

به نام خدا

# Data Base

## بانکهای اطلاعاتی

### (پایگاه داده)

#### تاریخچه بانکهای اطلاعاتی:

##### نسل‌های ذخیره‌سازی Data base

##### ۱- نسل اول (فایل‌های ساده ترتیبی)

- فایل‌های ساده ترتیبی مانند نوار مغناطیسی
- فایل فیزیکی و فایل منطقی یکی است دارای ساختار یکسانی بودند.
- برای ایجاد عملیات بهنگام‌سازی فایل دیگری (پدر) ایجاد می‌شود و فایل قبلی به نام old ذخیره شده است. به همین دلیل تکرار ذخیره سازی بسیار زیاد است.
- ساختار فایل ترتیبی بوده است
- نرم افزاری برای مدیریت وجود نداشته است.
- طراحی مشخصات فیزیکی بر عهده کاربر است.
- هر گونه تغییر در ساختار داده یا رسانه ذخیره سازی باعث تغییر در برنامه می‌شود.

##### فایل فیزیکی:

فایل فیزیکی فایلی است که طراحی می‌شود و در پایین ترین سطح ممکن قرار دارد.

##### فایل منطقی:

فایل منطقی دیدی از همان فایل فیزیکی است که در بالاترین سطح یعنی همان کاربر عادی می‌باشد.

##### ۲- نسل شیوه‌های دستیابی Access method

در این نسل حافظه های جانبی از نوع دیسک اختراع شد به همین دلیل شیوه ی دست یابی به صورت مستقیم شد و نه ترتیبی

با بوجود آمدن رسانه های مستقیم این نسل بوجود آمد.

- نرم افزار واسطی جهت شیوه دستیابی A.M
- جدا شدن نسبی فایل منطقی از فیزیکی
- کاربر همچنان درگیر مسائل فیزیکی است
- افزونگی بالا است.
- نرم افزاری برای مدیریت سیستم وجود ندارد.

### ۳- نسل سوم: نسل مدیریت داده‌ها

- نرم افزار کاملتری از نرم افزارهای شیوه دستیابی به عنوان واسط برنامه کاربردی فایل‌های محیط فیزیکی ایجاد شد.
- فایل‌های منطقی متعددی می‌توانند از داده‌های فیزیکی مشترک استفاده کنند.
- کاربران در يك محیط انتزاعی (Abstract) کار می‌کنند.
- صحت داده‌های ذخیره شده تا حدی تامین می‌شود.
- ترکیبی از انواع ساختارهای فایل به کار گرفته می‌شود.
- بازیابی به کمک چند کلید (Multi key retrieval) امکان پذیر است.

### ۴- نسل چهارم: (DBMS) data base manager system

- در این نسل نرم افزار پیچیده‌ای به نام سیستم مدیریت بانک اطلاعاتی (dbms) واسط است بین برنامه‌های کاربردی و محیط فیزیکی
- در این نسل برنامه‌های کاربردی از جنبه‌های و خصوصیات محیط فیزیکی ذخیره سازی مستقل شده اند که اصطلاحاً به آن استقلال داده فیزیکی می‌گویند.
- چند سطحی بودن بانک‌های اطلاعاتی و معماری ANSI
- مهمترین ویژگی این نسل چند سطحی شدن معماری بانک است . access, sql server ,oracle و غیره که همگی dbms هستند .

### ۵- نسل پنجم: نسل معروف (knowledge base)

- استنتاج - هوش مصنوعی - سیستم‌های خبره- پردازش زبان طبیعی
- Entity:** (موجودیت) هر شیء یا هر چیزی را که بخواهیم اطلاعاتی را روی آن ذخیره کنیم .
- صنعت خاصه: خصوصیات يك موجودیت را صفات خاصه می‌گویند ذخیره کنیم مانند نام و شماره دانشجویی.

Information: داده‌ها پس از اینکه مورد تفسیر قرار گرفتند تبدیل به اطلاعات می‌شوند

### رابطه (Relation ship)

رابطه بین موجودیت Relation ship

نکته: موجودیتها رابطه دارند.

موجودیت ها می‌تواند باهم رابطه داشته باشند.

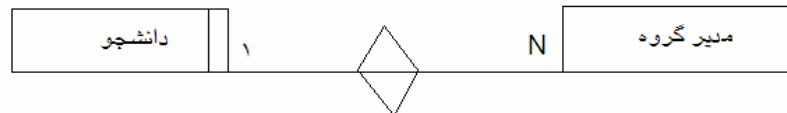
فیلد: فیلد همان صفات خاصه

رکورد: نمونه‌ای از یک Entity

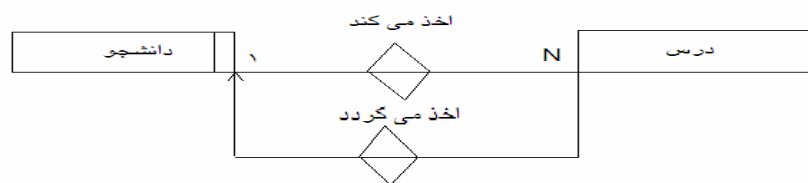
فایل: مجموعه‌ای از رکوردها که entity را شامل می‌باشند فایل گویند. ممکن است یک موجودیت با خودش رابطه داشته باشد.

## انواع رابطه Relation ship

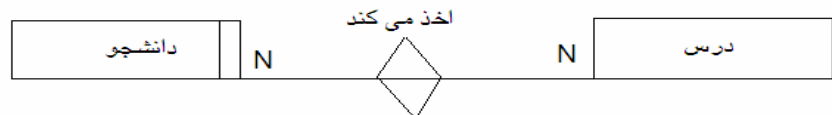
رابطه یک به یک



رابطه یک به چند



رابطه چند به چند ترکیبی از یک به چند و چند به یک می‌باشد



نکته: رابطه چند به چند قابل پیاده سازی نیست.



برای نمایش رابطه بین Entity ها از نمودارهایی به نام ER استفاده می شود. فضای اطلاعاتی: فضای اطلاعاتی ۳ بعدی است. به صورت روبه‌رو: مقدار، صفت خاصه، موجودیت

دام ارتباط (Connection trap)

S2 قطعه p3 را تهیه می‌کند

قطعه p3 در پروژه 4 بکار می‌رود.

S2 برای پروژه 4 قطعه تهیه می‌کند.

S2 قطعه p3 را برای پروژه 4 (غ).

استنتاج غلطی که از نمودار ER بوجود می‌آید دام ارتباط می‌گویند.

ممکن است با استفاده از نمودارهای ER استنتاج های غلط استخراج شود.

### صفت خاصه ی مشترک:

در بسیاری از موجودیت های یک بانک صفات خاصه ی مشترکی پیدا می شود که این صفات را به ارث می برند. مثلا در دانشگاه دانشجو و مدرس و درس دارای نام هستند.

