Banarsidas Chandiwala Institute Of Information Technology

Database management system
LAB MANUAL

Submitted to :-
Dr. Sushma Bahuguna

Submitted by :-
Prachi Sharma
04911104422
Mca 1st semester(2022-24)

INDEX

Q1. Create table to store customer information and solve the queries:

CUST_ID	CUST_NAME	STATE	COUNTRY	AMOUNT	PHONE
3 digit	20 Characters	10 Characters	10 Chars	8 digits including 2 decimal places	11 digits
Key Field	All capital	Default=”Delhi”	Default=”India”	Between 2000 to 15000	
	Not Null				

- WAQ to select customer name and id of those customers belonging to Germany.
- WAQ to display complete information of customer whose amount > 3000.
- WAQ to select id and country of customer whose name contain a substring as “et”,
- WAQ to display the average of amount of all customers.
- WAQ to display the complete information of “Peter”.
- WAQ to display the information of customer whose amount > 5000 and less than 7000.
- WAQ to select state and id of customer whose name contain “h” as third character.
- WAQ to display the maximum amount.
- WAQ to display the complete information of customer(s) belongs to Australia.
- WAQ to display name of customer whose amount >2000 and < 5000.
- WAQ to select id and phone of customer whose name start with “pe”.
- WAQ to display the maximum amount for country “Germany”.
- WAQ to display the complete information of “Smith”.
- WAQ to select state and id of customer whose name contain “o” as second character.
- WAQ to select id and country of customer whose name contain a substring as “oh”,

Q2. Create a table to store bank information and solve the queries:

ID	NAME	BRANCH	ACCOUNT NO	INTEREST	AMOUNT
10	ICICI	Delhi	34	4	56000
20	HDFC	Agra	56	5	43255
30	SBI	Delhi	77	3	67345
40	ICICI	Jaipur	89	3	87623
50	YES	Nagpur	20	5	45500
60	SBI	Agra	561	4	43255
70	YES	Delhi	771	3	67345
80	ICICI	Jaipur	891	7	87600
90	YES	Nagpur	201	5	45200

- WAQ to display complete information for ICICI bank.
- WAQ to select id and name of bank whose amount > 50000.
- WAQ to select name of bank whose branch name has “pur” as a substring.
- WAQ to select maximum amount among all bank.
- WAQ to display name and branch of bank whose no. of account > 50.
- WAQ to display average of amount for Delhi branch.
- WAQ to select name of bank whose branch name has “g” as a substring.
- WAQ to select minimum amount among all bank.
- WAQ to display id, name of bank whose interest >5 and less than 8.
- WAQ to display branch name whose amount > 20000 and < 55000.
- WAQ to count ID of HDFC bank.
- WAQ to display the sum of amount for Delhi branch.

- WAQ to update Delhi branch by Bangalore where amount > 60000.
- WAQ to delete the information of yes bank.
- WAQ to display name of bank where branch is Delhi and whose amount> 50000.
- WAQ to display branch of banks belong to HDFC bank and city may be Agra or Jaipur.
- WAQ to select maximum amount of HDFC bank.
- WAQ to display complete information for Delhi branch.
- WAQ to find distinct bank name.
- WAQ to arrange the data according to amount available.
- WAQ to delete all data from bank table.
- WAQ to select name and ID of bank where ID belongs to hdfc or yes bank.
- WAQ to select name and branch of bank where no. of account between 50 and 90.
- WAQ to select complete details of all bank whose interest between 2 to 6 and belong to IDBI and HDFC bank.
- WAQ to add a new column "no_user" in bank table with char datatype.
- WAQ to modify the data type of "no_user" column from char to int.
- WAQ to update the value no_user = 5 for ICICI and HDFC bank.
- WAQ to list the details of bank whose no of user column contains null value.

Q3. Create the table as shown below and perform the following query:-

DeptNo	DName	Loc
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

empno	ename	job	mgr	hiredate	sal	comm	deptno
7839	KING	PRESIDENT	null	17-11-1981	5000	Null	10
7698	BLAKE	MANAGER	7839	1-5-1981	2850	Null	30
7782	CLARK	MANAGER	7839	9-6-1981	2450	Null	10
7566	JONES	MANAGER	7839	2-4-1981	2975	Null	20
7788	SCOTT	ANALYST	7566	13-JUL-87	3000	Null	20
7902	FORD	ANALYST	7566	3-12-1981	3000	Null	20
7369	SMITH	CLERK	7902	17-12-	800	null	20
7499	ALLEN	SALESMAN	7698	20-2-1981	1600	300	30
7521	WARD	SALESMAN	7698	22-2-1981	1250	500	30
7654	MARTIN	SALESMAN	7698	28-9-1981	1250	1400	30
7844	TURNER	SALESMAN	7698	8-9-1981	1500	0	30
7876	ADAMS	CLERK	7788	13-JUL-87	51	null	20
7900	JAMES	CLERK	7698	3-12-1981	950	null	30
7934	MILLER	CLERK	7782	23-1-1982	1300	null	10

- Display the names of all the employees who are working as clerks and drawing a salary more than 3000.
- Display the names of employees who are working as clerks,salesman or analyst and drawing a salary more than 3000.
- Display the list of employees who have joined the company before 30-JUN-90 or after 31-DEC-90.
- Display the names of employees working in depart number 10 or 20 or 40 or employees working as CLERKS,SALESMAN or ANALYST.
- Display name,salary,hra,pf,da,total salary for each employee. The output should be in the order of total salary,hra 15% of salary,da 10% of salary,pf 5% salary,total salary will be(salary+hra+da)-pf.
- Display depart numbers and total number of employees working in each department.
- Display the various jobs and total salary for each job.

- Display the total salary drawn by ANALYST working in depart number 40.
- Display the names of employees whose names have second alphabet A in their names.
- Display the maximum salary being paid to CLERK.
- Display the names of the employee in descending order of salary.
- Display the name of the employee along with their annual salary(sal*12).The name of the employee earning highest annual salary should apper first.
- Display the depart numbers and total salary for each department.
- Display the depart numbers and max salary for each department.
- Display the various jobs and total salary for each job.
- Display the depart numbers with more than three employees in each dept.
- Display the employee number and name for employee working as clerk and earning highest salary among clerks.
- Display the names of salesman who earns a salary more than the highest salary of any clerk.
- Display the names of clerks who earn a salary more than the lowest salary of any salesman.
- Display the names of the employees who earn highest salary in their respective departments.
- Display the employee names who are working in accounting department.
- Display the names of employees from department number 10 with salary greater than that of any employee working in other department.
- Display the names of the employees from department number 10 with salary greater than that of all employee working in other departments.
- Display the maximum salary being paid to depart number 20.
- Display the average salary drawn by MANAGERS.
- Select Avg(Sal) from emp where Joj< {01/08/81};
- Select sum (fee) from student where where Joj> {01/08/81};

Q4. Given the following tables for a database LIBRARY:

Book_ID	Book_Name	Author_Nme	Publishers	Price	Type	Qty
C0001	Fast Cook	Lata Kapoor	EPB	355	Cookery	5
F0001	The Tears	William Hopkins	First Publ.	650	Fiction	20
T0001	My first c++	Brian & Brooke	EPB	350	Text	10
T0002	C++ Brainworks	A.W. Rossaine	TDH	350	Text	15
F0002	Thunderbolts	Anna Roberts	First publ.	750	Fiction	50

Book_id	Quantity_Issued
T0001	4
C0001	5
F0001	2

Write SQL statements for:-

- To show book name, author name and price of book of First Publ. publishers.
- To list the names from books of text type.
- To display the names and price from books in ascending order of their price.
- To increase the price of all books of EPB publishers by 50.
- To display the Book_Id, Book_Name and Quantity_Issued for all books which have been issued.
- To insert a new row in the table issued during the following data: “F0003”,1
- Give the output for the following SQL queries:

    1. select count(*) from book.
    2. select max(Price) from books where quantity >= 15.
    3. select book_Name, Author_Name from book where Publishers = “EPB”.
    4. select count (Distinct Publishers) from books where price > = 400;

Q5. With references to following relations PERSONAL and JOB answer the questions that follow:

Create following tables such that empno and sno are not null and unique, date of birth is after '12-Jan-1960', name is never blank, area and Native place is valid, hobby, dept is not empty, salary is between 4000 and 10000.

Empno	Name	DoBirth	Native_Place	Hobby
123	Amit	23-jan-1965	delhi	music
127	Manoj	12-dec-1976	mumbai	writing
124	Abhai	11-aug-1975	allahabad	music
125	Vinod	04-apr-1977	delhi	Sports
128	Abhay	10-mar-1974	mumbai	grdening
129	ramesh	28-ovt-1981	pune	sports

Sno	Area	App_date	Salary	Retd_date	Dept
123	Agra	25-jan-2006	5000	25-jan-2026	Marketing
127	Mathura	22-dec-2006	6000	22-dec-202	Finance
124	Agra	19-aug-2007	5500	19-aug-202	Marketing
125	Delhi	14-apr-2004	8500	14-apr-2018	Sales
128	pune	13-mar-2008	7500	13-mar-2028	Sales

- Show empno, name and salary of those who have sports as hobby.
- Show name of the eldest employee.
- Show number of employee area wise.
- Show youngest employees from ache native place.
- Show sno, name, hobby and salary in descending order of salary.
- Show the hobbies of those whose name pronounces as 'Abhay'.
- Show the appointment date and native place of those whose name starts with 'A' or ends in 'd'.
- Show the salary expense with suitable column heading of those who shall retire after 20-jan-2006.
- Show names of those who earn more than all of the employees of sales department.
- Increase salary of the employees by 5% of their present salary with hobby as music or they have completed atleast 3 years of services.

Q6. Write Pl/SQL code for

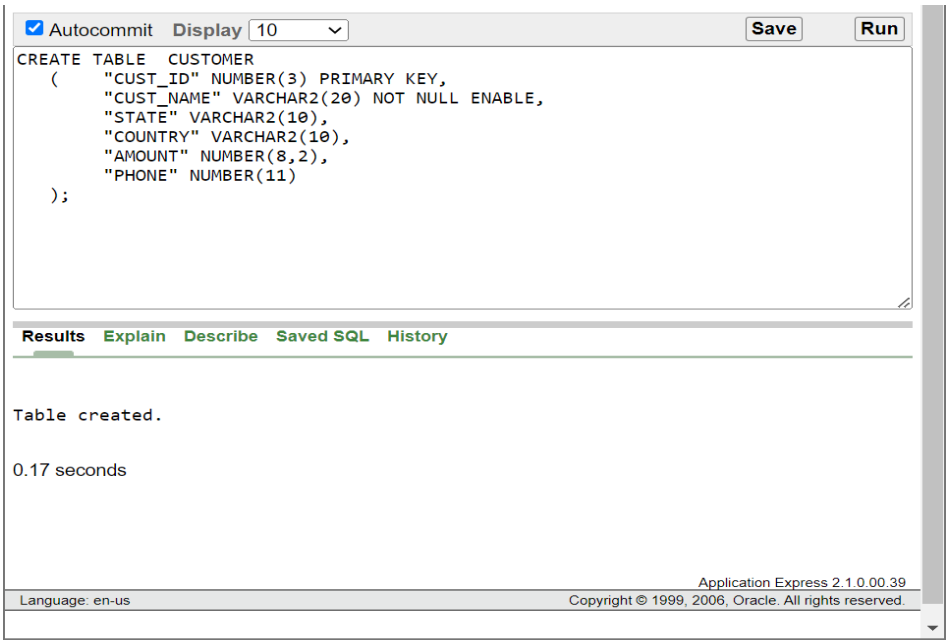
- To reverse a number and print, i.e, if num is 677 then it should print 776.
- To print a Fibonacci series.
- To check a number is Armstrong or not.
- To print the factorial of a given number.
- To evaluate whether a given number is prime or not.
- To perform the addition of two numbers.
- To get a number from keyboard and if it zero print "natural number", else print "not a natural number".
- To find the area and perimeter of given circle.
- To calculate the net salary if dfa is 30% of basic, hra is 10% of basic and pf is 7%. If basic salary is less than 8000, pf is 10% if basic sal between 8000 to 160000.
- To select record of emp table with cursor.
- To raise an error if no data found.

Q7. Write and explain the following Pl/SQL triggers on emp table

- Before UPDATE Trigger
- Before DELETE Trigger
- Before INSERT Trigger
- After UPDATE Trigger
- After DELETE Trigger
- After INSERT Trigger

1. Create table to store customer information and solve the queries:

CUST_ID	CUST_NAME	STATE	COUNTRY	AMOUNT	PHONE
3 digit	20 Characters	10 Characters	10 Chars	8 digits including 2 decimal places	11 digits
Key Field	All capital	Default=”Delhi”	Default=”India ”	Between 2000 to 15000	
	Not Null				



- Insert data in table “CUSTOMER”

insert into customer values(001,'Peter','Delhi','india',9543.15,9070405080);

insert into customer values(002,'John','Bavaria','Germany',9800,5040603010);

insert into customer values(003,'Peeyush','California','USA',8625,107080904);

insert into customer values(004,'Scarlett','London','England',13564,5054786945);

insert into customer values(005,'Rohan','Queensland','Australia',2598.515);

```
insert into customer values(006,'Suhesh','Mumbai','India',6845,5498419812);
insert into customer values(007,'Smith','Paris','France',5684.864,48496169498);
insert into customer values(008,'boii','Delhi','india',2458.15,9845965080);
insert into customer values(009,'sumeet','Bihar','india',6482.82,9858462130);
insert into customer values(010,'Harshet','Bavaria','Germany',4865.18,null);
insert into customer values(011,'Gorav','Bremen','Germany',11235,9549804985);
insert into customer values(013,'Richaard','caniel','Australia',11235,8976534284);
insert into customer values(014,'pooja','Delhi','India',9123.15,80700405080);
```

- **All data of table "CUSTOMER"**

CUST_ID	CUST_NAME	STATE	COUNTRY	AMOUNT	PHONE
1	Peter	Delhi	india	9543.15	9070405080
2	John	Bavaria	Germany	9800	5040603010
4	Scarlett	London	England	13564	5054786945
6	Suhesh	Mumbai	India	6845	5498419812
7	Smith	Paris	France	5684.86	48496169498
8	boii	Delhi	india	2458.15	9845965080
9	sumeet	Bihar	india	6482.82	9858462130
10	Harshet	Bavaria	Germany	4865.18	-
11	Gorav	Bremen	Germany	11235	9549804985

Download CSV

9 rows selected.

1. **WAQ to select customer name and id of those customers belonging to Germany.**

CUST_ID	CUST_NAME
2	John
10	Harshet
11	Gorav

Download CSV

3 rows selected.

2. **WAQ to display complete information of customer whose amount > 3000.**

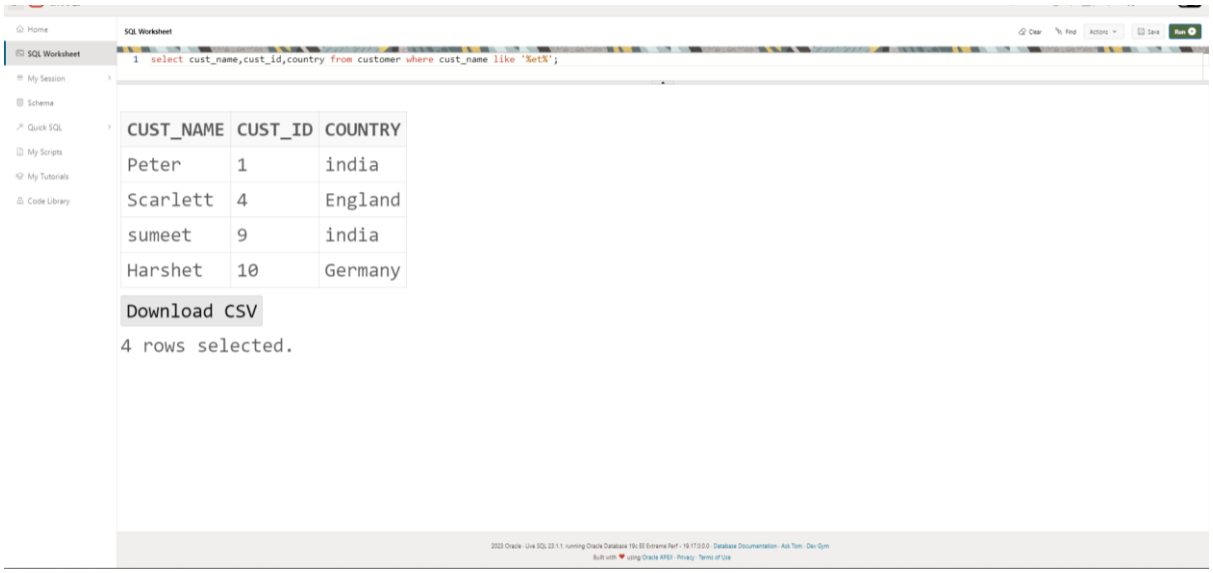
```
select * from customer where amount>3000;
```

CUST_ID	CUST_NAME	STATE	COUNTRY	AMOUNT	PHONE
1	Peter	Delhi	india	9543.15	9070405080
2	John	Bavaria	Germany	9800	5040603010
4	Scarlett	London	England	13564	5054786945
6	Suhesh	Mumbai	India	6845	5498419812
7	Smith	Paris	France	5684.86	48496169498
9	sumeet	Bihar	india	6482.82	9858462130
10	Harshet	Bavaria	Germany	4865.18	-
11	Gorav	Bremen	Germany	11235	9549804985

Download CSV

8 rows selected.

