

به نام آنکه جان را فکرت آموخت



**بخش سوم :**  
**طراحی منطقی**

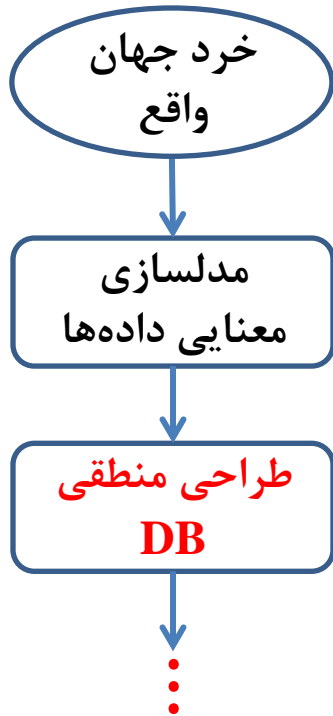
مرتضی امینی

نیمسال دوم ۹۴-۹۵

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



بخش سوم: طراحی منطقی پایگاه داده‌ها



مدل‌سازی داده‌ها می‌تواند در سطوح انتزاعی مختلفی صورت پذیرد.

سطح پایین‌تر از سطح مدل‌سازی معنایی داده‌ها، سطح طراحی منطقی است.

**سطح طراحی منطقی:** برای نمایش پایگاه داده‌ها در این سطح از مفاهیمی استفاده می‌شود که مستقل از

مفاهیم محیط فایلینگ پایگاه داده‌ها است.



بحث مقدماتی: دیدگاه کاربردی [و نه تئوریک]

برای طراحی منطقی پایگاه داده‌ها (و همچنین عملیات در DB و کنترل DB) هم امکان خاصی لازم است: یک **مدل داده (DM)**، که شامل یک **ساختار داده (DS)** است.

مفاهیم مطرح در طراحی منطقی پایگاه داده‌ها

ساختار داده جدولی : TDS

پایگاه داده جدولی : TDB

زبان پایگاهی جدولی : TDBL



چرا ساختار داده (در معنای عام)؟

برای نمایش نوع موجودیت‌ها و ارتباط بین آنها در سطح  
منطقی }  
فیزیکی }

دلایل لزوم ساختار داده (DS) در حیطه پایگاهی:

۱- تامین کننده محیط فرافایلی (محیط انتزاعی)

۲- مبنا و چارچوب طراحی منطقی DB

۳- مبنا و چارچوب طراحی زبان پایگاه داده‌ها DBL

---

۴- مبنا و چارچوب طراحی خود DBMS

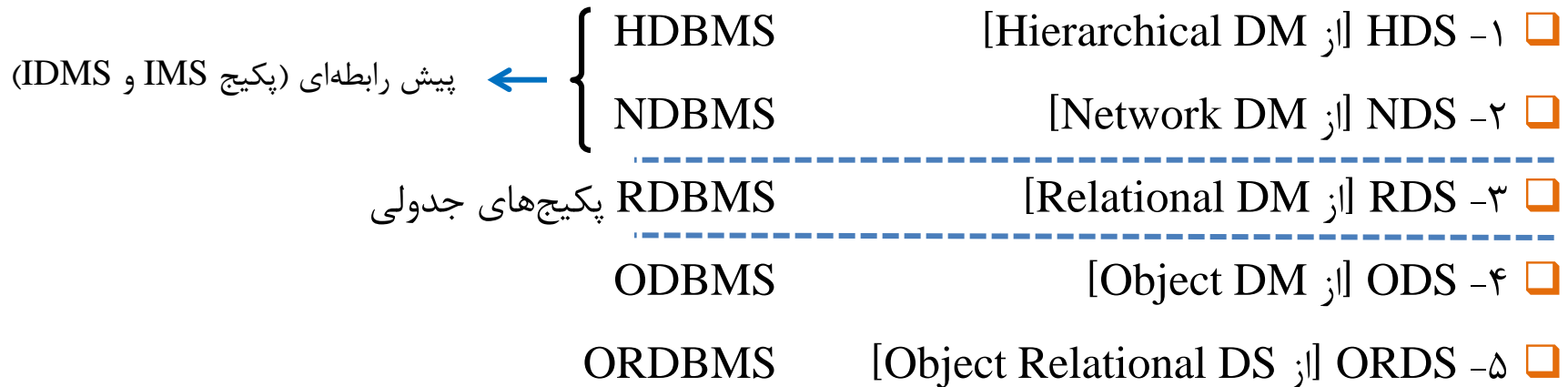
۵- ضابطه‌ای است برای مقایسه سیستم‌ها و ارزیابی آنها

۶- مبنایی است برای ایجاد و گسترش تکنیک‌های طراحی DB

۷- ...



