

CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY
DEVANG PATEL INSTITUTE OF ADVANCE TECHNOLOGY & RESEARCH
Department of Computer Engineering

Subject Code and Name: CE246 Database Management System

Semester: 4th

Academic year: 2022-23

Practical - 1

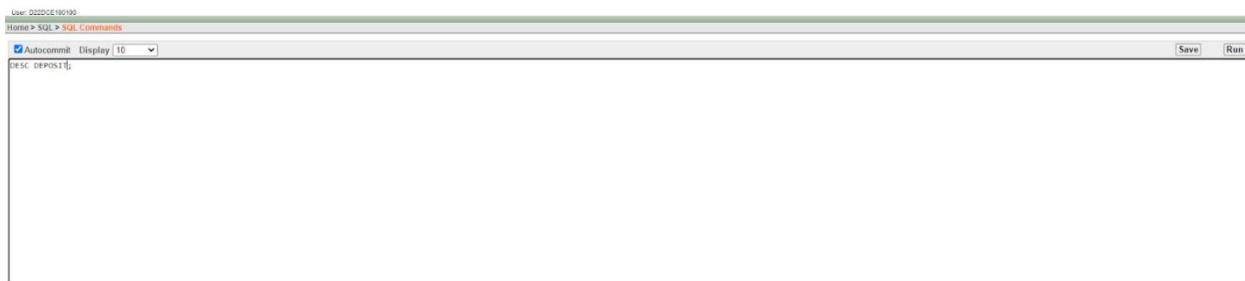
Aim: To Demonstrate DDL-create and DML-insert commands.

Implementation:

(1) Describe deposit, branch.

Ans:

Describe deposit



The screenshot shows a database interface with the following details:

- User: D22DCE190190
- Home > SQL > SQL Commands
- Autocommit: On
- Display: 10
- SQL Command: DESC DEPOSIT;
- Buttons: Save, Run

Below the command window, there is a results table for the 'DEPOSIT' table:

Object Type	Table	Column	Date Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DEPOSIT	ACNO	VARCHAR2	5	-	-	-	✓	-	-	-
	CNAME	VARCHAR2	10	-	-	-	✓	-	-	-
	ENAME	VARCHAR2	10	-	-	-	✓	-	-	-
	AMOUNT	NUMBER	-	8	2	-	✓	-	-	-
	ADATE	DATE	7	-	-	-	✓	-	-	-

Page Number: 1-5

Describe branch

User D22DCE190190
Home > SQL > SQL Commands
Autocommit: Display / 10 ✓ Save Run
DESC BRANCH;

Results Explain Describe Saved SQL History
Object Type TABLE Object BRANCH
Table Column Data Type Length Precision Scale Primary Key Nullable Default Comment
BRANCH_BNAME Varchar2 10 - - ✓ - -
CITY Varchar2 10 - - ✓ - -
1-2

(2) Describe borrow, customers

Ans:

Describe borrow

User D22DCE190190
Home > SQL > SQL Commands
Autocommit: Display / 10 ✓ Save Run
DESC BORROW;

Results Explain Describe Saved SQL History
Object Type TABLE Object BORROW
Table Column Data Type Length Precision Scale Primary Key Nullable Default Comment
BORROW_LOANNO Varchar2 5 - - ✓ - -
CHNAME Varchar2 10 - - ✓ - -
ENAME Varchar2 10 - - ✓ - -
AMOUNT Number - 8 2 - ✓ - -
1-4

Describe Customers

User D22DCE190190
Home > SQL > SQL Commands
Autocommit: Display / 10 ✓ Save Run
DESC CUSTOMERS;

Results Explain Describe Saved SQL History
Object Type TABLE Object CUSTOMERS
Table Column Data Type Length Precision Scale Primary Key Nullable Default Comment
CUSTOMERS_CNAME Varchar2 10 - - ✓ - -
CITY Varchar2 10 - - ✓ - -
1-2

(3) List all data from table DEPOSIT.**Ans:**

User: D22DCE190190
Home > SQL > SQL Commands
 Autocommit. Display 10

```
SELECT * FROM DEPOSIT;
```

ACCTNO	CNAME	BNAME	AMOUNT	ADATE
100	SANDIP	AMHEDAB	2000	31-MAR-96
101	SHIVANI	VIRAR	1000	05-SEP-95
102	KRANTI	NEHRU PLACE	5000	03-JUL-95
103	ANIL	VIRCE	1000	01-MAR-95
104	SUNIL	AJR	5000	04-JAN-96
105	MEHUL	KAROLBAGH	3500	17-NOV-95
106	MACHHURI	CHANDI	1200	17-DEC-95
107	PRAMOD	M.G.ROAD	3000	27-MAR-96
108	MINU	POWAI	7000	10-AUG-95

9 rows returned in 0.02 seconds

(4) List all data from table BORROW.**Ans:**

User: D22DCE190190
Home > SQL > SQL Commands
 Autocommit. Display 10

```
SELECT * FROM BORROW;
```

LOANNO	CNAME	BNAME	AMOUNT
261	ANIL	MUMBAI	1000
262	MEHUL	AJR	5000
311	SUNIL	DHARAMPETH	3000
321	MACHHURI	ANDHERI	2000
375	PRAMOD	VIRUAR	8000
481	KRANTI	NEHRU PLACE	3000

6 rows returned in 0.04 seconds

(5) List all data from table CUSTOMERS.**Ans:**

User: D22DCE190190
Home > SQL > SQL Commands
 Autocommit. Display 10

```
SELECT * FROM CUSTOMERS;
```

CNAME	CITY
NAREN	BOMBAY
ANIL	CALCUTTA
SUNIL	DEHR
MEHUL	BARODA
MANDIR	PATNA
MACHHURI	NAGPUR
PRAMOD	NAGPUR
SANDIP	SURAT
SHIVANI	GOA
KRANTI	BOMBAY

10 rows returned in 0.02 seconds

(6) List all data from table BRANCH.**Ans:**

User D22DCE190190
Home > SQL > SQL Commands
 Autocommit Display 10

```
SELECT * FROM BRANCH;
```

Results Explain Describe Saved SQL History

BNAME	CITY
M.G ROAD	BANGLORE
VRODE	NAGPUR
AIRI	NAGPUR
CHANDI	DELHI
DHARAMPETH	NAGPUR
ANDHERI	BOMBAY
VIRAR	BOMBAY
NEHRU PLACE	DELHI
POWAI	BOMBAY

9 rows returned in 0.00 seconds

(7) Give account no and amount of depositors.**Ans:**

User D22DCE190190
Home > SQL > SQL Commands
 Autocommit Display 10

```
SELECT ACTNO,AMOUNT FROM DEPOSIT;
```

Results Explain Describe Saved SQL History

ACTNO	AMOUNT
106	2000
107	1000
108	5000
109	1000
101	5000
102	3000
104	1200
105	3000
109	7000

9 rows returned in 0.00 seconds

(8) Give name of depositors having amount greater than 4000.**Ans:**

User D22DCE190190
Home > SQL > SQL Commands
 Autocommit Display 10

```
SELECT CNAME FROM DEPOSIT WHERE AMOUNT>4000;
```

Results Explain Describe Saved SQL History

CNAME
KRANTI
SUNIL
MINU

3 rows returned in 0.00 seconds

(9) Give name of customers who opened account after date '1-12-96'.

Ans:

```
User: D22DCE190190
Home > SQL > SQL Commands
 Autocommit: Display | 10 | ▾
SELECT CNAME FROM DEPOSIT WHERE ADATE > '1-12-96';
Save Run
```

ORA-01843: not a valid month!

(10) Give name of city where branch karolbagh is located.

Ans:

```
User: D22DCE190190
Home > SQL > SQL Commands
 Autocommit: Display | 10 | ▾
SELECT CITY FROM BRANCH WHERE BNAME='KAROLBAGH';
Save Run
```

CITY
DELHI

1 rows returned In 0.00 seconds CSV Export

(11) Give account no and amount of customer having account opened between date 1-12-96 and 1-6-96.

Ans:

```
User: D22DCE190190
Home > SQL > SQL Commands
 Autocommit: Display | 10 | ▾
SELECT ACTNO,AMOUNT FROM DEPOSIT WHERE ADATE BETWEEN '1-12-96' AND '1-6-96';
Save Run
```

ORA-01843: not a valid month!

(12) Give names of depositors having account at VRCE.

Ans:

The screenshot shows a MySQL Workbench interface. The SQL tab contains the following query:

```
SELECT CNAME FROM DEPOSIT WHERE SHARE='VRCE';
```

The results pane shows a single row:

CNAME
ANIL

Below the results, it says "1 rows returned in 0.00 seconds" and has a "CSV Export" button.

Conclusion: In this practical I learned how to create a table with different data types and insert data in it and also how to access various data from the table with some conditions.

Remarks: _____

Marks: _____

Signature: _____

Practical – 2

Aim: To Demonstrate DDL-create and DML-insert commands.

Implementation:

(1) Retrieve all data from employee, jobs and deposit.

