

Неделя 11. Лабораторная SQL Server

В SQL Server для обозначения одного произвольного символа используется подчеркивание «_», а для обозначения последовательности длиной 0 или более символов используется символ «%».

Задание. Напишите на SQL запрос, который выведет в обратном алфавитном порядке названия книг, выпущенных после 1999 года, фамилия автора (значение поля Author) у которых начинается с буквы «Г» или название издательства заканчивается буквой «а».

Для того чтобы проверить работу запроса необходимо вставить несколько строчек в таблицу:

```
USE [MyLib]
GO

INSERT INTO [dbo].[Book]
([Author]
,[Title]
,[Publisher]
,[BookYear])
VALUES
('Гербельт У.Д.'
,'Машина времени'
,'Клависин'
,2010)
GO

INSERT INTO [dbo].[Book]
([Author]
,[Title]
,[Publisher]
,[BookYear])
VALUES
('Гумилев И.С.'
,'Жираф'
,'СтихСтих'
,2001)
GO

INSERT INTO [dbo].[Book]
([Author]
,[Title]
,[Publisher]
,[BookYear])
VALUES
('Пушкин А.С.'
,'Капитанская дочка'
,'Астория'
,2012)
GO
```

Теперь таблица BOOK выглядит так:

	BookId	Author	Title	Publisher	BookYear
1	1	Дейт К.	Введение в системы баз данных. 8-е издание	Издательский дом "Вильямс"	2005
2	2	Карпова Т.С.	Базы данных	Питер	2001
3	3	Зеленин В.М.	Экспертные системы	NULL	1996
4	4	Гербельт У.Д.	Машина времени	Клависин	2010
5	5	Гумилев И.С.	Жираф	СтихСтих	2001
6	6	Пушкин А.С.	Капитанская дочка	Астория	2012

Выполним запрос:

SELECT Title

FROM Book

WHERE BookYear > 1999 AND (Author LIKE 'Г%' OR Publisher LIKE '%а')

ORDER BY Title DESC;

Вывод:

	Title
1	Машина времени
2	Капитанская дочка
3	Жираф

Задание. Напишите запрос, выводящий названия книг, содержащие символ подчеркивания во второй или любой следующей позиции. (Первый символ может быть произвольным, но если это подчеркивание, и оно единственное, такое название выводить не надо).

Для этого задания снова изменим данные в БД

BookId	Author	Title	Publisher	BookYear
1	Дейт К.	Введение в системы баз	Издательский	2005
2	Карпова Т.С.	Базы_данных	Питер	2001
3	Зеленин В.М.	Экспертные_системы		1996
4	Гербельт У.Д.	Машина времени	Америка	2010
5	Гумилев И.С.	Жираф	СтихСтих	2001
6	Пушкин А.С.	Капитанская_дочка	Литература	2012

В трех книгах заменим пробел на _

Выполним запрос:

```
SELECT Title
FROM Book
WHERE Title LIKE '%!_%' ESCAPE '!';
```

После LIKE используем последовательность ' _%!_%', которая говорит, что первый символ должен быть обязательно (знак _), потом может быть от 0 до многих символов (знак %), потом в title должен быть символ _ (об этом говорит знак !, тогда _ рассматривается не как символ подстановки, а просто как символ). Потом еще от 0 до многих символов (знак %).

Задание. Напишите запрос, выводящий без повторения список издательств, таких, что хотя бы одна книга этого издательства имеет статус «выдана»

Снова подправим для этого задания БД

BookId	Author	Title	Publisher	BookYear
1	Дейт К.	Введение в системы баз	Издательский	2005
2	Карпова Т.С.	Базы_данных	Питер	2001
3	Зеленин В.М.	Экспертные_системы	Питер	1996
4	Гербельт У.Д.	Машина времени	Америка	2010
5	Гумилев И.С.	Жираф	СтихСтих	2001
6	Пушкин А.С.	Капитанская_дочка	Литература	2012
9	Лермонтов М.	Мцыри	Литература	1999
10	Гоголь Н.В.	Шинель	Питер	2001

LibID	BookID	StatusID
1	2	1
2	2	3
3	3	3
4	9	3
5	10	3

Запрос:

```
SELECT DISTINCT Publisher
FROM (Book INNER JOIN BookInLib ON Book.BookId=BookInLib.BookID) INNER JOIN BookStatus
ON BookInLib.StatusID= BookStatus.StatusID
WHERE BookStatus.StatusName='выдана';
```

