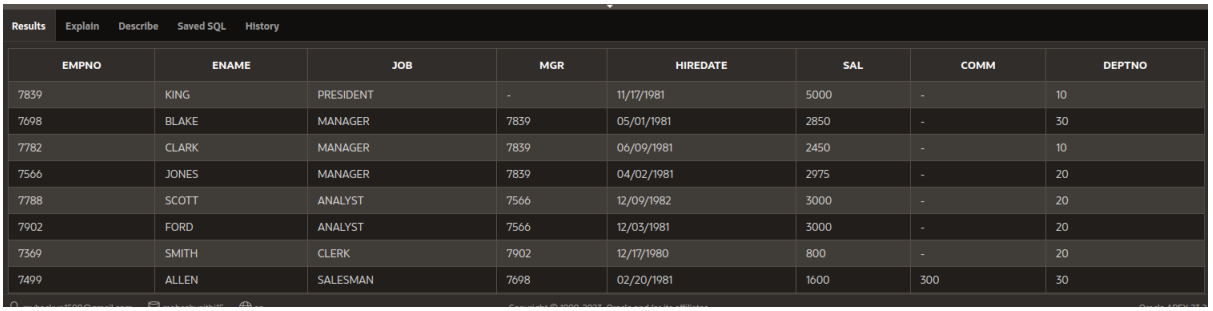
The Complete Oracle SQL Certification:

Select Clause: Basic Select clause as done earlier:

Here in oracle apex workspace we installed dataset emp-dept;

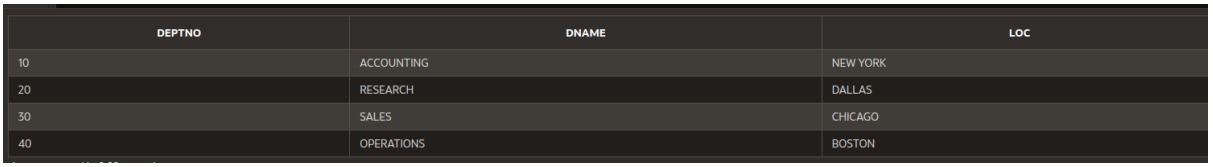
select * from emp;



Results | Explain | Describe | Saved SQL | History

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	-	11/17/1981	5000	-	10
7698	BLAKE	MANAGER	7839	05/01/1981	2850	-	30
7782	CLARK	MANAGER	7839	06/09/1981	2450	-	10
7566	JONES	MANAGER	7839	04/02/1981	2975	-	20
7788	SCOTT	ANALYST	7566	12/09/1982	3000	-	20
7902	FORD	ANALYST	7566	12/03/1981	3000	-	20
7369	SMITH	CLERK	7902	12/17/1980	800	-	20
7499	ALLEN	SALESMAN	7698	02/20/1981	1600	300	30

select * from dept;

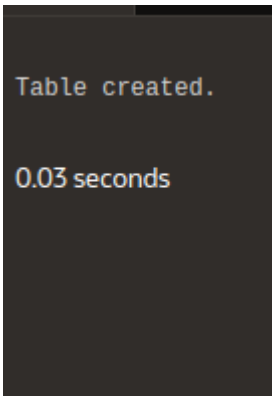


DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

The all Basic where , groupby orderby and filter queries are done earlier.

--- assignment 1 in udemy

--- create table
create table suppliers(supplier_id int,supplier_name varchar(30),city varchar(20),state varchar(20),total_spent numeric(10,2));

