

# Fundamentals Concepts of Data Bases

## اصول طراحی پایگاه داده ها

جلسه ۳ : معماری سیستم پایگاه داده

مدرس : اسماعیل نورانی

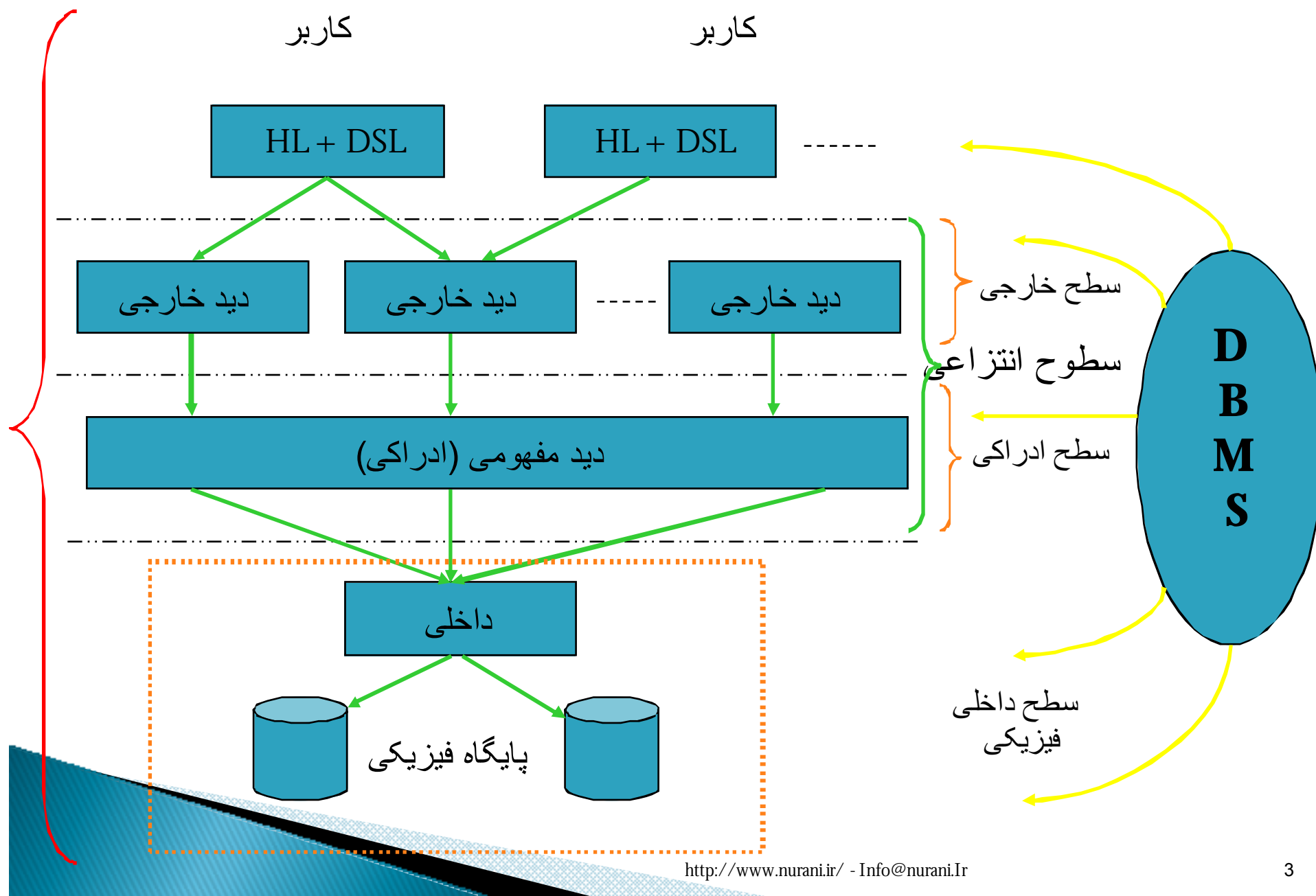
<http://www.nurani.ir/> - [Info@nurani.Ir](mailto:Info@nurani.Ir)

## معماري سيستم پايگاه داده :

در اوایل تکنولوژی پایگاه داده ها معماری استاندارد وجود نداشت. ANSI در اوایل سال 1970 معماری استاندارد ANSI را پیشنهاد کرد.

