**SQL COMMANDS**

**SCETION 6**

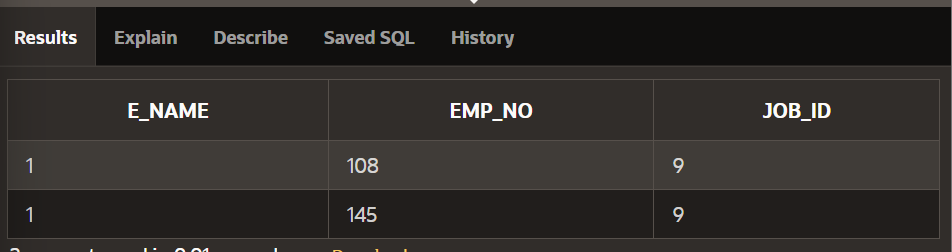
**HIREARICHAL LEVEL**

select level e\_name,emp\_no,job\_id

from employee

start with job\_id=9

connect by prior emp\_no =job\_id



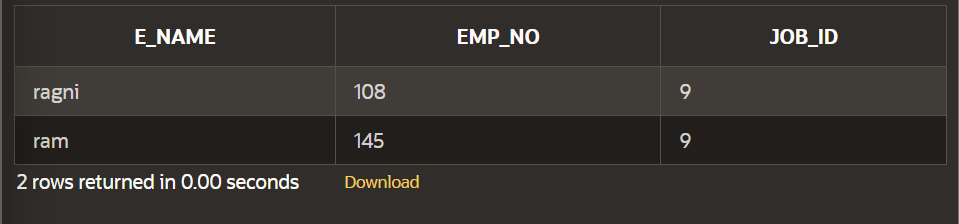
**HIERARICHAL START WITH**

select e\_name,emp\_no,job\_id

from employee

start with job\_id=9

connect by prior emp\_no =job\_id



**FULL OUTER JOIN**

select e.e\_name,j.job\_id,j.job\_name

from employee e full outer join job j

on (e.job\_id=j.job\_id);



**RIGHT OUTER JOIN**

select e.e\_name,j.job\_id,j.job\_name

from employee e right outer join job j

on (e.job\_id=j.job\_id);

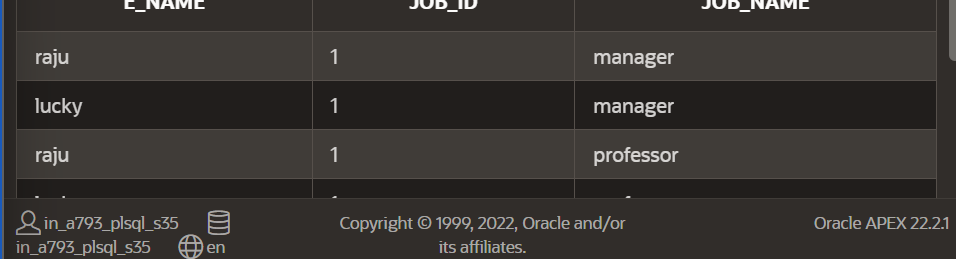


**LEFT OUTER JOIN**

select e.e\_name,j.job\_id,j.job\_name

from employee e left outer join job j

on (e.job\_id=j.job\_id);

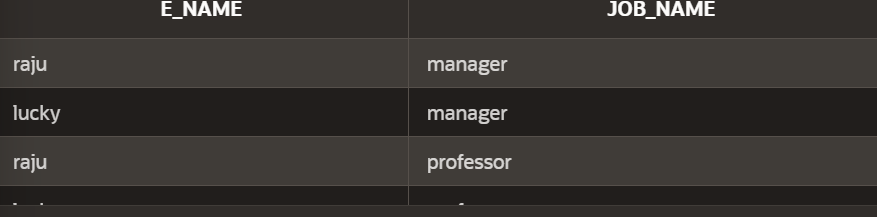


**ON CLAUSE**

select e\_name,job\_name

from employee e join job j

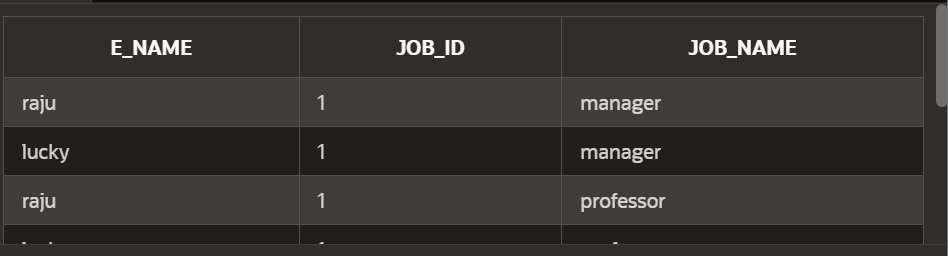
on (e.job\_id=j.job\_id);



**USING CLAUSE**

select e\_name,job\_id,job\_name

from employee join job using (job\_id);

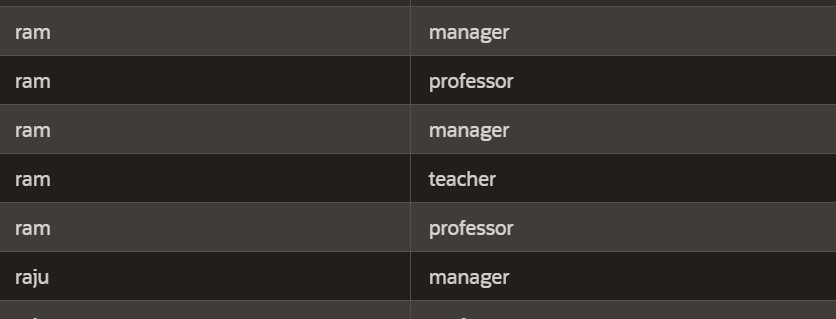


CROSS JOIN

select e\_name,job\_name

from employee cross join

job;

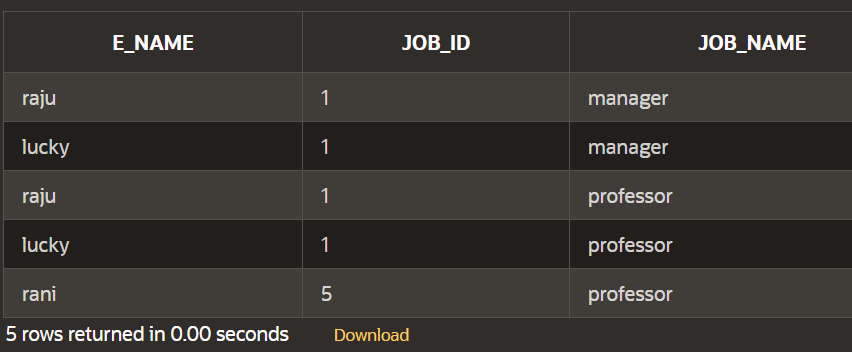


NATURAL JOIN:

select e\_name,job\_id,job\_name

from employee natural join job

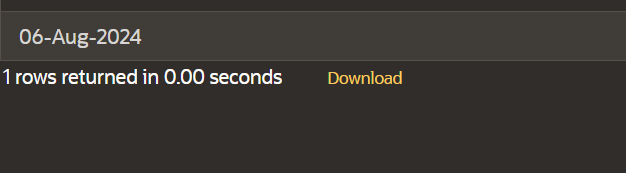
where job\_id<9;



**SYS DATE:**

select sysdate

from dual;

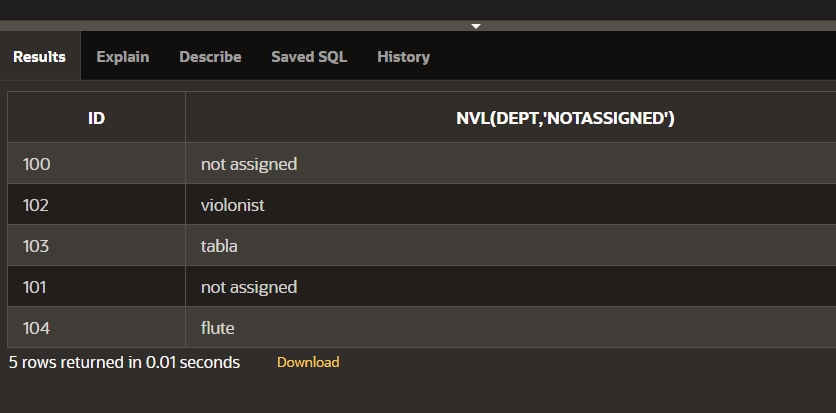


**SECTION 5**

**NVL FUNCTION:**

select id,nvl(dept,'not assigned')

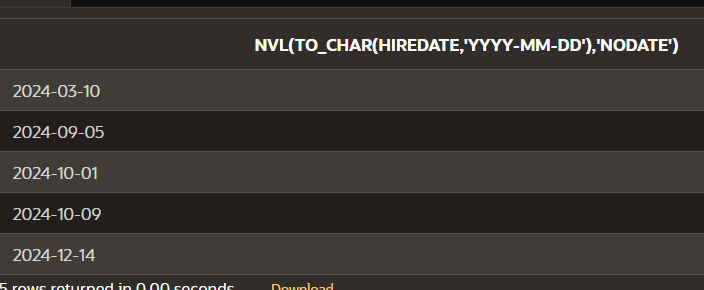
from singer;



**NVL DATE:**

SELECT NVL(TO\_CHAR(hiredate, 'YYYY-MM-DD'), 'no date')

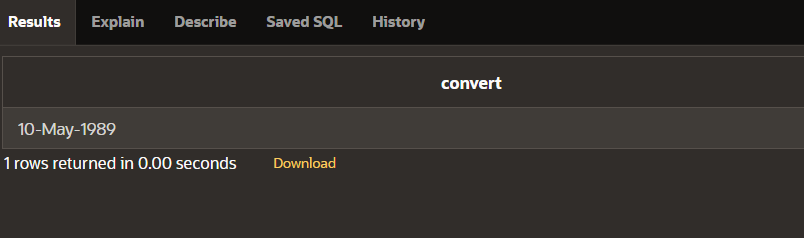
FROM singer;



**CHARACTER TO DATE**

select to\_date('may10,1989','fxmondd,yyyy') as "convert"

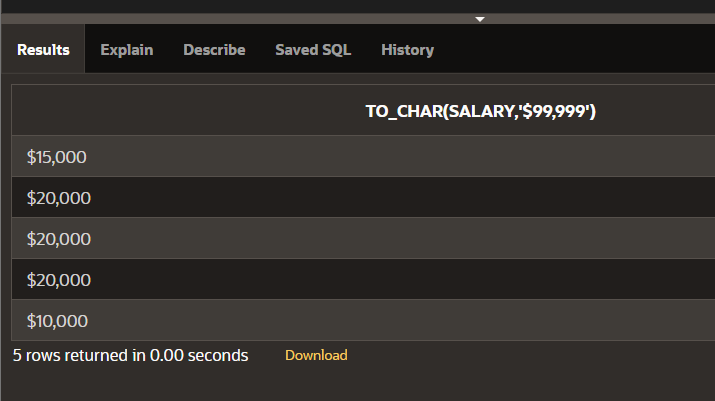
from dual;



**NUMBER TO CHARACTER:**

select to\_char(salary,'$99,999')

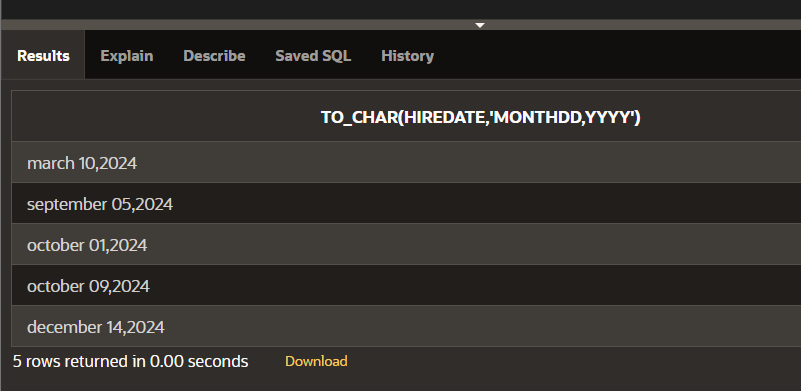
from singer;



