به نام آنکه جان را فکرت آموخت

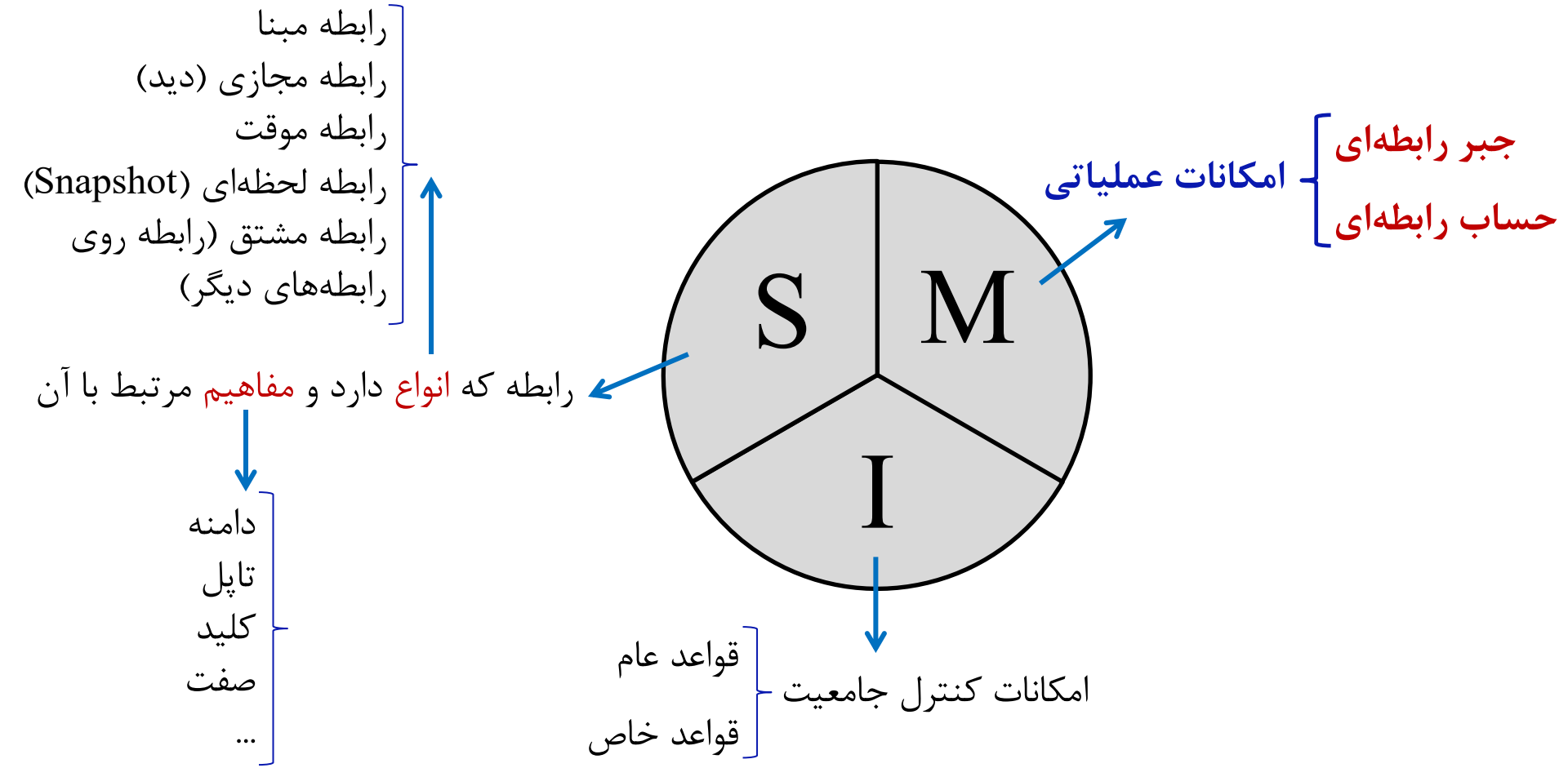
بخش نهم:
عملیات در پایگاه داده رابطه‌ای

مرتضی امینی

نیمسال اول ۱۴۰۲-۱۴۰۱

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

یادآوری: مدل داده‌ای



- عملگرها
  - عملگرهای متعارف
    - عملگرهای دو عملوندی $R_1\ op\ R_2$ $op \in \{\cup, \cap, , -, \times\}$
      - اجتماع - UNION
      - اشتراک - INTERSECT
      - تفاضل - MINUS
      - ضرب کارتزین - TIMES
  - عملگرهای خاص
    - گزینش یا تحدید - RESTRICT
    - پرتو یا تصویر - PROJECT
    - پیوند یا الصاق - JOIN