### تعریف data base :

بانک اطلاعاتی مجموعه ای است از داده های ذخیره شده در مورد انواع موجودیت ها یا انواع entity ها یک محیط عملیاتی و ارتباط بین آنها به صورت مستمر و مبتنی بر یک ساختار تعریف شده به صورت صوری با حداقل افزونگی تحت کنترل متمرکز و مورد استفاده ی یک یا چند کاربر به طور اشتراکی وهمزمان. تعریف شده به صورت صوری : یعنی اینکه کاربر داده های خود را ان طور که می بیند ذخیره کند و یک دید انتزاعی باید داشته باشد.

به صورت مستمر و مبتنی بر یک ساختار: یعنی داده های عملیاتی در یک ساختار به خصوص ذخیره شود مجتمع بودن بانک باعث می شود افزونگی کاهش یابد. تحت کنترل متمرکز: یعنی اینکه داده ها توسط یک سیستم مرکزی کنترل شود که در این صورت امنیت سیستم کنترل شود.

### حسن و مزایای اشتراکی شدن :

- ۱- عدم افزونگی
- ۲- امنیت (کنترل متمرکز)
- ۳- استفاده ی بهینه و سرعت بالا در اطلاعات

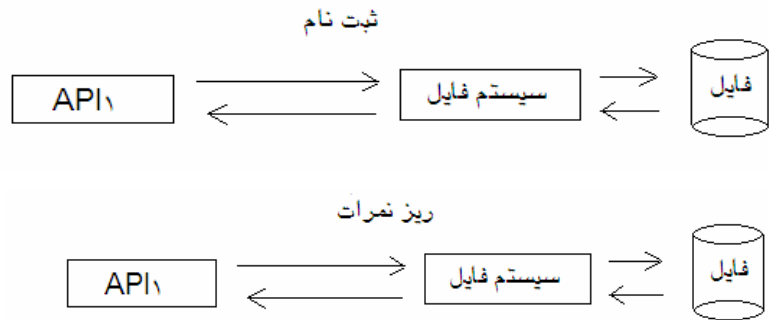
### روش ایجاد پایگاه داده:

#### ۱- فایل کلاسیک (غیربانکی)

#### ۲- روش بانکی

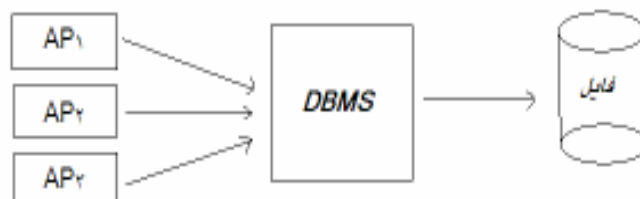
#### فایل کلاسیک:

در این روش برای هر برنامه ی کاربردی یک فایل وجود دارد و برای هر فایل نیز باید یک سیستم فایل نوشته شود. در این سیستم تجمع داده وجود ندارد در این سیستم ارث بری وجود ندارد در این روش اگر ساختار فایل تغییر کند برنامه ها نیز باید تغییر کند.



### بانکی:

در این روش وحدت ذخیره سازی وجود دارد ولی هر کاربر دید خاصی از داده ها دارد در این روش تغییر کردن فایل تاثیری بر روی برنامه های کاربردی ندارد به دلیل اینکه فقط یک فایل به وجود می آید داده ها مجتمع هستند در این روش مهمترین قسمت سیستم DBMS است.



## اجزاء و عناصر اصلی محیط بانک

### ۱- سخت افزارهای مورد نیاز

سخت افزار مورد نیاز بستگی به نوع بانک دارد که معمولا موارد زیر مورد نیاز است:

- ۱- یک ماشین محاسبه گر که معمولا کامپیوتر است (سخت افزار پردازنده مرکزی)
- ۲- یک سخت افزار ذخیره سازی (حافظه های جانبی)
- ۳- در صورت امکان سخت افزار ارتباطی (برای بانکهای شبکه ای)

### ۲- نرم افزار:

نرم افزار کاربردی: کاربر با سیستم DB ارتباط برقرار می کند

نرم افزار سیستم: که همان DBMS ارتباط برقرار می کند

### ۳- کاربر: افرادی که با بانک در ارتباط هستند:

DA: مدیر داده: مسئول داده ها یک مدیر است نه یک فرد فنی

شخصی است که کنترل داده های سازمان را به عهده دارد این فرد مشخص می کند که سازمان باید چه داده هایی داشته باشد. رابطه بین داده ها را درک میکند جدای از اینکه داده ها دستی باشند یا کامپیوتری این فرد یک مدیر است و نه یک فرد فنی

DBA: ( به مدیریت Data base گفته می شود که اداره کننده بانک می باشند)

مدیر بانک اطلاعاتی یک شخص فنی است که مسئول پیاده سازی تصمیمات مدیر داده است.  
DBP: همان برنامه نویسی است.

*End user*: کاربري که با سیستم کار می‌کنند.

۴- داده‌ها: اطلاعاتي که وارد می‌شود را داده می‌گویند.

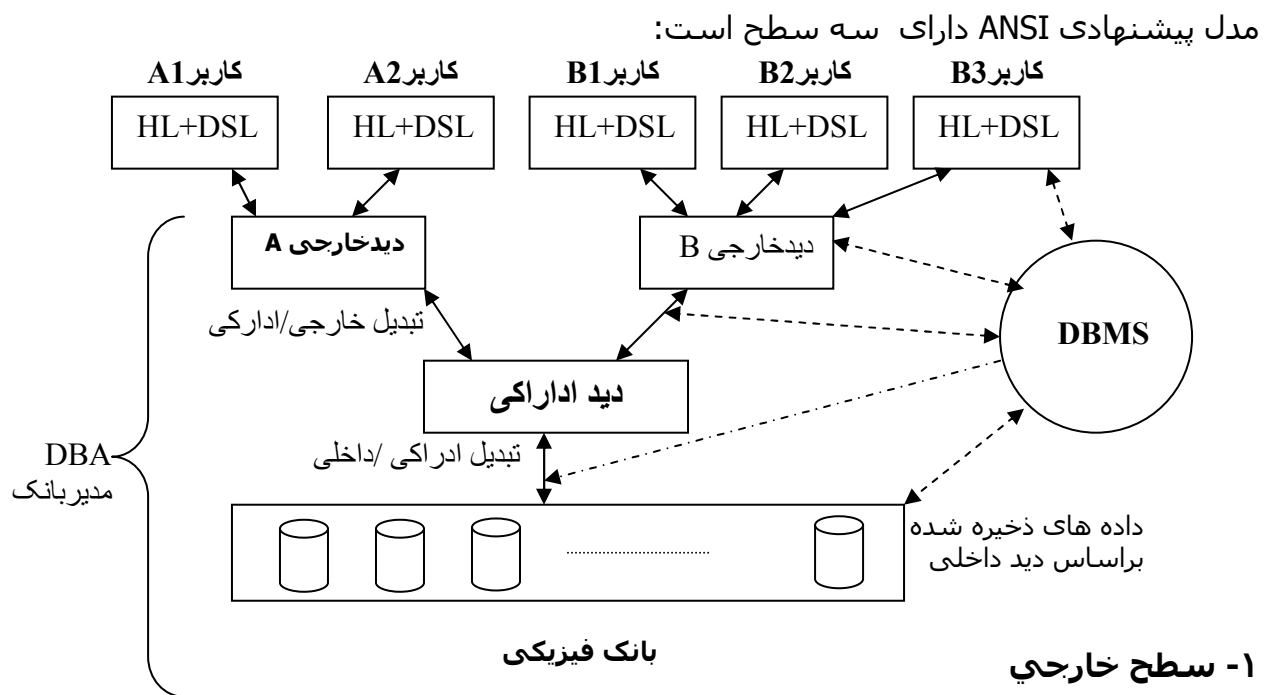
**بانک‌های اطلاعاتي توزیع شده:** بانک‌های اطلاعاتی که در شبکه قرار می‌گیرند بانک‌های اطلاعاتی