DISTINCT – выводим без повторений

Сначала делаем BOOK INNER JOIN BookInLib. Оператор INNER JOIN формирует таблицу из записей двух или нескольких таблиц. Каждая строка из первой (левой) таблицы, сопоставляется с каждой строкой из второй (правой) таблицы, после чего происходит проверка условия. Если условие истинно, то строки попадают в результирующую таблицу. Таким образом остаются только книги, которые имеют хотя бы какой-то статус

Потом так же добавляем с помощью INNER JOIN таблицу со статусами, что бы отобрать из всех записей только те, у которых статус «выдана»

Вывод:

	Publisher
1	Литература
2	Питер

Задание. Измените предыдущий запрос, чтобы он выводил без повторения список издательств, таких, что или ни одного экземпляра хотя бы одной книги этого издательства нет в библиотеке (в таблице BookInLib), или хотя бы одна книга этого издательства имеет статус «выдана».

Запрос:

```
SELECT DISTINCT Publisher
FROM (Book LEFT JOIN BookInLib ON Book.BookId=BookInLib.BookID)
LEFT JOIN BookStatus ON BookInLib.StatusID= BookStatus.StatusID
WHERE (BookInLib.LibID IS NULL) OR (BookStatus.StatusName ='выдана');
```

Заменяем INNER JOIN на LEFT JOIN и меняю WHERE

Во время выполнения LEFT JOIN сначала происходит формирование таблицы внутренним

соединением (оператор SQL INNER JOIN) левой и правой таблиц. Затем, в результат добавляются записи левой таблицы не вошедшие в результат формирования таблицы внутренним соединением. Для них, соответствующие записи из правой таблицы заполняются значениями NULL.

Вывод:

	Publisher
1	Америка
2	Издательский дом "Вильямс"
3	Литература
4	Питер
5	СтихСтих

Задание. По аналогии с рассмотренным примером напишите свой запрос, выводящий без повторения все сочетания разных издательств, упомянутых в таблице Book.

```
SELECT DISTINCT B.Publisher, B1.Publisher
FROM Book AS B, Book AS B1
WHERE B.Publisher <> B1.Publisher;
```

По сути делаем декартовое произведение таблицы саму с собой.

Вывод:

	Publisher	Publisher
1	Америка	Издательский дом "Вильямс"
2	Америка	Литература
3	Америка	Питер
4	Америка	СтихСтих
5	Издательский дом "Вильямс"	Америка
6	Издательский дом "Вильямс"	Литература
7	Издательский дом "Вильямс"	Питер
8	Издательский дом "Вильямс"	СтихСтих
9	Литература	Америка
10	Литература	Издательский дом "Вильямс"
11	Литература	Питер
12	Литература	СтихСтих
13	Питер	Америка
14	Питер	Издательский дом "Вильямс"
15	Питер	Литература
16	Питер	СтихСтих
17	СтихСтих	Америка
18	СтихСтих	Издательский дом "Вильямс"
19	СтихСтих	Литература
20	СтихСтих	Питер

Неделя 11. Лабораторная Oracle

Был создан новый пользователь

```
alter session set "_ORACLE_SCRIPT"=true;
create user hr -- имя пользователя (oraclelabs)
identified by hr -- пароль пользователя (oracle)
```

```

default tablespace users; -- табличное пространство для хранения данных таблиц и индексов

grant connect to hr; -- предоставление привилегий для подключения к БД

grant resource to hr; -- предоставление привилегий на создание основных объектов в своей схеме

grant create view to hr; -- предоставление привилегий на создание представлений

GRANT ALL PRIVILEGES TO hr;

```

Задание 2

Команда ниже содержит 4 ошибки, *найдите и исправьте их*.

```

SELECT last_name, first_name
'работает в должности: ' + jobid CURRENT JOB
FROM employees;

```

```

SELECT last_name, first_name, 'работает в должности: ' || job_id AS "CURRENT JOB"
FROM employees;

```

- 1 ошибка: jobid, пропущено нижнее подчеркивание. job_id
- 2 ошибка: пропущены кавычки для псевдонима. Можно поставить AS. AS "CURRENT JOB"
- 3 ошибка: пропущены ||
- 4 ошибка: пропущена запятая