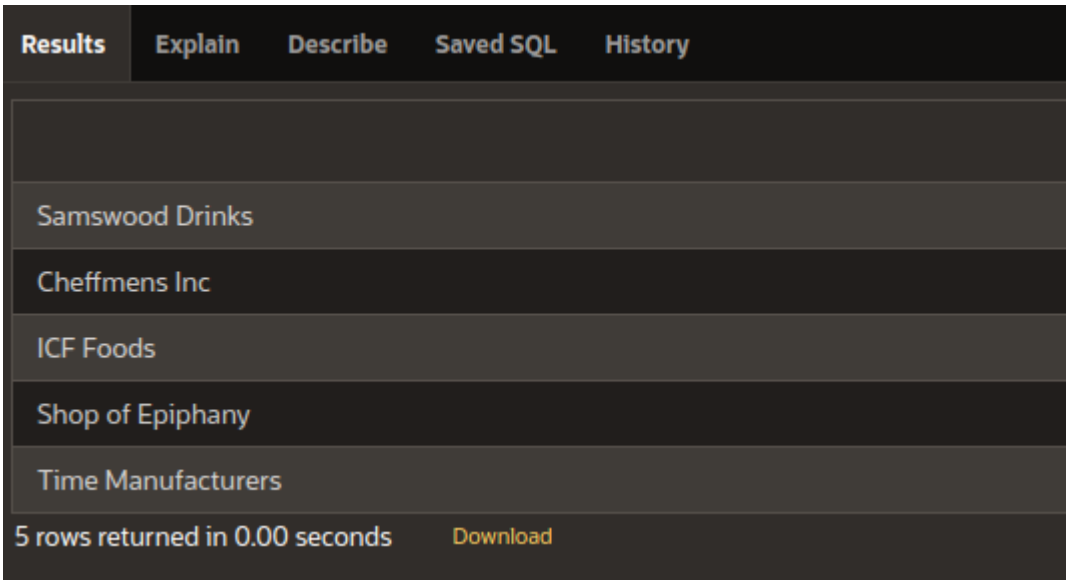


--- inserting values into the table
insert into suppliers values(100,'Shop of Epiphany','Augusta','Georgia',220500.00);
(200,'Instant Assemblers','Valdez','Alaska',37000.00),
(300.'Time Manufacturers','RedWood city','California',90500.00),
(400,'Roundhouse Inc.','New York City','New York',78150.00),

(500,'Smiths & Berries','Portland','Oregon',114600.00),
(600,'Wesson LLC','Yuma','Alaska',32000.00),
(700,'ICF Foods','Orlando','California',78700.00),
(800,'Cheffmens Inc','Toledo','Georgia',187500.00),
(900,'Samswood Drinks','Portland','Georgia',17800.00);

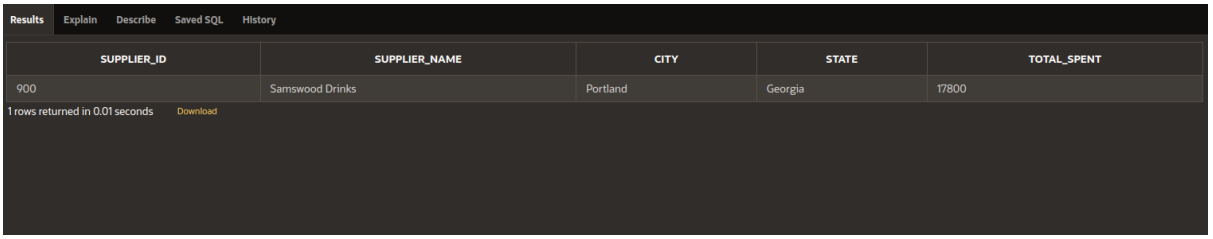
--- Write a query that retrieves suppliers that work in either Georgia or California.

select supplier_name from suppliers where state='Georgia' or state='California';



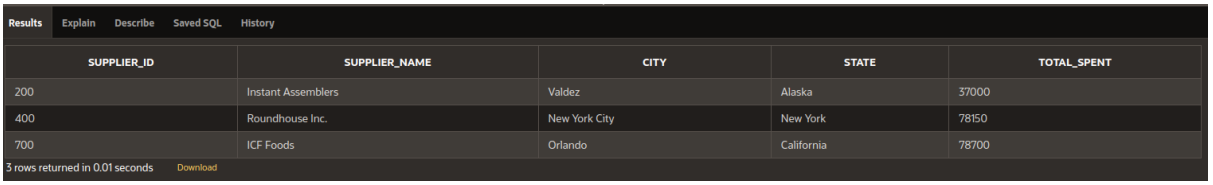
--- Write a query that retrieves suppliers with the characters "wo" and the character "l" or "i" in their name.