توزیع شده می‌گویند. انواع بانک‌های اطلاعاتی توزیع شده:

۱- بانک‌های اطلاعاتي با داده‌های توزیع شده

۲- بانک‌های اطلاعاتي با داده و سیستم توزیع شده

## معماري سیستم بانک اطلاعاتي:



۱- سطح خارجی

۲- سطح اداری (مفهومی)

۳- سطح داخلی

**سطح اداری:**

دید طراح بانک است از داده‌های ذخیره شده در بانک در سطح مفهومی (اداری) طراح بانک انواع موجودیت‌ها، ارتباط بین آن‌ها، انواع فیلدها و نوع آن‌ها (میدان)، و دیدهای خارجی را تعریف می‌کند. در این سطح DSL مفهومی وجود دارد یعنی طراح به وسیله DDL بانک را طراحی می‌کند و برای کنترل کردن داده‌ها از DCL استفاده می‌کند در این سطح طراح بانک از چگونگی ذخیره‌سازی و محیط گرافیکی مستقل است. نکته بعدی اینکه در این سطح طراح باید مدل بانک یا همان ساختار داده را مشخص کند. مثلاً در مدل رابطه‌ای ساختار آن جدول است، یعنی هر موجودیت را با یک جدول معرفی می‌کنیم.

**سطح خارجی:**

دید خاص کاربر است از داده‌های ذخیره شده در بانک و یک دید انتزاعی است و *End user* ها این سطح را

می‌بینند

سطح کاربر خاص است و کاربر با آن کار می کند، هرکاربر دارای دو زبان HL و DSL است که به این وسیله دو زبان را می تواند محاسبات را انجام دهد و سئوالات خود را از بانک بپرسد. در سطح خارجی کاربر در مورد داده ها یک تفکر انتزاعی دارد، هر کاربر می تواند دید خارجی مخصوص به خود داشته باشد، هر کاربر می تواند از داده هایی که ذخیره شده است دید خاصی داشته باشد که البته این دید را در سطح مفهومی (ادراکی) طراح بانک ایجاد می کند.

**زبان برنامه نویسی HL :** Hy Language (زبان میزبان)

این زبان یک زبان سطح بالای برنامه نویسی است که محاسبات و برنامه نویسی غیر بانکی با آن انجام می شود مانند زبان ویژوال بیسیک که در Access استفاده می شود.

**زبان برنامه نویسی DSL :**

یک زبان سطح بالاست که معمولاً به صورت میهمان در کنار HL قرار می گیرد و به وسیله این میزبان می توان:

الف) داده ها را تعریف کرد DDL

ب) با داده ها کار کرد DML

ج) داده ها را کنترل کرد DCL

مثلاً زبان SQL یک DSL است که در Access وجود دارد.

**اصل وحدت عملکرد:**

این اصل یعنی اینکه برای انجام یک عمل مشخص در سطوح خارجی و ادراکی فقط یک دستور وجود داشته باشد این اصل باعث می شود بین سطح خارجی و سطح ادراکی مترجمی وجود نداشته باشد.

نکته: کاربران مختلف در سطوح های مختلف دیده های متفاوتی از یک جدول خواهند داشت.

**سطح داخلی ( سطح فایلینگ ) :** در این سطح در واقع فایل های فیزیکی تعریف می شوند و

نحوه دسترسی اعلام می شود در واقع این سطح از معماری مانند سیستم های غیر بانکی هستند و هیچکدام از کاربران بانک اطلاعاتی اساساً با این قسمت کاری ندارند.

در این سطح در واقع فایل های محیط فیزیکی تعریف می شود. مانند محتوا، ساختار و نحوه دست یابی

نکته: طراح بانک دخالت چندانی در سطح داخلی ندارد البته سطح داخلی با سطح فیزیکی فرق میکند.

در سطح داخلی محل رکورد ، توالی رکوردها، تخصیص فضای ذخیره سازی و تکنیک هایی برای فشردن سازی و رمزگذاری داده ها تعریف می شود.

سطح ادراکی از سطح داخلی مستقل است.

**تبدیلات بین سطوح (سطح داخلی / ادراکی، سطح خارجی / ادراکی)**

تبدیلات بین سطوح:

۱- تبدیل داده ای

۲- تبدیل احکام

۳- تبدیل ساختار

۱. **تبدیل داده ای:** یعنی اینکه داده های تعریف شده در سطح خارجی به داده های تعریف شده در سطح

ادراکی تبدیل شود و برعکس همچنین باید داده ها در سطح ادراکی به سطح داخلی تبدیل شوند و برعکس

۲. **تبدیل احکام** : یعنی اینکه دستورات از سطح خارجی به دستورات سطح ادراکی تبدیل شوند و از سطح ادراکی به سطح داخلی و برعکس در اینجا وجود **ندارد**.

۳. **تبدیل ساختار**:

a. **تبدیل خارجی / ادراکی**

b. **تبدیل ادراکی / داخلی**

a. **تبدیل خارجی / ادراکی** : مکانیزمی برای برقراری تناظر بین دیده های خارجی مختلف و دیدادراکی است ، این وظیفه DBMS است. این نگاشت باعث می شود تعادلات موجود در سطح خارجی بدون هیچ تغییری در سطح ادراکی پیاده سازی شوند.

تغییراتی که در سطح ادراکی به وجود می آیند نباید در سطح خارجی اثر بگذارد. این تبدیل یعنی اینکه اگر در سطح ادراکی تغییراتی اعمال شد ، سطح خارجی تغییر نکند که این باعث استقلال داده منطقی می شود. مثلاً ممکن است در سطح ادراکی اطلاعات ساختار جدولی داشته باشند اما در سطح خارجی اطلاعات را بصورت گراف به کاربر نشان می دهیم حال اگر ساختار جدول ما تغییر کند گراف خارجی نباید تغییر کند.

b. **تبدیل ادراکی / داخلی** : اگر در سطح داخلی تغییراتی ایجاد شود ، سطح مفهومی (ادراکی) نباید تغییر کند که این باعث استقلال داده فیزیکی خواهد شد. در اصطلاح به تبدیلات - چه داخلی و چه خارجی- mapping (نگاشت) می گویند و همیشه تغییرات در سطح پایینی به وجود می آید و نباید در سطح بالایی اثر بگذارد.

