

# به نام آنکه جان را فکرت آموخت



## بخش هفتم: مفاهیم اساسی مدل داده رابطه‌ای

مرتضی امینی

نیمسال اول ۱۴۰۱-۱۴۰۰

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



☐ RDM مبنای تئوریک RDB و RDBMS

☐ واضع مدل: F. Codd

☐ مفاهیم زیر در طی سه بخش باقیمانده از این درس مرور می‌شوند:

☐ رابطه (Relation)

☐ دامنه (میدان)

☐ رابطه نرمال و غیرنرمال

☐ کلید در مدل رابطه‌ای

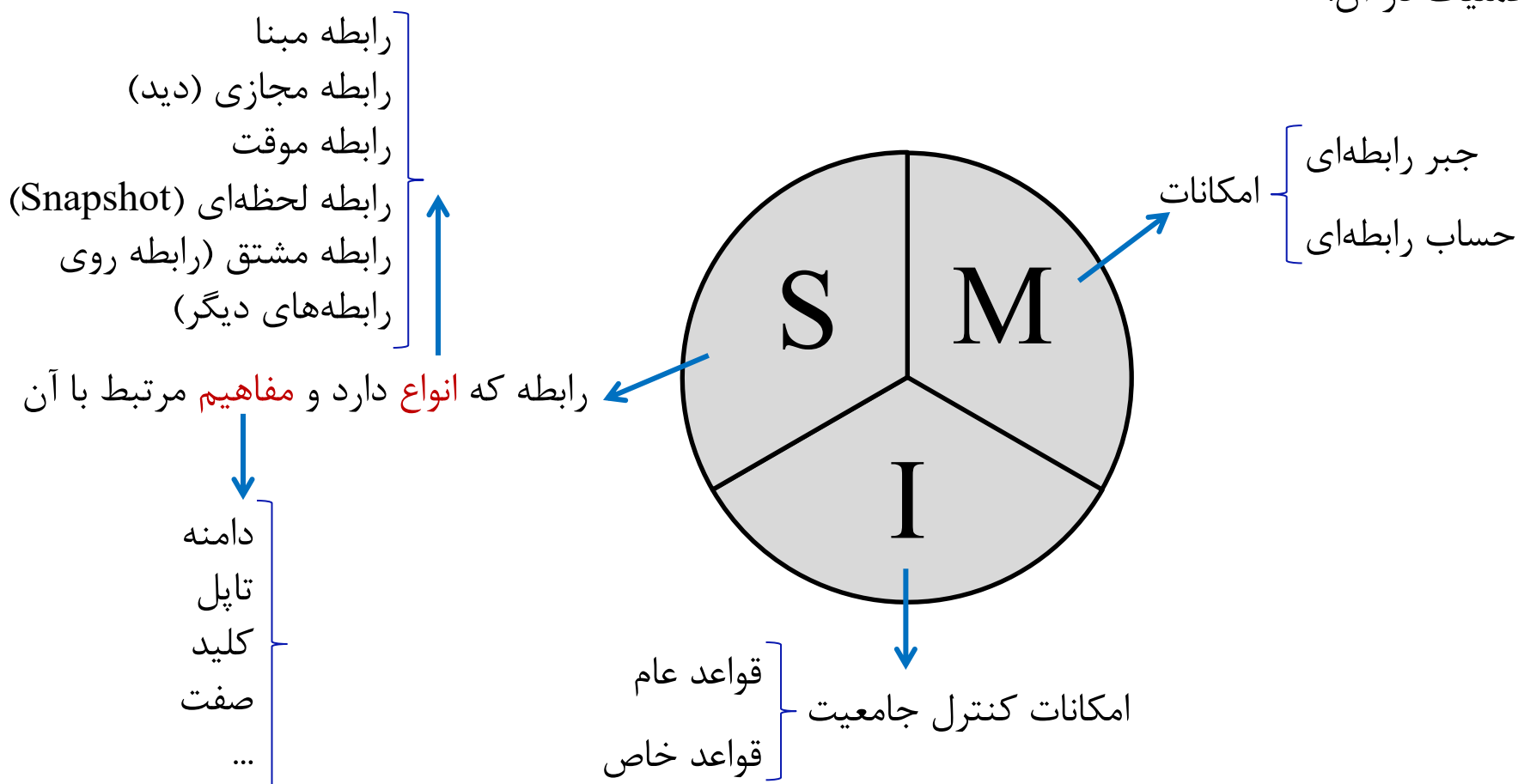
☐ قواعد جامعیت رابطه‌ای

☐ عملیات در RDB ←  
جبر رابطه‌ای  
حساب رابطه‌ای

☐ نرمال‌ترسازی



✓ **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 کارت‌زین چند مجموعه



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

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

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

به نحوی که  $d_{ji} \in D_j, \dots, d_{1i} \in D_1$



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

777      st7      bs      phys      d11

⋮      ⋮      ⋮      ⋮      ⋮

444      st4      bs      comp      d14

یک تاپل ۵-تایی →



(۲) [C.J. 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 که با  $R(A_1, \dots, A_m)$  نمایش داده می‌شود.
- بدنه [پیکر] (Body): مجموعه‌ای است از تاپل‌ها [همان مجموعه در تعریف اسلاید قبل].

رابطه دانشجو



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

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

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



### بخش هفتم: مفاهیم اساسی مدل داده رابطه‌ای

مجموعه عنوان را با  $H_R$  یا  $R(H)$  نیز نمایش می‌دهیم. به  $R(H)$  (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.



(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]

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

$R(A, B) \rightarrow$  متغیر رابطه‌ای

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

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