□ DSها [در حیطه دانش و تکنولوژی DB]:



□ TDS - ساختار داده جدولی:

□ عنصر ساختاری اساسی در Relational Model (RM): مفهوم **رابطه**

□ رابطه [Relation]: یک مفهوم ریاضی است ...

□ اما از دید کاربر [در عمل]: نمایش جدولی دارد.

▪ فعلا به جای RDS می‌گوییم TDS.

(البته رابطه و جدول تفاوت‌های جدی با هم دارند که در مباحث بعدی درس بدان پرداخته می‌شود.)



## اصطلاحات TDS:

نوع جدول ← { نام جدول  
نام و نوع ستون ها } برای نمایش نوع { موجودیت و/یا  
ارتباط

سطر ← برای نمایش نمونه { موجودیت  
ارتباط

ستون ← برای نمایش صفت

## عنصر ساختاری اساسی:

هر DS حداقل یک **عنصر ساختاری اساسی** دارد.

عنصری است که به کمک آن نوع موجودیت، نوع ارتباط، و یا هر دو آنها را نمایش می‌دهیم.



TDS فقط یک عنصر ساختاری اساسی دارد: همان **نوع جدول**

**نکته:** صفت **شناسه** در نوع موجودیت‌ها، حکم **کلید** را در جدول دارد.



## TDB چیست؟

از دید کاربر

$$\left\{ \begin{array}{l} \text{طراح} \\ \text{پیاده ساز DB} \\ \text{برنامه ساز AP} \end{array} \right\}$$

از لحاظ نوع: مجموعه‌ای است از تعدادی **نوع جدول** (که آنها را طراحی می‌کنیم)

از لحاظ محتوای داده‌ای [در سطح نمونه]: مجموعه‌ای است از نمونه‌های متمایز یک [چند] **نوع سطر**

▪ نوع سطر را همان نوع جدول مشخص می‌کند.



مفهوم کلید در مدل داده جدولی تعریف نشده است و برگرفته از مفاهیم تعریف شده در مدل داده‌ای رابطه است.

[از دیدگاه کاربردی] **کلید** امکان دسترسی به تک نمونه (از یک نوع موجودیت یا نوع ارتباط) را فراهم می‌نماید. لذا مقدار آن در سطرهاى جدولِ مبینِ موجودیت یا ارتباط، **یکتا** است.

[از دیدگاه کاربردی] یک یا چند صفت (ستون) تشکیل **کلید اصلی** را در یک جدول می‌دهند اگر مقادیر آن(ها) در سطرهاى جدول **یکتا** و **معلوم** باشد.

در مواقعی ممکن است بیش از یک کلید داشته باشیم. یکی از کلیدها که مقادیرش در همه سطرها معلوم است را کلید اصلی می‌گیریم (بقیه را با یکتا بودن مقادیر – با استفاده از UNIQUE در SQL – مشخص می‌نماییم).

در معرفی مدل داده رابطه‌ای، با انواع کلید و تعاریف آنها آشنا می‌شوید.