عملگرهای متعارف جبر رابطه‌ای

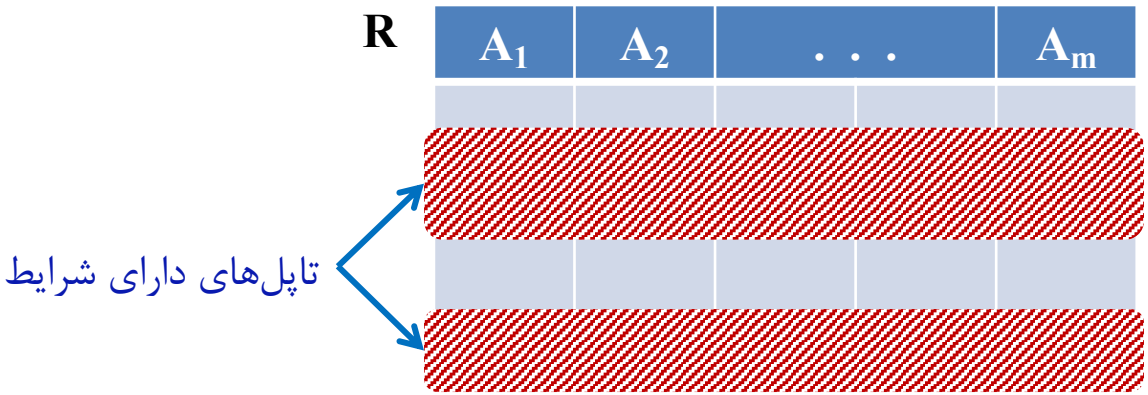
- **خاصیت بسته بودن:** حاصل ارزیابی هر عبارت جبر رابطه‌ای معتبر، باز هم یک رابطه است (که تاپل تکراری ندارد).
- برای **سه عملگر** $\cup$ ، $\cap$ **و** $-$ ، باید عملوندها نوع-سازگار (Type Compatible) باشند:
  - **پیش شرط:** $H_{R_1} = H_{R_2}$
  - $R_3 = R_1 \; op \; R_2 \longrightarrow H_{R_3} = H_{R_1} = H_{R_2} \qquad op \in \{\cup, \cap, , -\}$
    - بدنه نتیجه، حاصل انجام هر یک از اَعمال اجتماع، اشتراک و یا تفاضل دو مجموعه بدنه است.
- **در عملگر ضرب کارتزین (TIMES):**
  - **شرط:** در عنوان دو رابطه نباید صفت هم‌نام وجود داشته باشد. $H_{R_2} \cap H_{R_1} = \varnothing$
  - عنوان رابطه نتیجه برابر است با $H_{R_2} \cup H_{R_1}$ و بدنه نتیجه برابر ضرب کارتزین دو مجموعه بدنه است.
  - TIMES در SQL چگونه شبیه‌سازی می‌شود؟

عملگر گزینش

- **عملگر گزینش یا تحدید – RESTRICT**
- **نماد ریاضی:** σ_c ← شرط یا شرایط گزینش

یک عبارت بولی (منطقی) تشکیل شده از شرطهای ساده به صورت (A_i *theta* A_j) یا (A_i *theta* literal) که در آن *theta* یکی از عملگرهای $=$، $\neq$، $<$، $>$، $\leq$ و $\geq$ است و literal یک مقدار ثابت است.

- **شکل کلی:** $\sigma_c(R)$ **یا** R **WHERE** c **یا** **RESTRICT** R **WHERE** c
- تک عملوندی: Monadic
- **عملکرد** (در نمایش جدولی رابطه): زیرمجموعه‌ای افقی می‌دهد. ← عملگر تاپل(ها)یاب



عملگر گزینش (ادامه)

مشخصات کامل دانشجویان رشته فیزیک دوره کارشناسی را بدهید.

$$\sigma_{STJ='phys' \wedge STL='bs'}(STT)$$

```
SELECT STT.*
        FROM  STT
        WHERE  STJ='phys'  AND  STL='bs'
```

وقتی در شرط C (یا کلاز WHERE) بخشی از کلید را با شرط تساوی داده باشیم.

اگر $R' = \sigma_c(R)$ باشد آنگاه $CK_{R'} \subseteq CK_R$.