	LAST_NAME	FIRST_NAME	CURRENT JOB
1	Abel	Ellen	работает в должности: SA_REP
2	Davies	Curtis	работает в должности: ST_CLERK
3	De Haan	Lex	работает в должности: AD_VP
4	Ernst	Bruce	работает в должности: IT_PROG
5	Fay	Pat	работает в должности: MK_REP
6	Gietz	William	работает в должности: AC_ACCOUNT
7	Grant	Kimberely	работает в должности: SA_REP
8	Hartstein	Michael	работает в должности: MK_MAN
9	Higgins	Shelley	работает в должности: AC_MGR
10	Hunold	Alexander	работает в должности: IT_PROG
11	King	Steven	работает в должности: AD_PRES
12	Kochhar	Neena	работает в должности: AD_VP
13	Lorentz	Diana	работает в должности: IT_PROG
14	Matos	Randall	работает в должности: ST_CLERK
15	Mourgos	Kevin	работает в должности: ST_MAN
16	Rajs	Trenna	работает в должности: ST_CLERK
17	Taylor	Jonathon	работает в должности: SA_REP
18	Vargas	Peter	работает в должности: ST_CLERK
19	Whalen	Jennifer	работает в должности: AD_ASST
20	Zlotkey	Eleni	работает в должности: SA_MAN

Задание 3

Создайте запрос для получения в одном отчёте номера, фамилии, имени, должности, зарплаты, номера руководителя и номера отдела для:

- a. сотрудников высшего звена, т.е. таких, которые не имеют руководителя (менеджера) и
- b. сотрудников, не приписанных ни к одному отделу

Запрос:

```
SELECT employee_id, last_name, first_name, job_id, salary, manager_id, department_id
FROM employees
WHERE (manager_id IS NULL) OR( department_id IS NULL);
```

Таблица employees

EMPLOYEE_ID	LAST_NAME	FIRST_NAME	JOB_ID	SALARY	MANAGER_ID	DEPARTMENT_ID
1	100 King	Steven	AD_PRES	24000	(null)	90
2	101 Kochhar	Neena	AD_VP	17000	100	90
3	102 De Haan	Lex	AD_VP	17000	100	90
4	200 Whalen	Jennifer	AD_ASST	4400	101	10
5	205 Higgins	Shelley	AC_MGR	12000	101	110
6	206 Gietz	William	AC_ACCOUNT	8300	205	110
7	149 Zlotkey	Eleni	SA_MAN	10500	100	80
8	174 Abel	Ellen	SA_REP	11000	149	80
9	176 Taylor	Jonathon	SA_REP	8600	149	80
10	178 Grant	Kimberely	SA_REP	7000	149	(null)
11	124 Mourgos	Kevin	ST_MAN	5800	100	50
12	141 Rajs	Trenna	ST_CLERK	3500	124	50
13	142 Davies	Curtis	ST_CLERK	3100	124	50
14	143 Matos	Randall	ST_CLERK	2600	124	50
15	144 Vargas	Peter	ST_CLERK	2500	124	50
16	103 Hunold	Alexander	IT_PROG	9000	102	60
17	104 Ernst	Bruce	IT_PROG	6000	103	60
18	107 Lorentz	Diana	IT_PROG	4200	103	60
19	201 Hartstein	Michael	MK_MAN	13000	100	20
20	202 Fay	Pat	MK_REP	6000	201	20

Вывод по запросу:

EMPLOYEE_ID	LAST_NAME	FIRST_NAME	JOB_ID	SALARY	MANAGER_ID	DEPARTMENT_ID
1	100 King	Steven	AD_PRES	24000	(null)	90
2	178 Grant	Kimberely	SA_REP	7000	149	(null)

Задание 4

Вы хотите узнать, какие сотрудники подчиняются любым руководителям, кроме сотрудника высшего звена (с номером 100) и выполняете ледующий запрос:

```
SELECT employee_id, last_name, first_name, manager_id
FROM employees
WHERE manager_id NOT IN (100, null);
```

При просмотре таблицы «вручную» вы можете увидеть, что такие сотрудники существуют (например, это сотрудники с номерами 103 и 104), но в результат запроса они не попали. Как вы думаете, почему?

При наличии NULL-значения в наборе предикат NOT IN в предложении WHERE всегда будет давать пустой набор записей.

