```
SELECT column, group_function(column) FROM table
[WHERE condition]
[ORDER BY column]
[GROUP BY group_by_expression];
```

```
SELECT job_id, avg(salary)
FROM employees
GROUP BY job_id;
```

JOB_ID	AVG(SALARY)
IT_PROG	5760
AC_MGR	12008
AC_ACCOUNT	8300
ST_MAN	7280
PU_MAN	11000
AD_ASST	4400
AD_VP	17000
SH_CLERK	3215
FI_ACCOUNT	7920
FI_MGR	12008
PU_CLERK	2780

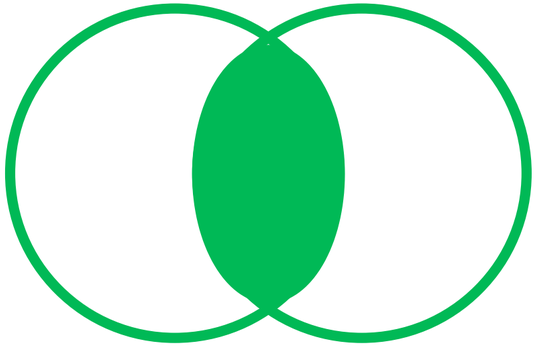
```
SELECT department_id, AVG(salary) FROM employees
WHERE AVG(salary) > 8000
GROUP BY department_id;
```

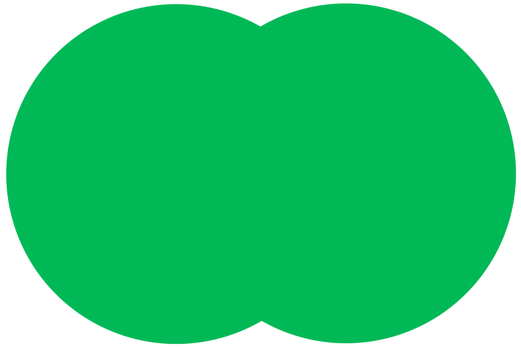
Error report - SQL Error: ORA-00934: групповая функция здесь не разрешена 00934. 00000 - "group function is not allowed here"

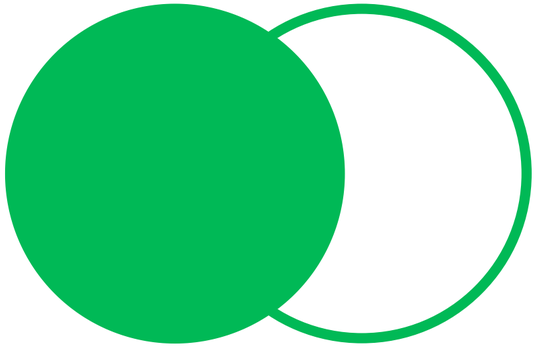
```
SELECT department_id, AVG(salary) FROM employees
GROUP BY department_id
HAVING AVG(salary) > 8000;
```

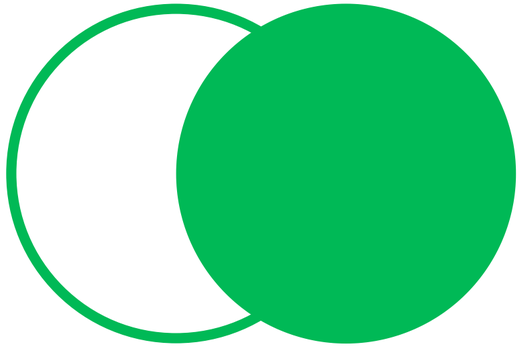
DEPARTMENT_ID	AVG(SALARY)
100	8601,33333
90	19333,3333
20	9500
70	10000
110	10154
80	8955,88235

JOIN









```
SELECT table1.column, table2.column
FROM table1
[NATURAL JOIN table2] |
[JOIN table2 USING (column_name)] |
[JOIN table2
    ON (table1.column_name = table2.column_name)] |
[LEFT|RIGHT|FULL OUTER JOIN table2
    ON (table1.column_name = table2.column_name)] |
[CROSS JOIN table2];
```

```
SELECT employee_id, city, department_name
FROM employees e
JOIN departments d
   ON d.department_id = e.department_id
JOIN locations l
   ON d.location_id = l.location_id;
```

EMPLOYEE_ID	CITY	DEPARTMENT_NAME
200	Seattle	Administration
201	Toronto	Marketing
202	Toronto	Marketing
203	London	Human Resources
204	Munich	Public Relations
205	Seattle	Accounting
206	Seattle	Accounting

```
SELECT employee_id, city, department_name
FROM employees e
JOIN departments d
   ON d.department_id = e.department_id
JOIN locations l
   ON d.location_id = l.location_id;
```

EMPLOYEE_ID	CITY	DEPARTMENT_NAME
200	Seattle	Administration
201	Toronto	Marketing
202	Toronto	Marketing
203	London	Human Resources
204	Munich	Public Relations
205	Seattle	Accounting
206	Seattle	Accounting

```
SELECT employee_id, last_name
FROM employees e
WHERE e.salary > (SELECT MIN(salary) FROM employees
                  WHERE department_id = 10)
```

EMPLOYEE_ID	LAST_NAME
202	Fay
203	Mavris
204	Baer
205	Higgins
206	Gietz

```
SELECT employee_id, job_id
FROM employees
UNION [ALL]
SELECT employee_id, job_id
                FROM job_history;
```

```
INSERT INTO table [(column [,column...])] VALUES (value [,value...]);
```

SELECT * FROM hr.regions;

REGION_ID	REGION_NAME
1	Europe
2	Americas
3	Asia
4	Middle East and Africa

```
INSERT INTO hr.regions VALUES (6,'Russia');
```

REGION_ID	REGION_NAME
1	Europe
2	Americas
3	Asia
4	Middle East and Africa
6	Russia

```
INSERT INTO regions_copy
SELECT *
FROM hr.regions;
```

```
5 rows inserted.
```

```
UPDATE table SET column=value [,column=value...] [WHERE condition];
```

```
UPDATE employees
SET salary = (SELECT max(salary)
                FROM employees);
```

```
SELECT DISTINCT salary
  FROM hr.employees;
```

```
107 rows updated.
```

```
    SALARY
----------
     24000
```

```
DELETE FROM table [WHERE condition];
```

```
SELECT count(*) FROM employees WHERE
department_id IS NULL;

DELETE FROM employees WHERE
department_id IS NULL;

SELECT count(*) FROM employees WHERE
department_id IS NULL;
```

```
  COUNT(*)
----------
         1

1 row deleted.

  COUNT(*)
----------
         0
```

```
TRUNCATE TABLE table;
```

```
SELECT count(*) FROM employees WHERE
department_id IS NULL;

DELETE FROM employees WHERE
department_id IS NULL;

SELECT count(*) FROM employees WHERE
department_id IS NULL;
```

```
  COUNT(*)
----------
         1

1 row deleted.

  COUNT(*)
----------
         0
```

```
SELECT count(1) FROM regions_copy;

INSERT INTO regions_copy
SELECT *
FROM hr.regions;

COMMIT;

SELECT count(1) FROM regions_copy;
```

```
  COUNT(1)
----------
         0

5 rows inserted.

Commit complete.

  COUNT(1)
----------
         5
```

