

به نام آنکه جان را فکرت آموخت



بخش سوم :
ساختار داده جدولی،
زبان پایگاهی جدولی
(طراحی منطقی بحث مقدماتی)

مرتضی امینی

نیمسال اول ۹۲-۹۳

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

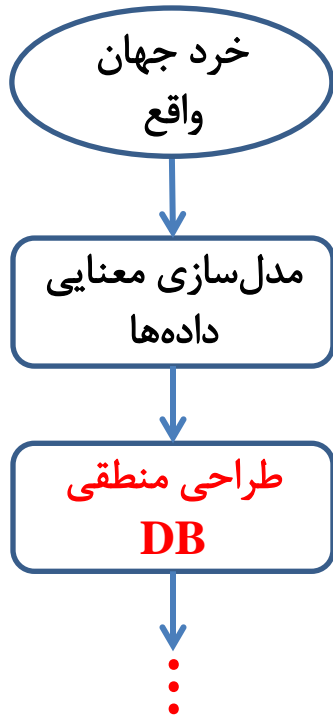


طراحی منطقی DB (بحث مقدماتی)

۲

بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

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



□ طراحی منطقی در سطح انتزاعی انجام می‌شود (ولی بالاترین سطح انتزاع سطحی است که در آن مدل‌سازی داده‌ها انجام می‌شود).

□ **سطح طراحی منطقی:** برای نمایش پایگاه داده‌ها در این سطح از مفاهیمی استفاده می‌شود که مستقل از مفاهیم محیط فایلینگ پایگاه داده‌ها است.



طراحی منطقی DB بحث مقدماتی (ادامه)

بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

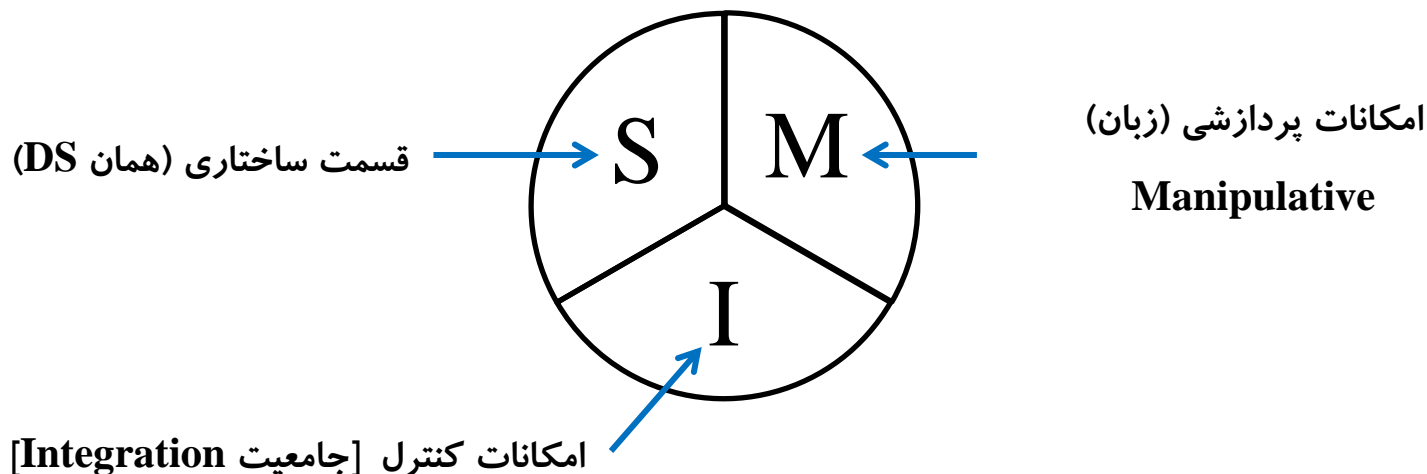
۳

□ 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 مدل داده استاندارد (استاندارد) از سه قسمت تشکیل شده است :








طراحی منطقی DB بحث مقدماتی (ادامه)

بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

۴

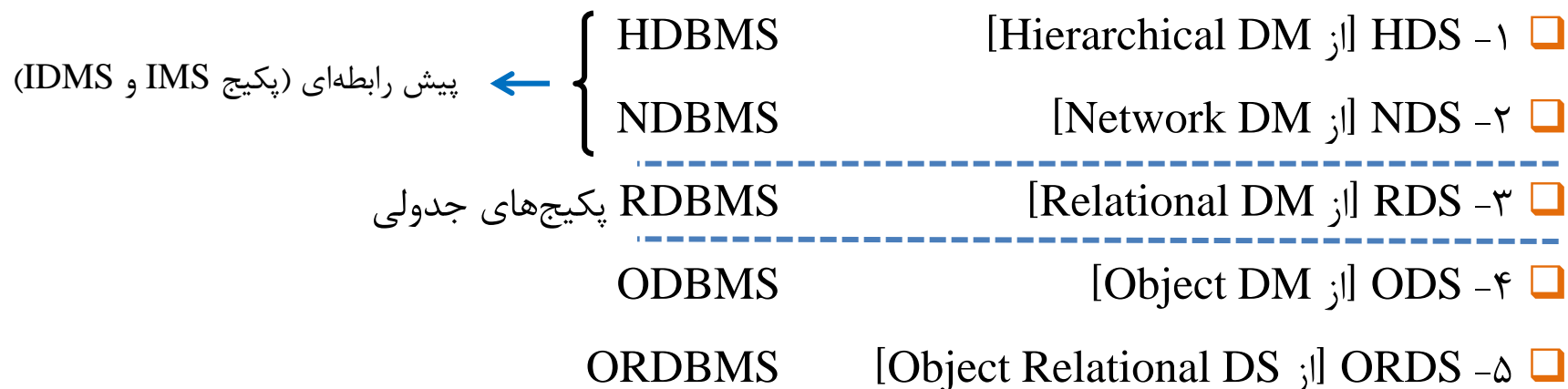
در این قسمت می‌پردازیم به: 

- Tabular Data Structure (TDS) : ساختار داده جدولی 
- Tabular DataBase (TDB) : پایگاه داده جدولی 
- Tabular DataBase Language (TDBL) : زبان پایگاهی جدولی 



بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

□ DS ها [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.



بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

□ چرا DS (در معنای عام)؟

□ برای نمایش داده‌ها و ارتباط بین آنها در سطح
} منطقی
} فیزیکی

□ لزوم DS در حیطه پایگاهی:

۱- تامین کننده محیط فرافایلی (محیط انتزاعی: DS (از DM) ترجیحاً باید مفاهیم ریاضی داشته باشد)

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 تئوری‌ها و تکنیک‌های طراحی DB

۷- ...





ساختار داده جدولی


۷

بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

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 می دهیم.



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

عنصر ساختاری اساسی در HDS و NDS؟





TDB چیست؟ ☐

☐ از دید کاربر $\left\{ \begin{array}{l} \text{طراح} \\ \text{پیاده‌ساز AP} \\ \text{برنامه‌ساز AP} \end{array} \right\}$

☐ از لحاظ نوع: مجموعه‌ای است از تعدادی نوع جدول (که آنها را طراحی می‌کنیم) : A Set of Table Types

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

☐ «نوع سطر» را همان «نوع جدول» مشخص می‌کند.

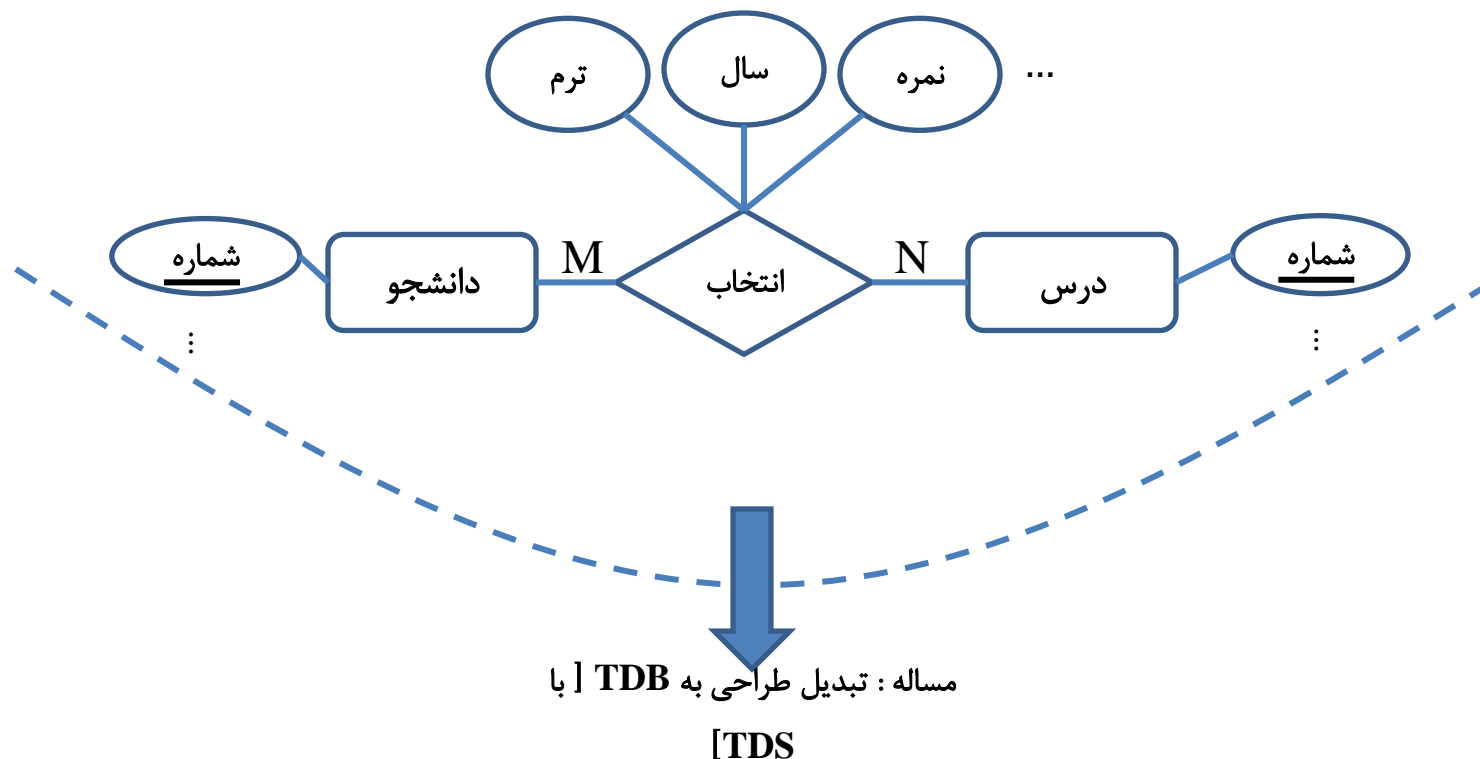


طراحی منطقی با TDS – ارتباط چند به چند

بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

۹

چندی M:N



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



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

بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

۱۰

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 – ارتباط چند به چند (ادامه)

بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

۱۱

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

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

برای نمایش نوع ارتباط M:N انتخاب درس

↓

STCOT

<u>STID</u>	<u>COID</u>	TR	YR . . .	GRADE
:	:	:	:	:
888	co2	1	87	19
888	co3	1	87	10
444	co2	1	87	13

❖ ستون‌های STID و COID در جدول STCOT **کلید خارجی** هستند و با خط چین مشخص می‌شوند.

□ کلید خارجی :

[کاربردی]: ستون c از جدول T1 در جدول T2 کلید خارجی است هرگاه در جدول T1 کلید اصلی باشد.