**نکته ! :** هر DBMS واقعي باید معماری ANSI را Support نماید .  
این اساسی ترین ضابطه در تشخیص نرم افزارهاست که آیا DBMS واقعي است یا شبه DBMS است یا DMS است .  
این اولین محک برای Package هاست .

# DBA



یک معماری سه سطحی یا چهار سطحی است که عناصر آن عبارتند از :

User	کاربر
Host language (HL)	زبان میزبان
Data sublanguage (DSL)	زبان فرعی داده ای
External view	دید خارجی
Conceptual view	دید اداری
Internal level	سطح داخلی
Physical Database	پایگاه فیزیکی
Mapping	نگاشت و تبدیلات بین سطوح
DBMS	نرم افزار
DBA	تیم تخصصی اداره کننده پایگاه
Abstractive	سطح انتزاعی (محیطی مستقل از Filing)

تمام سطوح تحت مدیریت DBA است و تمام سطوح تحت کنترل DBMS است .

## شرح عناصر معماری :

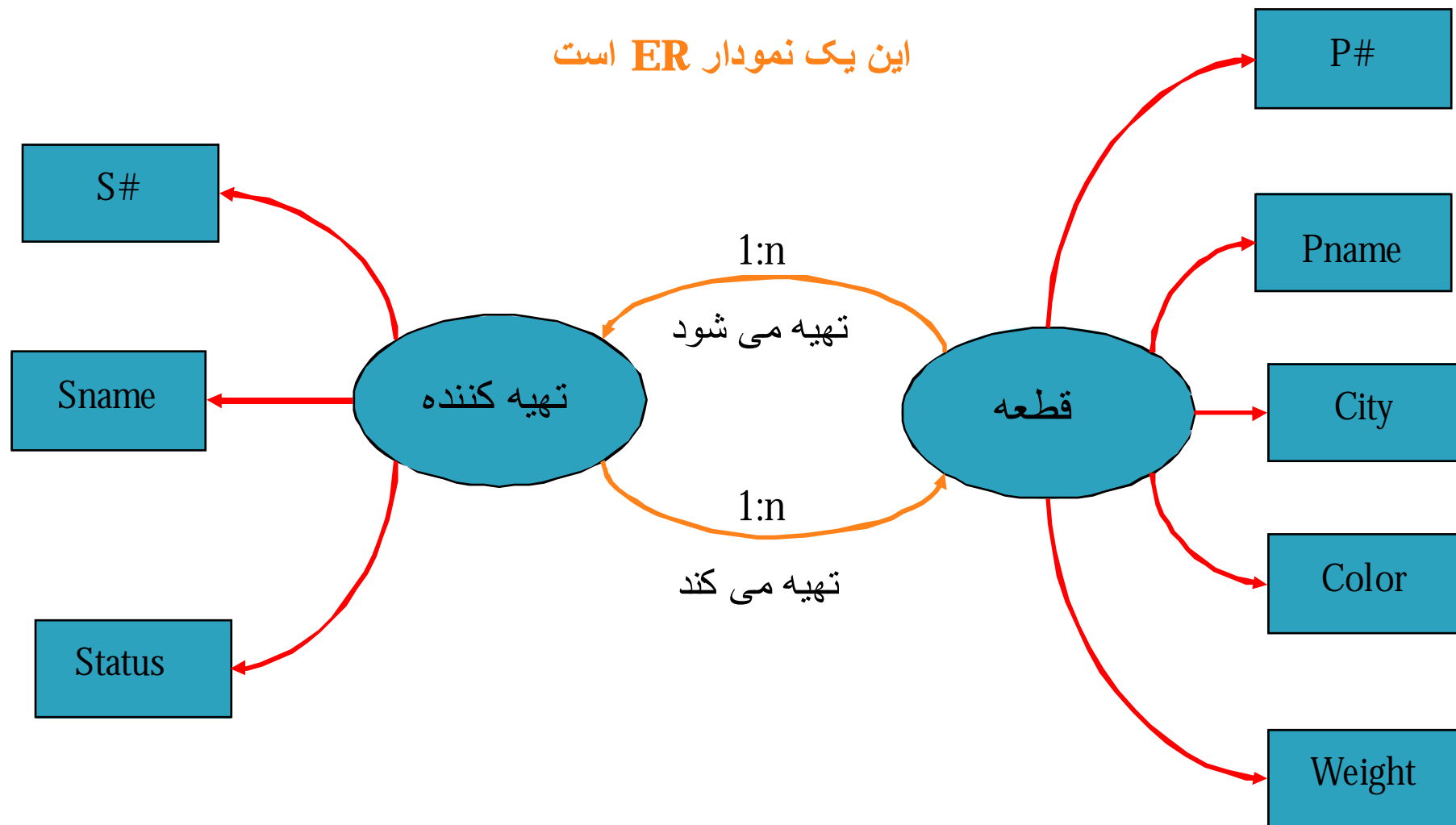
**دید ادراکی :** دید طراح است از کل داده های ذخیره شدنی در محیط عملیاتی. این دید جامع تمام نیاز های کاربران محیط است. این دید باید تحت **یک DS مشخص** طراحی شود.