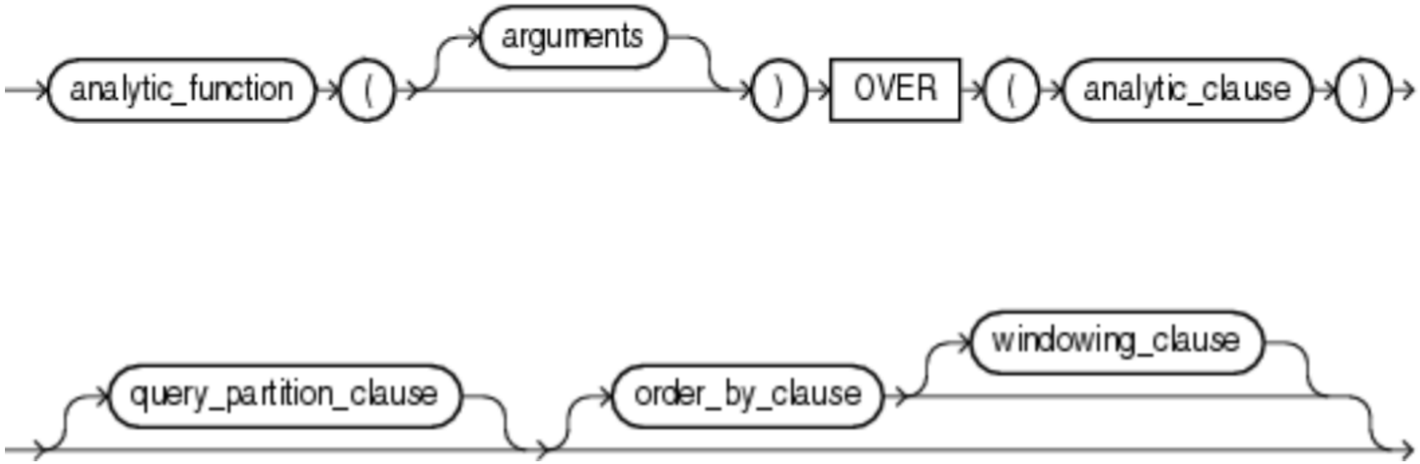
```
CREATE TABLE "HR"."REGIONS_COPY"
( "REGION_ID" NUMBER NOT NULL ENABLE,
"REGION_NAME" VARCHAR2(25 BYTE) )
```

```
CREATE TABLE "HR"."REGIONS_COPY"
   ( "REGION_ID" NUMBER NOT NULL ENABLE,
"REGION_NAME" VARCHAR2(25 BYTE)
   ) SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "USERS" ;
```

- Добавление новой столбец:
  - ALTER TABLE employees ADD (job_type number);
- Изменение столбца
  - ALTER TABLE employees MODIFY (job_type number(1) DEFAULT 0);
- Удаление столбца
  - ALTER TABLE employees DROP COLUMN job_type;
- Помечаем как столбце неиспользуемый
  - ALTER TABLE employees SET UNUSED COLUMN job_type;
- Удаление всех неиспользуемых столбцов
  - ALTER TABLE employees DROP UNUSED COLUMNS;
- Переименование столбца
  - ALTER TABLE employees RENAME COLUMN job_type TO jtype;

```
DROP TABLE [schema.]tablename;
```

Analytic Functions



Ranking

RANK ( ) OVER ([query_partition_clause] order_by_clause)

DENSE_RANK ( ) OVER ([query_partition_clause] order_by_clause)

```
select empno, ename, job, sal,
    rank() OVER (order by sal) as sal_rank,
    dense_rank() OVER (order by sal) as sal_dense_rank,
    row_number() OVER (order by sal) as sal_row_number,
    round(100 * percent_rank() OVER (order by sal), 2) as sal_percent_rank,
    ntile(4) OVER (order by sal) as sal_ntile
from scott.emp order by sal
```

EMPNO	ENAME	JOB	SAL	SAL_RANK	SAL_DENSE_RANK	SAL_ROW_NUMBER	SAL_PERCENT_RANK	SAL_NTILE
7369	SMITH	CLERK	800	1	1	1	0	1
7900	JAMES	CLERK	950	2	2	2	7.69	1
7876	ADAMS	CLERK	1100	3	3	3	15.38	1
7654	MARTIN	SALESMAN	1250	4	4	4	23.08	1
7521	WARD	SALESMAN	1250	4	4	5	23.08	2
7934	MILLER	CLERK	1300	6	5	6	38.46	2
7844	TURNER	SALESMAN	1500	7	6	7	46.15	2
7499	ALLEN	SALESMAN	1600	8	7	8	53.85	2
7782	CLARK	MANAGER	2450	9	8	9	61.54	3
7698	BLAKE	MANAGER	2850	10	9	10	69.23	3
7566	JONES	MANAGER	2975	11	10	11	76.92	3
7902	FORD	ANALYST	3000	12	11	12	84.62	4
7788	SCOTT	ANALYST	3000	12	11	13	84.62	4
7839	KING	PRESIDENT	5000	14	12	14	100	4

```
select empno, ename, job, comm,
      row_number() OVER (order by comm) as comm_row_number
from scott.emp where deptno in (10, 30)
```

EMPNO	ENAME	JOB	COMM	COMM_ROW_NUMBER
7844	TURNER	SALESMAN	0	1
7499	ALLEN	SALESMAN	300	2
7521	WARD	SALESMAN	500	3
7654	MARTIN	SALESMAN	1400	4
7698	BLAKE	MANAGER	–	5
7934	MILLER	CLERK	–	6
7839	KING	PRESIDENT	–	7
7900	JAMES	CLERK	–	8
7782	CLARK	MANAGER	–	9

```
select empno, ename, job, comm,
      row_number() OVER (order by comm NULLS FIRST) as comm_row_number
from scott.emp where deptno in (10, 30)
```