### **استقلال داده‌ای:**

۱- **استقلال داده‌ای فیزیکی**: تغییرات فیزیکی به وجود آمده در سطح داخلی نباید بر سطح ادراکی اثر بگذارد.

۲- **استقلال داده‌ای منطقی**: تغییرات در سطح ادراکی نباید در سطح خارجی اثر بگذارد.

### **زبانهای برنامه نویسی:**

**HL: (زبان میزبان)** يك زبان سطح بالا محسوب می شود که داخل آن *DSL* را پیاده سازی می کنند.

**DSL: (زبان فرعی داده‌ای): SQL** نوعی *DSL*

### **اصل وحدت عملگر:**

یعنی برای انجام يك عمل مشخص در سطوح خارجی و ادراکی دستور واحدی وجود داشته باشد

### **خصوصیات DSL:**

- **اصل وحدت عملگر باید در DSL رعایت شود.**

اصل وحدت عملگرها در يك سطح مشخص نیز مطرح است مثلاً هم برای درج داده ها و هم برای درج ارتباط بین آنها می بایست حکم واحدی وجود داشته باشد.

- **DSL باید داده های مختلف کاربران را بپذیرد .**



**نکته:** DSL يك زبان بياني است كه در كاربر درآن مي گويد چه مي خواهيد ولي رويه انجام كار را بيان نمي كند بر عكس c و پاسكال كه رويه (procedural) هستند و کاربران بايد رويه انجام كار را بيان كنند.

**زبان HL:** يك زبان برنامه نويسي مانند C و پاسكال مي تواند باشد زبان سطح بالايي كه احكام DSL در آن ادغام مي شود نيز زبان داده ميزبان مي گويند.

### انواع DSL:

**DSL مستقل:** نياز به زبان ميزبان ندارد.

**DSL نامستقل:** به زبان ميزبان نياز دارد.

**مزيت هاي به وجود آمدن سه سطح معماری:**

۱. به وجود آمدن کاربران خارجي مختلف با ديدهاي متفاوت
۲. ايجاد استقلال داده (چه منطقي و چه فزيكي)
۳. برقراري قوانين جامعيت

### كاتالوگ سيستم:

هر DBMS بايد شامل يك كاتالوگ باشد . كاتالوگ بانك (شبه داده) شامل:

الف) تمام ديدهاي خارجي و ادراكي

ب) تمام نگاشت ها mapping

ج) موجوديت ها و صفات خاصه آن ها و ميدان هر کدام از آنها

د) ارتباط بين موجوديت ها و index آن ها

و) تاريخ ايجاد بانك اصلاح بانك و ويرايش بانك تغيير بانك

د) واحدهاي اندازه گيري

ه) همه مشخصاتي كه در سيستم مفهومي تعريف شده است

به داده هاي موجود در كاتالوگ (ديكشنري) داده meta data مي گويند. درواقع كاتالوگ بانكي است از بانك طراحي شده ؛ كاتالوگ امكاني است براي كنترل و نگهداري بانك و در صورت نياز توسعه دادن آن در طول حيات آن.

### مزايای سيستم بانك اطلاعاتي :

**۱- مدلینگ داده هاي عملياتي بر اساس ساختار آن ها:** وجه تمايل سيستم هاي اطلاعاتي و غير اطلاعاتي اين است كه باعث مي شود کاربران ديده انتزاعي از داده هاي ذخيره شده داشته باشند. وجود ساختار داده هاي مانند مدل رابطه اي -شبكه اي سلسله مراتبي و و مدل شئي گرايي اين امكان را فراهم مي سازد

### ۲- وحدت ذخيره سازي كل داده هاي محيط عملياتي:

وجود سطح ادراكي اين امكان را فراهم مي كند منظور از وحدت ذخيره سازي داده ها اين است كه همه داده هاي مربوط به يك Entity در يك ساختار ذخيره مي شود.

**۳- اشتراكی شدن داده ها:** سيستم DBMS امكان مي دهد تا کاربران از داده هاي ذخيره شده به طور اشتراكی استفاده كنند و هر كاربر ديده مخصوص به خود داشته باشند مثلاً ممكن است

در مدل رابطه‌ای يك جدول داشته باشیم اما کاربران مختلف این جدول را به دیدگاه های مختلف ببینند در صورتی که يك فایل فیزیکی بیشتر وجود ندارد . (تعریف view)

#### ۴- کاهش میزان افزونگی:

وحدت ذخیره سازی باعث کاهش میزان افزونگی می‌شود.

۵- **تعدد شیوه‌های دستیابی و تسهیل دستیابی به داده‌ها:** وجود سطح ادراکی و نگاه‌های آن بین سطوح خارجی/ داخلی باعث می‌شود برای دستیابی داده‌ها از شیوه‌های متعدد دستیابی استفاده می‌شود (شیوه‌های دستیابی را مدل‌ها تعیین می‌کنند)

#### ۶- عدم وجود ناسازگاری داده‌ها: Inconsistency

منظور از ناسازگاری این است که افزونگی در بانک وجود داشته باشد و هنگام به هنگام سازی بعضی از داده‌ها را ویرایش کنیم و مابقی بدون ویرایش بماند مثلاً معدل دانشجو در ۲ جای مختلف ذخیره شود و هنگام تغییر کردن فقط یکی از آنها را تغییر دهیم با حذف افزونگی ناسازگاری نیز از بین می‌رود. البته تعریف میدان بر روی هر فیلد باعث جلوگیری از ناسازگاری می‌شود.

ناسازگاری هنگامی بروز می‌کند که بنا به دلیلی يك فقره اطلاع در بیش از يك نقطه از بانک ذخیره شود و لازم باشد که به هنگام درآید .

#### ۷- تأمین جامعیت:

**تعریف جامعیت:** بی نقص بودن داده‌ها جامعیت گویند

مفهوم میدان جامعیت را تأمین می‌کند مثلاً نمره منفی برای يك دانشجو نشان دهنده نقص بانک است یا دانشجویی که ثبت نام هویتی نداشته اما انتخاب واحد داشته است.

#### ۸- تأمین استقلال داده‌ای:

دو نوع استقلال داده وجود دارد استقلال داده فیزیکی و منطقی. استقلال داده یعنی مصونیت برنامه‌های کاربردی در قبال تغییراتی که در سطوح ادراکی و داخلی و فیزیکی ایجاد می‌شود. کلید استقلال داده فیزیکی نگاشت داخلی ادراکی است کلید استقلال داده منطقی نگاشت خارجی ادراکی است .

