

## 2 блок - соединение таблиц, операторы работы со множествами, некоррелированные подзапросы, управление транзакциями

2.1. Вывести фамилии сотрудников, их оклад, идентификатор отдела и плотный<sup>1</sup> ранг окладов в пределах отдела, в котором работает сотрудник. Ранг 1 должен быть у максимальной зарплаты.

Пример результата:

DEPARTMENT_ID	LAST_NAME	SALARY	SALARY_RANC
10	Whalen	4400	1
20	Hartstein	13000	1
20	Fay	6000	2
...	...	...	...
50	Chung	3800	10
50	Dilly	3600	11
50	Ladwig	3600	11
50	Rajs	3500	12
...	...	...	...

2.1

```
SELECT emp1.department_id, emp1.last_name, emp1.salary,  
COUNT(emp2.salary)+1 Salary_Rank  
FROM employees emp1  
LEFT OUTER JOIN employees emp2  
ON ((emp1.department_id = emp2.department_id AND emp1.salary <  
emp2.salary)  
or (emp1.department_id is null and emp2.department_id is null AND  
emp1.salary < emp2.salary))  
GROUP BY emp1.department_id, emp1.last_name, emp1.salary  
ORDER BY 1,4;
```

Суть в том, что мы соединяем две таблицы по равенству отдела и большинству зарплаты, либо же по отсутствию отдела. Таким образом, для каждого работника получаем строки со всеми сотрудниками с большей зарплатой, чем у него. Далее с помощью функции count получаем номер зарплаты в отделе (при плотном ранге нас интересуют уникальные зарплаты, при неплотном - без уникальности).

## 2.2 Вывести фамилии сотрудников, их оклад, идентификатор отдела и неплотный<sup>2</sup> ранг окладов в пределах

отдела, в котором работает сотрудник. Ранг 1 должен быть у максимальной зарплаты.

Пример результата:

DEPARTMENT_ID	LAST_NAME	SALARY	SALARY_RANC
10	Whalen	4400	1
20	Hartstein	13000	1
20	Fay	6000	2
...	...	...	...
50	Chung	3800	10
50	Dilly	3600	11
50	Ladwig	3600	11
50	Rajs	3500	13
...	...	...	...

Аналогично задаче выше

2.3. Вывести фамилии тех сотрудников, чей оклад выше среднего в отделе, в котором они работают. В результат вывести:

- 1) идентификатор отдела, в котором работает сотрудник;
- 2) средний оклад по отделу, округлённый до целого числа;
- 3) фамилию сотрудника;
- 4) оклад сотрудника.

Сведения должны быть отсортированы по возрастанию:

- 1) по идентификатору отдела, к которому приписан сотрудник;

## 2.3 2) по окладу, установленному сотруднику

Пример результата:

DEPARTMENT_ID	AVG_SAL	LAST_NAME	SALARY
20	9500	Hartstein	13000
30	4150	Raphaely	11000
50	3476	Rajs	3500
50	3476	Ladwig	3600
...	...	...	...
100	8601	Faviet	9000
100	8601	Greenberg	12008
110	10154	Higgins	12008

```
select emp1.department_id,round(emp2."Average") avg_sal,emp1.last_name,
emp1.salary
from (select employee_id,last_name,salary,department_id from employees)
emp1
join
```

```
(select avg(salary) "Average",department_id from employees group by
department_id) emp2
on (emp1.department_id=emp2.department_id and
emp1.salary>emp2."Average")
order by 1,4;
```

Находим среднее по отделам с группировкой, далее находим зарплату сотрудников и соединяем по условию. В конечном итоге группируем, как указано в условии.

2.4. Вывести фамилии сотрудников, начальники которых работают в другой стране. В результат вывести:

- 1) идентификатор сотрудника;
- 2) фамилию сотрудника;
- 3) название страны, где расположен офис сотрудника;
- 4) идентификатор непосредственного руководителя сотрудника;
- 5) фамилию непосредственного руководителя сотрудника;
- 6) название страны, где расположен офис непосредственного руководителя сотрудника.

