

# به نام آنکه جان را فکرت آموخت

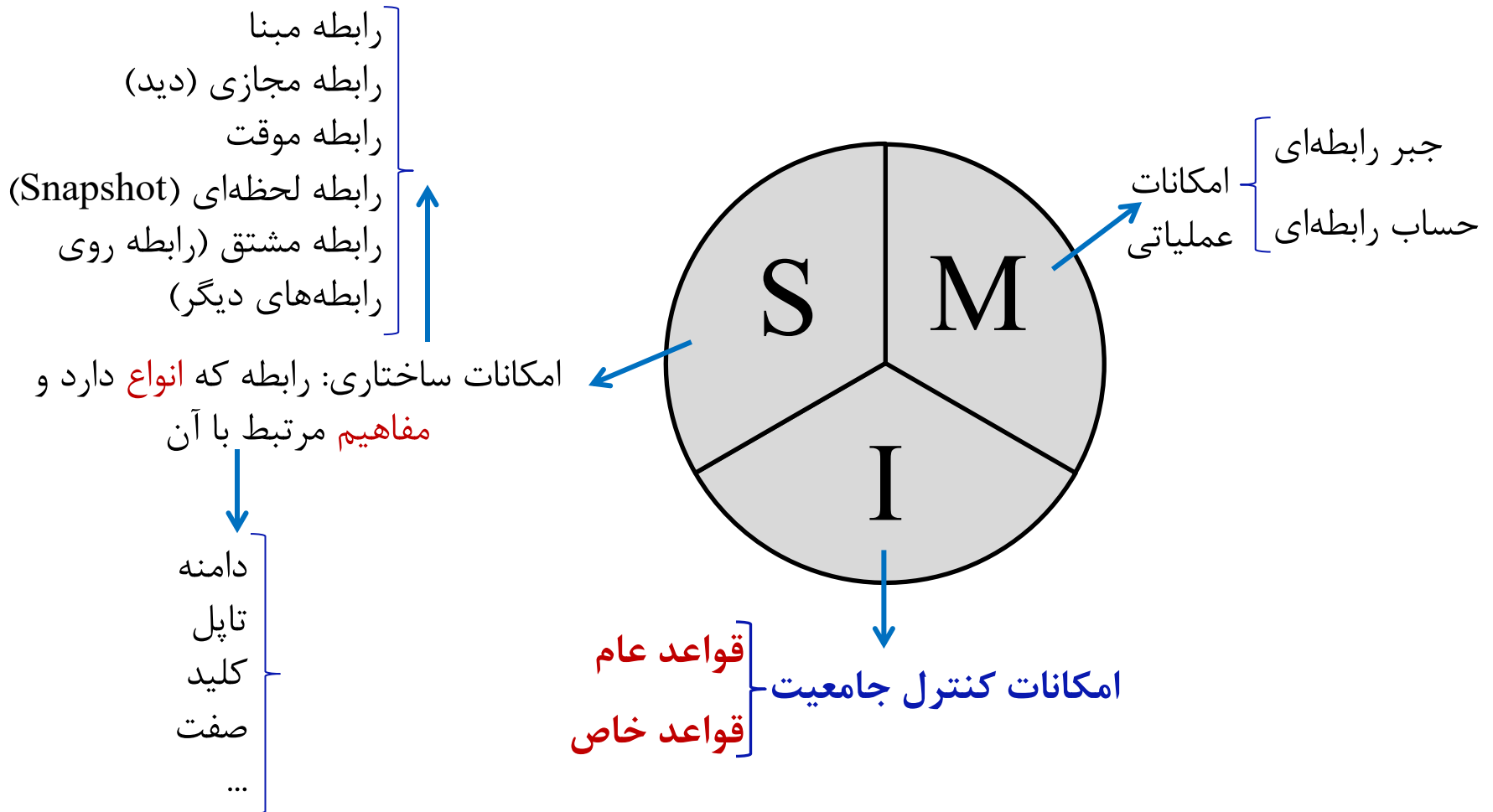


## بخش هشتم: جامعیت در مدل رابطه‌ای

مرتضی امینی

نیمسال اول ۱۴۰۲-۱۴۰۱

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





## جامعیت پایگاه داده‌ها (DB Integrity)



صحت و سازگاری داده‌های ذخیره شده در پایگاه داده‌ها

جنبه‌های کیفی داده (Data Quality Features)

مسئولیت کنترل جامعیت DB با RDBMS است.