**DATE TO CHARACTER:**

select to\_char(hiredate,'month dd,yyyy')

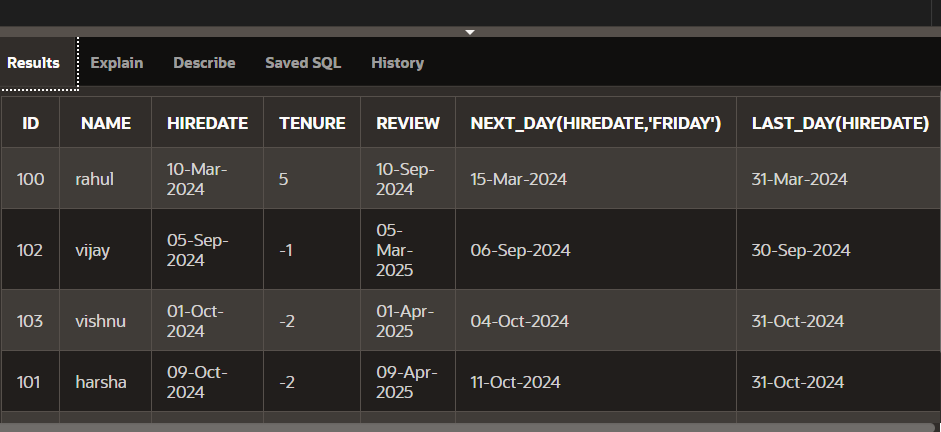
from singer;

****

**DATE FUNCTION**

select id,name,hiredate,round(months\_between(sysdate,hiredate)) as tenure, add\_months(hiredate,6) as review, next\_day(hiredate,'friday'),last\_day(hiredate)

from singer;



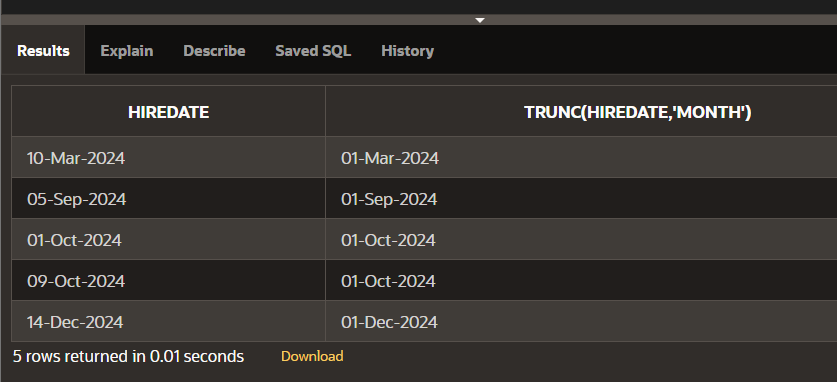
**SECTION 4**

**TRUNC**

select hiredate,

trunc(hiredate,'month')

from singer;

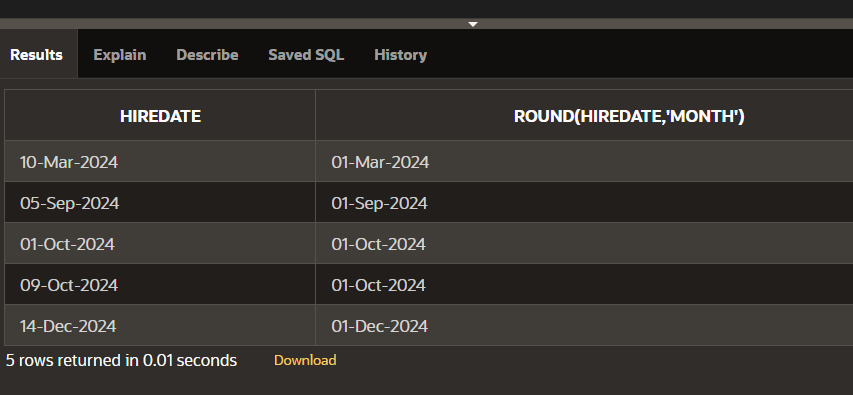


**ROUND TO HIREDATE**

select hiredate,

round(hiredate,'month')

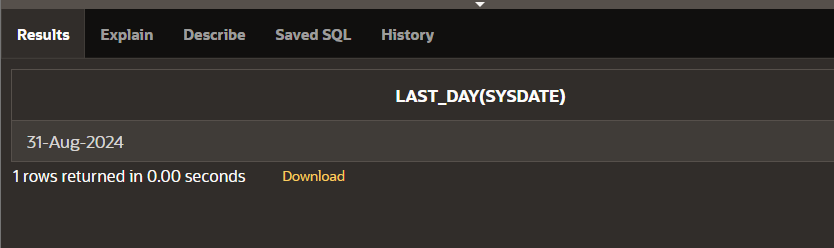
from singer;



**LAST DAY**

select last\_day(sysdate)

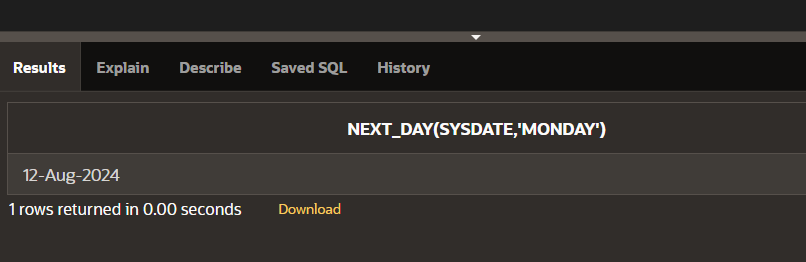
from dual;



**NEXT DAY**

select next\_day(sysdate,'monday')

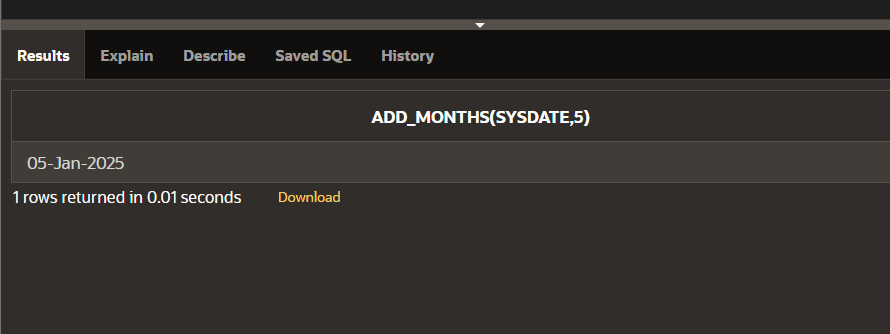
from dual;



**ADD MONTHS**

select add\_months(sysdate,5)

from dual;



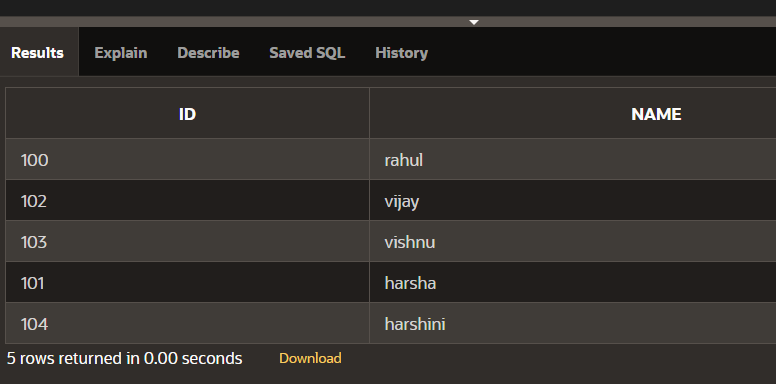
**MONTHS BETWEEN**

select id,name

from singer

where months\_between

(sysdate,hiredate)<100;



**SECTION\_7**

**EQUIJOIN AND CARTESIAN PRODUCT**

CREATE TABLE employ(

eno VARCHAR(14),

ename VARCHAR(14),

eadhress VARCHAR(15),

epno VARCHAR(15),

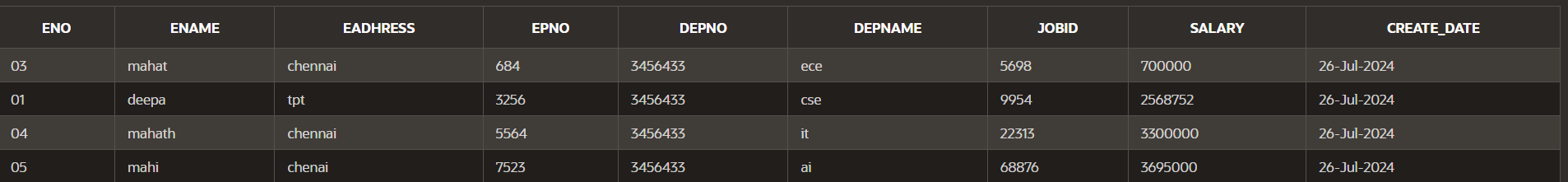
depno VARCHAR(14),

depname VARCHAR(14),

jobid VARCHAR(10),

salary VARCHAR(10),

create\_date DATE DEFAULT SYSDATE);



CREATE TABLE jobs (

job\_id VARCHAR(10) PRIMARY KEY,

job\_title VARCHAR(50) NOT NULL,

min\_salary DECIMAL(8, 2),

max\_salary DECIMAL(8, 2)

);

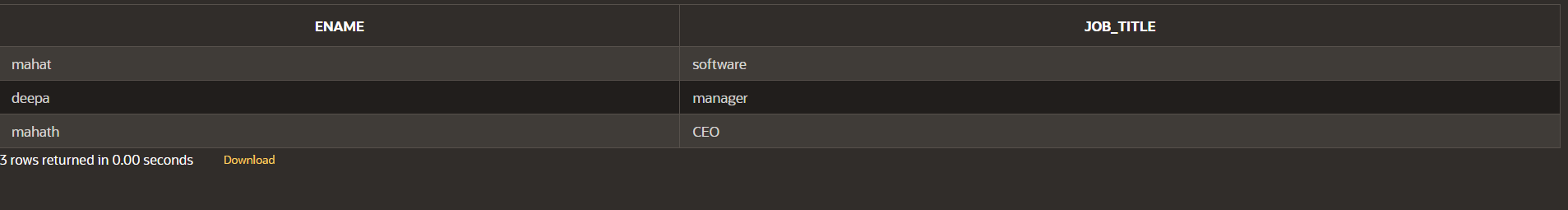


**PROPRIEATARY JOINS:**

SELECT employ.ename, jobs.job\_title

FROM employ,jobs

WHERE employ.jobid=jobs.job\_id;