3. WAQ to select id and country of customer whose name contain a substring as “et”



```
select cust_name,cust_id,country from customer where cust_name like '%et%';
```

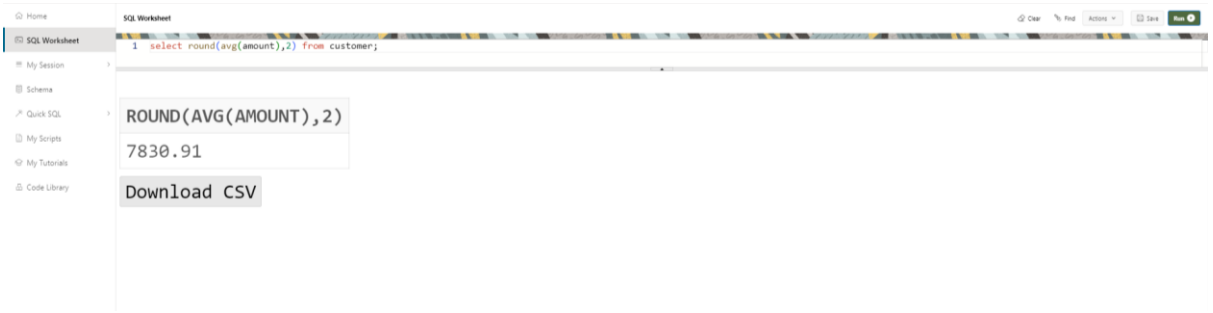
CUST_NAME	CUST_ID	COUNTRY
Peter	1	india
Scarlett	4	England
sumeet	9	india
Harshet	10	Germany

Download CSV

4 rows selected.

4. WAQ to display the average of amount of all customers.

SELECT AVG(AMOUNT) FROM customers;

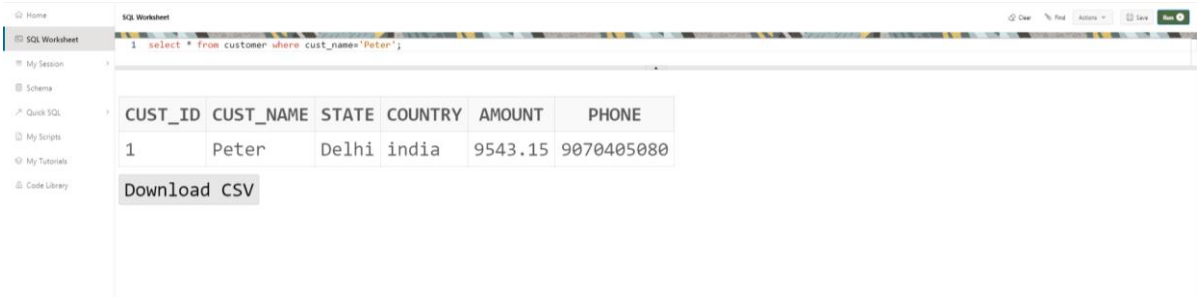


```
select round(avg(amount),2) from customer;
```

ROUND(AVG(AMOUNT),2)
7830.91

Download CSV

5. WaQ to display the complete information of “Peter”.

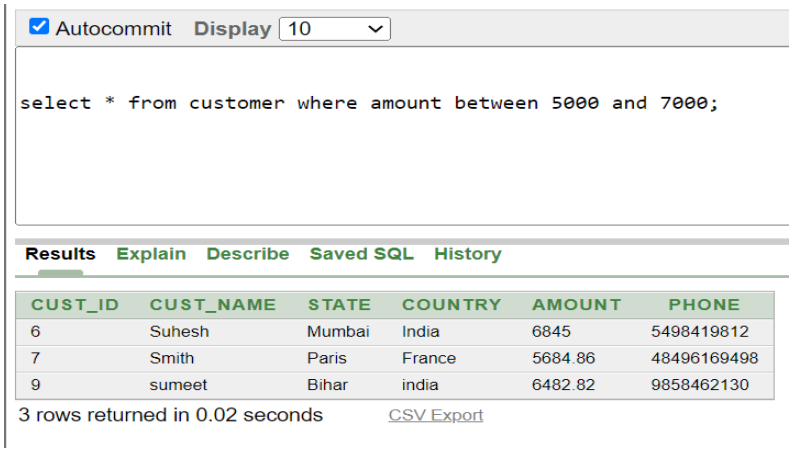


```
select * from customer where cust_name='Peter';
```

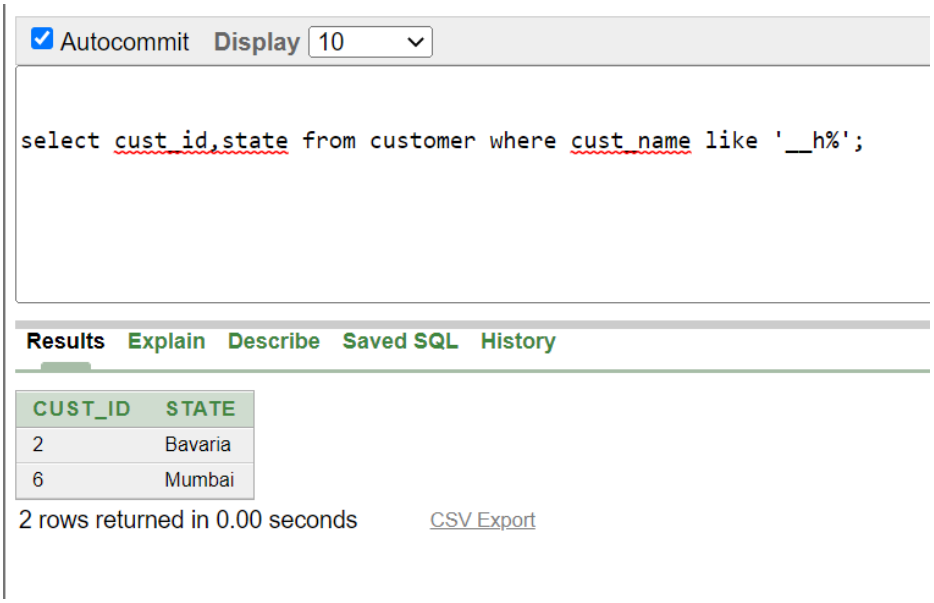
CUST_ID	CUST_NAME	STATE	COUNTRY	AMOUNT	PHONE
1	Peter	Delhi	india	9543.15	9070405080

Download CSV

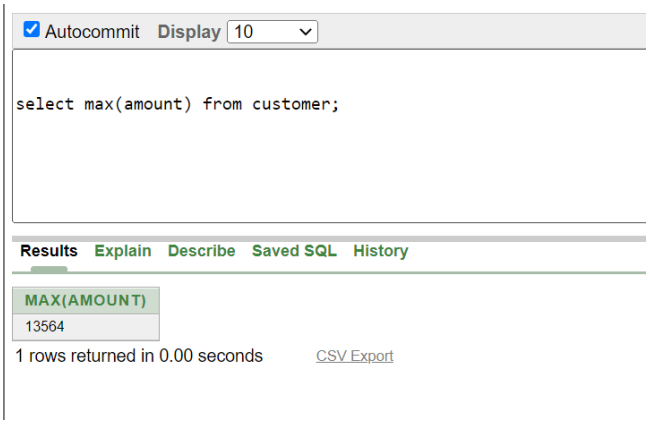
6. **WAQ to display the information of customer whose amount > 5000 and less than 7000.**



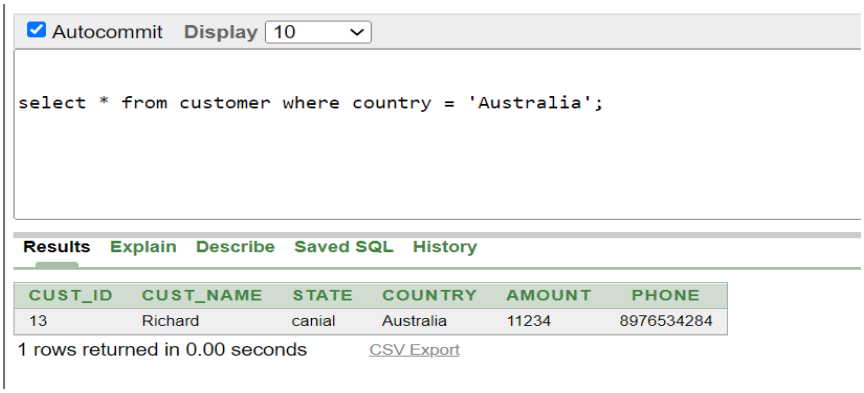
7. **WAQ to select state and id of customer whose name contain "h" as third character.**



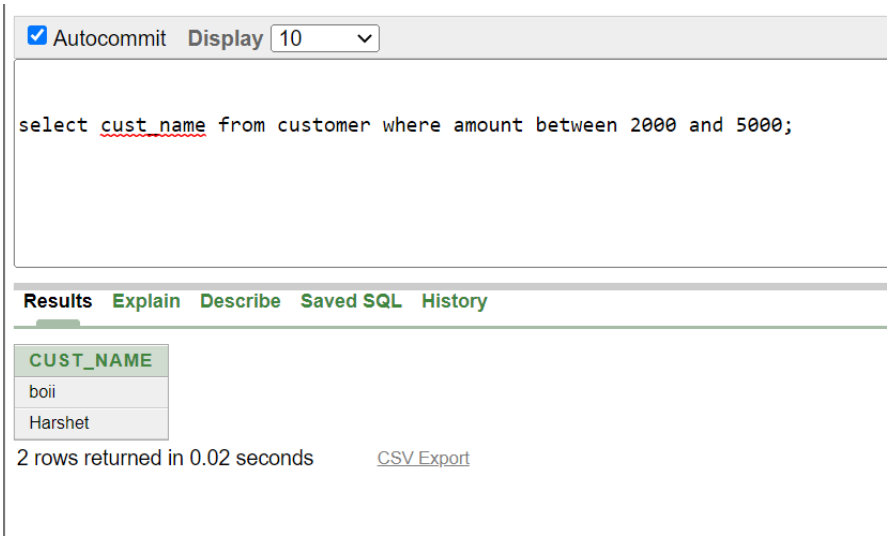
8. **WAQ to display the maximum amount.**



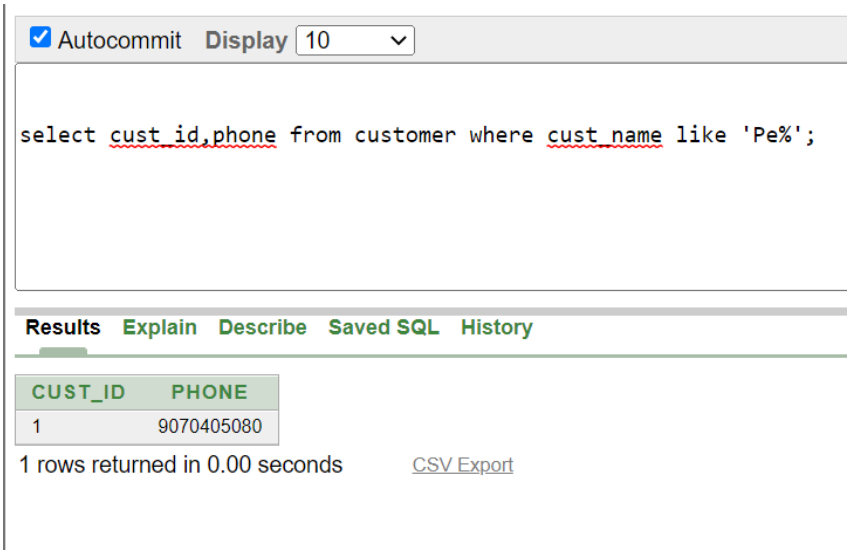
9. **WAQ to display the complete information of customer(s) belongs to Australia.**



10. **WAQ to display name of customer whose amount >2000 and amount < 5000.**

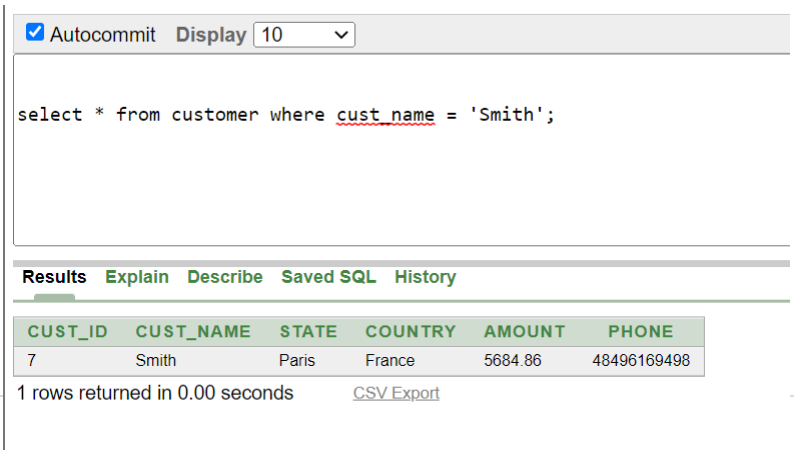


11. **WAQ to select id and phone of customer whose name start with “pe”.**

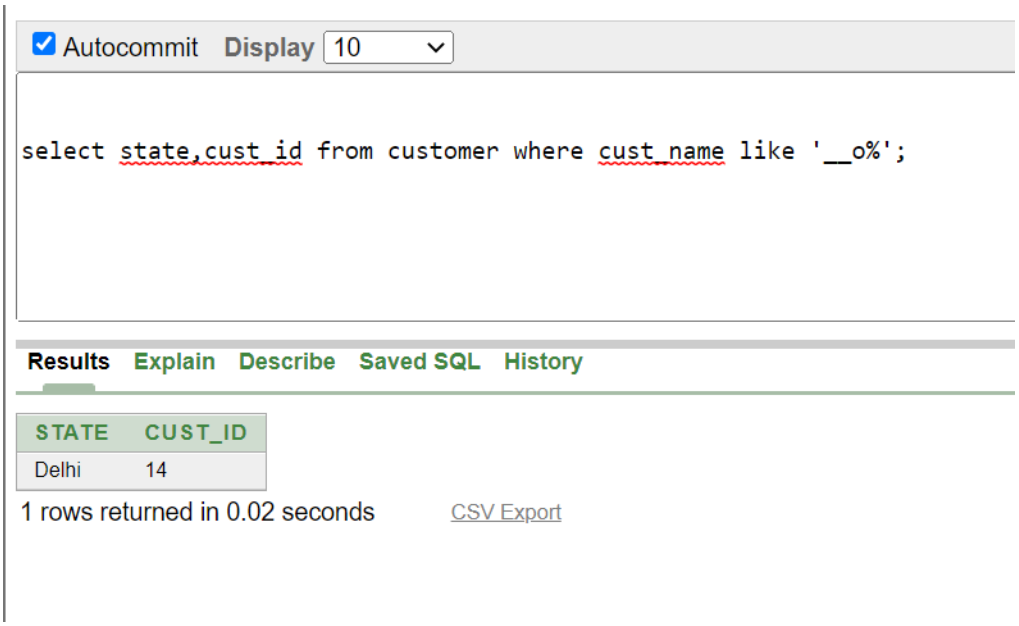


12. **WAQ to display the maximum amount for country “Germany”.**

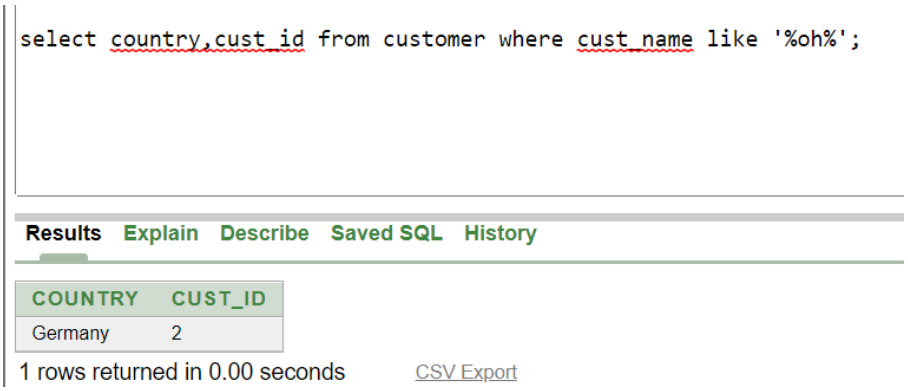
13. **WAQ to display the complete information of “Smith”.**



