

به نام آنکه جان را فکرت آموخت



## بخش پنجم: مفاهیم اساسی مدل داده رابطه‌ای

مرتضی امینی

نیمسال اول ۹۲-۹۳

(محتویات اسلایدها برگرفته از یادداشت‌های کلاسی استاد محمدتقی روحانی رانکوهی است.)



☐ RDM مبنای تئوریک RDB و RDBMS

☐ واضع مدل: F. Codd

☐ مباحث:

☐ رابطه (Relation)

☐ دامنه (میدان)

☐ رابطه نرمال و غیرنرمال

☐ کلید در مدل رابطه‌ای

☐ قواعد جامعیت رابطه‌ای

☐ عملیات در RDB ← جبر رابطه‌ای

☐ طراحی RDB ← حساب رابطه‌ای

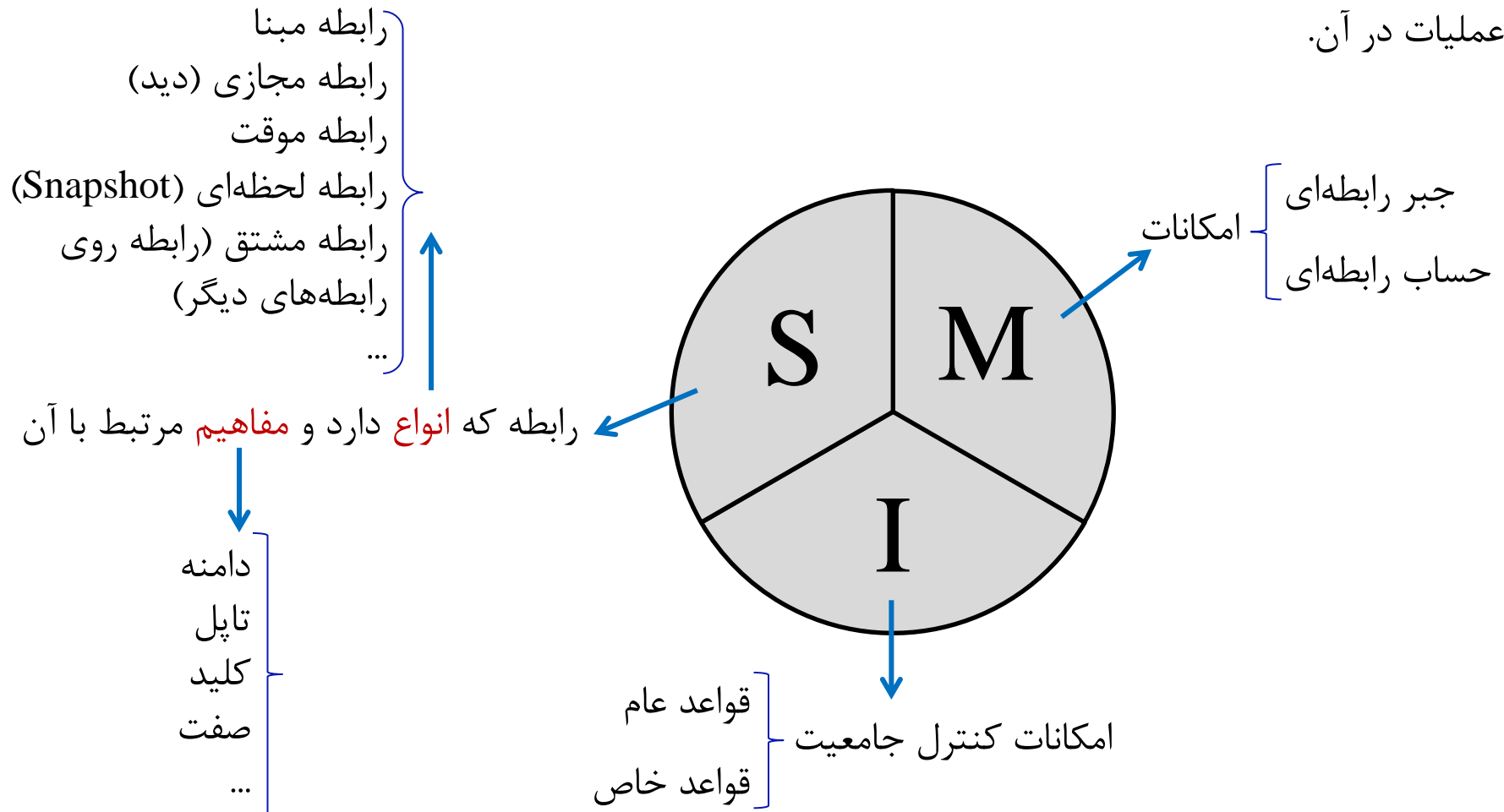
روش بالا به پایین

روش نرمال‌ترسازی (سنتز)



✓ **مدل داده** مجموعه‌ای است از امکانات برای طراحی منطقی و تعریف پایگاه داده‌ها، کنترل آن و نیز انجام

عملیات در آن.





در ریاضی: هر زیر مجموعه از ضرب کارترین چند مجموعه



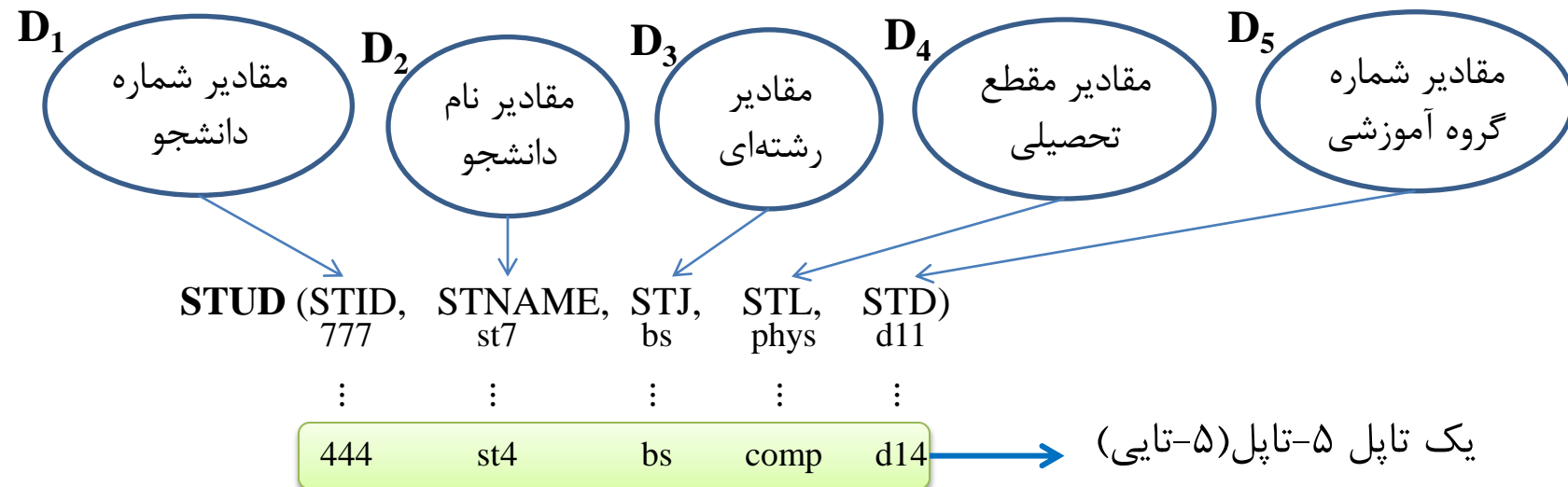
(۱) با فرض وجود  $m$  مجموعه از مقادیر موسوم به دامنه [میدان]  $D_1, \dots, D_m$ :



رابطه  $R$  با صفات  $A_1, \dots, A_n$  تعریف شده روی این  $m$  دامنه

مجموعه‌ای است از عناصر، هر یک به صورت  $\langle d_{1i}, d_{2i}, \dots, d_{ni} \rangle$  موسوم به  $n$ -تاپل (n-tuple)

به نحوی که  $d_{ni} \in D_n, \dots, d_{1i} \in D_1$





(۲) [Date] با فرض وجود  $m$  مجموعه از مقادیر موسوم به دامنه [میدان]  $D_1, \dots, D_m$  نه لزوماً متمایز،

رابطه  $R$  تعریف شده روی این  $m$  دامنه:

- دومجموعه
- عنوان [سرآیند] (Heading): مجموعه‌ای است نامدار از اسامی صفات یعنی  $\{A_1, \dots, A_n\}$  که با  $R(A_1, \dots, A_n)$  نمایش داده می‌شود.
  - بدنه [پیکر] (Body): مجموعه‌ای است از تاپل‌ها [همان مجموعه در تعریف اول].