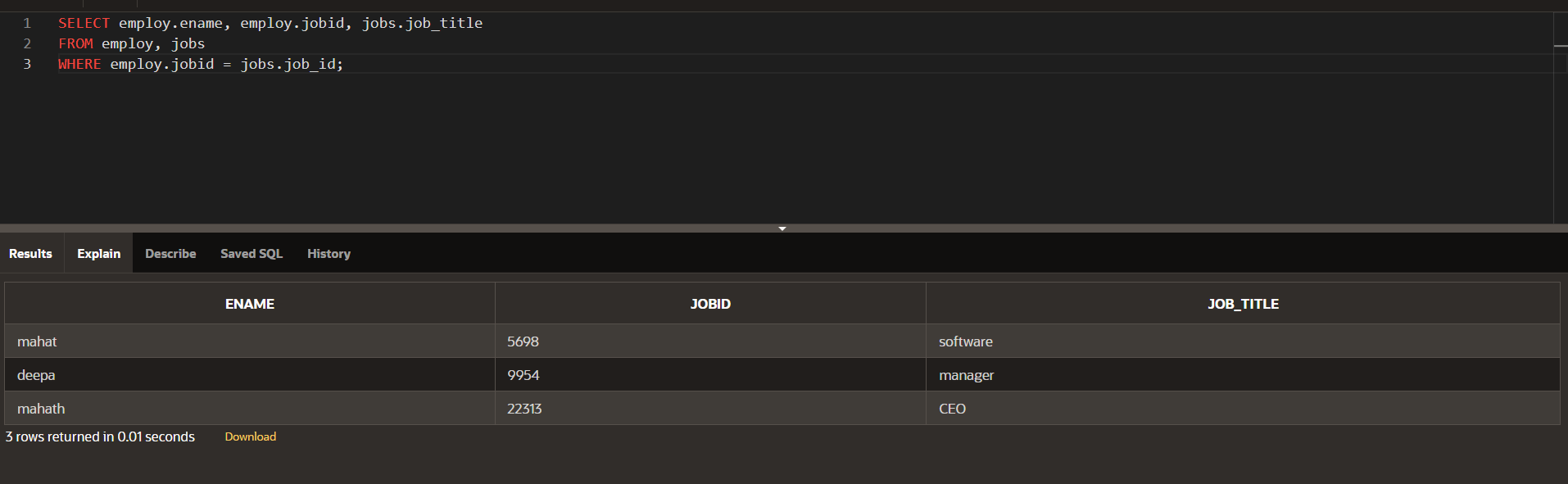


**EQUIJOIN:**

SELECT employ.ename, employ.jobid, jobs.job\_title

FROM employ, jobs

WHERE employ.jobid = jobs.job\_id;



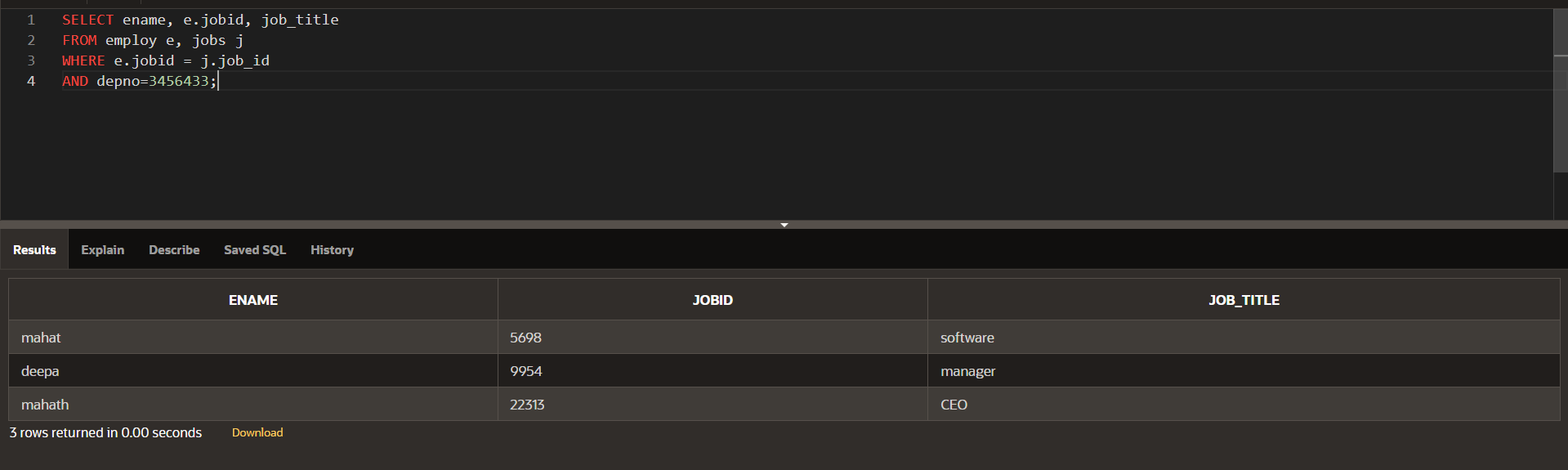
**ALIASES:**

SELECT ename, e.jobid, job\_title

FROM employ e, jobs j

WHERE e.jobid = j.job\_id

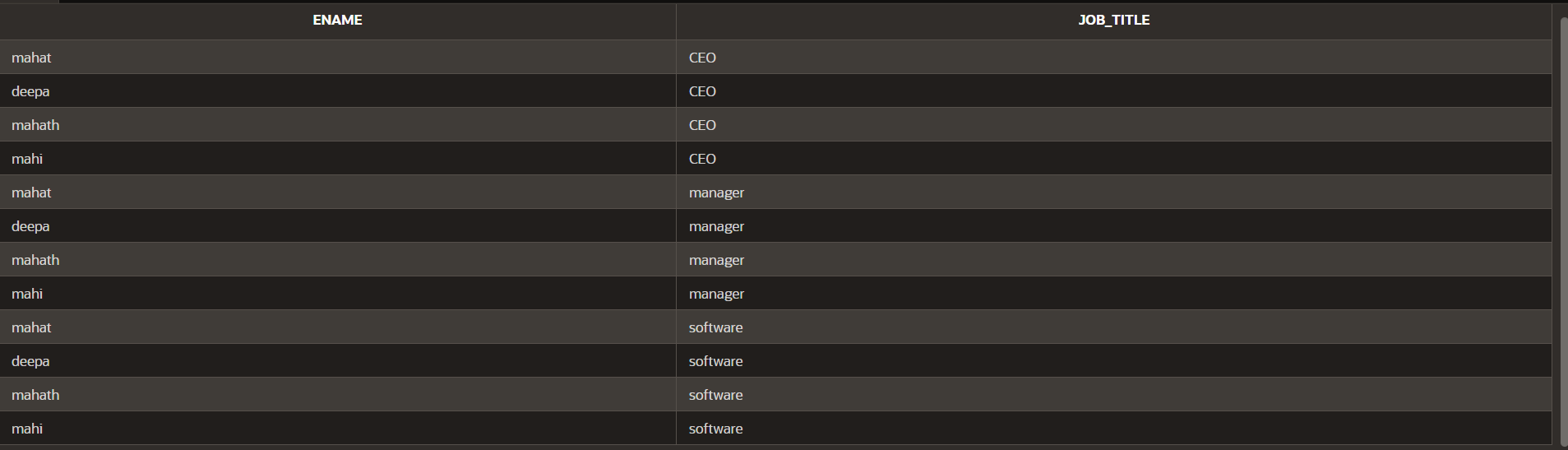
AND depno=3456433;



**CARTESIAN PRODUCTJOIN:**

SELECT employ.ename,jobs.job\_title

FROM employ,jobs;



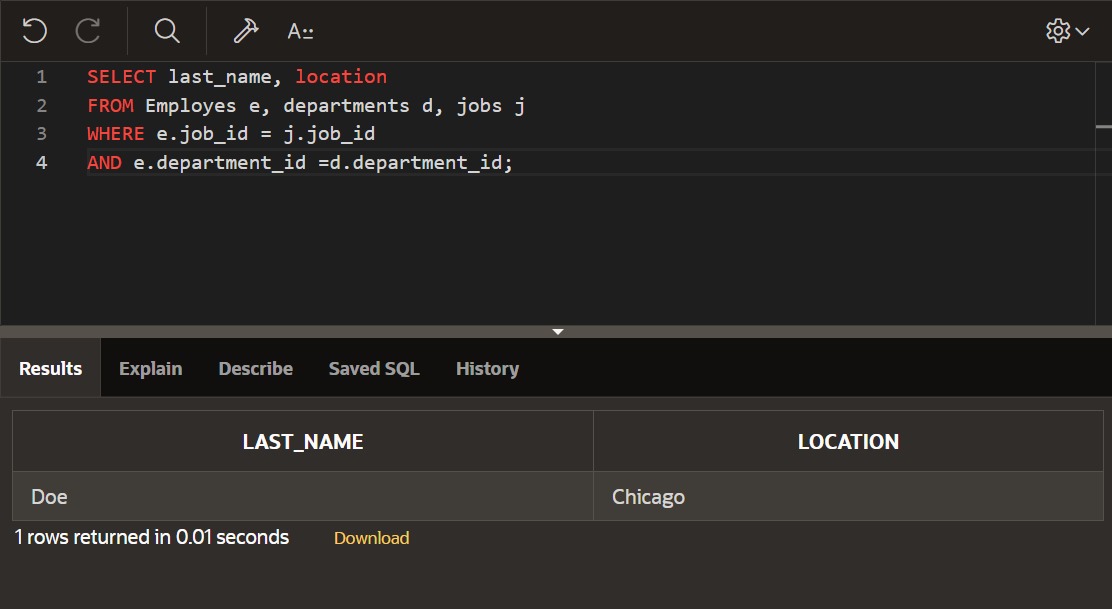
**JOIN:**

SELECT last\_name, location

FROM Employes e, departments d, jobs j

WHERE e.job\_id = j.job\_id

AND e.department\_id =d.department\_id;



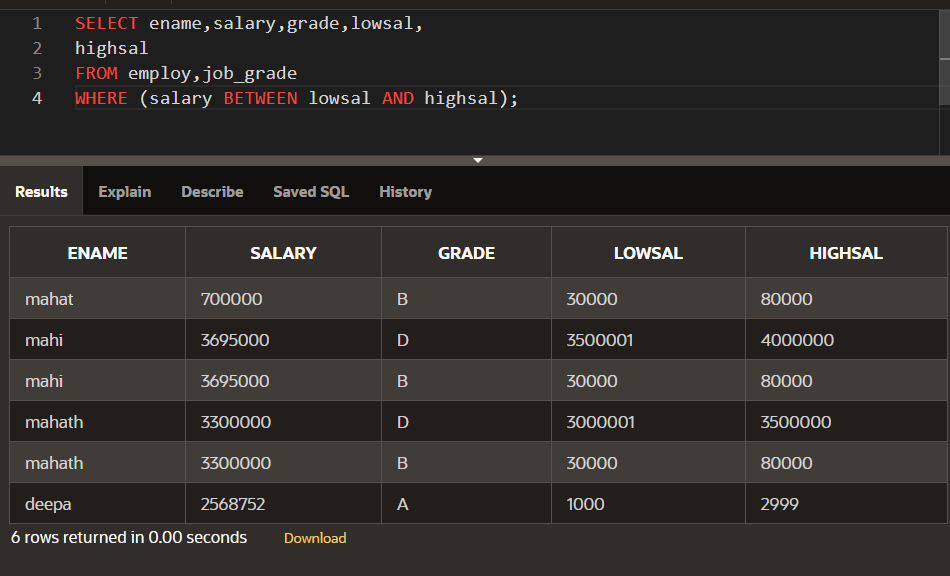
**NONEQUI JOIN:**

SELECT ename,salary,grade,lowsal,

highsal

FROM employ,job\_grade

WHERE (salary BETWEEN lowsal AND highsal);



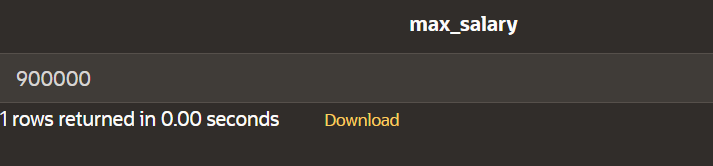
**SECTION\_8**

**MAXIMUM:**

select max(salary)

as "max\_salary"

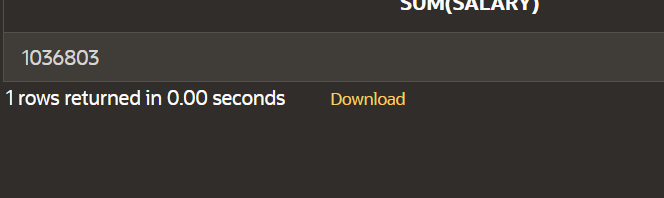
FROM employee;