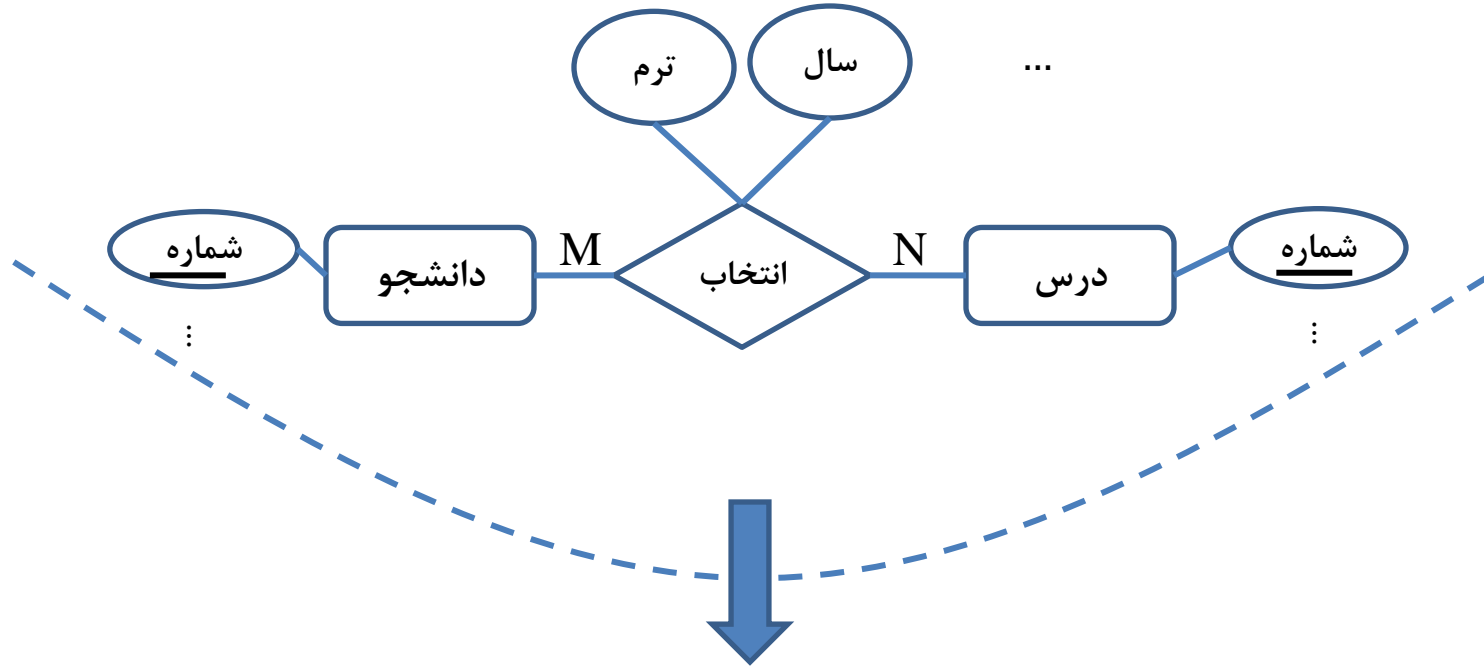




# طراحی منطقی با TDS – رابطه چند به چند

بخش سوم: طراحی منطقی پایگاه داده‌ها

چندی M:N



مساله: تبدیل به TDB [با TDS]

□ سه نوع جدول لازم داریم: ← } برای هر نوع موجودیت یک نوع جدول  
 برای نوع ارتباط M:N یک نوع جدول



## طراحی منطقی با TDS – رابطه چند به چند (ادامه)

بخش سوم: طراحی منطقی پایگاه داده‌ها

۱۰

خط زیرین  
نمایانگر کلید

STT	<u>STID</u>	STNAME	STLEV	STMJR	STDEID
	777	st7	bs	phys	d11
	888	st8	ms	math	d12
	444	st4	ms	phys	d11
	:	:	:	:	:

COT	<u>COID</u>	COTITLE	CREDIT	COTYPE	CEDEID
	:	:	:	:	:
	co3	programming	4	t (تئوری)	d13
	:	:	:	:	:



## طراحی منطقی با TDS – رابطه چند به چند (ادامه)

بخش سوم: طراحی منطقی پایگاه داده‌ها

۱۱

طبق قواعد معنایی محیط ممکن است سال و ترم هم جزو کلید باشند.

(در واقع اگر صفت چند مقداری مرکب برای رابطه باشند، جزو کلید محسوب می‌شوند.)