عملگر گزینش (ادامه)

- عملگر گزینش جابجایی‌پذیر است، یعنی:

$$\sigma_{c1}(\sigma_{c2}(R)) = \sigma_{c2}(\sigma_{c1}(R)) = \sigma_{c1 \wedge c2}(R)$$

- **عبارتهای جبری معادل:**

- R **WHERE** (C_1 **AND** C_2) ≡ (R **WHERE** C_1) **INTERSECT** (R **WHERE** C_2)
- R **WHERE** (C_1 **OR** C_2) ≡ (R **WHERE** C_1) **UNION** (R **WHERE** C_2)
- R **WHERE NOT** C ≡ R **MINUS** (R **WHERE** C)

عملگر پرتو

عملگر پرتو – PROJECT

- نماد ریاضی: **Π**
- شکل کلی: $\Pi_{\langle L \rangle}(R)$ یا $(R)[L]$ یا **PROJECT** R **OVER** (L)
  - $\langle L \rangle$ ← لیست صفات پرتو
- تک عملوندی: Monodic
- **عملکرد** (در نمایش جدولی رابطه): زیرمجموعه عمودی می‌دهد. ← عملگر ستون(ها)یاب

R

A_1	. . .	A_i	. . .	A_j	. . .	A_m

عملگر پرتو (ادامه)

- عملگر پرتو تکراری‌ها را حذف می‌کند. ⟵ چون جواب رابطه است، پس یک مجموعه است و عضو تکراری ندارد.

شماره و رشته تمام دانشجویان را بدهید.

$$\Pi_{\langle STID,STJ\rangle}(STT)$$

SELECT STID, STJ **FROM** STT

شماره دانشجویانی که درسی انتخاب نکرده‌اند.

$$R := \Pi_{\langle STID\rangle}(STT) - \Pi_{\langle STID\rangle}(STCOT)$$

شماره و مقطع تحصیلی دانشجویان رشته IT را بدهید.

$$\Pi_{\langle STID,STL\rangle}\left(\sigma_{STJ='IT'}(STT)\right)$$

عملگر پرتو (ادامه)

- اگر $R' = \Pi_{\langle L \rangle}(R)$ باشد آنگاه:
  - اگر $CK_R \subseteq L$ آنگاه $CK_{R'} = CK_R$،
  - اگر نه در بدترین حالت $CK_{R'} = L$.

کجای؟ اگر $R' = R_1 \; op \; R_2$ و $op \in \{\cup, \cap, -, \times\}$، آنگاه $CK_{R'} = ?$.

- SELECT در SQL استاندارد، در حالت کلی ترکیبی از دو عملگر RESTRICT و PROJECT است.

عملگر پرتو گسترش یافته

عملگر پرتو گسترش یافته – EXTENDED PROJECT

- نماد ریاضی: Π
- شکل کلی: $\Pi_{\langle F1,F2,\ldots,Fn\rangle}(R)$

  ← لیست صفات و یا توابع حسابی پرتو

- این عملگر امکان می‌دهد تا در لیست صفات پرتو، از توابع حسابی استفاده شود و صفت (صفاتی) با مقادیر حاصل از اجرای تابع (توابع) در رابطه جواب داشت.

مثال

رابطه‌ای با صفات شماره دانشجو، شماره درس و نمره دانشجو در درس، تغییریافته با فرمول G:=1.2*GRADE بدهید.

$$\Pi_{\langle STID,\ COID,\ (1.2*GRADE)\ RENAME\ AS\ G\rangle}(STCOT)$$

عملگر تغییر نام

عملگر تغییر نام – RENAME

- نماد ریاضی: ρ
- شکل کلی: $\rho_R(E)$

  R: نام رابطه حاصل از عبارت جبر رابطه‌ای E

- این عملگر برای نامیدن رابطه حاصل از یک عبارت جبر رابطه‌ای به کار می‌رود.
- **عملکرد:** $\rho_R(E)$ رابطه حاصل از عبارت جبر رابطه‌ای E را با نام R برمی‌گرداند.
- از عملگر RENAME برای دگرنامی صفت هم می‌توان استفاده کرد (مشابه آنچه در مثال اسلاید قبل آمد). مثلاً با دستور R **RENAME** A_i **AS** B_j، به صفت A_i از R، نام دیگر B_j داده می‌شود.

عملگر پیوند

- **عملگر پیوند JOIN** (مدل ریاضی عمومی)
  - **نام عمومی:** Theta Join
  - **نماد ریاضی:** $\bowtie_{Cond(s)}$ ← شرط پیوند
  - **فرض:** دو رابطه R_1 و R_2 نام صفت مشترک ندارند.
    - $R_1 (A_1, A_2, \ldots, A_n)$
    - $R_2 (B_1, B_2, \ldots, B_m)$
  - **شکل کلی:** $\mathbf{R_1 \bowtie_C R_2}$ یا $\mathbf{R_1\ \theta\text{-}JOIN_C\ R_2}$ یا فقط $\mathbf{R_1\ JOIN_C\ R_2}$
  - Theta
    - $=$ EQUI-JOIN
    - $\neq$ NOT EQUI-JOIN
    - $<$ LESS THAN-JOIN
    - $\leq$ LESS EQUI-JOIN
    - $>$ GREATER THAN-JOIN
    - $\geq$ GREATER EQUI-JOIN

عملگر پیوند (ادامه)

- شرط ساده پیوند (c): $R_1.A_i$ **theta** $R_2.B_j$

صفات پیوند

که باید **هم‌دامنه** و **ناهم‌نام** باشند.