Как следует переписать запрос, чтобы получить верный ответ?

```
SELECT employee_id, last_name, first_name, manager_id
FROM employees
WHERE manager_id NOT IN (100);
```

Вывод:

	EMPLOYEE_ID	LAST_NAME	FIRST_NAME	MANAGER_ID
1	200	Whalen	Jennifer	101
2	205	Higgins	Shelley	101
3	206	Gietz	William	205
4	174	Abel	Ellen	149
5	176	Taylor	Jonathon	149
6	178	Grant	Kimberely	149
7	141	Rajs	Trenna	124
8	142	Davies	Curtis	124
9	143	Matos	Randall	124
10	144	Vargas	Peter	124
11	103	Hunold	Alexander	102
12	104	Ernst	Bruce	103
13	107	Lorentz	Diana	103
14	202	Fay	Pat	201

Задание 6

Сформируйте запрос для определения полного числа лет, отработанных каждым сотрудником. Назовите столбец YEARS_WORKED. Выведите дополнительно номер сотрудника, его фамилию и дату приёма на работу. Отсортируйте результат в порядке возрастания числа проработанных лет. Если какие-то сотрудники трудятся в компании одинаковое количество лет, отсортируйте их по фамилиям (по алфавиту)

```
SELECT employee_id, last_name, hire_date, TRUNC (MONTHS_BETWEEN (sysdate, hire_date)/12) AS
"YEARS WORKED"
FROM employees
ORDER BY years_worked, last_name;
```

Сначала используем функцию MONTHS_BETWEEN, чтобы узнать сколько месяцев между настоящим временем и началом работы. Потом урезаем до целого числа.

Вывод:

	EMPLOYEE_ID	LAST_NAME	HIRE_DATE	YEARS_WORKED
1	178	Grant	24.05.99	20
2	124	Mourgos	16.11.99	20
3	149	Zlotkey	29.01.00	20
4	107	Lorentz	07.02.99	21
5	144	Vargas	09.07.98	21
6	202	Fay	17.08.97	22
7	143	Matos	15.03.98	22
8	176	Taylor	24.03.98	22
9	142	Davies	29.01.97	23
10	174	Abel	11.05.96	24
11	201	Hartstein	17.02.96	24
12	141	Rajs	17.10.95	24
13	206	Gietz	07.06.94	25
14	205	Higgins	07.06.94	25
15	102	De Haan	13.01.93	27
16	104	Ernst	21.05.91	28
17	103	Hunold	03.01.90	30
18	101	Kochhar	21.09.89	30
19	100	King	17.06.87	32
20	200	Whalen	17.09.87	32

Задание 8

Сформируйте запрос, позволяющий определить, в каких отделах трудится более 10 сотрудников. Выведите номер отдела и общее количество сотрудников. Отсортируйте результат по уменьшению общего количества сотрудников.

Предложение HAVING применяется после группировки для определения аналогичного предиката, фильтрующего группы по значениям агрегатных функций.

То есть сначала ведем подсчет сотрудников, потом группируем по id департамента, потом фильтруем группы с помощью Having.

```
SELECT department_id, COUNT (employee_id) AS "EMP_CNT"  
FROM employees  
GROUP BY department_id  
HAVING COUNT(employee_id) > 10  
ORDER BY emp_cnt DESC;
```

Вывод:

DEPARTM...	EMP_CNT
------------	---------

Так как сотрудников мало, изменим запрос для больше наглядности.

```
SELECT department_id, COUNT(employee_id) AS "EMP_CNT"  
FROM employees  
GROUP BY department_id  
HAVING COUNT(employee_id) >= 3  
ORDER BY emp_cnt DESC;
```

Вывод:

	DEPARTMENT_ID	EMP_CNT
1	50	5
2	60	3
3	80	3
4	90	3

Задание 9

Сформируйте запрос для определения количества сотрудников, нанятых в 2007 и 2008 годах и общего количества сотрудников.

Запрос:

```
SELECT  
COUNT(CASE WHEN EXTRACT(year FROM hire_date)=2007 THEN 1 ELSE NULL END) AS "CNT 2007",  
COUNT(CASE WHEN EXTRACT(year FROM hire_date)=2008 THEN 1 ELSE NULL END) AS "CNT 2008",  
COUNT(employee_id) AS "CNT TOTAL"  
FROM employees;
```

Используем блок WHEN-THEN-ELSE, таким образом определяя, когда добавлять 1, а когда нет.

