

中国科学院西安网络中心 编译 2005

ORACLE

Copyright © Oracle Corporation, 2001. All rights reserved.

| | | |
|------|-------|----|
| 进度表: | 时间 | 主题 |
| | 25 分钟 | 讲演 |
| | 30 分钟 | 练习 |
| | 55 分钟 | 总共 |

目标

完成本课后，您应当能够执行下列操作：

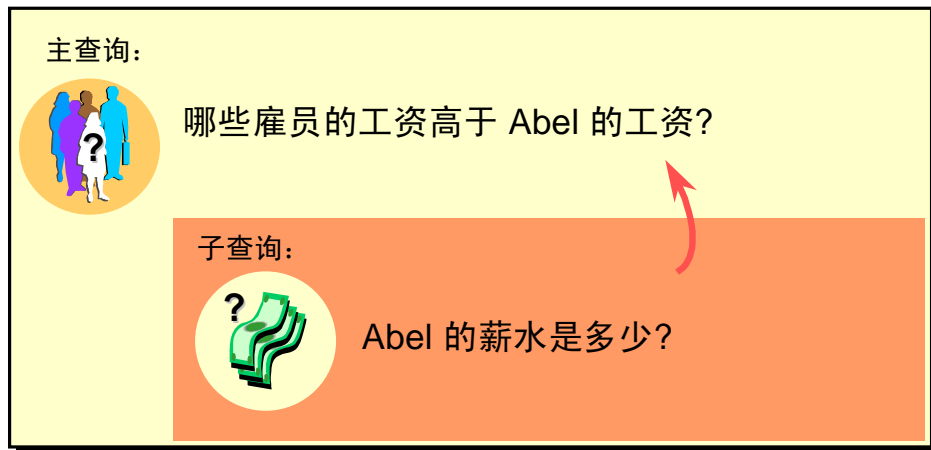
- 描述子查询能够解决的问题类型
- 定义子查询
- 列出子查询的类型
- 写单行和多行子查询

课程目标

在本课中，你将学习关于 **SELECT** 语句的更多的高级特性。你可以在另一个 **SQL** 语句的 **WHERE** 子句中写子查询以获得基于未知条件值。本课包括单行子查询和多行子查询。

用子查询解决问题

谁的薪水比 Abel's 多？



用子查询解决问题

假想你想要写一个查询来找出挣钱比 Abel 的薪水还多的人。为了解决这个问题，你需要两个查询：一个找出 Abel 的收入，第二个查询找出收入高于 Abel 的人。

你可以用组合两个查询的方法解决这个问题，放置一个查询到另一个查询中。

内查询或子查询返回一个值给外查询或主查询。使用一个子查询相当于执行两个连续查询并且用第一个查询的结果作为第二个查询的搜索值。

子查询语法

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

- 子查询（内查询）在主查询之前执行一次
- 子查询的结果被用于主查询（外查询）

子查询

子查询是一个 SELECT 语句，它是嵌在另一个 SELECT 语句中的子句。使用子查询你可以用简单的语句构建功能强大的语句。当你需要从表中用依赖于表本身的数据选择行时它们是非常有用的。

你可以将子查询放在许多的 SQL 子句中，包括：

- WHERE 子句
- HAVING 子句
- FROM 子句

在语法中：

operator 包括比较条件，例如 >、= 或 IN

注：比较条件分为两个种类：单行运算符 (>, =, >=, <, <>, <=) 和多行运算符 (IN, ANY, ALL)。

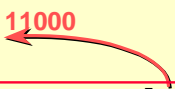
子查询通常涉及一个嵌套的 SELECT、子-SELECT 或内 SELECT 语句。子查询通常执行一次。并且它的输出被用于完成主或外查询的查询条件。

教师注释

另外，子查询可以被放在 CREATE VIEW 语句中、CREATE TABLE 语句、UPDATE 语句、INSERT 语句的 INTO 子句和 UPDATE 语句的 SET 子句中。

使用子查询

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



| LAST_NAME |
|-----------|
| King |
| Kochhar |
| De Haan |
| Hartstein |
| Higgins |

