

Practical -1

AIM: Evaluation of Database (File System, DBMS, RDBMS, DDBMS).

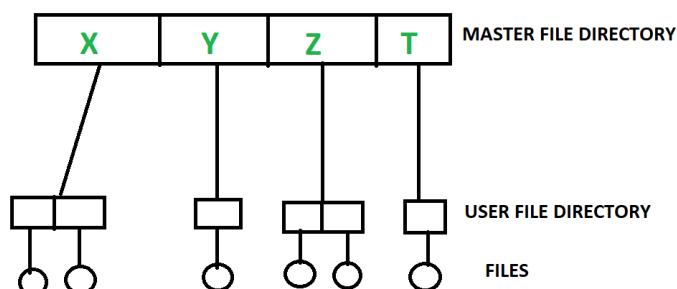
THEORY:

1. File System:

File system is basically a way of arranging the files in a storage medium like hard disk. File system organizes the files and helps in retrieval of files when they are required. File systems consists of different files which are grouped into directories. The directories further contain other folders and files. File system performs basic operations like management, file naming, giving access rules etc.

Example:

NTFS (New Technology File System), EXT(Extended File System).

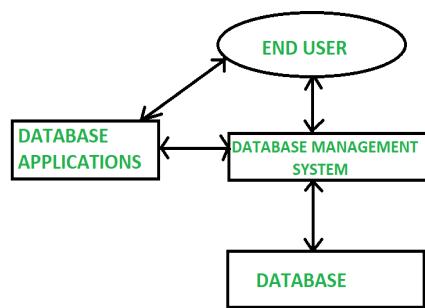


2. DBMS (Database Management System) :

Database Management System is basically a software that manages the collection of related data. It is used for storing data and retrieving the data effectively when it is needed. It also provides proper security measures for protecting the data from unauthorized access. In Database Management System the data can be fetched by SQL queries and relational algebra. It also provides mechanisms for data recovery and data backup.

Example:

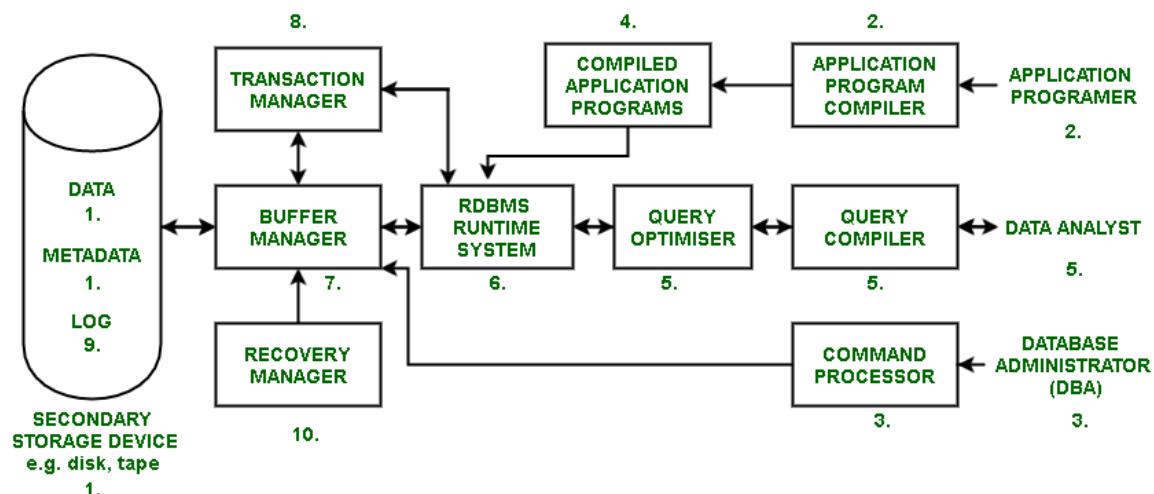
Oracle, MySQL, MS SQL server.