بر اساس اطلاعاتی که کاربر [تیم طراح - پیاده‌ساز] به سیستم می‌دهد.

قواعد یا محدودیت‌های جامعیتی (Integrity Rules/Constraints)

IRها [ICها] با استفاده از دستورات زبان پایگاهی به سیستم داده می‌شوند.

اعلانی: قواعد به نحوی اعلان می‌شوند.

اجرایی: قواعد در یک رویه به سیستم داده می‌شوند.



□ هر DBMS ای باید بتواند جامعیت پایگاه داده‌ها را کنترل و تضمین کند.

□ **دلیل:** زیرا همیشه ممکن است عواملی سبب نقض جامعیت شوند. از جمله:

□ اشتباه در برنامه‌های کاربردی (به ویژه اشتباهات معنایی)

□ اشتباه در وارد کردن داده‌ها

□ وجود افزونگی کنترل نشده

□ اجرای همروند تراکنشها به گونه‌ای که داده نامعتبر یا ناسازگار ایجاد شود.

□ خرابی‌های سخت‌افزاری و نرم‌افزاری که منجر به بروز ناسازگاری یا نقض صحت داده‌ها می‌شود.



❑ اعمال قواعد جامعیتی (IRها) یا محدودیت‌های جامعیتی (ICها) برای سیستم **سربار** دارد.

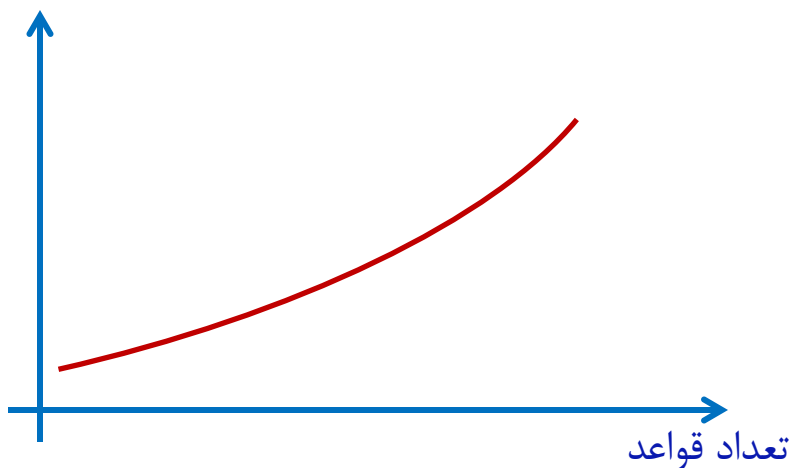
❑ منشأ سربار (دلایل بروز سربار) در DBMS

❑ انجام نگاشت‌ها (ناشی از معماری)

❑ اعمال و کنترل قواعد جامعیتی

❑ اعمال ضوابط و کنترل‌های امنیت داده‌ها در سطح DBMS

کار سیستم





## □ IRها [ICها] در مدل رابطه‌ای

۱- قواعد [محدودیت‌های] عام: ناوابسته به داده‌های محیط: فراقواعد (MetaRules)

۲- قواعد [محدودیت‌های] خاص: وابسته به داده‌های محیط: قواعد کاربری (User Defined)

یا قواعد فعالیت‌های محیط (Business Rules)

## □ قواعد عام در مدل رابطه‌ای

□ قاعده C1: جامعیت موجودیتی

□ قاعده C2: جامعیت ارجاعی



# قواعد عام در مدل رابطه‌ای – قاعده جامعیت موجودیتی C1

بخش هشتم: مفاهیم اساسی مدل داده رابطه‌ای

۷

## □ قاعده (محدودیت) C1 – قاعده جامعیت موجودیتی (Entity IR)

□ ناظر است به PK.