Так как нет людей с датой начала 2007 2008

	PHONE_NUMBER	HIRE_DATE	JOB_ID
1	515.123.4567	17.06.87	AD_PRES
2	515.123.4568	21.09.89	AD_VP
3	515.123.4569	13.01.93	AD_VP
4	515.123.4444	17.09.87	AD_ASST
5	515.123.8080	07.06.94	AC_MGR
6	515.123.8181	07.06.94	AC_ACCOUNT
7	011.44.1344.429018	29.01.00	SA_MAN
8	011.44.1644.429267	11.05.96	SA_REP
9	011.44.1644.429265	24.03.98	SA_REP
10	011.44.1644.429263	24.05.99	SA_REP
11	650.123.5234	16.11.99	ST_MAN
12	650.121.8009	17.10.95	ST_CLERK
13	650.121.2994	29.01.97	ST_CLERK
14	650.121.2874	15.03.98	ST_CLERK
15	650.121.2004	09.07.98	ST_CLERK
16	590.423.4567	03.01.90	IT_PROG
17	590.423.4568	21.05.91	IT_PROG
18	590.423.5567	07.02.99	IT_PROG
19	515.123.5555	17.02.96	MK_MAN
20	603.123.6666	17.08.97	MK_REP

То вывод выглядит так:

	CNT 2007	CNT 2008	CNT TOTAL
1	0	0	20

Изменим запрос на даты 1998, 1994.

Вывод:

	CNT 1998	CNT 1994	CNT TOTAL
1	3	2	20

Задание 10

Сформируйте запрос для вывода фамилий и номеров абсолютно всех служащих вместе с фамилиями и номерами их менеджеров. Присвойте соответствующие псевдонимы столбцам. Для сотрудника высшего звена в столбцах с номером и фамилией руководителя выведите 0 и 'не имеет' соответственно. Отсортируйте результат по возрастанию номера сотрудника.

Запрос:

```
SELECT e.employee_id empid, e.last_name empname, NVL(m.employee_id, 0) manid, NVL(m.last_name, 'не имеет') manname
FROM employees e LEFT JOIN employees m
ON (e.manager_id = m.employee_id)
ORDER BY empid;
```

Даем таблицам псевдонимы для удобства работы.

Далее с помощью функции NVL заменяем значения NULL на заданные.

Соединяем таблицы, сортируем результаты.

Вывод:

	EMPID	EMPNAME	MANID	MANNNAME
1	100	King	0	не имеет
2	101	Kochhar	100	King
3	102	De Haan	100	King
4	103	Hunold	102	De Haan
5	104	Ernst	103	Hunold
6	107	Lorentz	103	Hunold
7	124	Mourgos	100	King
8	141	Rajs	124	Mourgos
9	142	Davies	124	Mourgos
10	143	Matos	124	Mourgos
11	144	Vargas	124	Mourgos
12	149	Zlotkey	100	King
13	174	Abel	149	Zlotkey
14	176	Taylor	149	Zlotkey
15	178	Grant	149	Zlotkey
16	200	Whalen	101	Kochhar
17	201	Hartstein	100	King
18	202	Fay	201	Hartstein
19	205	Higgins	101	Kochhar
20	206	Gietz	205	Higgins

Задание 11

Сформируйте запрос для вывода сводной информации по отделам, предпоследняя буква в названии которых – n. Включите в результат следующие столбцы: номер, наименование, местоположение отдела, а также средний оклад работающих в нём сотрудников, округлённый до 2 цифр после запятой.

Обеспечьте вывод отделов, в которых нет сотрудников. Средний оклад в таких отделах должен быть равен нулю.

Отсортируйте результат по возрастанию номера отдела.

```
SELECT department_id, d.department_name, d.location_id, ROUND(AVG(NVL(e.salary, 0)), 2) avgсал
FROM employees e RIGHT JOIN departments d USING (department_id)
WHERE department_name LIKE '%n_'
GROUP BY department_id, d.department_name, d.location_id
ORDER BY department_id;
```

AVG-вычисляем среднее по группе

ROUND – округляем

Вывод:

	DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID	AVGSAL
1	20	Marketing	1800	9500
2	50	Shipping	1500	3500
3	110	Accounting	1700	10150
4	190	Contracting	1700	0

Неделя 12. Лабораторная Oracle

Задание 2

Сформируйте запрос для определения номеров, фамилий, идентификаторов должности и окладов сотрудников, руководитель которых носит фамилию 'Cambrault'.