این دید باید به کمک احکامی از Data sublanguage (DSL) تعریف شود.

**شمای ادراکی :** (conceptual scheme) به تعریف دید ادراکی شمای ادراکی می گویند.

شمای ادراکی نوعی برنامه است که طراح می نویسد. حاوی دستورات تعریف و کنترل داده ها که در کتابخانه مخصوصی نوشته می شود و نگهداری می شود که گاه موسوم به کاتالوگ سیستم یا Data Dictionary است.

این یک نمودار ER است



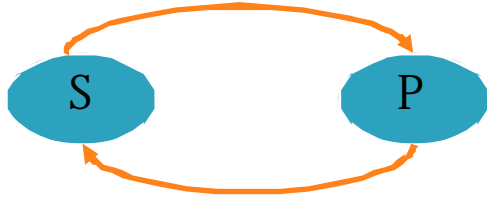
**نکته ! :** این نمودار ER باید در یک DS مشخص نگاشته Map شود هر DBMS یک DS خاص را تامین یا Support می نماید

1- رابطه ای Relational	} DS های رایج
2- سلسله مراتبی (درختی) Hierarchical	
3- شبکه ای Network	

اگر DBMS ای DS اش رابطه ای باشد به آن DBMS رابطه ای گویند .

به این دو Prerelational گویند . یعنی نسل قدیمی تر نسبت به Relational هستند .	}	اگر DBMS ای DS اش سلسله مراتبی باشد به آن DBMS سلسله مراتبی گویند
		اگر DBMS ای DS اش شبکه ای باشد به آن DBMS شبکه ای گویند .

بعد از سیستمهای رابطه ای سیستمهای Post relational به وجود آمد



معرفی مقدماتی **DS** رابطه ای برای ادامه مثال

رابطه (relation) : یک مفهوم ریاضی است

از دید کاربر رابطه نمایشی جدولی دارد (یک جورجدول است) هر جدول دو خصوصیت مهم دارد : یکی ستون و یکی سطر

برای مدل کردن سطح ادراکی پایگاه داده توسط DS رابطه ای : برای هر نوع موجودیت یک جدول تهیه می کنیم که هر ستونش بیانگر یک Attribute (صفت خاصه) آن موجودیت است .

**S**

	S#	Sname	Status	City
یک سطر →	<i>S1</i>	<i>Sn1</i>	<i>30</i>	<i>C1</i>

**P**

P#	Pname	Color	Weight	City
<i>P1</i>	<i>Pn1</i>	<i>red</i>	<i>100</i>	<i>C2</i>

اگر این جدول را به گونه ای برای کامپیوتر تعریف کنیم ، فقط نوع موجودیت (Entity Type) را معرفی کرده ایم که هر سطر یک نمونه از موجودیت خواهد بود.

Entity Occurrence

Entity instance



برای Modeling ارتباط 1:n دو سویه بین این دو موجودیت جدولی دیگر طراحی می کنیم.