## □ 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
نوع و 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- تاپل‌ها [در بدنه] نظم ندارند (مرتب نیستند) [چون مجموعه است].

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

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


۴- **فرض:** تمام صفات رابطه (نرمال)، تک مقدار 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.

## تفاوت‌های مفهوم رابطه و اصطلاح جدول: □


□ ۳ ویژگی اول رابطه، ۳ تفاوت



## مفهوم دامنه (میدان)

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

 معادل است با مفهوم Data Type در تئوری انواع (Type Theory).

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

مفروض  $R(H)$

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



□ **تمرین:** مثالی از یک رابطه ۵-تایی که

□ دو صفت آن از یک دامنه باشد.

□ سه صفت آن از یک دامنه باشد.

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

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



مثالی از شمای پایگاه رابطه‌ای

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

(در مدل تئوریک)

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

**تذکر:** توجه شود که در عمل این دستور را نداریم و به جای آن دستور CREATE TABLE داریم.

CREATE RELATION COT ....

CREATE RELATION STCOT ...



دستورات زیر در SQL مطالعه شود.



CREATE DOMAIN ☐

ALTER DOMAIN ☐

DROP DOMAIN ☐

مزایای مفهوم دامنه از دیدگاه مهندسی نرم‌افزار بررسی شود.





## □ 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 یا ساده) باشند.

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



صفت ساده چندمقداری

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 مرکب چندمقداری  
P# , QTY

NNSP ( S# , **PQTY** )

SP ( S# , P# , QTY )

S#	PQTY
s1	{ p1 100 p2 90 p3 50 }
s2	{ p1 60 p2 90 }
s3	p1 150

یک تاپل



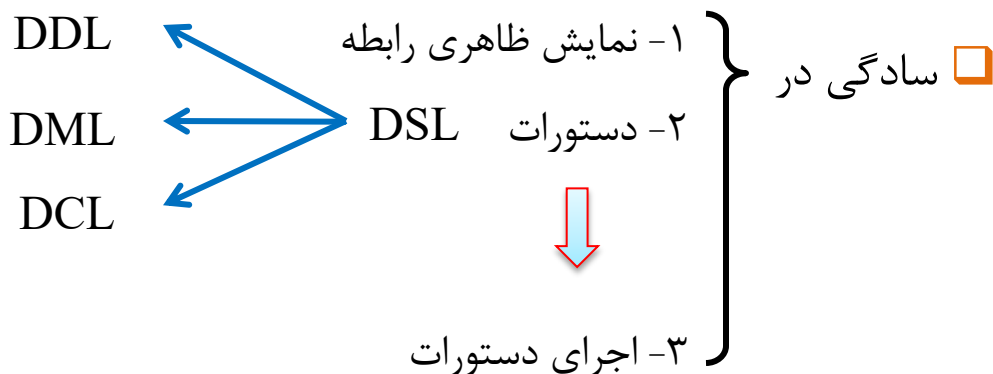
تبدیل به رابطه  
نرمال

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».