۹- **حفظ محرمانگی اطلاعات:** برای حفظ محرمانگی اطلاعات می‌توان از روش‌های مختلفی مانند کنترل دستیابی‌ها، ذخیره‌سازی داده‌ها بطور رمزی و یا رمزگذاری روی بانک‌ها استفاده کرد.

#### ۱۰- تسریع در دریافت پاسخ پرس جوها:

#### معایب بانک اطلاعاتی :

۱- به دلیل اینکه داده‌ها متمرکز هستند ممکن است داده‌ها به خطر بیفتند و راه حل آن هم پشتیبان‌گیری است .

۲- به دلیل اینکه داده‌ها متمرکز هستند ممکن است جامعیت داده‌ها به خطر بیفتد .

۳- سخت افزار اضافه ممکن است نیاز باشد . مانند UPS یا تجهیزات شبکه

۴- برنامه نویس و پیاده‌سازی تمام مفاهیم بانک اطلاعاتی پیچیده است .

**case : computer Aided software Engineering**

بعضي از DBMS داري ابزار كمكي (computer aided software engineering) هستند كه در طراحي و پياده سازي بانك مورد استفاده قرار مي گيرند از جمله مورد استفاده اين ابزارها مي توان از رسم نمودار EER ، كار با لغت نامه ها، تعريف شمائي بانك، تهيه نمودارها و گزارش ها تغيير شيء گرا و رسم نمودارها و مهندسي وارون كه گزارش طراحي را از سيستم ها پياده شده استخراج مي كند. نام برد، حتي ابزارهاي ويژه اي عرضه شده اند كه طراحي مفهومي عام مثل EER را گرفته و طراحي مفهومي خاص را در مدل رابطه توليد مي كنند.

**ساختار داده اي:**

به كمك ساختار داده اي مهمترين سطح بانك يعني سطح ادراكي تعريف مي شوند ساختار داده هاي متفاوتي تعريف مي شوند.

۱- مدل رابطه اي

۲- مدل سلسله مراتبي

۳- مدل شبكه اي

۴- مدل شيء گرايي

**هر مدلي سه قسمت دارد**

۱- **ساختار داده اي:**

۲- **عملگرهاي عمل كننده روي يك ساختار**

۳- **قواعد عام و براي تأمين جامعيت**

**ساختار رابطه اي:**

**تعريف:** مدل رابطه اي تعاريفي است كه در رياضيات براي رابطه وجود دارد  
نكته: در مورد رابطه اي هر زير مجموعه اي از ضرب كارتزين n ميدان است

**در مدل رابطه اي اساس كار چيست؟**

در مدل رابطه اي از ديد کاربر بانك اطلاعات به صورت جدول ذخيره مي شود. يعني ساختمان داده در اين مدل جدول است

مدل رابطه اي	ساختار جدولي
رابطه	جدول
تاپل	سطر
ميدان	مقادير مجاز در يك ستون
صفت خاصه	ستون

در این مدل طراحی موجودیتها را به صورت جدول نمایش می دهد بنابراین بانک تشکیل می شود از مجموعه ای از جداول .

برای برقراری ارتباط بین جداول ، جداول دیگری را طراحی می کنیم رابطه ها همان جداول هستند در این مدل برای بازیابی اطلاعات فقط به عملکرد سطرپای نیاز است .

### تعاریف در مورد مدل رابطه‌ای:

**درجه رابطه:** تعداد صفات خاصه رابطه درجه رابطه گفته می‌شود و همیشه ثابت است.

**کاردینالیتی:** به تعداد سطرهای جدول گفته می‌شود

تعداد تاپل‌های رابطه در يك لحظه از حیات آن کاردینالیتی رابطه است و متغیر است

**نکته:** در مورد رابطه‌ای سطح مفهومی کاملاً رابطه‌ای است و دید خارجی رابطه‌ای است ولی سطح داخلی رابطه ای نیست.

### خصوصیات رابطه:

- تاپل تکراری در رابطه وجود ندارد (به دلیل وجود کلید اصلی)

- تاپل‌ها نظم ندارند: به دلیل این که جدول به طور ناخودآگاه سطری دارند از جداول دقیقاً رابطه نیست. این ویژگی نشان می دهد که رابطه و جدول دقیقاً یکسان نیستند زیرا سطرهای جدول دارای ترتیب بالا و پایین هستند.

- صفت خاصه نظم ندارد ( ولی در جدول به طور اتوماتیک ستون ها نظم دارند)

- همه مقادیر صفت خاصه *A* تجزیه پذیر نیست: مقادیر در ستون‌ها باید اتمیک باشد یعنی قابل تجزیه به مقادیر دیگر باشد مانند روز ماه و سال

**تعریف 1NF:** رابطه‌است که خصوصیت تجزیه پذیر و نرمال شده را داشته باشد را *1NF* گویند. رابطه ای که دارای خصوصیت تجزیه ناپذیر باشد رابطه *normalize* گویند یا به عبارت دیگر تمام مقادیر صفت خاصه اتوماتیک باشند.

**نکته:** در مدل رابطه‌ای تمام رابطه‌ها باید نرمال باشد.

رابطه ایی که 1NF نباشد اصل وحدت عملگر رعایت نمی شود. مثلاً برای بازیابی همه عملگری که عنصری از يك مجموعه را بتواند بازیابی کند.

مثال) برای ذخیره کردن اطلاعات مربوط به قطعه و تهیه کننده و ارتباط بین آنها از جداول زیر استفاده می کنیم.