14. WAQ to select state and id of customer whose name contain “o” as second character.



15. WAQ to select id and country of customer whose name contain a substring as “oh”.



Q2. Create a table to store bank information and solve the queries:

ID	NAME	BRANCH	ACCOUNT NO	INTEREST	AMOUNT
10	ICICI	Delhi	34	4	56000
20	HDFC	Agra	56	5	43255
30	SBI	Delhi	77	3	67345
40	ICICI	Jaipur	89	3	87623
50	YES	Nagpur	20	5	45500
60	SBI	Agra	561	4	43255
70	YES	Delhi	771	3	67345
80	ICICI	Jaipur	891	7	87600
90	YES	Nagpur	201	5	45200

```
CREATE table bank (
"ID"        NUMBER(3) primary key,
"NAME"      VARCHAR2(20) NOT NULL,
"BRANCH"    VARCHAR2(20) NOT NULL,
"ACC_NO"    NUMBER(3) NOT NULL,
"INTEREST" NUMBER(3) NOT NULL,
"AMOUNT"    NUMBER(3) NOT NULL
);
```

Insert data in table "bank".

```
insert into bank values(10,'ICICI','Delhi',34,4,56000);
insert into bank values(20,'HDFC','Agra',56,5,43255);
insert into bank values(30,'SBI','Delhi',77,3,67345);
insert into bank values(40,'ICICI','Jaipur',89,3,87623);
insert into bank values(50,'YES','Nagpur',20,5,45500);
insert into bank values(60,'SBI','Agra',561,4,43255);
insert into bank values(70,'YES','Delhi',771,3,67345);
insert into bank values(80,'ICICI','Jaipur',891,7,87600);
insert into bank values(90,'YES','Nagpur',201,5,45200);
```

WAQ to display complete information for ICICI bank.

1. SELECT * FROM bank WHERE name = 'ICICI';

```
select * from bank where name='ICICI';
```

Results | Explain | Describe | Saved SQL | History

ID	NAME	BRANCH	ACC_NO	INTEREST	AMOUNT
10	ICICI	Delhi	34	4	56000
40	ICICI	Jaipur	89	3	87623
80	ICICI	Jaipur	891	7	87600

3 rows returned in 0.03 seconds CSV Export

2. **WAQ to select id and name of bank whose amount > 50000.**

```
select id,name from bank where amount>50000;
```

Results Explain Describe Saved SQL History

ID	NAME
10	ICICI
30	SBI
40	ICICI
70	YES
80	ICICI

5 rows returned in 0.00 seconds CSV Export

3. **WAQ to select name of bank whose branch name has "pur" as a substring.**

```
select distinct(name) from bank where branch like '%pur%';
```

Results Explain Describe Saved SQL History

NAME
ICICI
YES

2 rows returned in 0.02 seconds CSV Export

4. **WAQ to select maximum amount among all bank.**

```
select max(amount) from bank;
```

Results Explain Describe Saved SQL History

MAX(AMOUNT)
87623

1 rows returned in 0.00 seconds CSV Export

5. **WAQ to display name and branch of bank whose no. of account > 50.**

```
select name,branch from bank where acc_no>50;
```

Results Explain Describe Saved SQL History

NAME	BRANCH
HDFC	Agra
SBI	Delhi
ICICI	Jaipur
SBI	Agra
YES	Delhi
ICICI	Jaipur
YES	Nagpur

7 rows returned in 0.02 seconds CSV Export

6. **WAQ to display average of amount for Delhi branch.**

```
select round(avg(amount),2) from bank where branch='Delhi';
```

Results Explain Describe Saved SQL History

ROUND(AVG(AMOUNT),2)
63563.33

1 rows returned in 0.00 seconds CSV Export

7. **WAQ to select name of bank whose branch name has "g" as a substring.**

```
select name from bank where branch like '%g%';
```

Results Explain Describe Saved SQL History

NAME
HDFC
YES
SBI
YES

4 rows returned in 0.00 seconds CSV Export

8. **WAQ to select minimum amount among all bank.**

```
select min(amount) from bank ;
```

Results Explain Describe Saved SQL History

MIN(AMOUNT)
43255

1 rows returned in 0.00 seconds CSV Export

9. **WAQ to display id, name of bank whose interest >5 and less than 8.**

```
select id,name from bank where interest>5 and interest<(8);
```

Results Explain Describe Saved SQL History

ID	NAME
80	ICICI

1 rows returned in 0.01 seconds CSV Export

10. **WAQ to display branch name whose amount > 20000 and < 55000**

```
select branch from bank where amount>20000 and amount<(55000);
```

Results Explain Describe Saved SQL History

BRANCH
Agra
Nagpur
Agra
Nagpur

4 rows returned in 0.00 seconds CSV Export

11. WAQ to count ID of HDFC bank.

```
select count(id) from bank where name='HDFC';
```

Results Explain Describe Saved SQL History

COUNT(ID)
1

1 rows returned in 0.00 seconds CSV Export

12. WAQ to display the sum of amount for Delhi branch.

```
select sum(amount) from bank where branch='Delhi';
```

Results Explain Describe Saved SQL History

SUM(AMOUNT)
190690

1 rows returned in 0.00 seconds CSV Export

13. WAQ to update Delhi branch by Bangalore where amount > 60000.

```
update bank set branch='Bangalore' where branch='Delhi' and amount >60000;
```

Results Explain Describe Saved SQL History

```
2 row(s) updated.
```

0.00 seconds

14. WAQ to delete the information of yes bank.

```
delete from bank where name='YES';
```

Results Explain Describe Saved SQL History

3 row(s) deleted.

0.00 seconds

15. WAQ to display name of bank where branch is Delhi and whose amount> 50000.

```
select name from bank where branch='Delhi' and amount>50000;
```

Results Explain Describe Saved SQL History

NAME
ICICI

1 rows returned in 0.02 seconds CSV Export

16. WAQ to display branch of banks belong to HDFC bank and city may be Agra or Jaipur.

```
select max(amount) from bank where name='HDFC';
```

Results Explain Describe Saved SQL History

MAX(AMOUNT)
43255

1 rows returned in 0.00 seconds CSV Export

17. WAQ to select maximum amount of HDFC bank.

```
select branch from bank where name='HDFC' and branch in('Agra','Jaipur');
```

Results Explain Describe Saved SQL History

BRANCH
Agra

1 rows returned in 0.00 seconds CSV Export

18. WAQ to display complete information for Delhi branch.

```
select * from bank where branch='Delhi';
```

Results Explain Describe Saved SQL History

ID	NAME	BRANCH	ACC_NO	INTEREST	AMOUNT
10	ICICI	Delhi	34	4	56000

1 rows returned in 0.01 seconds CSV Export

19. WAQ to find distinct bank name.

```
select distinct(name) from bank;
```

Results Explain Describe Saved SQL History

NAME
ICICI
HDFC
SBI

3 rows returned in 0.00 seconds CSV Export

20. WAQ to arrange the data according to amount available.

```
select * from bank order by amount;
```

Results Explain Describe Saved SQL History

ID	NAME	BRANCH	ACC_NO	INTEREST	AMOUNT
20	HDFC	Agra	56	5	43255
60	SBI	Agra	561	4	43255
10	ICICI	Delhi	34	4	56000
30	SBI	Bangalore	77	3	67345
80	ICICI	Jaipur	891	7	87600
40	ICICI	Jaipur	89	3	87623

6 rows returned in 0.01 seconds CSV Export

21. WAQ to delete all data from bank table

```
truncate table bank;
```

Results Explain Describe Saved SQL History

```
Table truncated.
```

0.03 seconds

22. WAQ to select name and ID of bank where ID belongs to hdfc or yes bank.

```
select id,name from bank where name in('HDFC','YES');
```

Results Explain Describe Saved SQL History

ID	NAME
20	HDFC

1 rows returned in 0.01 seconds CSV Export

23. WAQ to select name and branch of bank where no. of account between 50 and 90.

```
select name,branch from bank where acc_no>50 and acc_no<(90);
```

Results Explain Describe Saved SQL History

NAME	BRANCH
HDFC	Agra
SBI	Bangalore
ICICI	Jaipur

3 rows returned in 0.00 seconds CSV Export

24. WAQ to select complete details of all bank whose interest between 2 to 6 and belong to IDBI and HDFC bank.

```
select * from bank where interest between 2 and 6 and name in('HDFC','IDBI');
```

Results Explain Describe Saved SQL History

ID	NAME	BRANCH	ACC_NO	INTEREST	AMOUNT
20	HDFC	Agra	56	5	43255

1 rows returned in 0.01 seconds CSV Export

25.

26.

27.

28.

25. WAQ to add a new column "no_user" in bank table with char datatype.

```
alter table bank add (no_user char);
```

Results Explain Describe Saved SQL History

Table altered.

0.03 seconds

26. WAQ to modify the data type of "no_user column from char to int.

```
alter table bank modify (no_user number(5));
```

Results Explain Describe Saved SQL History

Table altered.

0.03 seconds

27. WAQ to update the value no_user = 5 for ICICI and HDFC bank.

```
update bank set no_user=5 where name in('HDFC','ICICI');
```

Results Explain Describe Saved SQL History

```
4 row(s) updated.
```

0.02 seconds

28. WAQ to list the details of bank whose no of user column contains null value.

```
select * from bank where no_user is null;
```

Results Explain Describe Saved SQL History

ID	NAME	BRANCH	ACC_NO	INTEREST	AMOUNT	NO_USER
30	SBI	Bangalore	77	3	67345	-
60	SBI	Agra	561	4	43255	-

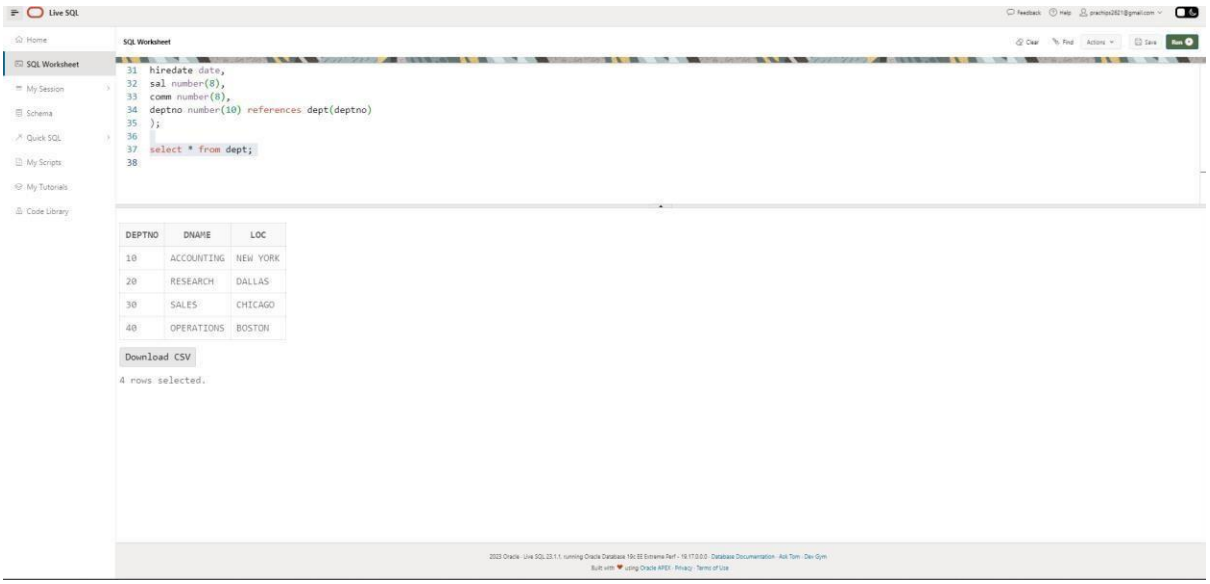
2 rows returned in 0.02 seconds CSV Export

Q3. Create the table as shown below and perform the following query:-

DEPTNO	DName	loc
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

empno	ename	job	mgr	hiredate	sal	comm	deptno
7839	KING	PRESIDENT	null	17-11-1981	5000	Null	10
7698	BLAKE	MANAGER	7839	1-5-1981	2850	Null	30
7782	CLARK	MANAGER	7839	9-6-1981	2450	Null	10
7566	JONES	MANAGER	7839	2-4-1981	2975	Null	20
7788	SCOTT	ANALYST	7566	13-JUL-87	3000	Null	20
7902	FORD	ANALYST	7566	3-12-1981	3000	Null	20
7369	SMITH	CLERK	7902	17-12-	800	null	20
7499	ALLEN	SALESMAN	7698	20-2-1981	1600	300	30
7521	WARD	SALESMAN	7698	22-2-1981	1250	500	30
7654	MARTIN	SALESMAN	7698	28-9-1981	1250	1400	30
7844	TURNER	SALESMAN	7698	8-9-1981	1500	0	30
7876	ADAMS	CLERK	7788	13-JUL-87	51	null	20
7900	JAMES	CLERK	7698	3-12-1981	950	null	30
7934	MILLER	CLERK	7782	23-1-1982	1300	null	10

```
CREATE TABLE DEPT (
DEPTNO number(5) PRIMARY KEY,
DNAME VARCHAR2(20) NOT NULL,
LOC VARCHAR2(20) NOT NULL
);

INSERT INTO DEPT VALUES (10 ,'ACCOUNTING' ,'NEW YORK');
INSERT INTO DEPT VALUES (20, 'RESEARCH', 'DALLAS'); INSERT INTO DEPT
VALUES (30 ,'SALES' ,'CHICAGO');
 INSERT INTO DEPT VALUES (40, 'OPERATIONS', 'BOSTON');
```



CREATE TABLE EMP (

EMPNO number(5) PRIMARY KEY,

ENAME VARCHAR2(20) NOT NULL,

JOB VARCHAR2(20) NOT NULL,

MGR number(5),

HIREDATE DATE NOT NULL,

SAL number(8, 2) NOT NULL,

COMM number(8, 2),

DEPTNO number(5),

FOREIGN KEY (DEPTNO) REFERENCES DEPT(DEPTNO)

);

INSERT INTO EMP VALUES(7934, 'MILLER', 'CLERK', 7782 ,'23-JAN-1982', 1300, null, 10);

INSERT INTO EMP VALUES(7900, 'JAMES','CLERK', 7698, '3-DEC-1981' ,950, null, 30);

INSERT INTO EMP VALUES(7876, 'ADAM', 'CLERK' ,7788, '13-JUL-1987', 51, null ,20);

INSERT INTO EMP VALUES(7844 ,'TURNER', 'SALESMAN', 7698, '8-SEP-1981', 1500, 0 ,30);

INSERT INTO EMP VALUES(7554, 'MARTIN', 'SALESMAN', 7698, '28-SEP-1981', 1250, 1400, 30);

INSERT INTO EMP VALUES(7521, 'WARD', 'SALESMAN', 7698, '22-FEB-1981', 1250, 500, 30);

INSERT INTO EMP VALUES(7499, 'ALLEN', 'SALESMAN', 7698, '20-FEB-1981', 1600, 300, 30);

INSERT INTO EMP VALUES(7369, 'SMITH', 'CLERK', 7902, '17-DEC-1980', 800, NULL, 20);

INSERT INTO EMP VALUES(7902, 'FORD', 'ANALYST', 7566, '03-DEC-1981', 3000, NULL, 20);