使用子查询

在幻灯片中，内查询确定雇员 Abel 的薪水，外查询取得内查询的结果，并且使用该结果显示所有薪水高于该数量的雇员。

教师注释

首先执行子查询（内查询）显示子查询返回的值，然后用内查询返回的结果执行外查询，最后，执行整个查询（包括子查询），显示相同的结果。

使用子查询的原则

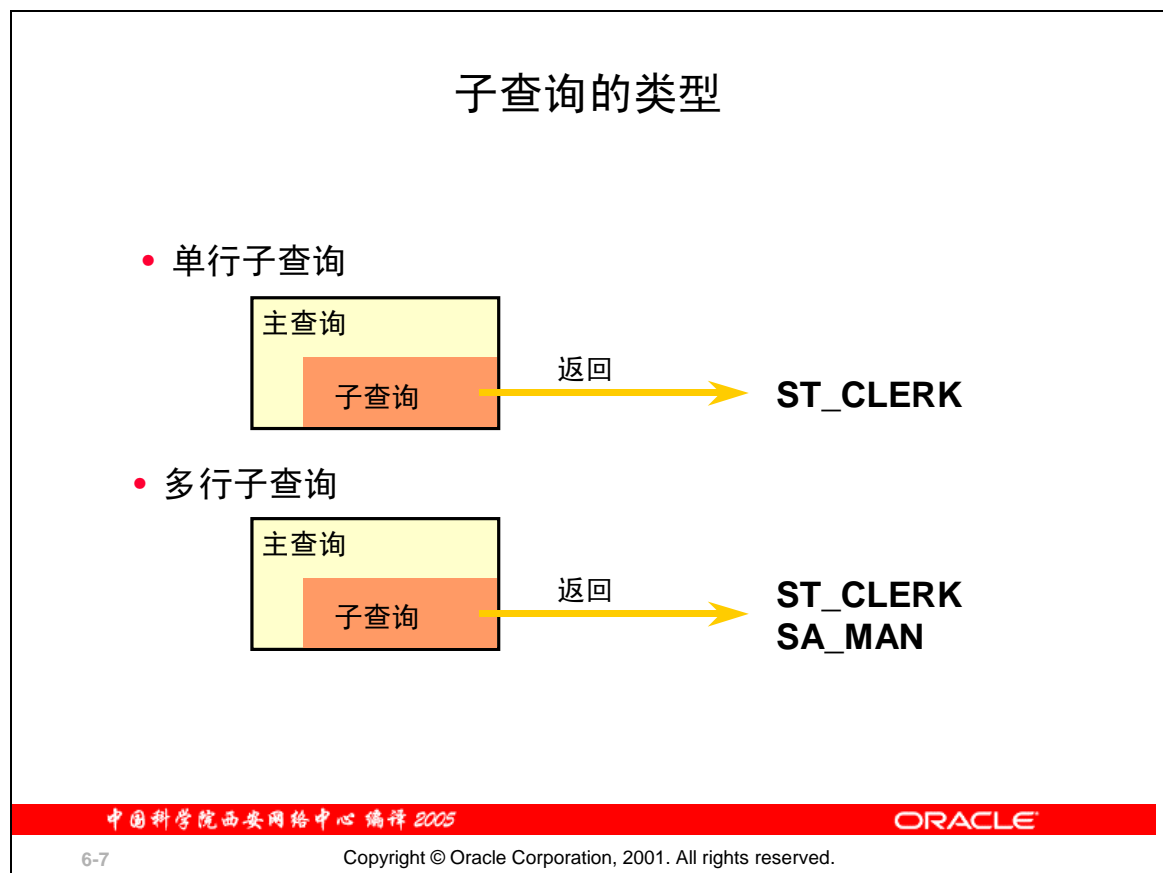
- 子查询放在圆括号中
- 将子查询放在比较条件的右边
- 在子查询中的 ORDER BY 子句不需要，除非你正在执行 Top-N 分析
- 在单行子查询中用单行运算符，在多行子查询中用多行运算符

