

Министерство образования и науки Украины
Харьковский национальный университет радиоэлектроники

Дисциплина:
«Организация баз данных и знаний»

для студентов дневной формы обучения
по направлению
6.050101 «Компьютерные науки»

Операторы над множествами: объединение, пересечение, разность

1. Количество атрибутов в предложениях ***SELECT*** должно совпадать.
2. Соответствующие атрибуты должны быть совместимы по типу данных.



Оператор UNION

Применяется для объединения результатов двух запросов. Результатом является выборка данных без дублирования кортежей.



```
select emp.employee_id from hr.employees emp
union
select jh.employee_id from hr.job_history jh
```

Row #	EMPLOYEE_ID
1	100
2	101
3	102
4	103
5	104
6	105
7	106
8	107

Row 1 (1x1) 7 lines 93 characters Insert Modified Exec time: 0,016 sec Data Set is Read-Only; 107 row(s) fetched

DB Navigator [HR@XE(1)] HR@XE(1):Untitled2

Оператор UNION ALL

Применяется для объединения результатов двух запросов. Результатом является выборка данных с повторяющимися кортежами.



```
select emp.employee_id, emp.department_id
from hr.employees emp
union all
select jh.employee_id, jh.department_id
from hr.job_history jh
```

EMPLOY...	DEPARTMENT_ID
100	90
101	90
101	110
101	110
102	90
102	60
103	60
104	60
105	60

Auto-Refresh every 20 sec(s)

Spool Result 1

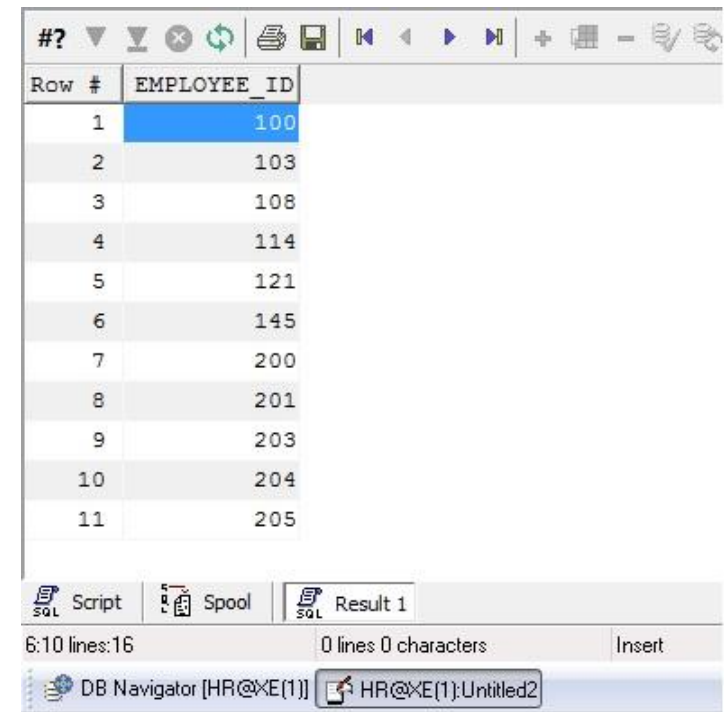
2x3] 6 lines 150 characters Insert Modified Exec time: 0,016 sec Data Set is Read-Only; 117 row(s) fetched

Navigator [HR@XE(1)] HR@XE(1):Untitled2

Оператор INTERSECT

Применяется для получения выборки, кортежи которой являются общими для двух запросов – реализация операции пересечения множеств.

```
select emp.employee_id  
from hr.employees emp  
intersect  
select dep.manager_id  
from hr.departments dep
```



Row #	EMPLOYEE_ID
1	100
2	103
3	108
4	114
5	121
6	145
7	200
8	201
9	203
10	204
11	205

Оператор MINUS

Реализация операции разности двух множеств.



