

پایگاه داده ها

SQL (ادامه)

زیرپرس و جویهای تودرتو



- زیرپرس و جو
- عبارتی به شکل `select-from-where` که داخل زیرپرس و جوی دیگر قرار می‌گیرد
- SQL امکان بررسی عضویت چندگانه‌ها در یک رابطه را فراهم می‌کند
- از رابط `in` برای بررسی عضویت در مجموعه‌ای از مقادیر تولید شده توسط عبارت `select` استفاده می‌شود

زیرپرس و جوهای تودرتو



• مثال

• نامهای مشتریانی را فهرست کنید که هم حساب بانکی و هم وام بانکی دارند

```
(select customer_name  
  from depositor)
```

intersect

```
(select customer_name  
  from borrower)
```

```
select distinct customer_name
```

```
from borrower
```

```
where customer_name in (select customer_name  
                           from depositor)
```

زیرپرس و جوهای تودرتو



• مثال

- نامهای مشتریانی را فهرست کنید که وام بانکی دریافت کردهاند، اما حساب بانکی ندارند

```
select distinct customer_name  
from borrower  
where customer_name not in (select customer_name  
                                from depositor)
```

- نامهای مشتریانی را فهرست کنید که وام بانکی دریافت کردهاند، اما نامشان Smith یا Jones نمیباشد

```
select distinct customer_name  
from borrower  
where customer_name not in ('Smith', 'Jones')
```

زیرپرس و جوهای تودرتو



- در SQL عبارت "بزرگتر از حداقل یکی از" با `> some` نمایش داده می شود

- مثال

- نامهای شعبه‌هایی را فهرست کنید که سرمایه آنها از سرمایه حداقل یکی از شعبه‌های مستقر در شهر Brooklyn بیشتر است

```
select distinct T.branch_name  
from branch as T, branch as S  
where T.assets > S.assets and S.branch_city = 'Brooklyn'
```

```
select branch_name  
from branch  
where assets > some (select assets  
                    from branch  
                    where branch_city = 'Brooklyn')
```

زیرپرس و جوهای تودرتو



- در SQL امکان انجام مقایسه با استفاده از `<some`، `<=some`، `>some`، `>=some` و `<>some` وجود دارد

• مثال

```
select branch_name
from branch
where assets <> some (select assets
                        from branch
                        where branch_city = 'Brooklyn')
```

• تمرین

- با استفاده از یک مثال نشان دهید که `=some` با `in` یکسان است، اما `<>some` با `not in` یکسان نیست

زیرپرس و جوهای تودرتو



- در SQL عبارت "بزرگتر از همه" با `all >` نمایش داده می‌شود

- مثال

- نام‌های شعبه‌هایی را فهرست کنید که سرمایه آن‌ها از سرمایه هر کدام از شعبه‌های مستقر در شهر Brooklyn بیشتر است

```
select branch_name  
from branch  
where assets > all (select assets  
                        from branch  
                        where branch_city = 'Brooklyn')
```

زیرپرس و جوهای تودرتو



- در SQL امکان انجام مقایسه با استفاده از `<all`، `<=all`، `>all`، `>=all`، `=all` و `<>all` وجود دارد
- تمرین
- با استفاده از یک مثال نشان دهید که `<>all` با `not in` یکسان است

زیرپرس و جوهای تودرتو



- در SQL امکان ترکیب توابع تجمیع وجود ندارد
- از `max (avg (...))` نمی توان استفاده کرد
- مثال

- نام شعبه‌ای که متوسط موجودی حساب‌های آن از سایر شعبه‌ها بیشتر است را پیدا کنید

```
select branch_name  
from account  
group by (branch_name)  
having avg (balance) >= all (select avg (balance)  
                                from account  
                                group by branch_name)
```

زیرپرس و جوهای تودرتو



- در SQL با استفاده از عبارت `exists` می توان وجود یا عدم وجود چندگانه ها در نتیجه یک زیرپرس و جو را بررسی کرد
- عبارت `exists` در صورتی مقدار `true` برمی گرداند که زیرپرس و جوی مورد نظر غیرتهی باشد
- مثال

• نام های مشتریانی را فهرست کنید که هم حساب بانکی و هم وام بانکی دارند

```
select distinct customer_name  
from borrower  
where exists (select *  
                from depositor  
                where depositor.customer_name = borrower.customer_name)
```