چرا نه، با کمتر از سه جدول طراحی انجام شود؟



تاثیر نوع مشارکت در طراحی، با چندی $M:N$ ؟



محدودیت الزامی بودن در مشارکت را چگونه باید اعمال یا اعلان کرد؟



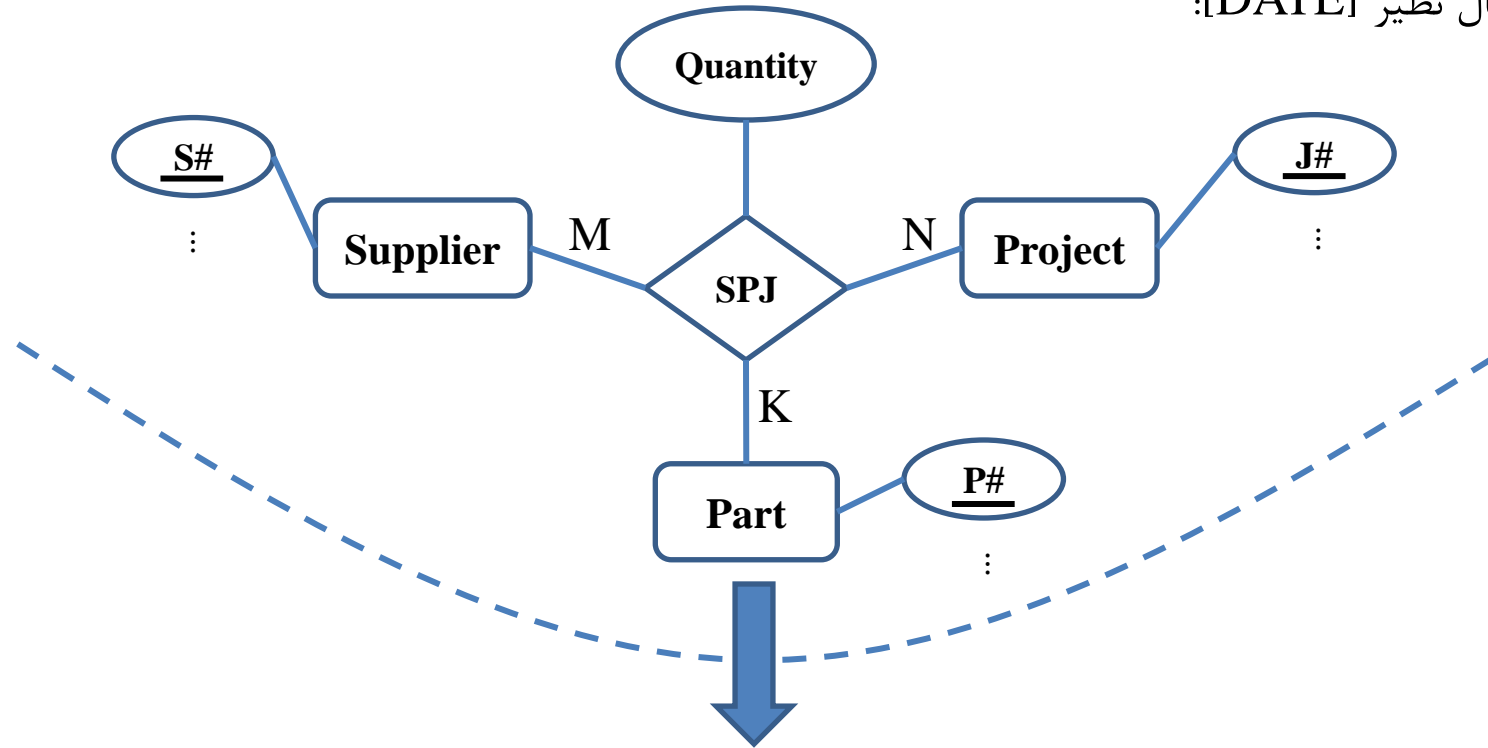


طراحی منطقی با TDS – ارتباط چند به چند (ادامه)

بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

۱۳

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 – ارتباط چند به چند (ادامه)

بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

۱۴

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

SPJ	<u>S#</u>	<u>P#</u>	<u>J#</u>	QTY
	s1	p1	j1	100
	s1	p1	j2	50
	⋮	⋮	⋮	⋮

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

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

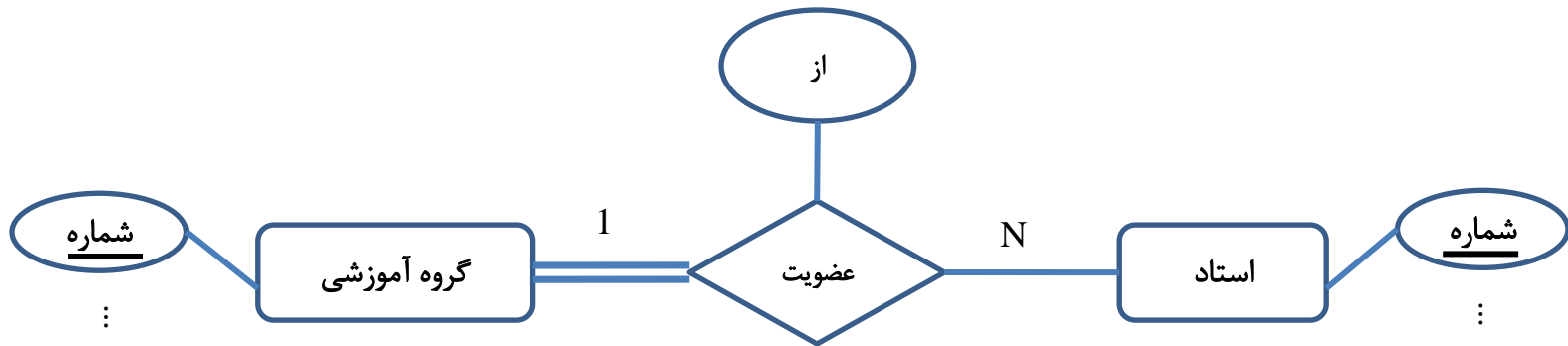


طراحی منطقی با TDS – ارتباط یک به چند

بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

۱۶

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 – ارتباط یک به چند (ادامه)

۱۷

بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

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

کلید خارجی

↑

PROF	<u>PRID</u>	PRNAME	RANK	...	PRPHONE	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 می شود.

اگر مشارکت سمت استاد الزامی باشد، طراحی چگونه خواهد بود؟





طراحی منطقی با TDS – ارتباط یک به یک

بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

۱۸

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



چندی 1:1



□ یک طرز طراحی ممکن :

□ دو نوع جدول داریم: } برای هر نوع موجودیت شرکت کننده در ارتباط یک نوع جدول
جدول نمایشگر نوع موجودیت سمت الزامی، کلید خارجی می گیرد.