Row #	EMPLOYEE_ID
1	101
2	102
3	104
4	105
5	106
6	107
7	109
8	110
9	111
10	112

Script Spool Result 1
Row 1 (1x1) 5 lines 92 characters Insert Modified Data Set is Read-Only: 96 row(s) fetched

```
select emp.employee_id
from hr.employees emp
      MINUS
select dep.manager_id
from hr.departments dep
```

```
select dep.manager_id
from hr.departments dep
MINUS
```

```
select emp.employee_id
from hr.employees emp
```

Row #	MANAGER_ID
1	

Script Spool	0 lines 0 characters	Insert	Modified	Exec time: 0,015 sec	Data Set is Read-Only: 1 row(s) fetched
--------------	----------------------	--------	----------	----------------------	---

Типы подзапросов

- 1. Однострочные подзапросы.** Возвращают во внешний оператор SQL одну или ноль строк.
- 2. Многострочные подзапросы.** Возвращают во внешний оператор SQL более одной строки.

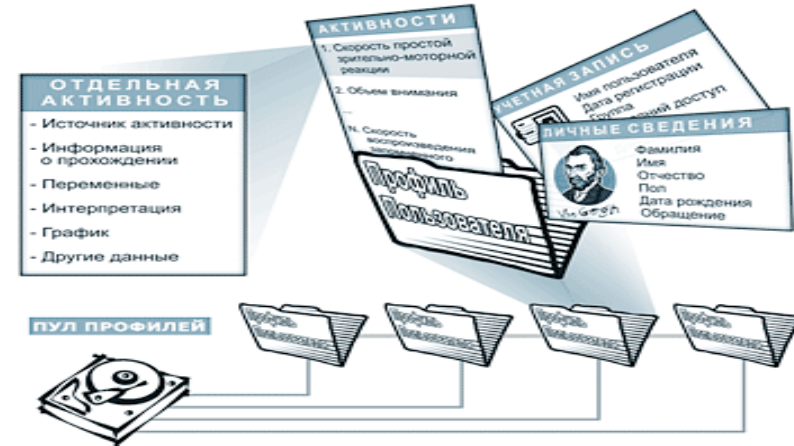
Подтипы подзапросов

- 1. Многостолбцовые подзапросы.** Возвращают во внешний оператор более одного столбца
- 2. Коррелированные подзапросы.** Ссылаются на один или несколько столбцов во внешнем операторе (используют таблицу внешнего запроса).
- 3. Вложенные подзапросы.** Помещаются внутри другого подзапроса. Подзапросы могут иметь уровень вложенности до 255.

Подзапросы предложения SELECT

Подзапрос может содержаться во фразах:

1. WHERE
2. HAVING
3. FROM
4. SELECT – (только для однострочных подзапросов!)



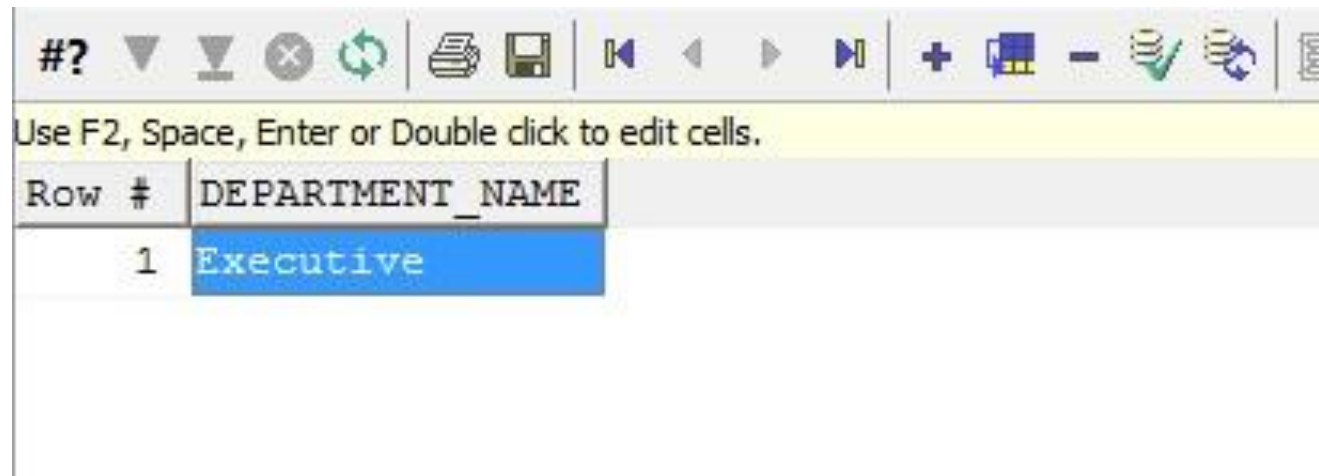
Подзапросы **не должны** содержать конструкцию
ORDER BY