Name	S#	City	Status
Ali	100	A	12
Reza	102	M	25
Ali	103	B	16
Hasan	105	M	21

جدول S

ID	Name	Color	Qty
9004	K	Red	10
9008	Y	Blue	12
9007	X	Yellow	11

جدول P

S#	C#	Qty
102	9007	50
100	9007	20
102	9005	70

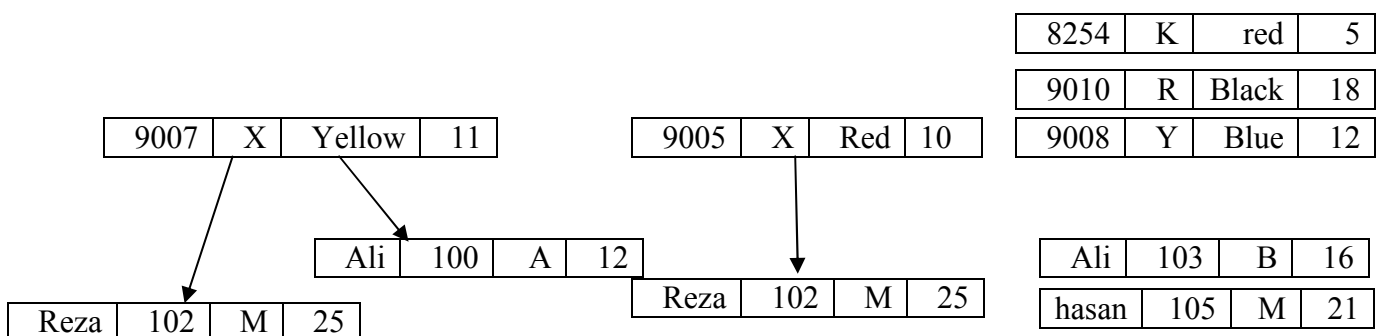
جدول SP

### ویژگیهای مدل رابطه ای

- ۱- دید کاربر بسیار واضح است و جدول محیطی مسطح و انتزاعی دارد .
- ۲- جداول دارای تئوری ریاضی می باشند .
- ۳- پیمایش جداول یا رابطه ها مستقل از جداول یا رابطه دیگر است .
- ۴- برای پاسخگویی به پرسش ها جستجو به صورت خطی انجام می شود.
- ۵- در این مدل داده ها و ارتباط بین آنها با مکانیزم واحدی نشان داده می شوند (جدول )
- ۶- برای طراحی بهینه رابطه ها قوانین تئوری نرمال سازی وجود دارد .
- ۷- افزونگی در مدل رابطه ای با توجه به قوانین نرمال سازی قابل حذف است .
- ۸- عملیات درج و حذف در مدل رابطه ای بسیار آسان است .

### مدل سلسله مراتبی:

این ساختار قدیمی داده ها و ارتباط بین آنها را به کمک یک درخت نمایش می دهد .  
روش سلسله مراتبی برای ارتباطات یک به چند بین موجودیها مناسب است .  
مثلا برای نمایش جداول بالا در مدل سلسله مراتبی از ساختار زیر استفاده می شود.



برای بازیابی نیاز به عملگر ریشه یاب (یا سطح یاب ) و عملگر وابسته یاب (فرزند یاب ) داریم.

### ویژگیهای شیوه سلسله مراتبی :

- تعریف: آنومالی یعنی اینکه از نظر اجرایی ناممکن یا دشوار است .
- ۱- عمل درج و حذف عناصر دارای آنومالی است (آنومالی یعنی دشواری انجام یک عمل در یک ساختار)
  - ۲- در این روش افزونگی بسیار بالاست .
  - ۳- پیمایش (غواصی) در سطوح پایین تر الزاما باید از سطوح بالا شروع شود .
  - ۴- در این روش کاربر وضوح دارد (یعنی مطلب را می فهمد) اما محیط انتزاعی آن مسطح نیست .
  - ۵- به هنگام سازی این روش بسیار مشکل است .
  - ۶- تئوری ریاضی در این روش وجود ندارد .
  - ۷- عملگرهای بازیابی به سادگی عملگر مدل رابطه ایی نیست.
- نرم افزار IMS ساختار سلسله مراتبی را استفاده می کند..

### مدل شبکه ای:

این مدل نمونه پیچیده ای از مدل سلسله مراتبی می باشد که به آن PLEX می گویند. در این مدل برای نمایش داده ها و موجودیت ها و رابطه بین آنها از گراف استفاده می شود. این مدل برای نمایش ارتباطات چند به چند بسیار مناسب است . چیزی به عنوان جهت در گرافها وجود ندارد . عمل بازیابی در مدل شبکه ای بسیار پیچیده تر از مدل سلسله مراتبی زیرا پیمایش گراف به دو روش سطحی و عمقی راههای متعددی دارد .

### خصوصیات محیط شبکه ای:

- ۱- از دید کاربر وضوح ندارد و محیط آن مسطح نیست .
- ۲- بدلیل استفاده از گراف برای ارتباط های چندبه چند مناسب است.
- ۳- عملگر بازیابی بسیار پیچیده است.
- ۴- اصل وحدت عملگر رعایت نمی شود. یعنی عملگرهای متفاوتی در سطوح مختلف نیاز است.
- ۵- عمل ذخیره سازی (درج حذف به هنگام سازی) آنومالی دارد
- ۶- اصل وحدت عملکرد رعایت نمی شود.

### مدل شیء گرایی :

در این روش برای هر موجودیت یک کلاس با طبقه ایجاد می شود که هر کلاس دارای خصوصیت و ویژگیهای خاصی است سپس عناصری را که می خواهیم ذخیره کنیم را عضو کلاس ها قرار می دهیم که در این صورت ارث بری از کلاسهای مختلف به وجود می آید .

### مفهوم میدان:

مقداری است که یک فیلد می تواند به خود بگیرد. مانند فیلد نمره میدان آن از صفر تا بیست است. به دلیل پیچیدگی پیاده سازی میدان برای بانک اطلاعاتی معمولا نرم افزار های بانک اطلاعات میدان را پیاده سازی نمی کند. و معمولا نرم افزارهای DBMS بعضی از میدانها را پیاده سازی می کند.

مانند پیاده سازی میدان برای آدرس دانشجویان

### نقش میدان در پایگاه اطلاعاتی:

در صورتی که میدان را به صورت واقعی پیاده سازی کنیم سه نوع کنترل بر روی بانک امکان پذیر می شود:

#### ۱- کنترل مقداری پرس و جوها:

مقادیر يك صنعت خاصه در طول حیات يك صفت خاصه از مقدار میدان برگرفته می شود به عبارت دیگر باید در میدان وجود داشته باشد (قاعده جامعیت خاص) مانند فیلد نمره که میدان آن بین صفر تا بیست است این پرسش که دانش آموزانی را استخراج کنید که نمره آنها بیش از بیست و دو است نادرست است.

#### ۲- کنترل سمانتیک (پرس و جوها):

ممکن است در صفت خاصه از يك نوع باشند (برای مثال عددی باشد) اما هر کدام ماهیت متفاوت باشد (مثلا شماره دانشجویی و شماره ی شناسنامه) این دو را نمیتوان را با یکدیگر مقایسه کرد مانند دانشجویی را پیدا کنید که شماره دانشجویی آن بزرگتر از شماره شناسنامه آن باشد این سؤال فاقد سمانتیک است.

#### ۳- تسهیل در پاسخگویی بعضی از پرس و جوها:

در بعضی از پرس و جوها سؤال در مورد رابطه ها یا خود بانک است مثلا در کدام جدول اطلاعاتی در مورد دانشجو است این سئوالی از کاتالوگ بانک است. یعنی اگر میدان ها در کاتالوگ بانک ذخیره شود این نوع سئوالهای پیشرفته را می توان پاسخ داد.

### مفاهیم مدل رابطه ای:

مدل رابطه ای توسط ریاضیدان معروف آقای کاد طراحی شده است.

تعریف رابطه همان تعریف ریاضی آن می باشد که دارای دامنه و برد است و چند تایی هایی مرتب را شامل می شود

هر رابطه از يك Heading و يك Body تشکیل می شود. معمولاً Heading ثابت است ولی body متغیر است .

Heading مجموعه اسمی صفات خاصه را Heading می گویند . Body مجموعه ای از چند تایی های مرتب که متغیرند.

رابطه هر زیر مجموعه ای از ضرب کارتین n میدان است.

