پایگاه داده ها

SQL (ادامه)

اصلاح پایگاه داده‌ها

- حذف (Deletion)
  - هر درخواست حذف مشابه با یک پرس و جو بیان می‌شود
  - مثال
    - همه حساب‌های بانکی در شعبه Perryridge را حذف کنید

delete from *account*
where *branch_name* = ‘Perryridge’

- همه وام‌های بانکی با میزان وام بین ۱۳۰۰ تا ۱۵۰۰ دلار را حذف کنید

delete from *loan*
where *amount* **between** 1300 **and** 1500

- همه وام‌های بانکی را حذف کنید

delete from *loan*

اصلاح پایگاه داده‌ها

- دستور **delete** فقط برروی یک رابطه عمل می‌کند
- مثال
  - همه حساب‌های بانکی در هر کدام از شعبه‌های مستقر در شهر **Brooklyn** را حذف کنید

```
delete from account
where branch_name in (select branch_name
                      from branch
                      where branch_city = 'Brooklyn')
```

اصلاح پایگاه داده‌ها

- مثال
  - همه حساب‌های بانکی با موجودی کمتر از متوسط موجودی حساب‌ها در بانک را حذف کنید

```
delete from account
where balance < (select avg (balance)
                 from account)
```

  - ابتدا متوسط موجودی حساب‌ها در بانک محاسبه می‌شود
  - سپس همه چندگانه‌ها از رابطه ***account*** با موجودی کمتر از متوسط موجودی حساب‌ها در بانک انتخاب می‌شوند
  - در نهایت چندگانه‌های فوق از رابطه ***account*** حذف می‌شوند

اصلاح پایگاه داده‌ها

- درج (Insertion)
  - برای افزودن داده‌ها به یک رابطه می‌توان یک چندگانه یا مجموعه‌ای از چندگانه‌ها را براساس نتیجه یک پرس و جو مشخص کرد
  - مثال
    - اطلاعات زیر را در پایگاه داده‌ها درج کنید
      - حساب بانکی A-973 در شعبه Perryridge ۱۲۰۰ دلار موجودی دارد

insert into *account*
  values (‘A-973’, ‘Perryridge’, 1200)

insert into *account* (*branch_name*, *balance*, *account_number*)
  values (‘Perryridge’, 1200, ‘A-973’)

اصلاح پایگاه داده‌ها

- **مثال**
  - **به همه مشتریان بانک که از شعبه Perryridge وام بانکی دریافت کرده‌اند، به عنوان جایزه یک حساب پس‌انداز با موجودی ۲۰۰ دلار بدهید. فرض کنید از شماره وام به عنوان شماره برای حساب پس‌انداز جدید استفاده شود**

```
insert into account
    select loan_number, branch_name, 200
    from loan
    where branch_name = 'Perryridge'

insert into depositor
    select customer_name, borrower.loan_number
    from borrower, loan
    where borrower.loan_number = loan.loan_number and
          branch_name = 'Perryridge'
```

اصلاح پایگاه داده‌ها

- به‌روزرسانی (Updating)
  - برای تغییر یک یا چند مقدار در یک چندگانه بدون تغییر همه مقادیر در آن چندگانه استفاده می‌شود
  - مثال
    - به هر حساب بانکی ۵ درصد سود پرداخت کنید

```
update account
set balance = balance * 1.05
```

    - به حساب‌های بانکی با موجودی بیش از ۱۰۰۰ دلار ۵ درصد سود پرداخت کنید

```
update account
set balance = balance * 1.05
where balance >= 1000
```

اصلاح پایگاه داده‌ها

- مثال
  - به حساب‌های بانکی با موجودی بیشتر از متوسط موجودی حساب‌ها در بانک ۵ درصد سود پرداخت کنید

```
update account
set balance = balance * 1.05
where balance > (select avg(balance)
                 from account)
```

اصلاح پایگاه داده‌ها

- مثال
  - به حساب‌های بانکی با موجودی بیش از ۱۰۰۰۰ دلار ۶ درصد سود و به سایر حساب‌های بانکی ۵ درصد سود پرداخت کنید

```
update account
set balance = balance * 1.06
where balance > 10000

update account
set balance = balance * 1.05
where balance <= 10000
```

  - اگر ترتیب دو دستور به‌روزرسانی فوق عوض شود، آنگاه حساب‌های بانکی با موجودی نزدیک به ۱۰۰۰۰ دلار بیش از ۱۱ درصد سود دریافت خواهند کرد
  - به‌روزرسانی فوق را می‌توان به شکل مناسب‌تری با استفاده از دستور **case** انجام داد

اصلاح پایگاه داده‌ها

update *account*
set *balance* = **case**
when *balance* <= 10000 **then** balance * 1.05
else balance * 1.06
end

- شکل کلی دستور **case** به صورت زیر است

case
when $pred_1$ **then** $result_1$
when $pred_2$ **then** $result_2$
...
when $pred_n$ **then** $result_n$
else $result_0$
end