Ans:

Employee:

User: D22DCE190
Home > SQL > SQL Commands
 Autocommit Display 10

```
select * from Employee
```

Results Explain Describe Saved SQL History

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE
101	Smith	800	-	20	shah	machine learning	fi_mgr	toronto	105	09-AUG-96
102	Snehal	1600	300	25	gupta	data science	lec	las vegas	-	14-MAR-96
103	Adama	1100	0	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95
104	Aman	3000	-	15	sharma	virtual reality	comp_op	mexico	12	02-OCT-97
105	Anita	5000	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-98
106	Sneha	2450	24500	10	joseph	big data analytics	fi_acc	melbourne	105	26-SEP-97
107	Anamika	2975	-	30	jha	artificial intelligence	it_prog	new york	-	15-JUL-97

7 rows returned in 0.00 seconds [CSV Export](#)

Job:

User: D22DCE190
Home > SQL > SQL Commands
 Autocommit Display 10

```
select * from job
```

Results Explain Describe Saved SQL History

JOB_ID	JOB_TITLE	MIN_SAL	MAX_SAL
it_prog	Programmer	4000	10000
mk_mgr	Marketing manager	9000	15000
fi_mgr	Finance manager	8200	12000
fi_acc	Account	4200	9000
lec	Lecturer	6000	17000
comp_op	Computer Operator	1500	3000

6 rows returned in 0.01 seconds [CSV Export](#)

Deposit:

User: D22DCE190
Home > SQL > SQL Commands
Autocommit Display 10

```
select * from deposit
```

Results Explain Describe Saved SQL History

ACTNO	CNAME	BNAME	AMOUNT	ADATE
100	ANIL	VRCE	1000	01-MAR-95
101	SUNIL	AJNI	5000	04-JAN-96
102	MEHUL	KAROLBAGH	3500	17-NOV-95
104	MADHURI	CHANDI	1200	17-DEC-95
105	PRMOD	M.G.ROAD	3000	27-MAR-96
106	SANDIP	ANDHERI	2000	31-MAR-96
107	SHIVANI	VIRAR	1000	05-SEP-95
108	KRANTI	NEHRU PLACE	5000	02-JUL-95
109	MINU	POWAI	7000	10-AUG-95

9 rows returned in 0.00 seconds [CSV Export](#)

(2) Give details of account no. and deposited rupees of customers having account opened between dates 01-01-06 and 25-07-06.

Ans:

User: D22DCE190
Home > SQL > SQL Commands
Autocommit Display 10

```
select A_no,Amount from deposit1 where a_date between '01-JAN-06' and '25-JUL-06';
```

Results Explain Describe Saved SQL History

A_NO	AMOUNT
101	7000
102	5000
103	6500

3 rows returned in 0.02 seconds [CSV Export](#)

(3) Display all jobs with minimum salary is greater than 4000.

Ans:

User: D22DCE190
 Home > SQL > SQL Commands

Autocommit Display 10

```
select job_title from JOB where min_sal > 4000;
```

Results Explain Describe Saved SQL History

JOB_TITLE
Marketing manager
Finance manager
Account
Lecturer

4 rows returned in 0.00 seconds [CSV Export](#)

(4) Display name and salary of employee whose department no is 20. Give alias name to name of employee.

Ans:

User: D22DCE190
 Home > SQL > SQL Commands

Autocommit Display 10

```
select emp_name "name" , emp_sal from Employee where dept_no =20;
```

Results Explain Describe Saved SQL History

Name	EMP_SAL
Smith	800
Adama	1100

2 rows returned in 0.00 seconds [CSV Export](#)

(5) Display employee no, name and department details of those employee whose department lies in (10,20).

Ans:

User: D22DCE190
 Home > SQL > SQL Commands

Autocommit Display 10

```
select emp_name,emp_no,dept_name from Employee where dept_no between 10 and 20;
```

Results Explain Describe Saved SQL History

EMP_NAME	EMP_NO	DEPT_NAME
Smith	101	machine learning
Adama	103	machine learning
Aman	104	virtual reality
Anita	105	big data analytics
Sneha	106	big data analytics

5 rows returned in 0.00 seconds [CSV Export](#)

(6) Display the non-null values of employees.**Ans:**

User: D22DCE190
Home > SQL > SQL Commands

Autocommit

```
select * from Employee where manager_id IS NOT NULL AND EMP_COMM IS NOT NULL;
```

Results Explain Describe Saved SQL History

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE
103	Adama	1100	0	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95
105	Anita	5000	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-98
106	Sneha	2450	24500	10	joseph	big data analytics	fi_acc	melbourne	105	26-SEP-97

3 rows returned in 0.00 seconds

(7) Display name of customer along with its account no (both columns should be displayed as one) whose amount is not equal to 8000 Rs.**Ans:**

User: D22DCE190
Home > SQL > SQL Commands

Autocommit

```
SELECT A_NO,CNAME FROM DEPOSIT1 WHERE AMOUNT != 8000;
```

Results Explain Describe Saved SQL History

A_NO	CNAME
101	Anil
102	sunil
103	jay
105	keyur
106	mayur

5 rows returned in 0.00 seconds

(8) Display the content of job details with minimum salary either 2000 or 4000.**Ans:**

User: D22DCE190
Home > SQL > SQL Commands

Autocommit

```
SELECT JOB_TITLE FROM JOB WHERE MIN_SAL=2000 OR MIN_SAL=4000;
```

Results Explain Describe Saved SQL History

JOB_TITLE
Programmer

1 rows returned in 0.00 seconds

Like predicate:

(1) Display all employee whose name start with ‘A’ and third character is “a”.

Ans:

```
User: D22DCE190
Home > SQL > SQL Commands
SELECT emp_name FROM Employee WHERE emp_name LIKE 'A_a%';

Results Explain Describe Saved SQL History
EMP_NAME
Adama
1 rows returned in 0.00 seconds CSV Export
```

(2) Display name, number and salary of those employees whose name is 5 characters long and first three characters are ‘Ani’.

Ans:

```
User: D22DCE190
Home > SQL > SQL Commands
SELECT emp_name , emp_no,emp_sal FROM Employee WHERE emp_name LIKE 'Ani%';

Results Explain Describe Saved SQL History
EMP_NAME EMP_NO EMP_SAL
Anita 105 5000
1 rows returned in 0.00 seconds CSV Export
```

(3) Display all information of employee whose second character of name is either ‘M’ or ‘N’.

Ans:

```
User: D22DCE190
Home > SQL > SQL Commands
SELECT * FROM Employee WHERE emp_name LIKE '_m%' OR emp_name LIKE '_n%';

Results Explain Describe Saved SQL History
```

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE
101	Smith	800	-	20	shah	machine learning	f_mgr	toronto	105	09-AUG-96
105	Anita	5000	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-98

(4) Find the list of all customer name whose branch is in ‘andheri’ or ‘dadar’ or ‘Virar’.

Ans:

```
User: D22DCE190
Home > SQL > SQL Commands
 Autocommit Display 10
SELECT cname FROM deposit1 WHERE cname LIKE 'andheri' or cname LIKE 'dadar' or cname Like 'virar';

Results Explain Describe Saved SQL History
CNAME
keyur
1 rows returned in 0.00 seconds CSV Export
```

(5) Display the job name whose first three character in job id field is ‘FI_’.

Ans:

```
User: D22DCE190
Home > SQL > SQL Commands
 Autocommit Display 10
SELECT job_title FROM job WHERE job_id LIKE 'fi%';

Results Explain Describe Saved SQL History
JOB_TITLE
Finance manager
Account
2 rows returned in 0.00 seconds CSV Export
```

(6) Display the title/name of job who’s last three character are ‘_MGR’ and their maximum salary is greater than Rs 12000.

Ans:

```
User: D22DCE190
Home > SQL > SQL Commands
 Autocommit Display 10
SELECT job_title FROM JOB WHERE job_id LIKE '%_mgr' and max_sal>12000;

Results Explain Describe Saved SQL History
JOB_TITLE
Marketing manager
1 rows returned in 0.01 seconds CSV Export
```

(7) Display the non-null values of employees and also employee name second character should be 'n' and string should be 5-character long.

Ans:

```
User: D22DCE190
Home > SQL > SQL Commands
 Autocommit Display 10
SELECT * FROM Employee WHERE emp_name is NOT NULL and Manager_id is NOT NULL and emp_name LIKE '_n%' and emp_name LIKE '_____';
Save Run

Results Explain Describe Saved SQL History
EMP_NO EMP_NAME EMP_SAL EMP_COMM DEPT_NO L_NAME DEPT_NAME JOB_ID LOCATION MANAGER_ID HIREDATE
105 Anita 5000 50000 10 patel big data analytics comp_op germany 107 01-JAN-98
1 rows returned in 0.00 seconds CSV Export
```

(8) Display the null values of employee and also employee name's third character should be 'a'.

Ans:

```
User: D22DCE190
Home > SQL > SQL Commands
 Autocommit Display 10
SELECT * FROM Employee WHERE EMP_COMM IS NULL OR MANAGER_ID IS NULL AND EMP_NAME LIKE '__a%';
Save Run

Results Explain Describe Saved SQL History
EMP_NO EMP_NAME EMP_SAL EMP_COMM DEPT_NO L_NAME DEPT_NAME JOB_ID LOCATION MANAGER_ID HIREDATE
101 Smith 800 - 20 shah machine learning fi_mgr toronto 105 09-AUG-98
104 Aman 3000 - 15 sharma virtual reality comp_op mexico 12 02-OCT-97
107 Anamika 2975 - 30 jha artificial intelligence it_prog new york - 15-JUL-97
3 rows returned in 0.00 seconds CSV Export
```

(9) What will be output if you are giving LIKE predicate as '%_%' ESCAPE '\'

Ans:

```
User: D22DCE190
Home > SQL > SQL Commands
 Autocommit Display 10
SELECT * FROM JOB WHERE JOB_ID LIKE '%\_%'ESCAPE'\';
Save Run

Results Explain Describe Saved SQL History
JOB_ID JOB_TITLE MIN_SAL MAX_SAL
it_prog Programmer 4000 10000
mk_mngr Marketing manager 9000 15000
fi_mngr Finance manager 8200 12000
fi_acc Account 4200 9000
comp_op Computer Operator 1500 3000
5 rows returned in 0.03 seconds CSV Export
```

Conclusion: In this practical I studied various options of LIKE predicate and how to find null and not null values and how to find length of particular data.

Remarks: _____

Marks: _____

Signature: _____

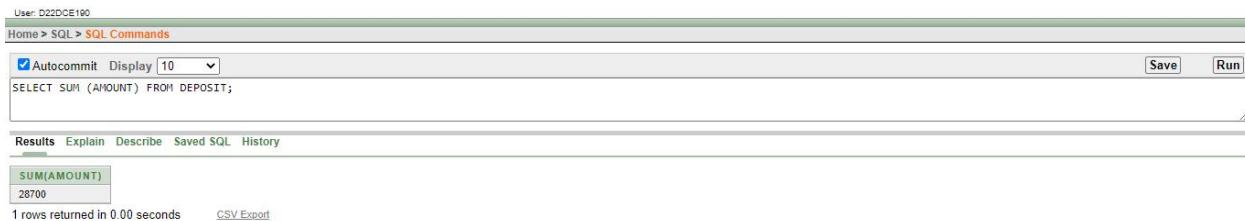
Practical – 3

Aim: To Perform various data manipulation commands, aggregate functions and sorting concept on all created tables .

