

# بسمه تعالی

جزوه پایگاه داده همراه با مثال ها

نویسنده : بهنام یوسفی

<http://www.Compfo.net>

ایمیل: [behnam.yosefi@gmail.com](mailto:behnam.yosefi@gmail.com)

تعاریف اولیه

۱- **محیط عملیاتی** : هر سازمان یا محیطی که بخواهیم برای آن یک سیستم مکانیزه (سیستم کامپیوتری) ایجاد نماییم را محیط عملیاتی میگوییم. مانند دانشگاه و بیمارستان و.....

۲- **موجودیت** : هر پدیده یا شی را در محیط عملیاتی موجودیت می گویند. به عنوان مثال در محیط عملیاتی دانشگاه موجودیت های استاد و دانشجو و درس وجود دارند.

۳- **صفت خاصه** : هر موجودیت دارای یک سری خصوصیات و ویژگی ها است که به آن صفت خاصه گفته میشود. صفت خاصه دارای ۲ مولفه است. نام صفت خاصه و مقدار صفت خاصه.

مثال:

محیط عملیاتی: دانشگاه

موجودیت : دانشجو

نام : شماره دانشجویی ، رشته و...

صفت خاصه

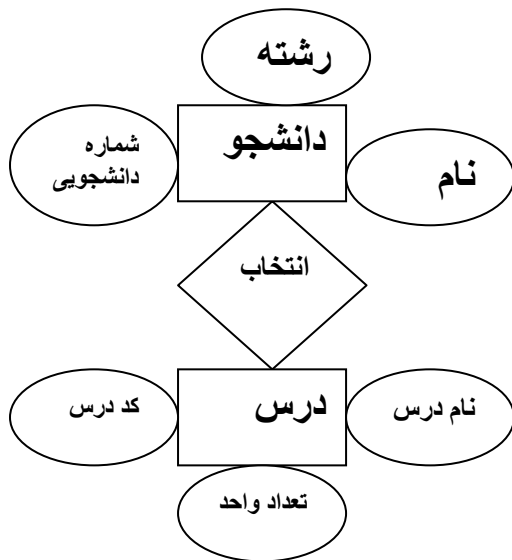
مقدار: ۵۶۷۲۷۲۷ ، گرافیک و...

جدول دانشجو

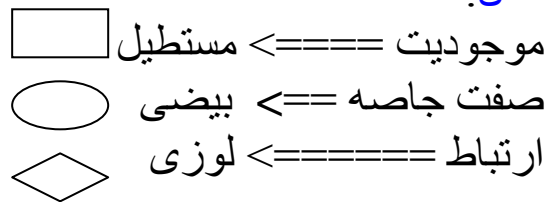
شماره دانشجویی	نام	نام خانوادگی	رشته
۶۷۶۵۵۵۶۵	زهرا	اکبری	کامپیوتر
۶۳۴۴۳۴۶۳۵۲	علی	حسینی	کامپیوتر

نمودار ER (entity relation ship): نمودار er ارتباط بین

موجودیت های یک محیط عملیاتی را نشان میدهد این نمودار دارای ۳ عنصر اصلی به صورت زیر است.



مثال:



بررسی اجزاء نمودار er

الف) موجودیت : موجودیت قوی ، موجودیت ضعیف

ب) صفت خاصه : صفت کلیدی، صفت ساده و مرکب، صفت تک مقداری و چند مقداری، صفت مشتق

ج) ارتباط : درجه ی ارتباط ، چندی ارتباط ، وضع مشارکت در ارتباط

الف) موجودیت : ۱) موجودیت قوی: موجودیتی که وجودش وابسته به موجودیت دیگر نبوده و به صورت مستقل در یک محیط عملیاتی مطرح می شود. مانند موجودیت استاد ، درس، دانشجو  
 ۲) موجودیت ضعیف : موجودیتی که وجودش وابسته به موجودیت دیگر باشد یعنی به صورت مستقل در یک محیط عملیاتی مطرح نمی شود مانند موجودیت عضو خانواده برای یک کارمند

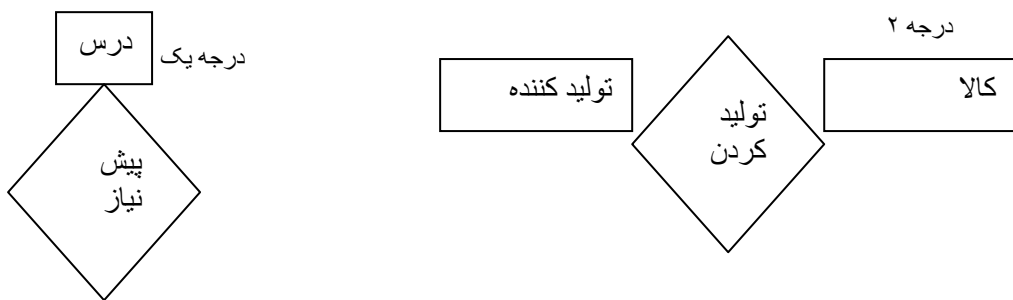
ب) صفت خاصه : ۱) صفت کلیدی: صفت کلیدی یعنی یک یا چند صفت خاصه که مقادیر آن ها در نمونه های مختلف یکتا باشد (یعنی در هیچ دو رکوردی مقدار تکراری نباشد) (\*رکورد=یک ردیف از جدول)

۲) صفت ساده و مرکب: صفتی که از ترکیب چندین صفت دیگر تشکیل شده باشد. مانند صفت آدرس که از ترکیب صفات شهر ، خیابان و .. درست شده است