چون نتیجه JOIN رابطه است و در headingاش صفت تکراری نباید وجود داشته باشد.

- **نکته:** اگر صفات پیوند هم‌نام باشند، حداقل یکی را باید دگرنامی کرد (به دلیل وجود این راه حل، حساسیتی در وجود صفت مشترک نداریم).
- در حالت کلی شرط پیوند می‌تواند به صورت زیر باشد که در آن c_1، ...، c_n قالب بالا (قالب شرط ساده پیوند) را دارند.

<c_1> AND <c_2> AND ... AND <c_n>

مثال

<R1.A1 = R2.B1> **AND** <R1.A2 = R2.B2>

عملگر پیوند (ادامه)

مثال: مشخصات کامل جفت تهیه‌کننده-قطعه از یک شهر را بدهید.

$$R_1 := S \bowtie_{S.CITY=P.PCITY} (P \text{ RENAME } CITY \text{ AS } PCITY)$$

S (S#, SNAME, STATUS, CITY)

S1	C1
S2	C2
S3	C3
S4	C4
S5	C5
S6	C6

P (P#, . . . , W, CITY)

P1	5	C1
P2	6	C2
P3	4	C1
P4	7	C4
P5	10	C5

R_1 (S#, . . . , CITY, P#, . . . , W, PCITY)

S1	C1	P1	5	C1
S1	C1	P3	4	C1
S2	C2	P2	6	C2
~~S3~~	تاپل پیوندشدنی ندارد.			
S4	C4	P4	7	C4
S5	C5	P5	10	C5
~~S6~~	تاپل پیوندشدنی ندارد.			

عملگر پیوند (ادامه)

❑ **عملکرد:**

$$R_3 = R_1 \bowtie_C R_2$$

$$H_{R_3} = H_{R_1} \cup H_{R_2}$$

- در بدنه R_3 تاپل‌های پیوندشدنی از دو رابطه قرار دارند.

❑ **خصوصیات:**

- $R_1 \bowtie_C R_2 = \sigma_C(R_1 \times R_2)$ حاصل Theta-Join در حالت عمومی، زیرمجموعه‌ای افقی از ضرب کارتزین است که در آن تاپل‌هایی از حاصلضرب که حائز شرط پیوند هستند حضور دارند.

⊂ وقتی در شرط پیوند، تساوی بخشی از کلید هر دو رابطه را داده باشیم.

نکته

اگر $R' = R_1 \bowtie_C R_2$ باشد، آنگاه $CK_{R'} \subseteq CK_{R_1} \cup CK_{R_2}$

❑ **در عمل:** دستور INNER JOIN در SQL ، پیاده‌سازی این نوع از پیوند است.

گونه‌های خاص عملگر پیوند – پیوند طبیعی

پیوند طبیعی (Natural Join)

- گونه‌ای از پیوند است که دو ویژگی دارد:

۱- Theta: =

۲- صفات پیوند یک بار در جواب می‌آیند. (صفت یا صفات پیوند باید هم‌نام هم باشند.)

مثال

$R_2 := S \bowtie P$

R_2 (S#, . . . , CITY, P#, . . . , W)

S1	C1	P1	5
S1	C1	P3	4
S2	C2	P2	6
S4	C4	P4	7
S5	C5	P5	10

گونه‌های خاص عملگر پیوند – پیوند طبیعی (ادامه)

- در **پیوند طبیعی**، پیوند روی تساوی مقادیر تمام صفات مشترک انجام می‌شود.

$$R_1: (A, B, C)$$

$$R_2: (A, F, C)$$

$$R' = R_1 \bowtie R_2$$

$$R': (A, B, C, F)$$

- اگر $H_{R_1} \cap H_{R_2} = \emptyset$، آنگاه $R_1 \bowtie R_2 = R_1 \times R_2$.
- اگر $H_{R_1} = H_{R_2}$، آنگاه $R_1 \bowtie R_2 = R_1 \cap R_2$.

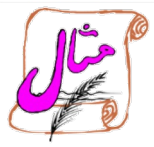
- **در عمل:** دستور NATURAL JOIN در SQL، پیاده‌سازی این نوع از پیوند است که پیوند را روی همه صفات مشترک انجام می‌دهد.

