داده یعنی اینکه اگر یک داده در چند محل ذخیره می شود باید مقادیر آنها یکسان باشد .

معماری پایگاه داده ها:

تعریف معماری پایگاه داده:

یکی از معروف ترین معماری های پایگاه داده ، معماری ANSI/SPARC می باشد که دارای سه سطح خارجی یا کاربر ، ادراکی یا مفهومی ، داخلی یا فیزیکی می باشد که هر کدام از این سطح ها دید متناظر خودشان را دارند .

تعریف سطح خارجی:

دید هر کاربر منفرد از سیستم پایگاه داده می باشد این سطح شامل تمامی دیدهای کاربران هست .

تعریف سطح مفهومی :

دیدي است که طراح بانک اطلاعاتی نسبت به کلیه موجودیتهای و ارتباطهای بین آنها دارد این سطح دید جامعی از سیستم است و در حقیقت نمایش تمامی اطلاعات بانک می باشد .

تعریف سطح داخلی :

یک نمایش سطح پائینی از تمامی اطلاعات یک بانک است در این سطح داده های بانک اطلاعاتی همانگونه که در محیط ذخیره سازی شده اند نمایش داده می شوند .

مثال:

(ID . Fname . Lname . Addres . Avg . Melli cod . Tell) دید مفهومی از دانشجو

(ID . Fname . Lname) دید خارجی ۱ از دانشجو

(ID . Lname Avg . Melli cod . Tell) دید خارجی ۱ از دانشجو

زبانهای مورد استفاده در پایگاه داده :

زبانهای میزبان (HL):

این زبانها برای ایجاد واسط کاربری استفاده می شوند مانند C#,VB,Delphi

زبانهای فرعی داده (DSL):

این زبانها برای انجام اعمال درج ، حذف ، ویرایش و جستجو بر روی پایگاه داده مورد استفاده قرار می گیرند .

برای مثال SQL

نکته : زبان SQL یک زبان DSL محسوب می شود و با نرم افزار SQLSERVER فرق دارد و هیچ ربطی به همدیگر ندارند.

زبان فرعی داده ها خود به سه دسته تقسیم می شود .

۱. DCL : زبان کنترل داده

۲. DDL : زبان توصیف داده

۳. DML : زبان ویرایش داده

نمودار ER :

تعریف ER:

یک مدل مفهومی از عام است مه به بررسی رابطه بین موجودیتها در یک سیستم پایگاه داده می پردازد .

تعریف موجودیت :

هر شیء ، پدیده یا مفهومی که می خواهیم در مورد آن اطلاعاتی ذخیره کنیم موجودیت نامیده می شود موجودیت را با نمودار ER نشان می دهند .

دارو

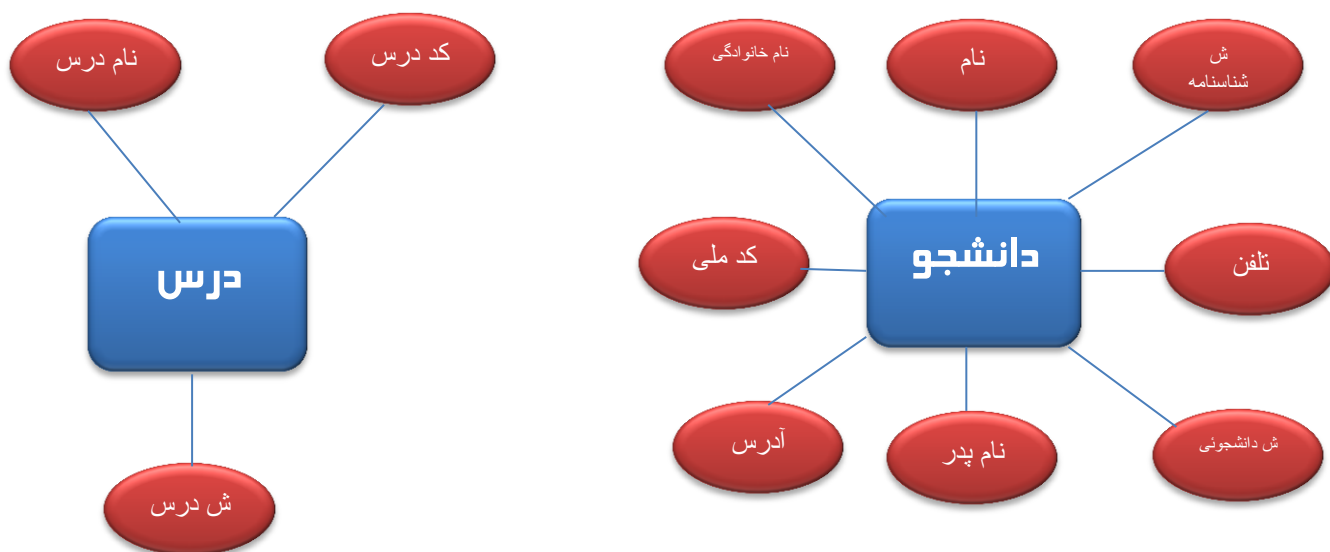
کتاب

درس

دانشجو

تعریف صفت :

ویژگیهای یک موجودیت را صفت می گویند و با بیضی نشان می دهند .



تعریف کلید :

صفت یا مجموعه ای از صفتها که برای یک موجودیت منحصر به فرد باشد .