□ هیچ جزء تشکیل‌دهنده PK نباید هیچ مقدار (Null) داشته باشد.

□ دلیل:

✓ PK عامل تمییز تاپل‌ها است.

✓ تاپل در مدل رابطه‌ای نمایشگر نمونه موجودیت است.

✓ PK عامل تمییز نمونه موجودیت‌ها است و تضمین‌کننده

دستیابی به تک‌موجودیت

۱- محدودیت یکتایی مقدار (با UNIQUE

فقط این محدودیت کنترل می‌شود)

۲- محدودیت هیچ‌مقدارناپذیری

□ مکانیزم اعمال C1: اعلان PK به سیستم **کنترل می‌کند** ←



## قواعد عام در مدل رابطه‌ای – قاعده جامعیت ارجاعی C2

بخش هشتم: مفاهیم اساسی مدل داده رابطه‌ای

۸

### □ قاعده (محدودیت) C2 – قاعده جامعیت ارجاعی (Referential IR)

□ ناظر است به FK.

□ اگر  $R_2.A_i$  در  $R_2$ ، کلید خارجی باشد، مقدار  $A_i$  در هر تاپل از  $R_2$  باید در  $R_1$  مقدار قابل انطباق (Matchable Value) داشته باشد.

□ به عبارت دیگر باید هر مقدار معلوم  $A_i$  در  $R_2$ ، در  $R_1$  نیز وجود داشته باشد. یعنی در عمل می‌تواند در  $R_2$  مقدار آن Null باشد (البته اگر جزء تشکیل‌دهنده کلید  $R_2$  نباشد).

□ دلیل نیاز به این قاعده:

- FK عامل ارجاع است؛ ارجاع به نمونه موجودیت (ارجاع مقداری و نه ارجاع از طریق اشاره‌گر).
- در واقعیت نمی‌توان به نمونه موجودیت ناموجود ارجاع داد.

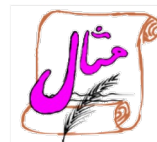




# قواعد عام در مدل رابطه‌ای – قاعده جامعیت ارجاعی C2 (ادامه)

بخش هشتم: مفاهیم اساسی مدل داده رابطه‌ای

۹



STT (STID, ...)

777  
888  
444

STCOT (STID, COID, ...)

777 CO1  
... ...  
444 CO4

INSERT INTO STCOT

VALUES ('999', 'CO9', ...)

□ چون برای 999 مقدار قابل انطباق در STT وجود ندارد، پس این درخواست رد می‌شود.



# قواعد عام در مدل رابطه‌ای – قاعده جامعیت ارجاعی C2 (ادامه)

بخش هشتم: مفاهیم اساسی مدل داده رابطه‌ای

۱۰

□ برای اعمال قاعده C2 در مدل رابطه‌ای لازم است:

۱- معرفی FKها به سیستم

CREATE TABLE STCOT

(STID CHAR(6) NOT NULL

COID CHAR(6) NOT NULL

TR CHAR(1)

YR CHAR(5)

GR DEC(2, 2))

CHECK (0 <= GR <= 20)

PRIMARY KEY (STID, COID)

FOREIGN KEY STID REFERENCES STT (STID)

ON DELETE CASCADE

ON UPDATE CASCADE

FOREIGN KEY COID REFERENCES COT (COID)

ON DELETE CASCADE

ON UPDATE CASCADE

۲- دادن گراف ارجاع

۳- مشخص کردن روش اعمال در عملیات حذف و به‌هنگام‌سازی مقدار کلید اصلی

(در درج روش خاصی لازم نیست و در صورت عدم

وجود تاپل مرجع، درخواست رد می‌شود.)

۲- گراف ارجاع

۳- روش اعمال (انتشار عمل)

۱- معرفی FK



# قواعد عام در مدل رابطه‌ای – قاعده جامعیت ارجاعی C2 (ادامه)

بخش هشتم: مفاهیم اساسی مدل داده رابطه‌ای

۱۱

□ روش‌های اعمال C2 در حذف (بعضاً در به‌هنگام‌سازی):

## ۱- روش CASCADE: انتشاری یا تسلسلی

در این روش با حذف (بروزرسانی) تاپل مرجع، تمام تاپل‌های رجوع کننده به آن حذف (بروزرسانی) می‌شوند.