گونه‌های خاص عملگر پیوند – نیم‌پیوند

نیم‌پیوند (Semijoin)

- در شکل عمومی با هر Theta نوشته می‌شود.
- **نماد:** $\ltimes_C$ (در چپ تعریف شده)
- **مدل ریاضی:** $R_3 := R_1 \ltimes_C R_2 = \Pi_{\langle H_{R_1} \rangle}(R_1 \bowtie_C R_2)$
- **عملکرد:**
  - $H_{R_3} = H_{R_1}$
  - در بدنه R_3: تاپل‌های پیوند شدنی از رابطه چپ

گونه‌های خاص عملگر پیوند – نیم‌پیوند (ادامه)



$$R_3 := S \ltimes_{S.CITY=P.PCITY} (P \text{ RENAME } CITY \text{ AS } PCITY)$$

R_3 (S#, . . . , CITY)

S1	C1
S2	C2
S4	C4
S5	C5

کاربرد این عملگر چیست؟

- **تمرین:** عملگر نیم‌پیوند در SQL شبیه‌سازی شود.

گونه‌های خاص عملگر پیوند – برون‌پیوند

برون‌پیوند (Outer Join)

- Theta هر چیزی می‌تواند باشد.
- سه گونه دارد:

۱- Left O. J. $\quad ⟕_C$

۲- Right O. J. $\quad ⟖_C$

۳- Full O. J. $\quad ⟗_C$

- عملکرد $R_4 := R_1 ⟗_C R_2$:
  - $H_{R_4} = H_{R_1} \cup H_{R_2}$
  - در بدنه R_4: تاپل‌های پیوند شدنی از دو رابطه و

تاپلهای‌های پیوندناشدنی از رابطه چپ گسترش‌یافته با هیچ‌مقدار (Null Value)

- **در عمل:** دستور LEFT/RIGHT/FULL OUTER JOIN در SQL ، پیاده‌سازی این نوع از پیوند است.

گونه‌های خاص عملگر پیوند – برون‌پیوند (ادامه)

$R_4 := S \overline{\bowtie} P$

R_4 (S#, . . . , CITY, P#, . . . , W)

S1	C1	P1	5
S1	C1	P3	4
S2	C2	P2	6
S4	C4	P4	7
S5	C5	P5	10
S3	C3	?	?
S6	C6	?	?

کلید R_4 (CK_{R4}) چیست؟ بی‌تردید کلید اصلی ندارد.

- مشکل Outer Join:

۱- از نظر ریاضی رابطه نیست، چون کلید اصلی ندارد.

۲- مصرف حافظه زیاد

این عملگرها در عمل چه کاربردی دارند؟

عملگر نیم‌تفریق

نیم‌تفریق (Semi Minus)

- R_1 **SEMIMINUS** R_2 = R_1 **MINUS** (R_1 **SEMIJOIN** R_2)
- عملکرد
  - $H_{R_5} = H_{R_1}$
  - در بدنه R_5: تاپل‌های پیوند نشدنی از رابطه چپ

عملگر تقسیم

عملگر تقسیم (Divide)

مفروضند رابطه‌های:

$$R_1(\underbrace{A_1, A_2, \ldots, A_n}_{X}, \underbrace{B_1, B_2, \ldots, B_m}_{Y})$$

$$R_2(B_1, B_2, \ldots, B_m)$$

شرط عمل:

- $R_3(X) := R_1(X,Y) \div R_2(Y) \longrightarrow H_{R_2} \subseteq H_{R_1}$

عملکرد:

۱- $H_{R_3} = X = H_{R_1} - H_{R_2}$

۲- در بدنه R_3: بخش X از تاپلهایی از R_1 که حاوی تمام مقادیر Y از R_2 باشند.

عملگر تقسیم (ادامه)

لیست تولیدکنندگانی که همه انواع قطعات را تولید کرده‌اند.

$R_1(S\#, P\#) \div R_2(P\#) = R_3(S\#)$

S1	P1	P1	S1
S1	P2	P2	
S1	P3	P3	
S2	P1		
S2	P2		
S3	P1		

$R_1(S\#, P\#) \div R_4(P\#) = R_5(S\#)$

S1	P1	P1	S1
S1	P2	P2	S2
S1	P3		
S2	P1		
S2	P2		
S3	P1		

Outer Divide چیست؟

عملگر تقسیم (ادامه)

- ضرب و تقسیم جبر رابطه‌ای لزوماً عکس هم نیستند.STNAME

- **تمرین:** لیست نام دانشجویانی که همه دروس عملی دانشکده کامپیوتر را با موفقیت گذرانده‌اند.

$$R := \Pi_{\langle STNAME \rangle}(STT \bowtie (\sigma_{Grade \geq 10}(\Pi_{\langle STID,COID \rangle}(STCOT)) \div \Pi_{\langle COID \rangle}(\sigma_{Type='p' \wedge Dept='CE'}(COT))))$$

- **تمرین:** عملگر تقسیم را در SQL شبیه‌سازی کنید.