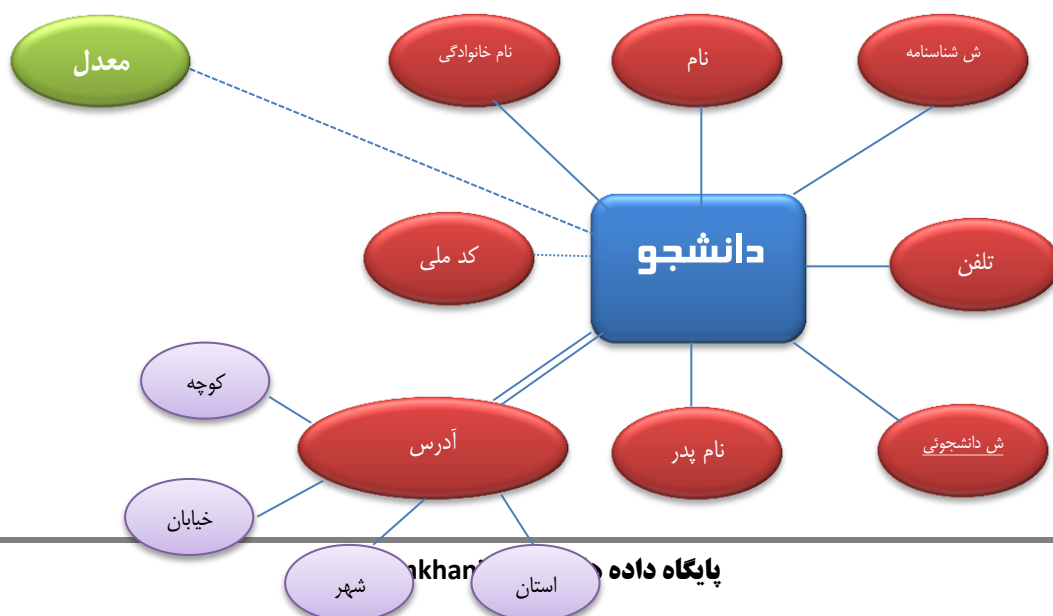
انواع صفت:

صفت ساده: صفتی می باشد که تجزیه پذیر نباشد مانند نام خانوادگی

صفت مرکب: صفتی است که تجزیه پذیر باشد مانند آدرس

صفت تک مقداری : صفتی که در هر زمان فقط یک مقدار می تواند داشته باشد مانند شماره دانشجویی

صفت چند مقداری : صفتی که در هر زمان می تواند چندین مقدار مختلف داشته باشد این نوع صفت را با دو خط موازی به موجودیت وصل می کنیم مانند شماره تلفن یا آدرس



نکته :

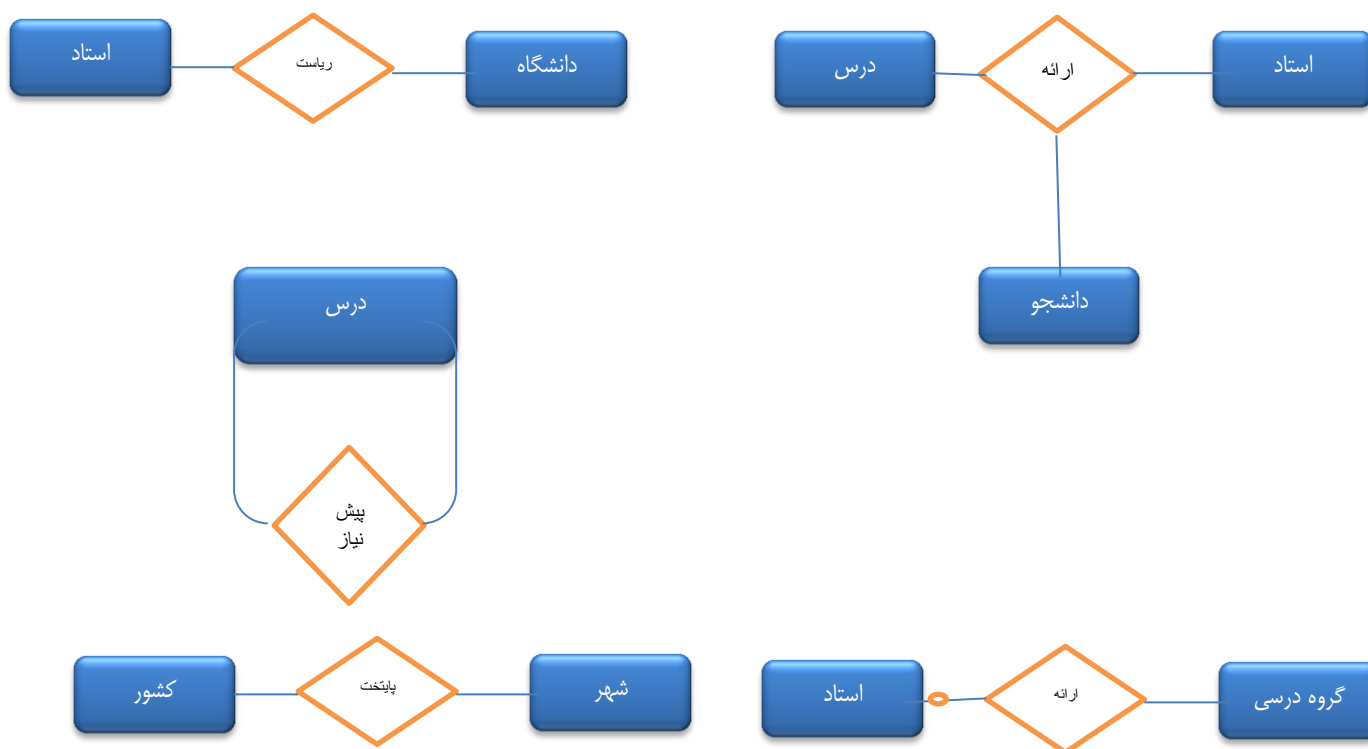
مرکب بودن و چند مقداری بودن هیچ ربطی به همدیگر ندارند ولی صفات ساده نیز می توانند چند مقداری باشند و ممکن است صفتی مرکب باشد ولی چند مقداری نباشد .