EMPNO	ENAME	JOB	COMM	COMM_ROW_NUMBER
7839	KING	PRESIDENT	–	1
7698	BLAKE	MANAGER	–	2
7782	CLARK	MANAGER	–	3
7934	MILLER	CLERK	–	4
7900	JAMES	CLERK	–	5
7844	TURNER	SALESMAN	0	6
7499	ALLEN	SALESMAN	300	7
7521	WARD	SALESMAN	500	8
7654	MARTIN	SALESMAN	1400	9

Tabibitosan Method

select * from LAB_SAMPLES order by 2

SAMPLE_ID	DATE_TAKEN
1	01-DEC-15
2	02-DEC-15
3	03-DEC-15
4	04-DEC-15
5	07-DEC-15
6	08-DEC-15
7	09-DEC-15
8	10-DEC-15
9	14-DEC-15
10	15-DEC-15
11	16-DEC-15
12	19-DEC-15
13	20-DEC-15

```
select date_taken,
      row_number() over(order by date_taken) as rn,
      date_taken-row_number() over(order by
date_taken) as delta
      from LAB_SAMPLES
order by 1
```

DATE_TAKEN	RN	DELTA
01-DEC-15	1	30-NOV-15
02-DEC-15	2	30-NOV-15
03-DEC-15	3	30-NOV-15
04-DEC-15	4	30-NOV-15
07-DEC-15	5	02-DEC-15
08-DEC-15	6	02-DEC-15
09-DEC-15	7	02-DEC-15
10-DEC-15	8	02-DEC-15
14-DEC-15	9	05-DEC-15
15-DEC-15	10	05-DEC-15
16-DEC-15	11	05-DEC-15
19-DEC-15	12	07-DEC-15
20-DEC-15	13	07-DEC-15

```
select min(date_taken) date_from,
       max(date_taken) date_to,
       count(*) num_samples
from (
  select date_taken,
         date_taken-row_number() over(order by date_taken) as delta
         from LAB_SAMPLES
     )
group by delta
order by 1
```

DATE_FROM	DATE_TO	NUM_SAMPLES
01-DEC-15	04-DEC-15	4
07-DEC-15	10-DEC-15	4
14-DEC-15	16-DEC-15	3
19-DEC-15	20-DEC-15	2

Predicates

```
select planet_name, name, radius
from  moons
where  planet_name in ('Earth','Mars','Saturn')
and rownum < 7
```

PLANET_NAME	NAME	RADIUS
Earth	Moon	1737.1
Mars	Deimos	6.2
Mars	Phobos	11.1
Saturn	Aegaeon	.25
Saturn	Aegir	3
Saturn	Albiorix	16

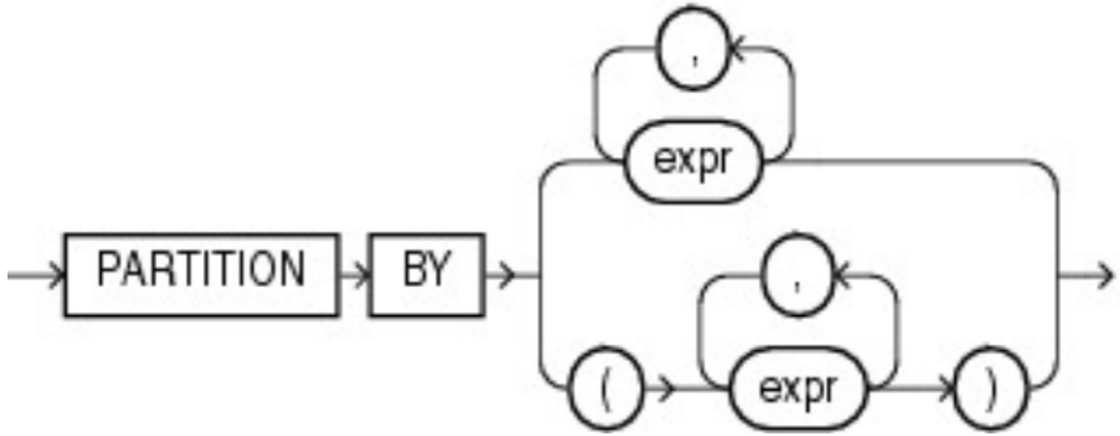
```
select planet_name, name, radius,
       row_number() over ( order by radius desc) as size_rank
from   moons
where  planet_name in ('Earth','Mars','Saturn')
and    row_number() over ( order by radius desc) <= 3
```

```
ORA-30483: window functions are not allowed here
```

```
select *
from (
  select planet_name, name, radius,
         row_number() over ( order by radius desc) as size_rank
  from   moons
  where  planet_name in ('Earth','Mars','Saturn')
)
where size_rank <= 3
```

PLANET_NAME	NAME	RADIUS	SIZE_RANK
Saturn	Titan	2575.5	1
Earth	Moon	1737.1	2
Saturn	Rhea	764.5	3

Partition



```
select *
from countries
order by continent, name
```

NAME	CONTINENT	POPULATION
Algeria	Africa	40400000
Angola	Africa	24383301
Benin	Africa	10653654
Botswana	Africa	2141206
Burkina	Africa	18450494
Burundi	Africa	10114505
Cameroon	Africa	23924000
Cape Verde	Africa	531239
Central African Republic	Africa	4998000

```
select *
from (
  select c.*,
        row_number() over ( order by population desc) as pop_rank
  from   countries c
  where  continent = CONTINENT NAME
)
where pop_rank <= 3
```

NAME	CONTINENT	POPULATION	POP_RANK
Nigeria	Africa	186988000	1
Ethiopia	Africa	92206000	2
Egypt			

NAME	CONTINENT	POPULATION	POP_RANK
Australia	Oceania	24016400	1
Papua New Guinea	Oceania	8083700	2
New Zealand	Oceania	4645120	3