Результаты подзапросов

1. Однострочный WHERE

```
select department_name
from hr.departments dep
where dep.department_id =
      (select emp.department_id
       from hr.employees emp
        where upper(emp.first_name) like 'NEENA')
```



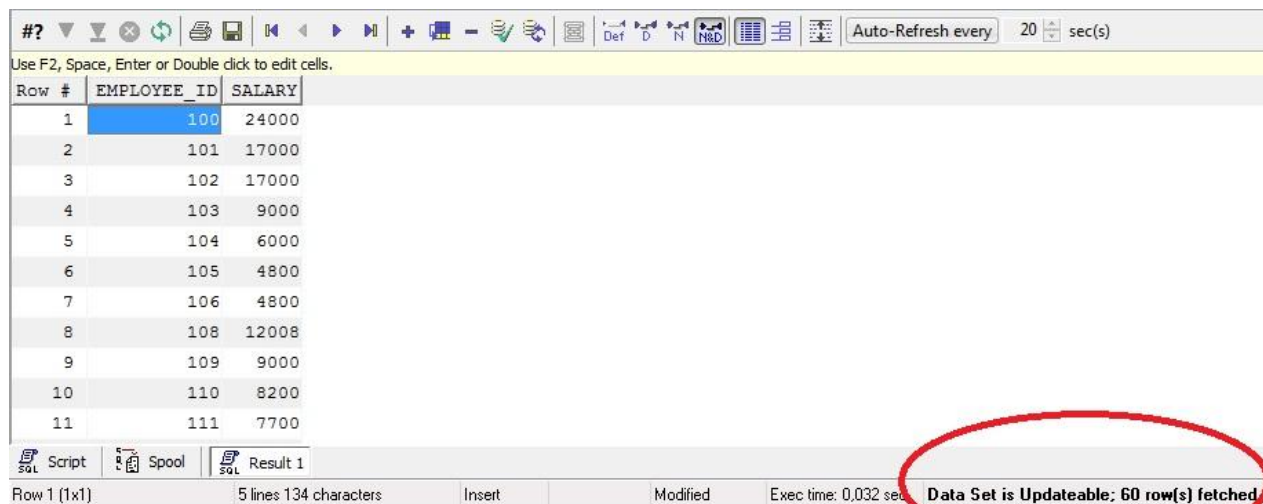
The screenshot shows a SQL query result in a grid tool. The toolbar at the top includes icons for help, dropdown, zoom, undo, redo, print, save, and navigation. Below the toolbar, a yellow banner reads "Use F2, Space, Enter or Double click to edit cells." The table has two columns: "Row #" and "DEPARTMENT_NAME". The first row shows the value "1" in the "Row #" column and "Executive" in the "DEPARTMENT_NAME" column.

Row #	DEPARTMENT_NAME
1	Executive

Результаты подзапросов

Результат выборки с использованием операторов сравнения в подзапросе

```
select emp.employee_id, emp.salary
from hr.employees emp
where salary >
    (select avg(salary)
     from hr.employees e
     where e.department_id=10)
```



Row #	EMPLOYEE_ID	SALARY
1	100	24000
2	101	17000
3	102	17000
4	103	9000
5	104	6000
6	105	4800
7	106	4800
8	108	12008
9	109	9000
10	110	8200
11	111	7700

Row 1 (1x1) 5 lines 134 characters Insert Modified Exec time: 0,032 sec Data Set is Updateable: 60 row(s) fetched

Подзапросы во фразе *HAVING*

Применяется для фильтрации групп строк на основании результата, который возвращает подзапрос.