STCOT

<u>STID</u>	<u>COID</u>	<u>TR</u>	<u>YR</u>
:	:	:	:
888	co2	1	87
888	co3	1	87
444	co2	1	87

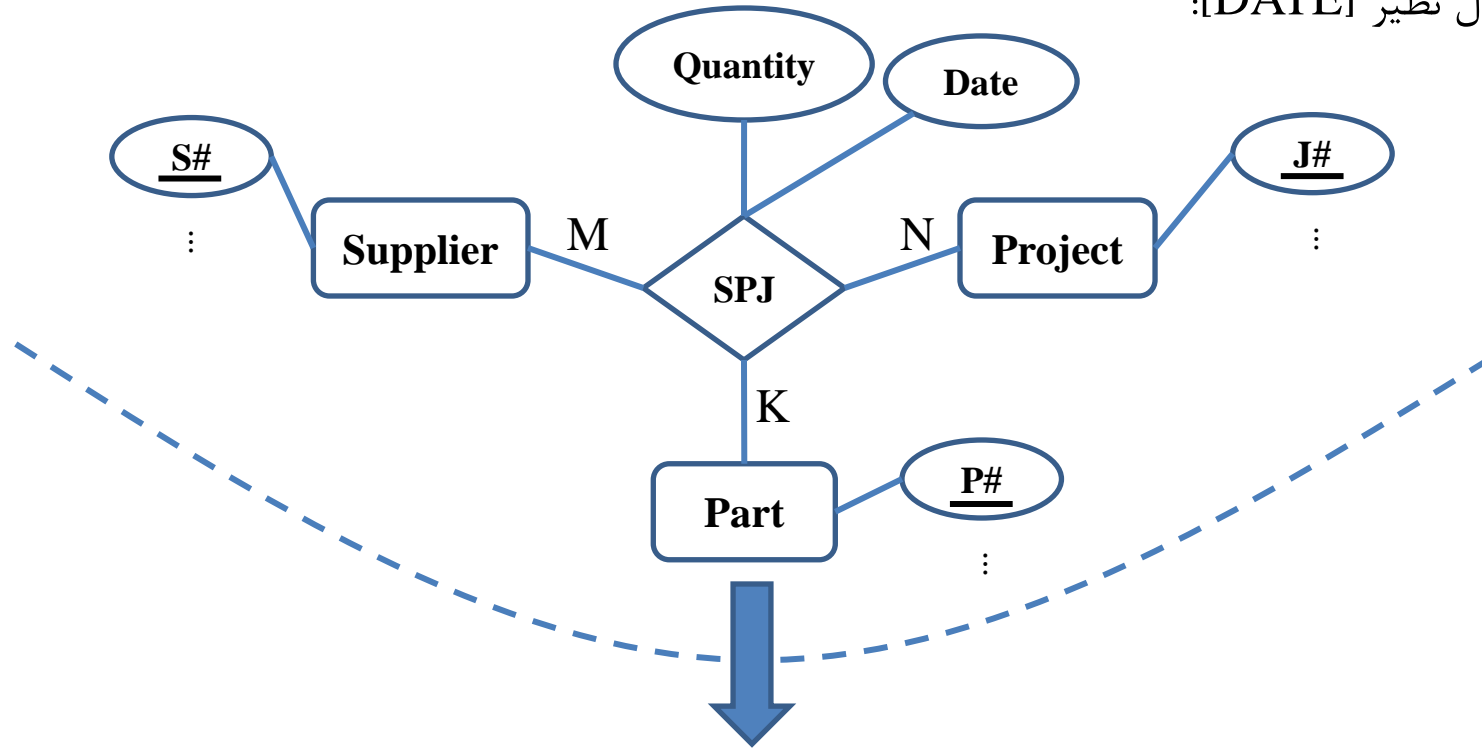


# طراحی منطقی با TDS – رابطه چند به چند (ادامه)

بخش سوم: طراحی منطقی پایگاه داده‌ها



مثال نظیر [DATE]:



مساله: تبدیل به TDB [با TDS]

چهار نوع جدول داریم: ← } برای هر نوع موجودیت یک نوع جدول  
 برای نوع ارتباط یک نوع جدول



# طراحی منطقی با TDS – رابطه چند به چند (ادامه)

بخش سوم: طراحی منطقی پایگاه داده‌ها

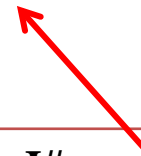
۱۳

Supplier	<u>S#</u>	SNAME	CITY	...
	s1	...	c1	...
	s2	...	c1	...
	:	:	:	:

طبق قواعد معنایی محیط ممکن است تاریخ هم جزو کلید بشود.  
(در واقع اگر صفت چند مقداری باشد، جزو کلید محسوب می‌شود.)