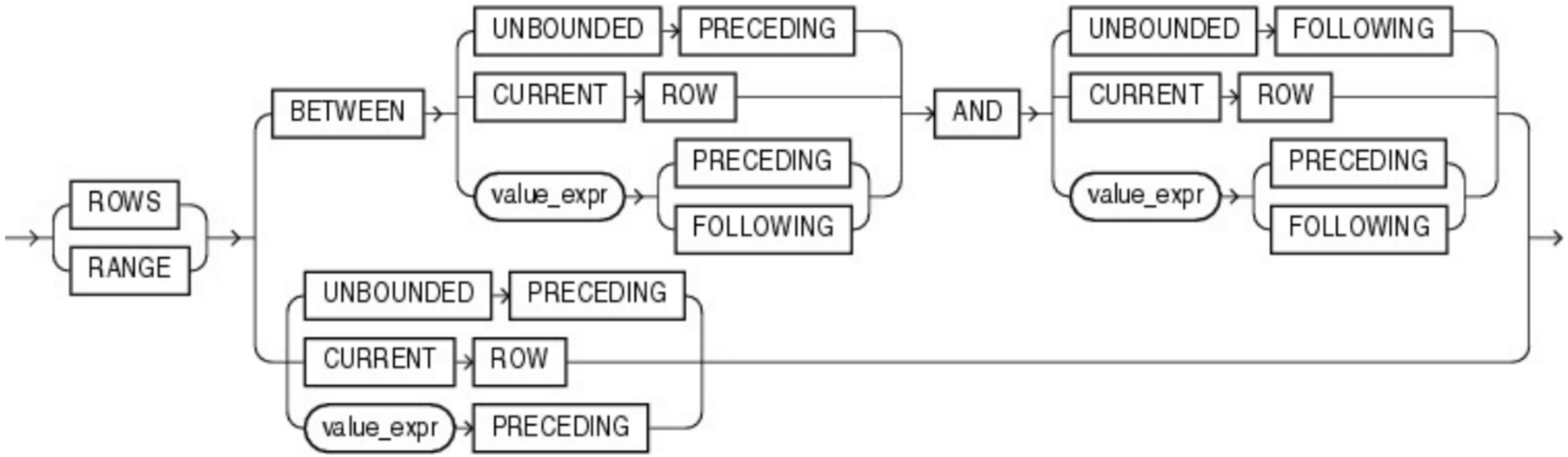
```
select *
from (
  select c.*,
        row_number() over (
            partition by continent
            order by population desc) as pop_rank
  from   countries c
)
where pop_rank <= 3
```

NAME	CONTINENT	POPULATION	POP_RANK
Nigeria	Africa	186988000	1
Ethiopia	Africa	92206000	2
Egypt	Africa	90174100	3
China	Asia	1374150000	1
India	Asia	1282600000	2
Indonesia	Asia	258705000	3
Germany	Europe	81292400	1
France	Europe	66539000	2
United Kingdom	Europe	64800000	3

```
select deptno, job, ename, sal,
        sum(sal) over () total_sal,
        sum(sal) over
           ( partition by deptno) sal_by_dept,
        sum(sal) over
           ( partition by deptno, job) sal_by_dept_job
   from emp
   order by deptno, job
```

DEPTNO	JOB	ENAME	SAL	TOTAL_SAL	SAL_BY_DEPT	SAL_BY_DEPT_JOB
10	CLERK	MILLER	1300	29025	8750	1300
10	MANAGER	CLARK	2450	29025	8750	2450
10	PRESIDENT	KING	5000	29025	8750	5000
20	ANALYST	SCOTT	3000	29025	10875	6000
20	ANALYST	FORD	3000	29025	10875	6000
20	CLERK	ADAMS	1100	29025	10875	1900
20	CLERK	SMITH	800	29025	10875	1900
20	MANAGER	JONES	2975	29025	10875	2975
30	CLERK	JAMES	950	29025	9400	950
30	MANAGER	BLAKE	2850	29025	9400	2850
30	SALESMAN	MARTIN	1250	29025	9400	5600
30	SALESMAN	WARD	1250	29025	9400	5600
30	SALESMAN	ALLEN	1600	29025	9400	5600
30	SALESMAN	TURNER	1500	29025	9400	5600

Window


ROWS
RANGE
BETWEEN
UNBOUNDED
PRECEDING
CURRENT
ROW
value_expr
PRECEDING
FOLLOWING
AND
UNBOUNDED
FOLLOWING
CURRENT
ROW
value_expr
PRECEDING
FOLLOWING
UNBOUNDED
PRECEDING
CURRENT
ROW
value_expr
PRECEDING

```
select *
from water
order by 3 desc
```

NAME	TYPE	SQUARE_KM
Pacific Ocean	Ocean	155557000
Atlantic Ocean	Ocean	76762000
Indian Ocean	Ocean	68556000
Southern Ocean	Ocean	20327000
Arctic Ocean	Ocean	14056000
Mediterranean Sea	Sea	2965800
Caribbean Sea	Sea	2718200
South China Sea	Sea	2319000
Bering Sea	Sea	2291900
Gulf of Mexico	Gulf	1592800
Okhotsk Sea	Sea	1589700
East China Sea	Sea	1249200
Hudson Bay	Bay	1232300
Japan Sea	Sea	1007800
Andaman Sea	Sea	797700
North Sea	Sea	575200
Red Sea	Sea	438000
Baltic Sea	Sea	422200

```
select name, type, square_km,
      sum(square_km) over ( order by square_km desc
        rows between unbounded preceding and current row
) as tot
from water
order by 3 desc
```

NAME	TYPE	SQUARE_KM	TOT
Pacific Ocean	Ocean	155557000	155557000
Atlantic Ocean	Ocean	76762000	232319000
Indian Ocean	Ocean	68556000	300875000
Southern Ocean	Ocean	20327000	321202000
Arctic Ocean	Ocean	14056000	335258000
Mediterranean Sea	Sea	2965800	338223800
Caribbean Sea	Sea	2718200	340942000
South China Sea	Sea	2319000	343261000
Bering Sea	Sea	2291900	345552900
Gulf of Mexico	Gulf	1592800	347145700
Okhotsk Sea	Sea	1589700	348735400
East China Sea	Sea	1249200	349984600
Hudson Bay	Bay	1232300	351216900
Japan Sea	Sea	1007800	352224700
Andaman Sea	Sea	797700	353022400
North Sea	Sea	575200	353597600
Red Sea	Sea	438000	354035600
Baltic Sea	Sea	422200	354457800

```
select name, type, square_km,
      sum(square_km) over (
         partition by type
         order by square_km desc
         rows between 1 preceding and 1 following
) as "3_ROW"
from water
order by decode(type,'Ocean',1,'Sea',2,3), square_km desc
```

NAME	TYPE	SQUARE_KM	3_ROW
Pacific Ocean	Ocean	155557000	232319000
Atlantic Ocean	Ocean	76762000	300875000
Indian Ocean	Ocean	68556000	165645000
Southern Ocean	Ocean	20327000	102939000
Arctic Ocean	Ocean	14056000	34383000
Mediterranean Sea	Sea	2965800	5684000
Caribbean Sea	Sea	2718200	8003000
South China Sea	Sea	2319000	7329100
Bering Sea	Sea	2291900	6200600
Okhotsk Sea	Sea	1589700	5130800
East China Sea	Sea	1249200	3846700
Japan Sea	Sea	1007800	3054700
Andaman Sea	Sea	797700	2380700
North Sea	Sea	575200	1810900
Red Sea	Sea	438000	1435400
Baltic Sea	Sea	422200	860200
Gulf of Mexico	Gulf	1592800	1592800
Hudson Bay	Bay	1232300	1232300

```
select
  dte,
  tweet_millions
from tweets
order by dte
```