S#	P#	Qty
S1	P1	100
S1	P2	80
S1	P3	100
S2	p1	60

یک نمونه محموله - یک نمونه  
ارتباط بین موجودیتها یعنی S1 P2

حال شمای ادراکی را طراحی می کنیم  
نمونه ساده شده شمای ادراکی به زبان طبیعی (natural language)

### اسم جدول **S** است

ستونها شامل

S# کاراکتر ۶  
Sname کاراکتر ۲۰  
Status عدد صحیح  
City کاراکتر ۱۲  
شناسه هر سطر S# است (identifier)

### اسم جدول **SP** است

ستونها شامل  
S# همان S# که در S است  
P# همان P# که در P است  
Qty عدد صحیح است

### اسم جدول **P** است

ستونها شامل

P# کاراکتر ۸  
Pname کاراکتر ۱۴  
Color کاراکتر ۱۰  
Weight عددی  
City کاراکتر ۱۲

# SQL : Structured Query Language

SQL : DSL استاندارد سیستمهای رابطه ای است . (زبان تعریف و کار با داده ها)

```
CREATE TABLE S  
(S# CHAR (6) ,  
Sname CHAR (20) ,  
STATUS INTEGER ,  
CITY CHAR (12) )  
PRIMARY KEY S# ;
```

ایجاد جدول S



این شمای ادراکی است در DBMS رابطه ای به نحوی که DBMS ، SQL را Support می نماید .

```
CREATE TABLE P  
(P# CHAR (8) ,  
Pname CHAR (14) ,  
WEIGHT INTEGER ,  
COLOR CHAR (10) ,  
CITY CHAR (12) )  
PRIMARY KEY P# ;
```

ایجاد جدول P



ایجاد جدول SP



```
CREATE TABLE SP  
(S# CHAR (6) ,  
P# CHAR (8) ,  
QTY INTEGER )  
PRIMARY KEY (S# , P#)
```

کاربران هرچه ببینند از همین دید ادراکی و از همین Table ها خواهند دید زیرا طراح این سطح را به طور جامع طراحی می کند و یک سطح بین کاربران و سطوح پائین تر است .

## دید خارجی External view :

۱ - دید کاربر خاص است نسبت به داده های ذخیره شده در پایگاه (در محدوده نیازهای اطلاعاتی اش)

۲ - هر کاربر دید های خاص خود را دارد

۳ - چند کاربر می توانند در یک دید مشترک باشند . همین اشتراک در دید است که تامین کننده Data sharing (اشتراک داده ها) در محیط انتزاعی می باشد  
(لازمه وجود محیط انتزاعی یک Data Structure است .)

**محیط انتزاعی:** محیط منفک و مجزا از *filig* . دید خارجی روی دید ادراکی تعریف می شود . (یعنی دید ادراکی باید ایجاد شده و در Catalog باشد تا کاربر بتواند روی این دید ادراکی دید خارجی مخصوص خود را ایجاد کند )

چون دید ادراکی روی یک DS مشخص ایجاد می شود پس دید خارجی هم مثل دید ادراکی در کادر یک DS مشخص تعریف می شود .

۵ - به تعریف دید خارجی شمای خارجی گفته می شود External Schema می گویند . نوعی برنامه است که Compile می شود و در Catalog سیستم یا در کتابخانه دیدهای خارجی ذخیره می شود تا موقع استفاده از پایگاه به کار گرفته شود .  
دیدهای زیر می تواند در سطح خارجی روی جداول S و P و SP تعریف شده باشد .

S				P			SP		
S#	Sname	Status	City	P#	Pname	City	S#	P#	QTY
S1	Sn1	10	C2						
S2	Sn2	15	C1						
S3	Sn3	17	C2						

User 1	S#	Sname
	S1	Sn1
	S2	Sn2
	S3	Sn3

User 2	SN	STA
	S1	10
	S3	17

تیم طراح تصمیم می گیرد که User x باید این دید را داشته باشد و بجز آن را نبیند .

این دید می گوید : نام من U1S1 است روی S تعریف شده و ستونهایش S# و Sname است .  
 این دید می گوید : نام من U1S2 است ستونهایش SN و STA از تهیه کنندگان ساکن C2 .

