



# 第4节 子查询

讲师：李玉婷

# 目 标

通过本章学习，您将可以：

- 描述子查询可以解决的问题。
- 定义子查询。
- 列出子查询的类型。
- 书写单行子查询和多行子查询。

**概念：** 出现在其他语句内部的select语句，称为子查询或内查询

内部嵌套其他select语句的查询，称为外查询或主查询

**示例：**

```
select first_name from employees where  
    department_id in(  
        select department_id from departments  
        where location_id=1700  
    )
```

# 注意事项

- 子查询要包含在**括号内**。

注意是(), 而不是{}

- 将子查询放在比较条件的右侧。

- 单行操作符对应单行子查询, 多行操作符对应多行子查询。**

子查询分类

按照子查询的结果集行数分类:

标量子查询 (一行一列)

列子查询 (一列多行)

行子查询 (一行多列)

表子查询 (多行多列)

子查询的位置:

select后面, 支持标量子查询;

from后面, 支持表子查询, 必须给表起别名

where/having后面, 支持标量, 行, 列子查询;

exists后面: 相关子查询, 支持表子查询, 返回0/1, 0表示不存在, 1表示存在, 可用IN 代替



## 子查询类型

- 单行子查询



- 多行子查询



# 单行子查询

- 只返回一行。
- 使用单行比较操作符。

操作符	含义
=	Equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
<>	Not equal to

## 子查询语法

```
SELECT    select_list  
FROM      table  
WHERE     expr operator
```

```
(SELECT      select_list  
FROM         table);
```

- 子查询（内查询）在主查询之前一次执行完成。
- 子查询的结果被主查询（外查询）使用。

# 使用子查询解决问题

谁的工资比 Abel 高？

Main Query:



谁的工资比 Abel 高？

Subquery



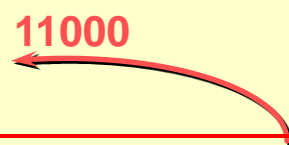
Abel的工资是多少？





## 子查询

```
SELECT last_name
FROM employees 11000
WHERE salary >
    (SELECT salary
     FROM employees
     WHERE last_name = 'Abel');
```



LAST_NAME
King
Kochhar
De Haan
Hartstein
Higgins

## 执行单行子查询

题目：返回`job_id`与141号员工相同，`salary`比143号员工多的员工姓名，`job_id` 和工资

LAST_NAME	JOB_ID	SALARY
Rajs	ST_CLERK	3500
Davies	ST_CLERK	3100

## 在子查询中使用组函数

题目：返回公司工资最少的员工的last\_name,job\_id和salary

LAST_NAME	JOB_ID	SALARY
Vargas	ST_CLERK	2500

## 子查询中的 HAVING 子句

- 首先执行子查询。
- 向主查询中的HAVING 子句返回结果。

题目：查询最低工资大于**50**号部门最低工资的部门**id**和其最低工资

## 非法使用子查询

```
SELECT employee_id, last_name
FROM employees
WHERE salary =
    (SELECT MIN(salary)
     FROM employees
     GROUP BY department_id);
```

```
ERROR at line 4:
ORA-01427: single-row subquery returns more than
one row
```

多行子查询使用单行比较符



## 子查询中的空值问题

```
SELECT last_name, job_id
FROM employees
WHERE job_id =
      (SELECT job_id
       FROM employees
       WHERE last_name = 'Haas');
```

no rows selected

子查询不返回任何行

## 多行子查询

- 返回多行。
- 使用多行比较操作符。

操作符	含义
IN/NOT IN	等于列表中的任意一个
ANY   SOME	和子查询返回的某一个值比较 即只要能够满足其中一个即可
ALL	和子查询返回的所有值比较 必须满足所有的

➤ 体会any和all的区别

题目：返回**location\_id**是**1400**或**1700**的部门中的所有员工姓名

分页查询，应用在分页：

limit子句：

索引从0开始，后面加上要显示的记录数，如

limit 0, 5 表示从0开始，显示5条记录

在select子句的末尾，执行也是在最后执行

select子句执行顺序：

1. from
2. join
3. on
4. where
5. group by
6. having 生成虚拟表
7. select
8. order by
9. limit

union子句，合并多个查询语句为一个结果

应用在查询多个表时，每个表之间没有连接关系，且查询的字段与列数都是一致的，才可以使用union子句。

同时，保证每一列的字段和类型是一致的，union关键字是去重的，使用union all显示所有的记录

```
select ...  
union  
select ...
```

# 在多行子查询中使用 ANY 操作符

题目：返回其它部门中比job\_id为‘IT\_PROG’部门任一工资低的员工的员工号、姓名、job\_id 以及salary

```
SELECT employee_id, last_name, job_id, salary
FROM employees 9000, 6000, 4800, 4200
WHERE salary < ANY
    (SELECT salary
     FROM employees
     WHERE job_id = 'IT_PROG')
AND job_id <> 'IT_PROG';
```

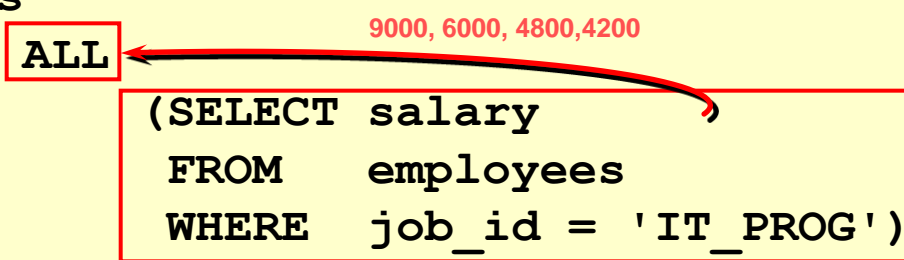
EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
124	Mourgos	ST_MAN	5800
141	Rajs	ST_CLERK	3500
142	Davies	ST_CLERK	3100
143	Matos	ST_CLERK	2600
144	Vargas	ST_CLERK	2500

...  
10 rows selected.

# 在多行子查询中使用 ALL 操作符

题目：返回其它部门中比job\_id为‘IT\_PROG’部门**所有**工资都低的员工的员工号、姓名、job\_id 以及salary

```
SELECT employee_id, last_name, job_id, salary
FROM employees
WHERE salary < ALL
      (SELECT salary
       FROM employees
       WHERE job_id = 'IT_PROG')
AND job_id <> 'IT_PROG';
```



EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
141	Rajs	ST_CLERK	3500
142	Davies	ST_CLERK	3100
143	Matos	ST_CLERK	2600
144	Vargas	ST_CLERK	2500



## 子查询中的空值问题

```
SELECT emp.last_name  
FROM   employees emp  
WHERE  emp.employee_id NOT IN  
                                (SELECT mgr.manager_id  
                                FROM    employees mgr);
```

no rows selected

# 总 结

通过本章学习，您已经学会：

- 如何使用子查询。
- 在查询时基于未知的值时，应使用子查询。

```
SELECT    select_list
FROM      table
WHERE     expr operator
          (SELECT select_list
           FROM    table);
```