3. RDBMS:

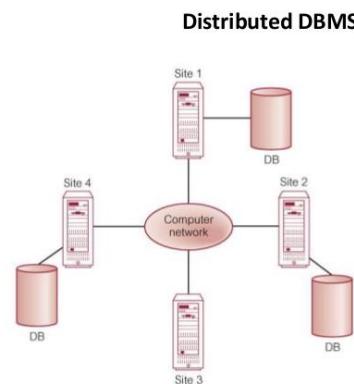
RDBMS stands for Relational Database Management System and it implements SQL. In the real-world scenario, people use the Relational Database Management System to collect information and process it, to provide service. E.g. In a ticket processing system, details about us (e.g. age, gender) and our journey (e.g. source, destination), are collected, and the ticket is provided to us.

RDBMS Architecture:



4. DDBMS

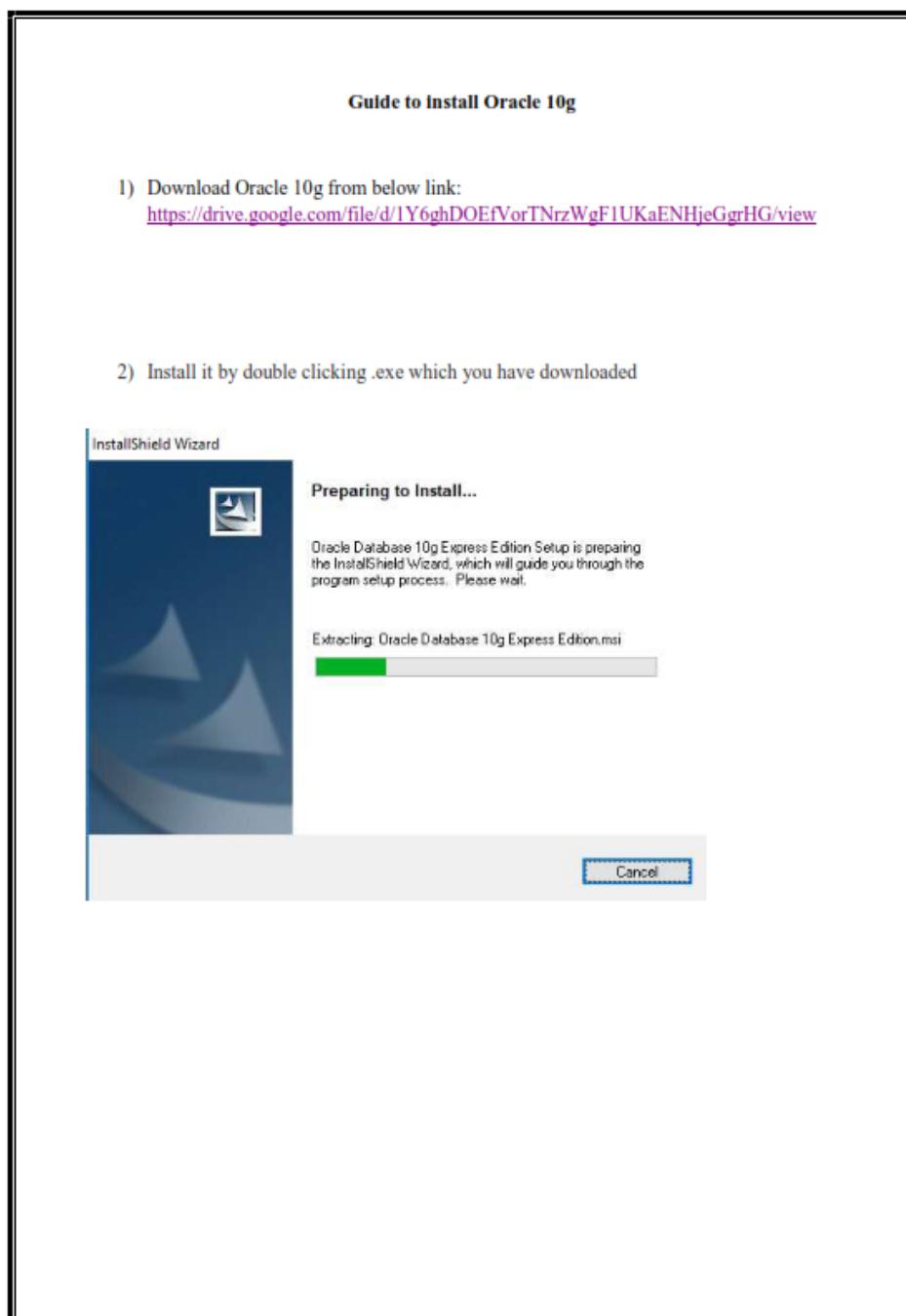
A distributed database is basically a database that is not limited to one system, it is spread over different sites, i.e., on multiple computers or over a network of computers. A distributed database system is located on various sites that don't share physical components. This may be required when a particular database needs to be accessed by various users globally. It needs to be managed such that for the users it looks like one single database.