Найти отдел, в котором средняя зарплата меньше максимальной средней зарплаты

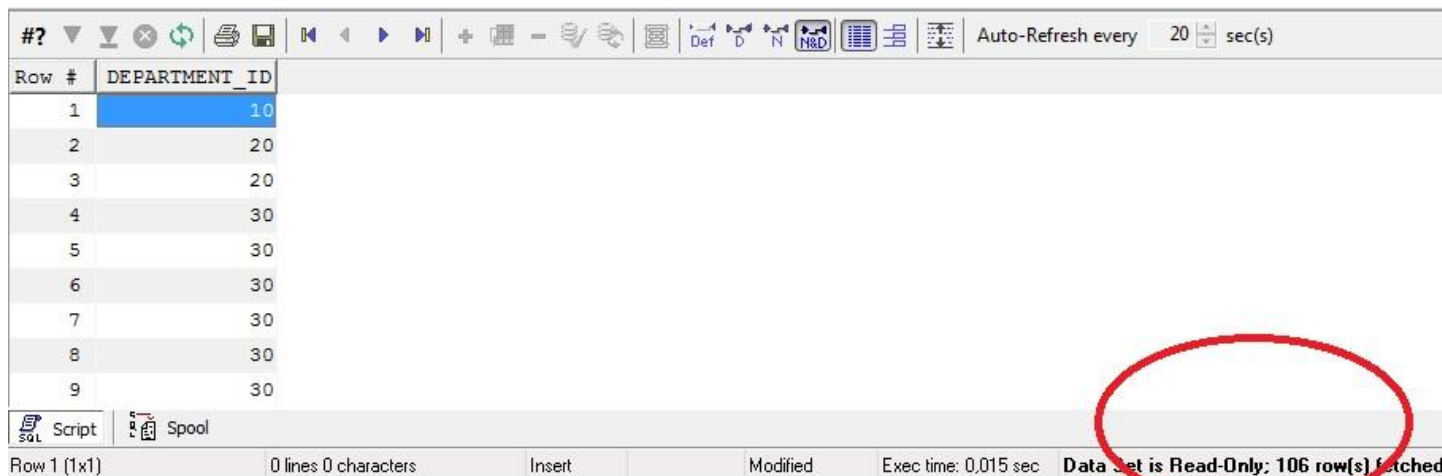
```
select emp.department_id, avg(salary)
from hr.employees emp
having avg(salary) <
    (select max(avg(salary)) from hr.employees e group
     by department_id)
group by emp.department_id
```

[illegible]

Подзапросы во фразе *FROM*

«Встроенные представления»

```
select a.department_id
from
  (select dep.department_id
   from hr.departments dep) a,
  (select department_id
   from hr.employees emp) b
WHERE a.department_id=b.department_id
```



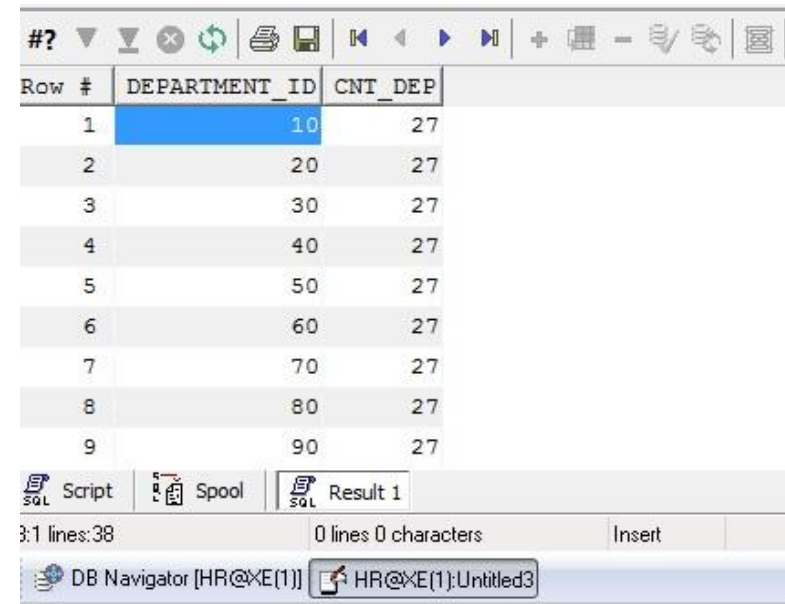
Row #	DEPARTMENT_ID
1	10
2	20
3	20
4	30
5	30
6	30
7	30
8	30
9	30