Пример результата:

2.4

EMP_I D	EMP_NA ME	EMP_COUN TRY	MAN_ ID	MAN_NA ME	MAN_COUN TRY
201	Hartstein	Canada	100	King	United States of America
204	Baer	Germany	101	Kochhar	United States of America
148	Cambraul t	United Kingdom	100	King	United States of America
...	...	...	...	...	...

```
select emp.employee_id emp_id,emp.last_name
emp_name,emp.country_name emp_country,
man.employee_id man_id,man.last_name man_name,man.country_name
man_country
from ((select
emp1.employee_id,emp1.last_name,cou1.country_name,emp1.manager_id
from ((select employee_id,last_name, department_id,manager_id from
employees) emp1
join (select department_id, location_id from departments) dep1 using
(department_id) join (select location_id,country_id from locations) loc1
using (location_id) join (select country_id, country_name from countries) cou1
using(country_id))) emp
JOIN
```

```
(select emp2.employee_id,emp2.last_name,cou2.country_name
from ((select employee_id,last_name, department_id,manager_id from
employees) emp2
join (select department_id, location_id from departments) dep2 using
(department_id) join (select location_id,country_id from locations) loc2
using (location_id) join (select country_id, country_name from countries) cou2
using(country_id))) man
on (emp.manager_id=man.employee_id and
emp.country_name!=man.country_name));
```

Соединяем таблицы employees->departments->locations->countries с целью получения страны работы сотрудника, далее делаем то же самое для менеджера и соединяем таблицы по равенству employee\_id=manager\_id и неравенства стран работы.

2.6. Создать запрос для определения списка городов, в которых расположены департаменты, суммарная заработная плата в которых выше средней суммарной заработной платы в департаментах этого города. Если к отделу не приписано ни одного сотрудника, считать суммарную заработную плату в этом отделе равной нулю и учитывать её при подсчёте среднего значения по городу. В результат вывести:

- 1) название города;
- 2) название департамента;
- 3) среднюю суммарную зарплату в городе, округлённую до двух знаков после запятой;
- 4) суммарную зарплату в департаменте.

Пример результата:

CITY	DEPARTMENT_	AVG	SUM_DEPT_
	NAME	_CITY_SUM_SAL	SAL
Seattle	Purchasing	7580,95	24900
Seattle	Accounting	7580,95	20300
Seattle	Finance	7580,95	51600
Seattle	Executive	7580,95	58000

2.6

```
select sum_dep.city city, sum_dep.department_name
department_name,aver_sal."Avg_Sal" avg_city_sum_sal,sum_dep."Sum_sal"
sum_dept_sal from ((select
emp1.*,loc1.location_id,dep1.department_name,loc1.city from ((select
department_id, sum(nvl(salary,0)) "Sum_sal",count(employee_id)
"Emp_count" from employees group by department_id) emp1 join (select
location_id,department_name,department_id from departments) dep1
```

```

on (emp1.department_id=dep1.department_id or (emp1.department_id is null
and dep1.department_id is null)) join (select location_id,city from locations)
loc1 on (dep1.location_id=loc1.location_id))) sum_dep
JOIN
(select avg("Sum_sal") "Avg_Sal",city from (select
emp1.*,loc1.location_id,dep1.department_name,loc1.city from ((select
department_id, sum(nvl(salary,0)) "Sum_sal" from employees group by
department_id) emp1 join (select location_id,department_name,department_id
from departments) dep1
on (emp1.department_id=dep1.department_id or (emp1.department_id is null
and dep1.department_id is null)) join (select location_id,city from locations)
loc1 on (dep1.location_id=loc1.location_id))) sum_dep
group by city) aver_sal
on (sum_dep.city=aver_sal.city and sum_dep."Sum_sal">aver_sal."Avg_Sal"));

```

Храним для отделов с городами их суммарную зарплату, далее соединяем с условием. Я не учитываю, что сотрудников не приписано, так как проверил этот столбец, и там 0 нет в принципе, округлить до 2 знаков можно с маской, так там до десятых округление.