select * from suppliers where supplier_name like '%wo%' and supplier_name like '%i%';



--- Write a query that retrieves suppliers on which a minimum of 37,000 and a maximum of 80,000 was spent.

select * from suppliers where total_spent>=37000.00 and total_spent<=80000.00;



/*Write a query that returns the supplier names and the state in which they operate meeting the following conditions:
1)belong in the state Georgia or Alaska
2)the supplier id is 100 or greater than 600
3)the amount spent is less than 100,000 or the amount spent is 220,000*/

select * from suppliers where state='Georgia' or state='Alaska' and
supplier_id>=100 and
total_spent>=100000.00 and total_spent<=220000.00;

Results | Explain | Describe | Saved SQL | History

SUPPLIER_ID	SUPPLIER_NAME	CITY	STATE	TOTAL_SPENT
900	Samswood Drinks	Portland	Georgia	17800
800	Cheffmens Inc	Toledo	Georgia	187500
100	Shop of Epiphany	Augusta	Georgia	220500

3 rows returned in 0.01 seconds Download

Single Row Functions in Oracle SQL:

All the single row functions are done at postgresql . all the concepts are similar.

select ename,hiredate,trunc(hiredate,'MONTH') from emp where
trunc(hiredate,'YEAR')='01/01/1982';

Results | Explain | Describe | Saved SQL | History

ENAME	HIREDATE	TRUNC(HIREDATE,'MONTH')
SCOTT	12/09/1982	12/01/1982
MILLER	01/23/1982	01/01/1982

2 rows returned in 0.01 seconds Download

Conversions are done at postgreSQL;

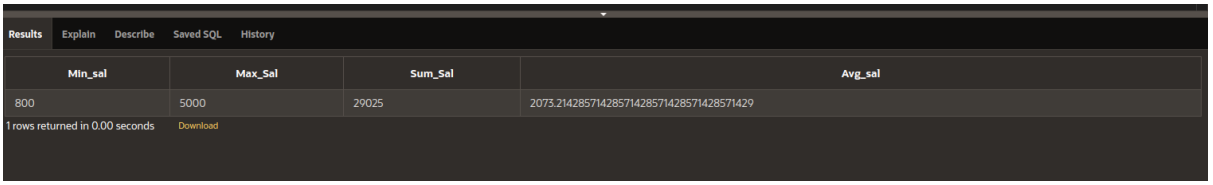
Syntax:
Select conversion() from dual;
Where conversion() → conversion datatype.

--- next day
select next_day(sysdate,1) from dual;
--- last day
select last_day(sysdate) from dual;

Grouping Min,Max,Avg and sum all these concepts are done at postgreSQL

--- min , max , sum , avg

```
select min(sal) as "Min_sal",max(sal) as "Max_Sal",
sum(sal) as "Sum_Sal",avg(sal) as "Avg_sal"
 from emp;
```



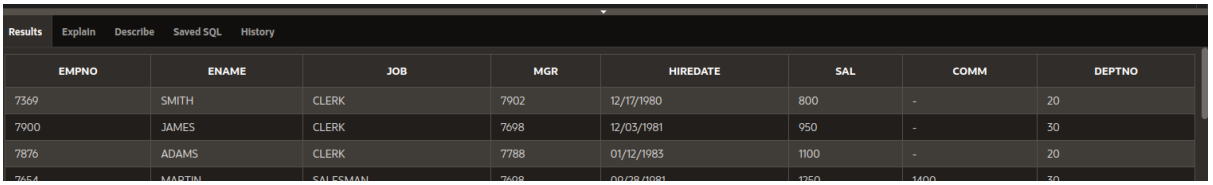
Results | Explain | Describe | Saved SQL | History

Min_sal	Max_Sal	Sum_Sal	Avg_sal
800	5000	29025	2073.214285714285714285714285714285714429