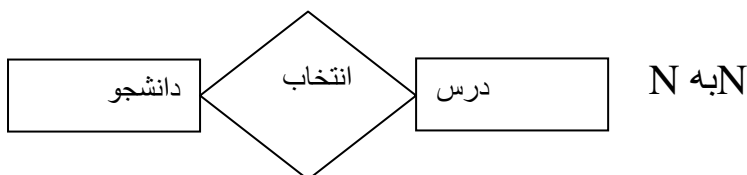
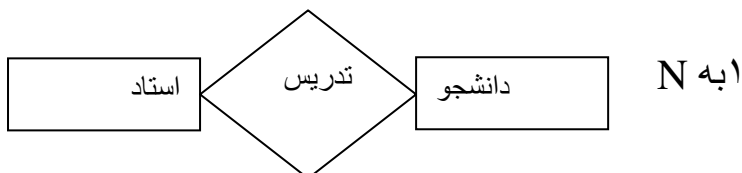
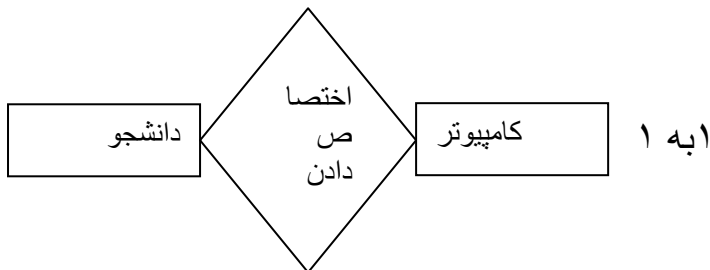
۳) صفت تک مقداری و چند مقداری: صفتی که در یک لحظه از زمان دارای چندین مقدار باشد را صفت چندمقداری می نامیم.  
 مثال) مدرک تحصیلی = چند مقداری

۴) صفت مشتق: می توان مقدار آن را با استفاده از صفت های دیگر به دست آورد. مانند صفت سن زیرا مقدار آن با استفاده از صفت تاریخ تولد نیز بدست می آید.

ج) ارتباط: ۱) درجه ی ارتباط: تعداد موجودیت هایی که در یک ارتباط شرکت دارند را درجه ی آن ارتباط می نامیم



۲) چندی ارتباط: ارتباط (۱ به ۱)  
ارتباط (۱ به N)  
ارتباط (N به N)



وضع مشارکت در ارتباط: مشارکت یک موجودیت به دو دسته تقسیم می شود ، مشارکت اجباری یا کامل و مشارکت اختیاری یا غیر کامل به عنوان مثال انتخاب واحد برای موجودیت دانشجو یک مشارکت اجباری می باشد اما عمل تدریس برای موجودیت استاد یک مشارکت اختیاری است .

تعریف پایگاه داده: عبارت است از مجموعه ای از داده های ذخیره شده در مورد موجودیت های یک محیط عملیاتی که به صورت مجتمع و مبتنی بر یک ساختار مشخص با حداقل افزونگی اطلاعات مورد استفاده ی یک یا چند کاربر به صورت مشترک و همزمان باشد.

**افزونگی** : افزونگی اطلاعات عبارت است از تکرار مقادیر یک یا چند صفت خاصه در نمونه های مختلف از یک فایل.  
مثال:

جدول اطلاعات تولید کنندگان

شماره تولید کننده	نام تولید کننده	وضعیت
S1	فن آوران	۱۰۰
S2	پولادین	۵۰
S3	ایران قطعه	۲۰

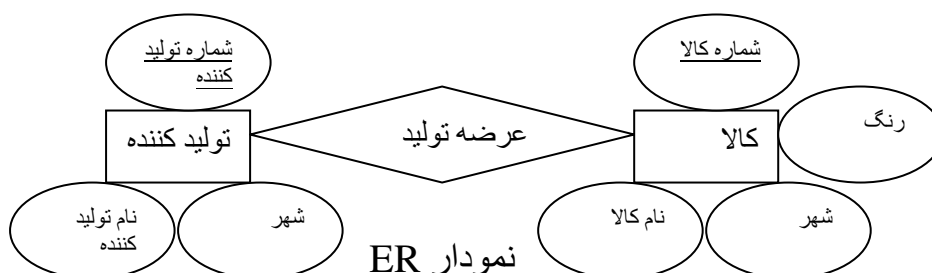
جدول میزان تولید

شماره تولید کننده	شماره کالا	میزان تولید	وضعیت
S1		۴۰۰	۱۰۰
S1	P1		
S1	P2	۵۰۰	۱۰۰
S2	P2	۳۰۰	۵۰
S2	P3	۴۰۰	۵۰
S3	P1	۵۰۰	۲۰
S3	P2	۲۰۰	۲۰

**توضیح** : در جدول میزان تولید، افزونگی اطلاعات رخ داده است ولی در جدول اطلاعات تولید کنندگان چنین نیست پس بهتر است فیلد وضعیت در جدول اطلاعات تولید کنندگان قرار گیرد.

ساختارهای پایگاه داده ها: ۱- ساختار رابطه ای ۲- ساختار سلسله مراتبی ۳- ساختار شبکه ای ۴- ساختار شی گرا.

**ساختار رابطه ای (جدولی)**: در ساختار رابطه ای عنصر اصلی جدول می باشد، در این ساختار برای هر موجودیت یک جدول و برای ارتباط بین آن ها جدول دیگر در نظر می گیریم. جدول ساختاری است مشخص که از تعدادی سطر و ستون تشکیل شده است. ستون های جدول نام صفات خاصه می باشند و به آن ها **فیلد** می گویم و ردیف های جدول مقادیر صفت خاصه می باشند که به آن ها **رکورد** گفته می شود.  
مثال:



جدول تولید کنندگان s# کلید

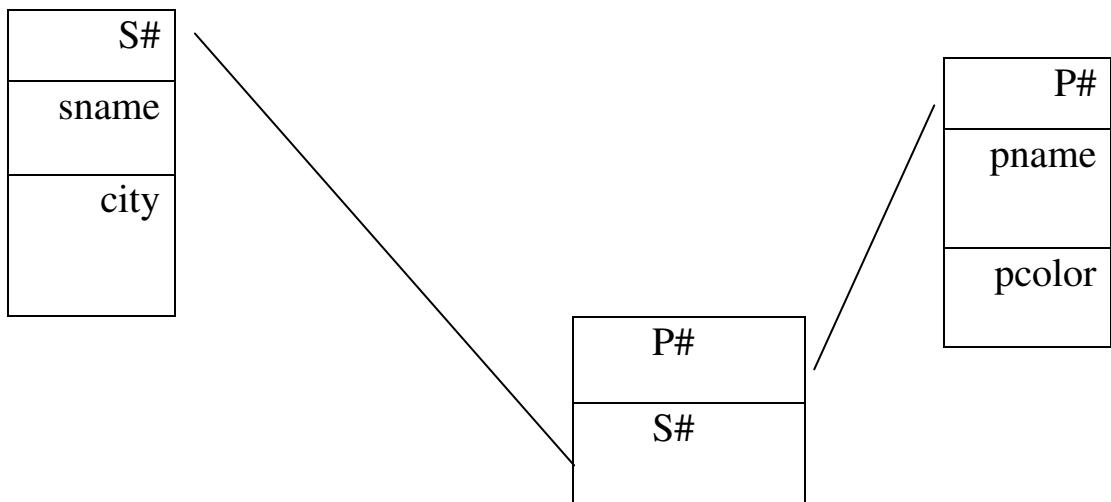
S#	sname	city
S1	فن اوران	تهران
S2	پولادین	تهران
S3	ایران قطعه	کرمان

جدول کالا p# کلید

P#	pname	pcolor	city
P1	اهن	قرمز	تهران
P2	فولاد	قرمز	تهران
P3	مس	زرد	کرمان

S#	P#
S1	P1
S1	P2
S2	P2
S3	P3

جدول انتخاب



برسی اعمال روی بانک در ساختار رابطه ای :

۱- عمل بازیابی (select): مثال ۱) شماره و نام تولید کنندگان واقع در تهران را مشخص نمایید.

```
Select s#,sname  
From s  
Where city='تهران'
```

مثال ۲) کلیه مشخصات قطعاتی را تعیین کنید که رنگ آن ها قرمز باشد.

```
Select *  
From p  
Where pcolor='قرمز'
```

۲- عمل درج (insert):

می خواهیم یک تولید کننده ی جدید با مشخصات زیر را به جدول تولیدکنندگان اضافه کنیم.

نام تولیدکننده : الومین  
شماره تولید کننده : s4  
شهر : اصفهان

```
Insert  
Into s  
Value('s4','الومین','اصفهان');
```

۳- عمل حذف (Delete):

می خواهیم مشخص کنیم کد شرکت s3 دیگر از قطعه p3 تولید نمی کند .

```
Delete  
From sp  
Where s#='s3' and p#='p3';
```

۴- عمل بهنگام سازی (update):

می خواهیم شهر تولید کننده ی s1 را به مقدار اهواز تغییر دهیم.

```
Update s  
Set city='اهواز'  
Where s#='s1';
```

**ساختار سلسله مراتبی:** در این ساختار دو عنصر اصلی وجود دارد. ۱- رکورد ۲-

پیوند پدر فرزندی. از رکورد برای نمایش موجودیت ها استفاده می شود. و برای نمایش ارتباط از پیوند پدر فرزندی استفاده می شود. که نشان دهند ارتباط یک به چند می باشد. در طرف یک ارتباط رکورد پدر و در طرف چند ارتباط رکورد فرزند قرار دارد.

مثال:

P: پدر

S: فرزند

P1	تهران	قرمز	اهن
----	-------	------	-----

S1	تهران	فن اوران	۴۰۰
----	-------	----------	-----

P2	تهران	قرمز	فولاد
----	-------	------	-------

S1	تهران	فن اوران	۵۰۰
S2	کرمان	پولادین	۵۰۰

P3	کرمان	زرد	مس
----	-------	-----	----

S3	کرمان	ایران قطعه	۳۰۰
----	-------	------------	-----

- انومالی:** شرایط نامطلوب در عملیات روی بانک (بازیابی، درج، حذف، بهنگام سازی) را انومالی گوئیم، این شرایط نامطلوب در ۳ وضعیت به وجود می آید.
- ۱- یک عمل امکان پذیر نباشد.
  - ۲- یک عمل امکان پذیر باشد اما عوارض نامطلوبی داشته باشد.
  - ۳- یک عمل با دشواری امکان پذیر باشد.

**بررسی اعمال روی بانک در ساختار سلسله مراتبی :**

۱- عمل بازیابی (select) :

مثال : شماره ی تولید کنندگانی که قطعه p2 را تولید کرده اند بیابید. با توجه به ساختار ارائه شده شرکت های s1 و s2 جواب مسعله می باشند. اما اگر بخواهیم مشخص کنیم که تهیه کننده ی s1 چه قطعاتی را تهیه کرده است باید کلیه سلسله مراتب ها را بررسی کرده و مشخص کنیم s1 چه قطعه را عرضه کرده است. پس به دشواری می شود (انومالی است)

## ۲- عمل درج (insert)

می خواهیم یک تولید کننده ی جدید با مشخصات زیر را به جدول تولیدکنندگان اضافه کنیم.

نام تولیدکننده : الومین

شماره تولید کننده : s4

شهر : اصفهان

\*\*\*با توجه به ساختار ارائه شده این عمل امکان پذیر نمی باشد زیرا تا هنگامی که ندانیم s4 چه قطعه ای را تهیه کره است نمی توانیم این شرکت را درج کنیم.

