

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ**
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**
КЕМЕРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Институт цифры

ОТЧЕТ
О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ 5

«Аналитические функции»

студента 1 курса, группы ПИМ-231

Давыдова Игоря Павловича

Направление 09.04.03 – «Прикладная информатика»

Руководитель:
К.т.н, доц. Завозкин С. Ю.

Работа защищена
« _____ »
“ _____ ” _____ 2023г.

Задание

1. Изучить предлагаемый теоретический материал.
2. Выполнить следующие задания:
 1. Получить для каждого отдела трех наиболее высокооплачиваемых специалистов по продажам. Требование можно интерпретировать так.
 - Выдать список специалистов по продажам, имеющих одну из трех максимальных зарплат. Другими словами, найти все различные значения зарплаты, отсортировать, выбрать три наибольших, и вернуть всех сотрудников, зарплата которых совпадает с одним из этих трех значений. (Рекомендуется использовать функцию `dense_rank()`).
 - Выдать до трех человек с максимальными зарплатами. Если четыре человека имеют одинаковую максимальную зарплату, в ответ не должно выдаваться ни одной строки. Если два сотрудника имеют максимальную зарплату и два — следующую по значению, ответ будет предполагать две строки - два сотрудника с максимальной зарплатой. (Рекомендуется использовать функцию `count(*)` и выражение `range unbounded preceding`).
 - Отсортировать специалистов по продажам по убыванию зарплат. Вернуть первые три строки. Если в отделе менее трех специалистов по продажам, в результате будет менее трех записей. (Рекомендуется использовать функцию `row_number()`).
 2. Необходимо создать отчет, в котором по отделам были бы представлены все сотрудники, причем, для каждого сотрудника выдана дата его приема на работу, за сколько дней до этой даты последний раз принимали сотрудника на работу, и через сколько дней после этого приняли на работу следующего сотрудника.
 3. Запрос с транспонированием. Выдать должности сотрудников отдела в виде столбцов:

DEPTNO	JOB_1	JOB_2	JOB_3
10	CLERK	MANAGER	PRESIDENT
20	ANALYST	ANALYST	CLERK
30	CLERK	MANAGER	SALESMAN

а не в виде строк:

DEPTNO	JOB
10	CLERK
10	MANAGER
10	PRESIDENT
20	ANALYST
20	CLERK
20	MANAGER
30	CLERK
30	MANAGER
30	SALESMAN

Ход работы

1. Получить для каждого отдела трех наиболее высокооплачиваемых специалистов по продажам. Требование можно интерпретировать так.
 - Выдать список специалистов по продажам, имеющих одну из трех максимальных зарплат. Другими словами, найти все различные значения зарплаты, отсортировать, выбрать три наибольших, и вернуть всех сотрудников, зарплата которых совпадает с одним из этих трех значений. (Рекомендуется использовать функцию `dense_rank()`).

```
SELECT *  
FROM (SELECT E.DEPT_ID, E.LAST_NAME, E.SALARY,  
dense_rank() over (PARTITION BY E.DEPT_ID ORDER BY E.SALARY  
DESC) dr  
FROM S_EMP E)  
WHERE dr <= 3  
ORDER BY DEPT_ID, SALARY DESC;
```

DEPT_ID NUMBER(7)	LAST_NAME VARCHAR2(25)	SALARY NUMBER(11, 2)	DR NUMBER
10	Quick-To-See	1450	1
31	Nagayamn	1400	1
31	Magee	1400	1
32	Velasquez	2550	1
33	Sedeghi	1515	1
34	Nguyen	1525	1
34	Smith	940	2
34	Patel	795	3
41	Maduro	1400	1
41	Urguhart	1200	2
42	Menchu	1250	1
42	Nozaki	1200	2
42	Patel	795	3
43	Biri	1100	1
43	Markarian	850	2
43	Newman	750	3
44	Catchpole	1300	1
44	Chang	800	2
45	Havel	1307	1
45	Schwartz	1100	2
45	Dancs	860	3
50	Ropeburn	1550	1
54	Ngao	1450	1

- Выдать до трех человек с максимальными зарплатами. Если четыре человека имеют одинаковую максимальную зарплату, в ответ не должно выдаваться ни одной строки. Если два сотрудника имеют максимальную зарплату и два — следующую по значению, ответ будет предполагать две строки - два сотрудника с максимальной зарплатой. (Рекомендуется использовать функцию count(*) и выражение range unbounded preceding).

```
SELECT *
FROM (SELECT E.DEPT_ID, E.LAST_NAME, E.SALARY,
COUNT(*) over (PARTITION BY E.DEPT_ID ORDER BY E.SALARY DESC
RANGE UNBOUNDED PRECEDING) cnt
FROM S_EMP E)
WHERE cnt <= 3
ORDER BY DEPT_ID, SALARY DESC;
```