هر چه گراف ارجاع سنگین‌تر باشد، کار سیستم در اینجا بیشتر است.  
**DELETE FROM STT**  
**WHERE STID='444'**

منجر می‌شود به

**DELETE FROM STCOT**  
**WHERE STID='444'**

## ۲- روش RESTRICTED: روش منوط به ... (یا مشروط به ...) یا روش تعویقی

در این روش اگر بخواهیم تاپل مرجع را حذف کنیم، خطا می‌دهد و ابتدا باید تاپل‌های ارجاع کننده به آن حذف شوند.



# قواعد عام در مدل رابطه‌ای – قاعده جامعیت ارجاعی C2 (ادامه)

بخش هشتم: مفاهیم اساسی مدل داده رابطه‌ای

۱۲

□ روش‌های اعمال C2 در حذف (و بعضاً در به‌هنگام‌سازی):

## ۳- روش SET TO NULL: روش هیچ‌مقدارگذاری یا Nullifying

در این روش با حذف تاپل مرجع، FK در تاپل‌های رجوع کننده Null می‌شود به شرط آنکه FK جزء سازنده PK نباشد و Not Null نیز نباشد.

## ۴- روش SET TO DEFAULT: روش درج پیش‌فرض

در این روش، با حذف تاپل مرجع، FK با مقدار پیش‌فرض جاگذاری می‌شود به شرط آنکه FK جزء سازنده PK نباشد و تاپلی با مقدار پیش‌فرض برای PK در رابطه مرجع وجود داشته باشد.



# قواعد عام در مدل رابطه‌ای – قاعده جامعیت ارجاعی C2 (ادامه)

بخش هشتم: مفاهیم اساسی مدل داده رابطه‌ای

۱۳

□ روش‌های اعمال C2 در حذف (و بعضاً در به‌هنگام‌سازی):

۵- روش NO ACTION: عدم اقدام به صورت موقت

تاپل مرجع حذف بشود ولی اقدام دیگری موقتاً انجام نشود. در این مورد طراح-پیاده‌ساز می‌پذیرد که **موقتاً** (معمولاً تا پایان یک تراکنش و نه بعد از آن) محدودیت C2 نقض شود.

□ در حالت وجود **چرخه ارجاع** کدام روش انجام شدنی است؟

□ نمی‌توان روش RESTRICTED را در حالت کلی اعمال کرد. با روش CASCADE هم ممکن است تاپل‌های ناخواسته حذف شود.

□ در این مواقع NO ACTION و یا SET TO NULL بر حسب شرایط می‌تواند راهگشا باشد.



□ قواعد خاص در مدل رابطه‌ای:

□ محدودیت دامنه‌ای (میدانی)


□ محدودیت صفتی

□ محدودیت رابطه‌ای

□ محدودیت پایگاهی



## محدودیت دامنه‌ای (میدانی)

 این محدودیت ناظر است به دامنه، مشخص‌کننده نوع و طیف مقادیر دامنه

 در همان دستور CREATE DOMAIN اعلان می‌شود.

دستور ایجاد دامنه '?...?' **CREATE DOMAIN** GRADE DEC(2, 2) **DEFAULT**

نام محدودیت (اختیاری) **CONSTRAINT** GRADECONST

**CHECK VALUE BETWEEN** (0, 20)



**DROP DOMAIN** GRADE دستور حذف دامنه



## □ محدودیت صفتی [ستونی]

□ این محدودیت ناشی می‌شود از محدودیت دامنه‌اش

□ صفت می‌تواند محدودیت‌های دیگری هم داشته باشد، به شرطی که ناقض محدودیت دامنه‌ای‌اش نباشد.

محدودیت‌های ناظر به صفت:




۱- صفت نمره باید بین ۰ تا ۲۰ باشد.