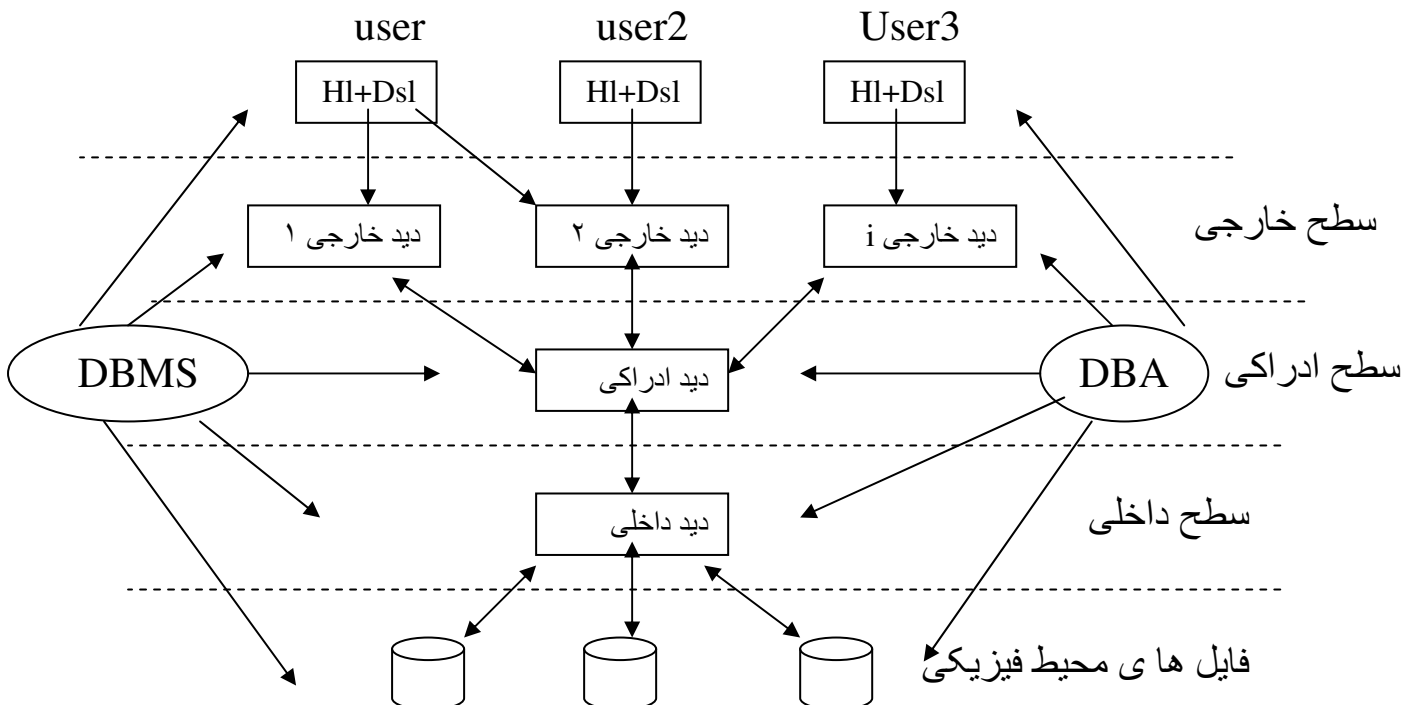
## ۳- حذف (delete):

می خواهیم مشخص کنیم کد شرکت s3 دیگر از قطعه p3 تولید نمی کند .  
\*\* این عمل امکان پذیر می باشد اما حذف اطلاعات ناخواسته ی دیگری را نیز سبب می شود.

## ۴- بهنگام سازی (update):

می خواهیم شهر تهیه کننده ی s1 را به مقدار اهواز تغییر دهیم برای انجام این عمل بایستی در تمامی نمونه هایی که شرکت s1 وجود دارد . این تغییر را اعمال کنیم . بنابراین حجم عملیات سیستم افزایش یافته و در نتیجه این عمل نیز در ساختار سلسله مراتبی دارای شرایط نامطلوب می باشد.

معماری پایگاه داده ها:



اجزای معماری پایگاه داده ها:

- کاربر (user)
- زبان میزبان یا hl (host language)
- زبان داده ای فرعی یا DSL (data sub language)
- دید خارجی (External view)
- دید ادراکی (conceptual view)
- دید داخلی (internal view)
- DBA ,DBMS

### دید ادراکی (conceptual view)

دید طراح بانک از کل داده های موجود در بانک اطلاعاتی می باشد. این دید در برگیرنده ی نیازهای اطلاعاتی کلیه کاربران می باشد.  
مثال:

جدول دانشجویان

شماره دانشجویی	نام	نام خانوادگی	رشته	ادرس

تعداد واحد	نام درس	کد درس

جدول درس

### دید خارجی (External view)

دید کاربر از داده های ذخیره در بانک اطلاعاتی می باشد. چندین کاربر ممکن است دید خارجی یکسانی داشته همچنین یک کاربر ممکن است دارای چندین دید باشد. دید خارجی نشان دهنده ی محدوده ای از پایگاه داده ها می باشد. که به نیازهای اطلاعاتی یک کاربر خاص پاسخ میدهد.  
دید خارجی منطقی از روی دید ادراکی طراحی و تعریف می شود بنابراین به جداول دید ادراکی اصطلاحاً جداول مبنا گفته می شود.  
مثال:

جدول دانشجویان

شماره دانشجویی	نام	نام خانوادگی	رشته	ادرس

View (دیدگاه)

شماره دانشجویی	نام	نام خانوادگی	رشته	ادرس

زبان میزبان **HL**: یک زبان سطح بالا می باشد که کاربر می تواند با استفاده از آن و نوشتن برنامه نیازهای اطلاعاتی خود را برطرف کند.

زبان داده ای فرعی (**DSL**): به احکام پیش بینی شده در یک زبان بانک اطلاعاتی گفته می شود. این احکام شامل دستوراتی برای ایجاد جداول و کارکردن با داده های موجود در آن ها می باشد  
**DSL** به ۳ دسته تقسیم می شود:

**DDL** (data definition language)  
**DML** (data manipulation language)  
**DCL** (data control language)

**DDL**: به معنای زبان تعریف داده ها می باشد و شامل دستوراتی برای ایجاد جداول و تعیین نوع داده ها می باشد.  
**DML**: به معنای زبان دستکاری داده ها می باشد و شامل دستورات بازیابی، درج، حذف و بهنگام سازی می باشد.  
**DCL**: به معنای زبان کنترل داده ها بوده که شامل دستوراتی برای کنترل داده های موجود در بانک اطلاعاتی می باشد.

Creat table **DDL**

Select,insert,delete,update **DML**

Grand ,revoke **DCL**

دید داخلی (internal view) در سطح داخلی فایل های محیط فیزیکی تعریف می شود. در سیستم های بانک اطلاعاتی طراحان و کاربران به مسائل این سطح نمی پردازند.