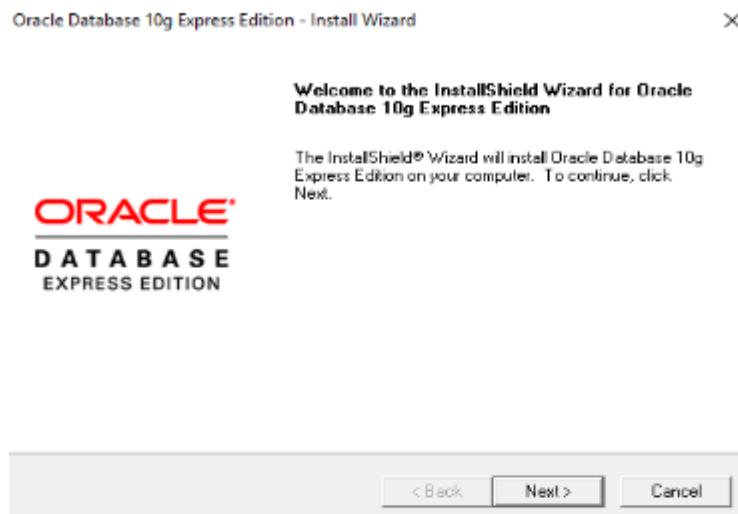
Practical -2

Aim: Introduction to Oracle (step by step installation, introduction of sql, plsql).

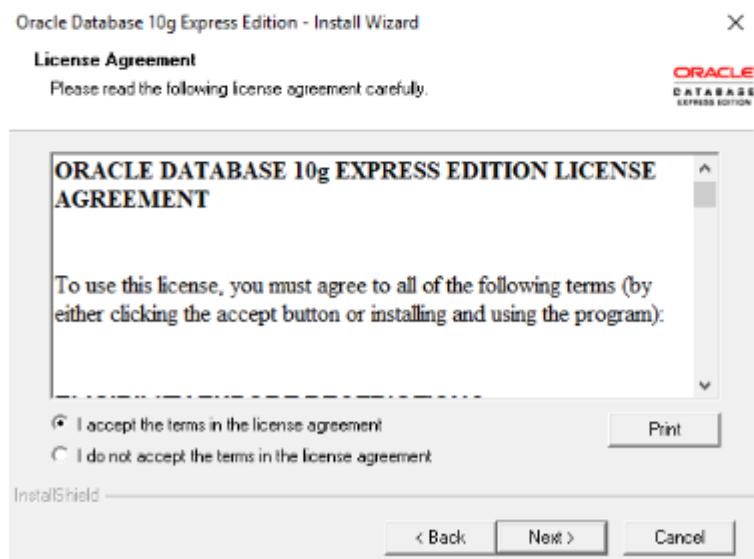
THEORY:



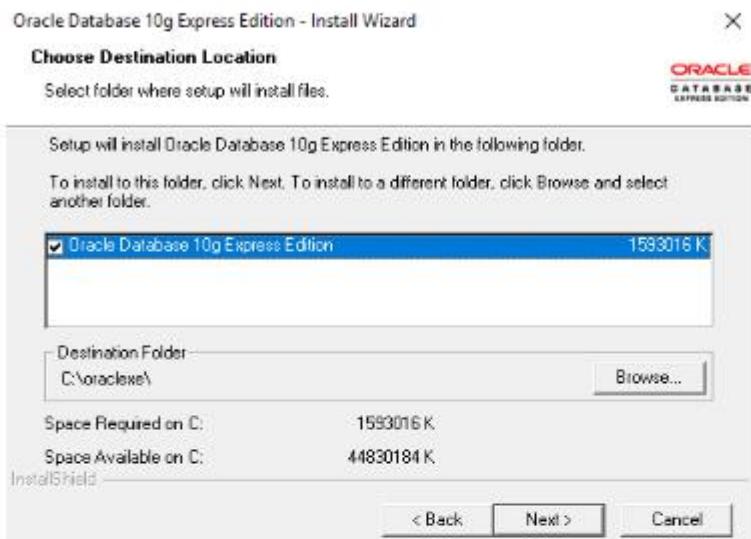
3) Click on Next button



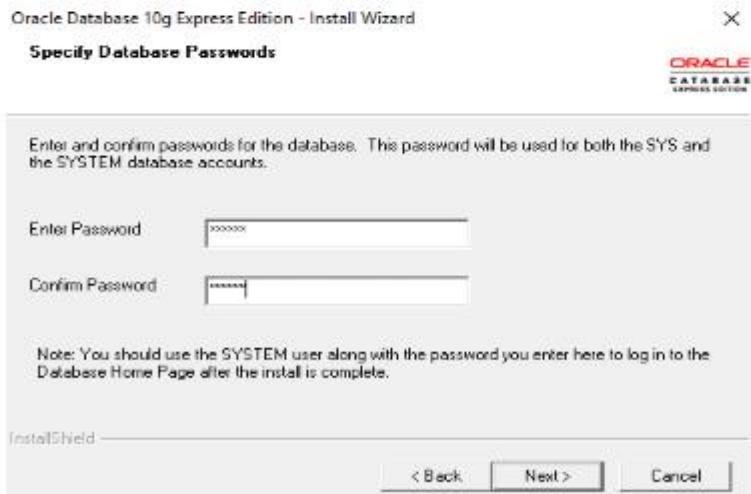
4) Accept license agreement and click on next button



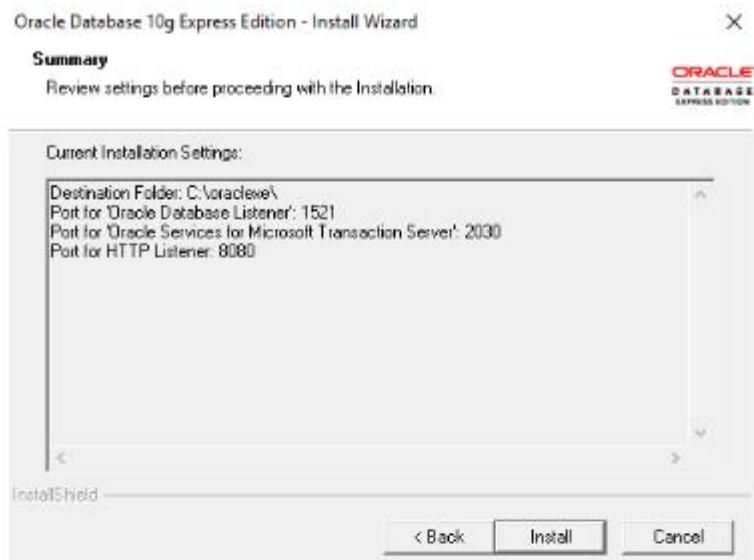
5) Click on next button



6) Enter password and confirm password for SYS and SYSTEM user. Please remember it because once installation will be over you have to enter it. To make it easy to remember give password as : "oracle"



7) Click on install button



8) Click on finish button.