پس Database از دید User1 تنها دو جدول u1s1 و u1s2 است . اگر user1 بخواهد کاری روی DB انجام دهد فقط می تواند روی همین دو تا جدول انجام دهد و لا غیر .

U2Px

P#	City
P 1	C2
P 2	C3
P 3	C4
P 4	C2
P 5	C3
P6	C2

U3Sx

S#	City
S1	C2
S2	C3
S3	C4
S4	C2
S5	C1

شماره تهیه کننده و قطعاتی که در یک شهر هستند .

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

تا بحال دید ها از نظر سطح خارجی بودند . حال می خواهیم دید ها را از نظر سطح که داخلی هستند را بررسی کنیم و مثال بزنیم



اینکه چند فایل بکار رفته و چه Pointer هایی استفاده شده باشد و غیره بستگی به نوع Mapping از سطح ادراکی به سطح داخلی دارد .

ترکیب یک OS با Hardware را Platform گویند .

تعریف u1s2 در شبه SQL :

```
CREATE VIEW U1S2 (SN,STA)  
ON S(S#,STATUS)
```

.  
.  
.

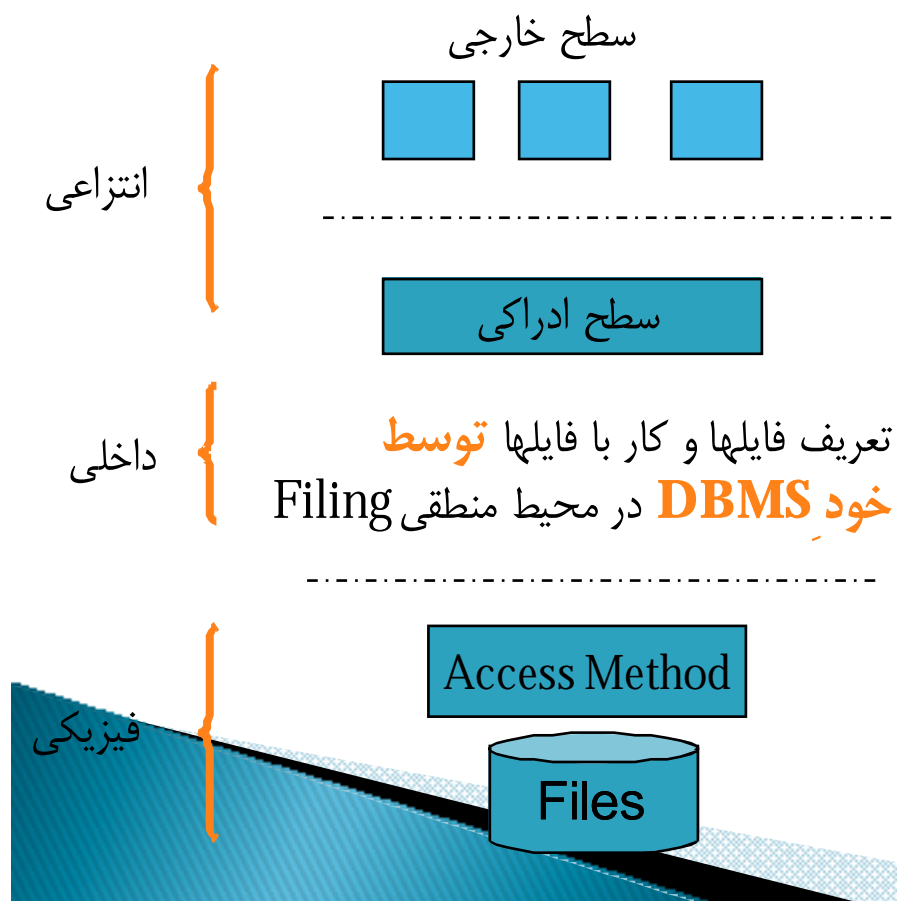
```
WHERE S.CITY = 'C2'
```

بعد از WHERE شرط یا شرایط مورد نظر را می آوریم  
وقتی سطح ادراکی شما جدولی است و مبتنی بر سطرها و ستونهایی است دیده ها نیز بخشی از جدواول  
یا بخشی از سطرها و ستونها هستند .

