پایگاه داده ها

SQL (ادامه)

مجوزها

- به منظور انجام هر گونه عملیات بر روی یک پایگاه داده‌ها، هر کاربر (یا برنامه کاربردی) باید ابتدا به آن پایگاه داده‌ها متصل شود
- به هر کاربر ممکن است انواع مختلفی از مجوزها داده شود
  - مجوز برای خواندن داده‌ها
  - مجوز برای افزودن داده‌های جدید
  - مجوز برای به روز رسانی داده‌ها
  - مجوز برای حذف داده‌ها

مجوزها

- هر کدام از انواع مجوزها یک امتیاز (privilege) نامیده می‌شود
- انواع امتیازات در SQL
  - select
  - insert
  - update
  - delete
- به هر کاربری که یک رابطه جدید را ایجاد می‌کند، همه امتیازات بر روی آن رابطه به صورت خودکار داده می‌شود

مجوزها

- دستورات DDL برای واگذاری و لغو امتیازات
  - دستور grant

grant <privilege list>
on <relation name or view name> **to** <user list>

  - مثال

grant select on *account* **to** *John*, *Mary*
grant update(*amount*) **on** *loan* **to** *John*, *Mary*

  - دستور revoke

revoke <privilege list>
on <relation name or view name> **from** <user list>

  - مثال

revoke select on *account* **from** *John*, *Mary*
revoke update(*amount*) **on** *loan* **from** *John*, *Mary*

عملیات پیوند

- در SQL علاوه بر مکانیزم ضرب دکارتی، مکانیزم‌های دیگری برای پیوند رابطه‌ها با هم وجود دارد
- در هر عملیات پیوند باید شرط پیوند و نوع پیوند را مشخص کرد
  - استفاده از شرط پیوند برای پیوندهای بیرونی اجباری و برای پیوندهای درونی اختیاری است

Join types
inner join
left outer join
right outer join
full outer join

Join Conditions
natural
on <predicate>
using $(A_1, A_1, \ldots, A_n)$

عملیات پیوند

- از عملیات پیوند معمولاً به عنوان زیرپرس و جو در عبارت from استفاده می‌شود
  - مثال
    - رابطه‌های *loan* و *borrower* را در نظر بگیرید

loan_number	*branch_name*	*amount*
L-170	Downtown	3000
L-230	Redwood	4000
L-260	Perryridge	1700

loan

customer_name	*loan_number*
Jones	L-170
Smith	L-230
Hayes	L-155

borrower

عملیات پیوند

select *

from (*loan* **inner join** *borrower*

on *loan.loan_number = borrower.loan_number*)

loan_number	*branch_name*	*amount*	*customer_name*	*loan_number*
L-170	Downtown	3000	Jones	L-170
L-230	Redwood	4000	Smith	L-230

select *

from (*loan* **left outer join** *borrower*

on *loan.loan_number = borrower.loan_number*)

loan_number	*branch_name*	*amount*	*customer_name*	*loan_number*
L-170	Downtown	3000	Jones	L-170
L-230	Redwood	4000	Smith	L-230
L-260	Perryridge	1700	*null*	*null*

عملیات پیوند

select *

from (*loan* **natural inner join** *borrower*)

loan_number	*branch_name*	*amount*	*customer_name*
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith

select *

from (*loan* **natural right outer join** *borrower*)

loan_number	*branch_name*	*amount*	*customer_name*
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-155	*null*	*null*	Hayes

عملیات پیوند

select *

from (*loan* **full outer join** *borrower*
using (*loan_number*))

loan_number	*branch_name*	*amount*	*customer_name*
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-260	Perryridge	1700	*null*
L-155	*null*	*null*	Hayes

عملیات پیوند

- مثال
  - نام‌های مشتریانی را فهرست کنید که حساب بانکی دارند، اما هیچ گونه وامی از بانک دریافت نکرده‌اند

select distinct *depositor.customer_name*
from (*depositor* **left outer join** *borrower*
on *depositor.customer_name* = *borrower.customer_name*)
where *borrower.customer_name* **is** *null*

توابع و رویه‌ها

- در SQL امکان تعریف توابع و رویه‌ها وجود دارد
- مثال
  - تابعی تعریف کنید که با فرض نام هر مشتری تعداد حساب‌های بانکی متعلق به آن مشتری را برگرداند

```
create function account_count(c_name varchar(20))
returns integer
begin
    declare a_count integer;
    select count(*) into a_count
    from depositor
    where depositor.customer_name = c_name;
    return a_count;
end
```

توابع و رویه‌ها

- **مثال**
- **نام‌ها و آدرس‌های مشتریانی را فهرست کنید که بیش از یک حساب بانکی دارند**

select *customer_name*, *customer_street*, *customer_city*
from *customer*
where *account_count* (*customer_name*) > 1

- **شکل کلی توابع**

create function <name> (<parameter list>) **returns** <type>
<body>

توابع و رویه‌ها

- **شکل کلی رویه‌ها**

create procedure <name> (<parameter list>)
<body>

- **مثال**
  - **رویه‌ای تعریف کنید که با فرض نام هر مشتری تعداد حساب‌های بانکی متعلق به آن مشتری را برگرداند**

```
create procedure account_count_proc (in c_name varchar(20),
                                     out a_count integer)
begin
    select count (*) into a_count
    from depositor
    where depositor.customer_name = c_name
end
```

توابع و رویه‌ها

- در SQL هر رویه را می‌توان با استفاده از دستور **call** از داخل رویه دیگر فراخوانی کرد
  - مثال

declare *a_count* **integer**;
call *account_count_proc* (‘Smith’, *a_count*);

توابع و رویه‌ها

- انواع دستورات
  - اعلان متغیرهای محلی

declare <name> <type>;

- مثال

declare *a_count* **integer**;

  - دستورات انتساب

set <variable> = <expression>;

- مثال

set *balance* = *balance* – *amount*;

  - دستورات مرکب

begin
...
end

توابع و رویه‌ها

- حلقه‌های while و repeat
  - مثال

```
declare n integer default 0;
while n < 10 do
    set n = n + 1;
end while
repeat
    set n = n – 1;
until n = 0
end repeat
```

توابع و رویه‌ها

- حلقه for
  - امکان تکرار بر روی همه نتایج یک پرس و جو را فراهم می‌کند
  - مثال
    - مجموع موجودی همه حساب‌های بانکی در شعبه Perryridge را مشخص کنید

```
declare n integer default 0;
for r as
    select balance from account
    where branch_name = 'Perryridge'
do
    set n = n + r.balance
end for
```

توابع و رویه‌ها

- دستورات شرطی
  - مثال

```
if r.balance < 1000 then
    set l = l + r.balance
elseif r.balance < 5000 then
    set m = m + r.balance
else
    set h = h + r.balance
end if
```

توابع و رویه‌ها

- مثال

```
create procedure withdraw (in account_number varchar(10),
                           in amount numeric(12,2));
begin
    declare newbalance numeric(12,2);
    select balance into newbalance
    from account
    where account.account_number = withdraw.account_number;
    set newbalance = newbalance – amount;
        ...
    update account
        set balance = newbalance
        where account.account_number = withdraw.account_number
end
```