Implementation:

(1) List total deposit from deposit.

Ans :



User: D22DCE190
Home > SQL > SQL Commands
 Autocommit Display 10
SELECT SUM (AMOUNT) FROM DEPOSIT;

Results Explain Describe Saved SQL History

SUM(AMOUNT)
28700

1 rows returned in 0.00 seconds [CSV Export](#)

(2) List total loan from karolbagh branch

Ans :



User: D22DCE190
Home > SQL > SQL Commands
 Autocommit Display 10
SELECT SUM(AMOUNT) FROM BORROW WHERE BNAME='KAROLBAGH';

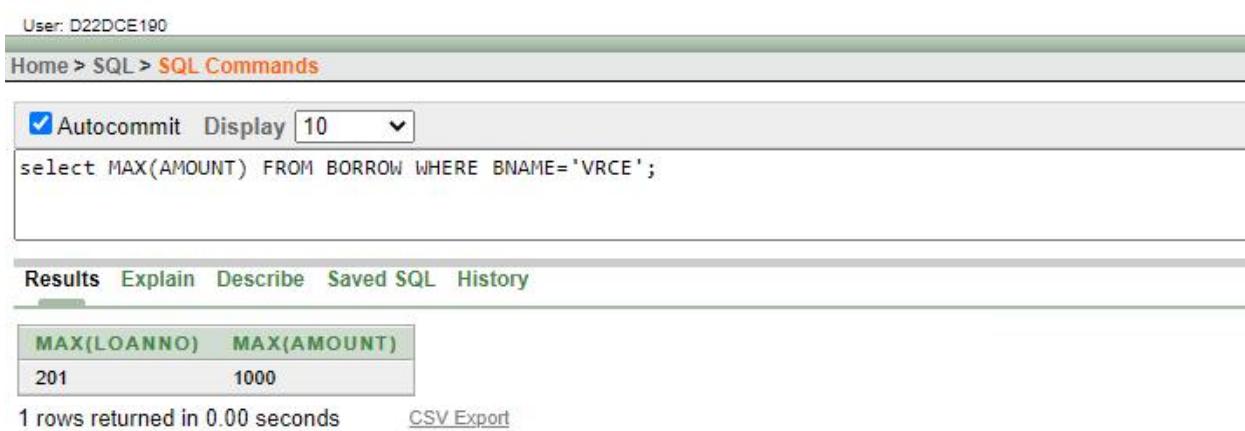
Results Explain Describe Saved SQL History

SUM(AMOUNT)
-

1 rows returned in 0.02 seconds [CSV Export](#)

(3) Give maximum loan from branch vrce.

Ans :



User: D22DCE190
Home > SQL > SQL Commands
 Autocommit Display 10
select MAX(AMOUNT) FROM BORROW WHERE BNAME='VRCE';

Results Explain Describe Saved SQL History

MAX(LOANNO)	MAX(AMOUNT)
201	1000

1 rows returned in 0.00 seconds [CSV Export](#)

(4) Count total number of customers**Ans :**

User: D22DCE190

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
SELECT COUNT(CNAME) AS "TOTAL" FROM CUSTOMERS;
```

Results Explain Describe Saved SQL History

TOTAL
10

1 rows returned in 0.00 seconds [CSV Export](#)

(5) Count total number of customer's cities.**Ans :**

User: D22DCE190

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
SELECT COUNT(CITY) AS "TOTAL" FROM CUSTOMERS;
```

Results Explain Describe Saved SQL History

TOTAL
10

1 rows returned in 0.00 seconds [CSV Export](#)

(6) Create table supplier from employee with all the columns.**Ans :**

User: D22DCE190

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
CREATE TABLE SUPPLIER AS SELECT * FROM EMPLOYEE ;
```

Results Explain Describe Saved SQL History

Table created.

(7) Create table sup1 from employee with first two columns.

Ans :

User: D22DCE190

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
CREATE TABLE SUP1 AS SELECT EMP_NO , EMP_NAME FROM EMPLOYEE ;
```

Results Explain Describe Saved SQL History

Table created.

(8) Create table sup2 from employee with no data

Ans :

User: D22DCE190

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
CREATE TABLE SUP2 AS SELECT * FROM EMPLOYEE WHERE EMP_SAL<800 ;
```

Results Explain Describe Saved SQL History

Table created.

(9) Insert the data into sup2 from employee whose second character should be 'n' and string should be 5 characters long in employee name field.

Ans :

User: D22DCE190

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
INSERT INTO SUP2 (SELECT * FROM EMPLOYEE WHERE EMP_NAME LIKE '_N___');|
```

Results Explain Describe Saved SQL History

0 row(s) inserted.

(10) Delete all the rows from sup1.

Ans :

User: D22DCE190

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
DELETE FROM SUP1;
```

Results Explain Describe Saved SQL History

7 row(s) deleted.

(11) Delete the detail of supplier whose sup_no is 103.

Ans:

User: D22DCE190

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
DELETE FROM SUPPLIER WHERE EMP_NO='103';
```

Results Explain Describe Saved SQL History

1 row(s) deleted.

(12) Rename the table sup2.

Ans :

User: D22DCE190

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
ALTER TABLE SUP2 RENAME TO SUP3;
```

Results Explain Describe Saved SQL History

Table altered.

(13) Destroy table sup1 with all the data.

Ans :

User: D22DCE190

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
DROP TABLE SUP1;
```

Results Explain Describe Saved SQL History

Table dropped.

(14) Update the value dept_no to 10 where second character of emp. name is ‘m’.

Ans :

User: D22DCE190

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
UPDATE EMPLOYEE SET DEPT_NO=10 WHERE EMP_NAME LIKE '_m%';
```

Results Explain Describe Saved SQL History

1 row(s) updated.

(15) Update the value of employee name whose employee number is 103.

Ans :

User: D22DCE190

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
UPDATE EMPLOYEE SET DEPT_NAME='PATEL' WHERE EMP_NO=103;
```

Results Explain Describe Saved SQL History

1 row(s) updated.

(16) Add one column phone to employee with size of column is 10.

Ans :

User: D22DCE190

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
ALTER TABLE EMPLOYEE ADD PHONE NUMBER(10);
```

Results Explain Describe Saved SQL History

Table altered.

(17) Modify the column emp_name to hold maximum of 30 characters.

Ans :

User: D22DCE190

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
ALTER TABLE EMPLOYEE MODIFY EMP_NAME VARCHAR2(30);
```

Results Explain Describe Saved SQL History

Table altered.

(18) Count the total no as well as distinct rows in dept_no column with a condition of salary greater than 1000 of employee

Ans :

User: D22DCE190

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
SELECT COUNT(DISTINCT DEPT_NO) FROM EMPLOYEE WHERE EMP_SAL>1000;
```

Results Explain Describe Saved SQL History

COUNT(DISTINCTDEPT_NO)
5

1 rows returned in 0.00 seconds CSV Export

(19) Display the detail of all employees in ascending order, descending order of their name and no.

Ans :

User: D22DCE190
Home > SQL > SQL Commands

Autocommit Display 10

```
SELECT * FROM EMPLOYEE ORDER BY EMP_NO DESC;
```

Results Explain Describe Saved SQL History

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE	PHONE
107	Anamika	2975	-	30	jha	artificial intelligence	it_prog	new york	-	15-JUL-97	-
106	Sneha	2450	24500	10	joseph	big data analytics	fi_acc	melbourne	105	26-SEP-97	-
105	Anita	5000	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-98	-
104	Aman	3000	-	15	sharma	virtual reality	comp_op	mexico	12	02-OCT-97	-
103	Adama	1100	0	20	wales	PATEL	mk_mgr	ontario	105	30-NOV-95	-
102	Snehal	1600	300	25	gupta	data science	lec	las vegas	-	14-MAR-96	-
101	Smith	800	-	10	shah	machine learning	fi_mgr	toronto	105	09-AUG-96	-

(20) Display the dept_no in ascending order and accordingly display emp_comm in descending order.

Ans :

User: D22DCE190
Home > SQL > SQL Commands

Autocommit Display 10

```
SELECT * FROM EMPLOYEE ORDER BY EMP_NO ASC , EMP_COMM DESC;
```

Results Explain Describe Saved SQL History

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE	PHONE
101	Smith	800	-	10	shah	machine learning	fi_mgr	toronto	105	09-AUG-96	-
102	Snehal	1600	300	25	gupta	data science	lec	las vegas	-	14-MAR-96	-
103	Adama	1100	0	20	wales	PATEL	mk_mgr	ontario	105	30-NOV-95	-
104	Aman	3000	-	15	sharma	virtual reality	comp_op	mexico	12	02-OCT-97	-
105	Anita	5000	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-98	-
106	Sneha	2450	24500	10	joseph	big data analytics	fi_acc	melbourne	105	26-SEP-97	-
107	Anamika	2975	-	30	jha	artificial intelligence	it_prog	new york	-	15-JUL-97	-

(21) Update the value of emp_comm to 500 where dept_no is 20.

Ans :

User: D22DCE190
Home > SQL > SQL Commands

Autocommit Display 10

```
SELECT * FROM EMPLOYEE ORDER BY EMP_NO ASC , EMP_COMM DESC;
```

Results Explain Describe Saved SQL History

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE	PHONE
101	Smith	800	-	10	shah	machine learning	fi_mgr	toronto	105	09-AUG-96	-
102	Snehal	1600	300	25	gupta	data science	lec	las vegas	-	14-MAR-96	-
103	Adama	1100	0	20	wales	PATEL	mk_mgr	ontario	105	30-NOV-95	-
104	Aman	3000	-	15	sharma	virtual reality	comp_op	mexico	12	02-OCT-97	-
105	Anita	5000	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-98	-
106	Sneha	2450	24500	10	joseph	big data analytics	fi_acc	melbourne	105	26-SEP-97	-
107	Anamika	2975	-	30	jha	artificial intelligence	it_prog	new york	-	15-JUL-97	-

(22) Display the emp_comm in ascending order with null value first and accordingly sort employee salary in descending order.

Ans :