1 rows returned in 0.00 seconds Download

Groupby , Having clause in Oracle SQL:
As same as in postgreSQL

```
--- groupby having
--- to get info of employee
--- group by president and orderby sal asc
select * from emp order by sal asc group by job;
```



Results | Explain | Describe | Saved SQL | History

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	12/17/1980	800	-	20
7900	JAMES	CLERK	7698	12/03/1981	950	-	30
7876	ADAMS	CLERK	7788	01/12/1983	1100	-	20
7654	MARTIN	SALESMAN	7698	09/28/1981	1250	1400	30

SubQueries:

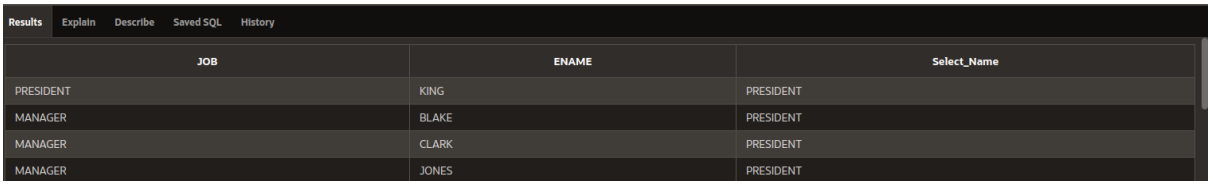
--- subqueries in Oracle

```
select * from (select ename,sal from emp);
```

As same as in the postgreSQL.

--- subqueries in Oracle

```
select job,ename,(select job from emp where ename='KING') as "Select_Name" from emp;
```



Results | Explain | Describe | Saved SQL | History

JOB	ENAME	Select_Name
PRESIDENT	KING	PRESIDENT
MANAGER	BLAKE	PRESIDENT
MANAGER	CLARK	PRESIDENT
MANAGER	JONES	PRESIDENT

JOINS in Oracle SQL;

Inner Joins,Outer Joins,Self Joins are same as in postgreSQL:

--- to join this

select e.empno,e.ename,e.job,d.deptno from emp e,dept d
where e.deptno=d.deptno;

Results | Explain | Describe | Saved SQL | History

EMPNO	ENAME	JOB	DEPTNO
7839	KING	PRESIDENT	10
7782	CLARK	MANAGER	10
7934	MILLER	CLERK	10
7566	JONES	MANAGER	20
7788	SCOTT	ANALYST	20

--- to inner join with condition

select e.empno,e.ename,e.job,d.deptno from emp e inner join dept d on e.deptno=d.deptno
where d.deptno=20

Results | Explain | Describe | Saved SQL | History

EMPNO	ENAME	JOB	DEPTNO
7566	JONES	MANAGER	20
7788	SCOTT	ANALYST	20
7902	FORD	ANALYST	20
7369	SMITH	CLERK	20
7876	ADAMS	CLERK	20

--- to right join with condition

select e.empno,e.ename,e.job,d.deptno from emp e right join dept d on e.deptno=d.deptno
where d.deptno=10;

Results | Explain | Describe | Saved SQL | History | Bottom Splitter

EMPNO	ENAME	JOB	DEPTNO
7839	KING	PRESIDENT	10
7782	CLARK	MANAGER	10
7934	MILLER	CLERK	10

3 rows returned in 0.01 seconds Download

Similarly same for left join.

--- to right outer join with condition

select e.empno,e.ename,e.job,d.deptno from emp e right outer join dept d on
e.deptno=d.deptno where e.job='SALESMAN';

Results | Explain | Describe | Saved SQL | History

EMPNO	ENAME	JOB	DEPTNO
7499	ALLEN	SALESMAN	30
7521	WARD	SALESMAN	30
7654	MARTIN	SALESMAN	30
7844	TURNER	SALESMAN	30

4 rows returned in 0.01 seconds Download

As same for left outer join

--- to full outer join with condition

select e.empno,e.ename,e.job,d.deptno from emp e full outer join dept d on e.deptno=d.deptno where e.deptno=20;