رابطه دانشجو



STUD (STID, STNAME, STJ, STL, STD)

اصطلاح	n
رابطه یگانی	۱
رابطه دوگانی	۲
رابطه n گانی	n

□ **درجه رابطه:** کاردینالیتی عنوان یا تعداد صفات رابطه



مجموعه عنوان را با  $H_R$  یا  $R(H)$  نیز نمایش می‌دهیم. به  $R(H)$  Intension (ذات، جوهر یا چکیده) رابطه هم گفته می‌شود. ☐

$R(H)$  ثابت در زمان است. یعنی اگر مجموعه صفات را عوض کنیم، از نظر ریاضی یک رابطه دیگر است. ☐

همین  $R(H)$  برای تعریف رابطه در سیستم کافی است. ☐



### CREATE RELATEION STUD

(STID, STNAME, STJ, STL, STD)

هر رابطه یک معنا دارد، بیانگر واقعیتی از یک محیط مشخص. به عنوان مثال وقتی می‌گوییم رابطه STUD را داریم، معنایش این است که در خردجهان واقع، نوع موجودیتی با نام STUD و با صفات STID و STNAME و ... و STD وجود دارد. ☐

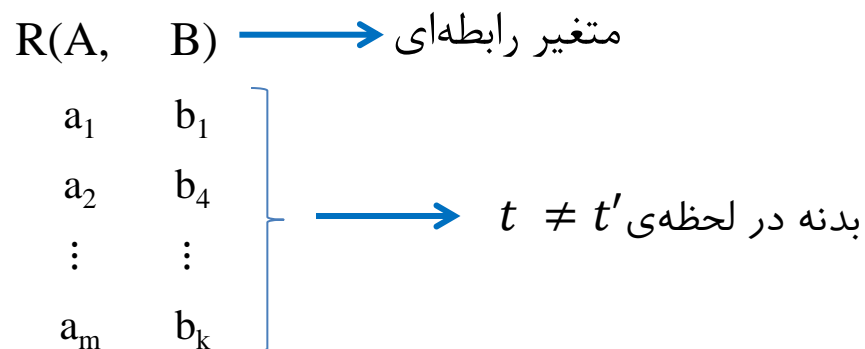
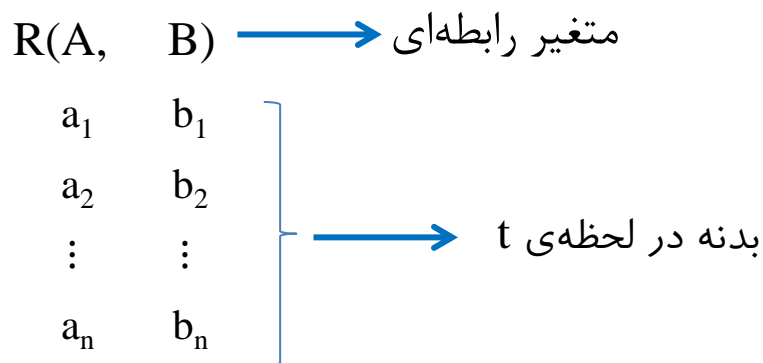


❑ **کاردینالیتی رابطه:** همان کاردینالیتی بدنه؛ تعداد تاپل‌ها (بزرگتر مساوی صفر؛ صفر در بدو تعریف)

❑ بدنه رابطه، متغیر در زمان است.

❑ به یک مقدار بدنه در یک لحظه مشخص instance گویند.

❑ به بدنه رابطه Extension (بسط یا گسترده) یا حالت رابطه گویند.





تاریخ (۳) [از نظر تئوری زبان‌های برنامه‌سازی] تشکیل شده است از یک متغیر رابطه‌ای و در هر لحظه از یک مقدار رابطه‌ای.

□  $R(H)$ : متغیر رابطه‌ای، متغیری از جنس رابطه  $[RELVAR]$  Relation Variable

□ بدنه  $(r)$ : مقدار رابطه‌ای  $[RELVAL]$  Relation Value

$R(A, B) \longrightarrow$  متغیر رابطه‌ای

$a_1$	$b_1$
$a_2$	$b_2$
$\vdots$	$\vdots$
$a_n$	$b_n$

$\longrightarrow$  یک مقدار رابطه‌ای  
(در لحظه بعد ممکن است  
مقدارش فرق کند.)





# مدل رابطه‌ای و مدل جدولی

۹

بخش پنجم: مفاهیم اساسی مدل داده رابطه‌ای

□ تناظر بین مفاهیم رابطه‌ای و اصطلاحات جدولی

اصطلاح	مفهوم رابطه‌ای
جدول (صرفاً امکانی است برای نمایش مفهوم رابطه‌ای و تفاوت‌های متعددی با رابطه دارد.)	رابطه
سطر	تاپل
ستون	صفت
مقادیر مجاز ستون	دامنه
تعداد ستون‌ها	درجه
تعداد سطرها	کاردینالیتی
؟ (به معنایی که در مدل رابطه‌ای داریم، در بحث‌های جدولی مطرح نیست.)	کلید ←



## ویژگی‌های رابطه: □

۱- صفات در عنوان رابطه نظم (مکانی) ندارند. [عنوان رابطه مجموعه است]  $R(A, B) = R(B, A)$

در حالی که در جدول، ستون‌ها می‌توانند نظم مکانی داشته باشند.

در مدل رابطه‌ای، تنها راه ارجاع به صفت رابطه، نام صفت است.

۲- تاپل‌ها [در بدنه] نظم ندارند (مرتب نیستند) [چون بدنه مجموعه است].

۳- رابطه، تاپل تکراری ندارد [چون بدنه مجموعه است].

۴- تمام صفات رابطه، تک مقداری هستند [ارجوع شود به مفهوم رابطه نرمال] (این ویژگی دلیل تکنیکی ← دارد و از ذات رابطه نتیجه نمی‌شود). یعنی در هر تاپل دقیقاً یک مقدار برای هر صفت وجود دارد.

□ **تاکید:** در RM هیچ یک از مفاهیم فایلینگ مطرح نیستند (مثل نظم، فیلد، رکورد، اشاره‌گر، آدرس که در سطح طراحی و فایلینگ فیزیکی مطرح است).



## تفاوت‌های مفهوم رابطه و اصطلاح جدول

۳ ویژگی اول رابطه، ۳ تفاوت

۴- در رابطه  $n \geq 0$  (درجه)، یعنی از نظر تئوری رابطه می‌تواند از درجه صفر باشد.

۵- رابطه می‌تواند بیش از دو بُعد داشته باشد (مثلاً Data Cube).