در حالت کلی چند طرز طراحی؟





طراحی منطقی با TDS – ارتباط یک به یک (ادامه)

بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

۱۹

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

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

طرزهای دیگر طراحی؟



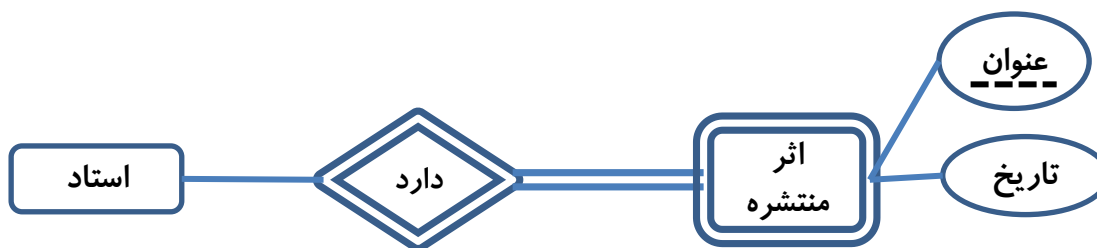


طراحی منطقی با TDS – ارتباط شناسا

۲۰

بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

ارتباط شناسا (ارتباط موجودیت ضعیف)



□ دو نوع جدول داریم: ← یکی برای نوع موجودیت قوی و یکی برای نوع موجودیت ضعیف
جدول نمایشگر نوع ضعیف، کلید خارجی می‌گیرد «جزء کلید».



طراحی منطقی با TDS – ارتباط شناسا (ادامه)

۲۱

بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

PROF	<u>PRID</u>	PRNAME	RANK	...	PRPHONE
	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 PTITLE، کلید اصلی جدول انتشارات را تشکیل می‌دهند.

حذف و بهنگام‌سازی در جدول PROF چه تأثیری باید بر جدول PUB داشته باشد؟



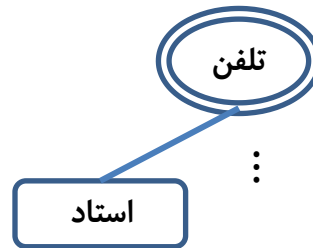
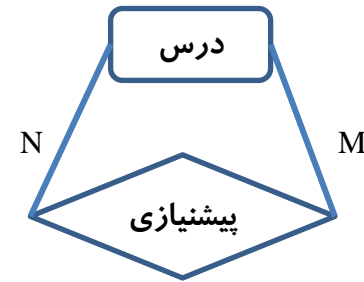
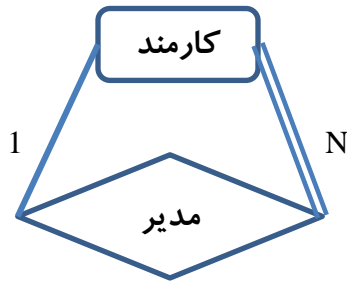


طراحی منطقی با TDS (ادامه)

بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

۲۴

تمرین: TDB را برای مدل‌سازی‌های زیر طراحی کنید. □





بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

□ دستورهای (SQL) Structured Query Language: {
Data Definition Language (DDL)
Data Manipulation Language (DML)
Data Control Language (DCL)

{
CREATE TABLE ایجاد جدول
DROP TABLE حذف جدول
ALTER TABLE تغییر جدول
□ چند دستور از DDL

■ به کمک این دستور ها پایگاه داده جدولی را ایجاد کرده یا تغییراتی در آن اعمال می کنیم.

□ **شِمای پایگاه داده ها** عبارت است از تعریف (توصیف) ساختهای منطقی طراحی شده (جدول های مبنا و ...) و نوعی «برنامه» است شامل تعدادی دستور برای تعریف و کنترل داده ها.



دستور تعریف جدول ☐

CREATE TABLE *TableName*

```
{ ( columnName dataType [NOT NULL | UNIQUE]
[DEFAULT defaultOption][CHECK (searchCondition)][, ...] ) }
[PRIMARY KEY (listOfColumns), ]
{ [UNIQUE (listOfColumns)][, ...] }
{ [FOREIGN KEY (listOfForeignKeyColumns)
REFERENCES ParentTableName [(listOfCandidateKeyColumns)],
[ON UPDATE referentialAction]
[ON DELETE referentialAction]][, ...] }
{ [CHECK (searchCondition)][, ...] }
```

تعریف جدول‌ها: شمای پایگاه جدولی

می‌توان جدول را به صورت موقت نیز (با استفاده از CREATE TEMPORARY TABLE) ایجاد کرد. جدول ☐

موقت حاوی داده‌های ناپایا است و پس از اینکه برنامه کاربر (SQL Session) اجرایش تمام بشود، این جدول توسط سیستم حذف می‌شود.



□ انواع داده‌های قابل استفاده در تعریف ستون‌ها عبارتند از:

□ کاراکتری: CHAR(n), VARCHAR(n)

□ بیتی: BIT [VARYING] (n)

□ عددی: NUMERIC(p, q), REAL, DECIMAL(p, q), INTEGER, SMALLINT,

FLOAT(p), DOUBLE PRECISION

□ زمانی: DATE, TIME, TIMESTAMP, INTERVAL

□

□ در برخی DBMS ها، نوع داده‌های خاصی پشتیبانی می‌شود که امکان ذخیره، بازیابی و پردازش داده‌های

از آن نوع را برای کاربر تسهیل می‌نماید. به طور مثال نوع داده جغرافیایی در PostgreSQL.



شمای پایگاه داده جدولی:



```
CREATE TABLE STT
( STID          CHAR(8) NOT NULL,
  STNAME        CHAR(25) ,
  STLEV         CHAR(12) ,
  STMJR         CHAR(4) ,
  STDEID        CHAR(4)
)
PRIMARY KEY STID ;
CHECK STMJR IN { 'bs', 'ms', 'doc', '???' }
```

```
CREATE TABLE COT
( COID          CHAR(6) NOT NULL,
  COTITLE       CHAR(16) ,
  CREDIT        SMALLINT ,
  COTYPE        CHAR(1) ,
  CODEID        CHAR(4) ,
)
PRIMARY KEY COID ;
```

محدودیت صفتی (ستونی) [کلاز کنترلی]



بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

CREATE TABLE SCT

(STID CHAR(8) NOT NULL ,

COID CHAR(6) NOT NULL ,

TR CHAR(1) ,

YR CHAR(5) ,

GRADE DECIMAL(2 , 2)

)

PRIMARY KEY (STID, COID)

CHECK 0 ≤ GRADE ≤ 20

محدودیت صفتی (ستونی) [کلاز کنترلی]

FOREIGN KEY (STID) REFERENCES STT (STID)

....

