

به نام آنکه جان را فکرت آموخت



بخش سوم : طراحی منطقی

دکتر عیسی زارع پور

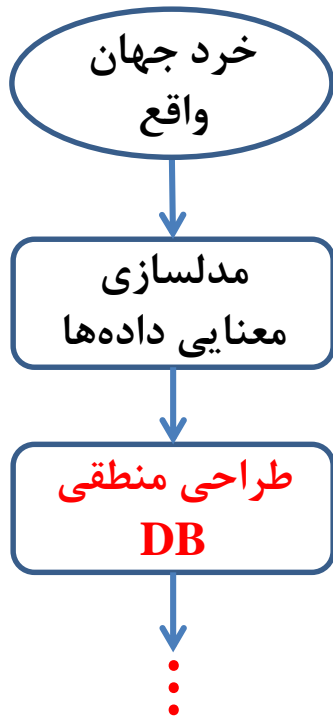
دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت

نیمسال اول ۹۹-۹۸

محتویات اسلایدها برگرفته از یادداشتهای کلاسی **استاد محمدتقی روحانی رانکوهی** است. اسلایدها توسط آقای **دکتر مرتضی امینی** (دانشکده مهندسی کامپیوتر دانشگاه صنعتی شریف) تهیه شده 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.

□ یک ساختار داده برای نمایش نوع موجودیت‌ها و ارتباط بین آنها در سطح منطقی و فیزیکی به کار برده می‌شود.

□ DSها [در حیطه دانش و تکنولوژی DB]:

← پیش رابطه‌ای (پکیج IMS و IDMS)	{	HDBMS	□ ۱- HDS [از Hierarchical DM]
		NDBMS	□ ۲- NDS [از Network DM]
پکیج‌های جدولی	{	RDBMS	□ ۳- RDS [از Relational DM]
		ODBMS	□ ۴- ODS [از Object DM]
		ORDBMS	□ ۵- ORDS [از Object Relational DS]



□ 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.

□ مفاهیم مطرح در طراحی منطقی پایگاه داده‌های رابطه‌ای

□ ساختار داده جدولی : TDS

□ پایگاه داده جدولی : TDB

□ زبان پایگاهی جدولی : TDBL

(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 لحاظ نوع: مجموعه‌ای است از تعدادی **نوع جدول** (که آنها را طراحی می‌کنیم) که به آن شمای پایگاه داده (Schema) هم گفته می‌شود.

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 سطرهای جدولِ مبین 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 سطرهای 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).

اگر کلید اصلی یک جدول در جدول دیگری استفاده شود (مثلا به دلیل وجود ارتباط)، به آن **کلید خارجی** گفته می‌شود.

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



برخی قواعد کلی طراحی مدل رابطه ای با استفاده از مدل ER

□ هر نوع موجودیت تبدیل به یک نوع جدول می شود. صفات نوع موجودیت ستونهای جدول خواهد بود و صفت شناسه نوع موجودیت به عنوان **کلید اولیه** در نظر گرفته می شود.

□ هر نوع موجودیت ضعیف تبدیل به یک نوع جدول می شود که مجموعه صفات نوع موجودیت ضعیف بعلاوه صفت شناسه موجودیت قوی مربوط به آن به عنوان ستونهای جدول خواهد بود و **کلید جدول ترکیب صفت شناسه موجودیت قوی با صفت ممیزه موجودیت ضعیف** خواهد بود.

□ برای صفات چند مقداری می توان یک جدول جدا در نظر گرفت که کلید آن ترکیب کلید جدول اصلی به همراه خود صفت چند مقداری خواهد بود. روشهای دیگری نیز برای طراحی صفت چند مقداری وجود دارد.

نکته: هر گاه کلید یک نوع جدول در نوع جدول دیگر استفاده شود به عنوان کلید خارجی خواهد بود.



برخی قواعد کلی طراحی مدل رابطه ای با استفاده از مدل ER

□ هر نوع ارتباط N به M از هر درجه ای ($n \geq 2$) تبدیل به یک نوع جدول تبدیل می شود که صفات شناسه نوع موجودیتهای شرکت کننده در ارتباط به علاوه صفت/صفات اختصاصی نوع ارتباط به عنوان ستونهای نوع جدول در نظر گرفته می شود. **کلید این جدول ترکیب صفت شناسه نوع موجودیتهای شرکت کننده در ارتباط خواهد بود.**

□ در ارتباطات 1 به N نیازی به ایجاد نوع جدول جدید نیست. بلکه کلید نوع جدول متناظر با نوع موجودیت سمت 1 به همراه صفات اختصاصی ارتباط به مجموعه ستونهای نوع جدول متناظر با نوع موجودیت سمت N اضافه خواهد شد.