```
User: D22DCE190
Home > SQL > SQL Commands

 Autocommit Display 10 ▾
SELECT * FROM EMPLOYEE ORDER BY EMP_COMM ASC NULLS FIRST, EMP_SAL DESC;

Results Explain Describe Saved SQL History
```

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE	PHONE
104	AMAN	3000	-	10	SHARMA	VIRTUAL REALITY	COMP_OP	MEXICO	12	02-OCT-97	-
107	ANAMIKA	2975	-	30	JHA	ARTIFICIAL INTELLIGENCE	IT_PROG	NEW YORK	-	15-JUL-97	-
101	SMITH	800	-	10	SHAH	MACHINE LEARNING	FI_MGR	TORONTO	105	09-AUG-96	-
102	SNEHAL	1600	300	25	GUPTA	DATA SCIENCE	LEC	LAS VEGAS	-	14-MAR-96	-
103	HARSHAD	1100	500	20	WALES	MACHINE LEARNING	MK_MGR	ONTARIO	105	30-NOV-95	-
106	SNEHA	2450	24500	10	JOSEPH	BIG DATA ANALYSTIC	FI_ACC	MELBOURNE	105	26-SEP-97	-
105	ANITA	5000	50000	10	PATEL	BIG DATA ANALYSTIC	COMP_OP	GERMANY	107	01-JAN-98	-

7 rows returned in 0.00 seconds [CSV Export](#)

(23) Display the emp_comm in ascending order with null value last and accordingly sort emp_no in descending order.

Ans :

```
User: D22DCE190
Home > SQL > SQL Commands

 Autocommit Display 10 ▾
SELECT * FROM EMPLOYEE ORDER BY EMP_COMM NULLS LAST , EMP_NO DESC;

Results Explain Describe Saved SQL History
```

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE	PHONE
103	Adama	1100	0	20	wales	PATEL	mk_mgr	ontario	105	30-NOV-95	-
102	Snehal	1600	300	25	gupta	data science	lec	las vegas	-	14-MAR-96	-
106	Sneha	2450	24500	10	joseph	big data analytics	fi_acc	melbourne	105	26-SEP-97	-
105	Anita	5000	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-98	-
107	Anamika	2975	-	30	jha	artificial intelligence	it_prog	new york	-	15-JUL-97	-
104	Aman	3000	-	15	sharma	virtual reality	comp_op	mexico	12	02-OCT-97	-
101	Smith	800	-	10	shah	machine learning	fi_mgr	toronto	105	09-AUG-96	-

7 rows returned in 0.00 seconds [CSV Export](#)

Conclusion: In this practical I Performed various data manipulation commands, aggregate functions and sorting concept on all created tables

Remarks: _____

Marks: _____

Signature: _____

Practical - 4

Aim: To Implement Single-row functions.

Implementation:

(1) Write a query to display the current date. Label the column Date

Ans:

User: D22DCE190
Home > SQL > SQL Commands
Autocommit Display 10 ▾
SELECT SYSDATE AS "CURRENT DATE" FROM DUAL;
Results Explain Describe Saved SQL History
CURRENT DATE
19-JAN-23
1 rows returned in 0.02 seconds CSV Export

(2) For each employee, display the employee number, salary, and salary increased by 15% and expressed as a whole number. Label the column New Salary

Ans:

User: D22DCE190
Home > SQL > SQL Commands
Autocommit Display 10 ▾
SELECT EMP_NO , EMP_SAL , ROUND(EMP_SAL*.15) FROM EMPLOYEE;
Results Explain Describe Saved SQL History
EMP_NO EMP_SAL ROUND(EMP_SAL*.15)
101 800 120
102 1600 240
103 1100 165
104 3000 450
105 5000 750
106 2450 368
107 2975 446
7 rows returned in 0.00 seconds CSV Export

(3) Modify your query no (2) to add a column that subtracts the old salary from the new salary. Label the column Increase

Ans:

User: D22DCE190

Home > SQL > SQL Commands

Home > SQL > SQL Commands

Autocommit Display 10

```
SELECT EMP_NO,EMP_NAME,EMP_SAL,ROUND(EMP_SAL+(EMP_SAL*15/100)) AS "NEW SALARY",ROUND (EMP_SAL+(EMP_SAL*.15)-EMP_SAL) AS "INCREMENT" FROM EMPLOYEE;
```

Results Explain Describe Saved SQL History

EMP_NO	EMP_NAME	EMP_SAL	NEW SALARY	INCREMENT
101	SMITH	800	920	120
102	SNEHAL	1600	1840	240
103	HARSHAD	1100	1265	165
104	AMAN	3000	3450	450
105	ANITA	5000	5750	750
106	SNEHA	2450	2818	368
107	ANAMIKA	2975	3421	446

7 rows returned in 0.00 seconds [CSV Export](#)

(4) Write a query that displays the employee's names with the first letter capitalized and all other letters lowercase, and the length of the names, for all employees whose name starts with J, A, or M. Give each column an appropriate label. Sort the results by the employees' last names.

Ans:

ORACLE Database Express Edition

User: D22DCE190

Home > SQL > SQL Commands

Autocommit Display 10

```
SELECT INITCAP(EMP_NAME), LENGTH(EMP_NAME) FROM EMPLOYEE WHERE EMP_NAME LIKE 'J%' OR EMP_NAME LIKE 'A%' OR EMP_NAME LIKE 'M%' ORDER BY EMP_NAME;
```

Results Explain Describe Saved SQL History

INITCAP(EMP_NAME)	LENGTH(EMP_NAME)
Aman	4
Anamika	7
Anita	5

3 rows returned in 0.01 seconds [CSV Export](#)

(5) Write a query that produces the following for each employee:

<employee last name> earns <salary> monthly

Ans:

User: D22DCE190

Home > SQL > **SQL Commands**

Autocommit Display 10

```
SELECT L_NAME || ' EARNS ' || EMP_SAL || ' MONTHLY ' FROM EMPLOYEE;
```

Results Explain Describe Saved SQL History

L_NAME 'EARNS' EMP_SAL 'MONTHLY'
shah EARNS 800 MONTHLY
gupta EARNS 1600 MONTHLY
wales EARNS 1100 MONTHLY
sharma EARNS 3000 MONTHLY
patel EARNS 5000 MONTHLY
joseph EARNS 2450 MONTHLY
jha EARNS 2975 MONTHLY

7 rows returned in 0.00 seconds [CSV Export](#)

(6) Display the name, date, number of months employed and day of the week on which the employee has started. Order the results by the day of the week starting with Monday.

Ans:

User: D22DCE190

Home > SQL > **SQL Commands**

Autocommit Display 10

[Save](#) [Run](#)

```
SELECT EMP_NAME, HIREDATE, ROUND(MONTHS_BETWEEN(SYSDATE, HIREDATE)) AS "M_EMPLOYED", TO_CHAR(HIREDATE,'DAY') AS "D_EMPLOYED" FROM EMPLOYEE ORDER BY TO_CHAR(HIREDATE,'D');
```

Results Explain Describe Saved SQL History

EMP_NAME	HIREDATE	M_EMPLOYED	D_EMPLOYED
Anamika	15-JUL-97	306	TUESDAY
Adama	30-NOV-95	326	THURSDAY
Snehal	14-MAR-96	322	THURSDAY
Aman	02-OCT-97	304	THURSDAY
Anita	01-JAN-98	301	THURSDAY
Sneha	26-SEP-97	304	FRIDAY
Smith	09-AUG-96	317	FRIDAY

7 rows returned in 0.00 seconds [CSV Export](#)

(7) Display the date of emp in a format that appears as Seventh of June 1994 12:00:00 AM.

Ans:

User: D22DCE190
Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
SELECT TO_CHAR(HIREDATE, 'DDSPTH "OF" MONTH YYYY FM HH:MI:SS AM') AS "DATE_FORMAT" FROM EMPLOYEE;
```

Results Explain Describe Saved SQL History

DATE_FORMAT
NINTH OF AUGUST 1996 12:0:0 AM
FOURTEENTH OF MARCH 1996 12:0:0 AM
THIRTIETH OF NOVEMBER 1995 12:0:0 AM
SECOND OF OCTOBER 1997 12:0:0 AM
FIRST OF JANUARY 1998 12:0:0 AM
TWENTY-SIXTH OF SEPTEMBER 1997 12:0:0 AM
FIFTEENTH OF JULY 1997 12:0:0 AM

7 rows returned in 0.00 seconds [CSV Export](#)

(8) Write a query to calculate the annual compensation of all employees (sal +comm.).