01-OCT-14	315.04
01-DEC-14	329.49
01-JAN-15	340
01-APR-15	376.6
01-MAY-15	388.8
01-JUL-15	413.2
01-AUG-15	425.41
01-OCT-15	449.81
01-NOV-15	462.01
01-DEC-15	474.21

```
elect  dte, tweet_millions,
  round(
    avg(tweet_millions) over
        ( order by dte
          range between interval '6' month preceding and
current row )) mov_avg
from tweets
order by dte
```

01-OCT-14	315.04	293
01-DEC-14	329.49	308
01-JAN-15	340	319
01-APR-15	376.6	340
01-MAY-15	388.8	359
01-JUL-15	413.2	380
01-AUG-15	425.41	401
01-OCT-15	449.81	411
01-NOV-15	462.01	428
01-DEC-15	474.21	445

select * from life_expectancy order by 1

RECORDED_YEAR	AGE
01-JAN-1543	33.94
01-JAN-1548	38.82
01-JAN-1553	39.59
01-JAN-1558	22.38
01-JAN-1563	36.66
.....	
01-JAN-2004	78.93
01-JAN-2005	79.13
01-JAN-2006	79.37
01-JAN-2007	79.56
01-JAN-2008	79.68
01-JAN-2009	80.18
01-JAN-2010	80.41
01-JAN-2011	80.8

```
select
  recorded_year, age,
  first_value(age) over
    ( order by recorded_year
      range between interval '5' year preceding
            and interval '5' year following ) as first_val
from life_expectancy
order by 1
```

01-JAN-1999	77.38	76.72
01-JAN-2000	77.86	76.62
01-JAN-2001	78.14	76.88
01-JAN-2002	78.26	77.14
01-JAN-2003	78.35	77.28
01-JAN-2004	78.93	77.38
01-JAN-2005	79.13	77.86
01-JAN-2006	79.37	78.14
01-JAN-2007	79.56	78.26
01-JAN-2008	79.68	78.35
01-JAN-2009	80.18	78.93
01-JAN-2010	80.41	79.13
01-JAN-2011	80.8	79.37

```
select
  recorded_year, age,
  first_value(age) over
    ( order by recorded_year
      range between interval '5' year preceding
            and interval '5' year following ) as first_val,
  last_value(age) over
    ( order by recorded_year
      range between interval '5' year preceding
            and interval '5' year following ) as last_val
from life_expectancy
order by 1
```

01-JAN-1999	77.38	76.72	78.93
01-JAN-2000	77.86	76.62	79.13
01-JAN-2001	78.14	76.88	79.37
01-JAN-2002	78.26	77.14	79.56
01-JAN-2003	78.35	77.28	79.68
01-JAN-2004	78.93	77.38	80.18
01-JAN-2005	79.13	77.86	80.41
01-JAN-2006	79.37	78.14	80.8
01-JAN-2007	79.56	78.26	80.8
01-JAN-2008	79.68	78.35	80.8
01-JAN-2009	80.18	78.93	80.8
01-JAN-2010	80.41	79.13	80.8
01-JAN-2011	80.8	79.37	80.8

select * from trips

TRIP_DATE	TRIP_COUNT
01-JUL-14	21228
02-JUL-14	26480
03-JUL-14	21597
04-JUL-14	14148
05-JUL-14	10890
06-JUL-14	11443
07-JUL-14	18280
08-JUL-14	25763
09-JUL-14	27817

```
select
  trip_date,
  trip_count,
    nth_value(trip_count,2) over (
      order by trip_count
      range between unbounded preceding and unbounded following ) as low_2
from trips
order by 1
```

TRIP_DATE	TRIP_COUNT	LOW_2
01-JUL-14	21228	11443
02-JUL-14	26480	11443
03-JUL-14	21597	11443
04-JUL-14	14148	11443
05-JUL-14	10890	11443
06-JUL-14	11443	11443
07-JUL-14	18280	11443
08-JUL-14	25763	11443
09-JUL-14	27817	11443

```
select
  trip_date,
  trip_count,
    nth_value(trip_count,2 from last) over (
      order by trip_count
      range between unbounded preceding and unbounded following ) as hi_2
from trips
order by 1
```

TRIP_DATE	TRIP_COUNT	HI_2
01-JUL-14	21228	42319
02-JUL-14	26480	42319
03-JUL-14	21597	42319
04-JUL-14	14148	42319
05-JUL-14	10890	42319
06-JUL-14	11443	42319
07-JUL-14	18280	42319
08-JUL-14	25763	42319
09-JUL-14	27817	42319
...	...	...
05-SEP-14	42319	42319
...	...	...
13-SEP-14	43205	42319

```
elect empno, ename, job, hiredate, sal , deptno
from external_emp
order by seq
```

EMPNO	ENAME	JOB	HIREDATE	SAL	DEPTNO
7782	CLARK	MANAGER	09-JUN-81	2450	10
7839	KING	PRESIDENT	17-NOV-81	5000	-
7934	MILLER	CLERK	23-JAN-82	1300	-
7369	SMITH	CLERK	17-DEC-80	800	20
7876	ADAMS	CLERK	12-JAN-83	1100	-
7902	FORD	ANALYST	03-DEC-81	3000	-
7566	JONES	MANAGER	02-APR-81	2975	-
7788	SCOTT	ANALYST	09-DEC-82	3000	-
7499	ALLEN	SALESMAN	20-FEB-81	1600	30
7521	WARD	SALESMAN	22-FEB-81	1250	-
7654	MARTIN	SALESMAN	28-SEP-81	1250	-
7698	BLAKE	MANAGER	01-MAY-81	2850	-
7844	TURNER	SALESMAN	08-SEP-81	1500	-
7900	JAMES	CLERK	03-DEC-81	950	-

```
select empno, ename, job, hiredate, sal ,
      last_value(deptno) over (order by SEQ)  deptno
from external_emp
order by seq
```