使用子查询的原则

- 一个子查询必须放在圆括号中。
- 将子查询放在比较条件的右边以增加可读性。
- Oracle8i 以前的版本中，子查询不包含 ORDER BY 子句。对一个 SELECT 语句只能用一个 ORDER BY 子句，并且如果指定了它就必须放在主 SELECT 语句的最后。从 Oracle8i 开始，ORDER BY 子句可以使用，并且在进行 Top-N 分析时是必须的。
- 在子查询中可以使用两种比较条件：单行运算符和多行运算符。

教师注释

一个子查询可以在有相互关系的子查询中被多次执行 A。学生可能会问可以写多少子查询，Oracle 服务器没有强制限制子查询的数目；限制只与查询所需的缓冲区大小有关。



子查询的类型

- 单行子查询：从内 SELECT 语句只返回一行的查询
- 多行子查询：从内 SELECT 语句返回多行的查询

注：还有多列子查询：从内 SELECT 语句返回多列的查询。

教师注释

多列子查询也是可用的。

单行子查询

- 仅返回一行
- 使用单行比较符

| 运算符 | 含义 |
|-----|-------|
| = | 等于 |
| > | 大于 |
| >= | 大于或等于 |
| < | 小于 |
| <= | 小于或等于 |
| <> | 不等于 |

中国科学院西安网络中心 编译 2005

ORACLE

6-8

Copyright © Oracle Corporation, 2001. All rights reserved.

单行子查询

单行子查询是从内查询返回一行的查询。在该子查询类型中用一个单行操作符。幻灯片中列出了单行操作符。

例

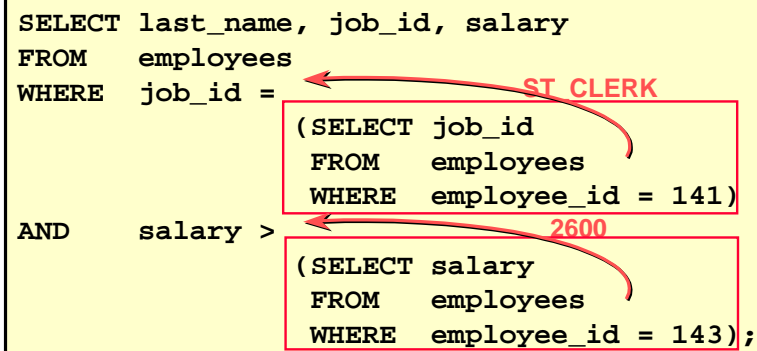
显示那些 job ID 与雇员 141 相同的雇员。

```
SELECT last_name, job_id
FROM employees
WHERE job_id =
      (SELECT job_id
       FROM employees
       WHERE employee_id = 141);
```

| LAST_NAME | JOB_ID |
|-----------|----------|
| Rajs | ST_CLERK |
| Davies | ST_CLERK |
| Matos | ST_CLERK |
| Vargas | ST_CLERK |

执行单行子查询

```
SELECT last_name, job_id, salary
FROM   employees
WHERE  job_id =
      (SELECT job_id
       FROM   employees
       WHERE  employee_id = 141)
AND    salary >
      (SELECT salary
       FROM   employees
       WHERE  employee_id = 143);
```



| LAST_NAME | JOB_ID | SALARY |
|-----------|----------|--------|
| Rajs | ST_CLERK | 3500 |
| Davies | ST_CLERK | 3100 |

执行单行子查询

SELECT 语句可以被考虑为一个查询块。幻灯片中的例子显示 job ID 与雇员 141 相同，并且薪水 高于雇员 143 的那些雇员。

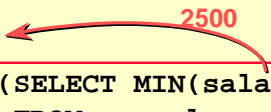
该例子可以由三个查询块组成：外查询和两个内查询。内查询块首先被执行，产生查询结果分别为 ST_CLERK 和 2600。然后处理外查询块，并且使用内查询的返回值来完成它的查询条件。