زیرپرس و جوهای تودرتو



- برای بیان این واقعیت که رابطه A رابطه B را در برمی گیرد از $\text{not exists (B except A)}$ استفاده می شود

$$B - A = \emptyset \Leftrightarrow B \subseteq A \quad \bullet$$

تمرین •

- نامهای مشتریانی را فهرست کنید که در همه شعبه های مستقر در شهر **Brooklyn** حساب بانکی دارند

زیرپرس و جوهای تودرتو



- در SQL با استفاده از عبارت **unique** می‌توان وجود یا عدم وجود چندگانه‌های تکراری در نتیجه یک زیرپرس و جو را بررسی کرد
- عبارت **unique** در صورتی مقدار **true** برمی‌گرداند که زیرپرس و جوی مورد نظر شامل چندگانه‌های تکراری نباشد
- مثال
- نام‌های مشتریانی را فهرست کنید که تنها یک حساب بانکی در شعبه **Perryridge** دارند

```
select T.customer_name
from depositor as T
where unique (select R.customer_name
              from account, depositor as R
              where T.customer_name = R.customer_name and
                    R.account_number = account.account_number and
                    account.branch_name = 'Perryridge')
```

روابط مشتق شده



- SQL اجازه می‌دهد تا یک زیرپرس و جو در عبارت **from** استفاده

شود

- مثال

- زیرپرس و جوی زیر را در نظر بگیرید

```
(select branch_name, avg(balance)  
from account  
group by branch_name)  
as branch_avg(branch_name, avg_balance)
```

- رابطه حاصل از زیرپرس و جو به نام *branch_avg* نام‌گذاری شده است

- خصیصه‌های *branch_name* و *avg_balance*

روابط مشتق شده



- اکنون پرس و جوی زیر را در نظر بگیرید

```
select branch_name, avg_balance
from (select branch_name, avg(balance)
from account
group by branch_name)
as branch_avg(branch_name, avg_balance)
where avg_balance > 1200
```

- پرس و جوی فوق با استفاده از عبارت **having**

```
select branch_name, avg(balance)
from account
group by branch_name
having avg(balance) > 1200
```

مقادیر تهی



- مقادیر برخی از خصیصه‌ها ممکن است تهی باشد
- بررسی مقادیر تهی
 - `is null`
 - مثال

```
select loan_number  
from loan  
where amount is null
```

```
select loan_number  
from loan  
where amount is not null
```

دیدها



- گاهی اوقات لازم است کاربران نتوانند تمام مدل منطقی را مشاهده کنند
- لازم است داده‌های خاصی از کاربران مخفی شود
- مثال
- فرض کنید یک کاربر خاص لازم است تا از نام شعبه و شماره وام مشتریان اطلاع داشته باشد، اما نباید میزان وام آن‌ها را مشاهده کند

```
select customer_name, borrower.loan_number, branch_name  
from borrower, loan  
where borrower.loan_number = loan.loan_number
```


دیدها



- دید (View)
- یک رابطه که بخشی از مدل منطقی نیست، اما برای برخی از کاربران به عنوان یک رابطه مجازی قابل رؤیت است
- مکانیزمی را برای مخفی سازی برخی داده ها از دید برخی کاربران فراهم می کند

• تعریف یک دید

create view v as <query expression>

- v نام دید

دیدها

• مثال

- یک دید شامل نام‌های شعبه‌ها و نام‌های مشتریان آن‌ها

```
create view all_customer as  
  (select branch_name, customer_name  
   from depositor, account  
   where depositor.account_number = account.account_number)  
union  
  (select branch_name, customer_name  
   from borrower, loan  
   where borrower.loan_number = loan.loan_number)
```

- نام‌های مشتریان شعبه Perryridge را فهرست کنید

```
select customer_name  
from all_customer  
where branch_name = 'Perryridge'
```

دیده‌ها



- نام‌های خصیصه‌های یک دید را می‌توان صریحاً مشخص کرد

• مثال

```
create view branch_total_loan(branch_name, total_loan) as  
select branch_name, sum(amount)  
from loan  
group by branch_name
```

- دیده‌های تعریف شده با استفاده از سایر دیده‌ها

• مثال

```
create view perryridge_customer as  
select customer_name  
from all_customer  
where branch_name = 'Perryridge'
```