....

FOREIGN KEY (COID) REFERENCES COT (COID)

.....

.....



❑ دستور حذف جدول DROP TABLE

DROP TABLE *tablename* [CASCADE|RESTRICT]

❑ **CASCADE** باعث می‌شود که همه «ساختارهای» وابسته به جدول (مانند دیدهای تعریف شده بر

روی آن) نیز به صورت خودکار حذف شود.

❑ **RESTRICT** در صورت وجود دیگر «ساختارهای» وابسته به جدول، از حذف آن جلوگیری می‌کند.

پیش‌فرض این دستور، **RESTRICT** است.

چه «ساختارهایی»؟



با اجرای این دستور چه پیش می‌آید؟



DROP TABLE SCT





□ دستور تغییر جدول ALTER TABLE

□ اضافه کردن ستون، تغییر تعریف ستون، حذف ستون و ...
ALTER TABLE *tableName*

[ADD [COLUMN] *columnName* *dataType* [NOT NULL][UNIQUE]

[DEFAULT *defaultOption*][CHECK (*searchCondition*)]]

[DROP [COLUMN] *columnName* [RESTRICT | CASCADE]]

[ADD [CONSTRAINT [*constraintName*]] *tableConstraintDefinition*

[DROP [CONSTRAINT *constraintName*] [RESTRICT | CASCADE]]

[ALTER [COLUMN] SET DEFAULT *defaultOption*

[ALTER [COLUMN] DROP DEFAULT]

اگر این دستور ADD COL و/یا DROP COL را نداشته باشد، چگونه شبیه سازی می شود؟



ALTER TABLE STT

ADD COLUMN STATE CHAR(10)

اضافه کردن ستون «وضعیت» به جدول اطلاعات دانشجو





بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

و نه دستورات

Data Manipulation (DM)

Data Definition (DD)

Data Controller (DC)

در شمای پایگاهی ← دستورات

این جدایی چه مزایایی دارد؟



در سیستم‌های جدولی، در تعدادی جدول

سیستم با شمای پایگاهی چه می‌کند؟



اطلاعات موجود در آن را **در جایی** به نحوی ذخیره می‌کند.

در جایی

به نحوی

کاتالوگ سیستم

دیکشنری سیستم

مَتا داده‌ها

حاوی متاداده‌ها و داده‌های کنترلی در مورد داده‌های کاربران



بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

مثالی از جدول‌های کاتالوگ:



SysTables

...	تعداد ستون	تاریخ	ایجاد کننده	نام جدول
	5	D1	C1	STT
	5	D2	C1	COT
	5	D2	C2	SCT
	⋮	⋮	⋮	⋮

جدولی که اطلاعات جدول‌های کاربر در آن است.

SysCols

...	طول	نوع	نام جدول	نام ستون
	8	CHAR	STT	STID
	25	CHAR	STT	STNAME
	⋮	⋮	⋮	⋮
	2,2	DEC	SCT	GR

جدولی که اطلاعات ستون‌های کاربر در آن است.



بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)



آیا برنامه ساز می تواند محتوای کاتالوگ را مستقیماً تغییر دهد؟ (با دستورات INSERT, DELETE, UPDATE)

تمرین: حداقل سه جدول دیگر برای کاتالوگ طراحی کنید.

تمرین: چه اطلاعاتی در کاتالوگ ذخیره می شود؟

تمرین: مثال از رفتار سیستم با کاتالوگ :



بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)



❑ CREATE TABLE PROF

(PRID

.....

RANK ...)



فرض :
۶ ستون



PRIMARY KEY PRID ...

INSERT INTO SYSTABLES

VALUES (PROF, C3, D7, 6, ...);



DROP TABLE COT;



DELETE FROM SYSTABLES

WHERE COT = نام جدول;



☐ ALTER TABLE STT
 ADD STDADR CHAR(80) ...



UPDATE TABLE SYSTABLE
 SET = 6 تعداد ستون
 WHERE 'STT' = نام جدول

INSERT INTO SYSCOLS
 VALUES('STDADR', 'STT', CHAR, 80)



□ عملیات در TDB : دستورهای DML

SELECT

بازیابی

INSERT

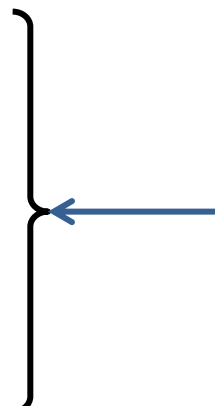
درج

DELETE

حذف

UPDATE

بهنگام سازی





دستور بازیابی SELECT ☐

```
SELECT [ALL | DISTINCT ] item(s) list  
FROM table(s)  
[WHERE condition(s)]  
[ORDER BY Col(s)]  
[GROUP BY Col(s)]  
[HAVING condition(s)]
```

☐ حاصل هر دستور SELECT معتبر، یک جدول است.

☐ از DISTINCT برای حذف سطرهای تکراری در جدول نتیجه استفاده می‌شود.

☐ در Clause WHERE می‌توان از =، <، >، <=، >=، <، >، <=، >=، <، >، <=، >=، <، >، <=، >= استفاده

کرد.



بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)



```
SELECT STT.STID AS SN,  
STT.STNAME AS SNAM  
FROM STT  
WHERE STT.STMJR='phys'  
AND  
STT.STLEV='bs'
```



```
SELECT STT1.STID AS SN,  
STT1.STNAME AS SNAM  
FROM STT AS STT1  
WHERE STT1.STMJR='phys'  
AND  
STT1.STLEV='bs'
```



بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)



یک کپی از جدول با نام جدید، نام‌گذاری جدول جواب:

(SELECT S.*

FROM S) AS MyS

■ مرتب شده:

ORDER BY SNAME یا 2

■ پیشفرض صعودی: (Ascending)



شماره ستون

■ نزولی (Descending): باید قید شود.