**SUM**:

select sum(salary)

from employee;

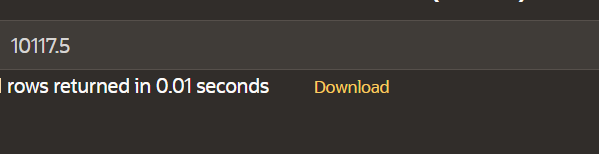


**AVERAGE:**

select avg(salary)

from employee

where job\_id=1;

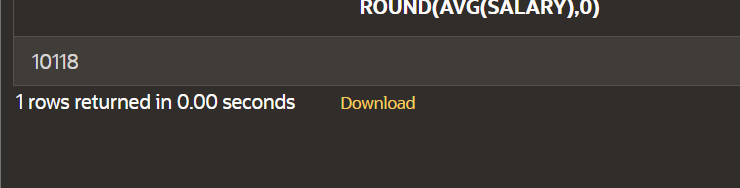


**ROUND AVERAGE:**

select round(avg(salary),0)

from employee

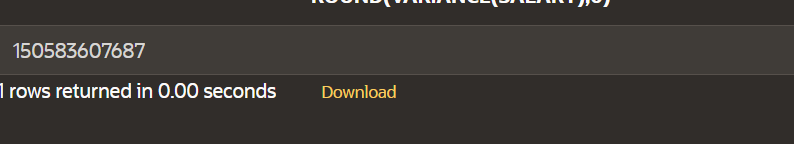
where job\_id=1;



**ROUND VARIANCE:**

select round(variance(salary),0)

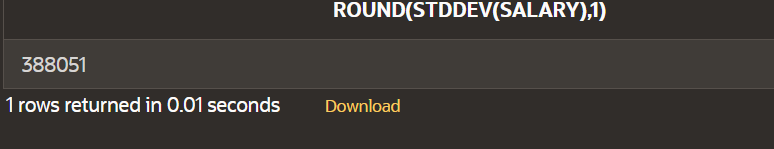
from employee



ROUND STDDEV:

select round(stddev(salary),1)

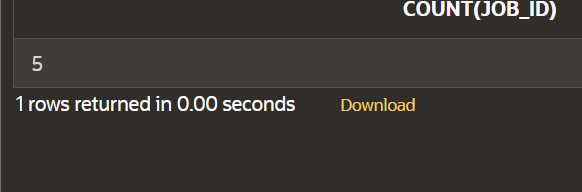
from employee



**COUNT**:

SELECT COUNT(job\_id)

FROM employee;

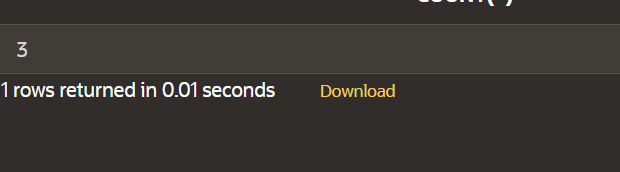


**COUNT ALL ROWS:**

SELECT COUNT(\*)

FROM employee

WHERE job\_id < 9;

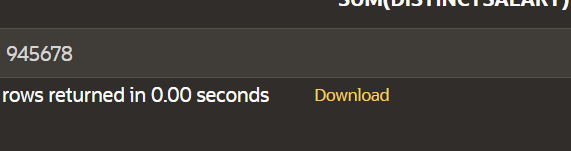


**SUM DISTINCT SALARY:**

SELECT SUM(distinct salary)

FROM employee

WHERE job\_id = 9;

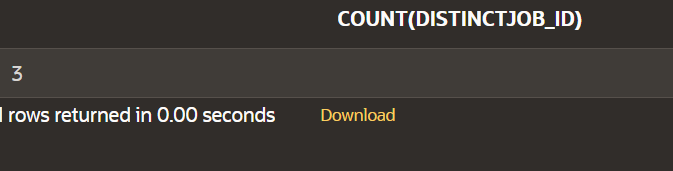


**COUNT(DISTINCT JOB\_ID):**

SELECT COUNT (DISTINCT

job\_id)

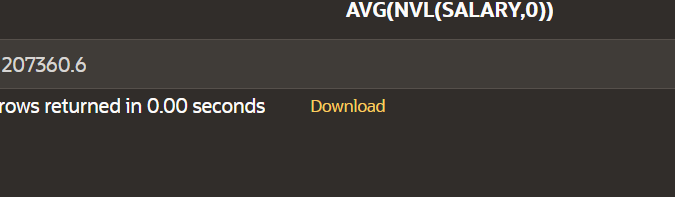
FROM employee**;**

****

**NULL:**

select avg(nvl(salary,0))

from employee;

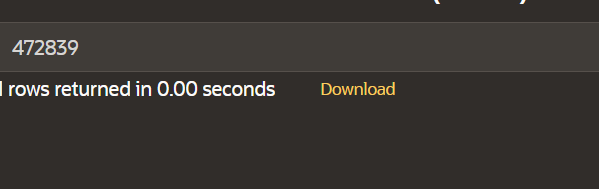


**GROUP AVERAGE:**

select avg(salary)

from employee

where job\_id=9;



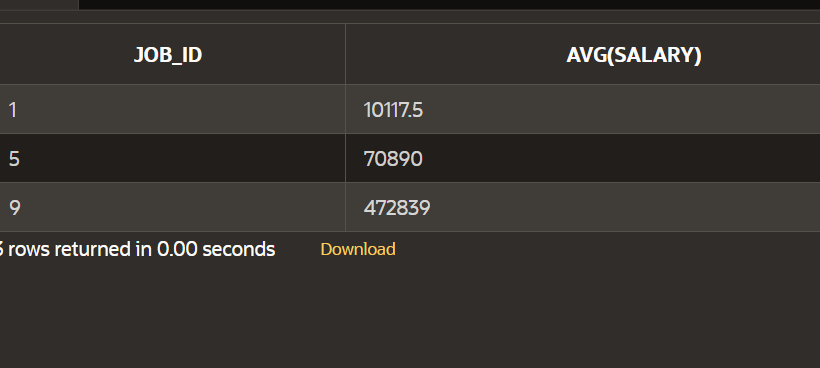
**GROUP BY USE:**

SELECT job\_id, AVG(salary)

FROM employee

GROUP BY job\_id

ORDER BY job\_id;

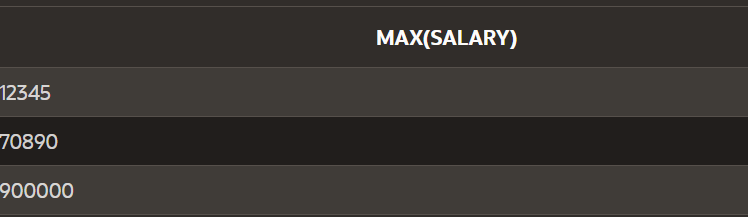


**GROUP BY MAX:**

SELECT MAX(salary)

FROM employee

GROUP BY job\_id;

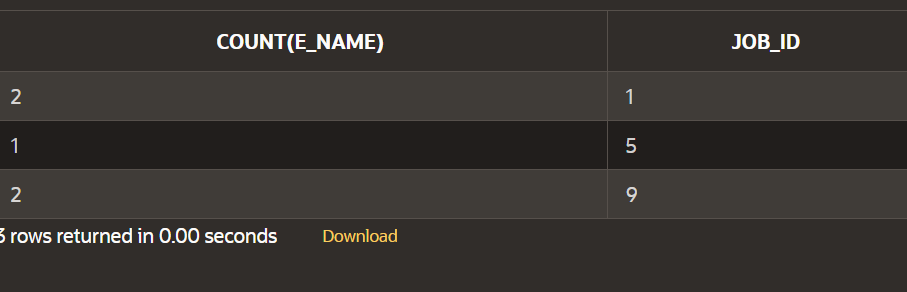


**GROUP BY COUNT:**

select count (e\_name),job\_id

from employee

group by job\_id;

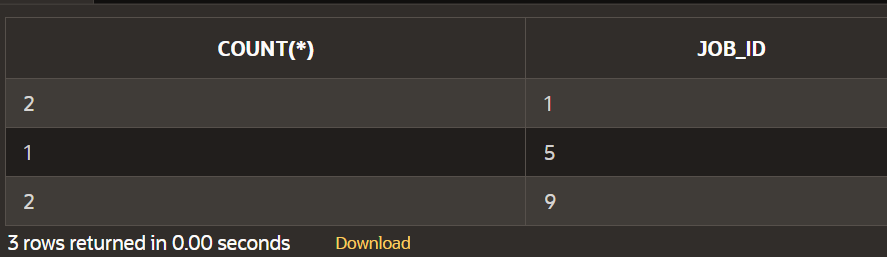
****

**COUNT BY COUNT (\*):**

select count (\*),job\_id

from employee

group by job\_id;



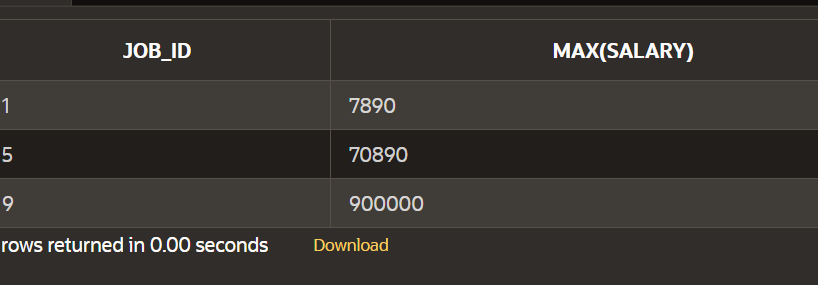
**GROUP BY WHERE:**

SELECT job\_id, MAX(salary)

FROM employee

WHERE e\_name != 'lucky'

GROUP BY job\_id;



**GROUP WITHIN GROUPS:**

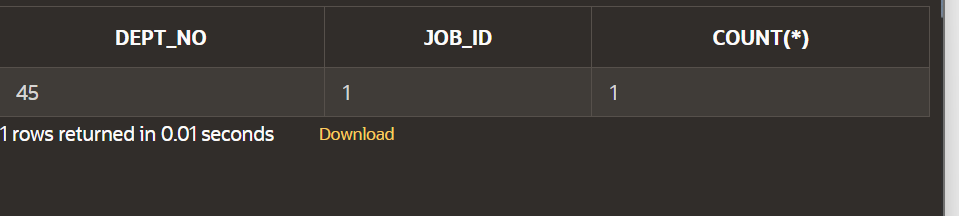
SELECT dept\_no, job\_id,

count(\*)

FROM employee

WHERE dept\_no > 40

GROUP BY dept\_no, job\_id;

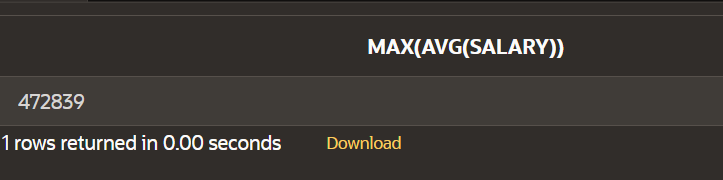


**NESTING GROUP FUNCTIONS:**

SELECT max(avg(salary))

FROM employee

GROUP by job\_id;



**SECTION 9**

**HAVING:**

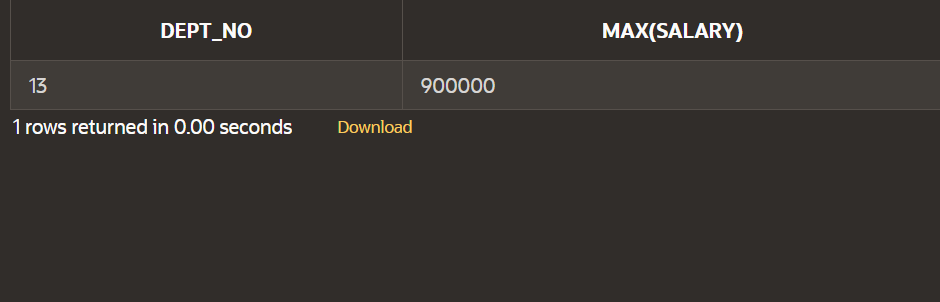
SELECT dept\_no,MAX(salary)