Part	<u>P#</u>	PNAME	CITY	...
	p1	...	c1	...
	p2	...	c2	...
	:	:	:	:

SPJ	<u>S#</u>	<u>P#</u>	<u>J#</u>	Date	QTY
	s1	p1	j1	d1	100
	s1	p1	j1	d2	50
	:	:	:	:	:



Project	<u>J#</u>	JNAME	CITY	...
	j1	...	c2	...
	j2	...	c1	...
	:	:	:	:

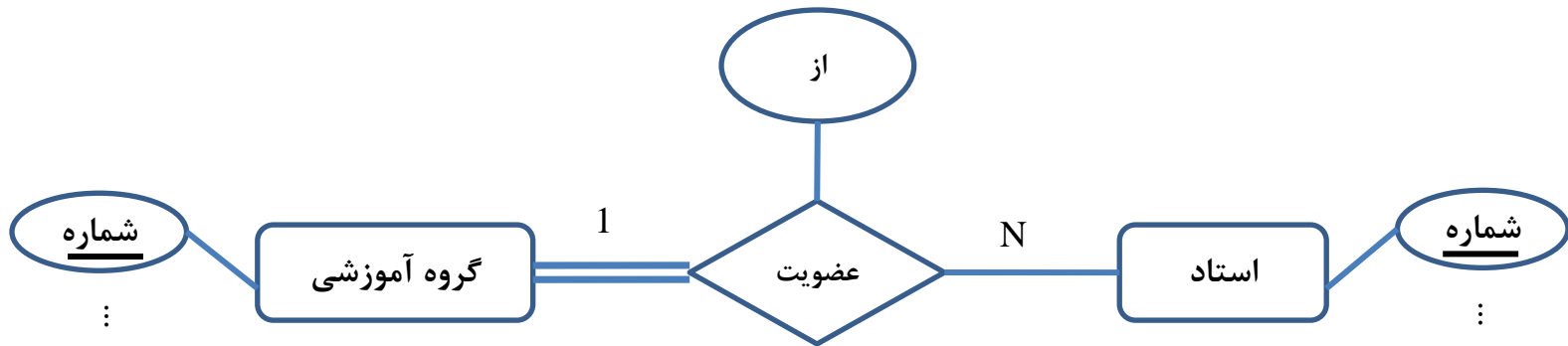


# طراحی منطقی با TDS – رابطه یک به چند

بخش سوم: طراحی منطقی پایگاه داده‌ها

۱۴

چندی 1:N



□ دو نوع جدول داریم: ←  
یکی برای نوع موجودیت سمت 1  
یکی برای نوع موجودیت سمت N و نیز خود ارتباط



# طراحی منطقی با TDS – رابطه یک به چند (ادامه)

بخش سوم: طراحی منطقی پایگاه داده‌ها

۱۵

DEPT	<u>DEID</u>	DETITLE	...	DEPHONE
	D11	Phys	...	...
	D12	Math	...	...
	:	:	:	:

PROF	<u>PRID</u>	PRNAME	RANK	...	FROM	<u>DEID</u>
	Pr100	...	استاد	...	d1	D13
	Pr200	...	استادیار	...	d2	D11
	Pr300	...	دانشیار	...	?	?

\* ستون DEID در جدول PROF **کلید خارجی** است و با خط چین مشخص می‌شود.

**کلید خارجی** [کاربردی]: ستون c از جدول T1 در جدول T2 کلید خارجی است هرگاه در جدول T1 کلید



اصلی باشد.