اصلاح پایگاه داده‌ها

- به‌روزرسانی دیدها
  - مثال
    - یک کاربر خاص لازم است تا از بین اطلاعات وام‌های بانکی تنها شماره وام‌ها و نام شعبه‌های آن‌ها را مشاهده کند

```
create view loan_branch as
    select loan_number, branch_name
    from loan
```

- درج چندگانه در ***loan_branch***

```
insert into loan_branch
    values ('L-37', 'Perryridge')
```

- درج چندگانه فوق باید با درج یک چندگانه در رابطه ***loan*** نمایش داده شود

('L-37', 'Perryridge', *null*)

اصلاح پایگاه داده‌ها

- هنگام اصلاح پایگاه داده‌ها از طریق دیدها ممکن است مشکلات متعددی مشاهده شود
- مثال
  - دید تعریف شده زیر را در نظر بگیرید

```
create view loan_info as
    select customer_name, amount
    from borrower, loan
    where borrower.loan_number = loan.loan_number
```

  - درج چندگانه در ***loan_info***

```
insert into loan_info
    values ('Johnson', 1900)
```

اصلاح پایگاه داده‌ها

- درج چندگانه (‘Johnson’, 1900) در رابطه دید *loan_info* با درج چندگانه (‘Johnson’, *null*) در رابطه *borrower* و چندگانه (*null*, *null*, 1900) در رابطه *loan* نمایش داده می‌شود
- با وجود درج دو چندگانه فوق در رابطه‌های *borrower* و *loan*، رابطه دید *loan_info* شامل چندگانه (‘Johnson’, 1900) نمی‌باشد

loan_number	*branch_name*	*amount*
L-11	Round Hill	900
L-14	Downtown	1500
L-15	Perryridge	1500
L-16	Perryridge	1300
L-17	Downtown	1000
L-23	Redwood	2000
L-93	Mianus	500
null	*null*	1900

loan

customer_name	*loan_number*
Adams	L-16
Curry	L-93
Hayes	L-15
Jackson	L-14
Jones	L-17
Smith	L-11
Smith	L-23
Williams	L-17
Johnson	*null*

borrower

اصلاح پایگاه داده‌ها

- تمرین
  - با درج چندگانه زیر چه مشکلی ممکن است مشاهده شود

insert into *all_customer*
values (‘Perryridge’, ‘John’)

- به دلیل مشکلات فوق، در بسیاری از سیستم‌های پایگاه داده‌ای تنها امکان به‌روزرسانی دیدهای ساده وجود دارد

قیود صحت

- قیود صحت اطمینان می‌دهند که تغییرات ایجاد شده توسط کاربران مجاز در پایگاه داده‌ها باعث به وجود آمدن ناسازگاری داده‌ها نمی‌شود

- مثال‌هایی از قیود صحت
  - شماره یک حساب بانکی نمی‌تواند تهی باشد
  - شماره هر حساب بانکی باید منحصر به فرد باشد
  - هر شماره وام در رابطه ***borrower*** باید با یک شماره وام در رابطه ***loan*** مطابقت کند
  - حقوق هر کدام از کارمندان بانک باید از ۶ دلار در ساعت بیشتر باشد

قیود صحت

- صحت ارجاعی (Referential Integrity)
  - اطمینان می‌دهد که مقادیر مشاهده شده برای مجموعه‌ای از خصیصه‌ها در یک رابطه برای مجموعه دیگری از خصیصه‌ها در رابطه دیگر نیز مشاهده می‌شوند
    - تنها مقادیر مشاهده شده در خصیصه‌های کلید اصلی رابطه مرجع ممکن است در خصیصه‌های کلید خارجی رابطه رجوع کننده مشاهده شوند
  - در زبان SQL، کلیدهای اصلی، کاندید و خارجی را می‌توان به عنوان بخشی از دستور create table مشخص کرد
    - عبارت primary key
      - خصیصه‌های تشکیل دهنده کلید اصلی را فهرست می‌کند

قیود صحت

- عبارت **unique key**
  - خصیصه‌های تشکیل دهنده کلید کاندید را فهرست می‌کند
- عبارت **foreign key**
  - خصیصه‌های تشکیل دهنده کلید خارجی و نام رابطه مرجع را فهرست می‌کند
  - به طور پیش‌فرض، کلید خارجی خصیصه‌های کلید اصلی رابطه مرجع را مورد ارجاع قرار می‌دهد
- مثال

create table *account*
(*account_number* **char**(10),
branch_name **char**(15),
balance **numeric**(12,2),
primary key (*account_number*),
foreign key (*branch_name*) **references** *branch*)

قیود صحت

- یک عمل حذف یا به‌روزرسانی بر روی رابطه مرجع ممکن است باعث تخطی از قید صحت ارجاعی شود
  - چندگانه‌ها در رابطه رجوع کننده باید به گونه‌ای تغییر داده شوند که قید صحت ارجاعی مجدداً برقرار شود
- مثال
  - تعریف یک قید صحت بر روی رابطه ***account***

create table *account*
(...
foreign key (*branch_name*) **references** *branch*
on delete cascade
on update cascade,
...)