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

درج کن □



ادامه مثال 

$I_1 : \text{INSERT INTO } \begin{Bmatrix} \text{NNSP} \\ \text{SP} \end{Bmatrix}$   
 $\text{TUPLE (S4 , P4 , 40);}$

$I_2 : \text{INSERT INTO SP}$   
 $\text{TUPLE (S2 , P3 , 30);}$

→ امکان پذیر

$I_2 : \text{INSERT INTO NNSP}$   
 $\text{TUPLE (S2 , P3 , 30);}$

→ 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

بخش هفتم: مفاهیم اساسی مدل داده رابطه‌ای

۲۰

نوع رابطه	مزایا	معایب
نرمال	سادگی (۱- ... ۲- ... ۳-...)	طولانی شدن کلید
	تقارن صفات (پیاده‌سازی در سطح فایلینگ ساده‌تر) (نقش تمام صفات در عبارت WHERE وقتی که شرط جستجو را با theta می‌دهیم، یکسان است، زیرا همه تک‌مقداری‌اند. SELECT.... FROM .... WHERE A<(=)(>) 'Single Value' چنین تقارنی در رابطه غیرنرمال وجود ندارد.)	افزونگی (ادراکی یا منطقی) (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.) سنگین و زمانگیر کردن کار طراحی منطقی پایگاه داده‌ها کاهش سرعت بازیابی در بعضی از پرسش‌ها
غیرنرمال	[عکس معایب رابطه نرمال]	پیچیدگی (۱- ... ۲- ... ۳-...) عدم تقارن صفات



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

بخش هفتم: مفاهیم اساسی مدل داده رابطه‌ای

۲۱

□ 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؟





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

۱- سوپرکلید (آبر کلید): SK

۲- کلید کاندید (کلید نامزد): CK

۳- کلید اصلی: PK

۴- کلید خارجی: FK



# کلید در مدل رابطه‌ای – سوپر کلید

۲۳

بخش هفتم: مفاهیم اساسی مدل داده رابطه‌ای

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

$H_R$

نمایشگر مقادیر  
صفات  $S$  از تاپل  $t_i$

سوپر کلید (Super Key)

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



اگر به ازای هر دو تاپل دلخواه و متمایز  $t_i$  و  $t_j$  از  $R$  داشته باشیم  $t_i(S) \neq t_j(S)$ ، آنگاه  $S$  یک سوپرکلید است.

اگر  $N$  تعداد SKهای رابطه  $R$  باشد،  $N \geq 1$  است، زیرا در بدترین حالت خود  $H$  سوپرکلید می‌شود. چون

بدنه، مجموعه است و در مجموعه تاپل تکراری نداریم.

$$1 \leq N \leq 2^m - 1$$

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

۲۴

بخش هفتم: مفاهیم اساسی مدل داده رابطه‌ای

## کلید کاندید (Candidate Key) □

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



۱- یکتایی مقدار

۲- کاهش ناپذیری (Irreducibility) یا کمینگی (Minimality)

- $K \subseteq H_R$  کاهش ناپذیر است هرگاه هر زیرمجموعه محض از  $K$ ، خود یکتایی مقدار نداشته باشد.
- هر زیرمجموعه از  $H_R$  به نحوی که یک صفت را از آن حذف کنیم دیگر یکتایی مقدار نداشته باشد.



رابطه	کلید کاندید
STT	STID
COT	COID
STCOT	(STID, COID)
S	S#
P	P#
SP	(S#, P#)





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

۲۵

بخش هفتم: مفاهیم اساسی مدل داده رابطه‌ای

□ CKها بر اساس قواعد معنایی و محدودیت‌های حاکم بر محیط به دست 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)

۲۶

بخش هفتم: 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.

□ رابطه می‌تواند بیش از یک CK داشته باشد.