۶- نمایش دقیق عنوان رابطه به صورت زیر است حال آنکه عنوان جدول چنین نیست.

عنوان رابطه مجموعه‌ای است از دوتایی‌ها منظم دامنه، صفت  $R(H): \{ \langle D_1: A_1 \rangle, \langle D_2: A_2 \rangle, \dots \}$

۷- نمایش دقیق تاپل رابطه به صورت زیر است حال آنکه سطر در جدول چنین نیست.

تاپل مجموعه‌ای است از سه‌تایی‌های منظم دامنه، صفت، مقدار  $TUPLE: \{ \langle D_1: A_1: V_1 \rangle, \langle D_2: A_2: V_2 \rangle, \dots \}$

۸- رابطه نمی‌تواند هیچ مقدار داشته باشد، ولی جدول می‌تواند.



## مفهوم دامنه (میدان) □

□ مجموعه‌ای است نامدار از مقادیر هم نوع، که حداقل یک صفت از رابطه، از آن **معنا**، **نوع** و **مقدار** می‌گیرد.

□ **STLEVEL** : (معنا : مقطع تحصیلی دانشجو - نوع : char(6) - مقادیری : { 'bs', 'ms', 'dr' } )



□ معادل است با مفهوم Data Type در تئوری انواع.

□ دامنه‌هایی که یک رابطه روی آن‌ها تعریف می‌شود، لزوماً متمایز نیستند.

مفروض  $R(H)$

لزوماً چنین نیست که  $(D_i \neq D_j) \Rightarrow A_i \neq A_j$  if  $A_i \in H, A_j \in H$

□ رابطه‌ی نمایشگر ارتباط پیشنهادی درس :  $COPRECO (COID, PRECOID)$



■ هر دو صفت از دامنه‌ی «شماره‌ی درس» هستند.



تمرین: مثالی از یک رابطه ۵-تایی که

دوصفت آن از یک دامنه باشد.

سه صفت آن از یک دامنه باشد.

اگر  $n$  درجه رابطه و  $m$  تعداد دامنه‌ها باشد، داریم:  $m \leq n$ .

برای تعریف یک رابطه در سیستم رابطه‌ای، از لحاظ تئوریک، ابتدا باید دامنه‌هایش را تعریف کرد.



مثالی از شمای پایگاه رابطه‌ای  
(در مدل تئوریک)

<b>CREATE DOMAIN SN</b>	<b>CHAR(8) DEFAULT '00000000'</b>
<b>CREATE DOMAIN SNAME</b>	<b>CHAR(20) DEFAULT 'noname'</b>
<b>CREATE DOMAIN SJ</b>	<b>CHAR(4) DEFAULT '?...?'</b>
<b>CREATE DOMAIN SL</b>	<b>CHAR(3) DEFAULT '?...?'</b>
<b>CREATE DOMAIN SD</b>	<b>CHAR(4) DEFAULT '?...?'</b>
<b>CREATE DOMAIN CN</b>	<b>CHAR(6) DEFAULT '?...?'</b>
<b>CREATE DOMAIN GR</b>	<b>DEC(2, 2) DEFAULT '?...?'</b>

...

```
CREATE RELATEION STUD
    (STID DOMAIN SN,
     STNAME DOMAIN SNAME,
     STJ DOMAIN SJ,
     STL DOMAIN STL,
     STD DOMAIN SD)
```

```
CREATE RELATION COUR ....
```



**CREATE RELATEION SCR**

(STID **DOMAIN** SN,  
COID **DOMAIN** CN,  
TR **DOMAIN** TERM ,  
YRYR **DOMAIN** YEAR,  
GRADE **DOMAIN** GRD)

...

مثالی از شمای پایگاه رابطه‌ای (ادامه)

(در مدل تئوریک)





دستورات زیر در SQL مطالعه شود.



CREATE DOMAIN ☐

ALTER DOMAIN ☐

DROP DOMAIN ☐

مزایای مفهوم دامنه از دیدگاه مهندسی نرم‌افزار بررسی شود.



چرا در مدل رابطه‌ای توصیه می‌شود میدان‌ها ساده باشند؟



تمرین: دامنه‌ی مرکب آدرس را تعریف کنید.





□ رابطه نرمال (بهنجار - عادی Flat Relation):

رابطه‌ای که تمام صفات آن تک‌مقداری (حداکثر دارای یک مقدار در هر تاپل) باشند.



□ رابطه غیر نرمال (Nested Relation):

رابطه‌ای که حداقل یک صفت آن چندمقداری باشد.



□ **توجه:** تعریف زیر درست نیست:

□ رابطه‌ای نرمال است که مقادیر تمام صفات آن اتمیک (تجزیه نشدنی) باشند.

□ تذکر: ساده یا مرکب بودن صفت نقشی در نرمال بودن و نبودن آن ندارد.



صفت چندمقداری ساده

NNCOPRECO ( COID , PRECOID )

COID	PRECOID
c01	$\begin{Bmatrix} c11 \\ c17 \\ c08 \end{Bmatrix}$
c02	$\begin{Bmatrix} c03 \\ c09 \end{Bmatrix}$
c03	c10

یک تاپل

COPRECO ( COID , PRECOID )

COID	PRECOID
c01	c11
c01	c17
c01	c08
c02	c03
c02	c09
c03	c10

یک تاپل

تبدیل به  
رابطه نرمال



صفت چندمقداری مرکب  
PTQY

NNSP ( S# , **P#, QTY** )

SP ( S# , P# , QTY )

S#	PQTY
s1	$\left\{ \begin{array}{ll} p1 & 100 \\ p2 & 90 \\ p3 & 50 \end{array} \right\}$
s2	$\left\{ \begin{array}{ll} p1 & 60 \\ p2 & 90 \end{array} \right\}$
s3	p1 150

یک تاپل

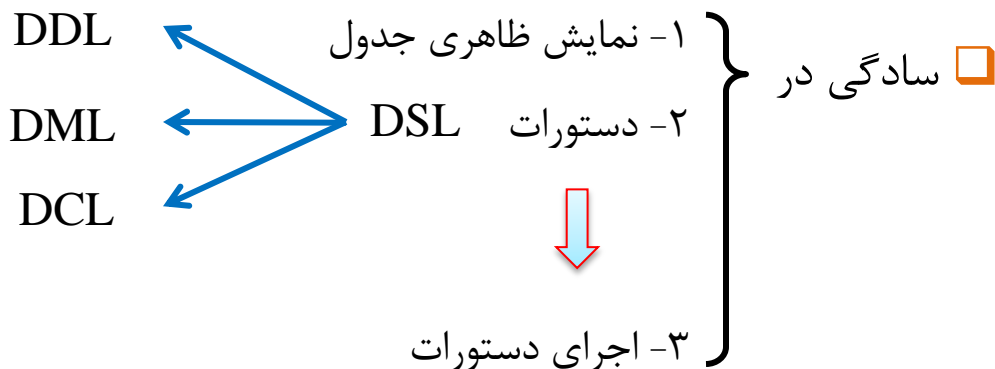
تبدیل به رابطه  
نرمال

S#	P#	QTY
s1	p1	100
s1	p2	90
s1	p3	50
s2	p1	60
s2	p2	90
s3	p1	150

یک تاپل



□ دلیل نرمال بودن رابطه در RM:



مثال □ برای درک موارد ۲ و ۳

$I_1: \langle s4, p4, 40 \rangle$  : در هر دو رابطه NNSP و SP منجر می شود به درج «تاپل

در رابطه» با همان دستور ساده «درج کن تاپل را».

درج کن □

$I_2: \langle s2, p3, 30 \rangle$  : با همان دستور ساده درج می شود در SP اما نه در NNSP.