تست) کدام تعریف نشان دهنده ی دید ادراکی است؟  
 الف) دید کاربر از بخش از بانک اطلاعاتی  
 ب) دید (data base admin) DBA از بخشی از بانک اطلاعاتی  
 ج) دید کاربر از کل بانک اطلاعاتی  
 د) دید DBA از کل بانک اطلاعاتی

مدل رابطه ای و جبر رابطه ای :

تعاریف اولیه:

۱- دامنه یا میدان: مجموعه ای است که مقدارهای یک صفت خاصه از آن انتخاب می شود.

مثال:  $\underline{D}_{city} = \{\text{تهران، کرمان، تبریز}\}$

S#	sname	City
		تهران
		کرمان
		تبریز
		قم

۲- رابطه: هر زیر مجموعه ای از حاصل ضرب دکارتی چند دامنه را رابطه می نامیم.

$D1 = \{1, 2\}$   
 $D2 = \{4, 5, 6\}$   
 $D1 * D2 = \{(1, 4), (1, 5), (1, 6), (2, 4), (2, 5), (2, 6)\}$   
 مثال:

$R1 = \{(1, 4), (1, 6), (2, 5)\}$

\*\*نمایش رابطه با جدول = در ساختار رابطه ای نیز از جدول برای نمایش موجودیت استفاده می کنیم بنابراین رابطه در ریاضیات معادل مفهوم موجودیت در ساختار رابطه ای می باشد.

۱	۴
۱	۶
۲	۵

چندگانه

چندگانه (tuple): به هر کدام یک از اجزای یک رابطه چندگانه یا tuple گفته می شود. به عنوان مثال زوج مرتب (2,4) یک چندگانه از رابطه ی R2 می باشد. چندگانه معادل مفهوم رکورد در ساختار رابطه ای می باشد.

**درجه رابطه:** تعداد صفات خاصه در یک رابطه را درجه ی ان می نامیم.  
(درجه ی رابطه نشان دهنده ی تعداد ستون ها یا فیلدهای یک جدول می باشد.)

**انواع کلید در مدل رابطه ای :**

سوپر کلید (super key) ، کلید کاندید (condidate key) ،  
کلید اصلی (primary key) ، کلید فرعی یا جانشین (alternative key)  
کلید خارجی (foreign key)

**سوپر کلید (super key):**

هر ترکیبی از نام صفات خاصه که در هیچ دو رکوردی مقدار یکسان نداشته باشد.

**کلید کاندید (condidate key)**

مجموعه ای از صفات خاصه در یک رابطه که دارای **دو شرط** زیر باشد کلید کاندید می نامیم.

۱- یکتای مقدار

۲- کمینه بودن: به معنای ان است که اگر هر یک از اجزای صفت خاصه ای را حذف نماییم یکتای مقدار عنصر دیگر از بین برود.

مثال

جدول s

S#	sname	city
S1	فن اوران	تهران
S2	پولادین	تهران
S3	ایران قطعه	کرمان

S#	P#	qty
S1	P1	400
S1	P2	500
S2	P2	400
S3	P3	300

جدول sp

**\*\*ترکیب صفات s#,p# در جدول sp کلید کاندید است.**

**\*\*ترکیب صفات s#,sname در جدول s کلید کاندید نیست.**

**کلید اصلی (primary key):**

از بین کلید های کاندید یک کلید به عنوان کلید اصلی معرفی می کنیم، دو ضابطه در تعیین کلید اصلی وجود دارد. الف) کوتاهتر بودن طول کلید ب) نقش و اهمیت کلید اصلی نسبت به سایر کلیدهای کاندید.

**کلید فرعی یا جانشین :** کلید کاندیدی که کلید اصلی نباشد را کلید فرعی یا جانشین می نامیم.

### کلید خارجی (foreign key):

کلید خارجی وسیله ای برای برقراری ارتباط بین جداول می باشد.  
 به عبارت دیگر کلید خارجی امکان ای برای ارجاع از یک جدول به جدول دیگر است. اگر صفتی در جدول کلید خارجی باشد ان صفت در جدول دیگر کلید اصلی خواهد بود.

### قوانین جامعیت:

جامعیت موجودیتی  
 جامعیت ارجاعی

**جامعیت موجودیتی:** این قانون بیان می کند که هیچ کدام از اجزای تشکیل دهنده ی کلید اصلی نباید توهی باشند.

**جامعیت ارجاعی:** چنان چه دو رابطه (جدول) معلوم و مشخص وجود داشته باشند اگر در رابطه ی اول صفاتی کلید خارجی نسبت به رابطه ی دوم باشند ان گاه ان صفات یا توهی خواهند بود یا مقادیر ان منطبق بر مقادیر کلید اصلی از رابطه ی دوم خواهد بود.  
 مثال:

کلید اصلی

جدول گروه های آموزشی

شماره گروه	نام گروه
۰۱	کامپیوتر
۰۲	حسابداری
۰۳	معماری

کلید خارج

شماره گروه	نام خانوادگی	نام	کد استاد
۰۱	.....	.....	۱۰۰
	.....	.....	۱۰۱
۰۳	.....	.....	۱۰۲

جدول استاد

عملگرهای جبر رابطه ای :

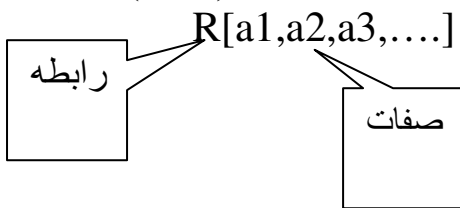
۱- عملگر انتخاب یا **select**: به عمل جداسازی چنگانه های (رکوردهای) خاص در یک رابطه، انتخاب یا **select** گفته می شود.

R(رابطه) where condition(شرط)

مثال: فقط رکوردهایی که شهر آن ها تهران است جدا کن از جدول s:

S where city='تهران'

۲- عملگر پرتو یا تصویر: به عمل جداسازی صفات خاص در یک رابطه (جدول) پرتو یا تصویر گفته می شود.