Results | Explain | Describe | Saved SQL | History

Saved SQL

EMPNO	ENAME	JOB	DEPTNO
7566	JONES	MANAGER	20
7788	SCOTT	ANALYST	20
7902	FORD	ANALYST	20
7369	SMITH	CLERK	20
7876	ADAMS	CLERK	20

EXISTS and NOT EXISTS Operator

--- exists and not exists in Oracle SQL

select * from emp where exists(select deptno from dept);

Results | Explain | Describe | Saved SQL | History

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	-	11/17/1981	5000	-	10
7698	BLAKE	MANAGER	7839	05/01/1981	2850	-	30
7782	CLARK	MANAGER	7839	06/09/1981	2450	-	10
7566	JONES	MANAGER	7839	04/02/1981	2975	-	20
7788	SCOTT	ANALYST	7566	12/09/1982	3000	-	20

select * from emp where not exists(select * from emp where job='PROGRAMMER');

Results | Explain | Describe | Saved SQL | History

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	-	11/17/1981	5000	-	10
7698	BLAKE	MANAGER	7839	05/01/1981	2850	-	30
7782	CLARK	MANAGER	7839	06/09/1981	2450	-	10
7566	JONES	MANAGER	7839	04/02/1981	2975	-	20
7788	SCOTT	ANALYST	7566	12/09/1982	3000	-	20

--- natural join
select * from emp natural join dept;

Results | Explain | Describe | Saved SQL | History

DEPTNO	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DNAME	LOC
10	7782	CLARK	MANAGER	7839	06/09/1981	2450	-	ACCOUNTING	NEW YORK
10	7934	MILLER	CLERK	7782	01/23/1982	1300	-	ACCOUNTING	NEW YORK
10	7839	KING	PRESIDENT	-	11/17/1981	5000	-	ACCOUNTING	NEW YORK
20	7902	FORD	ANALYST	7566	12/03/1981	3000	-	RESEARCH	DALLAS
20	7788	SCOTT	ANALYST	7566	12/09/1982	3000	-	RESEARCH	DALLAS

--- cross join are combination of right and left join
select * from emp cross join dept;

Results | Explain | Describe | Saved SQL | History

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DEPTNO	DNAME	LOC
7839	KING	PRESIDENT	-	11/17/1981	5000	-	10	10	ACCOUNTING	NEW YORK
7698	BLAKE	MANAGER	7839	05/01/1981	2850	-	30	10	ACCOUNTING	NEW YORK
7782	CLARK	MANAGER	7839	06/09/1981	2450	-	10	10	ACCOUNTING	NEW YORK
7566	JONES	MANAGER	7839	04/02/1981	2975	-	20	10	ACCOUNTING	NEW YORK
7788	SCOTT	ANALYST	7566	12/09/1982	3000	-	20	10	ACCOUNTING	NEW YORK