صفت مشتق یا محاسباتی :

این نوع صفات از روی بقیه فیلدها محاسبه می شوند و نیازی به ذخیره شدن ندارند مانند معدل برای دانشجو، این نوع صفات را با نقطه چین به موجودیت وصل می کنند .

ارتباط بین موجودیتها :

در یک سیستم پایگاه داده موجودیتهای مختلف با هم در ارتباط هستند ارتباط بین موجودیتها را با لوزی نشان می دهیم در داخل لوزی فعل یا کلمه ای مرتبط با آن می نویسیم در هر ارتباطی یک یا بیشتر از یک موجودیت در ارتباط شرکت می کنند .



نوع اتصال :

۱ به ۱

۱ به چند

چند به چند

نکته ۱ :

وجود هر موجودیتی در ارتباط می تواند اختیاری باشد اختیاری بودن را با دایره در کنار نام موجودیت مشخص می کنند.

نکته ۲:

تشخیص نوع اتصال از این لحاظ مهم می باشد که باید ارتباطهای چند به چند شناسایی شده و آنها را تبدیل به ارتباط های یک به چند کنیم زیرا ارتباط های چند به چند قابل پیاده سازی نمی باشند .

تعریف موجودیت مرکب :

بعضی از ارتباطات در نمودار ER ممکن است خود دارای صفاتی باشند بنابراین این ارتباط تبدیل به یک موجودیت می شود یکی از دلایل این کار وجود روابط چند به چند می باشد که باید آنها را به روابط یک به چند تبدیل کنیم.

نکته :

موجودیتهای مرکب حتماً باید کلید های اصلی موجودیتهای مرتبط را داشته باشند و معمولاً نام این موجودیت را ترکیبی از نام موجودیتهای قبلی قرار می دهیم .

نکته :

تشخیص نوع اتصال از این لحاظ مهم می باشد که باید ارتباط های چند به چند شناسایی شده و آنها را تبدیل به ارتباط های یک به چند کنیم زیرا ارتباط های چند به چند قابل پیاده سازی نمی باشند .

تعریف موجودیت مرکب :

بعضی از ارتباطات و نمودارهای ER ممکن است خود دارای صفاتی باشند بنابراین این ارتباط تبدیل به یک موجودیت می شود یکی از دلایل این کار وجود روابط چند به چند می باشد که باید آنها را به روابط یک به چند تبدیل کنیم .

نکته:

موجودیتهای مرکب حتماً باید کلیدهای اصلی موجودیتهای مرتبط را داشته باشند و معمولاً نام این موجودیت را ترکیبی از نام موجودیتهای قبلی قرار می دهیم .

نمره	نام درس	شماره درس
۱۰	پایگاه داده	۱
۰.۲۵	پایگاه داده	۱
۲۰	پایگاه داده	۱

نمره	نام دانشجو	شماره دانشجویی
۲۰	Ali	۸۸۸۰
۱۴	Hasan	۸۸۸۱
۱۵	Ali	۸۸۸۰

نمره	شماره دانشجویی	شماره درس
۲۰	۸۸۸۰	۱
۱۴	۸۸۸۱	۱
۱۵	۸۸۸۰	۲
۱۶	۸۸۸۱	۲

موجودیت وابسته یا ضعیف :

تعریف: بعضی از اوقات وجود یک موجودیت وابسته به موجودیت دیگری می باشد این نوع موجودیت را موجودیت وابسته می نامند .

نکته :

همیشه باید کلید اصلی موجودیت اصلی به عنوان یک کلید در موجودیت ضعیف آورده شود .

تعریف رابطه ارث پری :

در بعضی از سیستم ها ممکن است بعضی از موجودیتها صفات مشترک داشته باشند به جای تکرار این صفات مشترک می توان یک موجودیت جدید ایجاد کنیم که آن صفات را داشته باشد و بقیع موجودیتها این صفات را از آن موجودیت به ارث می برند .

تعمین :

پایگاه داده یک دانشگاه را در نظر بگیرید و نمودار ER این سیستم را با توجه به قوانین گفته شده رسم کنید .

- (۱) هر دانشجو تنها در یک رشته تحصیل می کند و در هر رشته دانشجویان متعددی تحصیل می کنند.
- (۲) هر رشته چندین درس دارد و هر درس ممکن است مربوط به چندین رشته باشد.
- (۳) برای هر نوع درس ممکن است چندین درس وجود داشته باشد ولی هر درس تنها مربوط به یک نوع درس می باشد.
- (۴) در هر ترم ممکن است هر درس توسط چندین استاد ارائه شود و یا اصلاً ارائه نشود و هر استاد ممکن است در طول ترم چندین درس را تدریس بکند یا نکند.
- (۵) هر درس پیش نیاز چندین درس می تواند باشد یا نباشد و هر درس ممکن است چندین پیش نیاز داشته باشد یا نداشته باشد .
- (۶) هر دانشجو ممکن است درسهای متعددی را بگیرد یا نگیرد یا هر درس ممکن است توسط چندین دانشجو اخذ شود یا نشود .