两个内查询返回单个值（分别是 ST_CLERK 和 2600），所以这种 SQL 语句被称为单行子查询。

注：外和内查询可以从不同的表中取得数据。

在子查询中使用组函数

```
SELECT last_name, job_id, salary
FROM   employees
WHERE  salary = (SELECT MIN(salary)
                  FROM   employees);
```



| LAST_NAME | JOB_ID | SALARY |
|-----------|----------|--------|
| Vargas | ST_CLERK | 2500 |

在子查询中使用组函数

你可以从主查询中显示数据，该主查询使用一个带组函数的单行子查询。子查询放在圆括号中并且放在比较条件的后面。

幻灯片中的例子显示所有其薪水等于最低薪水的雇员的 last name、job ID 和 salary。MIN 组函数返回单个的值 (2500) 给外函数。

带子查询的 HAVING 子句

- Oracle 服务器首先执行子查询
- Oracle 服务器返回结果到主查询的 HAVING 子句中

```

SELECT  department_id, MIN(salary)
FROM    employees
GROUP BY department_id
HAVING  MIN(salary) >
        (SELECT MIN(salary)
         FROM    employees
         WHERE   department_id = 50);

```

中国科学院西安网络中心 编译 2005

ORACLE

6-11

Copyright © Oracle Corporation, 2001. All rights reserved.

带子查询的 HAVING 子句

你不仅可以在 WHERE 子句中使用子查询，也可以在 HAVING 子句中使用子查询。Oracle 服务器执行子查询，并将结果返回到主查询的 HAVING 子句中。在幻灯片中的 SQL 语句显示所有其最低薪水小于 部门 50 的最低薪水的部门号和最低薪水。

| DEPARTMENT_ID | MIN(SALARY) |
|---------------|-------------|
| 10 | 4400 |
| 20 | 6000 |
| | 7000 |

7 rows selected.

例

找出平均薪水为最低平均薪水的工作岗位。

```

SELECT  job_id, AVG(salary)
FROM    employees
GROUP BY job_id
HAVING  AVG(salary) = (SELECT  MIN(AVG(salary))
                       FROM    employees
                       GROUP BY job_id);

```

什么是子查询错误？

```
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
```

单行运算符用于多行子查询

中国科学院西安网络中心 编译 2005

ORACLE

6-12

Copyright © Oracle Corporation, 2001. All rights reserved.

子查询错误

使用子查询的一个常见的错误是单行子查询返回返回了多行。

在幻灯片的 SQL 语句中，子查询包含一个 GROUP BY 子句，这就暗示该子查询将返回多行，每个对应它所找到的一组，在这种情况下，子查询的结果将是 4400、6000、2500、4200、7000、17000 和 8300。

外查询得到子查询的结果 (4400、6000、2500、4200、7000、17000、8300) 并且在它的 WHERE 子句中使用这些结果。WHERE 子句包含一个等号 (=) 运算符，这是一个单行比较运算符，只能使用一个值。= 操作符不能接受来自子查询的多个值，并且因此产生错误。

为了纠正该错误，改变 = 操作为 IN。

这个子查询是否有返回行？

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

```
no rows selected
```

子查询无返回值

子查询问题

子查询的另一个常见问题是内查询没有返回行。

在幻灯片的 SQL 语句中，子查询包含一个 WHERE 子句，推测起来，其目的是找名字为 Haas 的雇员，该语句是正确的，但在执行时选择无行返回。

没有名叫 Haas 的雇员，所以子查询无返回行，外查询得到子查询的结果 (null) 并且在 WHERE 子句中使用该结果，外查询找不到一个 job ID 等于 null 的雇员，所以也没有行返回。如果一个 job 存在 null 值，也没有返回行，因为比较两个空值还是空，因此 WHERE 子句的条件不为 true。

多行子查询

- 返回多于一行
- 使用多行比较符

| 操作 | 含义 |
|------------|-------------|
| IN | 等于列表中的任何成员 |
| ANY | 比较子查询返回的每个值 |
| ALL | 比较子查询返回的全部值 |

多行子查询

子查询返回多行被称为多行子查询。对多行子查询要使用多行运算符而不是单行运算符。多行运算符期待多个值。

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

例

查找各部门收入为部门最低的那些雇员。

内查询先被执行，产生一个查询结果，然后主查询块处理和使用由内查询返回的值完成它的搜索条件。事实上，在 Oracle 服务器看起来主查询象是下面这样：

```
SELECT last_name, salary, department_id
FROM   employees
WHERE  salary IN (2500, 4200, 4400, 6000, 7000, 8300, 8600,
17000);
```

在多行子查询中使用 ANY 运算符

```

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

```

9000, 6000, 4200

| 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.