$I_2$  : INSERT INTO SP

TUPLE (S4 , P4 , 40);

$I_1$  : INSERT INTO NNSP

TUPLE (S4 , P4 , 40);

ادامه مثال ☐

$I_3$  : INSERT INTO SP

TUPLE (S2 , P3 , 30);



امکان پذیر

$I_4$  : INSERT INTO NNSP

TUPLE (S2 , P3 , 30);



امکان ناپذیر

☐ دلیل: تاپلی با کلید S2 وجود دارد.

برای درج  $I_2$  در NNSP منطقاً چه باید کرد؟



✓ در رابطه غیر نرمال دستورات ساده‌ی تاپلی کار نمی‌کنند.



# مزایا و معایب رابطه نرمال و غیر نرمال

۲۲

بخش پنجم: مفاهیم اساسی مدل داده رابطه‌ای

نوع رابطه	مزایا	معایب
نرمال	<p>سادگی (۱- ... ۲- ... ۳-...)</p> <p>تقارن صفات</p> <p>(نقش تمام صفات در عبارت WHERE وقتی که شرط جستجو را با theta می‌دهیم، یکسان است، زیرا همه تک‌مقداری‌اند.</p> <p>SELECT....</p> <p>FROM ....</p> <p>WHERE A&lt;(=)(&gt;) 'Single Value'</p> <p>چنین تقارنی در رابطه غیرنرمال وجود ندارد.)</p>	<p>طولانی شدن کلید</p> <p>افزونگی (ادراکی یا منطقی)</p> <p>(این نوع افزونگی که در مرحله طراحی پیدا شده ممکن است منجر به افزونگی فیزیکی بشود یا نشود؛ بستگی دارد به نحوه پیاده‌سازی رابطه در سطح فایلینگ. اگر تناظر یک به یک باشد، که هر تاپل هم با یک رکورد پیاده‌سازی شود، افزونگی فیزیکی نیز پیش می‌آید.)</p> <p>دشواری در نمایش طبیعی ارتباط سلسله مراتبی بین اشیاء</p> <p>دشواری در نمایش مفهوم وراثت ←</p>
غیرنرمال	<p>[عکس معایب رابطه نرمال]</p>	<p>پیچیدگی (۱- ... ۲- ... ۳-...)</p> <p>عدم تقارن صفات</p>



## مزایا و معایب رابطه نرمال و غیر نرمال (ادامه)

بخش پنجم: مفاهیم اساسی مدل داده رابطه‌ای

۲۳

□ در عمل با کلید طولانی چه باید کرد؟

□ از یک کلید ساختگی استفاده می‌کنیم؛ یعنی یا خودمان به صورت دستی و یا خود سیستم به صورت

خودکار به هر سطر یک شماره می‌دهد.

این تکنیک چه مزایا و چه معایبی دارد؟



تأثیر تقارن صفات در سطح فایلینگ چیست؟





❑ اصطلاح **کلید**، یک اصطلاح عام است و گونه‌هایی دارد:

۱- سوپرکلید (اَبَر کلید): Super Key (SK)

۲- کلید کاندید (کلید نامزد): Candidate Key (CK)

۳- کلید اصلی: Primary Key (PK)

۴- کلید بدیل: Alternate Key (AK)

۵- کلید خارجی: Foreign Key (FK)





# کلید در مدل رابطه‌ای – سوپر کلید

۲۵

بخش پنجم: مفاهیم اساسی مدل داده رابطه‌ای

رابطه  $R(A_1, A_2, \dots, n)$  را در نظر می‌گیریم. ☐

$H_R$   
سوپر کلید (Super Key) ☐

هر زیر مجموعه  $S \subseteq H_R$  که یکتایی مقدار داشته باشد.



اگر  $t_i$  و  $t_j$  دو تاپل دلخواه و متمایز از  $R$  باشند و  $t_i(S) \neq t_j(S)$ ، آنگاه  $S$  یک سوپر کلید است. ☐

اگر  $M$  تعداد SKهای رابطه  $R$  باشد،  $M \geq 1$  است، زیرا در بدترین حالت خود  $H$  سوپر کلید می‌شود. چون ☐

بدنه، مجموعه است و تاپل تکراری نداریم.

اگر رابطه  $R$  درجه  $n$  و  $M$  تعداد SKهای رابطه  $R$  باشد: ☐

$$1 \leq M \leq 2^n - 1$$

کاربرد سوپر کلید: ☐

در عمل، فاقد کاربرد مستقیم، در تئوری در بحث طراحی. ☐

در SQL: با UNIQUE محدودیت یکتایی مقدار را اعمال می‌کنیم. ☐



# کلید در مدل رابطه‌ای – کلید کاندید

۲۶

بخش پنجم: مفاهیم اساسی مدل داده رابطه‌ای

کلید کاندید (Candidate Key) ☐

هر زیرمجموعه  $K \subseteq H_R$  که دو ویژگی داشته باشد:



۱- یکتایی مقدار

۲- کاهش‌ناپذیری (Irreducibility) یا کمینگی (Minimality)

▪  $K \subseteq H_R$  کاهش‌ناپذیر است هرگاه هر زیرمجموعه محض از  $K$ ، خود، یکتایی مقدار نداشته باشد.

▪ هر زیرمجموعه از  $H_R$  به نحوی که اگر یک صفت را از آن حذف کنیم دیگر یکتایی مقدار نداشته باشد.

رابطه	کلید کاندید
STUD	STID
COUR	COID
SCR	(STID, COID)
S	S#
P	P#
SP	(S#, P#)





## کلید در مدل رابطه‌ای – کلید کاندید (ادامه)

بخش پنجم: مفاهیم اساسی مدل داده رابطه‌ای

۲۷

□ CKها بر اساس قاعده (قواعد) معنایی محیط بدست می‌آیند.

دو حالت مختلف:



شماره ملی    شماره پروژه    شماره کارمند

EMP PROJ (E#,      J#,      ENC, ... )  
                    CK                      CK

□ هر کارمند در بیش از یک پروژه می‌تواند شرکت داشته باشد.

EMP PROJ (E#,      J#,      ENC, ... )  
                    CK                      CK

□ هر کارمند در حداکثر یک پروژه می‌تواند شرکت داشته باشد.



## کلید در مدل رابطه‌ای – کلید کاندید (ادامه)

۲۸

بخش پنجم: مفاهیم اساسی مدل داده رابطه‌ای

□ خصوصیات کلید کاندید:

□ هر SK, CK هم هست ولی عکس این مطلب صادق نیست.

□ هر رابطه حداقل یک CK دارد، زیرا در بدترین حالت، خود  $H_R$  می‌شود CK. (رابطه‌ی تمام کلید)



□ رابطه می‌تواند بیش از یک CK داشته باشد.

چه ویژگی‌هایی دارد؟ (مثال بزنید)

□ رابطه R حداکثر چند CK دارد؟

□ بیشترین تعداد CK زمانی است که به اندازه نصف تعداد صفات رابطه در CK شرکت کنند.

□ CKهای رابطه می‌توانند همپوشا باشند، یعنی حداقل در یک صفت مشترک باشند.

□ بنابراین اگر رابطه از درجه  $n$  باشد، بیشترین تعداد CK:  $C_n^m = \frac{n!}{m!(n-m)!}$  به نحوی که  $m = \left\lfloor \frac{n}{2} \right\rfloor$ .