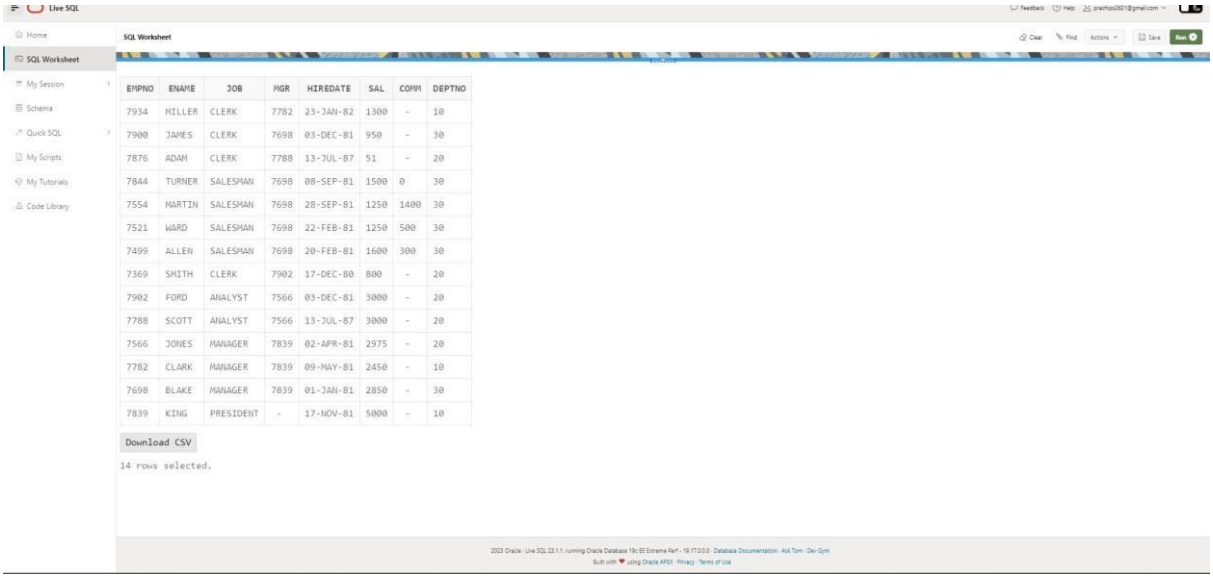
INSERT INTO EMP VALUES(7788, 'SCOTT', 'ANALYST', 7566, '13-JUL-1987', 3000, NULL, 20);

INSERT INTO EMP VALUES(7566, 'JONES', 'MANAGER', 7839, '02-APR-1981', 2975, NULL, 20);

INSERT INTO EMP VALUES(7782, 'CLARK', 'MANAGER', 7839, '09-MAY-1981', 2450, NULL, 10);

INSERT INTO EMP VALUES(7698, 'BLAKE', 'MANAGER', 7839, '01-JAN-1981', 2850, NULL, 30);

INSERT INTO EMP VALUES(7839, 'KING', 'PRESIDENT', NULL, '17-NOV-1981', 5000, NULL, 10);



EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7934	MILLER	CLERK	7782	23-JAN-82	1300	-	10
7900	JAMES	CLERK	7698	03-DEC-81	950	-	30
7876	ADAM	CLERK	7788	13-JUL-87	51	-	20
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7554	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7369	SMITH	CLERK	7902	17-DEC-80	800	-	20
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20
7788	SCOTT	ANALYST	7566	13-JUL-87	3000	-	20
7566	JONES	MANAGER	7839	02-APR-81	2975	-	20
7782	CLARK	MANAGER	7839	09-MAY-81	2450	-	10
7698	BLAKE	MANAGER	7839	01-JAN-81	2850	-	30
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10

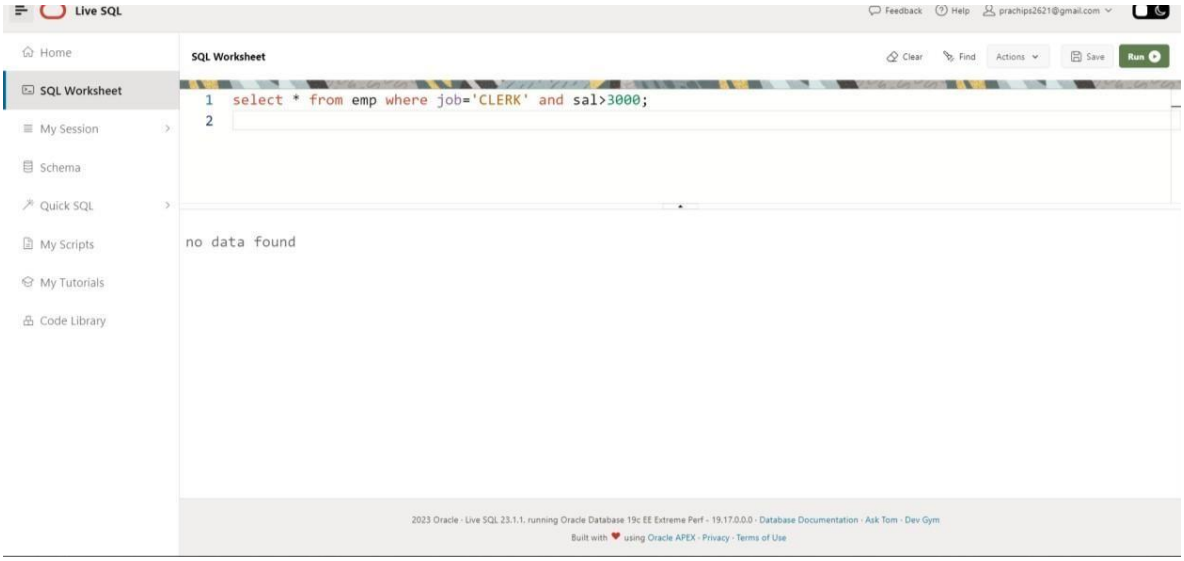
Download CSV

14 rows selected.

QUERY:

1. 2Display the names of all the employees who are working as clerks and drawing a salary more than 3000.

   SELECT ENAME FROM EMP WHERE JOB='CLERK' AND SAL > 3000;

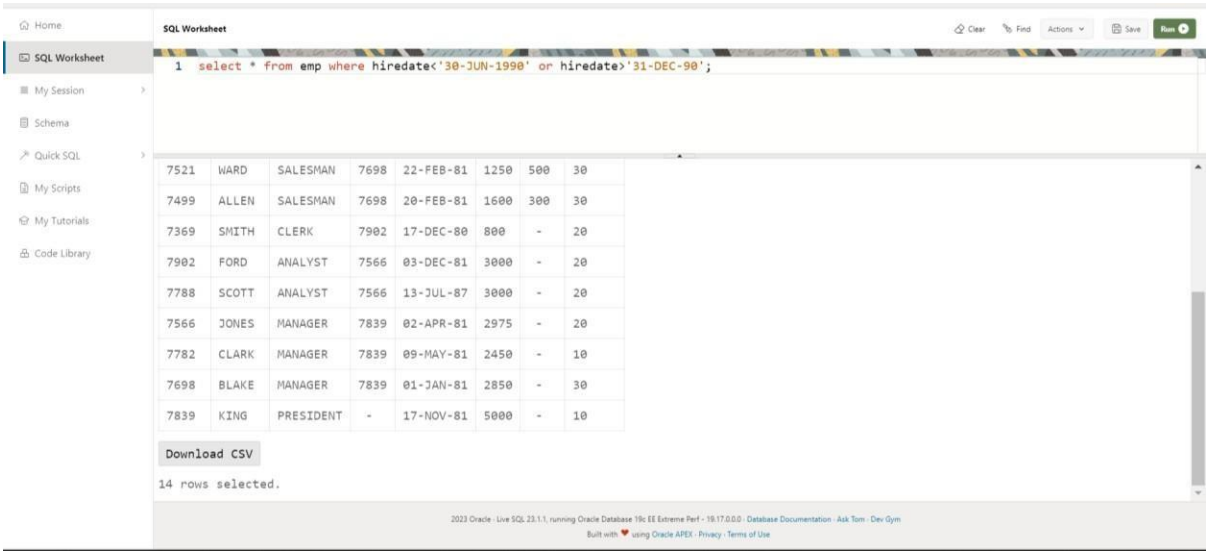


2. Display the names of employees who are working as clerks,salesman or analyst and drawing a salary more than 3000.

SELECT ENAME FROM EMP WHERE JOB IN ('CLERK','SALESMAN','ANALYST') AND SAL > 3000;

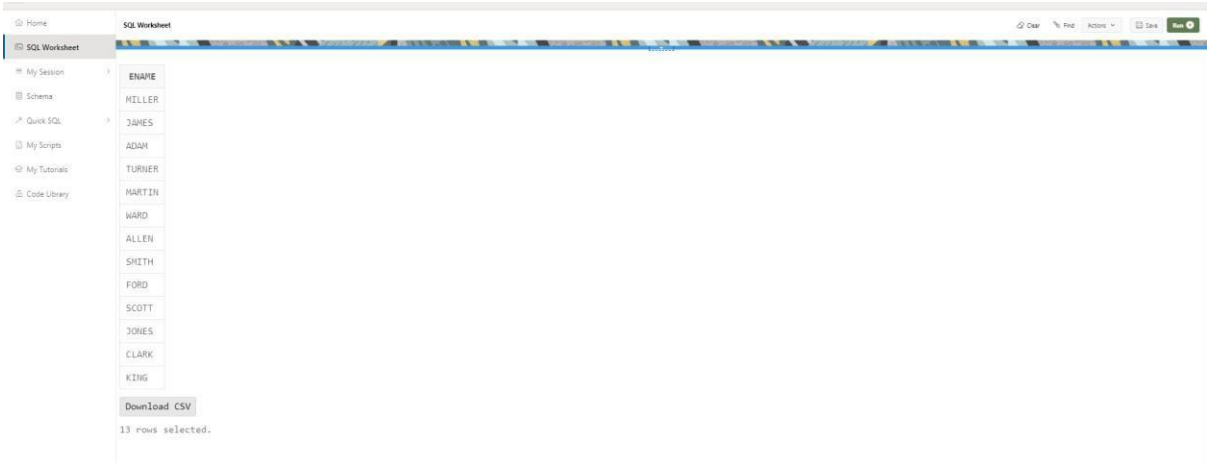
3. Display the list of employees who have joined the company before 30-JUN-90 or after 31-DEC-90.

SELECT ENAME FROM EMP WHERE HIREDATE < '30-MAY-1990' OR HIREDATE > '31-DEC-1990-12';



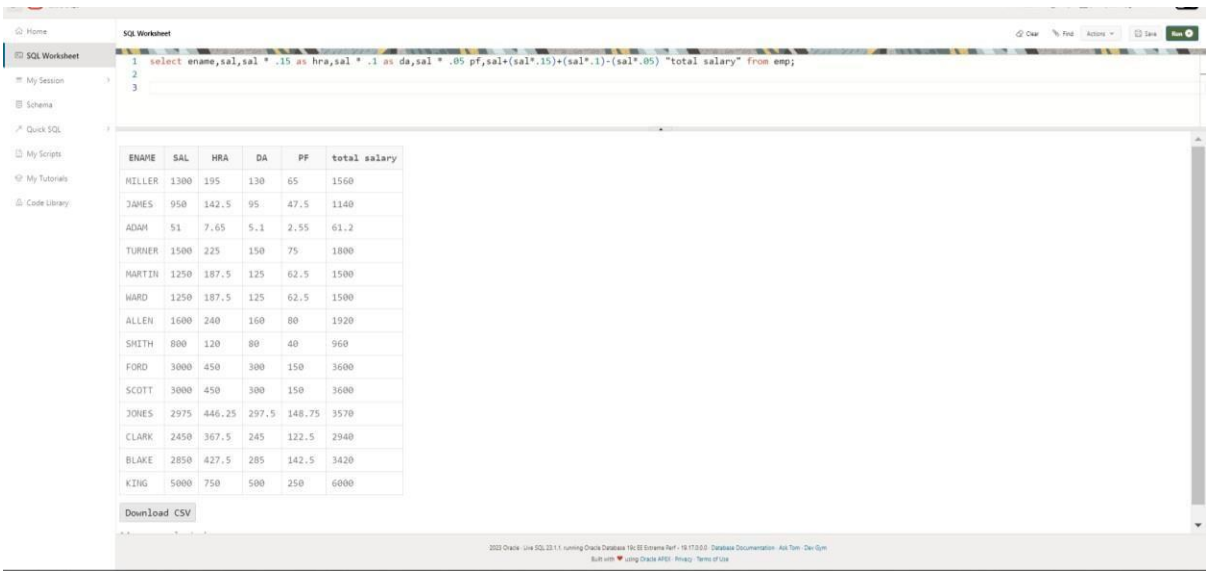
4. Display the names of employees working in depart number 10 or 20 or 40 or employees working asCLERKS,SALESMAN or ANALYST.

SELECT ENAME FROM EMP WHERE DEPTNO IN (10, 20, 40) OR JOB IN ('CLERK','SALESMAN','ANALYST');



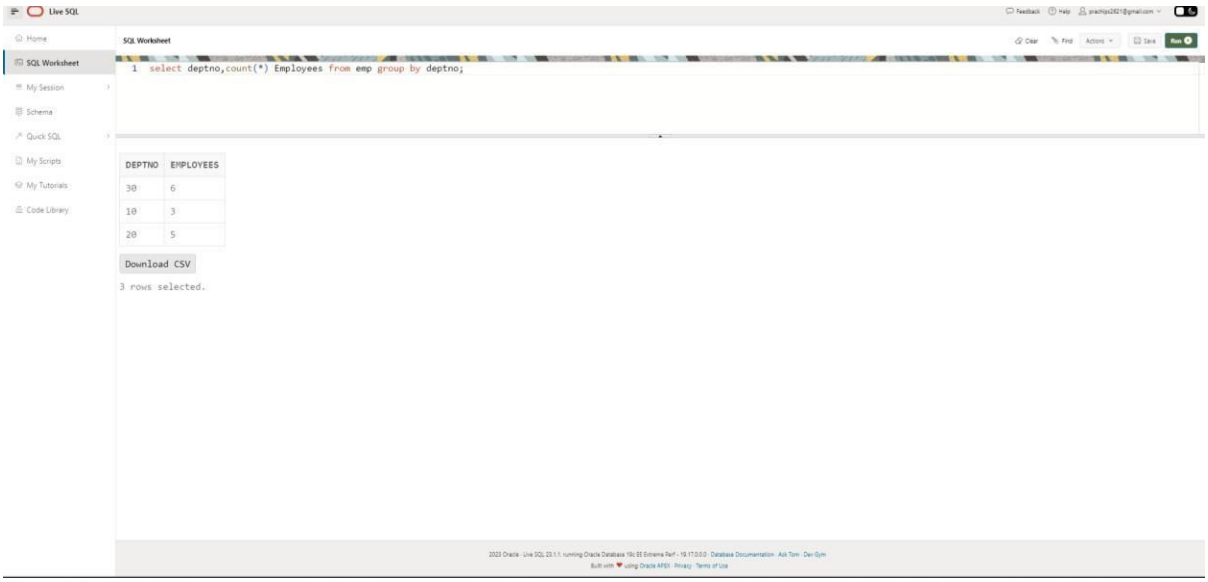
5. Display name,salary,hra,pf,da,total salary for each employee. The output should be in the order of total salary,hra15% of salary,da 10% of salary,pf 5% salary,total salary will be(salary+hra+da)-pf.