فصل دوم

جبر رابطه ای (مساب رابطه ای)

جبر رابطه ای جبری است که ورودی و خروجی آن رابطه می باشد و بر روی رابطه ها پردازش انجام می گیرد منظور از رابطه همان مفهوم موجودیت در نمودار ER می باشد و یا همان مفهوم جدول SQL می باشد.

عملگرهای جبر رابطه ای :

عملگرهای ساده :

۱. انتخاب یا گزینش δ

۲. پرتو Π project

عملگرهای مجموعه ای :

۱. اجتماع $\cup$

۲. اشتراک $\cap$

۳. تفاضل $-$

عملگرهای پیوند :

۱. ضرب دکارتی $\times$

۲. ضرب شرطی $\times \theta$

۳. پیوند طبیعی $\bowtie$

۴. نیم پیوند $\ltimes$

عملگر تقسیم $\div$

عملگر جایگزین $\leftarrow$

عملگر نامگذاری ρ

دو پایگاه داده را به صورت زیر فرض می کنیم :

(۱) پایگاه داده مربوط به تولید کننده قطعات و پروژه های در حال اجرا :

(شهر تولید کننده ، نام تولید کننده ، شماره تولید کننده) رابطه تولید کننده S(S#,Sname,city)

(رنگ قطعات ، نام قطعات ، شماره قطعات) رابطه قطعات P(P#,Pname,color)

(شهر پروژه ، نام پروژه ، شماره پروژه) رابطه پروژه های در حال اجرا J(J#,Jname,city)

(تعداد قطعه ، شماره پروژه ، شماره قطعه ، شماره تولید کننده) SPJ SPJ(S#,P#,J#,Qty)

(۲) پایگاه داده مربوط به دانشگاه :

(شماره دانشکده،معدل، شهر دانشجو ، نام دانشجو ، شماره دانشجویی) جدول دانشجو Stud(S#,Sname,city,avg,clg#)

(شماره دانشکده ، مدرک،تخصص، دفتر ، نام استاد) جدول استاد Prof(Pname,office,esp,degree,clg#)

(شماره دانشکده ، واحد درس، نام درس ، شماره درس) جدول درس Crs(C#,Cname,unit, clg#)

(نام رئیس دانشکده ، شهر دانشکده ، نام دانشکده ، شماره دانشکده) جدول دانشکده Clg(Clg#,Clgname,city,Pname)

(نمره، نام استاد،ترم، شماره دانشجویی ، شماره درس ، شماره SEC) SEC Sec(Sec#,C#,S#,term,Pname ,Score)

عملگر انتخاب (گزینش):

این عملگر برای انتخاب سطرهایی از جدول بکار میرود .

(نام جدول یا جداول مورد نیاز) شرط یا شرطها δ

مثال :

J#	Jname	city
1	A	Tehran
2	A	Tabriz
3	B	Urmieh
4	C	Tehran

تمامی مشخصات تمامی پروژه ها را بدست آورید ؟

$\delta(J)$

مثال :

تمامی پروژه های در حال اجرا در شهر تبریز را بدست آورید ؟

(J) city"tabriz" S

عملگر انتخاب δ (جدول) شرط δ سطر

عملگر پرتو Π (جدول) نام سطونهای مورد نظر Π ستون

در این نوع عملگر می توانیم ستونهای خاصی را از جدول مورد نظر انتخاب کنیم.

مثال: شماره دانشجویی و نام تمامی دانشجویان را بدست آورید؟

$\Pi S\#, Sname$ (stud)

مثال: نام شهرهایی را بدست آورید که در آنها پروژه ای در حال اجرا می باشد؟

$\Pi city$ (I)

مثال: شماره دانشکده هایی را بدست آورید که در شهر تهران واقع شده اند؟

$\Pi clg\#$ ($\delta city = "Tehran"$ (clg)

نکته:

همیشه عملگر δ (انتخاب) به عنوان ورودی برای عملگر پرتو Π داده می شود.

مثال: شماره تولید کنندگانی را بدست آورید که بیش از ۱۰۰ کیلو گرم از محصول شماره ۲۵۰ را تولید می کنند؟

$\Pi S\#$ ($\delta Qty > 100 \wedge P\# = 250$ (SPJ)

عملگرهای مجموعه ای:

عملگر اجتماع: $\cup$

عملگر اجتماع همانند عملگر اجتماع بر روی مجموعه ها در علوم ریاضی می باشد عملگر اجتماع را می توان بر روی جداولی استفاده کرد که با هم همتا باشند همتا بودن در جدول یعنی اینکه مقدار ستونهایشان یکسان باشد و همچنین نوع ستونهای متناظرشان یکسان باشد مثلاً سه جدول SPJ همتا هستند ولی جداول stud, prof همتا نیستند.

مثال: نام شهرهایی را بدست آورید که تولید کننده ای در آن زندگی میکند و یا پروژه ای در حال اجرا می باشد؟

$\Pi city$ (S) $\cup \Pi city$ (I)

مثال: لیست اسامی افراد یک دانشکده شامل دانشجویان و اساتید را بدست آورید؟

$\Pi Pname$ (prof) $\cup \Pi Sname$ (stud)

عملگر اشتراک: $\cap$

مثال : لیست دانشجویان و اساتید هم نام را بدست آورید ؟

$\Pi Pname^{(prof)} \cap \Pi Sname^{(stud)}$

عملگر تفاضل : -

نکته : عملگر تفاضل خاصیت جابجایی ندارد در حالیکه اجتماع و اشتراک خاصیت جابجایی داشتند .