## کلید در مدل رابطه‌ای – کلید کاندید (ادامه)

۲۹

بخش پنجم: مفاهیم اساسی مدل داده رابطه‌ای

☐ **نقش کلید کاندید:** تضمین کننده عملیات تاپلی (و نه مجموعه‌ای) یا امکان ارجاع به تک تاپل در رابطه را فراهم می‌کند. (با کم‌ترین صفات ممکن)

☐ هر زیرمجموعه از CK، یک SK است (تفاوتشان در این است که CK با کمترین تعداد صفات یکتایی مقدار را می‌دهد).

☐ CK(های) رابطه باید به سیستم معرفی شوند.



**CREATE RELATEION EMPROJ**

(E# ... NOT NULL,  
J# ... NOT NULL,  
ENC ... NOT NULL)

**CANDIDATE KEY (E#, J#)**

**CANDIDATE KEY (J#, ENC)**

☐ تئوری این را می‌گوید ولی در عمل، پکیج‌ها نمی‌پذیرند.



# کلید در مدل رابطه‌ای – کلید اصلی

۳۰

بخش پنجم: مفاهیم اساسی مدل داده رابطه‌ای

## کلید اصلی (Primary Key) ☐

کلید اصلی (PK) یکی از CKها است به انتخاب طراح.



☐ در عمل با عبارت PRIMARY KEY تعریف می‌شود.

## ضوابط انتخاب کلید اصلی: ☐

۱- شناسه رایج در محیط باشد.

۲- مقادیرش همیشه معلوم باشد

۳- کوتاه‌تر بودن طول

۴- حتی‌الامکان مقادیرش تغییر نکند.



## کلید در مدل رابطه‌ای – کلید اصلی (ادامه)

بخش پنجم: مفاهیم اساسی مدل داده رابطه‌ای

۳۱

□ دلایل لزوم انتخاب کلید اصلی:

۱- دلیل تاریخی: PK مفهوم آشنا تر برای طراحان است.

۲- ایجاد نمایه خودکار (Automatic Index) روی PK.

۳- در بحث جامعیت DB: چون محدودیت هیچ مقدارناپذیری را اگر به همه CKها بدهیم خیلی

محدودکننده است. یکی که این محدودیت را روی آن اعمال می‌کنند می‌شود PK.

چه دلایل دیگری متصور است؟



□ اصالت مفهومی در مدل رابطه‌ای با کلید کاندید (CK) است.



# کلید در مدل رابطه‌ای – کلید بدیل

۳۲

بخش پنجم: مفاهیم اساسی مدل داده رابطه‌ای

## کلید بدیل (Alternate Key) ☐

به هر کلید کاندید (CK) غیر از کلید اصلی (PK)، کلید بدیل (AK) گویند.



☐ در عمل متناظر ندارد.

☐ اگر  $N \geq 0$  تعداد AK های رابطه R باشد، داریم.



ممکن است فقط یک CK داشته باشیم که آن هم می‌شود PK و دیگر AK نداریم.

این مفهوم در عمل به چه کار می‌آید؟







# کلید در مدل رابطه‌ای – کلید خارجی

۳۳

بخش پنجم: مفاهیم اساسی مدل داده رابطه‌ای

## کلید خارجی (Foreign Key)

□ در بیان عملی:  $T_2.C$  در جدول  $T_2$ ، کلید خارجی است هرگاه در جدول  $T_1$ ، کلید اصلی باشد.

□ در بیان تئوری: صفت (ساده یا مرکب)  $R_2.A_i$  در  $R_2$  کلید خارجی است، هرگاه در  $R_1$ ، نه لزوماً متمایز از  $R_2$ ، کلید کاندید (CK) باشد.

□ تاکید: صفت (صفات) کلید خارجی باید هم‌میدان با صفت (صفات) کلید کاندید باشد و معمولاً هم‌نام با کلید کاندید است، ولی گاه لازم می‌شود که نام دیگری داشته باشد.



رابطه	کلید خارجی	دلیل: CK در
SCR	STID	STUD
SCR	COID	COUR
SPJ	S#	S
SPJ	P#	P
SPJ	J#	J



## کلید در مدل رابطه‌ای – کلید خارجی (ادامه)

۳۴

بخش پنجم: مفاهیم اساسی مدل داده رابطه‌ای

□ اگر  $N$  تعداد FKهای رابطه  $R$  باشد، داریم  $N \geq 0$ .

□ معرفی کلید خارجی با عبارت FOREIGN KEY انجام می‌شود.

□ نقش **کلید خارجی**: برای نمایش ارتباط(ها)ی صریح بین نوع موجودیت(ها) (و در نتیجه بین نمونه‌های آنها)

به کار می‌رود. منظور از «ارتباط صریح»، ارتباطی است که در مدل ER با لوزی مشخص شده است.



$S(\underline{S\#}, \dots)$                        $P(\underline{P\#}, \dots)$   
CK                                      CK

$SP(\overset{FK}{\underline{S\#}}, \overset{FK}{\underline{P\#}}, \dots)$   
CK

$SCR(\overset{FK}{\underline{STID}}, \overset{FK}{\underline{COID}}, \dots)$   
CK



# کلید در مدل رابطه‌ای – کلید خارجی (ادامه)

بخش پنجم: مفاهیم اساسی مدل داده رابطه‌ای

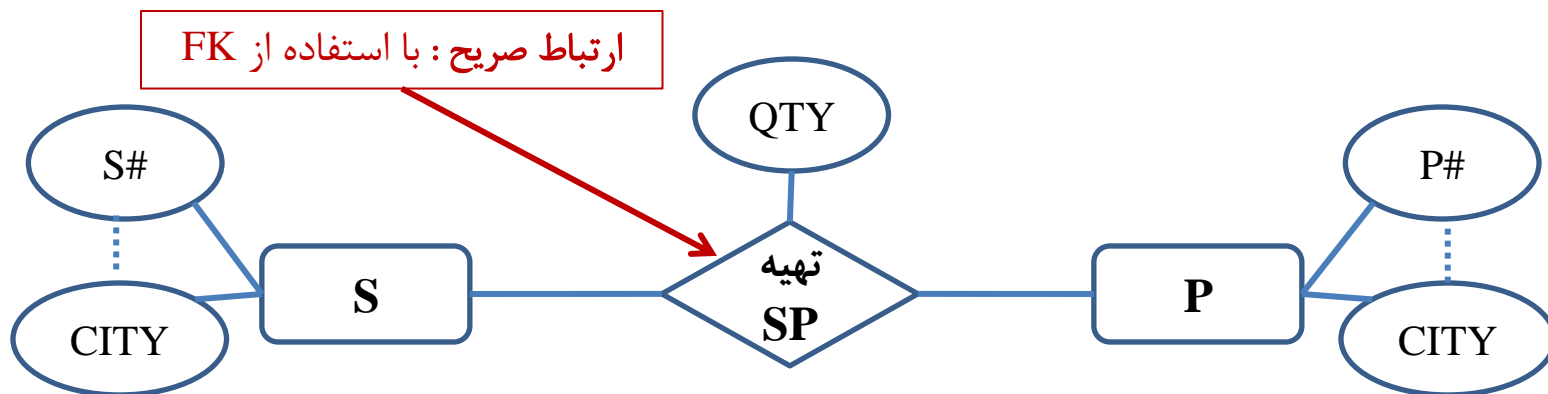
۳۵

□ آیا FK تنها امکان نمایش ارتباط است یا امکان دیگری هم وجود دارد؟

□ FK تنها امکان نیست.

□ وجود هر صفت مشترک [هم دامنه و در عمل، هم نام (نه لزوماً)]، در عنوان مثلاً دو رابطه، نمایشگر

نوعی ارتباط است بین دو نوع موجودیت که با آن دو رابطه نمایش داده‌ایم.