EMPNO	ENAME	JOB	HIREDATE	SAL	DEPTNO
7782	CLARK	MANAGER	09-JUN-81	2450	10
7839	KING	PRESIDENT	17-NOV-81	5000	-
7934	MILLER	CLERK	23-JAN-82	1300	-
7369	SMITH	CLERK	17-DEC-80	800	20
7876	ADAMS	CLERK	12-JAN-83	1100	-
7902	FORD	ANALYST	03-DEC-81	3000	-
7566	JONES	MANAGER	02-APR-81	2975	-
7788	SCOTT	ANALYST	09-DEC-82	3000	-
7499	ALLEN	SALESMAN	20-FEB-81	1600	30
7521	WARD	SALESMAN	22-FEB-81	1250	-
7654	MARTIN	SALESMAN	28-SEP-81	1250	-
7698	BLAKE	MANAGER	01-MAY-81	2850	-
7844	TURNER	SALESMAN	08-SEP-81	1500	-
7900	JAMES	CLERK	03-DEC-81	950	-

```
select empno, ename, job, hiredate, sal ,
  last_value(deptno IGNORE NULLS) over (order by SEQ)  deptno
from external_emp
order by seq
```

EMPNO	ENAME	JOB	HIREDATE	SAL	DEPTNO
7782	CLARK	MANAGER	09-JUN-81	2450	10
7839	KING	PRESIDENT	17-NOV-81	5000	10
7934	MILLER	CLERK	23-JAN-82	1300	10
7369	SMITH	CLERK	17-DEC-80	800	20
7876	ADAMS	CLERK	12-JAN-83	1100	20
7902	FORD	ANALYST	03-DEC-81	3000	20
7566	JONES	MANAGER	02-APR-81	2975	20
7788	SCOTT	ANALYST	09-DEC-82	3000	20
7499	ALLEN	SALESMAN	20-FEB-81	1600	30
7521	WARD	SALESMAN	22-FEB-81	1250	30
7654	MARTIN	SALESMAN	28-SEP-81	1250	30
7698	BLAKE	MANAGER	01-MAY-81	2850	30
7844	TURNER	SALESMAN	08-SEP-81	1500	30
7900	JAMES	CLERK	03-DEC-81	950	30

```
select time_id,
       sum(amount_sold) sold_per_day
from   sh.sales
group by time_Id
order by 1
```

TIME_ID	SOLD_PER_DAY
01-JAN-98	15406.48
02-JAN-98	19333.41
03-JAN-98	34392.76
04-JAN-98	32013.28
05-JAN-98	128209.37
06-JAN-98	4148.08
07-JAN-98	52989.75
08-JAN-98	22838.89
09-JAN-98	20505.11

```
create or replace
function LAST_CLOSE(p_close_date date)
return number is
begin
  return
    case to_char(p_close_date,'DY')
     when 'SUN' then 2
     when 'MON' then 3
     else 1
    end;
end;
```

```
select
  time_id,
  to_char(time_id,'DY') close_day,
  sum(sold_per_day)
    over (
      order by time_id
      range between LAST_CLOSE(time_id) preceding and 0 following) as close_off
from (
  select time_id,
        sum(amount_sold) sold_per_day
  from   sh.sales
  group by time_Id
  )
```

TIME_ID	CLOSE_DAY	CLOSE_OFF
01-JAN-98	THU	15406.48
02-JAN-98	FRI	34739.89
03-JAN-98	SAT	53726.17
04-JAN-98	SUN	85739.45
05-JAN-98	MON	213948.82
06-JAN-98	TUE	132357.45
07-JAN-98	WED	57137.83
08-JAN-98	THU	75828.64
09-JAN-98	FRI	43344

LAG / LEAD

select * from orders order by 1,2

ORDER_ID	STATUS_DATE	STATUS
11700	03-JAN-16	New
11700	04-JAN-16	Inventory Check
11700	05-JAN-16	Inventory Check
11700	06-JAN-16	Inventory Check
11700	07-JAN-16	Inventory Check
11700	08-JAN-16	Inventory Check
11700	09-JAN-16	Awaiting Signoff
11700	10-JAN-16	Awaiting Signoff
11700	11-JAN-16	Awaiting Signoff
11700	12-JAN-16	In Warehouse
11700	13-JAN-16	In Warehouse
11700	14-JAN-16	In Warehouse
11700	15-JAN-16	Awaiting Signoff
11700	16-JAN-16	Awaiting Signoff
11700	17-JAN-16	Payment Pending
11700	18-JAN-16	Payment Pending
11700	19-JAN-16	Awaiting Signoff
11700	20-JAN-16	Awaiting Signoff
11700	21-JAN-16	Delivery
11700	22-JAN-16	Delivery

```
select
  order_id,
  status_date,
  status,
  lag(status,1) over (partition by order_id order by status_date) lag_status,
  lead(status,1) over (partition by order_id order by status_date) lead_status
from ORDERS
order by 1,2
```

ORDER_ID	STATUS_DATE	STATUS	LAG_STATUS	LEAD_STATUS
11700	03-JAN-16	New	-	Inventory Check
11700	04-JAN-16	Inventory Check	New	Inventory Check
11700	05-JAN-16	Inventory Check	Inventory Check	Inventory Check
11700	06-JAN-16	Inventory Check	Inventory Check	Inventory Check
11700	07-JAN-16	Inventory Check	Inventory Check	Inventory Check
11700	08-JAN-16	Inventory Check	Inventory Check	Awaiting Signoff
11700	09-JAN-16	Awaiting Signoff	Inventory Check	Awaiting Signoff
11700	10-JAN-16	Awaiting Signoff	Awaiting Signoff	Awaiting Signoff
11700	11-JAN-16	Awaiting Signoff	Awaiting Signoff	In Warehouse
11700	12-JAN-16	In Warehouse	Awaiting Signoff	In Warehouse
11700	13-JAN-16	In Warehouse	In Warehouse	In Warehouse
11700	14-JAN-16	In Warehouse	In Warehouse	Awaiting Signoff
11700	15-JAN-16	Awaiting Signoff	In Warehouse	Awaiting Signoff
11700	16-JAN-16	Awaiting Signoff	Awaiting Signoff	Payment Pending
11700	17-JAN-16	Payment Pending	Awaiting Signoff	Payment Pending
11700	18-JAN-16	Payment Pending	Payment Pending	Awaiting Signoff
11700	19-JAN-16	Awaiting Signoff	Payment Pending	Awaiting Signoff
11700	20-JAN-16	Awaiting Signoff	Awaiting Signoff	Delivery
11700	21-JAN-16	Delivery	Awaiting Signoff	Delivery
11700	22-JAN-16	Delivery	Delivery	-

```
select
  order_id,
  status,
  lag(status_date) over (partition by order_id order by status_date)  from_date,
  status_date to_date
from (
  select
    order_id,
    status_date,
    status,
    lag(status) over (partition by order_id order by status_date) lag_status,
    lead(status) over (partition by order_id order by status_date) lead_status
  from ORDERS
  )
where lag_status is null
      or lead_status is null
      or lead_status <> status
order by 1,3 nulls first
```