مثال : شماره درسهایی را بدست آورید که در ترم اول سال تحصیلی ۹۱ ارائه نشده اند ؟

$\Pi C\#^{(crs)} - \Pi C\#^{(term=911)}$

آنهايي که کاراکتر هستند در داخل " کوتیشن " می نویسیم .

عملگر های پیوند :

ضرب دکارتی : $\times$

اگر جدول R_1 با M ستون و K سطر و R_2 با N ستون P سطر باشد R_1 و R_2 دارای $M+N$ ستون و $K \times P$ سطر خواهند بود .

x	y
1	۲
۱	۳
۲	۴

a	b
۱	۱
۲	۲

x	y	a	b
۱	۲	۱	۱
۱	۲	۲	۲
۱	۳	۱	۱
۱	۳	۲	۲
۲	۴	۱	۱
۲	۴	۲	۲

اگر همنام باشند دو تا y باشد.

نکته : اگر بین R_1 و R_2 ستون مشترک داشته باشیم هر دو تا ستون را در $R_1 * R_2$ ضرب خواهیم کرد فقط باید نام جدول را قبل از نام ستون مورد نظر بنویسیم .

نکته : ضرب دکرتی به صرفه نیست چونکه تعداد رکوردهایی که می دهد معمولاً زیاد می باشد و بیشتر اطلاعات معمولاً زائد می باشد .

عملگر ضرب شرطی : $\times \theta$

ضرب شرطی همانند ضرب دکارتی می باشد با این تفاوت که سطرهایی از دو جدول را انتخاب می کنیم و با یکدیگر پیوند می دهیم که شرط گفته شده (θ) را داشته باشد ..

مثال : در پایگاه داده مربوط به دانشگاه نام درس و شماره درسهایی را بدست آورید که توسط استادی به نام حسینی ارائه شده اند ؟

$\Pi C\#, Cname \ (CRS \times_{Pname} SEC = "hoseini" \wedge CRS.C\# = sec, C\#$

عملگر پیوند طبیعی: $\bowtie$

این عملگر همانند ضرب دکارتی است با این تفاوت که سطریهایی از دو جدول را که مقدار ستون مشترکشان یکسان باشد در کنار یکدیگر قرار می دهد .

مثال: نام و شماره درسهایی را بدست آورید که توسط استادی با نام حسینی ارائه شده اند ؟

$\Pi C\#, Cname \ (\delta_{Pname = "hoseini"} (CRS \bowtie SEC))$

CRS $\bowtie$ SEC						
C#	Cname	unit	Clg#	SEC#	S#	Pname
۱	DB	۴	۱	۱	۸۸	Hoseini
۲	Fs	۳	۱	۲	۸۸	Alizade
۳	Math	۳	۲	۲	۸۹	Fatahi

مثال: نام و شماره درسهایی را بدست آورید که توسط استاد حسینی ارائه شده اند ؟

$\Pi C\#, Cname \ (\delta_{Pname = "hoseini"}$

مثال: شماره پروژه هایی را بدست آورید که در آنها از تیر آهن استفاده شده است ؟

$\Pi J\# \ (\delta_{Pname = "تیر آهن"})$

مثال: نام تولید کنندگانی را بیابید که قطعات مشکی رنگ تولید می کنند ؟

$\Pi Sname \ (\delta_{color = "مشکی"})$

تمرین

(۱) نام تولید کنندگانی را بدست آورید که ساکن تهران هستند و قطعه تولیدی آنها تیر آهن است ؟

$\Pi Sname \ (\delta_{Pname = "Tirahan" \wedge city = Tehran} (S\# \bowtie P\# \bowtie SPJ))$

(۲) مشخصات کامل رؤسای دانشکده ها را بدست آورید ؟

$\Pi_{Pname} (Clg) \bowtie (Prof)$

(۳) نام تولید کنندگانی را بدست آورید که هیچ قطعه مشکی رنگی را تولید نکردند ؟

$\Pi_{Sname} (s) - \Pi_{Sname} (\delta_{color = "black"} (S \bowtie P \bowtie SPJ))$

عملگر نیم پیوند : $\circ$

مانند پیوند طبیعی است با این تفاوت که فقط ستونهای جدول اولی (جدول سمت چپ) در خروجی نمایش داده می شود .

مثال:

a	b
۱	۱
۱	۲
۲	۲
۲	۳

a	b
۱	۳
۲	۴
۲	۵
۴	۵

x	a	b
۱	۱	۳
۱	۲	۴
۱	۲	۵
۲	۲	۴
۲	۲	۵

نکته :

در نیم پیوند بعضی از ستونها حذف می شوند و حذف این ستونها ممکن است باعث ایجاد سطرهای تکراری شود ایت سطرها نیز حذف می شوند .

نکته :

اگر دو جدول ستون مشترک نداشته باشند پیوند طبیعی و ضرب دکارتی این دو جدول یکسان خواهد بود .

$$R1 * R2 = R2 * R1$$

$$R1 \circ R2 = R2 \circ R1$$

$$R1 \circ R2 \neq R2 \circ R1$$

مثال : با استفاده از نیم پیوند مشخصات دروسی را بیابید که در نیمسال اول سال ۹۰ در دانشکده شماره ۲ ارائه شده اند ؟

$$\delta_{term=901}^{CLG\#=2} (CRS \circ SEC)$$

$$\delta_{CLG\#=2} \circ \delta_{term=901}$$

عملگر تقسیم : $\div$

از این عملگر زمانی استفاده می کنیم که بخواهیم همه حالتها را یک اتفاق را بررسی کنیم .