□ تمرین: روش دیگر؟



جنبه‌های پیشرفته (Advanced features):

SELECT S#, CITY

FROM S

WHERE SNAME

$\left\{ \begin{array}{l} \text{LIKE} \\ \text{NOT LIKE} \end{array} \right\}$

'%N' → با N تمام شود

'M%' → با M شروع شود

'_ _ A _ _' → دقیقاً ۵ کاراکتر، کاراکتر سوم A



بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

SELECT P#

FROM P

WHERE WEIGHT BETWEEN (5,15)

یا

WHERE WEIGHT >=5 AND WEIGHT <=15

BETWEEN



□ شماره قطعاتی را بدهید که وزن آنها بین ۵ و ۱۵ است.



بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)



SELECT S#, CITY

FROM S

WHERE STATUS

$\left\{ \begin{array}{l} \text{IS NULL} \\ \text{IS NOT NULL} \end{array} \right\}$

بررسی برخورد یک package با NULL؟



NULL

?
≤
>
≥
=

?

NULL

یا

عدد

?

$\frac{VALUE}{NULL}$

?

NULL AND NULL



عملگرهای جبر مجموعه‌ها (دیدگاه کاربردی)

بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

۴۵

$tablename1$ op $tablename2$ [ALL] [CORRESPONDING [BY {column, [, column ...]}]]

$op \in \left\{ \begin{array}{l} \text{UNION} \\ \text{UNION ALL} \\ \text{INTERSECT} \\ \text{EXCEPT} \end{array} \right\}$

☐ شرط استفاده: برابری سرآیند: تعداد و هم‌نامی و هم‌نوعی ستون‌های (دو جدول) باید نوع-سازگار باشند

☐ اگر از گزینه CORRESPONDING BY استفاده شود، عمل درخواست شده روی ستون‌های تصریح شده انجام می‌شود.

☐ اگر CORRESPONDING بدون BY استفاده شود، عمل درخواست شده روی ستون‌های مشترک انجام می‌شود. (عمل بر روی زیرجدول نوع-سازگار انجام می‌شود)

☐ اگر از این گزینه استفاده نشود، عمل روی تمام ستون‌های دو جدول انجام می‌شود.

☐ **توجه:** حذف تکراری‌ها در نتیجه اجرای عملگرهای جبر مجموعه‌ها



بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

```
SELECT S.S#,  
FROM S  
INTERSECT
```

شماره تهیه‌کنندگانی را بدهید که حداقل یک قطعه تولید می‌کنند.



```
SELECT SP.S#,  
FROM SP
```

```
SELECT SP.S#,  
FROM SP  
EXCEPT
```

تست سازگاری پایگاه داده‌ها:



```
SELECT S.S#,  
FROM S
```

مدل دیگر




SP *EXCEPT* S Using S# یا Corresponding by S#



شماره تهیه‌کنندگانی را بدهید که هیچ قطعه‌ای تولید نمی‌کنند.



```
SELECT S.S#,  
FROM S  
EXCEPT  
SELECT SP.S#,  
FROM SP
```

تمرین: این مثال‌ها به طرز دیگر هم نوشته شود. 



Aggregation Functions ☐

AVG ☐

MIN ☐

MAX ☐

SUM ☐

COUNT(*) و COUNT ☐

بیشینه وضعیت تهیه کنندگان در شهرهای c1 یا c2



```
SELECT MAX ( STATUS ) AS SMAX
FROM S
WHERE CITY='c1'
OR
CITY='c2'
```



بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)



تعداد انواع قطعات تولیدی توسط تولیدکنندگان

```
SELECT COUNT (DISTINCT P#) AS N1  
FROM SP
```



تعداد انواع قطعات قابل تولید

```
SELECT COUNT (*) AS N2  
FROM P
```



تعداد کل قطعات تولیدی توسط s2

```
SELECT SUM (QTY) AS N3  
FROM SP  
WHERE S# = 's2'
```



NULL و توابع جمعی؟ (در سه پکیج بررسی شود)



GROUP BY

سطرهای جدول داده شده در کلاز FROM را گروه بندی می کند، به نحوی که مقدار ستون(های) گروه بندی در گروه یکسان است.

تعداد کل قطعات تولیدی توسط هر تولیدکننده



```
SELECT S# AS SN ,SUM (QTY) AS SQ  
FROM SP  
GROUP BY S#
```

جدول جواب

SN	SQ
s1	280
s2	100
s3	203
...	...

SP
گروه بندی
شده



S#	P#	QTY
s1	p1	100
s1	p2	50
s1	p4	130
s2	p2	30
s2	p3	70
s3	p5	203
...




بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)



در کلاز SELECT نمی توان نام ستونی را آورد که در کلاز GROUP BY نباشد، غیر از ستون‌هایی که با توابع جمعی به دست آمده‌اند.

HAVING

 امکانی است برای دادن شرط یا شرایط ناظر به گروه سطرها



شماره تهیه‌کنندگانی را بدهید که بیش از ۱۰۰ قطعه تولید کرده‌اند.

SELECT S#

FROM SP

GROUP BY S#

HAVING SUM (QTY) > 100



بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

تمرین : شماره دانشجویانی را بدهید که در ترم دوم سال ۸۷-۸۸ بیش از ۲۰ واحد گرفته باشند.

تمرین : شماره دانشجویانی را بدهید که در ترم دوم سال ۸۷-۸۸ بیش از ۷ درس گرفته باشند.

GROUP BY و HAVING در SQL افزونه‌اند، اما نوشتن QUERY بدون آنها پیچیده است.



HAVING بدون GROUP BY؟



به چند روش می‌توان یک کپی از جدول ساخت؟





بازیابی از بیش از یک جدول

۵۳

بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

روش اول

SELECT SNAME

FROM S, SP

WHERE SP.S# = S.S# AND SP.P# = 'p2'

قطعه 'p2' را بدهید:

تهیه کنندگان

نام



در جدول SP

در جدول S

شبیه سازی عملگر پیوند

عملگر پیوند چیست؟

SELECT T1.*, T2.*

FROM T1, T2

ضرب دکارتی در SQL



مکانیزم اجرا از دید برنامه ساز:

- به ازای هر سطر جدول S، بررسی می کند که آیا S# آن در SP وجود دارد یا نه و P# آن سطر در SP، p2 است یا نه. اگر درست بود SNAME آن سطر جزو جواب است.