Row 1 (1x1) 0 lines 0 characters Insert Modified Exec time: 0,015 sec Data Set is Read-Only; 106 row(s) fetched

Подзапросы во фразе *SELECT*

Подзапрос должен быть только однострочным.

```
select department_ID, (select count(department_id)
  from hr.departments dep ) cnt_dep
from hr.departments d
```



Row #	DEPARTMENT_ID	CNT_DEP
1	10	27
2	20	27
3	30	27
4	40	27
5	50	27
6	60	27
7	70	27
8	80	27
9	90	27

Подзапрос может не возвращать ни одной строки.

```
select department_ID, (select department_id from hr.employees where
department_id=-1) cnt_dep from hr.departments d
```

Многострочные подзапросы

Что делать, если подзапрос возвращает несколько строк?

```
select department_name
from hr.departments dep
where dep.department_id =
(select emp.department_id
from hr.employees emp
where upper(emp.first_name) like 'STEVEN')
```

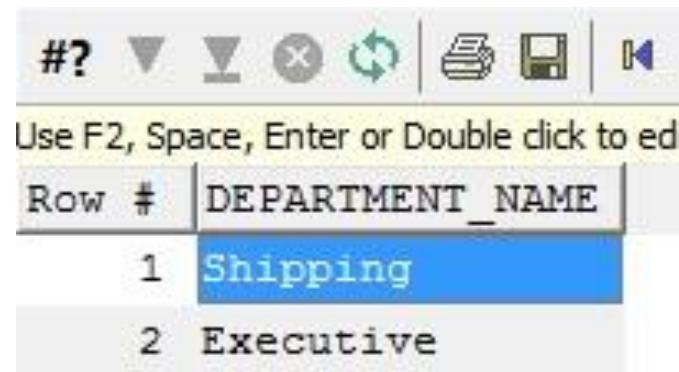


```
i 14:22:02 select department_name
i 14:22:02 from hr.departments dep
i 14:22:02 ...
! 14:22:02 line 2: ORA-01427: single-row subquery returns more than one row
i 14:22:02 **** SCRIPT ENDED 27-фев-2014 14:22:02 ****
i 14:22:02 End Script Execution
```

Использование оператора *IN* в многострочных подзапросах

Используется для проверки, входит ли значение в определенный список значений.

```
select department_name
from hr.departments dep
where dep.department_id IN
      (select emp.department_id
       from hr.employees emp
       where upper(emp.first_name) like 'STEVEN')
```



Row #	DEPARTMENT_NAME
1	Shipping
2	Executive

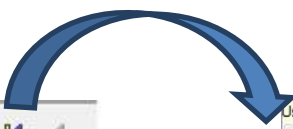
Использование оператора ANY

в многострочных подзапросах

Используется для сравнения значений с каждым значением из списка.
Перед оператором **ANY** в запросе указывается оператор сравнения.