۲- صفت سن کاهش نمی‌یابد (محدودیت پردازشی).

محدودیت ۱، یک **محدودیت وضعیتی** است ولی محدودیت ۲، یک **محدودیت گذاری** است.






محدودیت صفتی را چگونه می‌توان به سیستم اعلان کرد؟ 

۱- با تعریف دامنه‌اش اعلان می‌شود.

۲- در همان دستور CREATE TABLE با عبارت CHECK اعلان می‌شود.

جدول انتخاب درس 

**CREATE TABLE STCOT**

(STID ...

COID ...

TR ...

GR ...)

**CHECK (0 <= GR AND GR <= 20)**

۳- با ASSERTION اعلان می‌شود. (در اسلایدهای بعدی معرفی می‌شود)

۴- با TRIGGER به سیستم داده می‌شود. (در اسلایدهای بعدی معرفی می‌شود)



### □ محدودیت رابطه‌ای

□ ناظر است به تاپل‌های یک رابطه (درون رابطه‌ای Intra-relational).

□ حیطه اعمالش یک رابطه است و مقادیر مجاز یک متغیر رابطه‌ای را مشخص می‌کند.

□ باید در هر عملی که بر روی رابطه انجام می‌شود (که منجر به تغییر در متغیر رابطه‌ای می‌گردد)

کنترل شود.


تهیه‌کنندگان ساکن شهر C2 نمی‌توانند مقدار وضعیت بیش از ۱۵ داشته باشند.






## □ محدودیت پایگاهی

□ ناظر است به تاپل‌های بیش از یک رابطه که به نحوی با هم ارتباط معنایی [منطقی] دارند.

 رابطه بین جداول STT و STCOT  
یا رابطه بین جداول S و SP

 دانشجوی رشته کامپیوتر نمی‌تواند درس آمار و احتمال را از گروه آموزشی D13 (دانشکده ریاضی) انتخاب کند. رابطه‌های دخیل: STT، COT و STCOT

 تهیه‌کننده ساکن شهر C7 با وضعیت کمتر از ۱۵، نمی‌تواند قطعه آبی رنگ با وزن بیش از ۱۰ گرم به تعداد بیش از ۱۰۰ عدد تهیه کند.

□ محدودیت‌های رابطه‌ای و پایگاهی چگونه اعمال می‌شوند؟

■ با ASSERTION (اعلانی)

■ با TRIGGER (اجرایی)



## اظهار – ASSERTION ☐

☐ امکانی است اِعلانی برای بیان محدودیت‌های رابطه‌ای و پایگاهی [و صفتی]

**CREATE ASSERTION** *name*  
**CHECK** *condition(s)*

☐ در قسمت *condition(s)* می‌توان یک شرط ساده، یک عبارت بولی شامل چند شرط و نیز یک عبارت **SELECT** معتبر نوشت (همانطور که بعد از عبارت **WHERE** نوشته می‌شود).

☐ دستور حذف اظهار

**DROP ASSERTION** *name*



## امکانات بیان محدودیت‌ها – اظهار (ادامه)

۲۱

بخش هشتم: مفاهیم اساسی مدل داده رابطه‌ای

با این اظهار، محدودیت یکتایی مقادیر صفت کد ملی STNATID اعلان می‌شود.



```
CREATE ASSERTION UNC-CHECK  
CHECK (UNIQUE(SELECT STNATID FROM STT))
```

**نکته:** روش رایج در توصیف اظهار این است که مجموعه تاپلهایی که ناقض قاعده جامعیتی هستند را با استفاده از SELECT توصیف می‌کنیم و آن را داخل عبارت NOT EXISTS قرار می‌دهیم.