Делаем задание с помощью вложенного select-а

Но так как ни у кого нет руководителя по фамилии 'Cambrault', немного изменим запрос.

```
SELECT employee_id, last_name, job_id, salary
FROM employees
WHERE manager_id IN (SELECT employee_id FROM employees WHERE last_name = 'Mourgos');
```

	EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
1	141	Rajs	ST_CLERK	3500
2	142	Davies	ST_CLERK	3100
3	143	Matos	ST_CLERK	2600
4	144	Vargas	ST_CLERK	2500

Задание 3

Используя многостолбцовый подзапрос, сформируйте запрос, возвращающий номера, фамилии, идентификаторы должности и оклад для тех сотрудников, чей оклад совпадает с максимальным по их должности.

```
SELECT employee_id, last_name, job_id, salary
FROM employees
WHERE (job_id, salary) IN (SELECT job_id, max_salary FROM jobs);
```

Выполняем многостолбцовый подзапрос.

При выполнении видно, что никто не получает максимальной зарплаты.

EMPLOYEE...	LAST_NAME	JOB_ID	SALARY
-------------	-----------	--------	--------

Проверим это, выполнив запрос:

```
SELECT last_name, salary, max_salary
FROM employees e JOIN jobs j ON e.job_id = j.job_id;
```

	LAST_NAME	SALARY	MAX_SALARY
1	Gietz	8300	9000
2	Higgins	12000	16000
3	Whalen	4400	6000
4	King	24000	40000
5	De Haan	17000	30000
6	Kochhar	17000	30000
7	Hunold	9000	10000
8	Lorentz	4200	10000
9	Ernst	6000	10000
10	Hartstein	13000	15000
11	Fay	6000	9000
12	Zlotkey	10500	20000
13	Abel	11000	12000
14	Taylor	8600	12000
15	Grant	7000	12000
16	Matos	2600	5000
17	Davies	3100	5000
18	Rajs	3500	5000
19	Vargas	2500	5000
20	Mourgos	5800	8500

Задание 5

Создайте подзапрос для получения номеров и фамилий сотрудников, не являющихся руководителями других сотрудников. Используйте оператор ALL для сравнения с результатами подзапроса.

Оператор ALL возвращает true, если все значения подзапроса удовлетворяют условию

```
SELECT employee_id, last_name  
FROM employees  
WHERE employee_id != ALL (SELECT manager_id FROM employees WHERE manager_id IS NOT NULL);
```

Вывод:

EMPLOYEE_ID	LAST_NAME
1	174 Abel
2	142 Davies
3	104 Ernst
4	202 Fay
5	206 Gietz
6	178 Grant
7	107 Lorentz
8	143 Matos
9	141 Rajs
10	176 Taylor
11	144 Vargas
12	200 Whalen

Задание 7

Используйте раздел WITH при формировании запроса для определения наименования и фондов заработной платы (суммы окладов всех сотрудников) таких отделов, размер фондов которых больше 1/8 от общего фонда заработной платы всей компании.

Сначала группируем отделы, таким образом вычисляя в таблице employees с помощью функции SUM зарплату по отделам. Формируем таблицу salary_of_dept

Далее вычисляем общую зарплату всех сотрудников. Формируем таблицу sumsalary

Делаем запрос из таблицы salary_of_dept, где общий фонд больше чем восьмая от всего фонда.

```
WITH salary_of_dept AS  
(SELECT department_name, SUM(salary) AS salary_dept  
FROM employees JOIN departments USING(department_id)  
GROUP BY department_id, department_name),  
sumsalary AS  
(SELECT SUM(salary) AS sum_salary FROM employees)  
SELECT department_name, salary_dept  
FROM salary_of_dept  
WHERE salary_dept > (SELECT sum_salary FROM sumsalary)*(1/8);
```

DEPARTMENT_NAME	SALARY_DEPT
1 Sales	30100
2 Executive	58000

Задание 8

Используя SET-оператор, создайте запрос для определения номеров и идентификаторов должностей сотрудников, чья текущая должность совпадает с когда-либо занимаемой им ранее. Отсортируйте результат по убыванию номера сотрудника.