FROM employee

GROUP BY dept\_no

HAVING COUNT(\*)>1

ORDER BY dept\_no

****

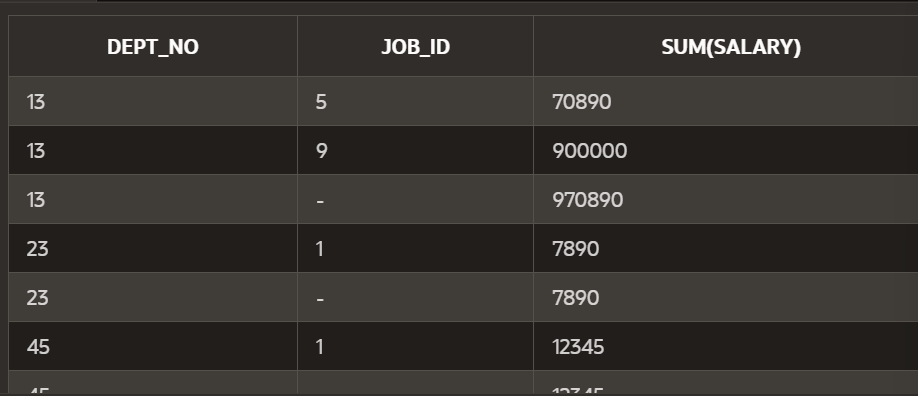
**ROLL UP:**

SELECT dept\_no, job\_id, SUM(salary)

FROM employee

WHERE dept\_no < 50

GROUP BY ROLLUP (dept\_no, job\_id);



**WITHOUT ROLLUP:**

SELECT dept\_no, job\_id, SUM(salary)

FROM employee

WHERE dept\_no < 50

GROUP BY (dept\_no, job\_id);



**CUBE:**

SELECT dept\_no, job\_id, SUM(salary)

FROM employee

WHERE dept\_no < 50

GROUP BY cube (dept\_no, job\_id);



**GROUPING SETS:**

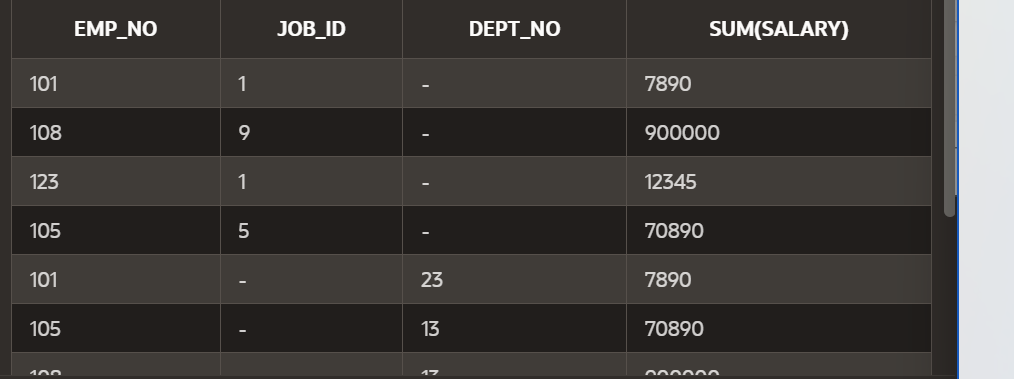
select emp\_no,job\_id,dept\_no,sum(salary)

from employee

where dept\_no<50

group by grouping sets

((emp\_no,job\_id),(job\_id,dept\_no),(dept\_no,emp\_no));



UNION:

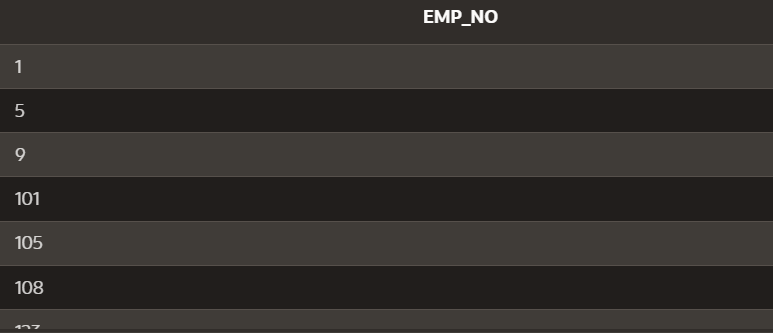
SELECT emp\_no

FROM employee

UNION

SELECT job\_id

FROM job;



**UNION ALL:**

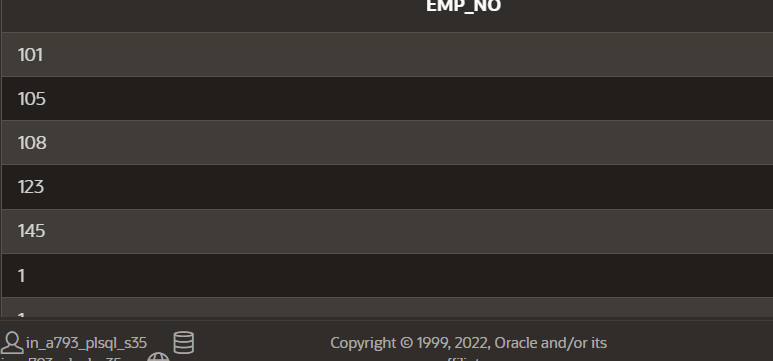
SELECT emp\_no

FROM employee

UNION all

SELECT job\_id

FROM job;



**SECTION-10**

**INTERSECT:**

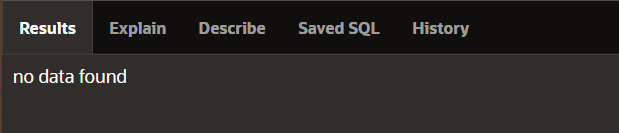
SELECT emp\_no

FROM employee

intersect

SELECT job\_id

FROM job;



**MINUS:**

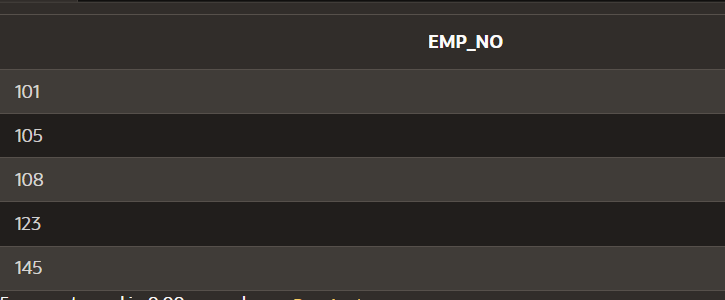
SELECT emp\_no

FROM employee

minus

SELECT job\_id

FROM job;



**SUBQUERY EXAMPLE:**

SELECT e\_name,

emp\_no

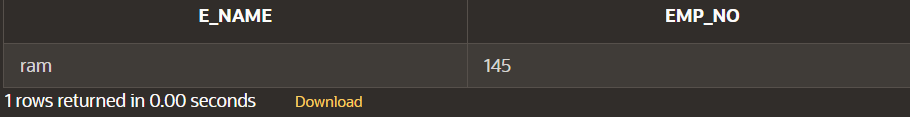
FROM employee

WHERE emp\_no >

(SELECT emp\_no

FROM employee

WHERE e\_name = 'lucky');



**SUBQUERY AND NULL:**

SELECT e\_name,

emp\_no

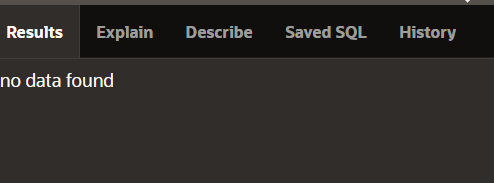
FROM employee

WHERE emp\_no >

(SELECT emp\_no

FROM employee

WHERE e\_name = 'anji');



**SUBQUERY FROM DIFFERENT TABLES:**

SELECT e\_name, job\_id, dept\_no

FROM employee

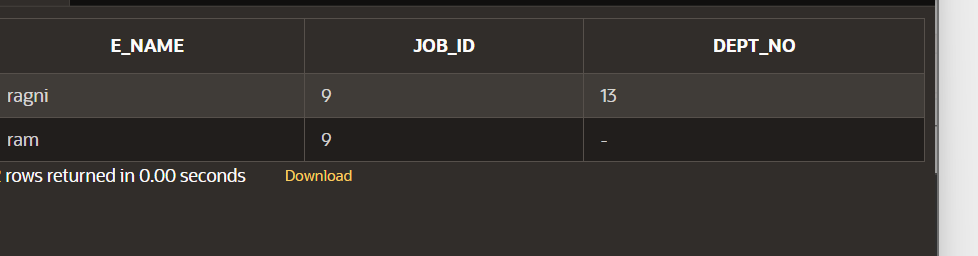
WHERE job\_id =

(SELECT job\_id

FROM job

WHERE job\_name = 'teacher')

ORDER BY job\_id;



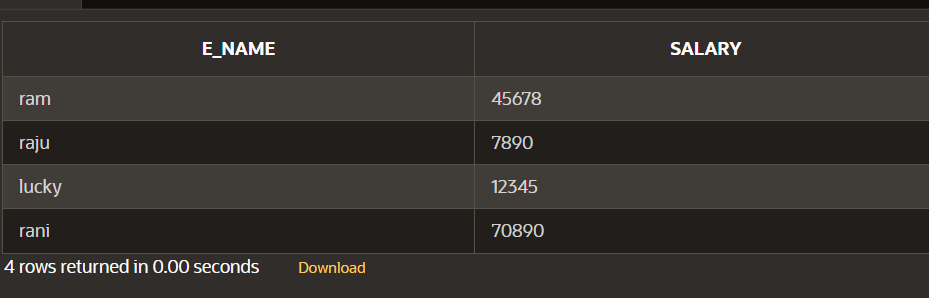
**GROUP FUNCTIONS IN SUBQUERIES:**

SELECT e\_name, salary

FROM employee

WHERE salary <

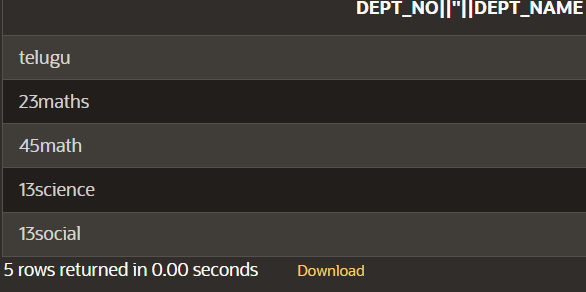
(SELECT AVG(salary)

FROM employee);

**CONCATENATION:**

select dept\_no||''||dept\_name

from employee;

****

**SECTION\_12**

**DEFAULT EXAMPLE:**

CREATE TABLE my\_employees (

hire\_date DATE DEFAULT SYSDATE,

first\_name VARCHAR2(15),

last\_name VARCHAR2(15)

);

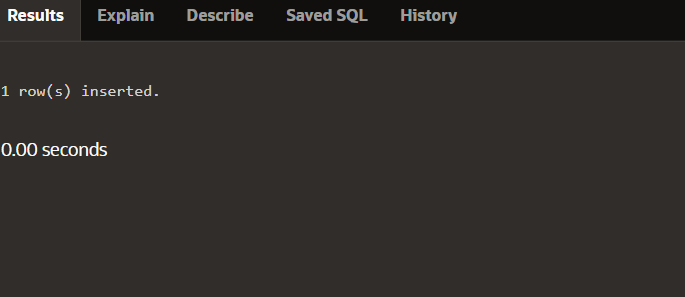
**EXPLICIT DEFAULT EXAMPLE:**

INSERT INTO my\_employees

(hire\_date, first\_name, last\_name)