با این اظهار این محدودیت که «جمع واحدهای انتخابی دانشجو در هر ترم-سال نباید بیش از ۲۰ واحد باشد»، اعلان می‌شود.



```
CREATE ASSERTION TOTCRED-CHECK  
CHECK (NOT EXISTS (SELECT STID  
FROM COT JOIN STCOT  
GROUP BY (STID, TR, YR)  
HAVING SUM(CREDIT) > 20) )
```



## امکانات بیان محدودیت‌ها – اظهار (ادامه)

بخش هشتم: مفاهیم اساسی مدل داده رابطه‌ای

۲۲

مثال همه دانشجویان ورودی ۹۵ دانشکده مهندسی کامپیوتر (CE) باید درس مبانی برنامه‌سازی (با کد ۴۰۱۱۱) را اخذ کرده باشند.

**CREATE ASSERTION ELEM-CHECK**

**CHECK (NOT EXISTS**

**( SELECT \* FROM STT**

**WHERE DEPT='CE' AND**

**STID LIKE '95%' AND**

**NOT EXISTS**

**( SELECT \* FROM STCOT**

**WHERE STCOT.STID = STT.STID**

**AND STCOT.COID='40111' ) )**



## TRIGGER – [راه‌انداز] رهانا

امکانی است اجرایی برای اعمال محدودیت‌های [صفتی]، رابطه‌ای و پایگاهی قبل یا بعد از بروز یک

رویداد و یا به جای یک رویداد (معمولاً تغییر دهنده داده‌ها).

**CREATE TRIGGER** *name* {BEFORE | AFTER | INSTEAD OF}

{INSERT | DELETE | UPDATE OF *columnlist*

ON *tablename*

[REFERENCING {OLD | NEW} [TABLE] AS *name* ]

[FOR EACH {ROW | STATEMENT}]

{(WHEN condition(s)

SQL Procedure

})

مفهوم نظری TRIGGER: مفهوم قاعده فعال [مفهوم محوری است در ADBMS ها]



ساختار (قاعده ECA): if **E**vent on **C**ondition, then **A**ction

↓  
Insert  
Delete  
Update



با **FOR EACH ROW** بعد از بروز رویداد در هر سطر عبارت رهانا اجرا شود. ☐

با **FOR EACH STATEMENT** فقط یک بار پس از بروز رویداد (با هر تعداد سطر متاثر از آن)، عبارت رهانا اجرا شود. ☐

مثال از **FOR EACH ROW**: این رهانا این محدودیت را که «حقوق کارمند هیچگاه کاهش نمی‌یابد» اعمال می‌کند.

**CREATE TRIGGER EMP-PAY-TRIG**

**BEFORE UPDATE OF ESAL**

**ON EMPL**

**REFERENCING OLD AS OEMPL, NEW AS NEMPL**

**FOR EACH ROW**

**(WHEN OEMPL.ESAL > NEMPL.ESAL**

**SIGNAL SQL State '7005' ('salary cannot be decreased')**

**)**

سطر جدید: بعد از بروزرسانی

سطر قدیم: قبل از بروزرسانی





از FOR EACH STATEMENT: این رهانا این محدودیت را که «میانگین حقوق کارمندان نباید از ۵ میلیون کمتر باشد» اعمال می‌کند. چون با یک درج یا بروزرسانی ممکن است چندین سطر درج یا بروز شوند و در حین اجرا موقتاً این محدودیت نقض شود، لذا با FOR EACH STATEMENT، بعد از اجرای کامل عملیات، برقراری محدودیت را کنترل می‌نماییم.

```
CREATE TRIGGER EMP-AVG-TRIG
```

```
AFTER UPDATE OF ESAL
```

```
ON EMPL
```

```
REFERENCING OLD TABLE AS OEMPL, NEW TABLE AS NEMPL
```

```
FOR EACH STATEMENT
```

```
(WHEN 5000000 > (SELECT AVG(ESAL) FROM NEMPL)
```

```
BEGIN
```

```
DELETE FROM EMPL
```

```
WHERE (EID, ENAME, ..., ESAL) IN (SELECT * FROM NEMPL);
```

```
INSERT INTO EMPL
```

```
SELECT * FROM OEMPL;
```

```
END)
```

جدول قدیم: قبل از بروزرسانی

جدول جدید: بعد از بروزرسانی

اگر بیش از یک عبارت باشد، آنها را داخل BEGIN و END قرار می‌دهیم.