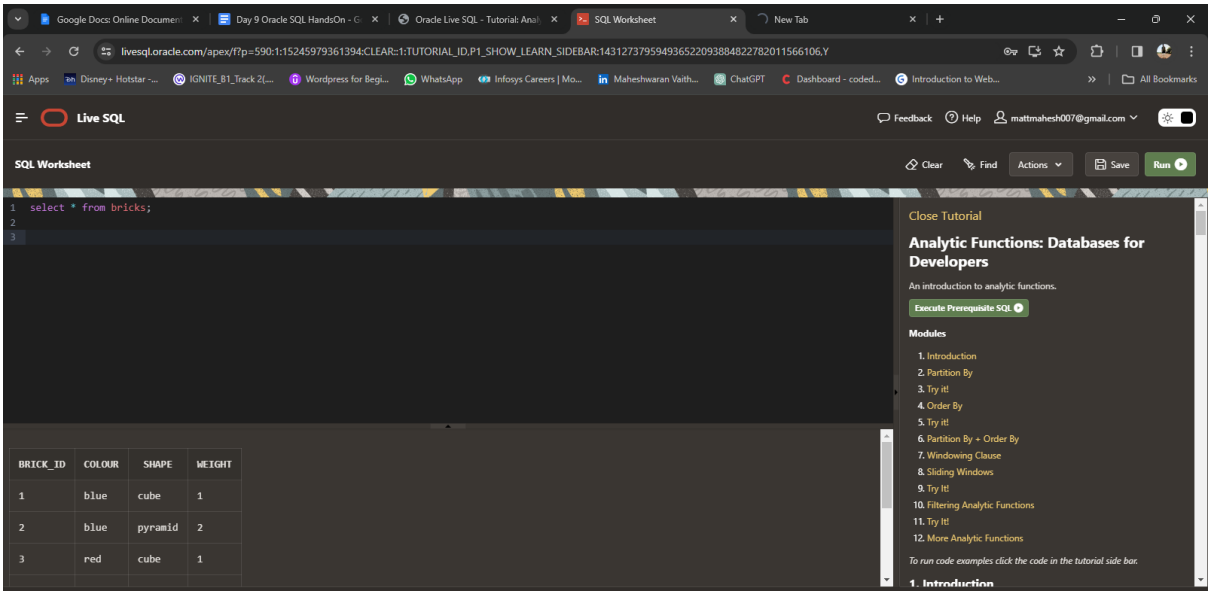
--- case statement

```
select ename,job,(
case job
when 'President' then 'HighProfile'
when 'Manager' then 'Employee'
when 'Analyst' then 'Good Perception'
when 'Clerk' then 'Assisstants'
else 'No comments Simply waste' end) as "COMMENT" from emp;
```

Results | Explain | Describe | Saved SQL | History

ENAME	JOB	COMMENT
KING	PRESIDENT	No comments Simply waste
BLAKE	MANAGER	No comments Simply waste
CLARK	MANAGER	No comments Simply waste
JONES	MANAGER	No comments Simply waste
SCOTT	ANALYST	No comments Simply waste

Partition in Oracle SQL



All the prerequisites are done

```
select b.*,(select count(*) from bricks) total_bricks from bricks b;
```

BRICK_ID	COLOUR	SHAPE	WEIGHT	TOTAL_BRICKS
1	blue	cube	1	6
2	blue	pyramid	2	6
3	red	cube	1	6

select b.*,(select sum(weight) from bricks where shape=b.shape) total_weight_of_bricks
from bricks b;

BRICK_ID	COLOUR	SHAPE	WEIGHT	TOTAL_WEIGHT_OF_BRICKS
1	blue	cube	1	4
3	red	cube	1	4
4	red	cube	2	4
2	blue	pyramid	2	6
5	red	pyramid	3	6

select b.*,count(*) over(partition by colour) total_count_bricks from bricks b;

BRICK_ID	COLOUR	SHAPE	WEIGHT	TOTAL_COUNT_BRICKS
2	blue	pyramid	2	2
1	blue	cube	1	2
6	green	pyramid	1	1
5	red	pyramid	3	3
4	red	cube	2	3

select b.*,sum(weight) over(partition by shape) sum_by_shape,
sum(weight) over(partition by colour) sum_by_weights,
max(weight) over(partition by shape) max_by_shape,
max(weight) over(partition by colour) max_by_colour
from bricks b;

BRICK_ID	COLOUR	SHAPE	WEIGHT	SUM_BY_SHAPE	SUM_BY_WEIGHTS	MAX_BY_SHAPE	MAX_BY_COLOUR
4	red	cube	2	4	6	2	3
1	blue	cube	1	4	3	2	2
3	red	cube	1	4	6	2	3
5	red	pyramid	3	6	6	3	3
2	blue	pyramid	2	6	3	3	2
6	green	pyramid	1	6	1	3	1

```
select b.*,
       count(*) over (
         partition by colour
         order by brick_id
       ) running_total,
       sum ( weight ) over (
         partition by colour
         order by brick_id
       ) running_weight
from   bricks b;
```