مثال : تولیدکنندگانی که همه قطعات مشکی رنگ را تولید کرده اند ؟

مثال : دانشجویانی را که تمام دروس ارائه شده با استادی بنام حسینی را اخذ کرده اند ؟

مثال : تولیدکنندگانی که در همه پروژه های شهر ارومیه شرکت کرده اند ؟

X	Y	Z	Y	Z	R1÷R2
۱	۱	۱	۱	۱	۲
۱	۲	۲	۲	۲	
۲	۱	۱	۳	۴	
۲	۲	۲			
۲	۳	۴			
۳	۲	۲			
۳	۳	۳			
۴	۱	۱			
۴	۵	۶			

نکته :

- (۱) اگر بخواهیم جدول R₁ را بر R₂ تقسیم کنیم باید تمام ستونهای R₂ در جدول R₂ باشند .
- (۲) دو جدول حاصل از تقسیم R₁ بر R₂ ستونهایی از R₁ که در R₂ وجود ندارند نمایش داده می شوند .
- (۳) سطریایی از R₁ در نتیجه ظاهر می شود که به ازای تمام سطریهای جدول R₂ تکرار شده باشند .

مثال :

کد تولیدکنندگانی را بدست آورید که تمام قطعات مشکی رنگ را تولید کرده اند ؟

$\Pi_{S\#,P\#}(SPJ) \div \Pi_{P\#}(\delta_{color="black"}(P))$

عملگر نامگذاری : ρ

از این عملگر زمانی استفاده می کنیم که بخواهیم اطلاعات یک جدول را با خودش مقایسه کنیم .

مثال :

نام دانشجویانی را بدست آورید که حداقل یک همشهری داشته باشند ؟

$Stud *_{stud.city=stud.S^stud.S\# \neq S.S\#} (ps(stud))$

مثال:

نام دانشجویانی را بدست آورید که معدلشان حداقل از یک دانشجو دیگر بیشتر باشد .

$Stud *_{stud.Avg>S.Avg} (ps(stud))$

عملگر جایگذاری : ←

اگر بخواهیم اطلاعات یک جدول را به صورت موقت ذخیره کنیم از این عملگر استفاده می کنیم.

مثال : شماره تولیدکنندگانی را بدست آورید که تمامی قطعات مشکی رنگ را تولید کرده اند ؟

$temp \leftarrow \Pi P\# (\delta_{color="black"}(p))$

$\Pi S\#, P\#(SPJ) \div temp$

حذف ، درج و اصلاح کردن :

$CRS \leftarrow CRS \cup \{10, web, 3, 4\}$ درج :

$CRS \leftarrow CRS - \{10, web, 3, 4\}$ حذف :

$CRS \leftarrow CRS - \delta_{unit=4}(CRS)$

$\delta_{score} \leftarrow score + 1 \quad (S_{unit=3}(CRS) \oslash SEC)$ اصلاح رکورد :

مثال : به نمره های کمتر از ۱۹ یک نمره اضافه کرده و به بالای ۱۹، ۲۰ دهید؟

$\delta_{score} \leftarrow score + 1 \quad (\delta_{score \leq 19}^{(SEC)})$

$\delta_{score} \leftarrow 20 \quad (\delta_{score \geq 19}^{(SEC)})$

فیلدهای محاسباتی با عملگر پرتو :

$\Pi a-b \text{ as } sub^{(R1)}$

a	b	sub
۳	۰	۳
۵	۲	۳
۴	۱	۳

توابع جمعی :

توابعی هستند که می توانند یک یا چند ورودی را دریافت کرده و یک خروجی را بدهند و در حقیقت برخی از عملیات محاسباتی یا مقایسه ای را می توان توسط این توابع پیاده سازی کرد . برخی از این توابع عبارتند از :

Max, Min, Sum, Count, Avg

نام جدول
 $\sum$ ستونهای مورد نظر , گروه بندی
 تابع تجمعی

مثال : با فرض جدول دانشجو به صورت زیر مشخصات دانشجو با بالاترین معدل را بدست آورید ؟

STUDENT (S#,Sname,Avreg,major)

$$\text{Student} \propto \sum_{\text{Max } (Avg), S\#}^{\text{student}}$$

مثال : مشخصات دانشجویان ممتاز برای هر رشته را بدست آورید؟

$$\text{Student} \propto (major) \sum_{\text{Max } (Avg), S\#}^{\text{student}}$$

فصل سوم :

:SQL

یک زبان پرس و جو ساخت یافته می باشد که برای کار با پایگاه داده مورد استفاده قرار می گیرد و پرس و جوی کاربران را بر روی DBMS اجرا می کند .

دستور ایجاد جدول :

نام جدول Creat table

(تعریف فیلدها)

)

مثال : جدول دانشجو را ایجاد کنید ؟

Stud Creat table

(S Number char(10) not null.

S Name char(20) not null.

City char(30) not null.

Avg double null.

clgNumber int(4) not null.

Primary key (SNumber)

Foreign key (clgNumber)References clg

On update cuscade

On delate cascade

Check (Avg<20 and Avg>0)

Char: حروف انگلیسی فقط می تواند ذخیره کند .