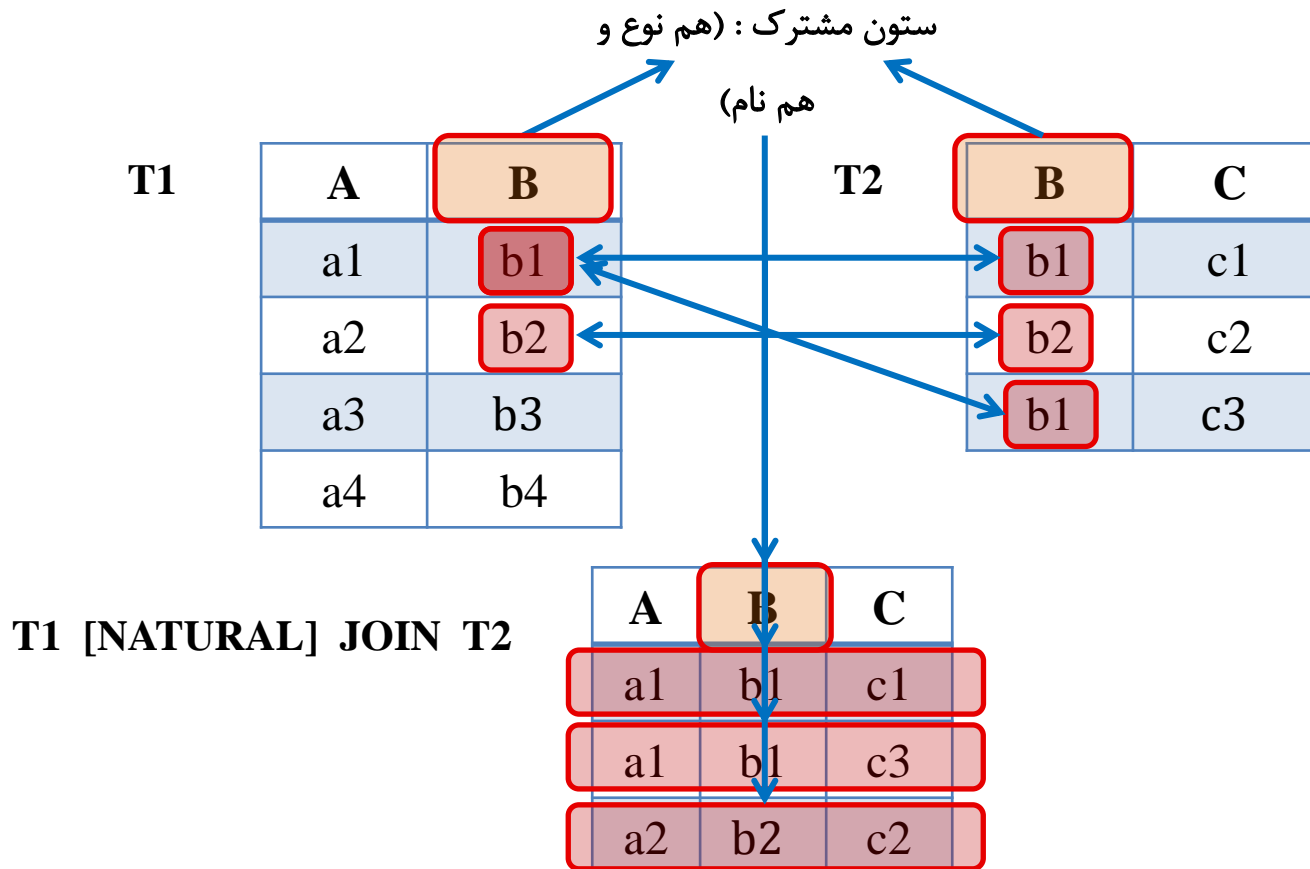
بازیابی از بیش از یک جدول – عملگر پیوند یا JOIN

بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

۵۴

□ پیوند: ارائه مقدماتی (غیر ریاضی)

□ T1 [NATURAL] JOIN T2





بازیابی از بیش از یک جدول – عملگر پیوند یا JOIN (ادامه)

بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

۵۵

توضیح مقدماتی عملگر پیوند:

صرف نظر از جزئیات تئوریک، سطرهای دو جدول را که مقدار ستون(های) مشترکشان یکسان است،

به هم پیوند می‌زند.

روش دوم

SELECT SNAME

FROM S [NATURAL] JOIN SP

WHERE P# = 'p2'

نام تهیه‌کنندگان قطعه 'p2' را بدهید:



S

S#	SNAME	...
s1	sn1	...
s2	sn2	...
s3	sn3	...
s3	sn4	...
...

SP

S#	P#	QTY
s1	p1	100
s1	p2	120
s1	p3	500
s2	p1	50
...

S [NATURAL] JOIN SP

S#	SNAME	...	P#	QTY
s1	sn1	...	p1	100
s1	sn1	...	p2	120
s1	sn1	...	p3	500
s2	sn2	...	p1	50
...



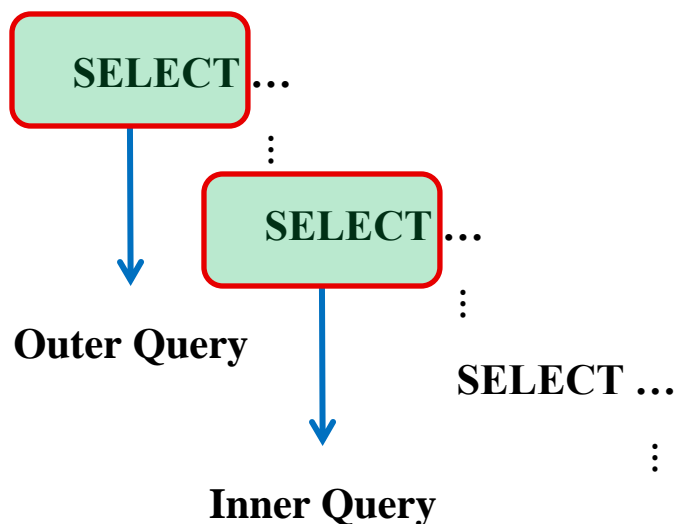
بازیابی از بیش از یک جدول – زیرپرسش

بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

۵۶

زیر پرسش یا SubQuery

یک SELECT است درون SELECT دیگر. ← پرسش تو در تو





بازیابی از بیش از یک جدول – عملگر تعلق

بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

۵۷

IN و NOT IN : عملگر تعلق □



روش سوم

SELECT SNAME

FROM S

WHERE S# IN (SELECT S# FROM SP

WHERE P# = 'p2')