2.7. Выбрать сотрудников компании, оклады которых наиболее близки к среднему окладу по подразделению, к которому они приписаны. Требуется вывести:

- 1) идентификатор сотрудника;
- 2) фамилию сотрудника;
- 3) идентификатор должности, которую занимает сотрудник;
- 4) идентификатор подразделения, к которому приписан сотрудник;
- 5) оклад, установленный сотруднику;
- 6) средний оклад по подразделению, к которому приписан

**2.7** сотрудник (округлить до целых).

Сведения должны быть отсортированы по возрастанию:

- 1) по идентификатору подразделения, к которому приписан сотрудник;
- 2) по окладу, установленному сотруднику;
- 3) по фамилии сотрудника.

Пример результата:

EMPLOYEE E_ID	LAST_NA ME	JOB_I D	DEPARTMENT _ID	SALA RY	AVG_S AL
200	Whalen	AD_AS ST	10	4400	4400
202	Fay	MK_RE P	20	6000	9500
201	Hartstein	MK_MA N	20	13000	9500
...	...	...	...	...	...

```

select tab1.employee_id employee_id, tab1.last_name last_name, tab1.job_id
job_id, tab1.department_id department_id, tab1.salary salary, tab1."Avg_Sal"
avg_sal
from ((select emp_sum.employee_id,
emp_sum.last_name,emp_sum.job_id,emp_sum.department_id,emp_sum.sala
ry,dep_sum."Avg_Sal",abs(emp_sum.salary-dep_sum."Avg_Sal") "Razn"
from (select employee_id,last_name,department_id, job_id,salary
from employees) emp_sum
join
(select job_id,department_id,round(avg(salary)) "Avg_Sal"
from employees
group by department_id,job_id) dep_sum
on (emp_sum.department_id=dep_sum.department_id and
emp_sum.job_id=dep_sum.job_id)) tab1
JOIN

```

```

(select
emp_sum.job_id,emp_sum.department_id,min(abs(emp_sum.salary-dep_sum.
"Avg_Sal")) "Razn"
from (select employee_id,last_name,department_id, job_id,salary
from employees) emp_sum
join
(select job_id,department_id,round(avg(salary)) "Avg_Sal"
from employees
group by department_id,job_id) dep_sum
on (emp_sum.department_id=dep_sum.department_id and
emp_sum.job_id=dep_sum.job_id)
group by emp_sum.department_id,emp_sum.job_id) tab2
on (tab1.department_id=tab2.department_id and tab1.job_id=tab2.job_id and
tab1."Razn"=tab2."Razn"))
order by 4,5,1;

```

Последовательно находим средний оклад по подразделениям и аналогично для работников. Далее соединяем по условию равенства отделов и должностей.

2.12. Написать запрос, выдающий сведения о нарастающей сумме окладов в пределах отдела.

Требуется вывести:

- 1) идентификатор отдела;
- 2) идентификатор сотрудника;
- 3) фамилию сотрудника;
- 4) оклад сотрудника;
- 5) накопленную сумму окладов.

Результат отсортировать по столбцам 1 и 4, перечисленным выше.

Пример результата:

DEPARTMENT_ID	EMPLOYEE_ID	LAST_NAME	SALARY	SUM_SAL
10	200	Whalen	4400	4400
20	202	Fay	6000	6000
20	201	Hartstein	13000	19000
...	...	...	...	...
50	132	Olson	2100	2100
50	128	Markle	2200	4300
50	136	Philtanker	2200	6500
...	...	...	...	...