مثال : char(30) اندازه اش کامل است ولی فارسی پشتیبانی نمی کند

Nchar(30) (national) فارسی را پشتیبانی می کند.

Varchar(30) اندازه اش متغیر است هم کاراکتری است فارسی را پشتیبانی نمی کند . (variable)

Nchar(40) فارسی را ساپورت می کند و هم اندازه اش متغیر است حداکثر ۴۰ تا اندازه اش می تواند باشد کم هم می تواند باشد مثلاً ۳ تا بزرگتر باشد فضای بیشتری اشغال می کند .

دستور جدول حذف : (ساختارش کلاً پا می شود) ; نام جدول drop Tale

دستور حذف رکوردهای جدول : ساختار می ماند اطلاعات داخل را خالی می کند . ; نام جدول delet Table

تغییر ساختار جدول : منظور از تغییرات ساختار جدول اضافه کردن یا ویرایش یک فیلد از جدول می باشد .

اضافه کردن ستونی :

Alter table Add (نام ستون نوع ستون Null/not Null) نام جدول

ویرایش ستونی از جدول:

Alter table modifg (نام ستون نوع ستون Null/not Null) نام جدول

مثال اضافه کردن ستون :

Alter table stud Add (Tell char(11) Null/not Null);

مثال اضافه کردن ستون :

Alter table stud modifg (Tell char(15) Null);

دستور select :

Select II نام ستونهای مورد نظر

From پیوند نام جدول

Where ; شرط δ

مثال : مشخصات تمامی دانشجویان را بدست آورید؟

Select * منظور همه ستونها all

From stud;

مثال : شماره دانشجویی و نام دانشجویان ساکن شهر ارومیه را بدست آورید؟

```
Select SNumber, SName
```

```
From stud
```

```
Where city="urmieh";
```

```
 $\Pi_{S\#,Sname} (\delta_{city="urmieh"} (stud))$ 
```

نکته: در SQL بر خلاف جبر رابطه ای مقادیر تکراری حذف نمی شوند برای حذف مقادیر تکراری از دستور **distinct** استفاده می کنیم.

مثال: نام دانشجویان را از جدول دانشجو بدست آورده و مقادیر تکراری آن را حذف کنید ؟

```
Select distinct SName
```

```
From stud;
```

یعنی اگر دو تا دانشجو هم نام باشند یکی از آنها را حذف می کند .

مثال: شماره تولیدکنندگانی را بدست آورید که محصول سیمان را تولید کرده اند ؟

```
 $\Pi_{S\#}(\delta_{pname="siman"} (P \bowtie SPJ))$ 
```

```
Select SNumber
```

ضرب دکارتی

```
From P,SPJ;
```

```
Where Pname ="siman" and P.P#=SPJ.P#
```

```
 $\Pi_{S\#} (P * SPJ_{pname="siman" \wedge P.P\#=SPJ.P\#})$ 
```

ضرب شرطی

مثال: اسامی دانشجویان ساکن ارومیه که معدل آنها ما بین ۱۷ تا ۲۰ می باشد بدست آورید ؟

```
Select SName
```

```
From stud
```

```
Where city="urmieh" and Avg bet ween (17and20) ;
```

```
Or Avg 17 and Avg <20
```

مثال: نام دانشجویانی را بدست آورید که ساکن یکی از شهر های ارومیه ، تبریز ، تکاب ، زنجان باشند ؟

```
Select SName
```

```
From stud
```

```
Where city in ( "urmieh", "Tabriz", "takab", "zanjan");
```

مثال : با توجه به بانک اطلاعاتی زیر اسامی ، شماره وام و میزان وام افرادی را بدست آورید که از شعبه ای بنام آزادی ما بین ۱۰۰۰ تا ۲۰۰۰ دلار وام دریافت کرده اند ؟

Branch (br- name , br- city, assets)	اطلاعات شعبه :
Customer (cu-name,cu-street, cu-city)	اطلاعات مشتری ها:
Loan (loan#,br- name,amount)	اطلاعات وامها :
Borrower (cu-name,loan#)	افراد وام گیرنده :
Account (account# , br-name, blance)	اطلاعات حسابها :
Depositor (cu-name,accont#)	افراد صاحب حساب :

جواب:

Select cu-name , loan #, account

From loan , borrower

Where br-name ="Azadi ' and amount between 1000\$ and 2000\$ loan .loan#=borrower.loan#

تغییر نام : as

نام ستون و جدول را می توانیم با دستور as تغییر دهیم .

مثال : نام شعبه هایی را بدست آورید که دارایی آنها حداقل از یکی از شعبه های شهر ارومیه بیشتر باشد ؟

SELECT br - name

FROM branch as b1 , branch as b2

WHERE b1.assets>b2.assets and b2.br – city ="urmieh"

توابع تجمعی : Max,Min,Sum,Count,Avg

مثال: میانگین موجودی همه حسابها را بدست آورید ؟

```
SELECT Avg(balance)
FROM Account
```

مثال : میانگین موجودی شعبه دانشگاه را بدست آورید؟

```
SELECT Avg(balance)
FROM Account
WHERE br -name="clg"
```

مثال : میانگین موجودی هر شعبه را بدست آورید؟

```
SELECT Avg(balance) , br- name
FROM Account
Group by br - name
```

مثال : نام شعبه هایی را بدست آورید که میانگین موجودی آنها از ۱۰۰ هزار دلار بیشتر باشد؟

```
SELECT Avg(balance) , br- name
FROM Account
Group by br – name
Having Avg(balance)>100000$
```

نکته :

توابع تجمعی را در قسمت WHERE نمی توان استفاده کرد اگر بخواهیم روی توابع تجمعی شرطی اعمال کنیم باید از دستور having استفاده کنیم .