Меньше, чем любое значение из подзапроса

```
select emp.last_name, salary
from hr.employees emp
where salary < ANY (select salary from hr.employees e
                    where e.department_id=30 )
```



Row #	SALARY
1	11000
2	3100
3	2900
4	2800
5	2600
6	2500

Row #	LAST_NAME	SALARY
81	Hunold	9000
82	Faviet	9000
83	Hall	9000
84	McEwen	9000
85	Sully	9500
86	Greene	9500
87	Bernstein	9500
88	Fox	9600
89	Baer	10000
90	Bloom	10000
91	King	10000
92	Tucker	10000
93	Vishney	10500
94	Zlotkey	10500

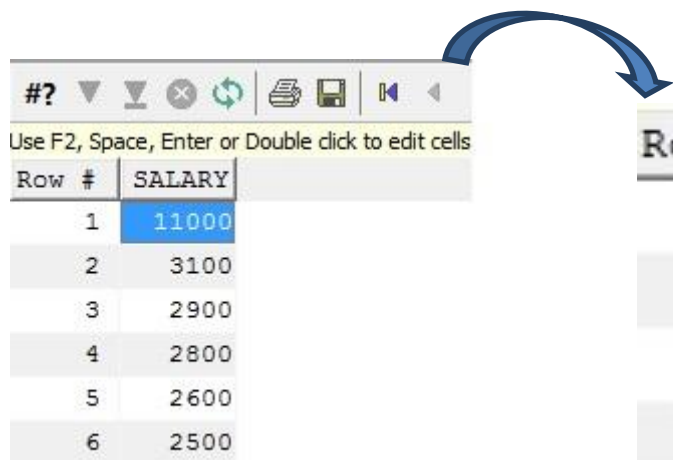
Использование оператора *ALL*

в многострочных подзапросах

Используется для сравнения значений со всеми значениями из списка. Перед оператором **ALL** в запросе указывается оператор сравнения.

Меньше, чем каждое значение из подзапроса

```
select emp.last_name, salary
from hr.employees emp
where salary < ALL (select salary from hr.employees e
                     where e.department_id=30 )
```



Row #	SALARY
1	11000
2	3100
3	2900
4	2800
5	2600
6	2500

Row #	LAST_NAME	SALARY
1	Gee	2400
2	Landry	2400
3	Philtanker	2200
4	Markle	2200
5	Olson	2100

Многостолбцовые подзапросы

Подзапрос может возвращать несколько атрибутов.

Найти сотрудников, у которых в отделе минимальная зарплата

```
select dep.department_name, emp.department_id,  
emp.last_name, salary  
from hr.departments dep, hr.employees emp  
where dep.department_id=emp.department_id  
and (emp.department_id, salary) IN  
  (select emp.department_id, min(salary)  
   from hr.employees emp  
   group by emp.department_id)  
order by dep.department_name;
```

Row #	DEPARTMENT_ID	MIN(SALARY)
1	100	6900
2	30	2500
3		7000
4	90	17000
5	20	6000
6	70	10000
7	110	8300
8	50	2100
9	80	6100
10	40	6500
11	60	4200
12	10	4400



Row #	DEPARTMENT_NAME	DEPARTMENT_ID	LAST_NAME	SALARY
1	Accounting	110	Gietz	8300
2	Administration	10	Whalen	4400
3	Executive	90	Kochhar	17000
4	Executive	90	De Haan	17000
5	Finance	100	Popp	6900
6	Human Resources	40	Mavris	6500
7	IT	60	Lorentz	4200
8	Marketing	20	Fay	6000
9	Public Relations	70	Baer	10000
10	Purchasing	30	Colmenares	2500
11	Sales	80	Kumar	6100
12	Shipping	50	Olson	2100

Коррелированные подзапросы

Результат внешнего запроса зависит от результата подзапроса, который использует таблицу внешнего запроса.

Строки внешнего запроса по очереди передаются в подзапрос. В свою очередь подзапрос построчно читает из внешнего запроса и применяет прочитанную строку к подзапросу, пока не будут обработаны все строки внешнего запроса.

После этого возвращаются результаты всего запроса в целом.

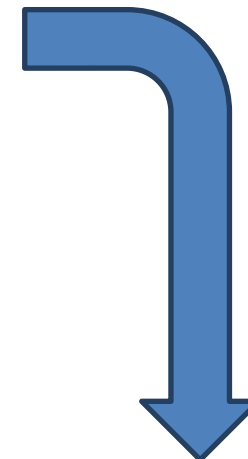


Коррелированные подзапросы (1)

Найти сотрудников, у которых значение зарплаты равно среднему значению зарплаты в отделе этого сотрудника.