- **تمرین:** Q3 و Q4 (صفحه A-3 از یادداشت‌های تکمیلی سری II) را بدون استفاده از عملگر DIVIDE بنویسید.

عملگر گسترش – EXTEND

- صفت یا صفاتی را به عنوان (heading) یک رابطه اضافه می‌کند. حاصل، رابطه دیگری است.

EXTEND STUD **ADD** STADDRESS

STUD (STID, …, STD, STADDRESS)

- در SQL با ALTER TABLE پیاده‌سازی شده ولی ALTER ستون(هایی) را به همان جدول اضافه می‌کند.

- با این عملگر می‌توانیم یک ستون محاسبه‌شدنی به رابطه اضافه نماییم.

عملگر تلخیص – SUMMARIZE

- تاپل‌های رابطه را گروه‌بندی می‌کند به نحوی که مقدار صفت (صفات) گروه‌بندی در هر گروه یکسان باشد؛ معمولاً با یک یا چند تابع جمعی استفاده می‌شود.
- این عملگردر SQL با GROUP BY پیاده‌سازی شده است.

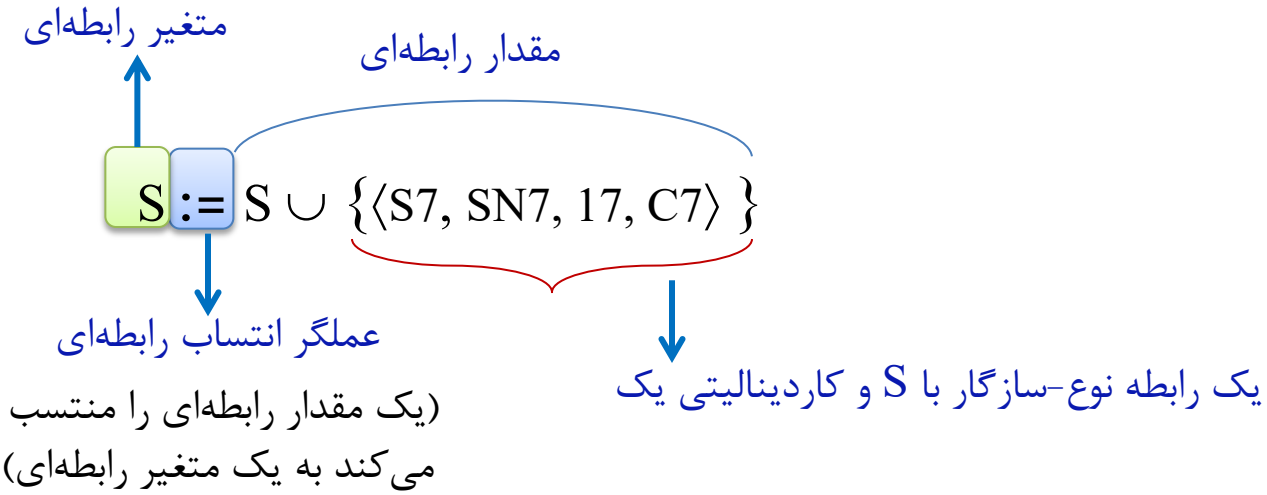
SUMMARIZE STCOT **BY** (STID) **ADD AVG**(GRADE) **AS** AVER

- برای این پرسش‌ها، اول عنوان (Heading) رابطه جواب را تعیین می‌کنیم.
- به جای AVG می‌توانیم از توابع جمع و یا گروهی دیگر مانند MIN (حداقل)، MAX (حداکثر)، SUM (جمع) و یا COUNT (شمارشگر تاپل‌ها) استفاده کنیم.

عملیات ذخیره‌سازی با جبر رابطه‌ای

- از لحاظ تئوریک می‌توان عملیات ذخیره‌سازی را هم با عملگرهای جبر رابطه‌ای انجام داد.

عمل	عملگر
درج	∪
حذف	−
به‌هنگام‌سازی	اول − بعد ∪



مقایسه دو رابطه

مقایسه دو رابطه

- دو رابطه R_1 و R_2 مقایسه‌شدنی (قابل قیاس) هستند، هر گاه نوع-سازگار باشند ($H_{R_2} = H_{R_1}$)
- در مقایسه رابطه R_1 با R_2، بدنه R_1 با بدنه R_2 مقایسه می‌شود از نظر هم مجموعگی، زیرمجموعگی و زبرمجموعگی

$$\Pi_{\langle STID \rangle}(STT) * \Pi_{\langle STID \rangle}(SCR)$$

$$* \in \{\subset, \supset, \subseteq, \supseteq, =, \neq\}$$

- پاسخ عمل مقایسه: یا T یا F. به طور مثال در رابطه فوق:
  - اگر $\supset$ باشد، پاسخ T است اگر حداقل یک دانشجو باشد که درسی انتخاب نکرده باشد.
  - اگر $\subset$ باشد، پاسخ T است اگر حداقل در یک عمل ذخیره‌سازی در این DB قاعده جامعیت C2 رعایت نشده باشد (حذف از دانشجو و یا درج در انتخاب درس).