SELECT ENAME, SAL, SAL*0.15 AS HRA, SAL*0.05 AS PF, SAL*0.1 AS DA, (SAL + SAL*0.15 + SAL*0.1 - SAL*0.05) AS TOTAL_SALARY FROM EMP ORDER BY TOTAL_SALARY;



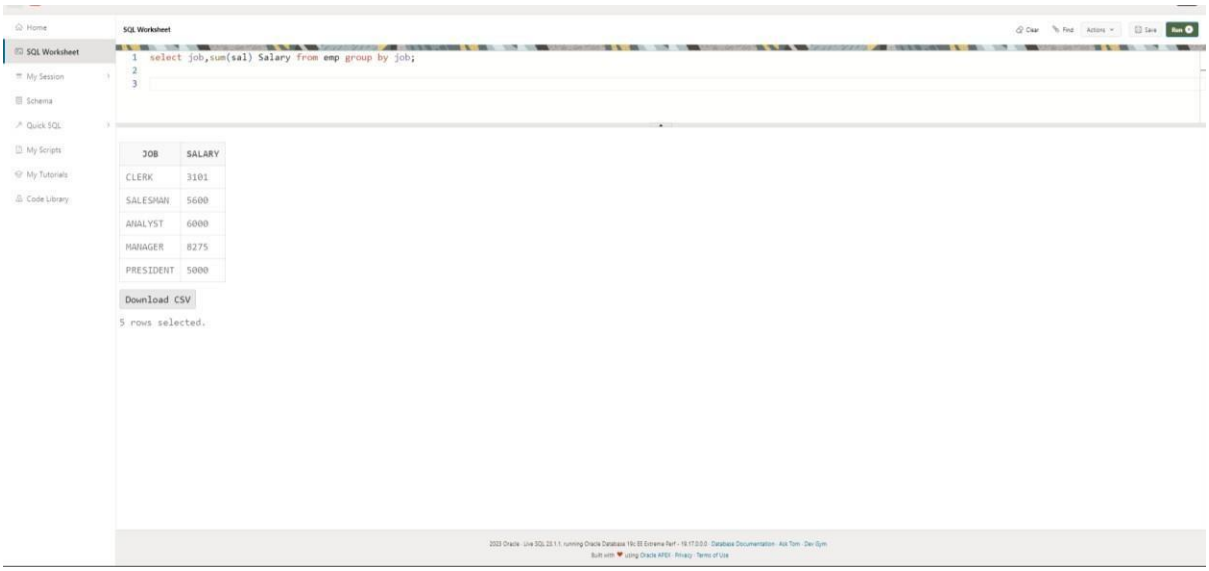
6. Display depart numbers and total number of employees working in each department.

SELECT DEPTNO, COUNT(DEPTNO) “No Of Emp” FROM EMP GROUP BY DEPTNO;



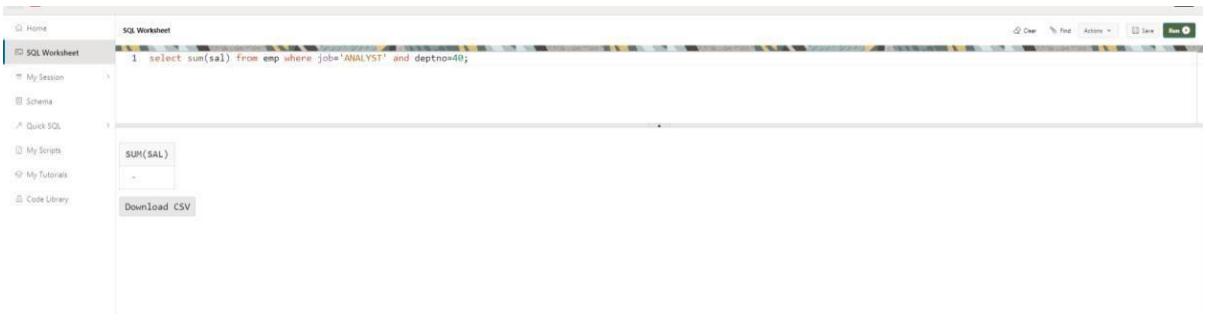
7. Display the various jobs and total salary for each job.

SELECT JOB , SUM(SAL) FROM EMP GROUP BY JOB;



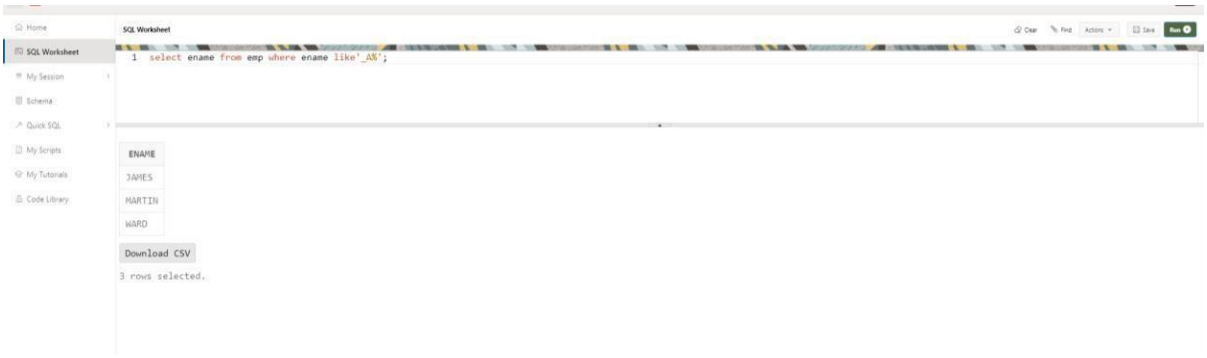
8. Display the total salary drawn by ANALYST working in depart number 40.

SELECT SUM(SAL) FROM EMP WHERE JOB='ANALYST' AND DEPTNO=40;



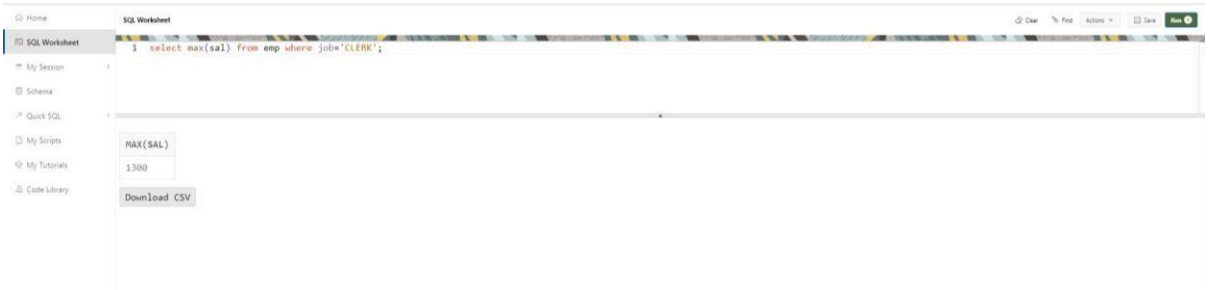
9. Display the names of employees whose names have second alphabet A in their names.

SELECT ENAME FROM EMP WHERE ENAME LIKE '_A%'



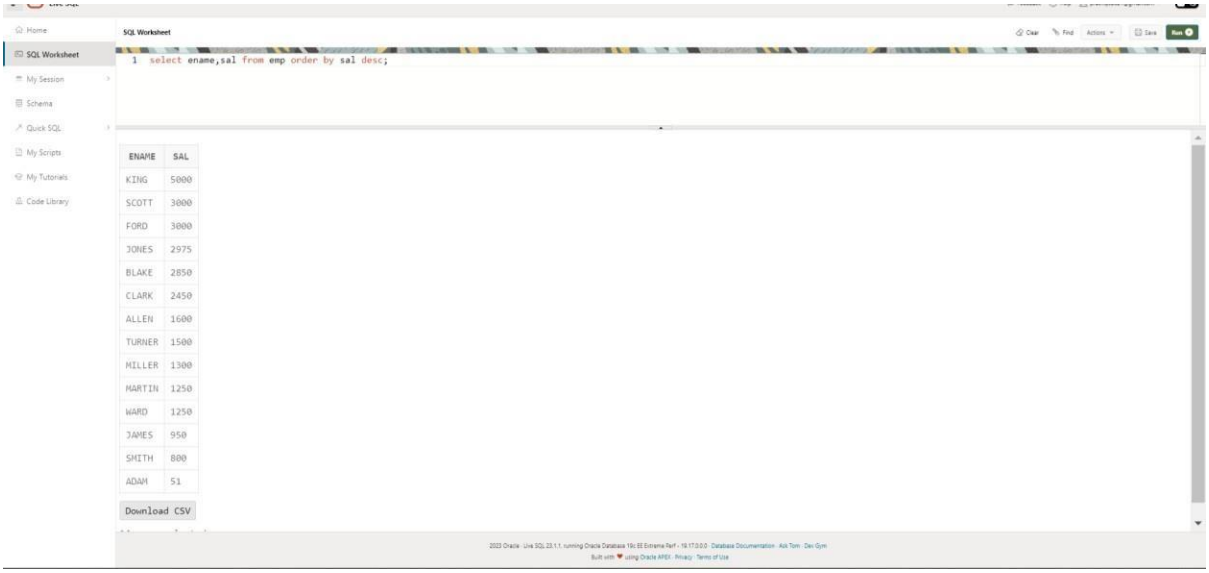
10. Display the maximum salary being paid to CLERK.

SELECT MAX(SAL) FROM EMP WHERE JOB='CLERK';



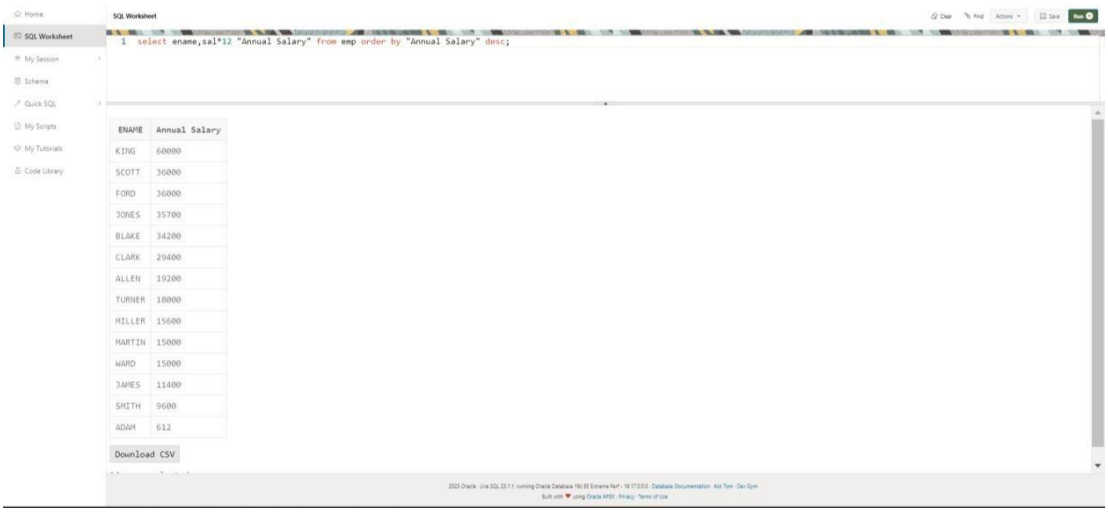
11. Display the names of the employee in descending order of salary.

SELECT ENAME FROM EMP ORDER BY SAL DESC;



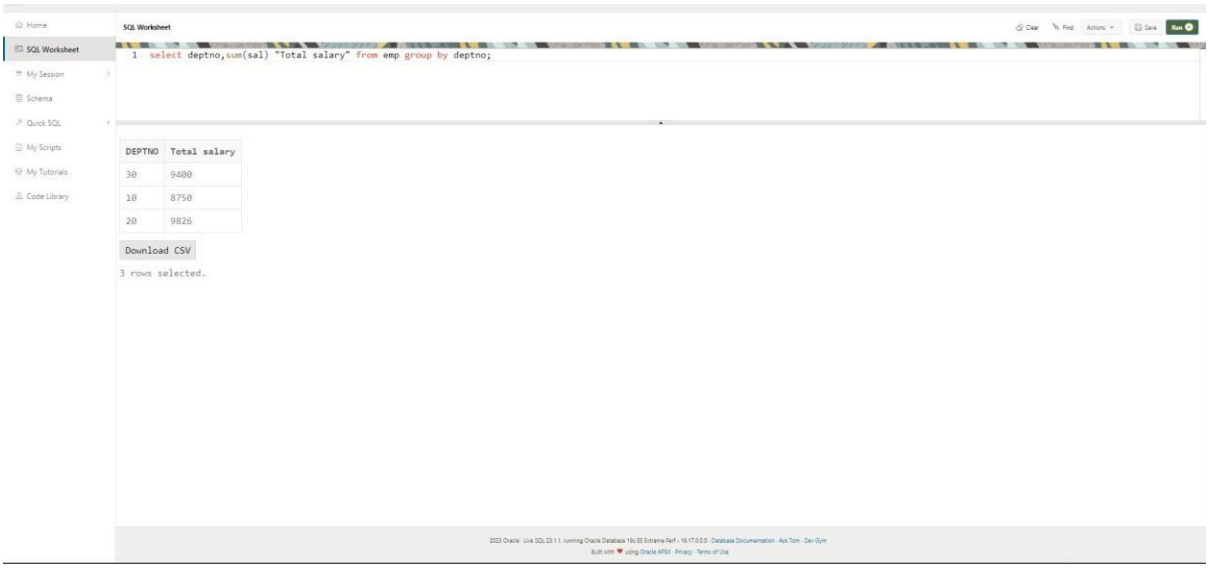
12. Display the name of the employee along with their annual salary(sal*12).The name of the employee earninghighest annual salary should apper first.

SELECT ENAME, SAL*12 AS ANNUAL_SALARY FROM EMP ORDER BY ANNUAL_SALARY DESC;



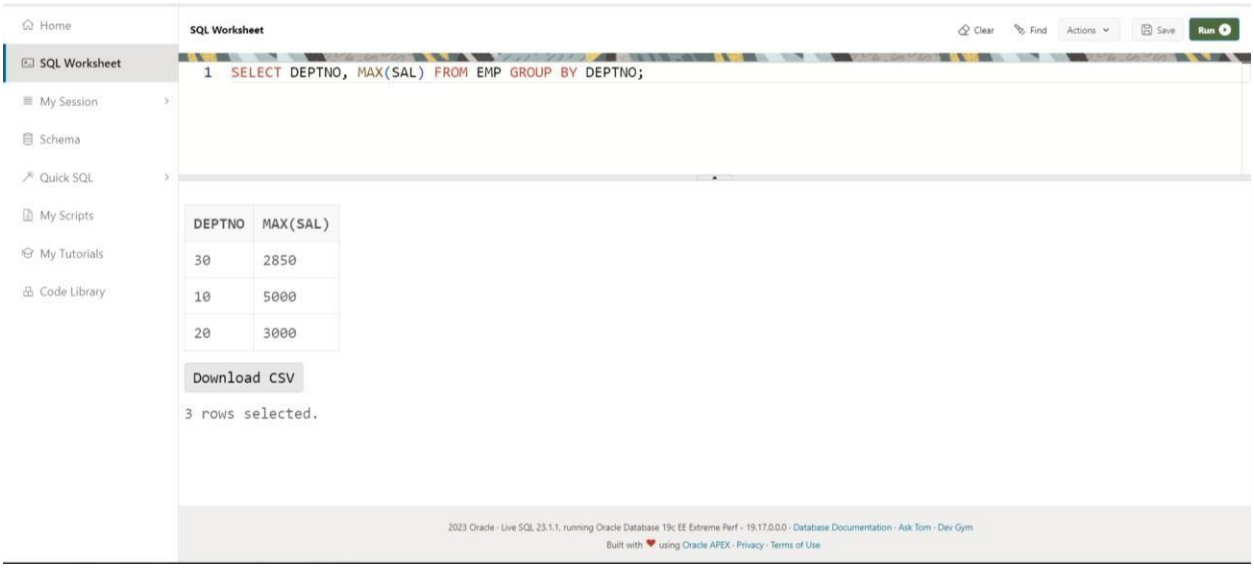
13. Display the depart numbers and total salary for each department.