مثال: از جدول sp ستون s# را جدا کن:

Sp[s#]

مثال: شماره و نام کالاهایی را تعیین کنید که رنگ آن ها قرمز باشد.

جبر رابطه ای (قرمز='pcolor') [p#,pname] where (p

1

2

p#	pname	Pcolor
P1	.....	قرمز
P2	.....	قرمز
P3	.....	زرد

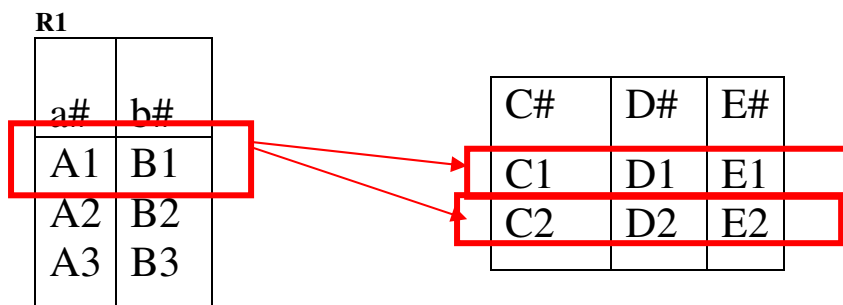
دستور sql :

Select p#,pname

From p

Where pcolor='قرمز'

۳- عملگر ضرب (product): به تناظر کلیدهای چندگانه های (رکوردهای) رابطه ی (جدول) اول با کلید چندگانه های (رکوردهای) رابطه ی (جدول) دوم ضرب آن دو رابطه گفته می شود.



R1 . R2

یا

R1 time R2

یا

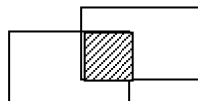
R1 product R2

A#	B#	C#	D#	E#
A1	B1	C1	D1	E1
A1	B1	C2	D2	E2
A2	B2	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C1	D1	E1
A3	B3	C2	D2	E2

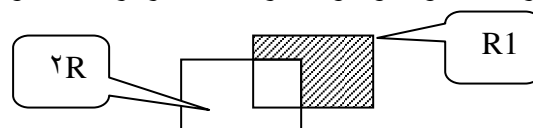
۴- عملگر اجتماع (union): اجتماع دو رابطه ، رابطه ای خواهد بود شامل تمامی چند گانه های رابطه اول و دوم.



۵- عملگر اشتراک (intersect): اشتراک دو رابطه رابطه ای خواهد بود شامل چندگانه های مشترک بین آن دو رابطه.



۶- عملگر تفریق (minus): نتیجه ی تفریق دو رابطه ، رابطه ای خواهد بود شامل چند گانه هایی که در رابطه اول وجود دارند اما در رابطه دوم وجود ندارند.



R1-R2

۷- عملگر الحاق یا پیوند (join) : با استفاده از عملگر الحاق می توان داده های دو جدول را بر اساس یک یا چند فیلد مشترک پیوند داد. ورکوردهایی از آن دو جدول را که مقدار آن فیلد مشترک برای آن ها یکسان باشد را انتخاب کرد.

A#	B#	C#
A1	B1	C1
A2	B2	C2

R1

B#	D#
B1	D1
B1	D2
B2	D3

R2

مرحله ۱) R1.R2

R1a#	R1b#	R1c#	R2b#	R2d#
A1	B1	C1	B1	D1
A1	B1	C1	B1	D2
A1	B1	C1	B2	D3
A2	B2	C2	B1	D1
A2	B2	C2	B1	D2
A2	B2	C2	B2	D3

مرحله ۲) R1.B#=R2.B#

R1a#	R1b#	R1c#	R2b#	R2d#
A1	B1	C1	B1	D1
A1	B1	C1	B1	D2
A2	B2	C2	B2	D3

حذف یکی از ستون ها (مرحله ۳)

R1a#	R1b#	R1c#	R2d#
A1	B1	C1	D1
A1	B1	C1	D2
A2	B2	C2	D3

مثال) نام تولید کنندگانی که کالای p2 را تولید کرده اند مشخص نمایید.

((sp where p#='p2')[s#] join s)[sname]

مثال) تولیدکنندگانی را مشخص کنید که کالایی با رنگ قرمز را تولید کرده باشند.  
 $((p \text{ where } pcolor='قرمز')[p\#] \text{ join } sp)[s\#] \text{ join } s)[sname]$

۸- عملگر تقسیم (divide by): نتیجه ی تقسیم دو رابطه رابطه ای خواهد بود شامل چندگانه هایی که به ازای تمامی چندگانه های دوم در رابطه اول وجود داشته باشد.

S#	P#
S1	P1
S1	P2
S1	P3
S2	P1
S2	P2
S3	P3
S3	P3

R2
P#
P1

p#
P1
P2

R1  
 R1 divid by R2: s# = s1,s2  
 R1 divid by R3: s# =s1,s2

مثال) نام تولید کنندگانی را تعیین کنید که همه کالا ه را تعیین کرده باشند.  
 $((sp[s\#,p\#] \text{ divide by } p[p\#]) \text{ join } s)[sname]$

S#	P#
S1	P1
S1	P2
S1	P3
S2	P2
S2	P3

p#
P1
P2
P3

divide by      ==>

S#
S1

(مثال) با توجه به جداول داده شده به سوالات زیر با جبر رابطه ای پاسخ دهید.  
 (st ( id,fullname,avg,address) جدول دانشجو  
 co (no,name,tedad,type) جدول درس  
 taken(id,no,mark) جدول انتخاب

(الف) نام و ادرس دانشجویانی را تعیین کنید که معدل ان ها کمتر از ۱۰ باشد.  
 (st where avg <10)[name]

(ب) نام دروس را تعیین کنید که دانشجو با نام علی اکبری ان ها را انتخاب کرده باشد.  
 (((st where fullname='علی اکبری')[id]join taken)[no]join co)[fullname]

(ج) نام دانشجویانی که تمام دروس اختیاری را انتخاب کرده اند مشخص کنید. (type=3 نشان دهنده ی دروس اختیاری )  
 ((taken[id,no] divide by (co where type=3)[no]) join st)[fullname]

## SQL

Sql مخفف عبارت structured query language است .  
 Sql به معنای زبان پرس وجوی ساخت یافته است.  
 Sql یک زبان بانک اطلاعاتی است که بر شیوه ی رابطه ای (جدولی ) استوار است.