ORDER_ID	STATUS	FROM_DATE	TO_DATE
11700	New	-	03-JAN-16
11700	Inventory Check	03-JAN-16	08-JAN-16
11700	Awaiting Signoff	08-JAN-16	11-JAN-16
11700	In Warehouse	11-JAN-16	14-JAN-16
11700	Awaiting Signoff	14-JAN-16	16-JAN-16
11700	Payment Pending	16-JAN-16	18-JAN-16
11700	Awaiting Signoff	18-JAN-16	20-JAN-16
11700	Delivery	20-JAN-16	22-JAN-16

```
elect empno, ename, job, hiredate, sal , deptno
from external_emp
order by seq
```

EMPNO	ENAME	JOB	HIREDATE	SAL	DEPTNO
7782	CLARK	MANAGER	09-JUN-81	2450	10
7839	KING	PRESIDENT	17-NOV-81	5000	-
7934	MILLER	CLERK	23-JAN-82	1300	-
7369	SMITH	CLERK	17-DEC-80	800	20
7876	ADAMS	CLERK	12-JAN-83	1100	-
7902	FORD	ANALYST	03-DEC-81	3000	-
7566	JONES	MANAGER	02-APR-81	2975	-
7788	SCOTT	ANALYST	09-DEC-82	3000	-
7499	ALLEN	SALESMAN	20-FEB-81	1600	30
7521	WARD	SALESMAN	22-FEB-81	1250	-
7654	MARTIN	SALESMAN	28-SEP-81	1250	-
7698	BLAKE	MANAGER	01-MAY-81	2850	-
7844	TURNER	SALESMAN	08-SEP-81	1500	-
7900	JAMES	CLERK	03-DEC-81	950	-

```
select empno, ename, job, hiredate, sal ,
      nvl(deptno, lag(deptno) over (order by SEQ))  deptno
from external_emp
order by seq
```

EMPNO	ENAME	JOB	HIREDATE	SAL	DEPTNO
7782	CLARK	MANAGER	09-JUN-81	2450	10
7839	KING	PRESIDENT	17-NOV-81	5000	10
7934	MILLER	CLERK	23-JAN-82	1300	-
7369	SMITH	CLERK	17-DEC-80	800	20
7876	ADAMS	CLERK	12-JAN-83	1100	20
7902	FORD	ANALYST	03-DEC-81	3000	-
7566	JONES	MANAGER	02-APR-81	2975	-
7788	SCOTT	ANALYST	09-DEC-82	3000	-
7499	ALLEN	SALESMAN	20-FEB-81	1600	30
7521	WARD	SALESMAN	22-FEB-81	1250	30
7654	MARTIN	SALESMAN	28-SEP-81	1250	-
7698	BLAKE	MANAGER	01-MAY-81	2850	-
7844	TURNER	SALESMAN	08-SEP-81	1500	-
7900	JAMES	CLERK	03-DEC-81	950	-

```
select empno, ename, job, hiredate, sal ,
      nvl(deptno, lag(deptno IGNORE NULLS) over (order by SEQ))
 deptno
from external_emp
order by seq
```

EMPNO	ENAME	JOB	HIREDATE	SAL	DEPTNO
7782	CLARK	MANAGER	09-JUN-81	2450	10
7839	KING	PRESIDENT	17-NOV-81	5000	10
7934	MILLER	CLERK	23-JAN-82	1300	10
7369	SMITH	CLERK	17-DEC-80	800	20
7876	ADAMS	CLERK	12-JAN-83	1100	20
7902	FORD	ANALYST	03-DEC-81	3000	20
7566	JONES	MANAGER	02-APR-81	2975	20
7788	SCOTT	ANALYST	09-DEC-82	3000	20
7499	ALLEN	SALESMAN	20-FEB-81	1600	30
7521	WARD	SALESMAN	22-FEB-81	1250	30
7654	MARTIN	SALESMAN	28-SEP-81	1250	30
7698	BLAKE	MANAGER	01-MAY-81	2850	30
7844	TURNER	SALESMAN	08-SEP-81	1500	30
7900	JAMES	CLERK	03-DEC-81	950	30

select * from LAB_SAMPLES order by 2

SAMPLE_ID	DATE_TAKEN
1	01-DEC-15
2	02-DEC-15
3	03-DEC-15
4	04-DEC-15
5	07-DEC-15
6	08-DEC-15
7	09-DEC-15
8	10-DEC-15
9	14-DEC-15
10	15-DEC-15
11	16-DEC-15
12	19-DEC-15
13	20-DEC-15

```
select
  date_taken,
  case
    when nvl(lag(date_taken) over (order by date_taken),date_taken)
!= date_taken-1
      then date_taken end loval
from lab_samples
order by 1
```

DATE_TAKEN	LOVAL
01-DEC-15	01-DEC-15
02-DEC-15	-
03-DEC-15	-
04-DEC-15	-
07-DEC-15	07-DEC-15
08-DEC-15	-
09-DEC-15	-
10-DEC-15	-
14-DEC-15	14-DEC-15
15-DEC-15	-
16-DEC-15	-
19-DEC-15	19-DEC-15
20-DEC-15	-

```
select date_taken, max(loval) over (order by date_taken) loval
from (
select date_taken,
 case
   when nvl(lag(date_taken) over (order by date_taken),date_taken) !=
    date_taken-1 then date_taken end loval
from lab_samples )
order by 1
```

DATE_TAKEN	LOVAL
01-DEC-15	01-DEC-15
02-DEC-15	01-DEC-15
03-DEC-15	01-DEC-15
04-DEC-15	01-DEC-15
07-DEC-15	07-DEC-15
08-DEC-15	07-DEC-15
09-DEC-15	07-DEC-15
10-DEC-15	07-DEC-15
14-DEC-15	14-DEC-15
15-DEC-15	14-DEC-15
16-DEC-15	14-DEC-15
19-DEC-15	19-DEC-15
20-DEC-15	19-DEC-15

```
select min(date_taken) range_start, max(date_taken) range_end
from (
  select date_taken,max(loval) over (order by date_taken) loval
  from (
    select date_taken,
      case
        when nvl(lag(date_taken) over (order by date_taken),date_taken) !=
          date_taken-1 then date_taken end loval
    from lab_samples))
group by loval
order by 1
```