در چه حالاتی استفاده از سه نوع جدول قابل توجیه است؟





# طراحی منطقی با TDS – رابطه یک به یک

بخش سوم: طراحی منطقی پایگاه داده‌ها

۱۶

چندی 1:1



□ دو نوع جدول داریم: } یکی برای نوع موجودیت سمت 1 غیرالزامی  
یکی برای نوع موجودیت سمت 1 الزامی و نیز خود ارتباط





## طراحی منطقی با TDS – رابطه یک به یک (ادامه)

بخش سوم: طراحی منطقی پایگاه داده‌ها

۱۷

DEPT	<u>DEID</u>	DETITLE	...	DEPHONE	<u>PRID</u>
	D11	Phys	...	...	...
	D12	Math	...	...	...
	:	:	:	:	:

یک طرز طراحی ممکن:

PROF	<u>PRID</u>	PRNAME	RANK	...
	Pr100	...	استاد	...
	Pr200	...	استادیار	...
	Pr300	...	دانشیار	...
	:	:	:	:

طرزهای دیگر طراحی؟



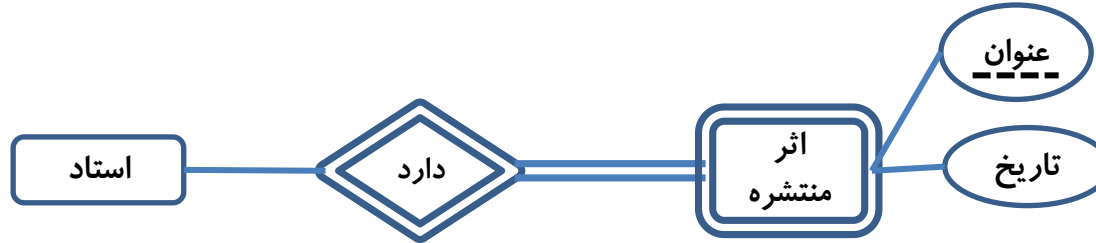


# طراحی منطقی با TDS – رابطه شناسا

بخش سوم: طراحی منطقی پایگاه داده‌ها

۱۸

رابطه شناسا (رابطه موجودیت ضعیف)



□ دو نوع جدول داریم: ← یکی برای نوع موجودیت قوی } یکی برای نوع موجودیت ضعیف و رابطه (حاوی شناسه موجودیت قوی)



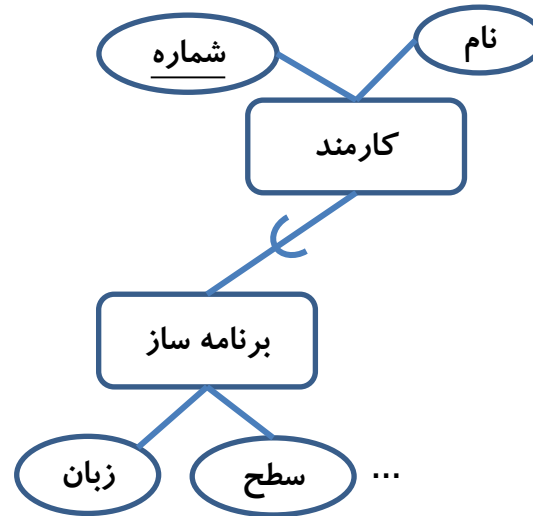
PROF	<u>PRID</u>	PRNAME	RANK	...
	Pr100	...	استاد	...
	Pr200	...	استادیار	...
	Pr300	...	دانشیار	...
	⋮	⋮	⋮	⋮

PUB	<u>PRID</u>	<u>PTITLE</u>	...	PDATE
	Pr100	Data Encryption...	...	...
	Pr100	Semantic Analysis of ...	...	...
	⋮	⋮	⋮	⋮

\* دو صفت PRID (کلید خارجی از جدول PROF) و TITLE، کلید اصلی جدول انتشارات را تشکیل می‌دهند.

حذف و بروزرسانی در جدول PROF چه تاثیری بر PUB باید داشته باشد.





یکی برای زبرنوع موجودیت (حاوی صفات عام یا مشترک) } دو نوع جدول داریم:  ←

یکی برای نوع زیرنوع موجودیت (حاوی صفات خاص زیرنوع و شناسه زبرنوع)



EMP	<u>EID</u>	ENAME	EBDATE	...	EPHONE
	E100	...	...	...	...
	E101	...	...	...	...
	E102	...	...	...	...
	⋮	⋮	⋮	⋮	⋮

PROG	<u><u>EID</u></u>	LANG	...	LEVEL
	E100	C++	...	...
	E102	Java	...	...
	⋮	⋮	⋮	⋮

\* EID (کلید خارجی از جدول EMP) کلید اصلی جدول PROG نیز هست.