VALUES

(DEFAULT, 'anjali','raju');



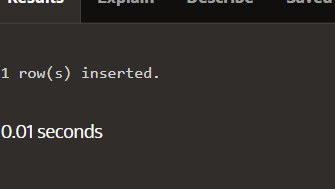
**IMPLICIT USE OF DEFAULT:**

INSERT INTO my\_employees

(first\_name, last\_name)

VALUES

('Angelina','Wright');

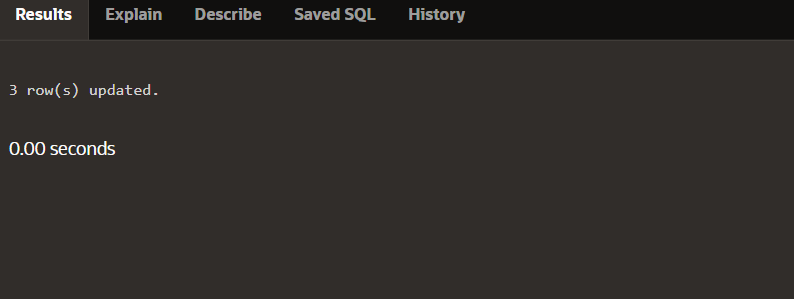


**EXPLICIT DEFAULT WITH UPDATE:**

UPDATE my\_employees

SET hire\_date = DEFAULT

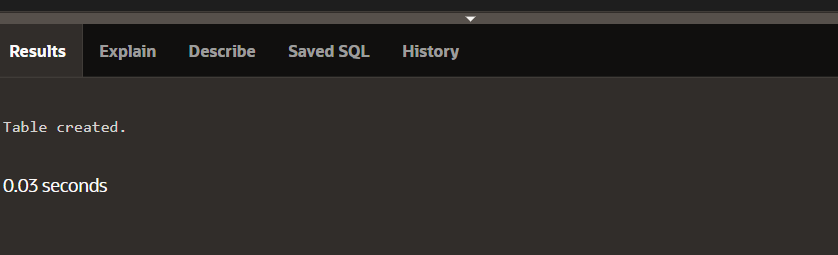
WHERE last\_name = 'Wright';



**CREATE A COPY OF THE TABLE:**

CREATE TABLE copy\_employee

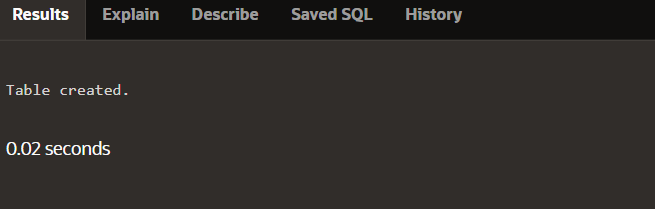
AS (SELECT \* FROM employee);



**CREATE A COPY OF JOB TABLE:**

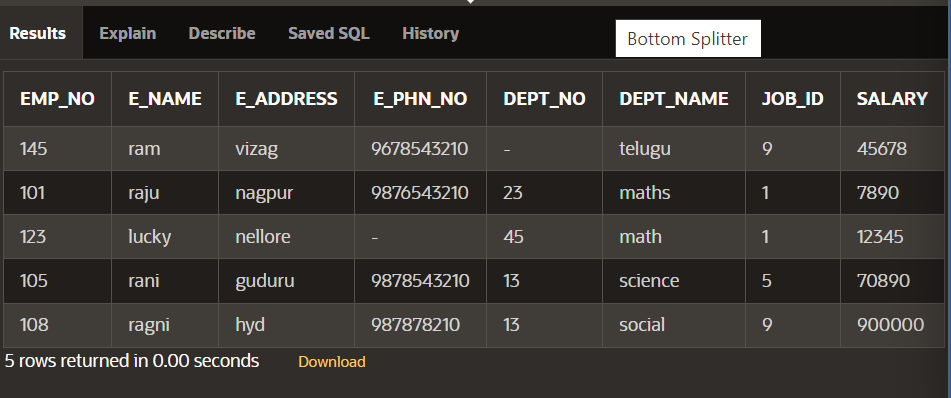
CREATE TABLE copy\_job

AS (SELECT \* FROM job);



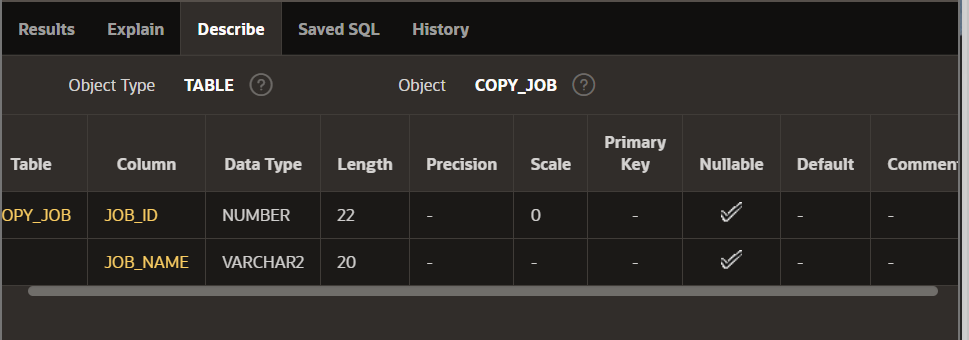
**DISPLAY COPY OF EMPLOYEE:**

SELECT \* FROM copy\_employee;



**DESCRIBE EMPLOYEE TABLE:**

DESCRIBE copy\_job;

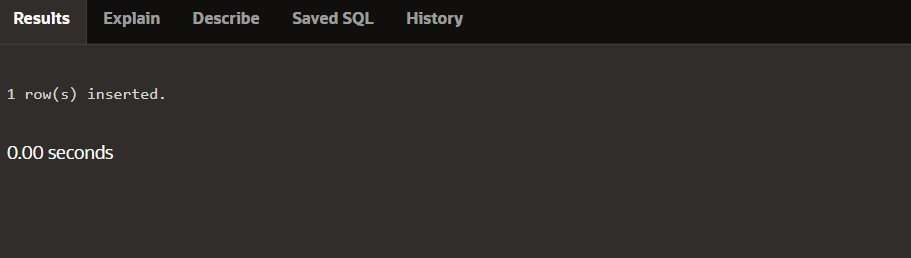


**INSERT INTO COPY OF JOB:**

INSERT INTO copy\_job

(job\_id, job\_name)

VALUES (200,'Human Resource');

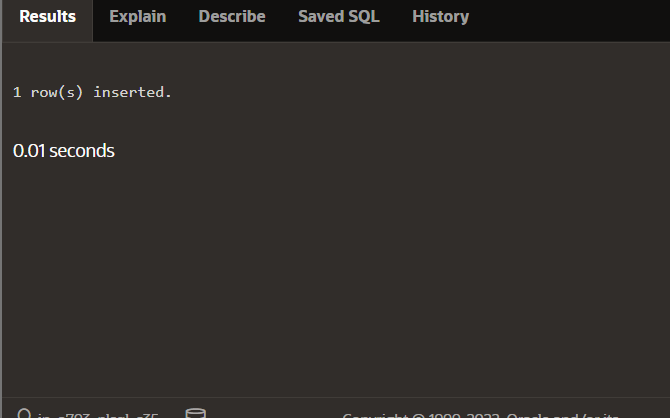


**Implicit insert into copy of job:**

INSERT INTO copy\_job

VALUES

(210,'Estate Management');

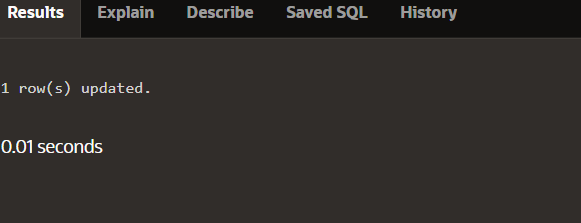


**Update**:

UPDATE copy\_employee

SET job\_id = '123456'

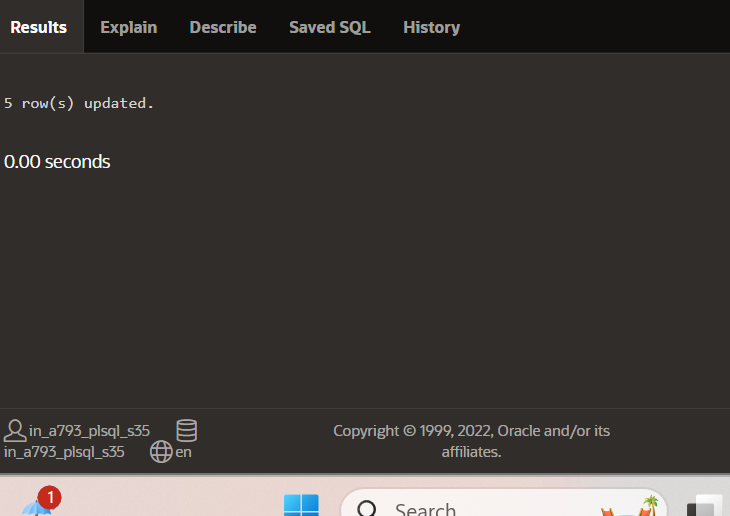
WHERE emp\_no = 145;



**Update copy of table:**

UPDATE copy\_employee

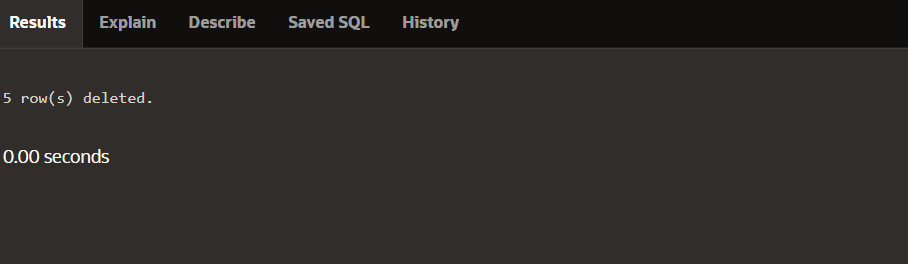
SET emp\_no = 243576, e\_name = 'raju'



**Delete:**

DELETE from copy\_employee

WHERE emp\_no = 243576;



Subquery delete:

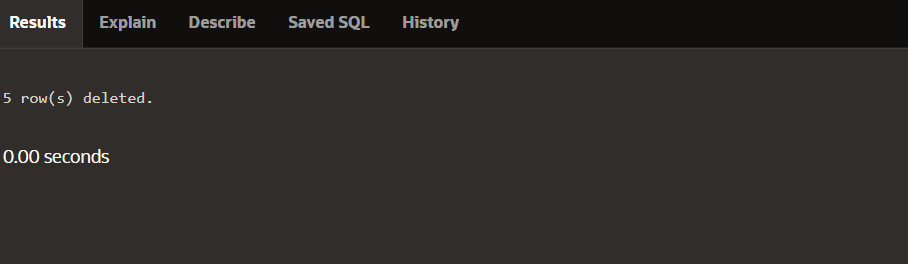
DELETE FROM copy\_employees

WHERE department\_id =

(SELECT department\_id

FROM departments

WHERE department\_name = 'Shipping');



**Section\_13**

**CREATE TABLE MY FRIENDS:**

CREATE TABLE my\_friends

(first\_name VARCHAR2(20),

last\_name VARCHAR2(30),

email VARCHAR2(30),

phone\_num VARCHAR2(12),

birth\_date DATE);