中国科学院西安网络中心 编译 2005

ORACLE

6-15

Copyright © Oracle Corporation, 2001. All rights reserved.

多行子查询 (续)

ANY 运算符 (和它的同义词, SOME 运算符) 比较一个值与一个子查询返回的每一个值。幻灯片中的例子显示不是 IT 程序员的雇员, 并且这些雇员的薪水少于 IT 程序员。挣钱最多的程序员的薪水是 \$9,000。

<ANY 意思是小于最大值。 >ANY 意思是大于最小值。 =ANY 等同于 IN。

<ALL 意思是小于最小值。 >ALL 意思是大于最大值。

教师注释

当使用 SOME 或者 ANY 时, 通常用 DISTINCT 关键字防止返回被多次选择的行。

在多行子查询中使用 ALL 运算符

```

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';

```

9000, 6000, 4200

| 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 |

多行子查询 (续)

ALL 运算符比较一个值与子查询返回的每个值。幻灯片的例子显示那些薪水低于工作岗位 IT_PROG 的最低薪水，并且工作岗位不是 IT_PROG 的所有雇员。

>ALL 意思是大于最大值，<ALL 意思是小于最小值。

NOT 运算符可以与 IN、ANY 和 ALL 运算符一起使用。

子查询中的空值

```
SELECT emp.last_name  
FROM   employees emp  
WHERE  emp.employee_id NOT IN  
  
      (SELECT mgr.manager_id  
       FROM   employees mgr);  
  
no rows selected
```

在子查询的结果集中返回空值

在幻灯片中的 SQL 语句显示所有没有下属的雇员 (即不是经理的雇员)。在逻辑上, 该 SQL 语句应该返回 12 行, 但实际上没有返回任何行。因为内查询返回的值之一是空值, 并因此整个查询无返回行, 原因是比较一个空值的所有条件结果还是空。所以, 只要空值可能是子查询结果集的一部分, 就不能用 NOT IN 运算符。NOT IN 运算符相当于 \neq ALL。

注意到空值作为一个子查询结果集的一部分，如果你使用 `IN` 操作符的话，不是一个问题。`IN` 操作符相当于 `=ANY`。例如为了显示有下属的雇员（即经理）使用下面的 `SQL` 语句：

```
SELECT emp.last_name
FROM   employees emp
WHERE  emp.employee_id IN
      (SELECT mgr.manager_id
       FROM   employees mgr);
```

作为选择，**WHERE** 子句也可以被指定在子查询中，以显示所有没有下属的雇员：

```
SELECT last_name FROM employees
WHERE employee_id NOT IN
            (SELECT manager_id
             FROM employees
             WHERE manager_id IS NOT NULL);
```

小结

在本课中，您应该已经学会如何：

- 确定什么时候子查询能够帮助解决问题
- 当一个查询基于未知值时写子查询

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

小结

在本课中，你应该已经学会了怎样使用子查询。子查询是一个被嵌在另一个 SQL 语句中的 SELECT 语句。当一个查询基于一个带有未知中间值的查找标准时，子查询是有用的。

子查询有下面的特性：

- 能传递一个数据行到包含一个单行运算符的主语句，例如 =, <>, >, >=, <, or <=
- 能传递多行数据到包含一个多行运算符的主语句，例如 IN
- 首先被 Oracle 服务器处理，并且由 WHERE 或 HAVING 子句使用其结果
- 可以包含组函数

练习 6 概览

本章练习包括下面的主题：

- 创建子查询以查询基于未知标准的值
- 使用子查询以找出那些值在一个数据集中而不在另一个数据集中

练习 6

在本练习中，你将用嵌套的 `SELECT` 语句写复合查询。

基于卷面的问题

你可能想创建内查询，前三个问题是相关的，在编码外查询之前确定查询能工作并且产生你预期的数据。

练习 6

1. 写一个查询显示与 Zlotkey 在同一部门的雇员的 last name 和 hire date, 结果中不包括 Zlotkey。

| LAST_NAME | HIRE_DATE |
|-----------|-----------|
| Abel | 11-MAY-96 |
| Taylor | 24-MAR-98 |

```
SELECT last_name, hire_date
FROM employees
WHERE department_id = (SELECT department_id
FROM employees
WHERE last_name = 'Zlotkey')
AND last_name <> 'Zlotkey';
```