2.12

```

select *
from (select tab1.department_id,tab1.last_name,tab1.salary,sum(tab2.salary)

```

```

from
((SELECT emp1.department_id, emp1.last_name, emp1.salary,
COUNT(emp2.salary)+1 Salary_Rank
FROM employees emp1
LEFT OUTER JOIN employees emp2
ON ((emp1.department_id = emp2.department_id AND emp1.salary <=
emp2.salary)
or (emp1.department_id is null and emp2.department_id is null AND
emp1.salary <= emp2.salary))
GROUP BY emp1.department_id, emp1.last_name, emp1.salary
ORDER BY 1,4) tab1
JOIN
(SELECT emp1.department_id, emp1.last_name, emp1.salary,
COUNT(emp2.salary)+1 Salary_Rank
FROM employees emp1
LEFT OUTER JOIN employees emp2
ON ((emp1.department_id = emp2.department_id AND emp1.salary <=
emp2.salary)
or (emp1.department_id is null and emp2.department_id is null AND
emp1.salary <= emp2.salary))
GROUP BY emp1.department_id, emp1.last_name, emp1.salary
ORDER BY 1,4) tab2
on (tab1.department_id=tab2.department_id and tab1.salary>=tab2.salary and
tab1.salary_rank<=tab2.salary_rank))
group by tab1.department_id,tab1.last_name,tab1.salary
order by 1,2,3)
order by 1,4;

```

Додумать надо, если ранг одинаковый - какое то условие добавить, чтобы накапливалось - а то пока не получилось отличить людей с одинаковыми зарплатами

2.8. Проверить столбцы First\_name, Last\_name, Salary таблицы Employees на уникальность значений и вывести все строки таблицы, в которых хотя бы в одном столбце встречается значение, которое в этом столбце не уникально (встречается несколько раз).

Пример результата:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	E_MAIL	...	SALARY	...
198	Donald	OConnell	DOCONNEL	...	2600	...
199	Douglas	Grant	DGRANT	...	2600	...
200	Jennifer	Whalen	JWHALEN	...	4400	...
...	...	...	...	...	...	...

2.8



```

select emp1.*
from employees emp1 join employees emp2
on (emp1.first_name=emp2.first_name and
emp1.employee_id!=emp2.employee_id)
union
select emp1.*
from employees emp1 join employees emp2
on (emp1.last_name=emp2.last_name and
emp1.employee_id!=emp2.employee_id)
union
select emp1.*
from employees emp1 join employees emp2
on (emp1.salary=emp2.salary and emp1.employee_id!=emp2.employee_id);

```

Соединяю таблицу с собой через равенство указанных столбцов и неравенство номеров сотрудников (чтобы с собой сотрудник не соединялся). Далее использую union для объединения результатов.

## 2.9. Создать запрос для вывода фамилий, последних должностей и дат приема на работу сотрудников, информация о работе которых в некоторые временные интервалы отсутствует.

Пример результата:

LAST_NAME	JOB_ID	HIRE_DATE
Raphaely	PU_MAN	07.12.94
Kaufling	ST_MAN	01.05.95
Whalen	AD_ASST	17.09.87

```

select *
from
(select last_name,job_id,to_char(start_date,'dd.mm.yy') hire_date
FROM(select * from
((select tab1.employee_id,tab1.last_name
FROM
((select
emp.employee_id,emp.last_name,emp.job_id,j_his.start_date,j_his.end_date
from employees emp
join job_history j_his
on (emp.employee_id=j_his.employee_id) order by 1,3) tab1
JOIN
(select
emp.employee_id,emp.last_name,emp.job_id,j_his.start_date,j_his.end_date
from employees emp
join job_history j_his

```

```

on (emp.employee_id=j_his.employee_id) order by 1,3) tab2
ON (tab1.end_date+1!=tab2.start_date and
tab1.employee_id=tab2.employee_id and tab1.start_date<tab2.start_date)))
tab_emp
join job_history on (tab_emp.employee_id=job_history.employee_id)) order by
5 desc))
where rownum=1;

```

Используя тот факт, что можно соединить таблицу с собой, чтобы понять, все ли промежутки по датам заполнены по равенству конечной и начальной дат.