**CREATE TABLE MY CDCOLLECTION:**

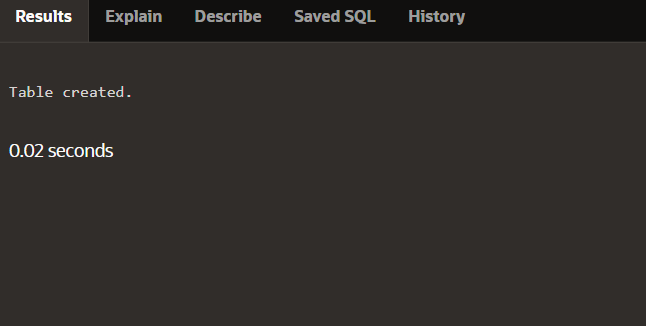
CREATE TABLE my\_cd\_collection

(cd\_number NUMBER(3),

title VARCHAR2(20),

artist VARCHAR2(20),

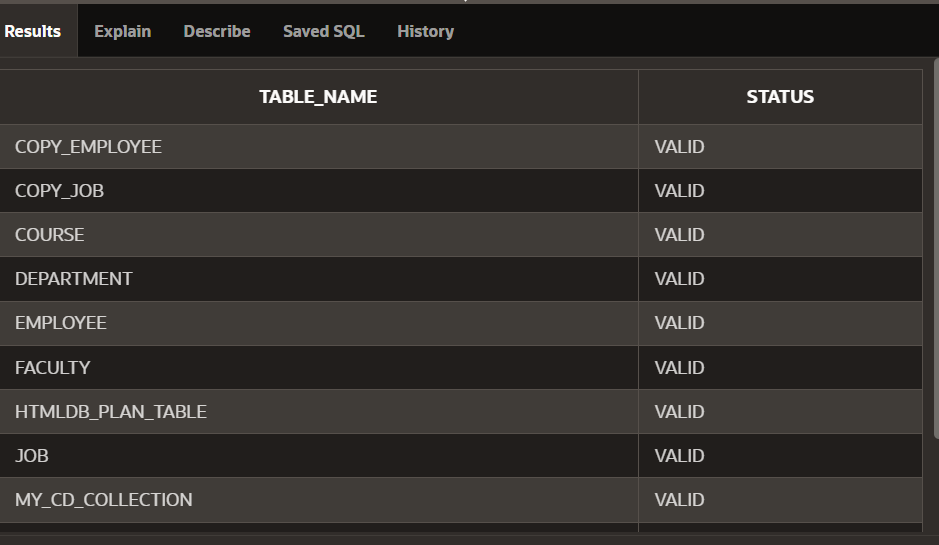
purchase\_date DATE DEFAULT SYSDATE);



**DATA DICTIONARY:**

SELECT table\_name, status

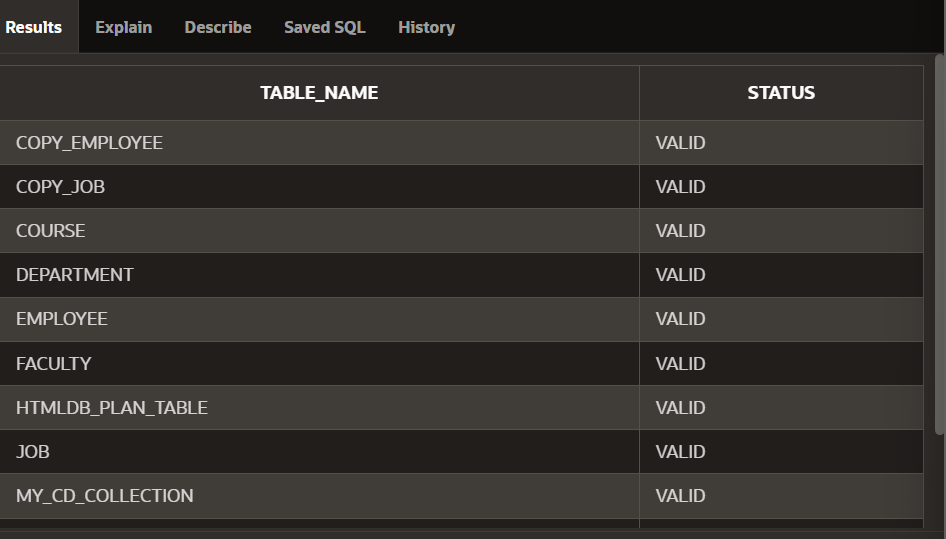
FROM USER\_TABLES;



**DATA DICTIONARY TABLE NAME:**

SELECT table\_name, status

FROM ALL\_TABLES;



**TIMES STAMP TABLE:**

CREATE TABLE time\_ex1

(exact\_time TIMESTAMP);

**TIMES STAMP TABLE INSERT ROW:**

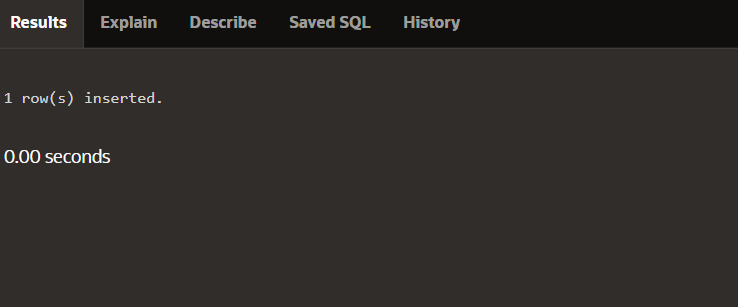
INSERT INTO time\_ex1

VALUES ('8-aug-2024 10:52:29.123456'); 

**TIMES STAMP TABLE INSERT ROW USING SYS UPDATE:**

INSERT INTO time\_ex1

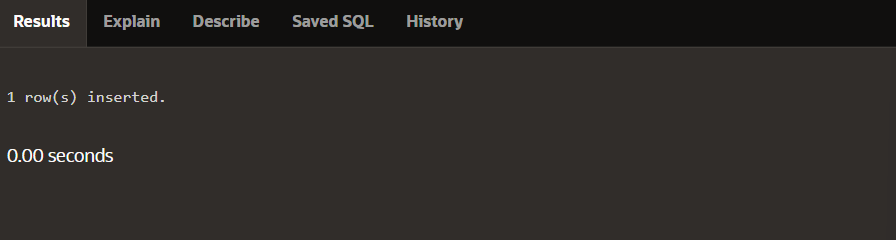
VALUES (SYSDATE);



**SYS UPDATE:**

INSERT INTO time\_ex1

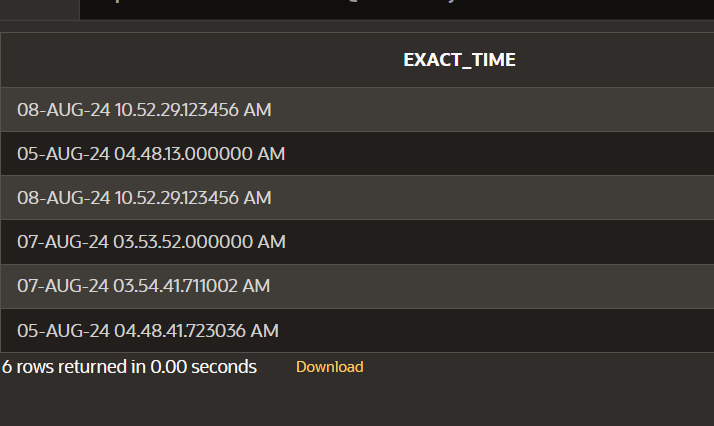
VALUES (SYSTIMESTAMP);



**DISPLAY TIME STAMP:**

SELECT \*

FROM time\_ex1;

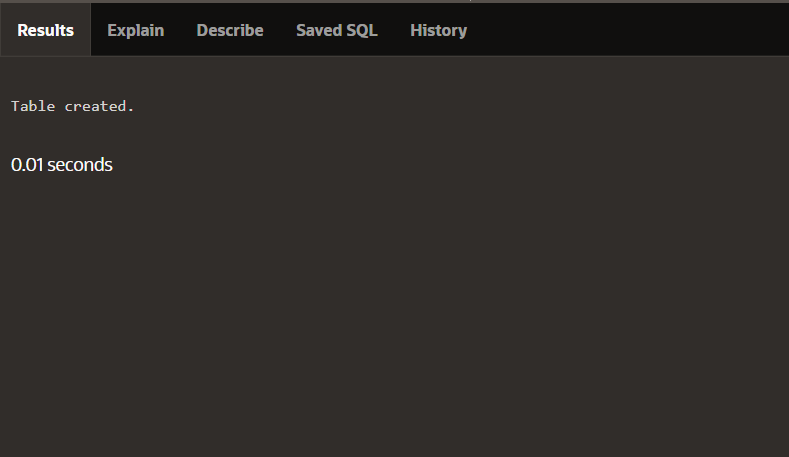


**CREATE TABLE:**

CREATE TABLE time\_ex4

(loan\_duration1 INTERVAL YEAR(3) TO MONTH,

loan\_duration2 INTERVAL YEAR(2) TO MONTH);

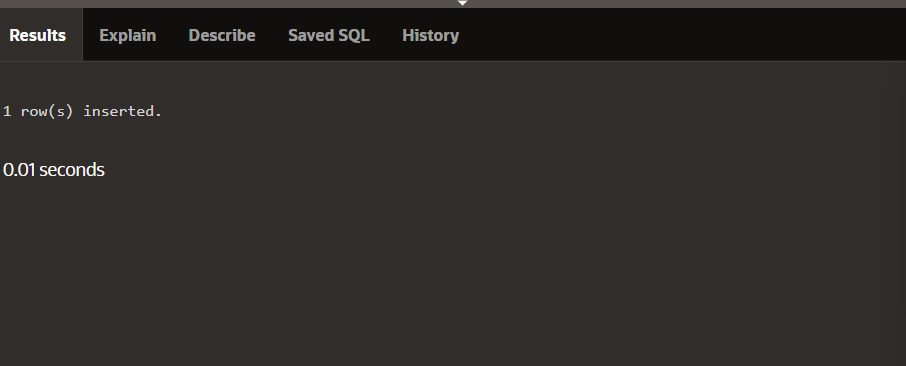


**INSERT :**

INSERT INTO time\_1(loan\_duration1, loan\_duration2)

VALUES (INTERVAL '120' MONTH(3),

INTERVAL '3-6' YEAR TO MONTH);



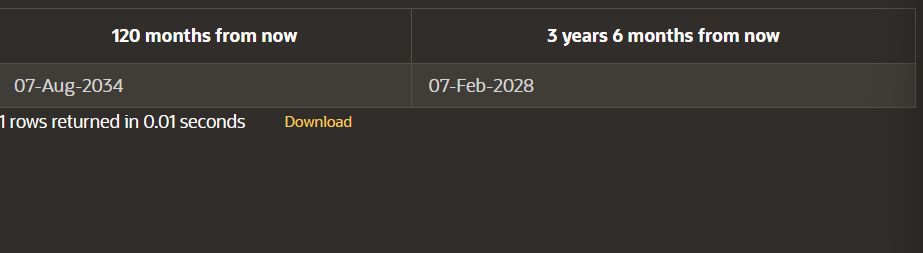
INTERVAL MONTH TO YEAR:

SELECT SYSDATE + loan\_duration1 AS "120 months from now",

SYSDATE + loan\_duration2 AS "3 years 6 months from

now"

FROM time\_1;