### کلیدها در مدل رابطه ای :

مفهوم مدل رابطه ای بر اساس کلیدها گذاشته شده است کلیدها در جداول مورد استفاده می باشد .

**کلید کاندید:** candidate key مجموعه ای از صفات خاصه (فیلدها یا ستون ها)  $(A_i, A_j, \dots)$

An در رابطه (جدول) که دارای دو خاصیت ذیل می باشد:

۱- **منحصر بودن (Uniqueness):** بدین معنا که در هر لحظه از حیات رابطه باید یکتا

باشد و در هیچ دو تاپلی تکرار نشود.

۲- **کاهش ناپذیری (Minimality):** هیچ زیر مجموعه ای دیگر از صفات خاصه منحصر به فرد

نیست یا به عبارت دیگر اگر یکی از عناصر کلید کاندید را حذف کنیم یکتایی خود را از

دست می دهد.

**نکته:** هر رابطه حداقل یک کلید کاندید داشته باشند.

**نکته:** در بدترین حالت همه ی فیلدها کلید کاندید رابطه است به چنین حالتی  $All\_key$  گویند.

کلید کاندید را برای جدول زیر مشخص کنید؟

نام	معدل	محل صدور	شماره شناسنامه	سال تولد	کد ملی	شماره دانشجویی

**کلید اصلی (primary key):** کلید کاندیدی است که توسط طراح بانک انتخاب می شود. طراح

بانک باید از بین کلیدهای کاندید فقط یکی را برای کلید اصلی معرفی کند دو عامل برای این

انتخاب مهم است:

الف) نقش و اهمیت کلید اصلی نسبت به سایر کلیدهای کاندید مثلاً: شماره نظام پزشکی در

تشخیص پزشکان.

ب) کوتاهترین کلید کاندید را به عنوان کلید اصلی انتخاب می کنند.

کلید اصلی تنها مکانیزم نشان دهی آدرس یا آدرس دهی در سطح تاپل است به همین دلیل

وجود کلید اصلی الزامی است.

**کلید بدیل (ALTERNET KEY):**

هر کلید کاندیده غیر از کلید اصلی کلید بدیل نامیده است.

**کلید خارجی (Foreign key):**

صفت خاصه  $X$  (یعنی ستون) از رابطه  $R_2$  کلید خارجی است اگر  $X$  در رابطه  $R_1$  کلید اصلی

باشد.

صفت خاصه  $X$  از رابطه  $R_2$  کلید خارجی است.

اگر  $X$  در رابطه  $R_1$  کلید اصلی باشد در جدول انتخاب واحد  $C\#$ ,  $S\#$  هر کدام جداگانه کلید

خارجی محسوب می شود.

انتخاب واحد	
S#	C#
100	15
100	13
110	12
115	15
110	12

جدول دانشجویان		
S#	نام	فامیلی
		ی



جدول دروس		
C#	نام درس	تعداد واحد
۱۵	ریاضی	۳
۱۲	فیزیک	۲
۱۳	فارسی	۱

۱۰۰	X	Y
۱۱۰	T	F
۱۱۵	K	L

**نکته:** کلید خارجی نباید حتما کلید اصلی باشد( می تواند

در جدول خودش کلید اصلی باشد) کلید خارجی امکانی است که برای ارجاع از یک رابطه به رابطه دیگر استفاده می شود و باعث پیوند جداول می شود. اما کلید خارجی تنها راه ایجاد ارتباط بین رابطه نیست بلکه وجود صفت خاصه میشترک بین دو رابطه عاملی است برای نمایش ارتباط بین جداول.

تکلیف : برای یک کتابخانه یک بانک طراحی کنید که کتاب ها اعضای کتابخانه و ناشران و کارمندان کتابخانه ذخیره می شوند.

رابطه بین جداول و کلید اصلی و خارجی و کاندید و میدان هر فیلد را مشخص کنید.

## قواعد جامعیت در مدل رابطه ای ( integrity rule )

در مدل رابطه ای قواعدی باید وجود داشته باشد که بر اساس آن جامعیت و بی نقص بودن

بانک تضمین شود.

### ۱- قواعد جامعیت خاص:

قواعدی هستند که در یک سیستم مشخص و خاص وضع می شود. و می توان آن را به مفهوم میدان ارتباط داد. مانند نمره دانش آموزان در سیستم آموزشی. قواعد جامعیت خاص مربوط به یک بانک خاص بوده له همین دلیل در بانکهای مختلف قوانین متفاوتی دیده میشود. برای اینکه قواعد جامعیت خاص رعایت شوند باید میدان را به معنای واقعی پیاده سازی

کرد.

### ۲- قواعد جامعیت عام (موجودتی و ارجاعی)

• **قاعده جامعیت موجودیتی:** هیچ بخشی از کلید اصلی نباید دارای مقدار NULL

باشد.

**مقدار NULL :** مقدار خاصی است برای نمایش مقدار ناشناخته یا مقدار غیر قابل اعمال بکار می رود و یا جای خالی و صفر فرق دارد و به طور کلی **NULL** می تواند هر مقداری که در میدان نیست باشد .

دلیل این قاعده جامعیت موجودیتی این است که کلید اصلی نقش يك نمونه از موجودیت را دارد و هر نمونه از روی شناسه اش که کلید اصلی است شناسه مي شود حال اگر این شناسه **NULL** باشد به معنای این است که این موجودیت وجود ندارد.

#### • قواعد جامعیت ارجاعي:

اگر صفت خاصه  $X$  کلید خارجی باشد در رابطه  $R_2$  (بنابراین در  $R_1$  کلید اصلی است) این صفت خاصه در رابطه  $R_2$  :

(الف) میتواند مقدار تهی داشته باشد (به شرط اینکه خودش کلید اصلی نباشد)  
(ب) اگر مقدار تهی نداشته باشد حتماً باید مقداری داشته باشد که در سطری از رابطه  $R_1$  موجود باشد.

#### مدل رابطه ای باید حداقل دارای :

(الف) ساختار داده ای رابطه ای (جدول)  
(ب) قواعد جامعیت (ج) عملگرهایی برای کار با مجموعه ها (عملگرهای جبر رابطه ای مانند اجتماع - گزینش - پرتو - پیوند و غیره باشد).

## جبر رابطه ایی

### عملگرهای جبر رابطه ای :

به مجموعه ای از قوانین و عملگرهای که امکان پردازش جداول را فراهم می سازند **جبر رابطه ای** می گویند.  
عملگرهای جبر رابطه ای دارای خاصیت بسته بودن هستند یعنی خروجی هر رابطه ای يك جدول است.

بانك مدل رابطه اي داراي زبان كار ب داده هاست و زبان زبان كار با داده ها مجموعه اي است از عملگرها كه بر روي جداول كار ميكنند اين عملگرها رابطه را به عنوان عملوند گرفته و يك رابطه را به عنوان نتيجه بر مي گردانند.