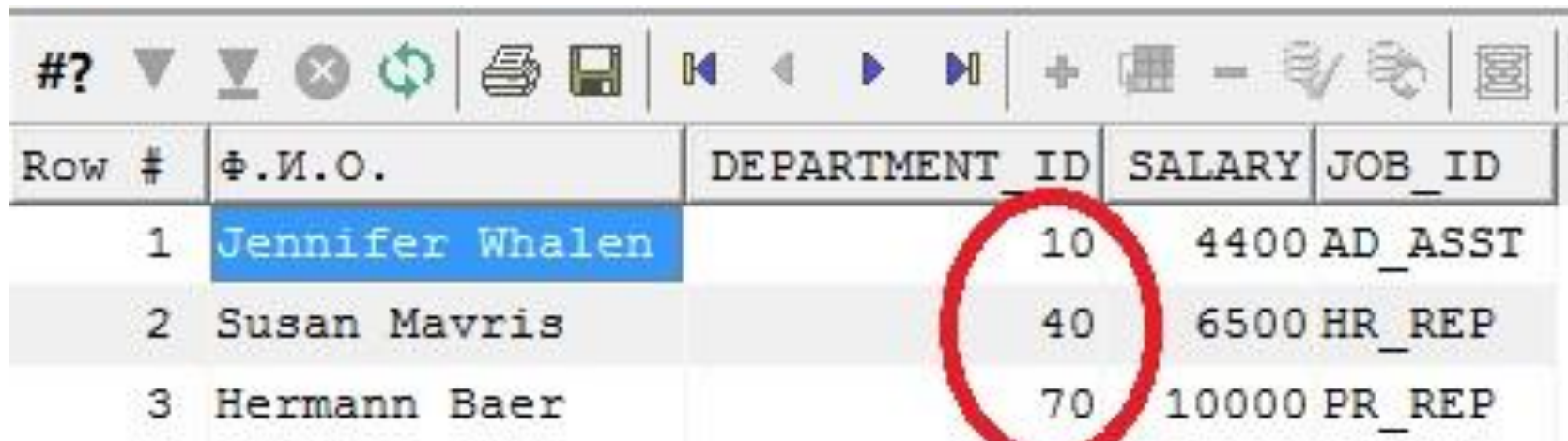
Дополнительная информация: *Средняя зарплата по отделам:*

```
select avg(salary), department_id from hr.employees
group by e.department_id
```

[illegible]

Результат коррелированного подзапроса

```
select emp.first_name||' '||emp.last_name "Ф.И.О.",  
department_id, emp.salary, emp.job_id  
from hr.employees emp  
where emp.salary =(select avg(emp1.salary) from  
                    hr.employees emp1  
                    where emp.department_id=emp1.department_id)  
order by emp.job_id, "Ф.И.О."
```



Row #	Ф.И.О.	DEPARTMENT_ID	SALARY	JOB_ID
1	Jennifer Whalen	10	4400	AD_ASST
2	Susan Mavris	40	6500	HR_REP
3	Hermann Baer	70	10000	PR_REP

Оператор *NOT IN* и значения *NULL*

Если список вхождений содержит *NULL* значение, то оператор *NOT IN* не вернет строк. Необходимо использовать функции обработки значений *NULL*.

```
select emp.employee_id, emp.manager_id,  
emp.department_id  
from hr.employees emp  
where emp.manager_id NOT IN  
      (select nvl(dep.manager_id, -1)  
       from hr.departments dep)
```

Row #	EMPLOYEE_ID	MANAGER_ID	DEPARTMENT_ID
1	161	146	80
2	160	146	80
3	159	146	80
4	158	146	80
5	157	146	80
6	156	146	80
7	205	101	110
8	204	101	70

Вложенные подзапросы

```
select emp.department_id, avg(salary)
from hr.employees emp
having avg(salary) <
    (select max(avg(salary)) from hr.employees e
     where department_id in
        (select department_id from hr.departments dep)
     group by department_id)
group by emp.department_id
```

[illegible]