### دستورات sql:

دستور create table : برای ایجاد جدول از این دستور استفاده می شود.

Create table s	create table p
(s# char(2) not null,	(p# char(2) notnull,
Sname char(20),	pname char(10) ,
City char(15),	city char(15),
Primary key (s#));	primary key (p#));

```
Create table sp
(s# char(2) notnull,
P# char(2) notnull,
Qty smallint,
Primary key(s#,p#)
Foreign key (s#) refrence s
```

On delete cascade  
 On update cascade  
 3 Foreign key (p#) refrence p  
 On delete cascade  
 On update cascade;

\*\*کار cascade: مثلا برای  
 P(خط ۳) اگر p# در جدول p  
 حذف یا update شود در  
 جدول sp هم به طور خودکار  
 انجام می شود.

دستور Drop table: برای حذف یک جدول از این دستور استفاده می شود.

Drop table s;

دستور Alter table: با استفاده از این دستور می توان ساختار یک جدول را  
 تغییر داد. به این صورت می توان ستونی را به سمت راست جدول اضافه کرد.

Alter table s  
 Add status smallint;

S#	sname	city	status

دستور select :

Select      فیلد (ها)  
 From        جدول(ها)  
 Where       شرط(ها);

مثال ۱) شماره و نام تولیدکنندگانی را مشخص کنید که واقع در تهران باشند.

```
Sql >>> select s#,sname
         from s
         where city='تهران';
```

```
>>> [s#,sname](تهران=city where s)
```

مثال ۲) کلیه مشخصات قطعاتی را تعیین کنید که رنگ ان ها قرمز باشد.

```
select *
From p
Where pcolor='قرمز';
```

مثال ۳) نام و رنگ قطعاتی را تعیین کنید که وزن آن ها کمتر از ۱۰ پوند باشد و شهر محل انبار آن ها تهران باشد.

```
Select pname , pcolor
From p
Where weight<10 and city='تهران';
```

مثال ۴) شماره قطعات و وزن هریک از آن ها را بر حسب گرم مشخص کنید.

```
Select p#.454*weight
From p;
```

\*\* هر پوند = ۴۵۴\*گرم

مثال ۵) شماره و وضعیت تولیدکنندگان واقع در تهران را بر اساس نزول مقادیر وضعیت مشخص کنید.

```
Select s#,status
From s
Where city='تهران'
Order by status DESC
```

\*\* صعودی = ASC ، نزولی = DESC ، پیش فرض صعودی است.

**نکته:** در هنگام استفاده از دستور order by می توان به جای نام ستون شماره آن را نیز ذکر کرد. ستون ها از چپ به راست و از عدد ۱ شماره گذاری می شوند. مثال بالا را با همین روش می زنیم.

```
1 2
Select s# , status
From s
Where city='تهران'
Order by 2 DESC
```

مثال ۶)

```
Select *
from p
where weight between 10 and 15;
یا
Select *
from p
where weight >=10 and <=15;
```

مثال ۷)

```
Select *
From p
Where weight in(12,13,16);
یا
Select *
From p
Where weight =12 or weight =13 or weight =16;
```

مثال ۸)

الف) مشخصات قطعاتی را بیابید که نام آن ها با حرف c شروع شود.

```
Select * from p where pname like 'c%';
```

access = % در \*

ب) مشخصات قطعاتی را بیابید که سه حرفی بوده حرف دوم آن c باشد.

```
Select * from p where pname like '_c_';
```

access = \_ ؟ در

ج) مشخصات قطعاتی را بیابید که در بین اسم آن ها حرف c نباشد.

```
Select * from p where pname notlike '%c%'
```

مثال ۹)

نام تولیدکنندگانی را مشخص نمایید که کالای p2 را تولید کرده باشند.

(روش اول)

```
((sp where p# = 'p2')[s#]join s)[sname]
```

(روش دوم)

```
Select sname from s.sp(s*sp)
Where s.s# = sp.s# and sp.p# = 'p2';
```

Select متداخل:

مثال ۱۰) با روش select متداخل

```
Select sname
From s
Where s# in(select s# from sp
Where p# = 'p2');
```

مثال ۱۱) نام تولیدکنندگانی را تعیین کنید که کالایی با رنگ قرمز را تولید کرده باشد.

```
Select sname
From s
Where s# in(select s# from sp
             Where p# in(select p# from p
                         Where pcolor='قرمز'));
```

مثال ۱۲)

Select p# from sp ;                      select **distinct** p# from sp;

S#	P#	qty
S1	P1	20
S2	P2	30
S1	P3	20
S2	P2	23
S3	P1	20
S3	P3	30

P#
P1
P2
P3
P2
P1
P3

P#
P1
P2
P3

\*\*\* distinct تکراری ها را حذف می کند.

مثال ۱۳) شماره قطعاتی را بیابید که وزن ان ها بیش از ۱۰ باشد یا توسط شرکت s2 تهیه شده باشد.

```
Select p#
From p
Where weight > 10
Union(اجتماع)
Select p#
From sp
Where s# = s2;
```

\*\* به خاطره این از **or** استفاده نکردیم چون با دو جدول سروکار داشتیم.

استفاده از توابع sql:

تابع **count**: این تابع تعداد رکوردهای یک جدول را می شمارد ( حتی اگر سطر تکراری وجود داشته باشد) در هنگام استفاده از تابع **count** باید از گزینه ی **distinct** استفاده نماییم اما در هنگام استفاده از **count(\*)** نوشتن گزینه ی **distinct** مجاز نمی باشد.

تابع **sum()**: این تابع مجموع یک فیلد عددی را بر می گرداند.  
 تابع **avg()**: این تابع میانگین یک فیلد عددی را بر می گرداند.