Having بعد از دستور group by نوشته می شود .

نکته :

Where ,having را با هم می توانیم در دستور select استفاده کنیم در اینصورت ابتدا دستور where می آید و سپس having را خواهیم داشت .

مثال : میانگین موجودی افرادی را بدست آورید که ساکن ارومیه بوده و حداقل دارای ۳ حساب باشند ؟

```
SELECT Avg(balance), cu – name
FROM account , customer , depositor
WHERE cu – city = "urmieh" and account. Account# depositor.account# and customer .cu –
name=depositor .cu-name
```

Group by cu – name

Having count (account#)>=3

پرس وجوهای تودر تو :

اگر بخواهیم از نتایج یک پرس و جو در پرس و جوی دیگری استفاده کنیم باید آنها را به صورت تودر تو بنویسیم .

نکته:

اگر بخواهیم نتیجه یک تابع محاسباتی را با یک تابع محاسباتی دیگر مقایسه کنیم باید از پرس و جوهای تودر تو استفاده کنیم.

مثال : نام شعبه هایی را پیدا کنید که میانگین موجودی آنها از میانگین موجودی کل بیشتر باشد؟

SELECT br – name

FROM account

Group by br – name

Having Avg (balance)>(select Avg (balance)

From account

مثال : نام شعبه هایی را پیدا کنید که میانگین موجودی آنها از میانگین موجودی تمام شعبه ها بیشتر باشد ؟

SELECT br – name

FROM account

Group by br – name

Having Avg (balance)> all (select Avg (balance)

From account

Group by br- name

View دید :

یک جدول مجازی است که نیازی به ذخیره شدن ندارد و معمولاً از روی جداول موجود می توانیم view ایجاد کنیم .

پرس و جو as نام دید view Creat

مثال : اگر جدول دانشجویی به صورت زیر داشته باشیم view ای برای دانشجویان کامپیوتر بسازید؟

STUDENT (S# , Sname, major)

Creat view com stud as

Select *


```
From Student
Ehere major="com"
```

مثال : نام رتبه اول رشته کامپیوتر را بدست آورید؟

STUDENT (S# , Sname, major,Avreg)

```
Creat view com stud as
Select *
From Student
Ehere major="com"
```

```
Select Sname , Max (Avreg)
From com stude
```

مثال : یک view ایجاد کنید که شامل نام تولیدکنندگانی باشد که سیمان تولید کردند؟

```
Creat view com S siman as
Select Sname
From S,P.SPJ
Where Pname="siman" and S.S#=SPJ.S# and P.P#=SPJ.P#
```

درج رکورد در رابطه :

```
Insert in to (نام فیلدها) values (مقادیر)
Insret into stud (S#,Sname, Avg, city, clg#)values (90,"Arash",17, 'urmieh", 2)
Insret into stud (S#,Sname, Avg, clg#, city)values (90,"Arash",17, 2, "urmieh")
Insret view stud (Sname,clg#,S#,city,)values ("Arash",2,90,"urmieh")
```

نکته :

۱. اگر نام فیلدها و دستور insert درج نشوند باید تمامی فیلدها را مقدار دهی نمائیم و ترتیب فیلدها همان ترتیب فیلدها در جدول می باشند ولی اگر نام فیلدها را در دستور Insert درج کنیم می توانیم ترتیب آنها را به هم بزنیم و همچنین هر ستونی که بخواهیم مقدار نداشته باشد را ذکر نمی کنیم.
۲. در داخل دستور Insert می توان از دستور select استفاده کنیم .

مثال : فرض کنیم می خواهیم برای دانشجویان حساب بانکی با مبلغ پیش فرض صد هزار تومان ایجاد کنیم دستور insert مربوطه را بنویسید؟

S Acc (SAcc#,Sname, blance)

Insert into SAcc

Select S#,Sname, 100000

From stud

حذف رکورد از رابطه :

Delete from نام جدول

Where شرط

مثال :

دانشجویان رشته کامپیوتر را از سیستم حذف کنید ؟

Delete from student

Where magor="com"

مثال :

شعبه هایی که دارای آنها از میانگین دارائیهها کمتر باشد حذف کنید؟

Delete from branch

Where asset<select Avg (assets)

اصلاح رکورد :

Update نام جدول

Set ومقدار = نام فیلد

Where شروط

مثال : نمره دانشجوی با شماره دانشجویی ۹۰۹۱ را به نمره ۲۰ تغییر دهید؟

Update sec

Set scare=20

Where S#=9091

مثال: ۵٪ سود به موجودی همه حسابها اضافه کنید؟

Update account

Set blance = blance * 5% + blance

مثال: ۱۰٪ سود به موجودی مشتریانی که موجودیشان بیشتر از میانگین موجودی ها باشد اضافه کنید؟

Update account

Set blance = blance * 10% + blance

Where blance = select Avg(blance) from account

اعطای مجوز به کاربران :

کاربران to جدول on دستورات Grant

کاربران from جدول on دستورات Revoke

مثال :

Grant select on stude to Ali

Revoke delete on stude from Ali

با استفاده از دستور Grant می توان مجوز جستجو ، حذف ، درج، و یا متغیر درون یک جدول را به کاربران اعطا کرد و به همین ترتیب می توان توسط دستور Revoke مجوزی را از کاربران گرفت .

پایان