□ 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 ولی زیرمجموعه

یکدیگر نمی‌توانند باشند. (چرا؟)

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



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

۲۷

بخش هفتم: مفاهیم اساسی مدل داده رابطه‌ای

☐ **نقش کلید کاندید:** تضمین‌کننده عملیات تاپلی (و نه مجموعه‌ای) یا امکان ارجاع به تک تاپل در رابطه را فراهم می‌نماید.

☐ 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. لذا از لحاظ **تئوری** باید داشته باشیم:



**CREATE RELATEION EMPROJ**

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

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

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

☐ **تذکر:** تئوری این را می‌گوید ولی در عمل، سمپادها نمی‌پذیرند و

صرفاً می‌توان یک CK را به عنوان PK در آنها تعریف کرد. بقیه CKها را می‌توان UNIQUE تعریف کرد.



## 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 هفتم: مفاهیم اساسی مدل داده رابطه‌ای

۲۹

□ دلایل لزوم انتخاب کلید اصلی:

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

۲- ایجاد شاخص اتوماتیک روی PK.

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

کننده است. کلید CKای که این محدودیت را روی آن اعمال می‌کنند می‌شود PK.

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



# کلید در مدل رابطه‌ای – کلید خارجی

بخش هفتم: مفاهیم اساسی مدل داده رابطه‌ای

۳۰

## □ کلید خارجی (Foreign Key)

□ **در عمل:**  $T_2.C$  در  $T_2$ ، کلید خارجی است هرگاه در  $T_1$ ، کلید اصلی باشد.

□ **در 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.

□ 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
STCOT	STID	STT
STCOT	COID	COT
SPJ	S#	S
SPJ	P#	P
SPJ	J#	J



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

۳۱

بخش هفتم: مفاهیم اساسی مدل داده رابطه‌ای

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

□ معرفی کلید خارجی با عبارت FOREIGN KEY انجام می‌شود.

□ نقش **کلید خارجی**: برای نمایش ارتباطهای صریح بین نوع موجودیت‌ها (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

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



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

بخش هفتم: مفاهیم اساسی مدل داده رابطه‌ای

۳۲

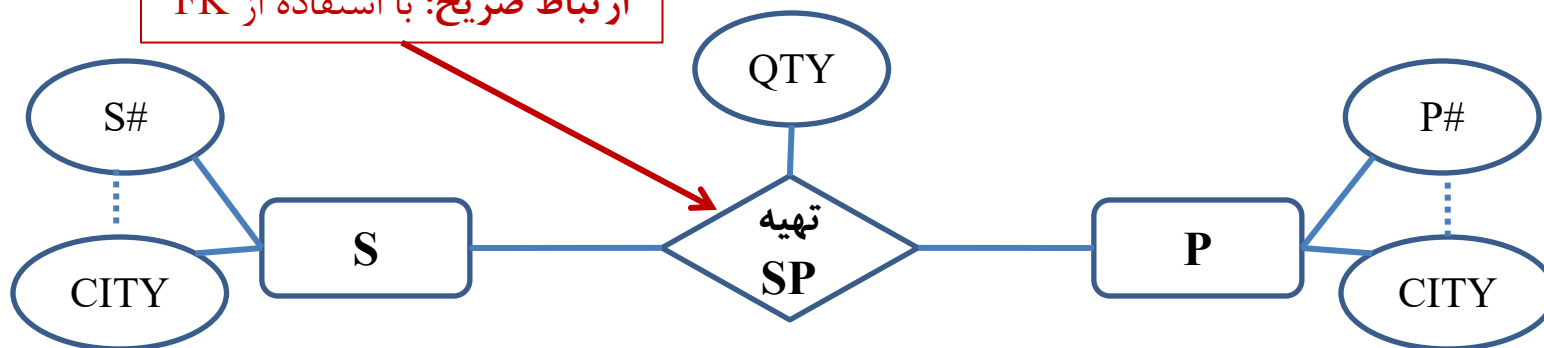
□ آیا FK تنها امکان نمایش ارتباط است یا امکان دیگری هم وجود دارد؟

□ 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.