روش چهارم

یا
= ANY

روش پنجم

یا
= SOME

عملگر تعلق

□ مکانیزم اجرا:


- سیستم ابتدا SELECT درونی را اجرا می کند، آنگاه به ازای هر سطر S بررسی می کند که S# در مجموعه جواب SELECT درونی هست یا نه.




بازیابی از بیش از یک جدول – پرسش های بهم بسته

بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

۵۸

 دو پرسش درونی و بیرونی (در یک پرسش تو در تو) را **بهم بسته (Correlated)** گوئیم هر گار در کلاز WHERE پرسش درونی به ستونی از جدول موجود در کلاز FROM پرسش بیرونی، ارجاع داشته باشیم.

 **توجه:** نحوه اجرای پرسش های بهم بسته با طرز اجرای پرسش های نابهم بسته متفاوت است: در حالت بهم بسته، سیستم پرسش درونی را به ازای هر سطر از جدول پرسش بیرونی یک بار اجرا می کند.

روش ششم

SELECT SNAME

FROM S

WHERE 'p2' IN (SELECT P# FROM SP
WHERE SP.S# = S.S#)

روش هفتم

یا

= ANY

روش هشتم

یا

= SOME

CORRELATED یا زیرپرسش بهم بسته





بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

$$\text{theta} \in \left\{ \begin{array}{c} = \\ \neq \\ < \\ \leq \\ \geq \\ > \end{array} \right\} \quad \left\{ \begin{array}{cc} \text{theta} & \text{ANY} \\ \text{theta} & \text{SOME} \\ \text{theta} & \text{ALL} \end{array} \right\} \quad \square \text{ امکان}$$

شماره تهیه کنندگانی را بدهید که مقدار وضعیت آنها بیشینه نباشد.



1- SELECT S#

FROM S

WHERE STATUS < ANY (SELECT DISTINCT STATUS FROM S)

2- SELECT S#

FROM S

WHERE STATUS < (SELECT MAX (STATUS) FROM S)

چون جواب SELECT تک مقداری است نیازی به ANY نیست.



بازیابی از بیش از یک جدول (ادامه)

بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

۶۰



روش نهم

```
SELECT SNAME
FROM S
WHERE 0 < ( SELECT COUNT(*)
            FROM SP
            WHERE SP.S# = S.S#
            AND
            SP.P# = 'p2' )
```




NOT EXISTS و EXISTS ☐

☐ امکان بررسی وجود یا عدم وجود سطر در جدول بازگشتی

روش دهم

```
SELECT SNAME  
FROM S  
WHERE
```

```
EXISTS ( SELECT *  
          FROM SP  
          WHERE SP.S# = S.S#  
                AND  
                SP.P# = 'p2' )
```

«وف» یا «WFF»

روش‌های دیگر؟





دستورهای INSERT, UPDATE, DELETE ☐

درج :INSERT ☐

INSERT INTO *table-name*
VALUES (*one row*) | *subquery*

به‌نگام‌سازی :UPDATE ☐

UPDATE *table-name*
SET *col = value / scalar ...*
:
WHERE *condition(s) / subquery*

حذف :DELETE ☐

DELETE FROM *table-name*
WHERE *condition(s) / subquery*



درج سطری (سطر کامل - سطر ناقص):



INSERT INTO STT

VALUES { ('222' , 'st2' , 'IT' , 'bs' , 'D17')
('333' , 'st3' , Null , 'ms' , Null) }

درج گروهی:



CREATE TEMPORARY TABLE T1

(STN,)

اطلاعات دانشجویان مقطع کارشناسی ارشد

رشته کامپیوتر در جدول موقت T1 درج شود.

INSERT INTO T1

(**SELECT STT.***

FROM STT

WHERE STJ = 'comp'

AND

STL = 'ms')



بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

بهنگام سازی چند سطر:



تعداد واحد تمام درس های عملی گروه آموزشی D11 را برابر یک کن.

```
UPDATE COT
SET CREDIT = '1'
WHERE COTYPE = 'p' AND CODEID = 'D11'
```

بهنگام سازی در بیش از یک جدول:



```
UPDATE STT
SET STID = 88104444
WHERE STID = 88107777

UPDATE STCOT
SET STID = 88104444
WHERE STID = 88107777
```

اگر دستور دوم اجرا نشود؟





بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

نمره دانشجویان گروه آموزشی D111 در درس 'com222' در ترم دوم سال ۸۵-۸۶ را ناتمام



اعلان کن.

UPDATE STCOT

SET STCOT.GRADE = 'U'

WHERE STCOT.TR = '2' AND STCOT.YRYR = '85-86'

AND STCOT.COID = 'COM222'

AND STID IN (SELECT STID

FROM STT

WHERE STT.STDEID = 'D111');



بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)



حذف تکدرس: درس com111 را برای دانشجوی 88104444 حذف کنید.

DELETE FROM STCOT

WHERE STID = 88104444

AND

COID = 'COM111'

آیا این حذف باید انتشار یابد؟



حذف از بیش از یک جدول:

DELETE FROM DEPT

WHERE DEID = 'D333'

UPDATE STT

SET DEID = 'Null'

WHERE DEID = 'D333'



بخش سوم: ساختار داده جدولی، زبان پایگاهی جدولی (طراحی منطقی بحث مقدماتی)

☐ مطالعه شود :

☐ پرسش بازگشتی

☐ SQL ادغام شده

☐ SQL پویا

☐ نوشتن رویه

☐ نوشتن تابع

☐ امکانات شیء- رابطه‌ای

☐ مدیریت تراکنش



پرسش و پاسخ ...

amini@sharif.edu