2.13. Написать команду для удаления одной из двух записей, которые отличаются лишь значениями в двух столбцах. При этом записи должны удовлетворять условиям:

$R1.Col1 = R2.Col2$  и  $R2.Col1 = R1.Col2$

Например, для таблицы:

Col1	Col2	Col3	Col4
one	two	three	four
two	one	three	four
two	four	one	three
one	two	four	five

2.13

Результат должен быть такой:

Col1	Col2	Col3	Col4
one	two	three	four
two	four	one	three
one	two	four	five

```

select col1,column1,column2,column3 from table_test
minus
select tt2.col1,tt2.column1,tt2.column2,tt2.column3
from (select col1,column1,column2,column3,rownum r1 from table_test) tt1
join (select col1,column1,column2,column3,rownum r2 from table_test) tt2
on (tt1.Col1=tt2.Column1 and tt1.Column1=tt2.Col1 and
tt1.Column2=tt2.Column2 and tt1.Column3=tt2.Column3 and r1<r2);

```

Используя соединение с условиями, учитываю rownum, чтобы из каждой пары строк хранить одну.

2.14. Дана таблица из трёх столбцов, в каждой ячейке которой может содержаться любое количество любых символов. Необходимо вывести содержимое таблицы, выполнив следующее условие: записи, которые могут быть получены из других записей перестановкой значений в столбцах, должны выводиться только один раз.

Например, для таблицы:

Col1	Col2	Col3
ab	a	a
a	b	b
a	a	b
a	de	tu
de	tu	a
a	tu	de
m	l	3n
l	3n	m

Один из вариантов результата такой:

Col1	Col2	Col3
ab	a	a
a	b	b
a	a	b
de	tu	a
l	3n	m

2.14

```
select "Col1","Col2","Col3" from tab_test
minus
select tt2."Col1",tt2."Col2",tt2."Col3"
from (select "Col1","Col2","Col3",rownum r1 from tab_test) tt1 cross join
(select "Col1","Col2","Col3",rownum r2 from tab_test) tt2
where ((tt1."Col1"||tt1."Col2"||tt1."Col3"=tt2."Col1"||tt2."Col3"||tt2."Col2")
or (tt1."Col1"||tt1."Col2"||tt1."Col3"=tt2."Col2"||tt2."Col1"||tt2."Col3")
or (tt1."Col1"||tt1."Col2"||tt1."Col3"=tt2."Col2"||tt2."Col3"||tt2."Col1")
or (tt1."Col1"||tt1."Col2"||tt1."Col3"=tt2."Col3"||tt2."Col1"||tt2."Col2")
or (tt1."Col1"||tt1."Col2"||tt1."Col3"=tt2."Col3"||tt2."Col2"||tt2."Col1")) and
r1>r2;
```

Решается аналогично предыдущей, только учитываются все возможные конкатенации 3х столбцов (все комбинации кроме текущей, так как строки в таблице не могут быть идентичными).

2.15. Из таблицы Employees необходимо выбрать такие пары окладов, суммы которых также содержатся в этой таблице. Также необходимо вывести идентификаторы сотрудников, с окладами, удовлетворяющими условию задачи.

Результат представить в виде:

Оклады	Сотрудники
3000,14000->17000	187,145->101
3000,14000->17000	187,145->102
3000,14000->17000	197,145->101
3000,14000->17000	197,145->102
...	...

```
select emp1.salary||','||emp2.salary||'-'>'||emp3.salary as "Оклады",
emp1.employee_id||','||emp2.employee_id||'-'>'||emp3.employee_id as
"Сотрудники"
from
((select employee_id,salary from employees) emp1 join
(select employee_id,salary from employees) emp2
on (emp1.employee_id!=emp2.employee_id) join
(select employee_id,salary from employees) emp3
on (emp1.salary+emp2.salary=emp3.salary and
emp1.employee_id!=emp3.employee_id and
emp2.employee_id!=emp3.employee_id));
```

Использую два соединения таблицы с собой, первое для перебора всех возможных пар сотрудников, а второе - для подсчета равной суммы. Далее корректирую вывод.