RANGE_START	RANGE_END
01-DEC-15	04-DEC-15
07-DEC-15	10-DEC-15
14-DEC-15	16-DEC-15
19-DEC-15	20-DEC-15

LISTAGG

select deptno, ename from emp

DEPTNO	ENAME
20	SMITH
30	ALLEN
30	WARD
20	JONES
30	MARTIN
30	BLAKE
10	CLARK
20	SCOTT
10	KING
30	TURNER
20	ADAMS
30	JAMES
20	FORD
10	MILLER

```
select deptno,
        listagg( ename, ',')
          within group (order by empno) members
   from   emp
group by deptno
order by 1
```

DEPTNO	MEMBERS
10	CLARK,KING,MILLER
20	SMITH,JONES,SCOTT,ADAMS,FORD
30	ALLEN,WARD,MARTIN,BLAKE,TURNER,JAMES

```
select *
from named_moons
order by 2,1,3
```

PLANET	CLASS	MOON	RADIUS
Eris	Dwarf	Dysnomia	342
Haumea	Dwarf	Hiiaka	195
Haumea	Dwarf	Namaka	100
Earth	Earth	Moon	1737.1
Mercury	Inner	-	-
Venus	Inner	-	-
Jupiter	Superior	Adrastea	8.2
Jupiter	Superior	Aitne	1.5
Jupiter	Superior	Amalthea	83.45
Jupiter	Superior	Ananke	14
Jupiter	Superior	Aoede	2
Jupiter	Superior	Arche	1.5
Jupiter	Superior	Autonoe	2

```
elect class,
      planet,
      min(radius) min_r,
      listagg(radius,',') within group ( order by planet ) members
from named_moons
group by class, planet
```

CLASS	PLANET	MIN_R	MEMBERS
Dwarf	Eris	342	342
Dwarf	Haumea	100	100,195
Earth	Earth	1737.1	1737.1
Inner	Venus	-	-
Inner	Mercury	-	-
Superior	Mars	6.2	11.1,6.2
Superior	Pluto	5	23,30.5,5,6.5,603.6
Superior	Saturn	.25	.25,1,1.25,1.6,106.6,11,12,12.8,133,15.3,16,16,198.2,2,2,2.5,20,252.3,2575.5,3,3,3,3,3,3,3,3,3,3,3,3.5,3.5,3
Superior	Uranus	5	10,10,15,15,20.1,21.4,235.8,25.7,32,36,39.8,40.3,46.8,49,5,5,5,5.5,578.9,584.7,6,6,67.6,75,761.4,788.9,81
Superior	Jupiter	1	1,1,1,1,1,1,1,1,1,1,1.5,1.5,1.5,1.5,1.5,1.5,1.6,1.9,1.9,10,14,1560.7,18,1818.1,19,2,2,2,2,2,2,2,2.2,2.5,2.
Superior	Neptune	20	1353.4,170,20,21,210,22,30,31,33,41,75,88,97

```
select class,
      planet,
      min(radius),
      listagg(min(radius),',') within group ( order by planet )
        over ( partition by class ) class_radii
from named_moons
group by class, planet
order by 1,2
```

CLASS	PLANET	MIN(RADIUS)	CLASS_RADII
Dwarf	Eris	342	342,100
Dwarf	Haumea	100	342,100
Earth	Earth	1737.1	1737.1
Inner	Mercury	-	-
Inner	Venus	-	-
Superior	Jupiter	1	1,6.2,20,5,.25,5
Superior	Mars	6.2	1,6.2,20,5,.25,5
Superior	Neptune	20	1,6.2,20,5,.25,5
Superior	Pluto	5	1,6.2,20,5,.25,5
Superior	Saturn	.25	1,6.2,20,5,.25,5
Superior	Uranus	5	1,6.2,20,5,.25,5

KEEP

```
select deptno, min(sal)
from emp
group by deptno
```

DEPTNO	MIN(SAL)
30	950
20	800
10	1300

```
select d.deptno, d.min_sal, e.empno
from emp e,
     (  select deptno, min(sal) min_sal
        from emp
        group by deptno
     ) d
where d.deptno = e.deptno
and   e.sal    = d.min_sal
```

DEPTNO	MIN_SAL	EMPNO
30	950	7900
20	800	7369
10	1300	7934

```
select deptno, min(sal),
        min(empno) KEEP ( dense_rank FIRST order by sal) empno
from emp
group by deptno
```

DEPTNO	MIN(SAL)	EMPNO
10	1300	7934
20	800	7369
30	950	7900

PL / SQL

```
BEGIN
DBMS_OUTPUT.put_line ('Hello World!');
END;
```

```
DECLARE
  l_msg VARCHAR2 (100) := 'Hello World!';
BEGIN
  DBMS_OUTPUT.put_line (l_msg);
EXCEPTION
  WHEN OTHERS
  THEN
    DBMS_OUTPUT.put_line (SQLERRM);
END;
```

```
CREATE OR REPLACE PROCEDURE
hello_world
IS
  l_msg VARCHAR2 (100) := 'Hello World!';
BEGIN
  DBMS_OUTPUT.put_line (l_msg);
END;
```

```
BEGIN
  hello_world;
END;
```

```
DECLARE
  l_name  employees.last_name%TYPE;
BEGIN
  SELECT last_name
    INTO l_name
    FROM employees
   WHERE employee_id = 1;

  DBMS_OUTPUT.put_line (l_name);
END;
```

```
CREATE OR REPLACE PROCEDURE display_dep_salary (
   p_dep_id IN NUMBER
)
IS
BEGIN
  FOR l_row IN (SELECT * FROM employees e where
e.department_id = p_dep_id)
  LOOP
     DBMS_OUTPUT.put_line (l_row.salary);
  END LOOP;
END display_dep_salary;
```

```
BEGIN
   DISPLAY_DEP_SALARY(10);
END;
```

OUTPUT: 4400

```
CREATE OR REPLACE PROCEDURE display_dep_salary (
   p_dep_id IN NUMBER
)
IS
BEGIN
  FOR l_row IN (SELECT * FROM employees e where e.department_id = p_dep_id)
  LOOP
     DBMS_OUTPUT.put_line(l_row.salary);
     IF l_row.salary <= 1000
        THEN
           DBMS_OUTPUT.put_line('Зарплата '|| l_row.salary || ' <= 1000.');
        ELSE
           DBMS_OUTPUT.put_line('Зарплата '|| l_row.salary || ' > 1000.');
    END IF;
  END LOOP;
END display_dep_salary;
```

```
4400
Зарплата 4400 > 1000.
```

Thanks to Connor McDonald

https://blogs.oracle.com/author/connor-mcdonald

Thanks to Tom Kyte

https://blogs.oracle.com/author/tom-kyte

1 HD

@almorozovv

Thank you!