## سطح داخلی :

در واقع سطح file definition است درون خود سیستم توسط خود DBMS در این سطح با پایگاه برخورد منطقی دارد . اما به مسائل فیزیکی پایگاه نمی پردازد .  
(مسائل فیزیکی پایگاه بر عهده Access Method (شیوه دستیابی) است)

### در سیستم DB



### در سیستم های فایلینگ



## DBMS می داند :

- ۱- چه فایل‌هایی وجود دارد .
- ۲- استراتژی دستیابی به هر فایل چیست و ساختارش چگونه است ؟
- ۳- اندازه فایل چقدر است .
- ۴- کلید یا کلیدها را می شناسد.
- ۵- ترتیب منطقی رکوردها را می داند .
- ۶- نحوه نگاشت ساختارهای ادراکی در فایلها را . (چه فایلی مربوط به چه جدولی است ) .

### چند نکته ! :

**D-logy** : هر چند به مسائل مربوط به سطح داخلی پایگاه توسط DBMS پرداخته می شود اما DBA و تیمش اختیاراتی در طراحی این سطح دارند . مثلاً می تواند درخواست ایجاد شاخص نماید (Index) . یا اینکه انواع کلید را معرفی کند اما میزان دخالت DBA در ایجاد سطح داخلی و نهایتاً فیزیکی پایگاه در DBMS های مختلف فرق می کند . در DBMS های قدیمی تر این دخالت بیشتر است . (در Pre relational بیشتر است) اما در جدیدها کمتر زیرا اتوماتیسم DBMS ضعیف تر بوده است .

**D-logy** : معمولاً یک DBMS قوی بیش از یک ساختار File را اجازه می دهد و این که چندین Access Method وجود خواهد داشت . یک DBMS کلاسیک حداقل دو روش دستیابی را ارائه می دهد . اما DBMS هایی هم هستند که شش و یا پنج روش دستیابی را ارائه می دهند .

## مفاهیم مبنایی DM رابطه ای ( Data Model رابطه ای)

- تعریف رابطه
- خصوصیات رابطه
- مفهوم میدان (دامنه)
- کلید در DM رابطه ای
- قواعد جامعیت در DM رابطه ای
- مشخصات کلی یک سیستم رابطه ای
- عملیات در رابطه ها

## تعریف رابطه :

تعریف مقدماتی :

فرض می کنیم که :  $D_1, D_2, \dots, D_n$  ،  $n$  میدان از مقادیر باشند . رابطه مجموعه ایست که هر عنصر آن به صورت  $\langle d_{i1}, d_{i2}, \dots, d_{in} \rangle$  است

N-Tuple

(  $n$  تایی )

$$i = 1, 2, 3, \dots, m$$

به نحوی که  $d_{i1} \in D_1$

$d_{ij} \in D_j$

میدان مقادیر Status ها



مثال :  $S(S\#, sname, status, city)$



میدان مقادیر Sname ها

مجموعه مقادیر City ها

S(S#	Sname	Status	City)
S1	Sn1	10	C1
S2	sn2	20	C2
S3	sn3	15	C2
S4	sn4	14	C3

یک تاپل  
Tuple



تعریف کامل تر (از DATE) : با فرض داشتن  $n$  میدان  $D1, D2, \dots, Dn$  رابطه از دو مجموعه تشکیل شده است :

مجموعه عنوان : Heading یا سررابطه  
مجموعه بدنه : Body یا پیکر رابطه

(Attribute names)

مجموعه اسامی صفات خاصه

بنحوی که این صفات خاصه از میدانهایی که صفات خاصه روی آنها تعریف می شود ، مقدار می گیرد  
هر رابطه دارای یک نام است و با داشتن نام و مجموعه عنوان رابطه **جوهر رابطه** معلوم است.

نام رابطه       $R (A1 , A2 , \dots , An)$   
مجموعه عنوان رابطه

ذات یا جوهر رابطه که ثابت در زمان است .

منطقا باید میدانهای زیرین نیز به نحوی به سیستم معرفی شوند.