تابع `min()`: این تابع حداقل مقدار یک فیلد عددی را بر می گرداند.  
تابع `max()`: این تابع حداکثر مقدار یک فیلد عددی را بر می گرداند.

مثال ۱۴) کل عرضه ی انجام شده را مشخص کنید.

```
Select sum (qty) from sp;
```

مثال ۱۵) کل عرضه ی انجام شده توسط شرکت s1 را مشخص کنید.

```
select sum(qty) from sp where s# = 's1';
```

مثال ۱۶) کل عرضه ی انجام شده را به تفکیک شماره کالا مشخص کنید.

```
Select p#,sum(qty) from sp
```

```
Group by p#;
```

\*\* در این مثال `group by` بر حسب `p#` دسته بندی می کند . تنها فیلدهایی جلوی `group by` می توانند بیابند که در جلوی `select` آمده باشند.

---

با توجه به جدول زیر به سوالات مطرح شده پاسخ دهید.

Taken (ID,NO,term,year,mark)

۱- تعداد درس هایی که دانشجو با شماره ۷۰۰ در آن ها قبول نشده است را مشخص کنید.

```
Select count(distict NO) from taken
```

```
Where ID = 700 and mark <10;
```

۲- بیش ترین نمره و کمترین نمره در درس با کد ۰۱ را در ترم دوم سال ۸۷-۸۸ مشخص کنید.

```
Select min(mark),max(mark)
```

```
From taken
```

```
Where NO='01' and term ='2' and year ='88-87';
```

۳- شماره دانشجویانی را تعیین کنید که نمره ی آن ها در درس با کد ۰۱ برای ترم اول سال ۸۷-۸۸ هنوز اعلام نشده باشد.

```
Select ID
```

```
From taken
```

```
Where NO='01' and term ='1' and year ='88-87'
```

```
And mark='is null';
```

مثال ۱۷) حل مثال ۱۱ با روش دیگر:

```
Select sname
From s,p,sp
Where p.p# = sp.p#
And sp.s#=s.s#
And p.color='قرمز';
```

دستور insert:

شکل استفاده از این دستور به ۲ صورت زیر است.

۱) insert into table name values(value1,value2,...) (حالت ۱)

۱) insert into s values's4','الومین','اصفهان','تهران' (مثال حات ۱)

۱) insert into p(p#,city) values('p4','تهران') (مثال حات ۱)

۲) insert into table name suboary(نتیجه ی یک پرسجو) (حالت ۲)

۲) insert into temp (مثال حلت ۲)

```
Select p#,sum(qty)
```

```
From sp
```

```
Group by p#;
```

(شکل حالت ۲)

جدول temp

جدول sp

S#	P#	Qty
S1	P1	100
S1	P2	200
S2	P2	300
S3	P3	200

S#	Sum(qty)
P1	100
P2	500
P3	200

دستور update: شکل کلی

```
Update table name set field1=value,[field2=value]
```

```
[where condition)
```

مثال) رنگ قطه p2 را به مقدار زرد تغییر داده و به وزن ان ۵ واحد اضافه کنید.

```
Update p set pcolor='زرد', weight=weight+5
```

```
Where p#='p2';
```

مثال) شماره ی تهیه کننده ی s1 را به مقدار s5 تغییر دهید.

```
Update s set s#='s5' where s#='s1';
```

دستور delete: شکل کلی:

Delete from table name  
[where condition(s)]

delete from sp where s#=s3 and p#='p3'; (مثال)  
( فقط داده های فیلدها را پاک می کند نه خود جدول را ) delete from sp: (مثال)

دیدگاه (view): جدولی است مجازی برای مخفی سازی اطلاعات و برقراری امنیت از آن استفاده می کنیم. شکل تهریف دیدگاه به صورت زیر می باشد.  
Create view نام دیدگاه as suboary;

انواع دیدگاه

- ۱- دیدگاه سطری
- ۲- دیدگاه ستونی
- ۳- دیدگاه سطری و ستونی
- ۴- دیدگاه اماری
- ۵- دیدگاه الحاق

دیدگاه سطری: انتخاب کلیه ستون ها با ذکر شرط.

create view test (مثال)  
as select \* from s where city='تهران';

S#	sname	city	Status
S1	فن اوران	تهران	100
S2	پولادین	تهران	150

دیدگاه test

\*\* user فقط به این دو سطر دسترسی دارند و به سطر های بعدی از جدول s دسترسی ندارند.

دیدگاه ستونی: انتخاب بعضی از ستون ها بدون ذکر شرط.

create view test as select p#,pname from p; (مثال)

P#	pname

دیدگاه test

\*\* user فقط به p# و pname دسترسی دارد و در جدول p به بقیه فیلد ها دسترسی ندارد.

دیدگاه سطری و ستونی: انتخاب بعضی از ستون ها با ذکر شرط.

```
create view test
As select s#,p#
From sp where qty>300;
```

دیدگاه اماری: در بیان این دیدگاه از توابع sql استفاده می کنیم.

```
create view test(PNO ,max-qty)
As select p#,max(qty)
From sp
Group by p#;
```

PNO	Max-qty

\*\*\* test(PNO,max-qty) یعنی این که فیلد های جدول test، caption، شان PNO و max-qty .

دیدگاه الحاق:

```
Create view test
As select s.*,p.*(همه ی فیلد ها از ۲ جدول)
From s,p
Where s.city=p.city
```

---

استفاده از **having** در **group by**: عبارت **having** همواره با **group by** استفاده می شود، نقش **having** در **group by** مانند نقش **where** در سطر است. به عبارت دیگر از **having** برای در نظر گرفتن **group** ها استفاده می شود همان طور که از **where** برای در نظر گرفتن سطرهایی در جدول استفاده می شود.

(مثال) شماره ی قطعه ی تمام قطعاتی که توسط بیش از یک تهیه کننده تهیه شده است را بیابید.

```
Select p#
From sp
Group by p#
Having count(*)>1;
```

مثال) شماره تهیه کنندگانی را مشخص کنید که بیش از ۳ قطعه تهیه کرده باشد.

```
Select s#  
From sp  
Group by s#  
Having count(s#)>3;
```

موفق باشید