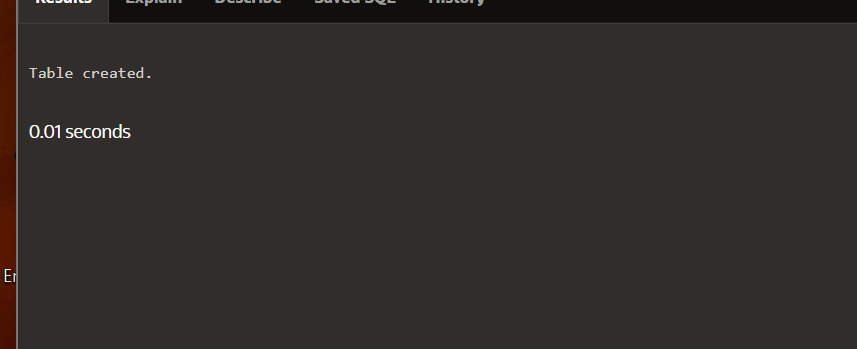


**DAY TO SECOND:**

CREATE TABLE time\_2

(day\_duration1 INTERVAL DAY(3) TO SECOND,

day\_duration2 INTERVAL DAY(3) TO SECOND);



**DAY TO SECOND INSERT ROW:**

INSERT INTO time\_2 (day\_duration1, day\_duration2)

VALUES (INTERVAL '25' DAY(2), INTERVAL '4 10:30:10' DAY TO

SECOND);



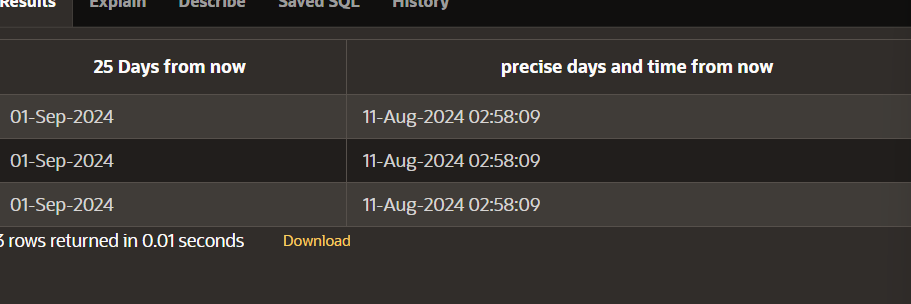
**DAY TO SECOND SELECT:**

SELECT SYSDATE + day\_duration1 AS "25 Days from now",

TO\_CHAR(SYSDATE + day\_duration2, 'dd-Mon-yyyy hh:mi:ss')

AS "precise days and time from now"

FROM time\_2;



**SECTION 14**

**CREATING TABLE:**

CREATE TABLE clients (

client\_number NUMBER(4),

first\_name VARCHAR2(14),

last\_name VARCHAR2(13));

CREATING CONSTRAINTS

CREATE TABLE clients

(client\_number NUMBER(4) CONSTRAINT clients\_client\_num\_pk

PRIMARY KEY,

first\_name VARCHAR2(14),

last\_name VARCHAR2(13));

COLUMN LEVEL

CREATE TABLE clients

(client\_number NUMBER(4) CONSTRAINT clients\_cient\_num\_pk PRIMARY KEY,

last\_name VARCHAR2(13) CONSTRAINT clients\_last\_name\_nn NOT NULL,

email VARCHAR2(80) CONSTRAINT clients\_emil\_uk UNIQUE);

**TABLE LEVEL:**

CREATE TABLE clients (

client\_number NUMBER(6) NOT NULL,

first\_name VARCHAR2(20),

last\_name VARCHAR2(20),

phone VARCHAR2(20),

email VARCHAR2(10) NOT NULL,

CONSTRAINT clients\_phone\_email\_uk UNIQUE (email,phone));

)

**VIOLATION:**

CREATE TABLE clients(

client\_number NUMBER(6),

first\_name VARCHAR2(20),

last\_name VARCHAR2(20),

phone VARCHAR2(20) CONSTRAINT phone\_email\_uk

UNIQUE(email,phone),

email VARCHAR2(10) CONSTRAINT NOT NULL,

CONSTRAINT emailclients\_email NOT NULL,

CONSTRAINT clients\_client\_num\_pk PRIMARY KEY (client\_number));

**UNIQUE CONSTRAINT:**

INSERT INTO clients (client\_number, first\_name, Last\_name, phone,

email)

VALUES (7234, 'Lonny', 'Vigil', 4072220091, 'lbv@lbv.net');

PRIMARY KEY

CREATE TABLE clients

(client\_number NUMBER(4) CONSTRAINT clients\_client\_num\_pk

PRIMARY KEY,

first\_name VARCHAR2(14),

last\_name VARCHAR2(13));

**TABLE LEVEL:**

CREATE TABLE copy\_job\_history

(employee\_id NUMBER(6,0),

start\_date DATE,

job\_id VARCHAR2(10),

department\_id NUMBER(4,0),

CONSTRAINT copy\_jhist\_id\_st\_date\_pk PRIMARY KEY(employee\_id,

start\_date));

FOREIGN KEY

CREATE TABLE copy\_employees

(employee\_id NUMBER(6,0) CONSTRAINT copy\_emp\_pk PRIMARY KEY,

first\_name VARCHAR2(20),

last\_name VARCHAR2(25),

department\_id NUMBER(4,0),

email VARCHAR2(25),

CONSTRAINT c\_emps\_dept\_id\_fk FOREIGN KEY (department\_id)

REFERENCES departments(department\_id));

ON DELETE CASCADE

CREATE TABLE copy\_employees

(employee\_id NUMBER(6,0) CONSTRAINT copy\_emp\_pk PRIMARY KEY,

first\_name VARCHAR2(20),

last\_name VARCHAR2(25),

department\_id NUMBER(4,0),

email VARCHAR2(25),

CONSTRAINT cdept\_dept\_id\_fk FOREIGN KEY (department\_id)

REFERENCES copy\_departments(department\_id));

**SECTION\_15**

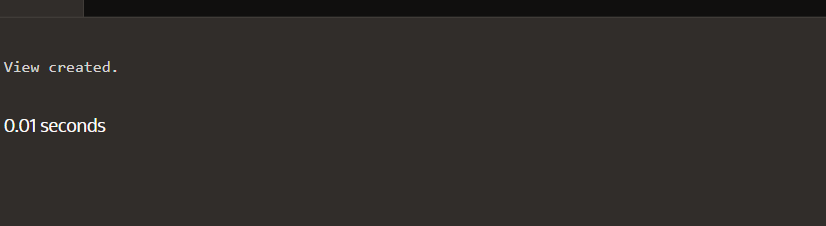
**VIEW:**

CREATE VIEW view\_employee

AS SELECT emp\_no,e\_name,job\_id

FROM employee

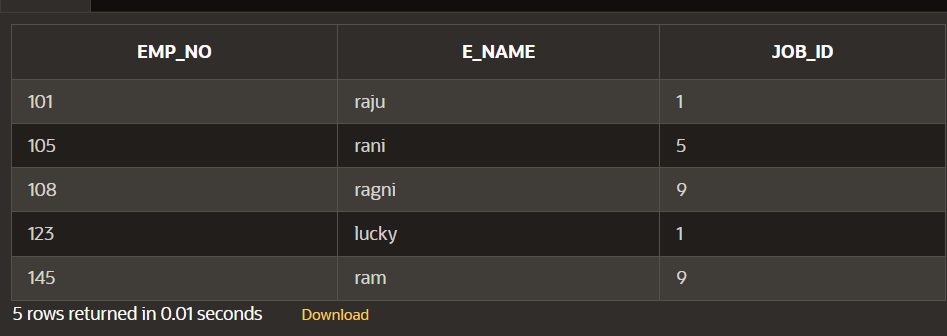
WHERE emp\_no BETWEEN 100 and 243569;



**VIEWING THE TABLE:**

SELECT \*

FROM view\_employee;



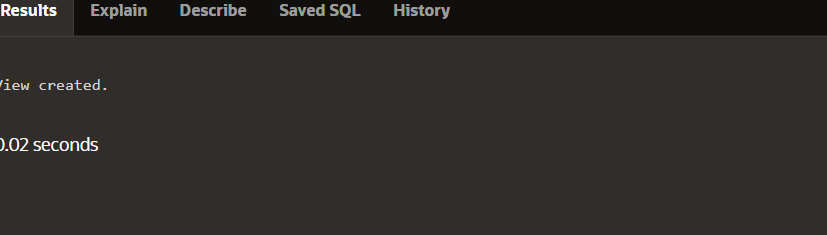
**CREATING A VIEW:**

CREATE OR REPLACE VIEW view\_employee

AS SELECT job\_id, emp\_no, e\_name

FROM employee

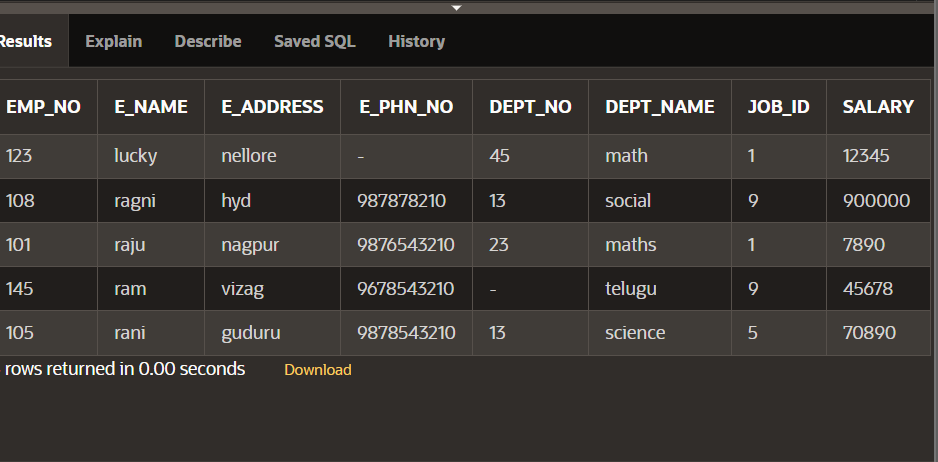
WHERE emp\_no LIKE '%43567';



**VIEW:**

SELECT \* FROM employee

ORDER BY e\_name;



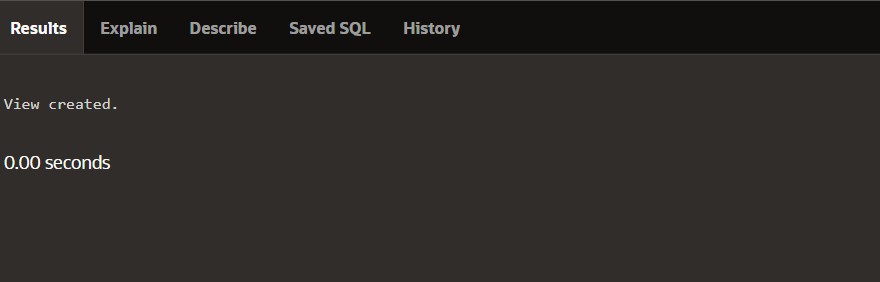
**SIMPLE VIEW:**

CREATE OR REPLACE VIEW view\_employee

AS SELECT job\_id, emp\_no, e\_name

FROM employee

WHERE emp\_no LIKE '%43567';



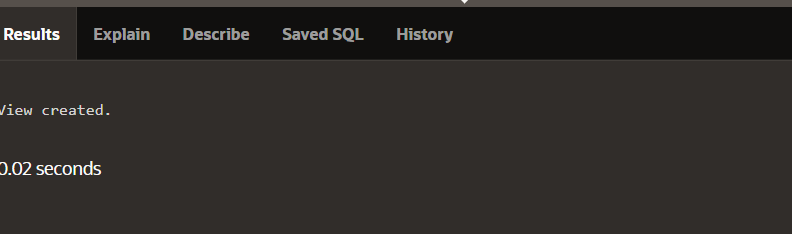
**Simple view with columns :**

CREATE OR REPLACE VIEW view\_employee

AS SELECT job\_id as id , emp\_no as e\_no, e\_name as name

FROM employee

WHERE emp\_no LIKE '%43567';



**MODIFYING A VIEW:**

CREATE OR REPLACE VIEW view\_employee

AS SELECT emp\_no,job\_id,e\_name,salary

FROM employee

WHERE emp\_no LIKE '%43567';