کامل بودگی جبر رابطه‌ای

- جبر رابطه‌ای **زبانی** است از نظر رابطه‌ای **کامل** (Relational Completeness) یعنی هر رابطه معتبر متصور از مجموعه رابطه‌های ممکن را می‌توان به کمک یک عبارت جبر رابطه‌ای بیان کرد.
- جبر رابطه‌ای ضابطه تشخیصِ کامل بودن زبان‌های رابطه‌ای است.
  - اگر هر رابطه‌ای را که با جبر رابطه‌ای می‌توان نشان داد، با زبانی مدعی کامل بودن رابطه‌ای بتوان نشان داد، آن زبان از نظر رابطه‌ای **کامل** است.
- **کاربردهای جبر رابطه‌ای:**
  - عملیات بازیابی
  - عملیات ذخیره‌سازی
  - تعریف انواع رابطه‌های مشتق (رابطه مجازی، لحظه‌ای و ...) مثال: تعریف دید (View) در SQL
  - ....

مباحث تکمیلی در جبر رابطه‌ای

- برای نوشتن یک پرسش (Query)، اصولا به ترتیب زیر باید مشخص کنیم که:

۱- از چه رابطه‌هایی استفاده کنیم.

۲- از چه عملگرهایی استفاده کنیم (حتی‌الامکان با کمترین تعداد عملگر)

۳- چه ترتیبی از عملگرها را استفاده کنیم.

- مثال‌هایی از کاربرد جبر رابطه‌ای را در عملیات در RDB (در یادداشت‌های تکمیلی سری II) (صفحه A-1 و A-2) مطالعه نمایید.

حساب رابطه‌ای

- **حساب رابطه‌ای** شاخه‌ای است از منطق ریاضی، منطق مسندات.
- حساب رابطه‌ای و جبر رابطه‌ای معادلند. یعنی هر رابطه‌ای را که بتوان با یک عبارت جبر رابطه‌ای نوشت، می‌توان با عبارتی از حساب رابطه‌ای هم نوشت و برعکس.
- حساب رابطه‌ای حالت **توصیفی** دارد ولی جبر رابطه‌ای حالت **دستوری** دارد.

Descriptive

به کمک عبارات منطقی، شرایط ناظر به رابطه را برای سیستم توصیف می‌کنیم.

Prospective

دستورات عملیاتی به سیستم می‌دهیم.

- حساب رابطه‌ای هم ضابطه تشخیص زبان‌های رابطه‌ای کامل است.

حساب رابطه‌ای – متغیر تاپلی

- **متغیر تاپلی (Tuple Variable) یا متغیر طیفی (Range Variable):**
  - متغیری است که مقادیر آن تاپل‌های یک رابطه است (هر لحظه یک تاپل).

RANGVAR SX **RANGES OVER** S;

RANGVAR PX **RANGES OVER** P;

RANGVAR SPX **RANGES OVER** SP;

سورها (Quantifiers)

- سور وجودی EXISTS X (F): حداقل یک مقدار برای متغیر X وجود دارد به نحوی که به ازای آن، فرمول F به درست ارزیابی شود.
- سور همگانی (عمومی) FOR ALL X (F): به ازای تمام مقادیر متغیر X، فرمول F به درست ارزیابی می‌شود.

با فرض اینکه X از مجموعه اعداد صحیح مثبت مقدار می‌گیرد.

EXISTS X (X<10) حاصل ارزیابی: TRUE

FOR ALL X (X<10) حاصل ارزیابی: FALSE

حساب رابطه‌ای – سورها (ادامه)

- **یادآوری:** بین این دو سور روابط زیر وجود دارد.

FOR ALL X (F) = NOT EXISTS X (NOT F)

EXISTS X (F) = NOT (FORALL X (NOT F))

FORALL X (F) $\Rightarrow$ EXISTS X (F)

NOT EXISTS X (F) $\Rightarrow$ NOT FORALL X (F)

- بر اساس روابط فوق می‌توان روابط پیچیده دیگری را نیز استنباط کرد مانند روابط هم ارزی زیر:

FORALL X (F AND G) = NOT EXISTS X (NOT(F) OR NOT(G))

FORALL X (F OR G) = NOT EXISTS X (NOT(F) AND NOT(G))

EXISTS X (F OR G) = NOT FORALL X (NOT(F) AND NOT(G))

EXISTS X (F AND G) = NOT FORALL X (NOT(F) OR NOT(G))

حساب رابطه‌ای – فرمول خوش‌ساخت

تعریف

یک **فرمول خوش ساخت (WFF)** به صورت زیر تعریف می‌شود:

- اگر R یک رابطه و T یک تاپل یا متغیر تاپلی تعریف شده روی R باشد، آنگاه R(T) یک فرمول اتمی است.