SELECT DEPTNO, SUM(SAL) FROM EMP GROUP BY DEPTNO;



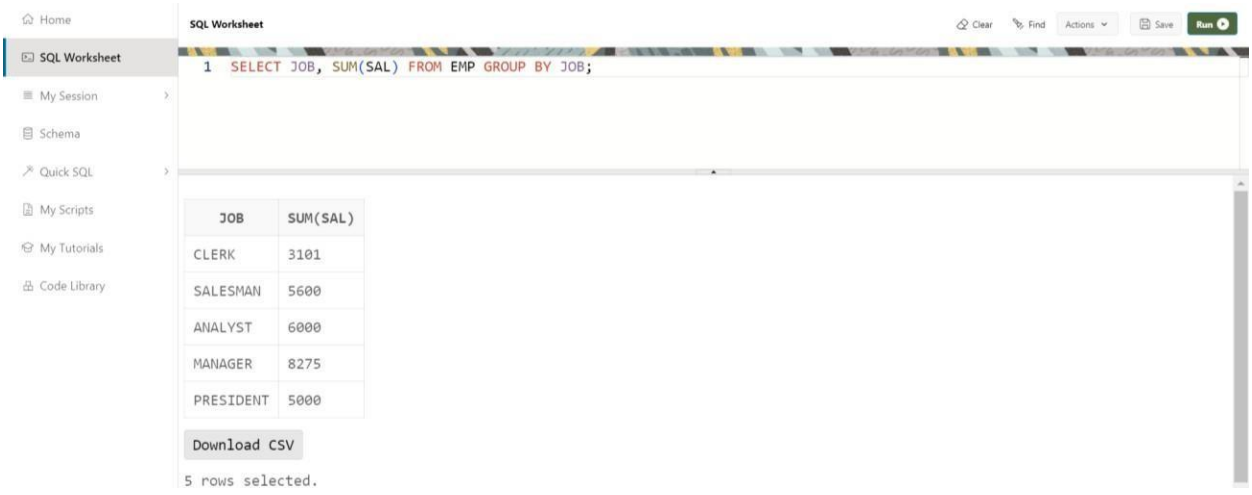
14. Display the depart numbers and max salary for each department.

SELECT DEPTNO, MAX(SAL) FROM EMP GROUP BY DEPTN0



15. Display the various jobs and total salary for each job.

SELECT JOB, SUM(SAL) FROM EMP GROUP BY JOB;



16. Display the depart numbers with more than three employees in each dept.

SELECT DEPTNO FROM EMP GROUP BY DEPTNO HAVING COUNT(EMPNO) > 3;

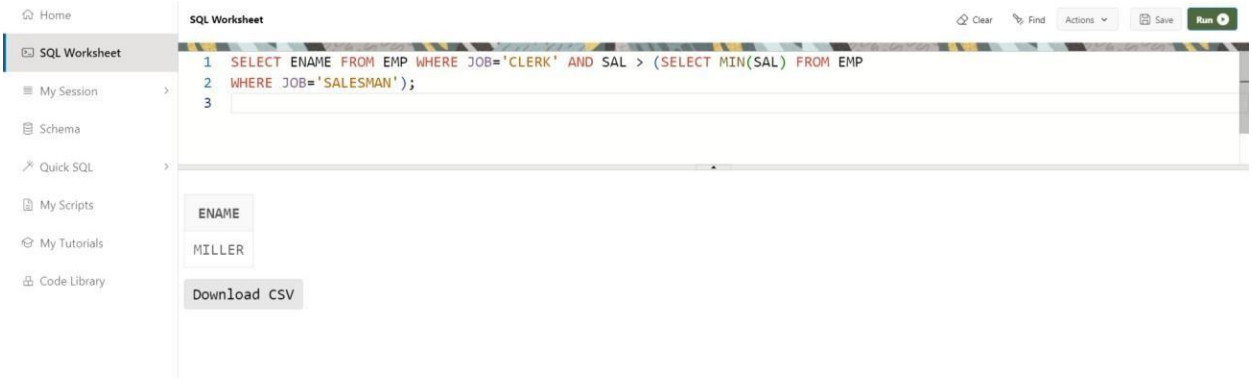
17. Display the employee number and name for employee working as clerk and earning highest salary among clerks.

SELECT EMPNO, ENAME FROM EMP WHERE JOB='CLERK' AND SAL = (SELECT MAX(SAL) FROM EMP WHERE JOB='CLERK');

18. Display the names of salesman who earns a salary more than the highest salary of any clerk.

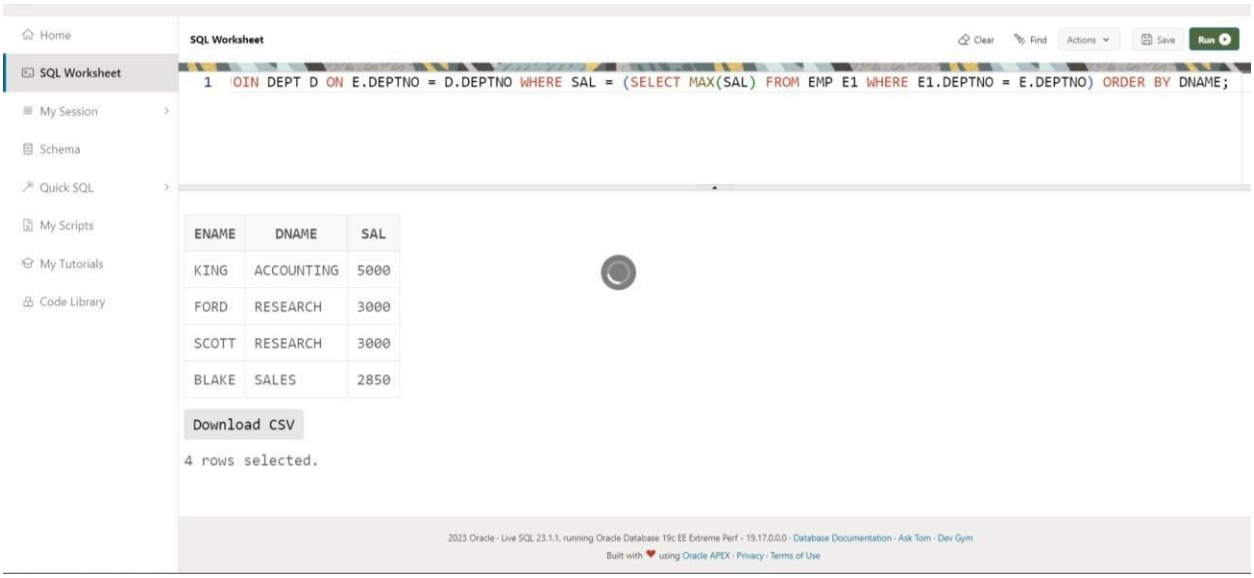
SELECT ENAME FROM EMP WHERE JOB='SALESMAN' AND SAL > (select max(sal) from emp where job = 'CLERK');

19. Display the names of clerks who earn a salary more than the lowest salary of any salesman.**SELECT ENAME FROM EMP WHERE JOB='CLERK' AND SAL > (SELECT MIN(SAL) FROM EMP WHERE JOB='SALESMAN');**



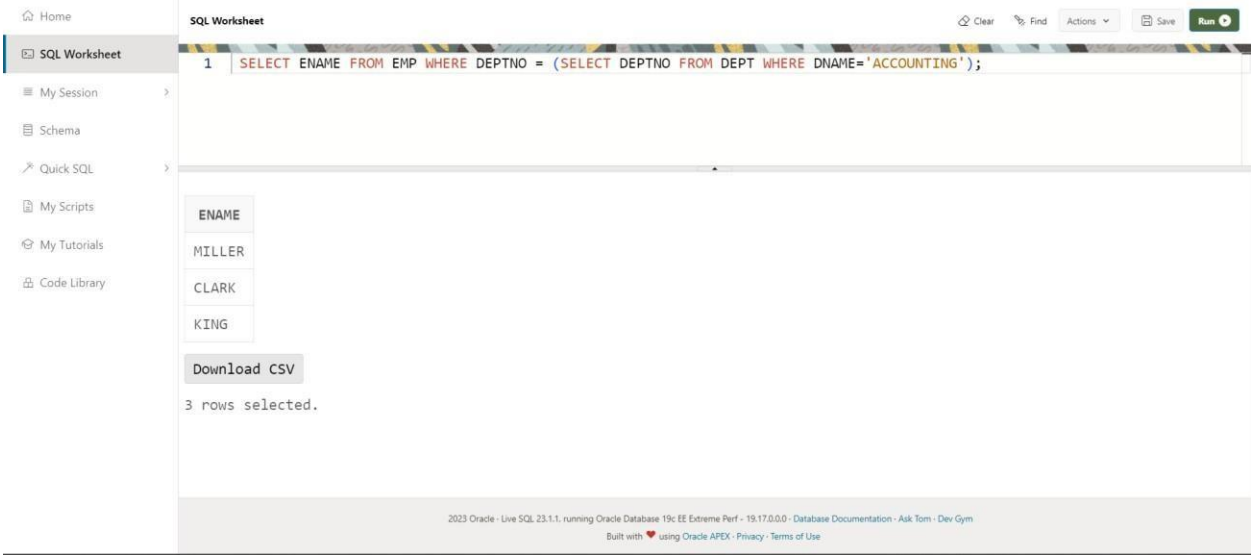
20. Display the names of the employees who earn highest salary in their respective departments.

SELECT ENAME, DNAME, SAL FROM EMP E JOIN DEPT D ON E.DEPTNO = D.DEPTNO WHERE SAL = (SELECT MAX(SAL) FROM EMP E1 WHERE E1.DEPTNO = E.DEPTNO) ORDER BY DNAME;



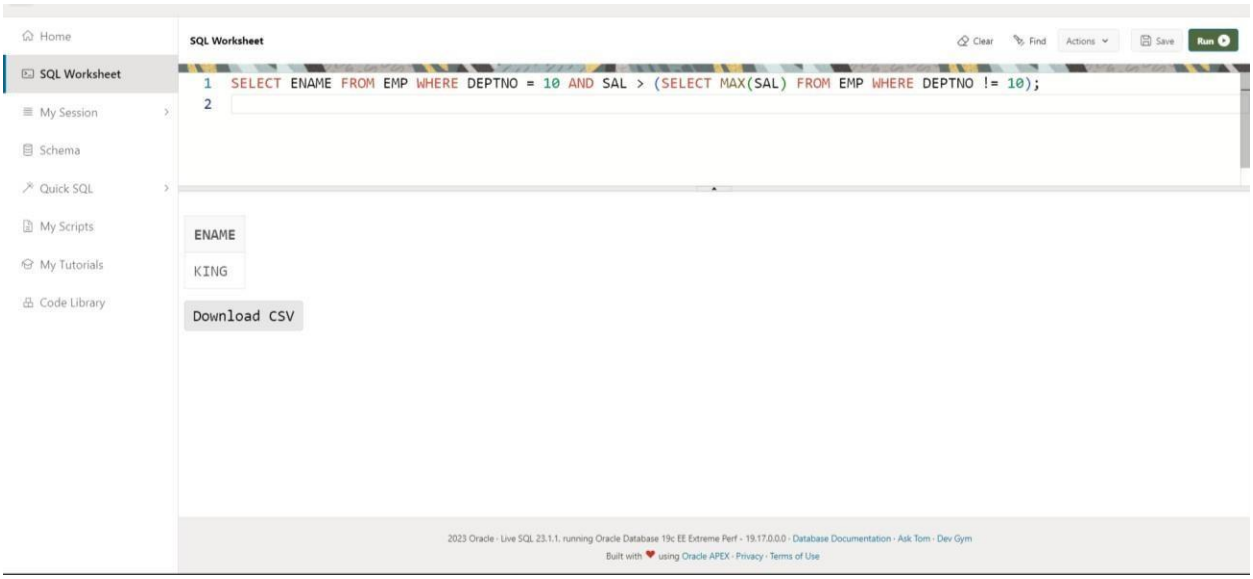
21. Display the employee names who are working in accounting department.

SELECT ENAME FROM EMP WHERE DEPTNO = (SELECT DEPTNO FROM DEPT WHERE DNAME='ACCOUNTING');



22. Display the names of employees from department number 10 with salary greater than that of any employeeworking in other department.

SELECT ENAME FROM EMP WHERE DEPTNO = 10 AND SAL > (SELECT MAX(SAL) FROM EMP WHERE DEPTNO != 10);

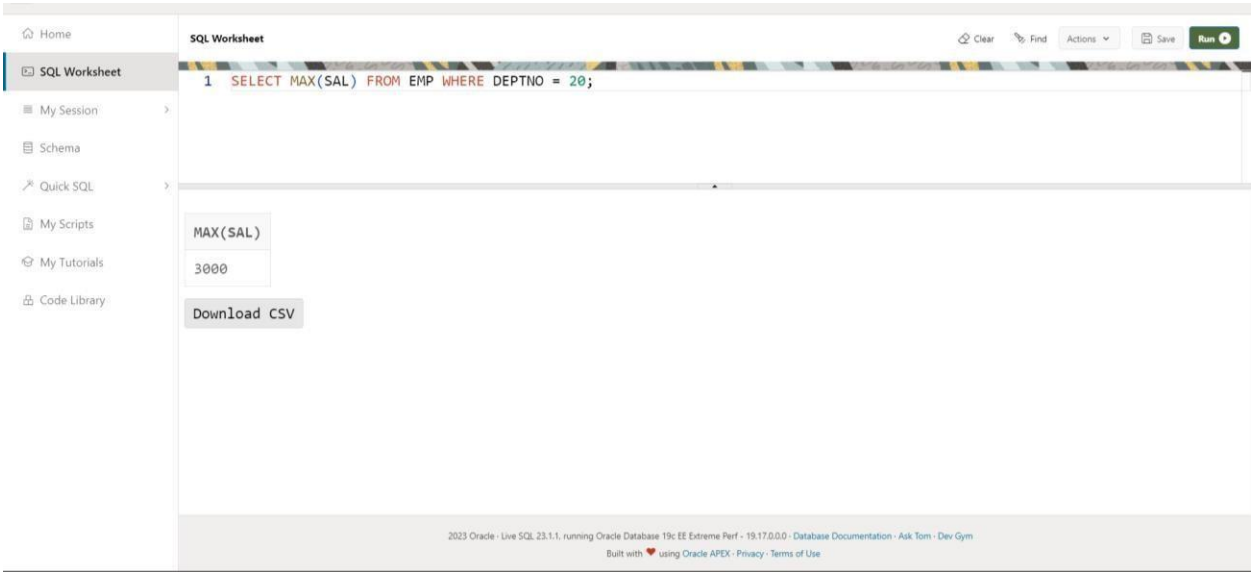


23. Display the names of the employees from department number 10 with salary greater than that of all employeeworking in other departments.

SELECT ENAME FROM EMP WHERE DEPTNO = 10 AND SAL > (SELECT MAX(SAL) FROM EMP WHERE DEPTNO != 10 GROUP BY DEPTNO);

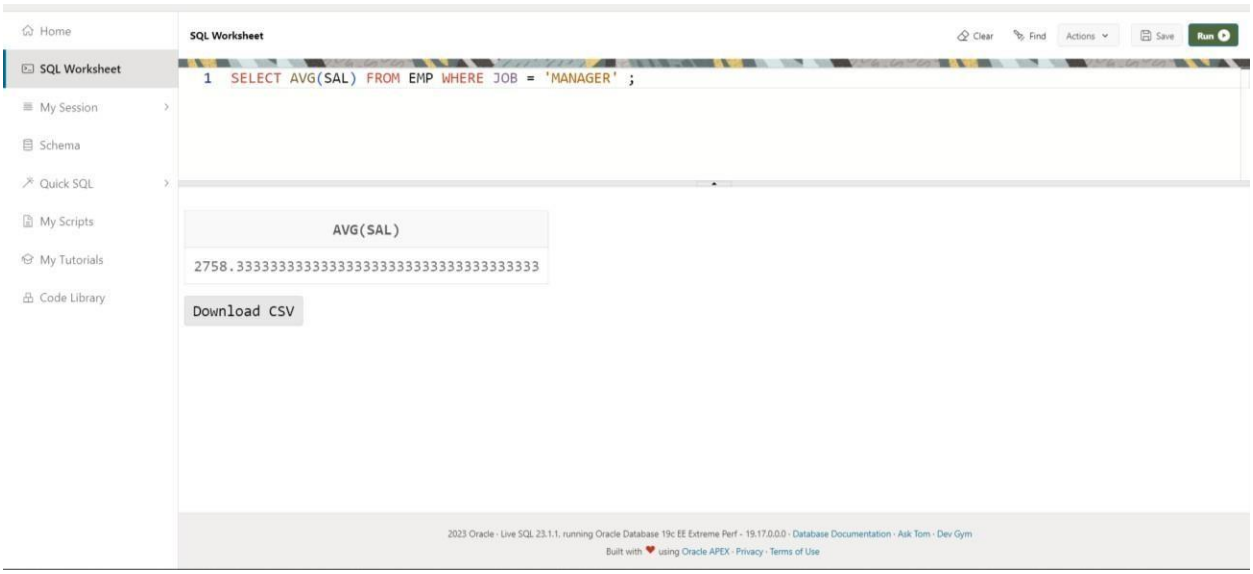
24. Display the maximum salary being paid to depart number 20.