→ S (S#, ..., CITY )  
→ P (P#, ..., CITY )  
→ SP (S#, P#, ... )  
ارجاع

ارتباط ضمنی: از طریق هر صفت مشترک؛  
صفت هم‌معنا (از یک میدان) و نه لزوماً هم‌نام



## بحث تکمیلی: کلید خارجی – گراف ارجاع

۳۶

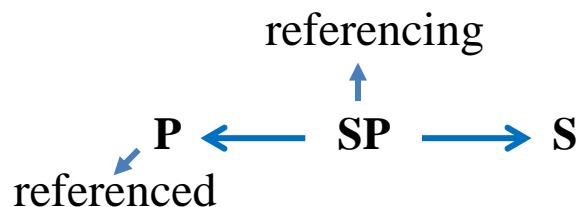
بخش پنجم: مفاهیم اساسی مدل داده رابطه‌ای

مفهوم گراف ارجاع

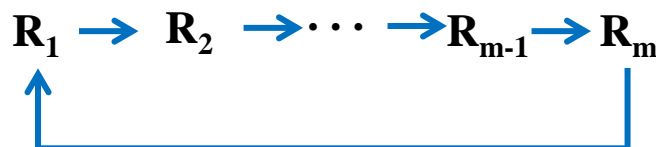
FK امکانی است برای ارجاع از یک رابطه به رابطه‌ای دیگر

هر مقدار معلوم FK، امکانی است برای ارجاع مقداری از تاپل (هایی) از رابطه (هایی) به تاپلی از رابطه (هایی).

گراف ارجاع امکانی است برای نمایش ارجاعات بین رابطه‌ها.



شکل کلی مسیر ارجاع:



با این ارجاع می شود چرخه ارجاع

مسیر ارجاع می تواند چرخه‌ای باشد.

□ چرخه ارجاع می‌تواند تک‌رابطه‌ای باشد و این در صورتی است که یک رابطه خود ارجاع (Self-Referencing) داشته باشیم.

□ هنگامی که FK تعریف می‌کنیم باید معنایش را نیز بگوییم.

مثال □ چرخه ارجاع بین دو رابطه کارمند و اداره.

شماره کارمند مدیر اداره

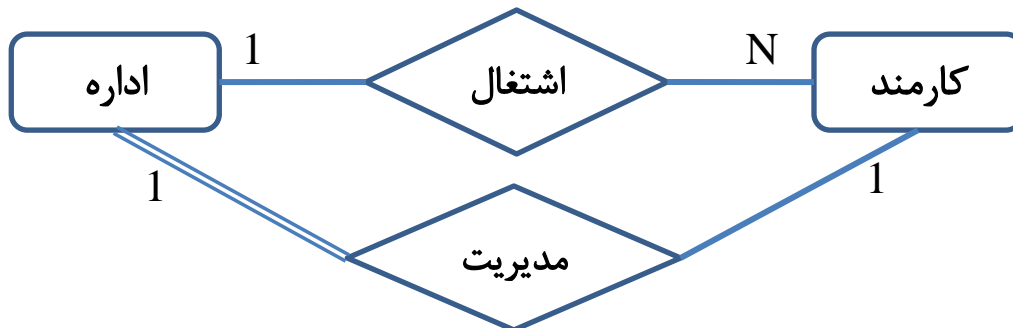
**DEPT (D#, DTITLE, ..., E#)**

شماره اداره محل کار

**EMPL (E#, ENAME, ..., D#)**

**DEPT**  $\longleftrightarrow$  **EMPL**

□ بر اساس کدام مدل‌سازی این طراحی انجام شده است؟





چرخه ارجاع تک‌رابطه‌ای کارمند با خودش.



شماره مدیر

EMPL (E#, ENAME, ENC, ..., EPHONE, EMANAGER#)



EMPL

نکته‌های مثال اخیر: □

□ مثالی است از حالتی که در آن R1 و R2 در تعریف FK، لزوماً متمایز نیستند.

□ رابطه EMPL به خود رجوع کننده (خود ارجاع) است.

□ اگر n درجه EMPL باشد و m تعداد دامنه‌هایش باشد، داریم:  $m \leq n - 1$

□ لزوم دگر نامی شماره کارمندی مدیر، چون عنوان رابطه (Heading)، مجموعه‌ای از نام صفات است.

□ **تمرین:** این طراحی بر اساس کدام مدل‌سازی انجام شده است؟



بخش پنجم: مفاهیم اساسی مدل داده رابطه‌ای

چرخه ارجاع سه رابطه‌ای



PROF (PRID, PRNAME, ..., DEID)

دانشکده استاد

DEPT(DEID, DTITLE, ....., UNID)

UNIV(UNID, UNAME, ..., UNPRESNUM)

شماره استادی رئیس دانشگاه



**تمرین:** این طراحی بر اساس کدام مدل‌سازی انجام شده است؟

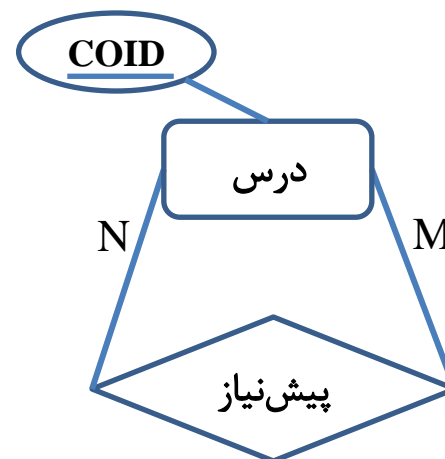


به صرف وجود دور در ERD چرخه ارجاع ایجاد نمی‌شود.



COUR (COID, ...)

COPRECO(COID, PRECOID)



□ در چه وضعی چرخه ارجاع پدید می‌آید؟

□ باید به چندی ارتباط‌ها توجه شود.





## جامعیت پایگاه داده‌ها (DB Integrity) ☐

صحت، سازگاری [، دقت و اعتبار] داده‌های ذخیره شده در پایگاه داده‌ها



جنبه‌های کیفی داده (Data Quality Features)

☐ مسئولیت کنترل جامعیت DB با RDBMS است.

☐ بر اساس اطلاعاتی که کاربر [تیم طراح - پیاده‌ساز] به سیستم می‌دهد.

قواعد یا محدودیت‌های جامعیتی (Integrity Rules/Constraints)

☐ IRها [ICها] با استفاده از دستورات زبان پایگاهی به سیستم داده می‌شوند.

← اعلانی: قواعد به نحوی اعلان می‌شوند.

← اجرایی: قواعد در یک رویه به سیستم داده می‌شوند.



□ هر DBMS ای باید بتواند جامعیت پایگاه داده‌ها را کنترل و تضمین کند.

□ **دلیل:** زیرا همیشه ممکن است عواملی به طور مستقیم یا غیر مستقیم سبب نقض جامعیت شوند. از جمله:

□ ناقص بودن قواعد و محدودیت‌های جامعیتی

□ اشتباه در برنامه‌های کاربردی (به ویژه اشتباهات معنایی)

□ اشتباه در وارد کردن داده‌ها

□ وجود افزونگی کنترل نشده

□ اجرای همروند تراکنش‌ها به گونه‌ای که داده نامعتبر ایجاد شود.

□ خرابی‌های سخت‌افزاری و نرم‌افزاری



❑ اعمال IRها برای سیستم فزونکاری دارد.