### انواع عملگرها :

۱- مجموعه اي

۲- رابطه اي (ساده)

عملگرهاي مجموعه اي بر روي رابطه هايي اعمال مي شوند كه نسبت به عملگر اجتماع با يكديگر سازگار باشند يعني درجه آنها يكسان بوده (تعداد ستون ها ) و صفت خاصه / م از هر رابطه از يك ميدان باشند.

**عملگر اجتماع :** اجتماع دو رابطه ، رابطه اي است كه تايل هاش در يك يا هر دو رابطه وجود دارد ركورد هاي تكراري حذف مي شوند.

$$R_3 = R_1 \text{ UNION } R_2 \quad \text{مانند:}$$

**عملگر اشتراك :** اشتراك دو رابطه اي است كه تايل هاش در هر دو رابطه وجود داشته باشند مانند

$$R_4 = R_1 \text{ INTERSECT } R_2$$

عملگرهاي اجتماع و اشتراك خاصيت جابجايي و شركت پذيري دارند اما تفاضل اين خاصيت ها را ندارد.

### ضرب كارتزين :

حاصل ضرب دو رابطه اي است كه تايل هاش اثر الساق هر يك از دو تايل دو رابطه به دست مي آيد.

$$R_3 = R_1 \text{ TIMES } R_2$$

به طور مثال :

$R_1$		$R_2$		$R_3$		$R_1$		$R_2$
A		X		A A		A	۱	X ۱
B		Y		A Y		B	۲	Y ۲
C				B X		C	۳	Z ۳
				B Y				
				C M				
				C Y				

### عملگر گزينش (محدوديت) : RESTRICT :

شرط  $SELECT R_1 WHERE$

این عملگر از بین سطرها تعدادی را انتخاب می کند

$SELECT R WHERE C_1 OR C_2$

$(SELECT R WHERE C_1) UNION (SELECT R WHERE NOT C_2)$

$SELECT R WHERE NOT C_1$

$SELECT R WHERE NOT C_1$

$SELECT R MINUS (SELECT R WHERE C_1)$

### (عملگر تصویر یا پرتو) $PROJECT$ :

این عملگر مجموعه ای عمودی از یک رابطه استخراج می کند و هیچ گونه شرطی را اعمال نمی کند و سطرهای تکراری را حذف تمامی می کند زیرا خروجی  $PROJECT$  نیز یک رابطه است از تئوری مجموعه ها پیروی می کند در نتیجه تکراری نداریم بنابراین سطرهای تکراری حذف می شوند

$PROJECT R$  [نام]

$R$

نام	فامیلی
علی	اکبری
رضا	حسینی
محسن	فاطمی
علی	قاضیان
محمد	رضایی

### عملگر پیوند ( $JOIN$ ):

این عملگر ابتدا در جدول را که دارای بند مشترکی هستند را در هم ضرب کارتیزین می کند سپس به عملگر  $JOIN$  ضرب کارتیزین را  $FILTER$  می کند.

$R_1 = JOIN R_2 OVER STD$

$R_1$

NAME	STD	AYG	STD	CODE
ALI	۱	۱۲	۱	۱۰۰
REZA	۲	۱۴	۲	۱۰۵
...	...	...	۱	۱۰۵

NAME	R1.STD	AVG	R2.STD	CODE
ALI	1	12	1	100
ALI	۱	۱۲	۲	105
ALI	۱	۱۲	۱	۱۰۵
REZA	۲	۱۴	۱	۱۰۰
REZA	۲	۱۴	۲	۱۰۵
REZA	۲	۱۴	۱	۱۰۵

**نکته:** بوسیله TIMES و select می توان join را پیاده سازی کرد.

- اگر در join عملگر مقایسه = (تساوی) باشد به آن join تساوی می گویند.
- اگر در join تساوی صفت خاصه مشترک (دو فیلد مشترک) در حاصل آن تکرار نشود به آن join طبیعی می گویند.

SELECT (R<sub>1</sub> TIMES R<sub>2</sub>) WHERE R<sub>1</sub>.STD=R<sub>2</sub>.STD GIVING

TEMP

PROJECT TEMP [ NAME,STD,CODE]

عملگر تقسیم: این عملگر رابطه R<sub>1</sub> از درجه M+N را بر رابطه R<sub>2</sub> از درجه N تقسیم می کند و خارج قسمت رابطه است از درجه M صنعت خاصه PM+I را بر رابطه R<sub>1</sub> و صفت خاصه I ام از رابطه R<sub>2</sub> (I=1,I=2,...N) باید روی یک تعریف شده باشند فرض می کنیم M صفت خاصه اول از رابطه R<sub>1</sub> صفت خاصه مرکب NoX صفت خاصه آخرش صفت مرکب Y باشد. رابطه R<sub>1</sub> را می توان مجموعه ای از جفت مقادیر (Y,X) تصور کرد و همچنین رابطه R<sub>2</sub> را می توان مجموعه ای از مقادیر (Y) تلقی کرد. در این صورت نتیجه تقسیم R<sub>1</sub> بر R<sub>2</sub> و به بیان دیگر نتیجه ارزیابی

عبارت R<sub>1</sub> DIVIDEBY R<sub>2</sub>

تهیه کننده

قطعه

محموله SP

S#	SNAME	STATUS	CITY
S1	SN1	20	C2

P#	PNAME	COLOR	CITY	WEGHT
P1	PN1	RED	C2	12

S#	P#	QTY
S1	P1	300

S2	SN2	10	C3
S3	SN3	30	C3
S4	SN4	20	C2
S5	SN5	30	C1

P2	PN2	GREEN	C3	17
P3	PN3	BLUE	C4	17
P4	PN4	RED	C2	14
P5	PN5	BLUE	C3	12
P6	PN6	RED	C2	19

S1	P2	200
S1	P3	400
S1	P4	200
S1	P6	100
S2	P1	300
S2	P2	400
S4	P5	400
S3	P3	200
S4	P2	200
S4	P4	300

۱-اسامي تهیه کنندگان قطعه P<sub>2</sub> را بدست آورید.

```
SELECT SP WHERE P#=P2 GIVING T1
      S JOIN T1 GIVING T
PROJECT T2 [SNAM]
```

۲-اسامي تهیه کنندگانی را بیابید که اقلای یک قطعه قرمز تهیه می کنند.

```
SELECT P WHERE COLOR ='RED' GIVING TEMP1
PROJECT TEMP1[P#] GIVING TEMP2
TEMP2 JOIN SP GIVING TEMP3
PROJECT TEMP3 [S#] GIVING TEMP4
TEMP4 JOIN S1 GIVING TEMP5
PROJECT TEMP5[SNAME]GIVING TEMP6
```

۳-اسامي تهیه کنندگانی را بدهید که تمام قطعات را تهیه می کنند.

```
PROJECT SP[ S#,P#] GIVING TEMP1
PROJECT P[P#]GIVING TEMP2
TEMP1 DIVIDEBY TEMP2 GIVING TEMP3
S JOIN TEMP3 GIVING TEMP4
PROJECT TEMP4[SNAME]
```