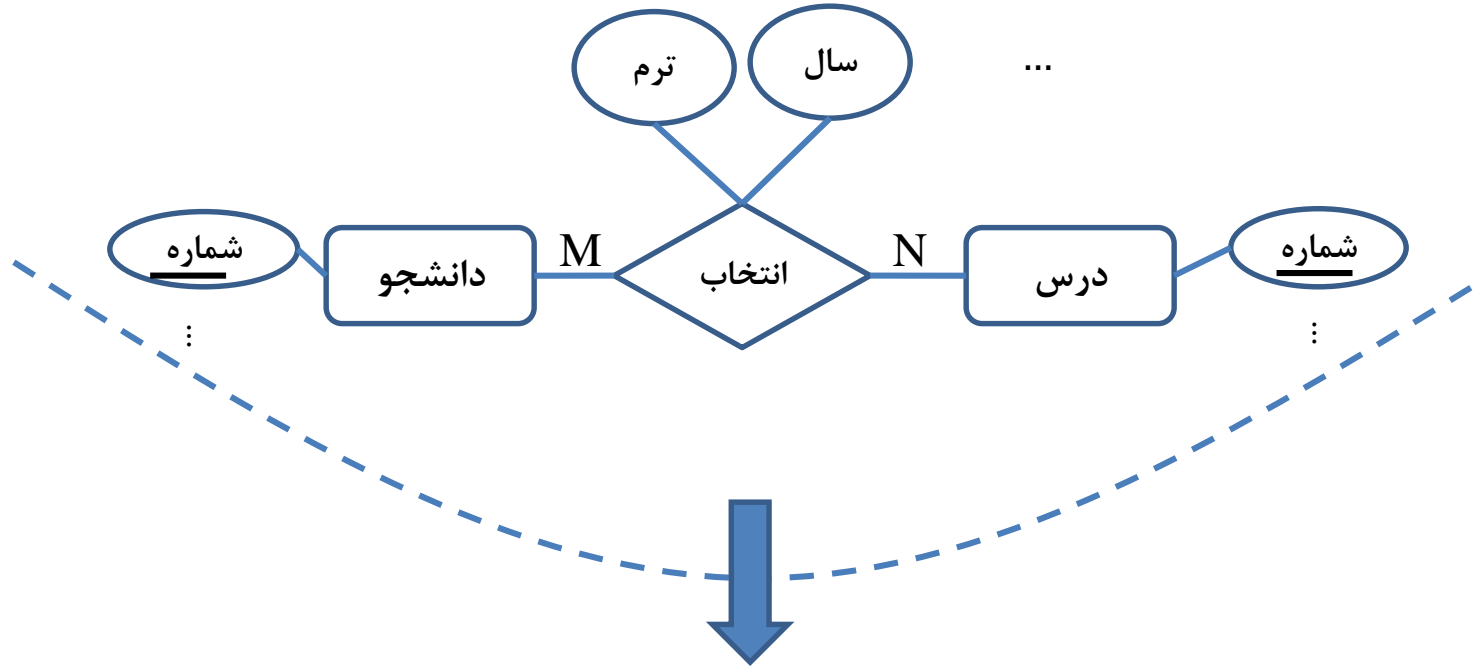
□ نکته: کلید نوع جدول سمت 1 به عنوان **کلید خارجی** در جدول سمت N خواهد بود.

□ در ارتباطات 1 به 1 صفت اختصاصی ارتباط به مجموعه ستونهای یکی از جداول متناظر با نوع موجودیت های مشارکت کننده در ارتباط اضافه می شود.