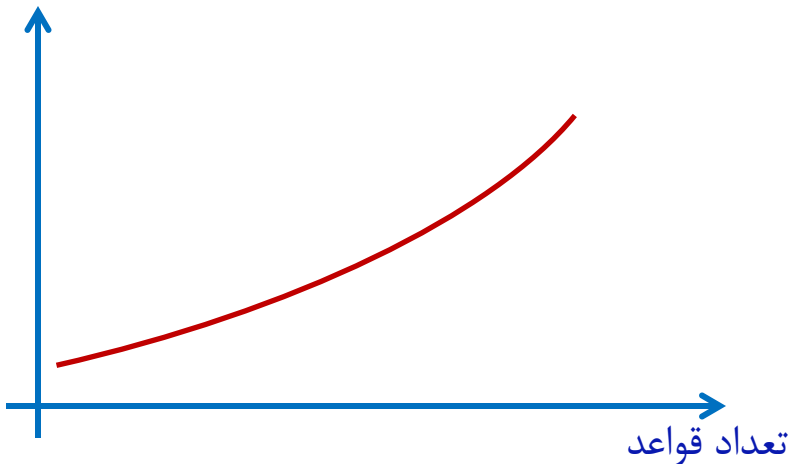
❑ منشأ فزونکاری (دلایل بروز فزونکاری) در DBMS

❑ انجام نگاشت‌ها (ناشی از معماری سه سطحی)

❑ اعمال قواعد جامعیتی

❑ اعمال ضوابط امنیت داده‌ها (در سطح DBMS)

کار سیستم





## □ IRها [ICها] در مدل رابطه‌ای

۱- قواعد [محدودیت‌های] عام: ناوابسته به داده‌های محیط: فراقواعد – متاقواعد (MetaRules)

۲- قواعد [محدودیت‌های] خاص: وابسته به داده‌های محیط: قواعد کاربری (User Defined)

یا قواعد فعالیت‌های محیط (Business Rules)

## □ قواعد عام در مدل رابطه‌ای

□ قاعده C1: قاعده جامعیت موجودیتی

□ قاعده C2: قاعده جامعیت ارجاعی



# قواعد عام در مدل رابطه‌ای – قاعده جامعیت موجودیتی C1

بخش پنجم: مفاهیم اساسی مدل داده رابطه‌ای

۴۵

## قاعده (محدودیت) C1 – قاعده جامعیت موجودیتی (Entity IR)

□ ناظر است به PK.

□ هیچ جزء تشکیل‌دهنده PK نباید هیچ مقدار (Null) داشته باشد.

□ دلیل:

✓ PK عامل تمییز تاپل‌ها است.

✓ تاپل در مدل رابطه‌ای نمایشگر نمونه موجودیت است. عامل تمییز خود نمی‌تواند ناشناخته باشد.

✓ PK عامل تمییز نمونه موجودیت‌ها است.

۱- محدودیت یکتایی مقدار (با UNIQUE

□ مکانیزم اعمال C1: اعلان PK به سیستم **کنترل می‌کند** فقط این محدودیت کنترل می‌شود)

۲- محدودیت هیچ‌مقدارناپذیری



## قواعد عام در مدل رابطه‌ای – قاعده جامعیت ارجاعی C2

بخش پنجم: مفاهیم اساسی مدل داده رابطه‌ای

۴۶

### □ قاعده (محدودیت) C2 – قاعده جامعیت ارجاعی (Referential IR)

□ ناظر است به FK.

□ اگر  $A_i$  در  $R_2$ ، کلید خارجی باشد،  $A_i$  باید در  $R_1$  مقدار قابل انطباق (Matchable Value) داشته باشد.

□ به عبارت دیگر باید هر مقدار معلوم  $A_i$  در  $R_2$ ، در  $R_1$  نیز وجود داشته باشد. البته در عمل می‌تواند در  $R_2$  مقدار آن Null باشد (البته اگر جزء تشکیل‌دهنده کلید  $R_2$  نباشد).

□ دلیل:

- FK عامل ارجاع است؛ ارجاع به نمونه موجودیت (ارجاع مقداری و نه ارجاع از طریق اشاره‌گر).
- در واقعیت نمی‌توان به نمونه موجودیت ناموجود ارجاع داد.



# قواعد عام در مدل رابطه‌ای – قاعده جامعیت ارجاعی C2 (ادامه)

بخش پنجم: مفاهیم اساسی مدل داده رابطه‌ای

۴۷



**STUD** (STID, ...)

777,	...
888,	...
444,	...

**SCR** (STID, COID, ...)

777	CO1	....
...	...	....
444	CO4	....

**INSERT INTO SCR**

**VALUES** ('999', 'CO9', ...)

□ چون برای 999 مقدار قابل انطباق در STUD وجود ندارد، پس این درخواست رد می‌شود.



# قواعد عام در مدل رابطه‌ای – قاعده جامعیت ارجاعی C2 (ادامه)

بخش پنجم: مفاهیم اساسی مدل داده رابطه‌ای

۴۸

□ برای اعمال قاعده C2 در مدل رابطه‌ای لازم است:

۱- معرفی FK (ها) به سیستم

۲- دادن گراف ارجاع

۳- مشخص کردن روش اعمال در عملیات حذف و

بهنگام‌سازی مقدار کلید اصلی

(در درج روش خاصی لازم نیست و در صورت عدم

وجود تاپل مرجع، درخواست رد می‌شود.)

**CREATE Relation SCR**

(STID CHAR(6) NOT NULL

COID CHAR(6) NOT NULL

TR CHAR(1)

YR CHAR(5)

GR DEC(2, 2))

**CHECK** (0 <= GR <= 20)

**PRIMARY KEY** (STID, COID)

**FOREIGN KEY** **STID** REFERENCES STUD

ON DELETE CASCADE

ON UPDATE CASCADE

**FOREIGN KEY** **COID** REFERENCES COUR

ON DELETE CASCADE

ON UPDATE CASCADE

گراف ارجاع

روش اعمال (انتشار عمل)





# قواعد عام در مدل رابطه‌ای – قاعده جامعیت ارجاعی C2 (ادامه)

بخش پنجم: مفاهیم اساسی مدل داده رابطه‌ای

۴۹

روش‌های اعمال C2 در حذف (بعضاً در به‌هنگام‌سازی): □

## ۱- روش CASCADE: انتشاری یا تسلسلی

در این روش با حذف تاپل مرجع، تمام تاپل‌های رجوع کننده به آن حذف می‌شوند.  
هر چه گراف ارجاع سنگین‌تر باشد، کار سیستم در اینجا بیشتر است.

```
DELETE FROM STUD  
WHERE STID='444'
```



```
DELETE FROM SCR  
WHERE STID='444'
```

## ۲- روش RESTRICTED: روش منوط به ... (یا مشروط به ...) یا روش تعویقی

در این روش اگر بخواهیم تاپل مرجع را حذف کنیم، ابتدا باید تاپل‌های رجوع کننده به آن حذف شوند. (تا زمانی که تاپل(های) رجوع کننده وجود داشته باشد، تاپل مرجع حذف نشود)



# قواعد عام در مدل رابطه‌ای – قاعده جامعیت ارجاعی C2 (ادامه)

بخش پنجم: مفاهیم اساسی مدل داده رابطه‌ای

۵۰

❑ روش‌های اعمال C2 در حذف (و بعضاً در به‌هنگام‌سازی):

۳- روش **SET TO NULL**: روش هیچ‌مقدارگذاری یا **Nullifying**

در این روش با حذف تاپل مرجع، FK در تاپل‌های رجوع کننده Null می‌شود به شرط آنکه FK جزء سازنده PK نباشد.

۴- روش **SET TO DEFAULT**: روش درج پیش‌فرض

در این روش، با حذف تاپل مرجع، FK با مقدار پیش‌فرض جاگذاری می‌شود به شرط آنکه FK جزء سازنده PK نباشد.