BRICK_ID	COLOUR	SHAPE	WEIGHT	RUNNING_TOTAL	RUNNING_WEIGHT
1	blue	cube	1	1	1
2	blue	pyramid	2	2	3
6	green	pyramid	1	1	1
3	red	cube	1	1	1
4	red	cube	2	2	3
5	red	pyramid	3	3	6

```
select b.*,
       count(*) over (
         order by weight, brick_id
         rows between unbounded preceding and current row
       ) running_total,
```

```
      sum ( weight ) over (
        order by weight, brick_id
        rows between unbounded preceding and current row
      ) running_weight
from   bricks b
order  by weight, brick_id;
```

BRICK_ID	COLOUR	SHAPE	WEIGHT	RUNNING_TOTAL	RUNNING_WEIGHT
1	blue	cube	1	1	1
3	red	cube	1	2	2
6	green	pyramid	1	3	3
2	blue	pyramid	2	4	5
4	red	cube	2	5	7
5	red	pyramid	3	6	10

Sliding window:

```
select b.*,
      count (*) over (
        order by weight
        range between 2 preceding and 1 preceding
      ) count_weight_2_lower_than_current,
      count (*) over (
        order by weight
        range between 1 following and 2 following
      ) count_weight_2_greater_than_current
from   bricks b
order  by weight;
```

BRICK_ID	COLOUR	SHAPE	WEIGHT	COUNT_WEIGHT_2_LOWER_THAN_CURRENT	COUNT_WEIGHT_2_GREATER_THAN_CURRENT
1	blue	cube	1	0	3
3	red	cube	1	0	3
6	green	pyramid	1	0	3
4	red	cube	2	3	1
2	blue	pyramid	2	3	1
5	red	pyramid	3	5	0

— rank , row_number and dense rank

```
select brick_id, weight,
       row_number() over ( order by weight ) rn,
       rank() over ( order by weight ) rk,
       dense_rank() over ( order by weight ) dr
from   bricks;
```

BRICK_ID	WEIGHT	RN	RK	DR
1	1	1	1	1
3	1	2	1	1
6	1	3	1	1
4	2	4	4	2
2	2	5	4	2
5	3	6	6	3

Lead and lag functions

```
select b.*,
       lag ( shape ) over ( order by brick_id ) prev_shape,
       lead ( shape ) over ( order by brick_id ) next_shape
from   bricks b;
```

BRICK_ID	COLOUR	SHAPE	WEIGHT	PREV_SHAPE	NEXT_SHAPE
1	blue	cube	1	-	pyramid
2	blue	pyramid	2	cube	cube
3	red	cube	1	pyramid	cube
4	red	cube	2	cube	pyramid
5	red	pyramid	3	cube	pyramid
6	green	pyramid	1	pyramid	-

```
select b.*,
       first_value ( weight ) over (
```

```
      order by brick_id
    ) first_weight_by_id,
    last_value ( weight ) over (
      order by brick_id
      range between current row and unbounded following
    ) last_weight_by_id
from   bricks b;
```

BRICK_ID	COLOUR	SHAPE	WEIGHT	FIRST_WEIGHT_BY_ID	LAST_WEIGHT_BY_ID
1	blue	cube	1	1	1
2	blue	pyramid	2	1	1
3	red	cube	1	1	1
4	red	cube	2	1	1
5	red	pyramid	3	1	1
6	green	pyramid	1	1	1

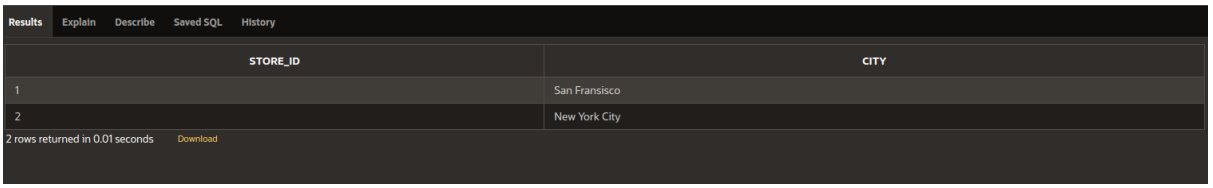
--- exploring table creation

```
create table stores (store_id number not null,city varchar2(20));
```