DEPT_ID NUMBER(7)	LAST_NAME VARCHAR2(25)	SALARY NUMBER(11, 2)	CNT NUMBER
10	Quick-To-See	1450	1
31	Nagayamn	1400	2
31	Magee	1400	2
32	Velasquez	2550	1
33	Sedeghi	1515	1
34	Nguyen	1525	1
34	Smith	940	2
34	Patel	795	3
41	Maduro	1400	1
41	Urguhart	1200	2
42	Menchu	1250	1
42	Nozaki	1200	2
42	Patel	795	3
43	Biri	1100	1
43	Markarian	850	2
43	Newman	750	3
44	Catchpole	1300	1
44	Chang	800	2
45	Havel	1307	1
45	Schwartz	1100	2
45	Dancs	860	3
50	Ropeburn	1550	1
54	Ngao	1450	1

- Отсортировать специалистов по продажам по убыванию зарплат. Вернуть первые три строки. Если в отделе менее трех специалистов по продажам, в результате будет менее трех записей. (Рекомендуется использовать функцию row_number()).

```
SELECT *
FROM (SELECT E.DEPT_ID, E.LAST_NAME, E.SALARY,
ROW_NUMBER() over (PARTITION BY E.DEPT_ID ORDER BY E.SALARY
DESC) RN
FROM S_EMP E)
WHERE RN <= 3
ORDER BY DEPT_ID, SALARY DESC;
```

DEPT_ID NUMBER(7)	LAST_NAME VARCHAR2(25)	SALARY NUMBER(11, 2)	RN NUMBER
10	Quick-To-See	1450	1
31	Nagayamn	1400	1
31	Magee	1400	2
32	Velasquez	2550	1
33	Sedeghi	1515	1
34	Nguyen	1525	1
34	Smith	940	2
34	Patel	795	3
41	Maduro	1400	1
41	Urguhart	1200	2
42	Menchu	1250	1
42	Nozaki	1200	2
42	Patel	795	3
43	Biri	1100	1
43	Markarian	850	2
43	Newman	750	3
44	Catchpole	1300	1
44	Chang	800	2
45	Havel	1307	1
45	Schwartz	1100	2
45	Dancs	860	3
50	Ropeburn	1550	1
54	Ngao	1450	1

2. Необходимо создать отчет, в котором по отделам были бы представлены все сотрудники, причем, для каждого сотрудника выдана дата его приема на работу, за сколько дней до этой даты последний раз принимали сотрудника на работу, и через сколько дней после этого приняли на работу следующего сотрудника.

```

SELECT E.DEPT_ID, E.LAST_NAME, E.START_DATE,
       LAG(E.START_DATE, 1, NULL) OVER (PARTITION BY E.DEPT_ID ORDER
BY E.START_DATE, E.LAST_NAME) AS "LAST_HIRE",
       E.START_DATE - LAG(E.START_DATE, 1, NULL) OVER (PARTITION BY
E.DEPT_ID ORDER BY E.START_DATE, E.LAST_NAME) AS "DAYS_LAST",
       LEAD(E.START_DATE, 1, NULL) OVER (PARTITION BY E.DEPT_ID ORDER
BY E.START_DATE, E.LAST_NAME) AS "NEXT_HIRE",
       LEAD(E.START_DATE, 1, NULL) OVER (PARTITION BY E.DEPT_ID ORDER
BY E.START_DATE, E.LAST_NAME) - E.START_DATE AS "DAYS_NEXT"
FROM S_EMP E
ORDER BY DEPT_ID, START_DATE;