2. 创建一个查询显示所有其薪水高于平均薪水的雇员的雇员号和名字。按薪水的升序排序。

| EMPLOYEE_ID | LAST_NAME | SALARY |
|-------------|-----------|--------|
| 103 | Hunold | 9000 |
| 149 | Zlotkey | 10500 |
| 174 | Abel | 11000 |
| 205 | Higgins | 12000 |
| 201 | Hartstein | 13000 |
| 101 | Kochhar | 17000 |
| 102 | De Haan | 17000 |
| 100 | King | 24000 |

8 rows selected.

```
SELECT employee_id, last_name
FROM employees
WHERE salary > (SELECT AVG(salary)
FROM employees);
```

3. 写一个查询显示所有工作在任一雇员的的名字中包含一个 u 的部门的雇员的雇员号和名字。保存 SQL 语句到一个名为 lab6_3.sql 的文本文件中, 运行该查询。

| EMPLOYEE_ID | LAST_NAME |
|-------------|-----------|
| 124 | Mourgos |
| 141 | Rajs |
| 142 | Davies |
| 143 | Matos |
| 144 | Vargas |
| 103 | Hunold |
| 104 | Ernst |
| 107 | Lorentz |

8 rows selected.

```
SELECT employee_id, last_name
FROM employees
WHERE department_id IN (SELECT department_id
FROM employees
WHERE last_name like '%u%');
```

4. 显示所有部门地点号 (department location ID) 是 1700 的雇员的 last name、department number 和 job ID。

| LAST_NAME | DEPARTMENT_ID | JOB_ID |
|-----------|---------------|------------|
| Whalen | 10 | AD_ASST |
| King | 90 | AD PRES |
| Kochhar | 90 | AD_VP |
| De Haan | 90 | AD_VP |
| Higgins | 110 | AC_MGR |
| Gietz | 110 | AC_ACCOUNT |

6 rows selected.

```
SELECT last_name, department_id, job_id
FROM employees
WHERE department_id IN (SELECT department_id
FROM departments
WHERE location_id = 1700);
```

5. 显示每个向 King 报告的雇员的名字和薪水。

| LAST_NAME | SALARY |
|-----------|--------|
| Kochhar | 17000 |
| De Haan | 17000 |
| Mourgos | 5800 |
| Zlotkey | 10500 |
| Hartstein | 13000 |

```

SELECT last_name, salary
FROM employees
WHERE manager_id = (SELECT employee_id
FROM employees
WHERE last_name = 'King');

```

6. 显示在 Executive 部门的每个雇员的 department number、last name 和 job ID。

| DEPARTMENT_ID | LAST_NAME | JOB_ID |
|---------------|-----------|---------|
| 90 | King | AD_PRES |
| 90 | Kochhar | AD_VP |
| 90 | De Haan | AD_VP |

```

SELECT department_id, last_name, job_id
FROM employees
WHERE department_id IN (SELECT department_id
FROM departments
WHERE department_name = 'Executive');

```

如果你还有时间，完成下面的习题：

7. 修改 lab6_3.sql 中的查询显示所有收入高于平均薪水并且工作在有任一雇员的名字中带有 u 的部门的雇员的 employee numbers、last names 和 salaries。保存 lab6_3.sql 到 lab6_7.sql。运行 lab6_7.sql 中的语句。

| EMPLOYEE_ID | LAST_NAME | SALARY |
|-------------|-----------|--------|
| 103 | Hunold | 9000 |

```

SELECT employee_id, last_name, salary
FROM employees
WHERE department_id IN (SELECT department_id
FROM employees
WHERE last_name like '%u%')
AND salary > (SELECT AVG(salary)
FROM employees);

```