SELECT MAX(SAL) FROM EMP WHERE DEPTNO = 20;



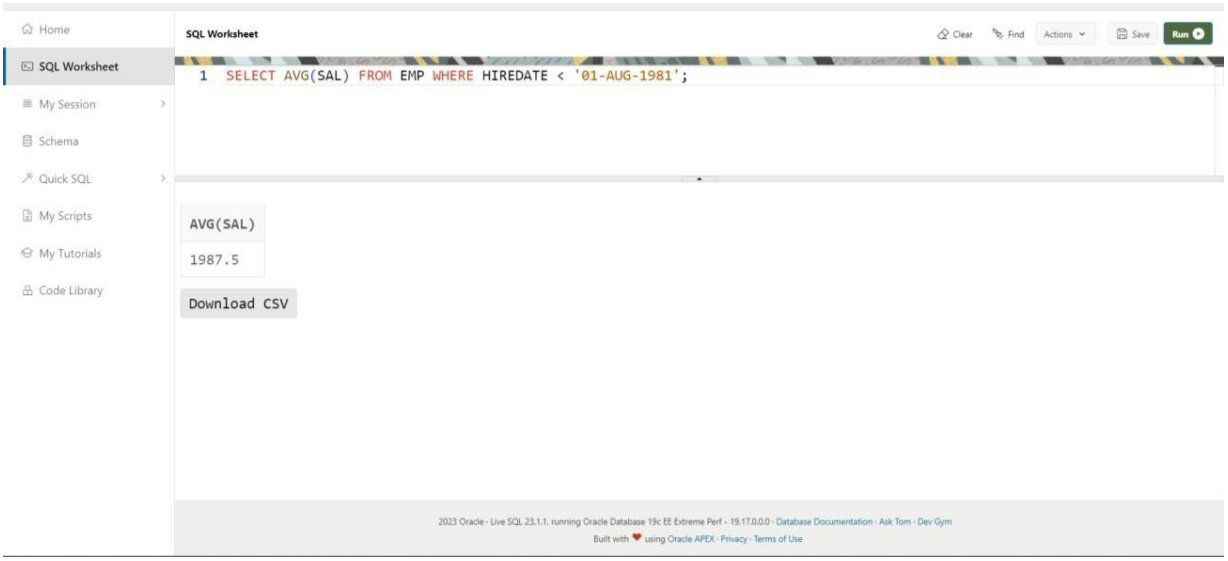
25. Display the average salary drawn by MANAGERS.

SELECT AVG(SAL) FROM EMP WHERE JOB = 'MANAGER' ;



26. Select Avg(Sal) from emp where Joj< {01/08/81};

SELECT AVG(SAL) FROM EMP WHERE HIREDATE < '01-AUG-1981';



Q.4 Given the following tables for a database LIBRARY:

Book_ID	Book_Name	Author_Nme	Publishers	Price	Type	Qty
C0001	Fast Cook	Lata Kapoor	EPB	355	Cookery	5
F0001	The Tears	William Hopkins	First Publ.	650	Fiction	20
T0001	My first c++	Brian & Brooke	EPB	350	Text	10
T0002	C++ Brainworks	A.W. Rossaine	TDH	350	Text	15
F0002	Thunderbolts	Anna Roberts	First publ.	750	Fiction	50

Book_id	Quantity_Issued
T0001	4
C0001	5
F0001	2

```
create table BOOKS(
   Book_ID varchar2(5) primary key,
   Book_Name varchar2(20),
   Author_Name varchar2(20),
   Publishers varchar2(20),
   Price number(3),
   Type varchar2(10),
   Qty number(2));

create table ISSUED(
   Book_ID varchar2(5) primary key references BOOKS(Book_ID),
   Quantity_Issued number(2));

insert into BOOKS values('C0001', 'Fast Cook', 'Lata Kapoor', 'EPB', 355, 'Cookery', 5);
insert into BOOKS values('F0001', 'The Tears', 'William Hopkins', 'First Publ.', 650, 'Fiction', 20);
insert into BOOKS values('T0001', 'My first c++', 'Brian & Brooke', 'EPB', 350, 'Text', 10);
insert into BOOKS values('T0002', 'C++ Brainworks', 'A.W. Rossaine', 'TDH', 350, 'Text', 15);
insert into BOOKS values('F0002', 'Thunderbolts', 'Anna Roberts', 'First Publ.', 750, 'Fiction', 50);

insert into ISSUED values('T0001', 4);
insert into ISSUED values('C0001', 5);
insert into ISSUED values('F0001', 2);
```

Write SQL statements for:-

1. **To show book name, author name and price of book of First Publ. Publishers.**

```
select Book_Name, Author_Name, Price from BOOKS where Publishers='First Publ.';
```

BOOK_NAME	AUTHOR_NAME	PRICE
The Tears	William Hopkins	650
Thunderbolts	Anna Roberts	750

2. To list the names from books of text type.

select Book_Name from BOOKS where Type='Text';

BOOK_NAME
My first c++
C++ Brainworks

3. To display the names and price from books in ascending order of their price.

select Book_Name, Price from BOOKS order by Price;

BOOK_NAME	PRICE
My first c++	350
C++ Brainworks	350
Fast Cook	355
The Tears	650
Thunderbolts	750

4. **To increase the price of all books of EPB publishers by 50.**

update BOOKS set Price=Price+50 where Publishers='EPB';

5. **To display the Book_Id, Book_Name and Quantity_Issued for all books which have been issued.**

select i.Book_ID, Book_Name, Quantity_Issued from BOOKS b, ISSUED i where b.Book_ID=i.Book_ID;

BOOK_ID	BOOK_NAME	QUANTITY_ISSUED
T0001	My first c++	4
C0001	Fast Cook	5
F0001	The Tears	2

6. **To insert a new row in the table issued during the following data: "F0003",1.**

insert into ISSUED values('F0003', 1);

```
ORA-02291: integrity constraint (SQL_IKUALSHUOYQEUDJAVCWWSTYSI.SYS_C00112416164) violated - parent key not found ORA-06512: at "SYS.DBMS_SQL", line 1721
```

7. **Give the output for the following SQL queries:**
   1. **select count(*) from BOOKS.**

COUNT(*)
5

   2. **select max(Price) from BOOKS where Qty>=15.**

MAX(PRICE)
750

   3. **select Book_Name, Author_Name from BOOKS where Publishers='EPB'.**

BOOK_NAME	AUTHOR_NAME
Fast Cook	Lata Kapoor
My first c++	Brian & Brooke

   4. **select count (Distinct Publishers) from books where price > = 400;**

COUNT(DISTINCTPUBLISHERS)
2

Q.5.With references to following relations PERSONAL and JOB answer the questions that follow: Createfollowing tables such that empno and sno are not null and unique, date of birth is after '12-Jan-1960', name is never blank, area and Native place is valid, hobby, dept is not empty, salary is between 4000 and 10000.

Empno	Name	DoBirth	Native_Place	Hobby
123	Amit	23-jan-1965	delhi	music
127	Manoj	12-dec-1976	mumbai	writing
124	Abhai	11-aug-1975	allahabad	music
125	Vinod	04-apr-1977	delhi	Sports
128	Abhay	10-mar-1974	mumbai	grdening
129	ramesh	28-ovt-1981	pune	sports

Sno	Area	App_date	Salary	Retd_date	Dept
123	Agra	25-jan-2006	5000	25-jan-2026	Marketing
127	Mathura	22-dec-2006	6000	22-dec-202	Finance
124	Agra	19-aug-2007	5500	19-aug-202	Marketing
125	Delhi	14-apr-2004	8500	14-apr-2018	Sales
128	pune	13-mar-2008	7500	13-mar-2028	Sales

```
CREATE TABLE PERSONAL (
Empno number(5),
EName VARCHAR(30) NOT NULL,
 DoBirth DATE NOT NULL CHECK (DoBirth > '12-Jan-1960'),
 Native_Place VARCHAR(255) NOT NULL,
 Hobby VARCHAR(255) NOT NULL,
constraint p_pk primary key (Empno)
```

);

INSERT INTO PERSONAL VALUES (123, 'Amit', '23-Jan-1965', 'Delhi', 'music');

INSERT INTO PERSONAL VALUES (127, 'Manoj', '12-dec-1976', 'Mumbai', 'writing');

INSERT INTO PERSONAL VALUES (124, 'Abhai', '11-aug-1975', 'Allahabad', 'music');

INSERT INTO PERSONAL VALUES (125, 'Vinod', '04-apr-1977', 'Delhi', 'Sports');

INSERT INTO PERSONAL VALUES (128, 'Abhay', '10-mar-1974', 'Mumbai', 'gardening');

INSERT INTO PERSONAL VALUES (129, 'Ramesh', '28-oct-1981', 'Pune', 'sports');

CREATE TABLE JOB (

Sno number(5) references PERSONAL(Empno),

Area VARCHAR(30) NOT NULL,

App_date DATE NOT NULL,

Salary number(6,2) NOT NULL CHECK (Salary BETWEEN 4000 AND 10000),

Retd_date DATE NOT NULL,

Dept VARCHAR(30) NOT NULL,

constraint j_pk primary key (Sno)

);

INSERT INTO JOB VALUES (123, 'Agra', '25-jan-2006', 5000, '25-jan-2026', 'Marketing');

INSERT INTO JOB VALUES (127, 'Mathura', '22-dec-2006', 6000, '22-dec-202', 'Finance');

INSERT INTO JOB VALUES (124, 'Agra', '19-aug-2007', 5500, '19-aug-202', 'Marketing');

INSERT INTO JOB VALUES (125, 'Delhi', '14-apr-2004', 8500, '14-apr-2018', 'Sales');

INSERT INTO JOB VALUES (128, 'Pune', '13-mar-2008', 7500, '13-mar-2028', 'Sales');

Queries:

1. Show empno, name and salary of those who have sports as hobby:

SELECT Empno, EName, Salary FROM PERSONAL JOIN JOB ON PERSONAL.Empno = JOB.Sno WHERE Hobby = 'Sports';

```
47  SELECT Empno, EName, Salary FROM PERSONAL JOIN JOB ON PERSONAL.Empno = JOB.Sno WHERE Hobby = 'Sports';
48
```

EMPNO	ENAME	SALARY
125	Vinod	8500

2. Show name of the eldest employee:

SELECT EName FROM PERSONAL where rownum = 1 ORDER BY DoBirth ASC ;

```
48  SELECT EName FROM PERSONAL where rownum = 1 ORDER BY DoBirth ASC ;
```

ENAME
Amit

3. Show number of employee area wise:

SELECT Area, COUNT(Sno) as "Number of Employees" FROM JOB GROUP BY Area;

```
49  SELECT Area, COUNT(Sno) as "Number of Employees" FROM JOB GROUP BY Area;
```

AREA	Number of Employees
Pune	1
Delhi	1
Mathura	1
Agra	2

4. Show youngest employees from each native place:

select max(DoBirth) "Date_Of_Birth" , Native_Place from personal group by Native_Place;

```
50  select max(DoBirth) "Date_Of_Birth" , Native_Place from personal group by Native_Plac
```

Date_Of_Birth	NATIVE_PLACE
28-OCT-81	Pune
04-APR-77	Delhi
12-DEC-76	Mumbai
11-AUG-75	Allahabad

5. Show sno, name, hobby and salary in descending order of salary

SELECT Sno, EName, Hobby, Salary FROM JOB JOIN PERSONAL ON JOB.Sno = PERSONAL.Empno ORDER BY Salary DESC;

```
51  SELECT Sno, EName, Hobby, Salary FROM JOB JOIN PERSONAL ON JOB.Sno = PERSONAL.Empno ORDER BY Salary DESC;
52
```

SNO	ENAME	HOBBY	SALARY
125	Vinod	Sports	8500
128	Abhay	gardening	7500
127	Manoj	writing	6000
124	Abhai	music	5500
123	Amit	music	5000

6. Show the hobbies of those whose name pronounces as ‘Abhay’:

SELECT Hobby FROM PERSONAL WHERE EName LIKE 'Abhay';

```
52  SELECT Hobby FROM PERSONAL WHERE EName LIKE 'Abhay';
53
```

HOBBY
gardening

7. Show the appointment date and native place of those whose name starts with ‘A’ or ends in ‘d’: **SELECT App_date, Native_Place FROM JOB JOIN PERSONAL ON JOB.Sno = PERSONAL.Empno WHEREEName LIKE 'A%' OR EName LIKE '%d';**

```
53  SELECT App_date, Native_Place FROM JOB JOIN PERSONAL ON JOB.Sno = PERSONAL.Empno WHERE EName LIKE 'A%' OR EName LIKE '%d';
```

APP_DATE	NATIVE_PLACE
25-JAN-06	Delhi
19-AUG-07	Allahabad
14-APR-04	Delhi
13-MAR-08	Mumbai

8. Show the salary expense with suitable column heading of those who shall retire after 20-jan-2006:

SELECT SUM(Salary) AS "Salary Expense" FROM JOB WHERE Retd_date > '20-jan-2006';

```
54  SELECT SUM(Salary) AS "Salary Expense" FROM JOB WHERE Retd_date > '20-jan-2006';
```

Salary Expense
21000

9. Show names of those who earn more than all of the employees of sales department:

SELECT EName FROM JOB JOIN PERSONAL ON JOB.Sno = PERSONAL.Empno WHERE Salary > (SELECT MAX(Salary) FROM JOB WHERE Dept = 'Sales');

6.Write Pl/SQL code for

To reverse a number and print, i.e, if num is 677 then it should print 776:

```
DECLARE
num NUMBER := 677;
reversed_num NUMBER := 0;
temp NUMBER;
BEGIN
temp := num;
WHILE temp > 0 LOOP
reversed_num := reversed_num * 10 + (temp MOD 10);
temp := temp DIV 10;
END LOOP;
DBMS_OUTPUT.PUT_LINE('Reversed number: ' || reversed_num);
END;
```

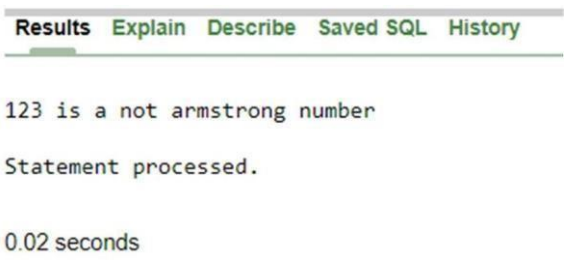
To print a Fibonacci series:

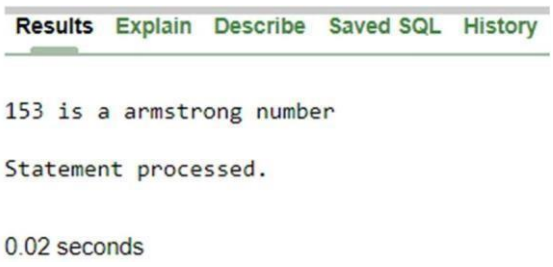
```
DECLARE
n NUMBER := 10;
 a NUMBER := 0;
 b NUMBER := 1;
 temp NUMBER;
 BEGIN
 DBMS_OUTPUT.PUT_LINE(a); DBMS_OUTPUT.PUT_LINE(b);
 FOR i IN 1..n LOOP
 temp := a + b;
 a := b;
 b := temp;
DBMS_OUTPUT.PUT_LINE(temp);
END LOOP;
END;
```


```
0
1
1
2
3
5
8
13
21
34
55
89
144
233
377

Statement processed.

0.00 seconds
```

To check a number is Armstrong or not:

```
 declare
a number:=677; b number:=0;
c number;

begin c:=a; loop
if a=0 then exit;
end if; b:=b+power(mod(a,10),3); a:=floor(a/10);
end loop; if b=c then
dbms_output.put_line('c is a armstrong number'); else
dbms_output.put_line('c is a not armstrong number'); end if;
end;
```