مجموعه بدنه ( Body ) :

$\{ \langle a_{i1}, a_{i2}, \dots, a_{in} \rangle \}$

$a_{i1} \in D_1$  (مقادیر صفت خاصه  $A_1$ )

$a_{ij} \in D_j$  (مقادیر صفت خاصه  $A_j$ )

به مجموعه Body :

بسط یا گستره رابطه : ( Extension رابطه ) گفته می شود  
و تعداد Tuple های رابطه را Cardinality می گویند .

کاردینالیتی رابطه : تعداد عناصر مجموعه بدنه . (تعداد سطرها)

کاردینالیتی رابطه : تعداد تاپلهای بسط رابطه .

درجه رابطه :

کاردینالیتی مجموعه عنوان ، یعنی تعداد صفات خاصه عنوان رابطه (تعداد ستونها)  
مثال : S از درجه ۴ است. SP از درجه ۳ است. P از درجه ۵ است.

اصطلاح	درجه
Unary رابطه	۱
Binary رابطه	۲
Ternary رابطه	۳
n-ary رابطه	.
	.



مفهوم جدولی	مفهوم رابطه ای
جدول	رابطه
عنوان جدول	مجموعه عنوان
سطر	تاپل
ستون	صفت خاصه
تعداد ستونها	درجه
تعداد سطرها	کاردینالیتی
مجموعه مقادیر ستون	میدان
؟(خود جدول چیزی به نام کلید ندارد)	کلید

## خصوصیات رابطه :

۱- تاپل تکراری ندارد ← چون بدنه مجموعه است و مجموعه عنصر تکراری نمی تواند داشته باشد.

۲- صفات خاصه فاقد نظم هستند چون عنوان مجموعه است نه  $n$  تایی مرتب

۳- تاپلها فاقد نظم هستند چون بدنه مجموعه است

در سیستمهای موجود هر دو نوع نظم ارائه شده است . یعنی هم می توان رابطه ای داشت منظم شده بر مبنای سطرها یا ستونها بسته به تقاضای طراح و هم برای صفات خاصه مجموعه عنوان قائل به موقعیت Position شد . می توان گفت ستون اول ، ستون دوم ، ستون سوم یعنی ترتیب قائل شویم برای صفات خاصه رابطه در نمایش جدولی .

۴- مقادیر تمام صفات خاصه اتمیک (Atomic : تجزیه نشدنی) هستند .

در تقاطع هر سطر و ستون جدول یک داده ساده داریم که از لحاظ مفهومی غیر قابل تجزیه است .

## ادامه بحث مفاهیم مبنایی مدل رابطه ای :

مقدار اتومیک یا تجزیه نشدنی : مقداری است که اگر آنرا به اجزا تجزیه کنیم اجزای بدست آمده فاقد Semantic (معنی) باشد . از این نظر مقدار اتومیک را واحد Semantic (Semantic Unit Of Data) (از دید کاربر در معنای عام : از جمله طراح ، dba و ...)

**مثال :** مقدار صفت خاصه رنگ

آبی ← غیر قابل تجزیه از لحاظ سمتیک  
سیاه ← غیر قابل تجزیه از لحاظ سمتیک  
آب + ی  
سی + آه

مقدار ساده یا اتومیک است .

این اجزای حاصله از تجزیه در مقوله شناخت مقادیر رنگ اشیا فاقد معنا هستند . در صورتیکه این تجزیه در مقوله زبانشناسی ممکن است دارای معنی باشد .

مثال : تاریخ yy mm dd : مرکب ، هم از نظر Syntax و هم از نظر Semantic . این مقدار یک مقدار اتومیک نیست زیرا تجزیه می شود به سه مقدار دارای معنا .

**نکته :** یک مقدار مرکب یا تجزیه شدنی لزوماً نباید تجزیه شود ، بستگی دارد به دید طراح . همین تاریخ را در یک دید ممکن است اتومیک ببینیم و در دید دیگر بسته به کاربرد ممکن است تجزیه شود .

**نکته :** خاصیت ۴ (اتومیک) : از نظر ریاضی یک رابطه لزومی ندارد دارای خاصیت چهارم باشد ولی این نظر آقای Codd است . دلیلش را در ادامه بحث می بینیم . (از لحاظ ریاضی لزومی ندارد میدانهایی که رابطه روی آنها تعریف می شود اتومیک باشد) .