طراحی منطقی با TDS – رابطه چند به چند

بخش سوم: طراحی منطقی پایگاه داده‌ها



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

۱۱

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

۱۲

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>	Grade
:	:	:	:	
888	co2	1	87	15
888	co3	1	87	17
444	co2	1	87	8
444	co2	1	88	14

STCOT

<u>STID</u>	<u>COID</u>	<u>TR</u>	<u>YR</u>	Grade
:	:	:	:	
888	co2	1	87	15
888	co3	1	87	17
444	co2	1	87	8
444	co2	1	88	14

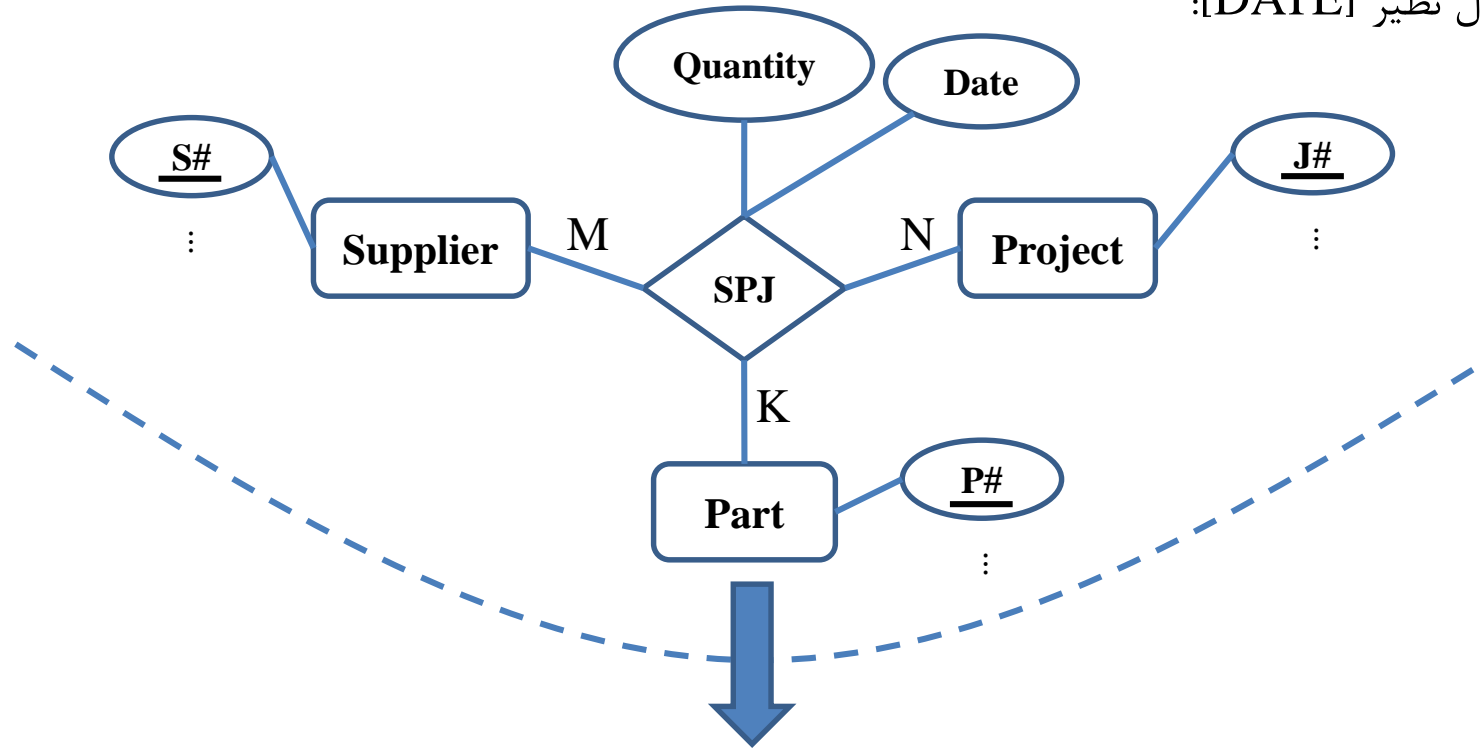


طراحی منطقی با TDS – رابطه چند به چند (ادامه)

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

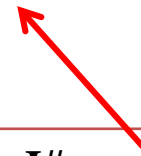
۱۴

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	...
	:	:	:	:

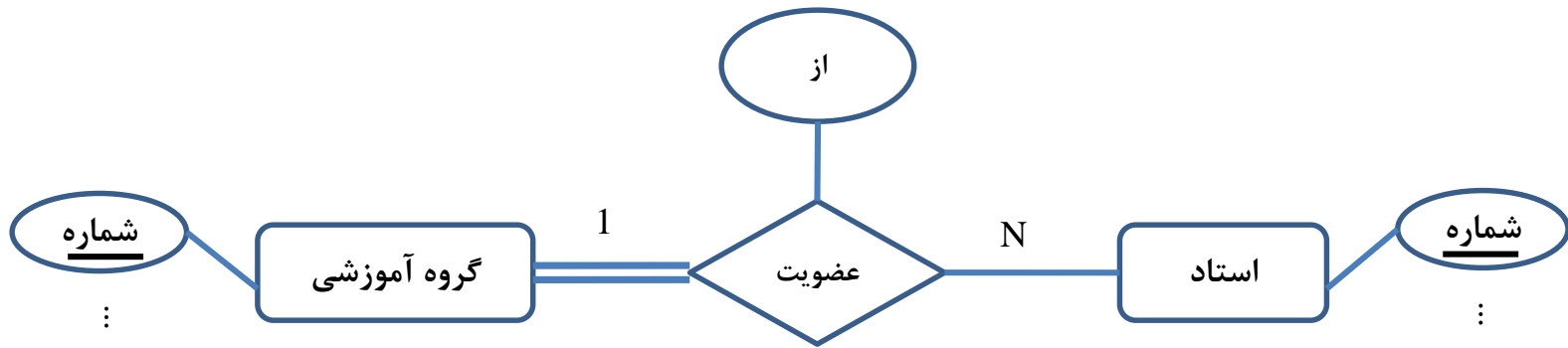


طراحی منطقی با TDS – رابطه یک به چند

بخش سوم: طراحی منطقی پایگاه داده‌ها

۱۵

مثال 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

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

۱۷

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

۱۸

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

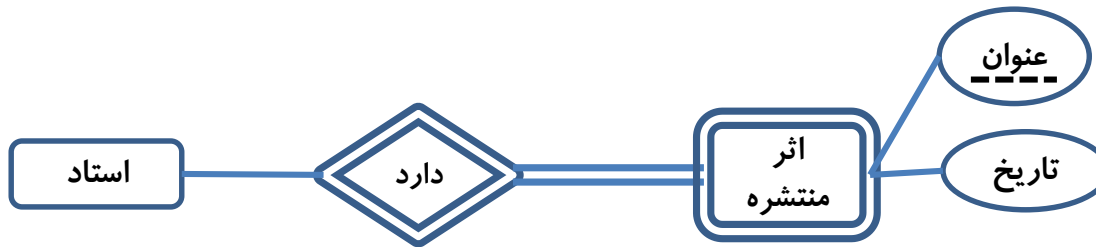
یک طرز طراحی ممکن:

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

طرزهای دیگر طراحی؟



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)



طراحی منطقی با TDS – رابطه شناسا (ادامه)

بخش سوم: طراحی منطقی پایگاه داده‌ها

۲۰

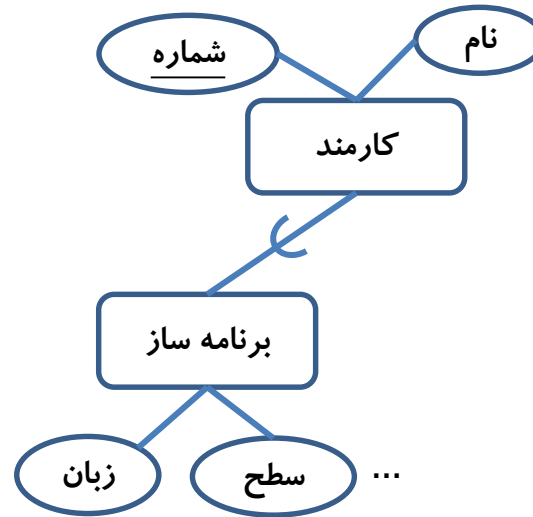
PROF	<u>PRID</u>	PRNAME	RANK	...
	Pr100	...	استاد	...
	Pr200	...	استادیار	...
	Pr300	...	دانشیار	...
	⋮	⋮	⋮	⋮

PUB	<u>PRID</u>	PTITLE	...	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)؟