---inserting values into the table

```
insert into stores values(1,'San Fransisco');
insert into stores values(2,'New York City');
```

```
select * from stores;
```



--- create table with primary key

```
create table testpk(t_id number not null,t_name varchar(20),constraint test_pk primary
key(t_id));
```

```
insert into testpk values(107,'Nijan');
```

```
select * from testpk;
```

Results | Explain | Describe | Saved SQL | History

T_ID	T_NAME
102	Pradeep
107	Nijan
101	Mahesh
104	Jai

alter table testpk modify t_name varchar(20) not null;

Basic crud operations in sql are done earlier.

Copying all the data to the table

--- creating an table with existing table

create table emp_info as select * from emp;

select * from emp_info;

Results | Explain | Describe | Saved SQL | History

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	-	11/17/1981	5000	-	10
7698	BLAKE	MANAGER	7839	05/01/1981	2850	-	30
7782	CLARK	MANAGER	7839	06/09/1981	2450	-	10
7566	JONES	MANAGER	7839	04/02/1981	2975	-	20
7788	SCOTT	ANALYST	7566	12/09/1982	3000	-	20
7902	FORD	ANALYST	7566	12/03/1981	3000	-	20
7369	SMITH	CLERK	7902	12/17/1980	800	-	20

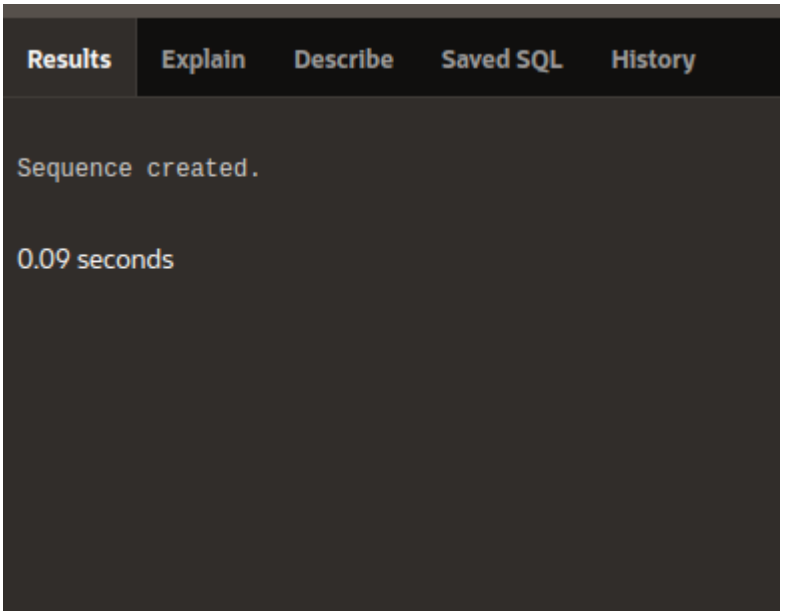
— merge into
merge into emp_info ei using emp e on(ei.empno=e.empno) when matched then
update set ei.ename=e.ename;

Results | Explain | Describe | Saved SQL | History

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	-	11/17/1981	5000	-	10
7698	BLAKE	MANAGER	7839	05/01/1981	2850	-	30
7782	CLARK	MANAGER	7839	06/09/1981	2450	-	10
7566	JONES	MANAGER	7839	04/02/1981	2975	-	20
7788	SCOTT	ANALYST	7566	12/09/1982	3000	-	20
7902	FORD	ANALYST	7566	12/03/1981	3000	-	20

--- sequence creation

create sequence prod_seq
minvalue 1
maxvalue 9999999999
start with 100
increment by 1
nocache;



Sequence handson done in postgresql
Delete , Truncate and Drop commands are already done earlier.

--- index in oracle sql
create unique index idx_emp on emp_info(ename);

create index idx_job on emp_info(job);

--- to drop that index

drop index idx_job;

Views and union intersection and except are already done in postgreSQL.