```

DEPT_ID NUMBER(7)	LAST_NAME VARCHAR2(25)	START_DATE DATE	LAST_HIRE DATE	DAYS_LAST NUMBER	NEXT_HIRE DATE	DAYS_NEXT NUMBER
10	Quick-To-See	07.04. 1990 00:00:00	(null)	(null)	(null)	(null)
31	Magee	14.05. 1990 00:00:00	(null)	(null)	17.06. 1991 00:00:00	399
31	Nagayamn	17.06. 1991 00:00:00	14.05. 1990 00:00:00	399	(null)	(null)
32	Velasquez	03.03. 1990 00:00:00	(null)	(null)	(null)	(null)
33	Sedeghi	18.02. 1991 00:00:00	(null)	(null)	(null)	(null)
34	Smith	08.03. 1990 00:00:00	(null)	(null)	17.10. 1990 00:00:00	223
34	Patel	17.10. 1990 00:00:00	08.03. 1990 00:00:00	223	22.01. 1992 00:00:00	462
34	Nguyen	22.01. 1992 00:00:00	17.10. 1990 00:00:00	462	(null)	(null)
41	Urguhart	18.01. 1991 00:00:00	(null)	(null)	07.02. 1992 00:00:00	385
41	Maduro	07.02. 1992 00:00:00	18.01. 1991 00:00:00	385	(null)	(null)
42	Menchu	14.05. 1990 00:00:00	(null)	(null)	09.02. 1991 00:00:00	271
42	Nozaki	09.02. 1991 00:00:00	14.05. 1990 00:00:00	271	06.08. 1991 00:00:00	178
42	Patel	06.08. 1991 00:00:00	09.02. 1991 00:00:00	178	(null)	(null)
43	Biri	07.04. 1990 00:00:00	(null)	(null)	26.05. 1991 00:00:00	414
43	Markarian	26.05. 1991 00:00:00	07.04. 1990 00:00:00	414	21.07. 1991 00:00:00	56
43	Newman	21.07. 1991 00:00:00	26.05. 1991 00:00:00	56	(null)	(null)
44	Chang	30.11. 1990 00:00:00	(null)	(null)	09.02. 1992 00:00:00	436
44	Catchpole	09.02. 1992 00:00:00	30.11. 1990 00:00:00	436	(null)	(null)
45	Havel	27.02. 1991 00:00:00	(null)	(null)	17.03. 1991 00:00:00	18
45	Dancs	17.03. 1991 00:00:00	27.02. 1991 00:00:00	18	09.05. 1991 00:00:00	53
45	Schwart,z	09.05. 1991 00:00:00	17.03. 1991 00:00:00	53	(null)	(null)
50	Ropeburn	04.03. 1990 00:00:00	(null)	(null)	(null)	(null)
54	Ngao	08.03. 1990 00:00:00	(null)	(null)	(null)	(null)

3. Запрос с транспонированием. Выдать должности сотрудников отдела в виде столбцов:

```

SELECT DEPT_ID,
       MAX(DECODE(RN, 1, TITLE, NULL)) AS "TITLE_1",
       MAX(DECODE(RN, 2, TITLE, NULL)) AS "TITLE_2",
       MAX(DECODE(RN, 3, TITLE, NULL)) AS "TITLE_3",
       MAX(DECODE(RN, 4, TITLE, NULL)) AS "TITLE_4",
       MAX(DECODE(RN, 5, TITLE, NULL)) AS "TITLE_5"
FROM (SELECT E.DEPT_ID, E.TITLE, SUM(E.SALARY), COUNT(*),
       ROW_NUMBER() OVER (PARTITION BY E.DEPT_ID ORDER BY
       SUM(E.SALARY) DESC) RN
FROM S_EMP E
GROUP BY E.DEPT_ID, E.TITLE)
GROUP BY DEPT_ID;

```

DEPT_ID NUMBER(7)	TITLE_1 VARCHAR2(25)	TITLE_2 VARCHAR2(25)	TITLE_3 VARCHAR2(25)	TITLE_4 VARCHAR2(25)	TITLE_5 VARCHAR2(25)
10	VP, Finance	(null)	(null)	(null)	(null)
31	Sales Representative	VP, Sales	(null)	(null)	(null)
32	President	(null)	(null)	(null)	(null)
33	Sales Representative	(null)	(null)	(null)	(null)
34	Stock Clerk	Sales Representative	(null)	(null)	(null)
41	Stock Clerk	Warehouse Manager	(null)	(null)	(null)
42	Stock Clerk	Warehouse Manager	(null)	(null)	(null)
43	Stock Clerk	Warehouse Manager	(null)	(null)	(null)
44	Warehouse Manager	Stock Clerk	(null)	(null)	(null)
45	Stock Clerk	Warehouse Manager	(null)	(null)	(null)
50	VP, Administration	(null)	(null)	(null)	(null)
54	VP, Operations	(null)	(null)	(null)	(null)

Заключение

В ходе выполнения данной лабораторной работы были изучены и применены на практике аналитические функции SQL Oracle. Они представляют собой инструмент для подсчета данных в таблицах. Были рассмотрены такие задачи как:

- Поиск максимального значения зарплаты сотрудников
- Составление отчета о сотрудниках, относительно их даты приема на работу
- Использование транспонированных запросов (вывод данных в несколько столбцов, исходя из определенной характеристики, в нашем случае – номер отдела)