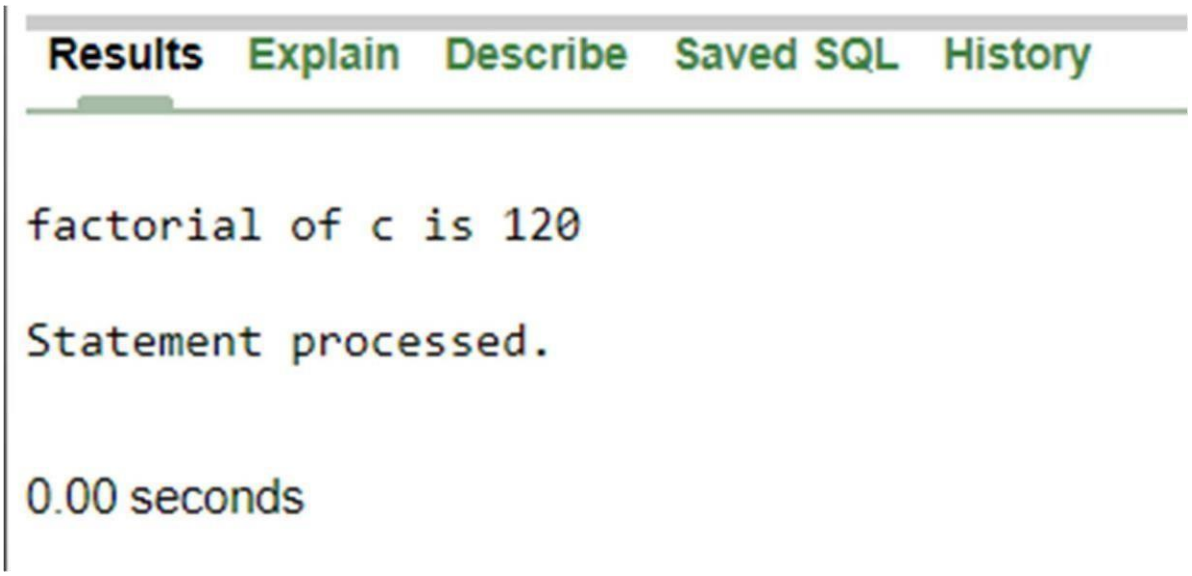




To print the factorial of a given number:

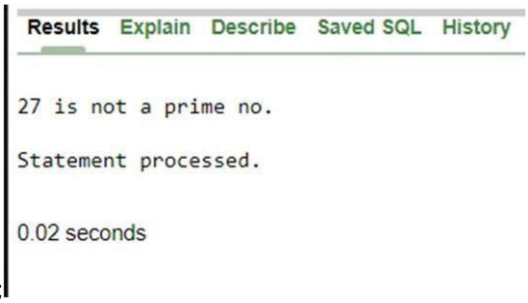
```
DECLARE
 num NUMBER := 5;
 factorial NUMBER := 1;
 BEGIN
FOR i IN 1..num LOOP
 factorial := factorial * i;
END LOOP;
DBMS_OUTPUT.PUT_LINE('Factorial of ' || num || ' is ' || factorial);
```

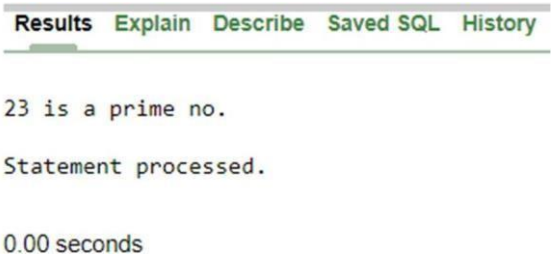
END;



To evaluate whether a given number is prime or not:

```
declare
a number:=0; b number:=2; c number:=0;

begin loop
if b>(ceil(a/2)) then exit;
else
if mod(a,b)=0 then
dbms_output.put_line(a||' is not a prime no.'); c:=1;
exit; end if; b:=b+1;
end if; end loop; if c=0 then
dbms_output.put_line(a||' is a prime no.'); end if;
```

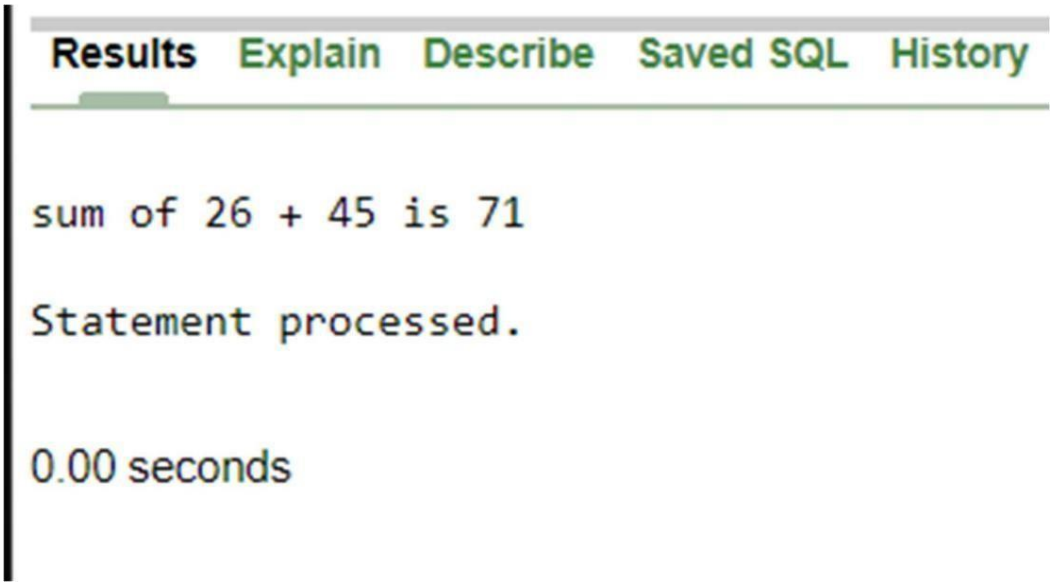




end;

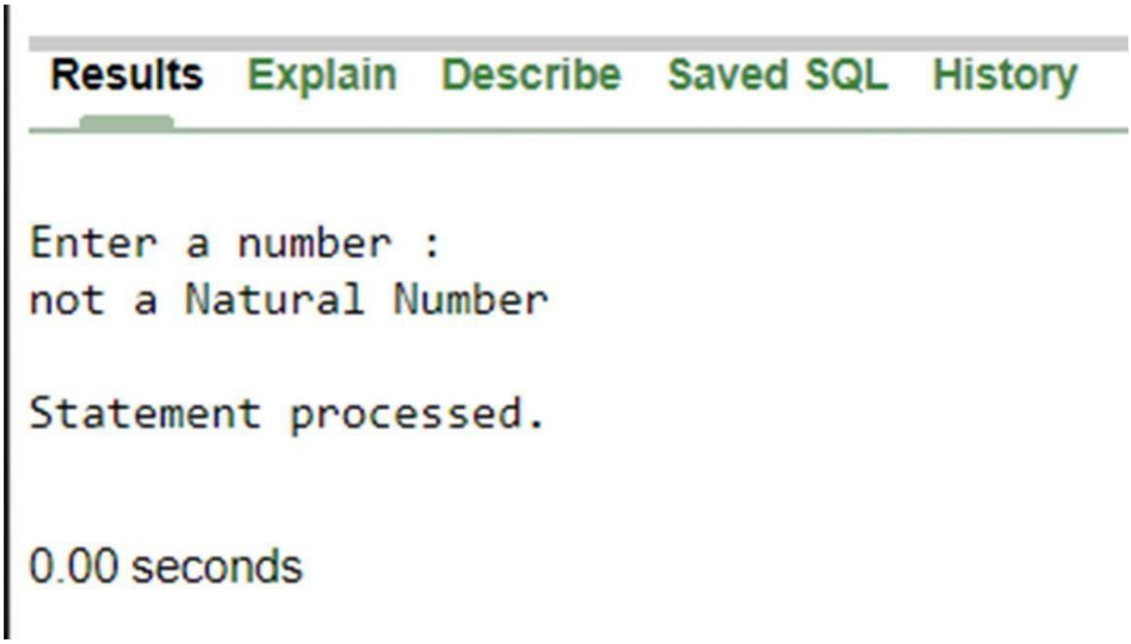
To perform the addition of two numbers

```
DECLARE
num1 NUMBER := 26;
num2 NUMBER := 45;
sum NUMBER;
BEGIN
sum := num1 + num2;
DBMS_OUTPUT.PUT_LINE('The sum of ' || num1 || ' and ' || num2 || ' is ' || sum);
END;
```



To get a number from keyboard and if it zero print “natural number”, else print “not a natural number”:

```
declare
a number;

begin
dbms_output.put_line('Enter a number : '); a:=&a;
if a=0 then dbms_output.put_line('Natural Number'); else
dbms_output.put_line('not a Natural Number'); end if;
end;
```



Top of Form To find the area and perimeter of given circle:

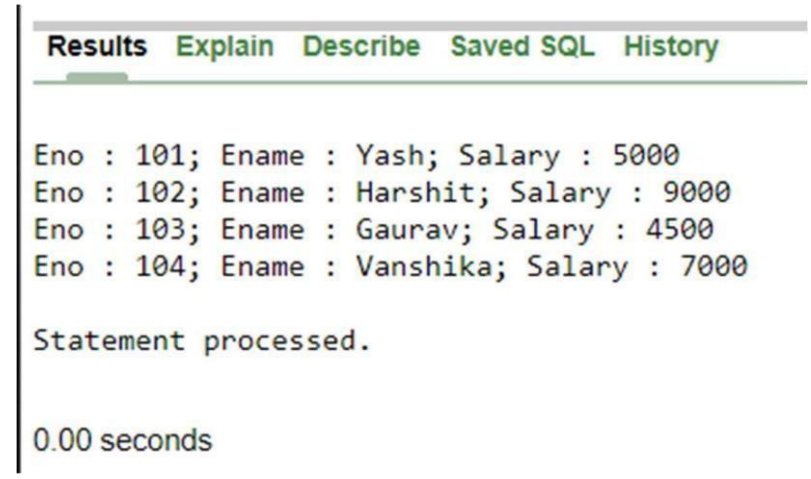
```
declare
diameter number:=3; area number; perimeter number;
begin
 area:=power(diameter,2)*3.14; perimeter:=2*diameter*3.14;
dbms_output.put_line('Area of circle is : '||area);
dbms_output.put_line('Perimeter of circle is : '||perimeter); end
```


```
Area of circle is : 28.26
Perimeter of circle is : 18.84

Statement processed.
```

0.00 seconds

To calculate the net salary if dfa is 30% of basic, hra is 10% of basic and pf is 7%. If basic salary is less than 8000, pf is 10% if basic sal between 8000 to 160000.

```
declare
eno emp.empno%type; ename emp.ename%type; sal emp.sal%type;
cursor c_emp is select empno,ename,sal from emp; begin
open c_emp;
loop fetch c_emp into eno,ename,sal; if c_emp%notfound then
exit; end if;
dbms_output.put_line('Eno : '||eno||'; Ename : '||ename||'; Salary : '||sal); end loop;
close c_emp; end
```



To select record of emp table with cursor:

```
DECLARE
CURSOR emp_cursor IS SELECT * FROM emp;
 emp_rec emp%ROWTYPE;
BEGIN
OPEN emp_cursor;
LOOP
 FETCH emp_cursor INTO emp_rec;
EXIT WHEN emp_cursor%NOTFOUND;
DBMS_OUTPUT.PUT_LINE(emp_rec.empno || ' ' || emp_rec.ename);
END LOOP;
 CLOSE emp_cursor;
END;
```

To raise an error if no data found:

```
 DECLARE CURSOR emp_cursor IS SELECT * FROM emp;
emp_rec emp%ROWTYPE; no_data_found EXCEPTION;
PRAGMA EXCEPTION_INIT(no_data_found, -1403);
BEGIN
 OPEN emp_cursor;
 FETCH emp_cursor INTO emp_rec;
 IF emp_cursor%NOTFOUND THEN RAISE no_data_found;
 ELSE DBMS_OUTPUT.PUT_LINE(emp_rec.empno || ' ' || emp_rec.ename);
 END IF;
CLOSE emp_cursor;
EXCEPTION WHEN no_data_found THEN
DBMS_OUTPUT.PUT_LINE('No data found in the table.');
END;
```

```
ORA-20009: No data found in emp Table
```

Q7.Write and explain the following Pl/SQL triggers on emp table.

10. Before UPDATE Trigger

```
create or replace trigger before_update_trigger
before update on emp
for each row
begin
dbms_output.put_line('Old Salary : '||:old.sal);
dbms_output.put_line('New Salary : '||:new.sal);
dbms_output.put_line('Before executing update statement...');
end;
```

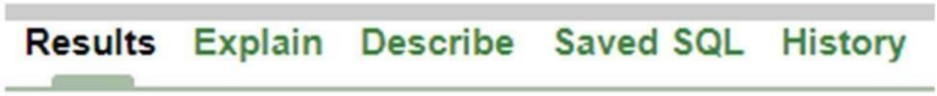

```
Trigger created.

0.25 seconds
```

update emp set sal=5500 where ename='Prachi';


```
Old Salary : 5000
New Salary : 5500
Before executing update statement...

1 row(s) updated.

0.00 seconds
```

11. Before DELETE Trigger

```
create or replace trigger before_delete_trigger
before delete on emp
for each row
begin
dbms_output.put_line('Old Salary : '||:old.sal);
dbms_output.put_line('New Salary : '||:new.sal);
dbms_output.put_line('Before executing Delete statement...');
end;
```




```
Trigger created.
```

0.25 seconds

o delete from emp where ename='Pragya';


```
Old Salary : 7000
New Salary :
Before executing Delete statement...

1 row(s) deleted.
```

0.02 seconds

12. Before INSERT Trigger

```
create or replace trigger before_insert_trigger
before insert on emp
for each row
```

```
begin
dbms_output.put_line('Old Salary : '||:old.sal);
dbms_output.put_line('New Salary : '||:new.sal);
dbms_output.put_line('Before executing insert statement...');
end;
```


```
Trigger created.
```

0.25 seconds

o insert into emp values(104,'Vanshika',7000,'18-MAY-97');

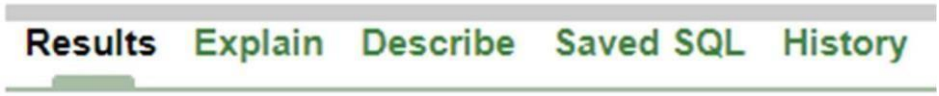

```
Old Salary :
New Salary : 7000
Before executing insert statement...

1 row(s) inserted.
```

0.00 seconds

13. After INSERT Trigger

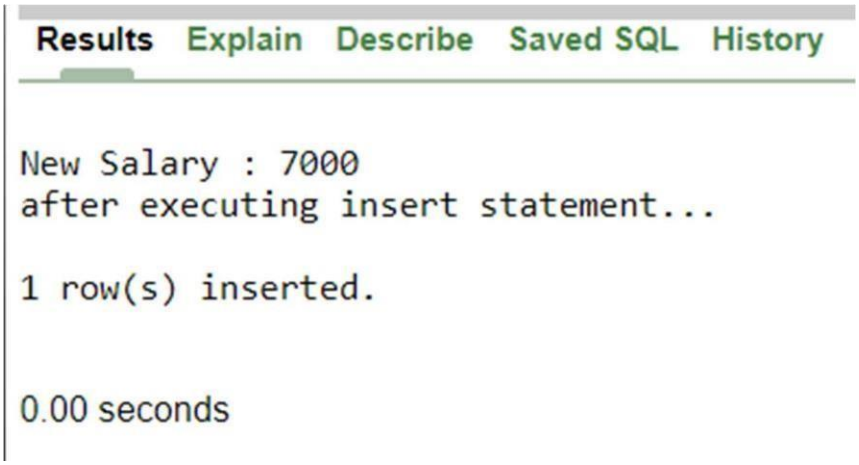
```
create or replace trigger after_insert_trigger
after insert on emp
for each row
begin
dbms_output.put_line('Old Salary : '||:old.sal);
dbms_output.put_line('New Salary : '||:new.sal);
```

```
dbms_output.put_line('after executing insert statement...');
end;
```




```
Trigger created.

0.25 seconds
```

o insert into emp values(104,'Vanshika',7000,'18-MAY-97');



14. After UPDATE Trigger

```
create or replace trigger after_update_trigger
after update on emp
for each row
begin
dbms_output.put_line('New Salary : '||:new.sal);
dbms_output.put_line('after executing update statement...');
end;
```


```
Trigger created.

0.25 seconds
```

o update emp set sal=7500 where ename='Vanshika';

15. After DELETE Trigger

```
create or replace trigger after_delete_trigger
after delete on emp
for each row
begin
dbms_output.put_line('New Salary : '||:new.sal);
dbms_output.put_line('after executing delete statement...');
end;
```


```
Trigger created.
```

0.25 seconds

o delete from emp where ename='Vanshika'; and "sal" columns along with the current date and time into the "emp_audit" table.


```
New Salary :
after executing delete statement...

1 row(s) deleted.
```

0.00 seconds