Ans:

User: D22DCE190
Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
SELECT SUM(EMP_SAL+EMP_COMM) AS "COMPENSATION" FROM EMPLOYEE;
```

Results Explain Describe Saved SQL History

COMPENSATION
84950

1 rows returned in 0.00 seconds [CSV Export](#)

Conclusion: In this practical I learnt about Single-row functions and different type of time functions by which we can get current date and etc. and also implemented function by which we can convert initial letter capital.

Remarks: _____

Marks: _____

Signature: _____

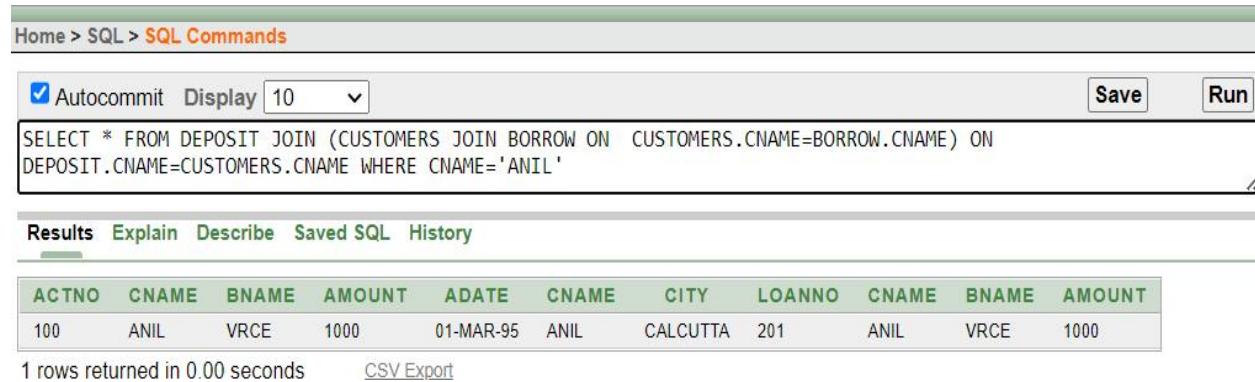
Practical - 5

Aim: Displaying data from Multiple Tables (join)

Implementation:

(1) Give details of customers ANIL.

Ans:



The screenshot shows a MySQL query interface. The SQL command entered is:

```
SELECT * FROM DEPOSIT JOIN (CUSTOMERS JOIN BORROW ON CUSTOMERS.CNAME=BORROW.CNAME) ON DEPOSIT.CNAME=CUSTOMERS.CNAME WHERE CNAME='ANIL'
```

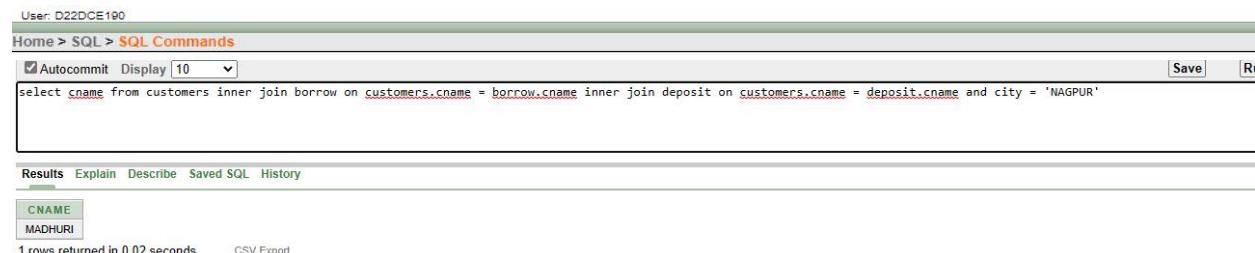
The results table has the following columns and data:

ACTNO	CNAME	BNAME	AMOUNT	ADATE	CNAME	CITY	LOANNO	CNAME	BNAME	AMOUNT
100	ANIL	VRCE	1000	01-MAR-95	ANIL	CALCUTTA	201	ANIL	VRCE	1000

1 rows returned in 0.00 seconds [CSV Export](#)

(2) Give name of customer who are borrowers and depositors and having living city Nagpur.

Ans:



The screenshot shows a MySQL query interface. The SQL command entered is:

```
select cname from customers inner join borrow on customers.cname = borrow.cname inner join deposit on customers.cname = deposit.cname and city = 'NAGPUR'
```

The results table has the following column and data:

CNAME
MADHURI

1 rows returned in 0.02 seconds [CSV Export](#)

(3) Give city as their city name of customers having same living branch.

Ans:

ORACLE Database Express Edition

User: D22DCE190

Home > SQL > SQL Commands

Autocommit Display 10

```
select city from customers inner join borrow on customers.cname = borrow.cname inner join branch on borrow.bname = branch.bname
```

Results Explain Describe Saved SQL History

CITY
NAGPUR
NAGPUR
NAGPUR
BOMBAY
DELHI

5 rows returned in 0.01 seconds [CSV Export](#)

(4) Write a query to display the last name, department number, and department name for all employees.

Ans:

User: D22DCE190

Home > SQL > SQL Commands

Autocommit Display 10

```
SELECT L_NAME,DEPT_NO,DEPT_NAME FROM EMPLOYEE
```

Results Explain Describe Saved SQL History

L_NAME	DEPT_NO	DEPT_NAME
shah	10	machine learning
gupta	25	data science
wales	20	PATEL
sharma	15	virtual reality
patel	10	big data analytics
joseph	10	big data analytics
jha	30	artificial intelligence

7 rows returned in 0.00 seconds [CSV Export](#)

(5) Create a unique listing of all jobs that are in department 30. Include the location of the department in the output

Ans:

ORACLE® Database Express Edition

User: D22DCE190

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
select job.job_id,employee.location from job inner join employee on job.job_id = employee.job_id and dept_no = 30
```

Results Explain Describe Saved SQL History

JOB_ID	LOCATION
IT_PROG	NEW YORK

1 rows returned in 0.01 seconds [CSV Export](#)

(6) Write a query to display the employee's name, department number, and department name for all employees who work in NEW YORK.

Ans:

User: D22DCE190

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
SELECT EMP_NAME,DEPT_NO,DEPT_NAME FROM EMPLOYEE WHERE LOCATION = 'NEW YORK'
```

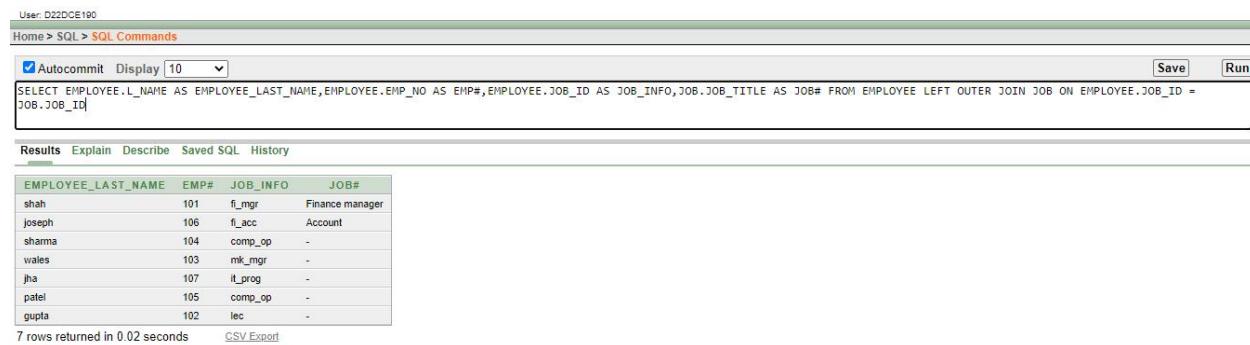
Results Explain Describe Saved SQL History

EMP_NAME	DEPT_NO	DEPT_NAME
ANAMIKA	30	ARTIFICIAL INTELLIGENCE

1 rows returned in 0.00 seconds [CSV Export](#)

(7) Display the employee's last name and employee number along with their manager's last name and manager number. Label the columns Employee, Emp#, Manager, and Mgr#, respectively.

Ans:



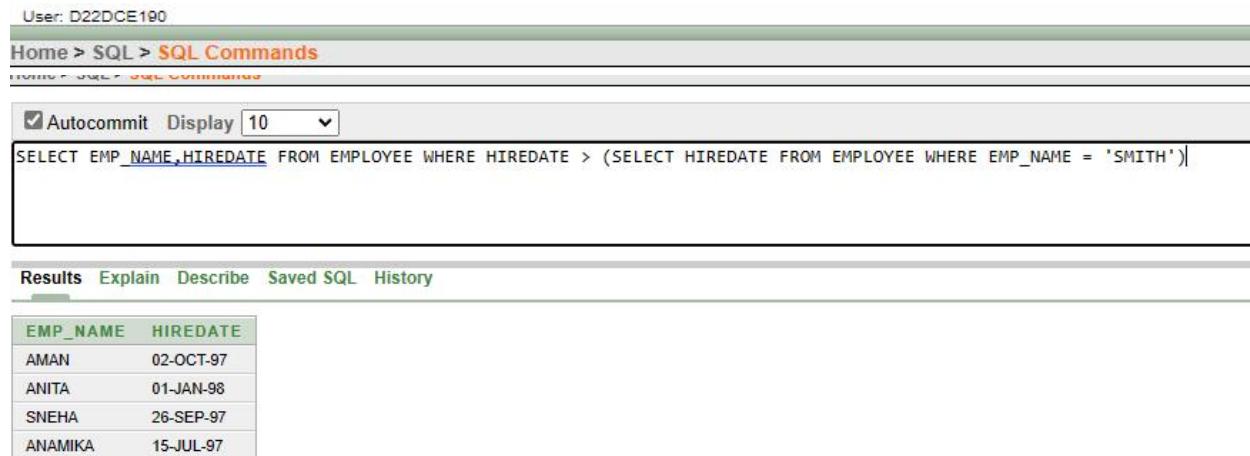
```
User: D22DCE190
Home > SQL > SQL Commands
 Autocommit Display 10  
SELECT EMPLOYEE.L_NAME AS EMPLOYEE_LAST_NAME,EMPLOYEE.EMP_NO AS EMP#,EMPLOYEE.JOB_ID AS JOB_INFO,JOB.JOB_TITLE AS JOB# FROM EMPLOYEE LEFT OUTER JOIN JOB ON EMPLOYEE.JOB_ID = JOB.JOB_ID
Results Explain Describe Saved SQL History
```

EMPLOYEE_LAST_NAME	EMP#	JOB_INFO	JOB#
shah	101	fi_mgr	Finance manager
joseph	106	fi_acc	Account
sharma	104	comp_op	-
wales	103	mkt_mgr	-
jha	107	it_prog	-
patel	105	comp_op	-
gupta	102	lec	-

7 rows returned in 0.02 seconds [CSV Export](#)

(8) Create a query to display the name and hire date of any employee hired after employee "smith".

Ans:



```
User: D22DCE190
Home > SQL > SQL Commands
 Autocommit Display 10  
SELECT EMP_NAME,HIREDATE FROM EMPLOYEE WHERE HIREDATE > (SELECT HIREDATE FROM EMPLOYEE WHERE EMP_NAME = 'SMITH')
Results Explain Describe Saved SQL History
```

EMP_NAME	HIREDATE
AMAN	02-OCT-97
ANITA	01-JAN-98
SNEHA	26-SEP-97
ANAMIKA	15-JUL-97

Conclusion: In this practical I learnt about how to display data from different tables. The purpose of JOINS in SQL is to access data from multiple tables based on logical relationships between them. JOINS are used to fetch data from database tables and represent the result dataset as a separate table.

Remarks: _____

Marks: _____

Signature: _____

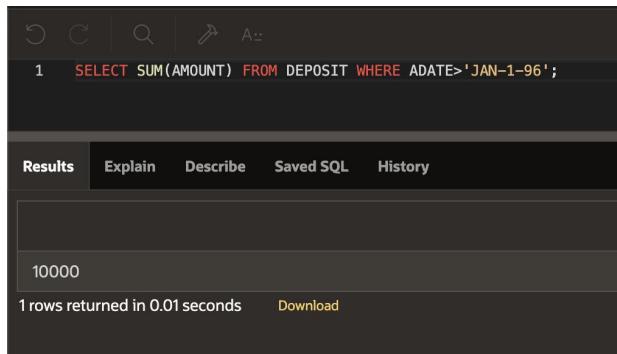
Practicle-6

Aim: To apply the concept of Aggregating Data using Group functions.

Implementation:

(1) List total deposit of customer having account date after 1-jan-96.

Ans:



The screenshot shows a dark-themed SQL query interface. At the top, there is a toolbar with icons for back, forward, search, and other database operations. Below the toolbar, a code editor window displays the following SQL query:

```
1 SELECT SUM(AMOUNT) FROM DEPOSIT WHERE ADATE > 'JAN-1-96';
```

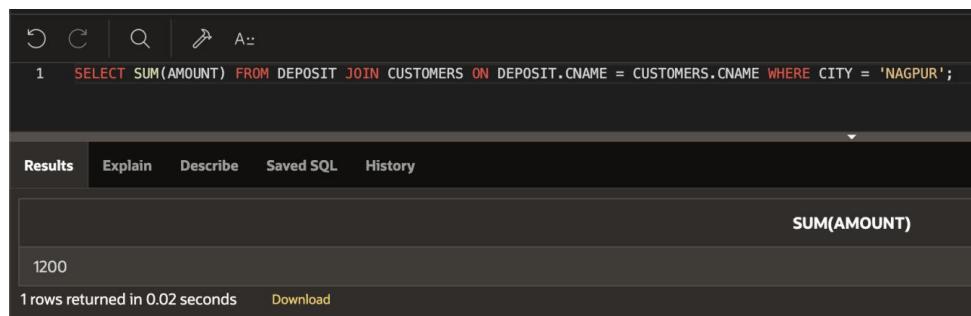
Below the code editor is a navigation bar with tabs: Results (which is selected), Explain, Describe, Saved SQL, and History. The main area is labeled "Results" and contains the following output:

SUM(AMOUNT)
10000

At the bottom of the results area, it says "1 rows returned in 0.01 seconds" and has a "Download" link.

(2) List total deposit of customers living in city Nagpur.

Ans:



The screenshot shows a dark-themed SQL query interface. At the top, there is a toolbar with icons for back, forward, search, and other database operations. Below the toolbar, a code editor window displays the following SQL query:

```
1 SELECT SUM(AMOUNT) FROM DEPOSIT JOIN CUSTOMERS ON DEPOSIT.CNAME = CUSTOMERS.CNAME WHERE CITY = 'NAGPUR';
```

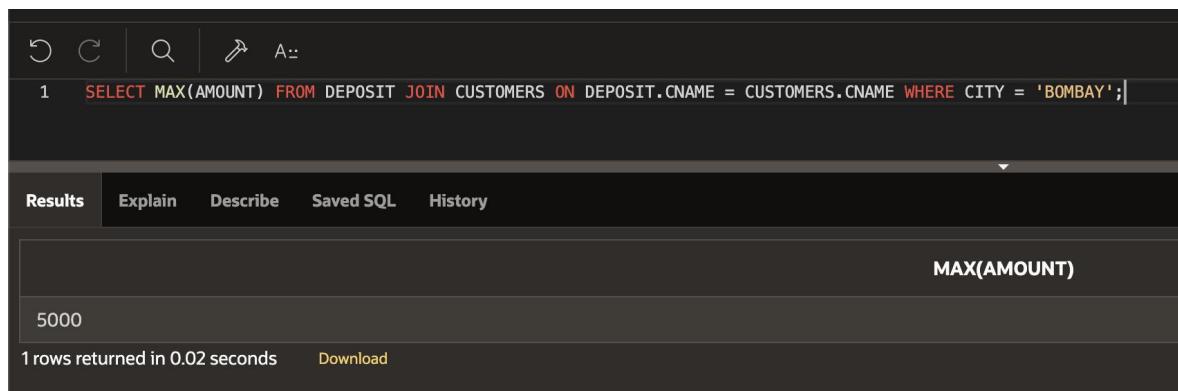
Below the code editor is a navigation bar with tabs: Results (selected), Explain, Describe, Saved SQL, and History. The main area is labeled "Results" and contains the following output:

SUM(AMOUNT)
1200

At the bottom of the results area, it says "1 rows returned in 0.02 seconds" and has a "Download" link.

(3) List maximum deposit of customers living in bombay.

Ans:



The screenshot shows a dark-themed SQL query interface. At the top, there is a toolbar with icons for back, forward, search, and other database operations. Below the toolbar, a code editor window displays the following SQL query:

```
1 SELECT MAX(AMOUNT) FROM DEPOSIT JOIN CUSTOMERS ON DEPOSIT.CNAME = CUSTOMERS.CNAME WHERE CITY = 'BOMBAY';
```

Below the code editor is a navigation bar with tabs: Results (selected), Explain, Describe, Saved SQL, and History. The main area is labeled "Results" and contains the following output:

MAX(AMOUNT)
5000

At the bottom of the results area, it says "1 rows returned in 0.02 seconds" and has a "Download" link.

(4) Display the highest, lowest, sum, and average salary of all employees. Label the columns Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number.

Ans:

```
1  SELECT ROUND(MAX(EMP_SAL)) AS MAXIMUM, ROUND(MIN(EMP_SAL)) AS MINIMUM, ROUND(SUM(EMP_SAL)) AS SUM, ROUND(AVG(EMP_SAL)) AS AVERAGE FROM EMPLOYEE;
```

MAXIMUM	MINIMUM	SUM	AVERAGE
5000	800	16925	2418

1 rows returned in 0.01 seconds [Download](#)

(5) Write a query that displays the difference between the highest and lowest salaries. Label the column DIFFERENCE.

Ans:

```
1  SELECT ROUND(MAX(EMP_SAL)) - ROUND(MIN(EMP_SAL)) AS DIFFERENCE FROM EMPLOYEE;
```

DIFFERENCE
4200

1 rows returned in 0.05 seconds [Download](#)

(6) Create a query that will display the total number of employees and, of that total, the number of employees hired in 1995, 1996, 1997, and 1998

Ans:

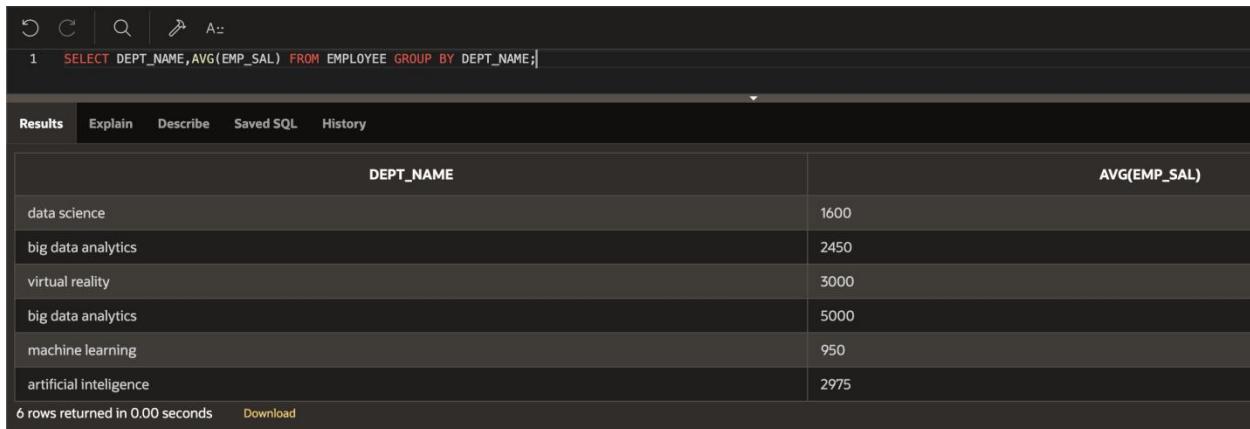
```
1  SELECT TO_CHAR(HIREDATE, 'YYYY'), COUNT(*) FROM EMPLOYEE GROUP BY TO_CHAR(HIREDATE, 'YYYY');
```

TO_CHAR(HIREDATE,'YYYY')	COUNT(*)
0095	1
0096	2
0098	1
0097	3

4 rows returned in 0.01 seconds [Download](#)

(7) Find the average salaries for each department without displaying the respective department numbers.

Ans:



The screenshot shows a SQL query results interface. At the top, there are icons for refresh, copy, search, and a dropdown menu. Below that is a code editor with the following SQL query:

```
1 SELECT DEPT_NAME, AVG(EMP_SAL) FROM EMPLOYEE GROUP BY DEPT_NAME;
```

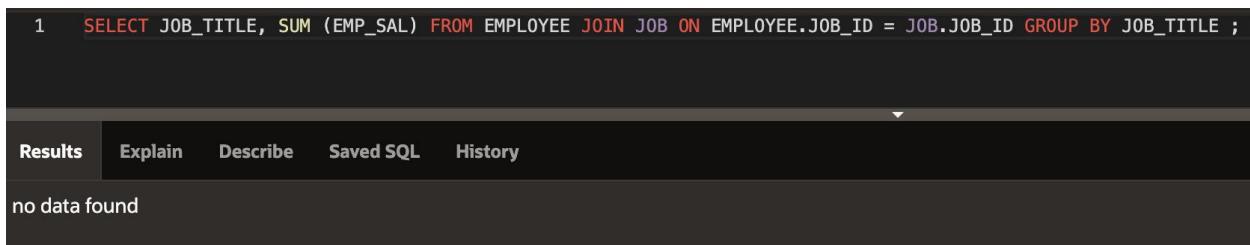
Below the code editor is a navigation bar with tabs: Results, Explain, Describe, Saved SQL, and History. The Results tab is selected. The main area displays a table with two columns: DEPT_NAME and AVG(EMP_SAL). The data is as follows:

DEPT_NAME	AVG(EMP_SAL)
data science	1600
big data analytics	2450
virtual reality	3000
big data analytics	5000
machine learning	950
artificial intelligence	2975

At the bottom left, it says "6 rows returned in 0.00 seconds". At the bottom right, there is a "Download" button.

(8) Write a query to display the total salary being paid to each job title, within each department.

Ans:



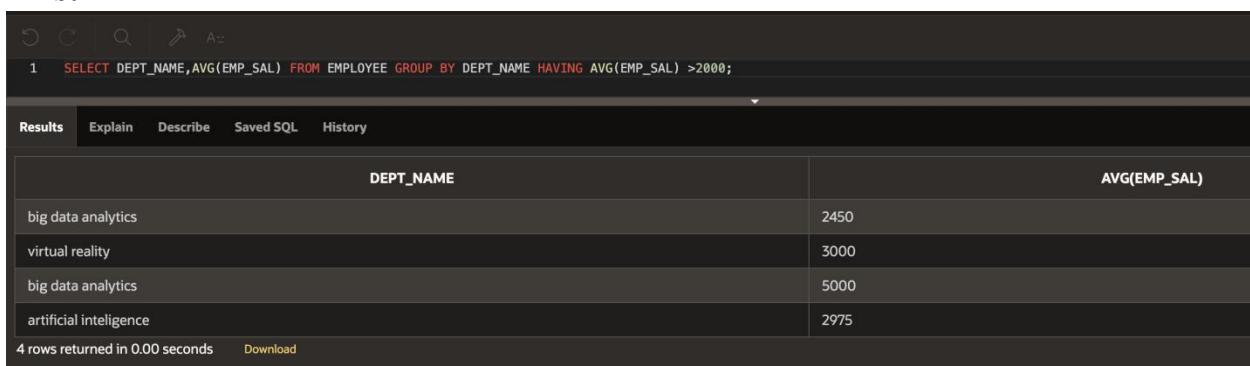
The screenshot shows a SQL query results interface. At the top, there are icons for refresh, copy, search, and a dropdown menu. Below that is a code editor with the following SQL query:

```
1 SELECT JOB_TITLE, SUM (EMP_SAL) FROM EMPLOYEE JOIN JOB ON EMPLOYEE.JOB_ID = JOB.JOB_ID GROUP BY JOB_TITLE ;
```

Below the code editor is a navigation bar with tabs: Results, Explain, Describe, Saved SQL, and History. The Results tab is selected. The main area displays the message "no data found".

(9) Find the average salaries > 2000 for each department without displaying the respective department numbers.

Ans:



The screenshot shows a SQL query results interface. At the top, there are icons for refresh, copy, search, and a dropdown menu. Below that is a code editor with the following SQL query:

```
1 SELECT DEPT_NAME, AVG(EMP_SAL) FROM EMPLOYEE GROUP BY DEPT_NAME HAVING AVG(EMP_SAL) >2000;
```

Below the code editor is a navigation bar with tabs: Results, Explain, Describe, Saved SQL, and History. The Results tab is selected. The main area displays a table with two columns: DEPT_NAME and AVG(EMP_SAL). The data is as follows:

DEPT_NAME	AVG(EMP_SAL)
big data analytics	2450
virtual reality	3000
big data analytics	5000
artificial intelligence	2975

At the bottom left, it says "4 rows returned in 0.00 seconds". At the bottom right, there is a "Download" button.

(10) Display the job and total salary for each job with a total salary amount exceeding 3000 and sorts the list by the total salary.

Ans:

```

1 ✓ SELECT JOB_TITLE, SUM (EMP_SAL) FROM EMPLOYEE JOIN JOB ON EMPLOYEE.JOB_ID = JOB.JOB_ID
2   | GROUP BY JOB_TITLE HAVING SUM(EMP_SAL) > 3000 ORDER BY SUM(EMP_SAL);

Results Explain Describe Saved SQL History
no data found

```

(11) List the branches having sum of deposit more than 5000 and located in city bombay.

Ans:

```

SELECT DEPOSIT.BNAME,SUM(AMOUNT) FROM DEPOSIT JOIN BRANCH ON DEPOSIT.BNAME = BRANCH.BNAME WHERE CITY='BOMBAY' GROUP BY DEPOSIT.BNAME HAVING SUM(AMOUNT) > 5000;

Results Explain Describe Saved SQL History
BNAME                                     SUM(AMOUNT)
POWAI                                         7000
1 rows returned in 0.01 seconds    Download

```

Remarks: _____

Marks: _____

Signature: _____

Practical-7

Aim: To solve queries using the concept of sub query.

Implementation:

(1) Write a query to display the last name and hire date of any employee in the same department as smith. Exclude smith

Ans:

```
1  SELECT L_NAME,HIREDATE FROM EMPLOYEE WHERE DEPT_NAME = (SELECT DEPT_NAME FROM EMPLOYEE WHERE EMP_NAME = 'Smith') AND EMP_NAME != 'Smith'
```

Results		Explain	Describe	Saved SQL	History
L_NAME	HIREDATE				
Wales	11/30/0095				

1 rows returned in 0.01 seconds Download

(2) Give name of customers who are depositors having same branch city of mr. sunil.

Ans:

```
1  SELECT CNAME FROM DEPOSIT JOIN BRANCH ON DEPOSIT.BNAME = BRANCH.BNAME WHERE CITY = (SELECT CITY FROM BRANCH JOIN DEPOSIT ON BRANCH.BNAME = DEPOSIT.BNAME WHERE CNAME = 'SUNIL'))
```

Results		Explain	Describe	Saved SQL	History
CNAME					
ANIL					
SUNIL					

(3) Give deposit details and loan details of customer in same city where pramod is living.

Ans:

```
1  SELECT * FROM DEPOSIT JOIN BORROW ON DEPOSIT.CNAME = BORROW.CNAME JOIN CUSTOMERS ON DEPOSIT.CNAME = CUSTOMERS.CNAME WHERE CITY = (SELECT CITY FROM CUSTOMERS WHERE CNAME = 'PRAMOD'))
```

Results											Explain	Describe	Saved SQL	History	
ACTNO	CNAME	BNAME	AMOUNT	ADATE	LOANNO	CNAME	BNAME	AMOUNT	CNAME	CITY					
104	MADHURI	CHANDI	1200	12/17/0095	321	MADHURI	ANDHERI	2000	MADHURI	NAGPUR					

1 rows returned in 0.01 seconds Download

(4) Create a query to display the employee numbers and last names of all employees who earn more than the average salary. Sort the results in ascending order of salary.

Ans:

```
1  SELECT EMP_NO,L_NAME FROM EMPLOYEE WHERE EMP_SAL > (SELECT AVG(EMP_SAL) FROM EMPLOYEE)44
```

Results		Explain	Describe	Saved SQL	History
EMP_NO		L_NAME			
105		Patel			
106		Joseph			
104		Sharma			
107		Jha			

4 rows returned in 0.01 seconds Download

(5) Give names of depositors having same living city as mr. anil and having deposit amount greater than 2000

Ans:

```
1  SELECT DEPOSIT.CNAME FROM DEPOSIT JOIN CUSTOMERS ON DEPOSIT.CNAME = CUSTOMERS.CNAME
2  WHERE CITY = (SELECT CITY FROM CUSTOMERS WHERE CNAME = 'ANIL' AND AMOUNT > 2000)
```

Results		Explain	Describe	Saved SQL	History
no data found					

(6) Display the last name and salary of every employee who reports to ford.

Ans:

```
1  SELECT EMP_NO,L_NAME FROM EMPLOYEE WHERE EMP_SAL > (SELECT AVG(EMP_SAL) FROM EMPLOYEE)
```

Results		Explain	Describe	Saved SQL	History
EMP_NO		L_NAME			
105		Patel			
106		Joseph			
104		Sharma			
107		Jha			

4 rows returned in 0.01 seconds Download

(7) Display the department number, name, and job for every employee in the accounting department.

Ans:

```
1  SELECT DEPT_NO,EMP_NAME,JOB_TITLE FROM EMPLOYEE JOIN JOB ON EMPLOYEE.JOB_ID = JOB.JOB_ID  
2  WHERE JOB_TITLE = (SELECT JOB_TITLE FROM JOB WHERE JOB_TITLE = 'ACCOUNT')
```

Results Explain Describe Saved SQL History

no data found

(8) List the name of branch having highest number of depositors.

Ans:

```
1  SELECT BNAME FROM DEPOSIT GROUP BY BNAME HAVING COUNT(BNAME) = (SELECT MAX(COUNT(*)) FROM DEPOSIT GROUP BY BNAME)
```

Results Explain Describe Saved SQL History

BNAME

CHANDI

M.G.ROAD

ANDHERI

VIRAR

NEHRUPLACE

VRCE

AJNII

KAROLBAGH

POWAI

(9) Give the name of cities where in which the maximum numbers of branches are located.

Ans:

```
1  SELECT CITY FROM BRANCH GROUP BY CITY HAVING COUNT(CITY) = (SELECT MAX(COUNT(*)) FROM BRANCH GROUP BY CITY);
```

Results Explain Describe Saved SQL History

CITY
BOMBAY
DELHI
NAGPUR

3 rows returned in 0.00 seconds [Download](#)

(10) Give name of customers living in same city where maximum depositors are located.

Ans:

```
1  SELECT CNAME FROM CUSTOMERS WHERE CITY = (SELECT CITY FROM CUSTOMERS GROUP BY CITY HAVING COUNT(CITY) = (SELECT MAX(COUNT(*)) FROM CUSTOMERS GROUP BY CITY))
```

Results Explain Describe Saved SQL History

CNAME
SHIVANI
KRANTI
NAREN

3 rows returned in 0.01 seconds [Download](#)

Remarks: _____

Marks: _____

Signature: _____

Practical-8

Aim: Manipulating Data

Implementation

(1) Give 10% interest to all depositors.

Ans:

```
1
2 UPDATE DEPOSIT SET AMOUNT=AMOUNT + (AMOUNT * 0.1);

Results Explain Describe Saved SQL History

9 row(s) updated.
```

(2) Give 10% interest to all depositors having branch vrce

Ans:

```
1 UPDATE DEPOSIT SET AMOUNT=AMOUNT + (AMOUNT * 0.1) WHERE BNAME='VRCE';
2

Results Explain Describe Saved SQL History

9 row(s) updated.
```

(3) Give 10% interest to all depositors living in nagpur and having branch city bombay.

Ans:

```
1 UPDATE DEPOSIT SET AMOUNT=AMOUNT + (AMOUNT * 0.1) WHERE CNAME IN(SELECT CNAME FROM CUSTOMERS WHERE CITY = 'NAGPUR')
2 | AND BNAME IN(SELECT BNAME FROM BRANCH WHERE CITY = 'BOMBAY');

Results Explain Describe Saved SQL History

0 row(s) updated.
```

(4) Transfer 10 Rs from account of anil to sunil if both are having same branch.

Ans:

1 UPDATE DEPOSIT1 SET AMOUNT=AMOUNT-10 WHERE CNAME='ANIL';	2 UPDATE DEPOSIT SET AMOUNT=AMOUNT+10 WHERE CNAME='SUNIL';	3 SELECT * FROM DEPOSIT;	4																																																		
<table border="1"> <tr> <th>Results</th> <th>Explain</th> <th>Describe</th> <th>Saved SQL</th> <th>History</th> </tr> <tr> <td colspan="5" style="text-align: center;">Describe</td> </tr> <tr> <td colspan="5">1 row(s) updated.</td> </tr> </table>				Results	Explain	Describe	Saved SQL	History	Describe					1 row(s) updated.																																							
Results	Explain	Describe	Saved SQL	History																																																	
Describe																																																					
1 row(s) updated.																																																					
1 UPDATE DEPOSIT1 SET AMOUNT=AMOUNT-10 WHERE CNAME='ANIL';	2 UPDATE DEPOSIT SET AMOUNT=AMOUNT+10 WHERE CNAME='SUNIL';	3 SELECT * FROM DEPOSIT;	4																																																		
<table border="1"> <tr> <th>Results</th> <th>Explain</th> <th>Describe</th> <th>Saved SQL</th> <th>History</th> </tr> <tr> <td colspan="5" style="text-align: center;">Describe</td> </tr> <tr> <td colspan="5">1 row(s) updated.</td> </tr> </table>				Results	Explain	Describe	Saved SQL	History	Describe					1 row(s) updated.																																							
Results	Explain	Describe	Saved SQL	History																																																	
Describe																																																					
1 row(s) updated.																																																					
1 UPDATE DEPOSIT1 SET AMOUNT=AMOUNT-10 WHERE CNAME='ANIL'; Undo 2 UPDATE DEPOSIT SET AMOUNT=AMOUNT+10 WHERE CNAME='SUNIL'; 3 SELECT * FROM DEPOSIT;																																																					
<table border="1"> <thead> <tr> <th>ACTNO</th> <th>CNAME</th> <th>BNAME</th> <th>AMOUNT</th> <th>ADATE</th> </tr> </thead> <tbody> <tr><td>100</td><td>ANIL</td><td>VRCE</td><td>1100</td><td>03/01/0095</td></tr> <tr><td>107</td><td>SHIVANI</td><td>VIRAR</td><td>1100</td><td>09/05/0095</td></tr> <tr><td>102</td><td>MEHUL</td><td>KAROLBAGH</td><td>3850</td><td>11/17/0095</td></tr> <tr><td>106</td><td>SANDIP</td><td>ANDHERI</td><td>2200</td><td>03/31/0096</td></tr> <tr><td>101</td><td>SUNIL</td><td>AJNI</td><td>5520</td><td>01/04/0096</td></tr> <tr><td>104</td><td>MADHURI</td><td>CHANDI</td><td>1320</td><td>12/17/0095</td></tr> <tr><td>105</td><td>PRAMOD</td><td>M.G.ROAD</td><td>3300</td><td>03/27/0096</td></tr> <tr><td>108</td><td>KRANTI</td><td>NEHRUPLACE</td><td>5500</td><td>07/02/0095</td></tr> <tr><td>109</td><td>MINU</td><td>POWAI</td><td>7700</td><td>08/10/0095</td></tr> </tbody> </table>				ACTNO	CNAME	BNAME	AMOUNT	ADATE	100	ANIL	VRCE	1100	03/01/0095	107	SHIVANI	VIRAR	1100	09/05/0095	102	MEHUL	KAROLBAGH	3850	11/17/0095	106	SANDIP	ANDHERI	2200	03/31/0096	101	SUNIL	AJNI	5520	01/04/0096	104	MADHURI	CHANDI	1320	12/17/0095	105	PRAMOD	M.G.ROAD	3300	03/27/0096	108	KRANTI	NEHRUPLACE	5500	07/02/0095	109	MINU	POWAI	7700	08/10/0095
ACTNO	CNAME	BNAME	AMOUNT	ADATE																																																	
100	ANIL	VRCE	1100	03/01/0095																																																	
107	SHIVANI	VIRAR	1100	09/05/0095																																																	
102	MEHUL	KAROLBAGH	3850	11/17/0095																																																	
106	SANDIP	ANDHERI	2200	03/31/0096																																																	
101	SUNIL	AJNI	5520	01/04/0096																																																	
104	MADHURI	CHANDI	1320	12/17/0095																																																	
105	PRAMOD	M.G.ROAD	3300	03/27/0096																																																	
108	KRANTI	NEHRUPLACE	5500	07/02/0095																																																	
109	MINU	POWAI	7700	08/10/0095																																																	

9 rows returned in 0.00 seconds [Download](#)

(5) Give 100 Rs more to all depositors if they are maximum depositors in their respective branch.

Ans:

```
1 UPDATE DEPOSIT SET AMOUNT=AMOUNT - 100 WHERE BNAME IN(SELECT BNAME FROM BRANCH GROUP BY BNAME  
2 HAVING COUNT(BNAME)=(SELECT MAX(COUNT(CNAME)) FROM DEPOSIT GROUP BY BNAME));
```

Results Explain Describe Saved SQL History

7 row(s) updated.

(6) Delete depositors of branches having number of customers between 1 to 3.

Ans:

```
1 DELETE FROM DEPOSIT WHERE BNAME IN(SELECT BNAME FROM DEPOSIT GROUP BY BNAME HAVING COUNT(BNAME)  
2 IN (SELECT COUNT(BNAME) FROM DEPOSIT GROUP BY BNAME HAVING COUNT(BNAME)>=1 AND COUNT(BNAME)<=3))
```

Results Explain Describe Saved SQL History

9 row(s) deleted.

(7) Delete deposit of vijay.

Ans:

```
1 DELETE FROM DEPOSIT WHERE CNAME = 'VIJAY';  
2
```

Results Explain Describe Saved SQL History

0 row(s) deleted.

(8) Delete borrower of branches having average loan less than 1000.

Ans:

```
1  DELETE FROM BORROW WHERE BNAME IN (SELECT BNAME FROM BORROW GROUP BY BNAME HAVING AVG(AMOUNT)<1000)
```

Results Explain Describe Saved SQL History

0 row(s) deleted.

Remarks: _____

Marks: _____

Signature: _____

Practical-9

Aim:Add and Remove constraint

Implementation

(1)Add primary key constraint on job_id in job table.

Ans:

```
1  ALTER TABLE JOB ADD CONSTRAINT ADDING_TO_JOB_ID PRIMARY KEY(JOB_ID);
```

Results	Explain	Describe	Saved SQL	History
Table altered.				

(2)Add foreign key constraint on employee table referencing job table.

Ans:

Object Type TABLE Object EMPLOYEE										
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment	
EMPLOYEE	EMP_NO	Number	-	3	0	-	✓	-	-	
	EMP_NAME	Varchar2	30	-	-	-	✓	-	-	
	EMP_SAL	Number	-	8	2	-	✓	-	-	
	EMP_COMM	Number	-	6	1	-	✓	-	-	
	DEPT_NO	Number	-	3	0	-	✓	-	-	
	L_NAME	Varchar2	30	-	-	-	✓	-	-	
	DEPT_NAME	Varchar2	30	-	-	-	✓	-	-	
	JOB_ID	Varchar2	15	-	-	-	✓	-	-	
	LOCATION	Varchar2	30	-	-	-	✓	-	-	
	MANAGER_ID	Number	-	5	0	-	✓	-	-	
	HIREDATE	Date	7	-	-	-	✓	-	-	
	PHONE	Number	-	10	0	-	✓	-	-	

1 - 12

(3)Add composite primary key on lock table (lock table does not exist, while creating table add composite key)

Ans:

```
1  CREATE TABLE VALID_LOCK( COLUMN1 NUMBER(3) , COLUMN2 NUMBER(3) , CONSTRAINT TWO_COLS_AS_PK PRIMARY KEY(COLUMN1,COLUMN2));
```

Results Explain Describe Saved SQL History

Table created.

(4)Remove primary key constraint on job_id

Ans:

Home > SQL > **SQL Commands**

Autocommit Display 10 ▾

```
ALTER TABLE JOB DROP PRIMARY KEY;
```

Results Explain Describe Saved SQL History

Table dropped.

(5)Remove foreign key constraint on employee table

Ans:

```
1  ALTER TABLE EMPLOYEE DROP CONSTRAINT ADDING_FK;
```

Results Explain Describe Saved SQL History

Table altered.

Remarks: _____

Marks: _____

Signature: _____