  [R(T) یعنی، T یک عنصر (تاپلی) از R است.]

- اگر T_i یک متغیر تاپلی روی رابطه R و A یک صفت از R باشد و T_j یک متغیر تاپلی بر روی S و B یک صفت از S باشد، آنگاه $T_i.A \text{ theta } T_j.B$ یک فرمول اتمی است (theta یک از عملگرهای متعارف مقایسه‌ای است).

- $T_i.A \text{ theta } C_2$ و $C_1 \text{ theta } T_j.B$ و $C_1 \text{ theta } C_2$ نیز که در آن C_i یک مقدار ثابت است، فرمول اتمی هستند.

- اگر F_1 و F_2 فرمول باشند، آنگاه $(F_1 \text{ AND } F_2)$، $(F_1 \text{ OR } F_2)$، $\text{NOT}(F_1)$ نیز فرمول هستند.

- اگر F یک فرمول و T یک متغیر تاپلی باشد، آنگاه EXISTS T(F) و FORALL T(F) نیز فرمول هستند.

حساب رابطه‌ای – عبارت حساب رابطه‌ای

تعریف

اگر X (Y و ...) متغیرتاپلی روی رابطه $R(A_1, A_2, \ldots, A_n)$ (S و ...) باشد در اینصورت شکل کلی **عبارت حساب رابطه‌ای** بدین صورت است

(target-items) [**WHERE** F]

که در آن target-items فهرستی از صفات متغیر تاپلی X (و Y و ...) به صورت $X.A_1, X.A_2, \ldots, X.A_n$ (به طور مشابه برای Y و ...) و F یک فرمول خوش‌ساخت است.

در واقع عبارت حساب رابطه‌ای توصیف کننده مجموعه تاپلهایی است که شرایط F را ارضا می‌نمایند.

مثال

- STX.STID شماره تمام دانشجویان در رابطه STT
- STX.STID **WHERE** STX.STDEID=‘D11’ شماره دانشجویان گروه آموزشی D11
- (STX.STID, STX.STL) **WHERE EXISTS** STCOX (STX.STID=STCOX.STID **AND** STCOX.COID=‘COM11’)

شماره دانشجویی و مقطع تحصیلی آنهایی که درس COM11 را انتخاب کرده‌اند.

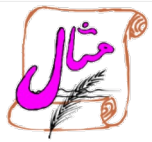
حساب رابطه‌ای – عبارت حساب رابطه‌ای (ادامه)

شماره همه تهیه کنندگان

- SX.S#

نام تهیه کنندگان شهرستان C2 که وضعیت آنها بزرگتر از 15 باشد.

- SX.SNAME **WHERE** SX.CITY=‘C2’ **AND** SX.STATUS> 15



نام تهیه کنندگانی که حداقل یک قطعه آبی‌رنگ تهیه کرده‌اند.

- SX.SNAME **WHERE EXISTS** SPX (SPX.S#=SX.S# **AND**
  EXISTS PX (PX.P#=SPX.P# **AND** PX.COLOR=‘Blue’))

نام جفت تهیه کنندگانی که در یک شهر بوده و حداقل یک قطعه مشترک تولید کرده‌اند.

- SX.SNAME, SY.SNAME **WHERE** SX.CITY=SY.CITY **AND NOT** (SX.S#=SY.S#)
  AND EXISTS SPX (**EXISTS** SPY SPX.S#=SX.S# **AND**
  SPY.S#=SY.S# **AND** SPX.P#=SPY.P#)

حساب رابطه‌ای – عبارت حساب رابطه‌ای (ادامه)

عنوان درسهای را بدهید که که تمام دانشجویان رشته کامپیوتر در ترم دوم ۹۸-۹۹ در آنها قبول شده‌اند.

- COX.TITLE **WHERE FORALL** STX (**NOT** STX.STJ=‘CE’ **OR**
  EXISTS STCOX (STCOX.STID=STX.STID **AND** STCOX.COID=COX.COID **AND**
  STCOX.YR=‘98-99’ **AND** STCOX.TR=‘2’ **AND** STCOX.GRADE>=10) **)**

مثال‌های بیشتر در کتاب‌های مرجع و یادداشتهای تکمیلی سری II.

پرسش و پاسخ . . .

amini@sharif.edu