قیود صحت

- عبارت **on delete cascade** با اعلان کلید خارجی مرتبط شده است
  - در صورتی که حذف یک چندگانه در رابطه ***branch*** باعث تخطی از قید صحت ارجاعی شود، سپس چندگانه‌هایی از رابطه ***account*** که شعبه حذف شده را مورد ارجاع قرار می‌دهند نیز حذف می‌شوند
- عبارت **on update cascade** با اعلان کلید خارجی مرتبط شده است
  - در صورتی که به‌روزرسانی یک چندگانه در رابطه ***branch*** باعث تخطی از قید صحت ارجاعی شود، سپس مقادیر خصیصه ***branch_name*** در چندگانه‌های رجوع کننده از رابطه ***account*** نیز به‌روزرسانی می‌شوند
  - از عبارت‌های **set null** و **set default** می‌توان پس از عبارت **cascade** استفاده کرد
  - مقادیر خصیصه‌های کلید خارجی می‌توانند تهی باشند

قیود صحت

- قید Not Null
  - مقدار تهی عضو همه دامنه‌ها است
  - مقدار تهی مقداری مجاز برای هر خصیصه در SQL است
  - برای برخی از خصیصه‌ها مقادیر تهی ممکن است نامناسب باشند
  - برای محدود کردن دامنه خصیصه‌ها به گونه‌ای که شامل مقادیر تهی نباشند، از قید **not null** استفاده می‌شود
    - مثال

account_number **char**(10) **not null**
balance **numeric**(12, 2) **not null**

قیود صحت

- عبارت Check
  - از عبارت **check** می‌توان هنگام اعلان یک رابطه استفاده کرد
    - شرطی را مشخص می‌کند که توسط هر چندگانه در رابطه باید برآورده شود
  - مثال
    - در رابطه ***branch*** خصیصه ***branch_name*** را به عنوان کلید اصلی تعریف کنید. همچنین اطمینان حاصل کنید که مقادیر خصیصه ***assets*** منفی نمی‌باشند

```
create table branch
    (branch_name    char(15),
     branch_city    char(30),
     assets         numeric(16,2),
     primary key (branch_name),
     check (assets >= 0))
```

قیود صحت

- مثال

create table *student*
(*name* **char**(15) **not null**,
student_id **char**(10),
degree_level **char**(15),
primary key (*student_id*),
check (*degree_level* **in** (‘Bachelors’, ‘Masters’, ‘Doctorate’)))

قیود صحت

- مثال
  - شِمای سیستم بانکی

customer **(*customer_name*, *customer_street*, *customer_city*)**
branch **(*branch_name*, *branch_city*, *assets*)**
loan **(*loan_number*, *branch_name*, *amount*)**
borrower **(*customer_name*, *loan_number*)**
account **(*account_number*, *branch_name*, *balance*)**
depositor **(*customer_name*, *account_number*)**

create table *customer*
(*customer_name* **char**(20),
customer_street **char**(30),
customer_city **char**(30),
primary key (*customer_name*))

قیود صحت

create table *branch*
(*branch_name* **char**(15),
branch_city **char**(30),
assets **numeric**(16,2),
primary key (*branch_name*),
check (*assets* >= 0))

create table *account*
(*account_number* **char**(10),
branch_name **char**(15),
balance **numeric**(12,2),
primary key (*account_number*),
foreign key (*branch_name*) **references** *branch*
on delete cascade
on update cascade,
check (*balance* >= 0))

قیود صحت

create table *depositor*
(*customer_name* **char**(20),
account_number **char**(10),
primary key (*customer_name*, *account_number*),
foreign key (*customer_name*) **references** *customer*
on delete cascade
on update cascade,
foreign key (*account_number*) **references** *account*
on delete cascade
on update cascade)

قیود صحت

- مثال
  - شِمای سیستم کارکنان

employee (*employee_name*, *street*, *city*)
works (*employee_name*, *company_name*, *salary*)
company (*company_name*, *city*)
manages (*employee_name*, *manager_name*)

- با استفاده از عبارت **check** قید صحتی بنویسید که اطمینان دهد
  - هر کارمند برای شرکتی کار می‌کند که در همان شهر محل سکونت کارمند مستقر است

قیود صحت

```
create table works
    (employee_name char(20),
     company_name  char(15),
     salary        int,
     primary key (employee_name, company_name),
     foreign key (employee_name) references employee,
     foreign key (company_name) references company,
     check ((employee_name, company_name) in
              (select employee_name, company_name
               from employee, company
               where employee.city = company.city)))
```