INTERSECT - Возвращает все различные значения, входящие в результаты выполнения запросов, указанных как слева, так и справа от оператора INTERSECT.

```
SELECT employee_id, job_id
FROM employees
INTERSECT
SELECT employee_id, job_id
FROM job_history
ORDER BY employee_id DESC;
```

	EMPLOYEE_ID	JOB_ID
1	200	AD_ASST
2	176	SA_REP

Задание 9

Используя SET-оператор и коррелированный подзапрос в разделе SELECT, создайте запрос для вывода фамилий и имён всех сотрудников. Если сотрудник является руководителем какого-либо отдела, в третьем столбце для него необходимо вывести название этого отдела, если нет – строку «не является руководителем отдела». В последней строке результата выведите общее количество сотрудников, являющихся руководителями отделов.

UNION ALL – оператор для объединения результирующего набора данных нескольких запросов, а вот данный оператор, выведет уже абсолютно все строки, даже дубли.

Сначала производим запрос в бд employees, где третьим столбиком будет название отдела, в котором он является менеджером. Название отдела вытаскиваем с помощью вложенного SELECT. Если работник не является руководителем, чтобы не выводил NULL, используем функцию NVL. Потом выполняем запрос для подсчета всех руководителей, используя соединение таблиц JOIN. Соединяем все результаты запроса командой UNION ALL.

```
SELECT last_name, first_name
       ,NVL((SELECT department_name FROM departments d WHERE e.employee_id = d.manager_id), 'не
является руководителем отдела') deptname
FROM employees e
UNION ALL
SELECT 'Общее число ', 'руководителей отделов:', TO_CHAR(COUNT(*))
FROM employees e JOIN departments d ON (e.employee_id = d.manager_id);
```

LAST_NAME	FIRST_NAME	DEPTNAME
1 King	Steven	Executive
2 Kochhar	Neena	не является руководителем отдела
3 De Haan	Lex	не является руководителем отдела
4 Hunold	Alexander	IT
5 Ernst	Bruce	не является руководителем отдела
6 Lorentz	Diana	не является руководителем отдела
7 Mourgos	Kevin	Shipping
8 Rajs	Trenna	не является руководителем отдела
9 Davies	Curtis	не является руководителем отдела
10 Matos	Randall	не является руководителем отдела
11 Vargas	Peter	не является руководителем отдела
12 Zlotkey	Eleni	Sales
13 Abel	Ellen	не является руководителем отдела
14 Taylor	Jonathon	не является руководителем отдела
15 Grant	Kimberely	не является руководителем отдела
16 Whalen	Jennifer	Administration
17 Hartstein	Michael	Marketing
18 Fay	Pat	не является руководителем отдела
19 Higgins	Shelley	Accounting
20 Gietz	William	не является руководителем отдела
21 Общее число руководителей отделов: 7		

Задание 10

Создайте таблицу EMP_DEPT на основе содержимого таблиц EMPLOYEES и DEPARTMENTS. Таблица должна содержать 5 столбцов: номер сотрудника, фамилия, номер отдела, к которому он приписан, наименование отдела, квартал и год, в котором сотрудник был принят на работу. Столбцы с номерами сотрудника и отдела должны иметь названия empno и deptno соответственно, последний столбец - hire_quarter

CREATE TABLE emp_depts AS

SELECT employee_id empno, last_name, department_id deptno, department_name, to_char(hire_date, 'yyyy-q') hire_quarter

FROM employees LEFT JOIN departments USING(department_id);

EMPNO	LAST_NAME	DEPTNO	DEPARTMENT_NAME	HIRE_QUARTER
1	200 Whalen	10 Administration		1987-3
2	201 Hartstein	20 Marketing		1996-1
3	202 Fay	20 Marketing		1997-3
4	124 Mourgos	50 Shipping		1999-4
5	141 Rajs	50 Shipping		1995-4
6	142 Davies	50 Shipping		1997-1
7	143 Matos	50 Shipping		1998-1
8	144 Vargas	50 Shipping		1998-3
9	103 Hunold	60 IT		1990-1
10	104 Ernst	60 IT		1991-2
11	107 Lorentz	60 IT		1999-1
12	149 Zlotkey	80 Sales		2000-1
13	174 Abel	80 Sales		1996-2
14	176 Taylor	80 Sales		1998-1
15	100 King	90 Executive		1987-2
16	101 Kochhar	90 Executive		1989-3
17	102 De Haan	90 Executive		1993-1
18	205 Higgins	110 Accounting		1994-2
19	206 Gietz	110 Accounting		1994-2
20	178 Grant	(null) (null)		1999-2

Создаем таблицу на основе подзапроса.

Делаем слияние двух таблиц с помощью LEFT JOIN