حذف و بروزرسانی در جدول EMP چه تاثیری بر PROG باید داشته باشد (و بالعکس)؟



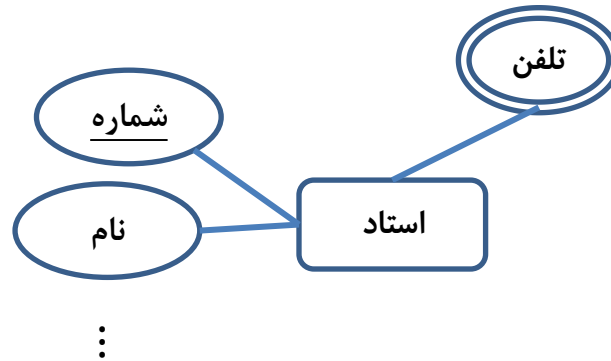


# طراحی منطقی با TDS – صفت چندمقداری

بخش سوم: طراحی منطقی پایگاه داده‌ها

۲۲

صفت چندمقداری



□ دو نوع جدول داریم: ← } یکی برای نوع موجودیت (حاوی صفات تک‌مقداری)  
یکی برای صفت (ساده یا مرکب) چندمقداری



# طراحی منطقی با TDS – صفت چندمقداری (ادامه)

بخش سوم: طراحی منطقی پایگاه داده‌ها

۲۳

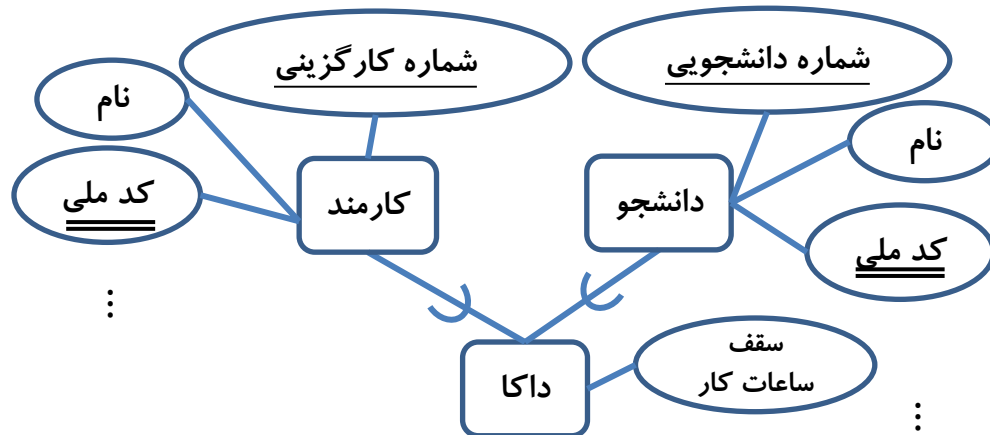
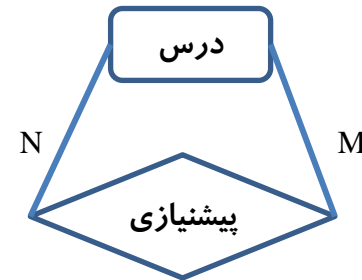
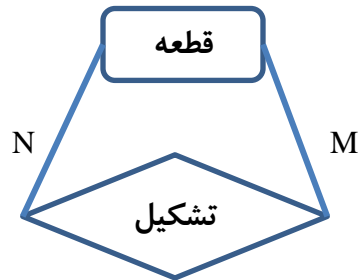
**PROF**

<u>PRID</u>	PNAME	RANK	...	...
Pr100	...	...	...	...
Pr101	...	...	...	...
Pr102	...	...	...	...
⋮	⋮	⋮	⋮	⋮