ارتباط صریح: با استفاده از FK



→ 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

۳۳

بخش هفتم: مفاهیم اساسی مدل داده رابطه‌ای

### مفهوم گراف ارجاع

FK امکانی است برای ارجاع از یک رابطه به رابطه‌ای دیگر

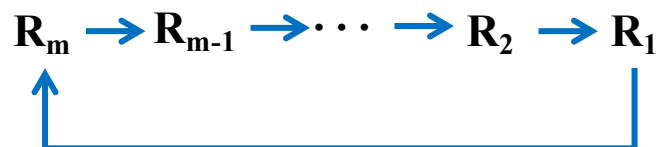
هر مقدار معلوم FK، امکانی است برای ارجاعِ **مقداری**، از تاپل(هایی) از رابطه(هایی) به تاپلی از رابطه(هایی).

**تعریف** **گراف ارجاع** امکانی است برای نمایش ارجاعات بین رابطه‌ها که در آن هر گره، نمایانگر یک رابطه و هر یال جهت‌دار، نمایانگر ارجاع از یک رابطه (حاوی کلید خارجی) به رابطه دیگر (حاوی کلید کاندید) است.

$P \leftarrow SP \rightarrow S$



شکل کلی مسیر ارجاع:



با این ارجاع می‌شود چرخه ارجاع

مسیر ارجاع می‌تواند **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.

چرخه ارجاع بین دو رابطه کارمند و اداره.



شماره کارمند مدیر اداره

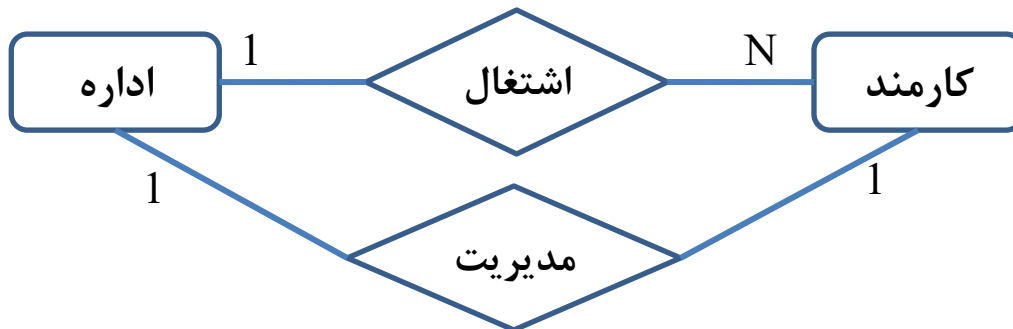
DEPT (D#, DTITLE, ..., E#)  
Unique

شماره اداره محل کار

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

DEPT  $\longleftrightarrow$  EMPL

□ بر اساس کدام مدل‌سازی این طراحی انجام شده است؟





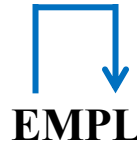
بخش هفتم: مفاهیم اساسی مدل داده رابطه‌ای

چرخه ارجاع تک‌رابطه‌ای کارمند با خودش.



شماره مدیر

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



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

□ 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 هفتم: 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؟

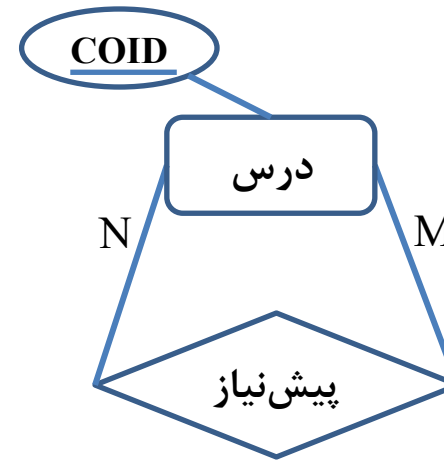


صرف وجود دور در ERD چرخه ارجاع ایجاد نمی‌شود.



COT (COID, ...)

COPRECO(COID, PRECO)



□ در چه وضعی چرخه ارجاع پدید می‌آید؟

□ باید به چندی ارتباط‌ها توجه شود.



**پرسش و پاسخ . . .**

**amini@sharif.edu**