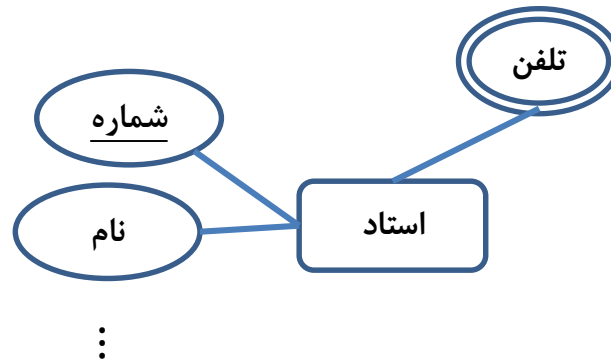


طراحی منطقی با TDS – صفت چندمقداری

بخش سوم: طراحی منطقی پایگاه داده‌ها

۲۳

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

۲۴

PROF

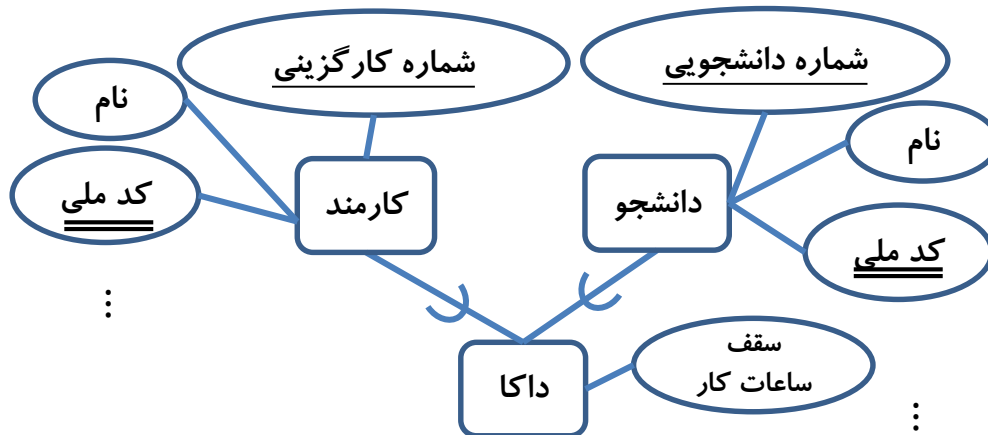
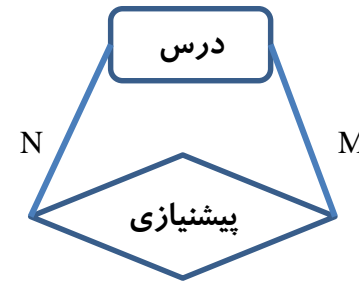
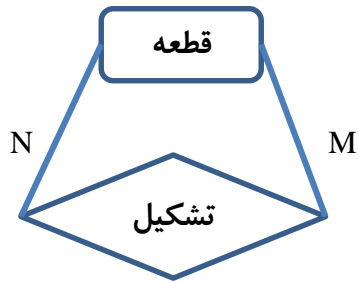
<u>PRID</u>	PNAME	RANK
Pr100
Pr101
Pr102
⋮	⋮	⋮	⋮	⋮