**PROFTEL**

<u>PRID</u>	<u>TEL</u>
Pr100	09121234567
Pr100	02177889911
Pr101	09352348762
⋮	⋮

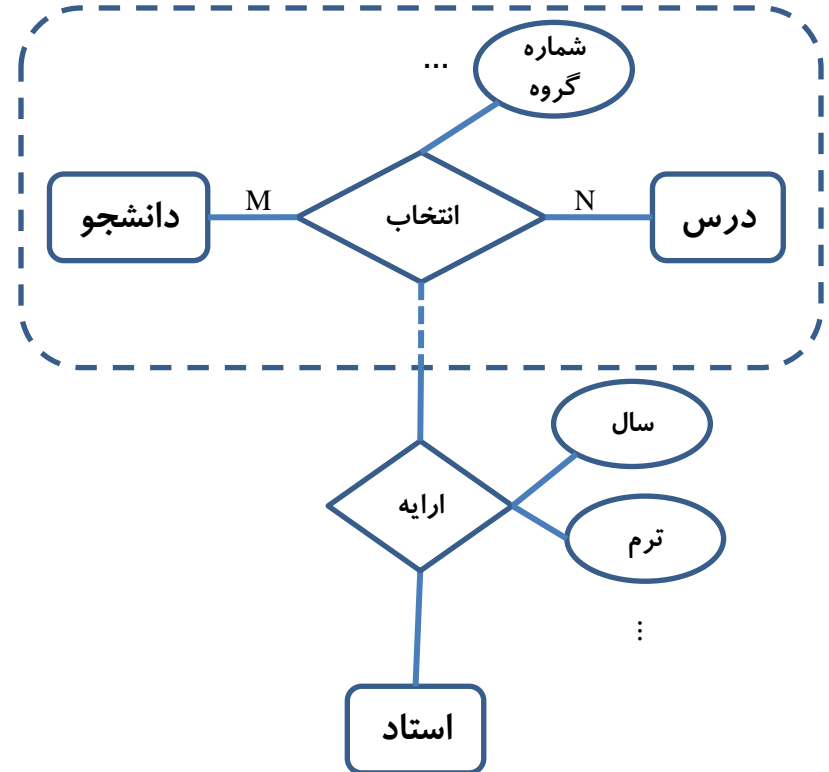
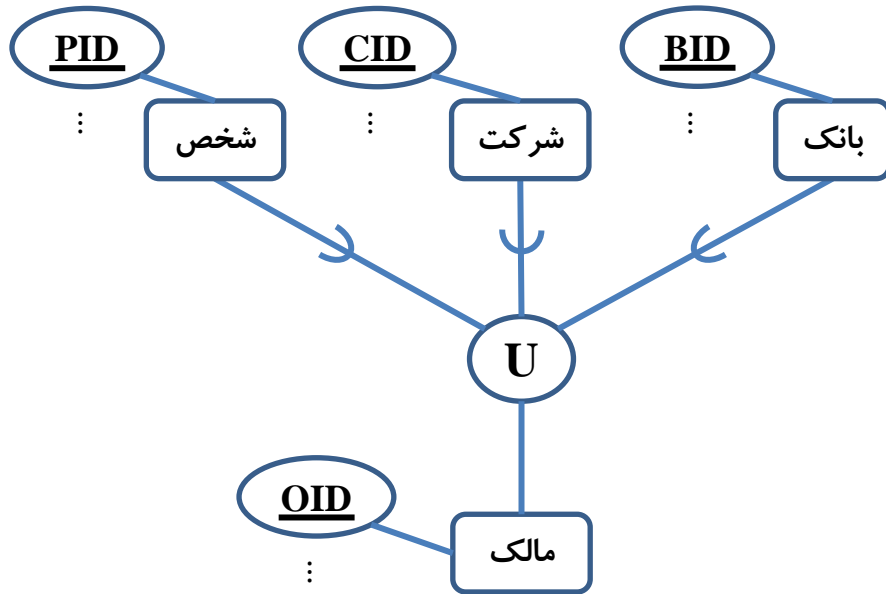
تمرین: TDB را برای مدل‌سازی‌های زیر طراحی کنید. □







تمرین: TDB را برای مدل‌سازی‌های زیر طراحی کنید. □





## پرسش و پاسخ ...

[amini@sharif.edu](mailto:amini@sharif.edu)