# قواعد عام در مدل رابطه‌ای – قاعده جامعیت ارجاعی C2 (ادامه)

بخش پنجم: مفاهیم اساسی مدل داده رابطه‌ای

۵۱

□ روش‌های اعمال C2 در حذف (و بعضاً در بهنگام‌سازی):

۵- روش **NO ACTION**: عدم اقدام

برای این روش دو پیشنهاد داده شده است:

۵-۱- عدم اقدام مطلق: مثلاً مجاز نبودن عمل حذف تاپل مرجع و نمایش خطا

۵-۲- انجام عمل خواسته شده و نه اقدام دیگر: تاپل مرجع حذف بشود ولی اقدام دیگری انجام

نشود. در این مورد طراح-پیاده‌ساز می‌پذیرد که موقتاً محدودیت C2 نقض شود.

□ در حالت وجود **چرخه ارجاع** کدام روش انجام شدنی است؟



□ نمی‌توان روش RESTRICTED را در حالت کلی اعمال کرد. با روش CASCADE هم ممکن است

تاپل‌های ناخواسته حذف شود.

□ در این مواقع NO ACTION را انتخاب می‌کنیم.



☐ قواعد خاص در مدل رابطه‌ای:

☐ محدودیت دامنه‌ای (میدانی)


☐ محدودیت صفتی

☐ محدودیت رابطه‌ای

☐ محدودیت پایگاهی



## محدودیت دامنه‌ای (میدانی)

 این محدودیت ناظر است به دامنه، مشخص‌کننده نوع و طیف مقادیر دامنه

 در همان دستور CREATE DOMAIN اعلان می‌شود.

دستور ایجاد دامنه `CREATE DOMAIN GRADE DEC(2, 2) DEFAULT '...?'`

نام محدودیت (اختیاری) `CONSTRAINT GRADECONST`

`CHECK VALUE BETWEEN (0, 20)`

دستور حذف دامنه `DROP DOMAIN GRADE`





## □ محدودیت صفتی [استونی]

□ این محدودیت ناشی می‌شود از محدودیت دامنه‌اش

□ صفت می‌تواند محدودیت‌های دیگری هم داشته باشد، به شرطی که ناقض محدودیت دامنه‌ای‌اش نباشد.

محدودیت‌های ناظر به صفت:



۱- صفت  $Y$  تابع صفت  $X$  است (وابستگی تابعی دارد).

۲- مقادیر صفت  $B$  زیرمجموعه‌ای از مقادیر صفت  $A$  است. وابستگی شمولی  $B\{\text{values}\} \subseteq A\{\text{values}\}$

۳- صفت سن کاهش نمی‌یابد (محدودیت پردازشی).

محدودیت ۱ و ۲، محدودیت‌های وضعیتی هستند ولی محدودیت ۳، محدودیت گذاری است.



❑ محدودیت صفتی را چگونه می‌توان اِعلان یا اِعمال کرد؟

۱- با تعریف دامنه‌اش اِعلان می‌شود.

۲- در همان دستور CREATE TABLE با عبارت CHECK اِعلان می‌شود.

مثال  
جدول انتخاب درس

**CREATE TABLE STCOT**

(STID ...

COID ...

TR ...

GR ...)


**CHECK** (0 <= GR <= 20)


۳- با ASSERTION اِعلان می‌شود.

۴- با TRIGGER به سیستم داده می‌شود (اجرای).



## محدودیت رابطه‌ای

 ناظر است به تاپل‌های یک رابطه (درون رابطه‌ای Intra-relational).

 حیطه اعمالش یک رابطه است (مقادیر مجاز یک متغیر رابطه‌ای را مشخص می‌کند).

تعداد واحد درس‌های عملی حداکثر ۲ واحد است.




تهیه‌کنندگان ساکن شهر C2 نمی‌توانند مقدار وضعیت بیش از ۱۵ داشته باشند.








## محدودیت پایگاهی


 ناظر است به تاپل‌های بیش از یک رابطه که به نحوی با هم ارتباط معنایی [منطقی] دارند.

 مثال رابطه STT و STCOT

یا رابطه S و SP

 مثال دانشجوی رشته کامپیوتر نمی‌تواند درس آمار و احتمال را از گروه آموزشی D13 (دانشکده ریاضی)

انتخاب کند. رابطه‌های دخیل: STUD، COUR و SCR

 مثال تهیه‌کننده ساکن شهر C7 با وضعیت کمتر از ۱۵، نمی‌تواند قطعه آبی رنگ با وزن بیش از ۱۰ گرم به تعداد بیش از ۱۰۰ تهیه کند.

 محدودیت‌های رابطه‌ای و پایگاهی چگونه اعمال می‌شوند؟

■ با ASSERTION (اعلانی)

■ با TRIGGER (اجرایی)



## اظهار – ASSERTION ☐

☐ امکانی است اِعلانی برای بیان محدودیت‌های رابطه‌ای و پایگاهی [او صفتی]

```
CREATE ASSERTION name  
    [ BEFORE | AFTER action  
    ON tablename ]  
    CHECK condition(s)
```

☐ در قسمت *condition(s)* می‌توان یک شرط ساده، یک عبارت بولی شامل چند شرط و نیز یک عبارت SELECT معتبر نوشت (همانطور که بعد از عبارت WHERE نوشته می‌شود).

☐ دستور حذف اظهار

```
DROP ASSERTION name
```



## امکانات بیان محدودیت‌ها – اظهار (ادامه)

بخش پنجم: مفاهیم اساسی مدل داده رابطه‌ای

۵۹

با این اظهار، محدودیت یکتایی مقادیر صفت کد ملی STNATID اعلان می‌شود.



```
CREATE ASSERTION UNC-CHECK  
CHECK ( UNIQUE (SELECT STNATID FROM STT) )
```

با این اظهار این محدودیت که «جمع واحدهای انتخابی دانشجو در هر ترم-سال نباید بیش از ۲۰ واحد

باشد»، اعلان می‌شود.



```
CREATE ASSERTION TOTCRED-CHECK
```

...

```
CHECK (NOT EXISTS (SELECT STID  
FROM COUR JOIN STCOT  
GROUP BY (STID, TR, YR)  
HAVING SUM(CREDIT) > 20) )
```



## TRIGGER – [راه‌انداز] ☐

☐ امکانی است اجرایی برای اعمال محدودیت‌های [صفتی،] رابطه‌ای و پایگاهی.

```
CREATE TRIGGER name
  {BEFORE | AFTER | INSTEAD OF}
  {INSERT | DELETE | UPDATE OF columnlist}
  ON tablename
  [REFERENCING { OLD ROW | NEW ROW | OLD TABLE | NEW TABLE } AS name ]
  [FOR EACH {ROW | STATEMENT}]
  {(WHEN condition(s)
    SQL 2008 Procedure
  )}
```

☐ مبنای تئوریک TRIGGER: مفهوم قاعده فعال [مفهوم محوری است در ADBMS ها]

ساختار (قاعده ECA): if **E**vent on **C**ondition, then **A**ction





این رهانا این محدودیت را که «حقوق کارمند هیچگاه کاهش نمی‌یابد» اعمال می‌کند.



**EMPL (E#, ENAME, ENC, ..., EPHONE, EMPSAL)**

**CREATE TRIGGER EMP-PAY-TRIG**

**BEFORE UPDATE OF EMPSAL**

**ON EMPL**

**REFERENCING OLD AS OEMPL, NEW AS NEMPL**

**FOR EACH ROW**

**(WHEN OEMPL.EMPSAL > NEMPL.EMPSAL**

**SIGNAL.SQL State '7005' ('salary cannot be decreased')**

**)**

مطالعه یادداشتهای تکمیلی در خصوص رهانا 



**پرسش و پاسخ ...**

**amini@sharif.edu**