این رهانا باعث حفظ سازگاری در جدول PROF می‌شود تا همواره صفت SALAUG حاوی آخرین میزان افزایش حقوق استاد باشد.



```
CREATE TRIGGER EMP-PAY-TRIG
  AFTER UPDATE OF PSALARY
  ON PROF
  REFERENCING OLD AS OPROF, NEW AS NPROF
  FOR EACH ROW
  (UPDATE PROF
    SET SALAUG=NPROF.PSALARY – OPROF.PSALARY
    WHERE PROF.PID=OPROF.PID
  )
```



از کاربردهای رهانا، استفاده از آن در انجام عملیات ذخیره‌سازی از دید خارجی است (به خصوص در



سمپادهایی که از عملیات در دید خارجی پشتیبانی نمی‌کنند).

STT1 (STID, NAME, MAJOR, LEVEL)

STT2 (STID, DEPT, BDATE, NATID)

**CREATE VIEW CE-STT**

**AS SELECT STID, NAME, MAJOR**

**FROM STT1 JOIN STT2**

**WHERE DEPT='CE' AND LEVEL='BS'**

**CREATE TRIGGER INS-VIEW-TRIG**

**INSTEAD OF INSERT ON CE-STT**

**REFERENCING NEW AS NST**

**FOR EACH ROW**

**BEGIN**

**INSERT INTO STT1 VALUES ( NST.STID, NST.NAME, NST.MAJOR, 'BS')**

**INSERT INTO STT2 VALUES ( NST.STID, 'CE', NULL, NULL)**

**END**



## امکانات بیان محدودیت‌ها – رهانا (ادامه)

بخش هشتم: مفاهیم اساسی مدل داده رابطه‌ای

۲۸

این رهانا باعث اعمال قاعده C2 در عمل حذف در جدول COT می‌شود.



```
CREATE TRIGGER DEL-TRIG
  BEFORE DELETE
  ON COT
  REFERENCING OLD AS OCOT
  FOR EACH ROW
  (DELETE FROM STCOT
   WHERE STCOT.COID=OCOT.COID )
```



بخش هشتم: مفاهیم اساسی مدل داده رابطه‌ای

سازگاری مجموع حقوق (TOTAL\_SAL) در هر بخش با مقادیر حقوق (SALARY) کارمندان



کارمند EMPL (EID, ENAME, SALARY, DNO)

بخش DEPT (DNO, DNAME, TOTAL\_SAL, MANAGER)

```
(R1) CREATE TRIGGER TOTALSAL1
      AFTER INSERT ON EMPL
      FOR EACH ROW
      WHEN (NEW.DNO IS NOT NULL)
      UPDATE DEPT
      SET TOTAL_SAL = TOTAL_SAL + NEW.SALARY
      WHERE DNO = NEW.DNO
```

```
(R2) CREATE TRIGGER TOTALSAL2
      AFTER UPDATE OF SALARY ON EMPL
      FOR EACH ROW
      WHEN (NEW.DNO IS NOT NULL)
      UPDATE DEPT
      SET TOTAL_SAL = TOTAL_SAL + NEW.SALARY - OLD.SALARY
      WHERE DNO = NEW.DNO
```



```
(R3) CREATE TRIGGER TOTALSAL3  
AFTER UPDATE OF DNO ON EMPL  
FOR EACH ROW  
BEGIN  
    UPDATE DEPT  
        SET TOTAL_SAL = TOTAL_SAL + NEW.SALARY  
        WHERE DNO = NEW.DNO  
    UPDATE DEPT  
        SET TOTAL_SAL = TOTAL_SAL - OLD.SALARY  
        WHERE DNO = OLD.DNO  
END
```

```
(R4) CREATE TRIGGER TOTALSAL4  
AFTER DELETE ON EMPL  
FOR EACH ROW  
WHEN (OLD.DNO IS NOT NULL)  
    UPDATE DEPT  
        SET TOTAL_SAL = TOTAL_SAL - OLD.SALARY  
        WHERE DNO = OLD.DNO
```



**پرسش و پاسخ ...**

**amini@sharif.edu**