PROFTEL

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

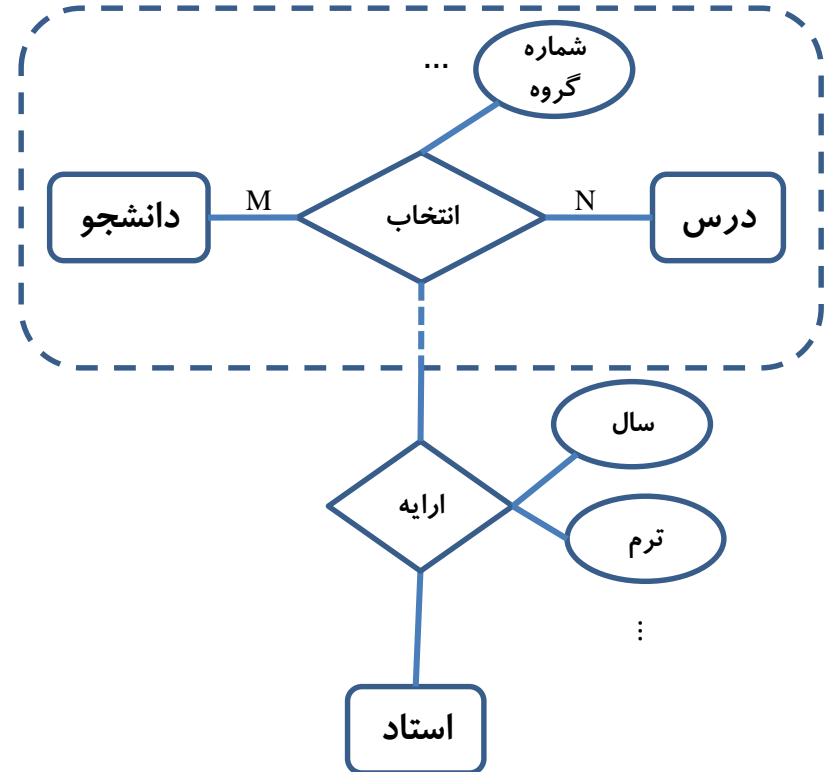
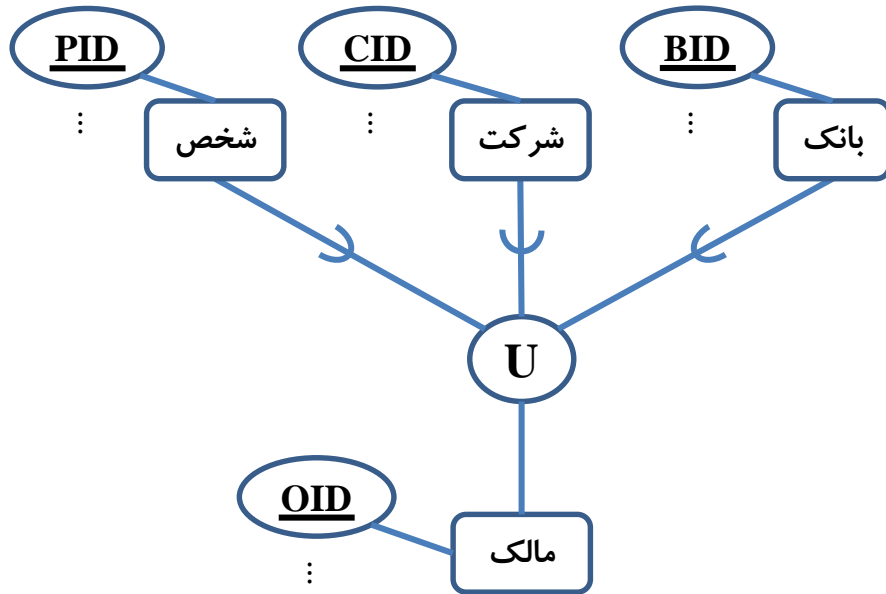


تمرین: TDB را برای مدل‌سازی‌های زیر طراحی کنید. □





تمرین: TDB را برای مدل‌سازی‌های زیر طراحی کنید. □







تمرین سر کلاسی - جدولهای مربوط به این پایگاه داده را طراحی کنید


بخش سوم: طراحی منطقی پایگاه داده‌ها


۲۷

میخواهیم یک سیستم برای مدیریت اعضا و روند امانت کتاب در یک کتابخانه ایجاد کنیم 

- کتابخانه تعدادی عضو دارد.
- اطلاعات شماره عضویت (یکتا)، نام، نام خانوادگی، سطح تحصیلات، آدرس، شماره تلفن، سن، تاریخ عضویت، تاریخ انقضای عضویت و ایمیل اعضا مورد نیاز می باشد.
- کتابخانه تعدادی کتاب در موضوعات مختلف دارد.
- هر کتاب دارای عنوان، نویسنده، ناشر، سال انتشار، شماره ویرایش، موضوع (ات) و شماره شابک می باشد. شابک شماره منحصر به فرد برای هر کتاب است.
- فرض: از هر کتاب تنها یک جلد موجود است.
- اعضا باید هر سال نسبت به تمدید عضویت خود اقدام کنند.
- هر عضو می تواند چند کتاب در مدتی محدود به امانت بگیرد.
- هر عضو می تواند مدت امانت کتاب را تا دو بار تمدید کند.

اگر بخواهیم این سیستم را به صورت آنلاین بین چند کتابخانه استفاده کنیم چه تغییراتی باید در مدل بدهیم؟ 

اگر بخواهیم اطلاعات ناشرین را در سیستم داشته باشیم چه تغییراتی باید در مدل بدهیم؟ 

اگر بخواهیم اطلاعات نویسندگان و موضوعات کتاب ها را در سیستم داشته باشیم چه تغییراتی باید در مدل بدهیم؟ 



پرسش و پاسخ . . .

ایمیل : zarepour@iust.ac.ir

ارتباط حضوری: ساعت مشخص شده در برنامه هفتگی به عنوان رفع

اشکال دانشجویی (روزهای یکشنبه و سه شنبه ساعت ۱:۳۰ تا ۳ عصر)

www.ezarepour.ir