9) Enter username as SYS OR SYSTEM and enter your password (Entered in step: 6)



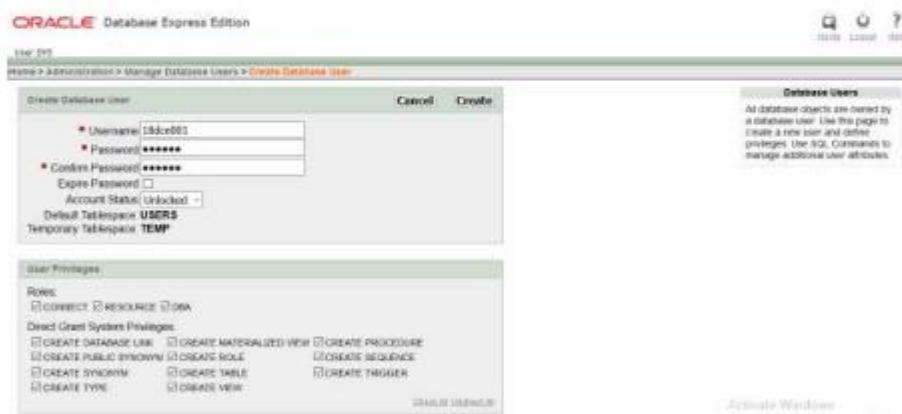
10) Click on Administration



11) Now click on “database user drop down button”. From that click on “create user”.



12) Enter your college roll no in username and give password (NEW) and confirm password. Don't check expire password, make account status unblocked if it is not. Give all privileges to your user. Finally click on “create” button.



13) This page will be shown to you. Now click on “logout” button.

The screenshot shows the Oracle Database Express Edition interface. The title bar reads "ORACLE® Database Express Edition". Below it, a navigation bar shows "User: 19DCE109" and "Home > Administration > Manage Database Users". A success message "User Created." with a green checkmark is displayed. There is a search bar labeled "Search Username" and a "Create" button. Below the search bar are two user icons: one for "19DCE109" and another for "hr". At the bottom right, there is a timestamp "1:3".

14) Click on login

The screenshot shows the Oracle Database Express Edition login page. The title bar reads "ORACLE® Database Express Edition". Below it, a green header bar displays the message "You are now logged out.". At the bottom left, there is a "Login" link.

15) Enter username and password that you just created and click on “login” button



16) Click on SQL



17) Click on SQL Commands



18) Congratulations!!! Now you are ready to code SQL and PLSQL.



19) Thank You!!!

Introduction

SQL:

Structured Query Language (SQL) is a standard Database language which is used to create, maintain and retrieve the relational database.

Introduction to PL/SQL:

PL/SQL is a block structured language that enables developers to combine the power of SQL with procedural statements. All the statements of a block are passed to oracle engine all at once which increases processing speed and decreases the traffic. PL/SQL stands for “Procedural language extensions to SQL.” PL/SQL is a database-oriented programming language that extends SQL with procedural capabilities. It was developed by Oracle Corporation within the early 90’s to boost the capabilities of SQL.

PL/SQL adds selective (i.e. if...then...else...) and iterative constructs (ie. loops) to SQL. PL/SQL is most helpful to put in writing triggers and keep procedures. Stored procedures square measure units of procedural code keep during a compiled type inside the info.

Conclusion: From this practical I learned how to install oracle & learned about sql & pl/sql.

Practical – 3

Aim: (i) To study DDL-CREATE and DML-INSERT Commands.

CREATE:

Syntax:

```
CREATE TABLE <TABLE_NAME> (
    FIELD_NAME <DATA_TYPE>,
);
```

Snapshot:

The screenshot shows a SQL command window with the following details:

- User: 19DCE109
- Navigation: Home > SQL > SQL Commands
- Autocommit: Checked
- Display: 10
- SQL Command:
CREATE TABLE TABLE1(BNAME VARCHAR2(18), CITY VARCHAR2(18));
- Bottom menu: Results (selected), Explain, Describe, Saved SQL, History

Table created.

0.53 seconds

INSERT:

Syntax:

```
INSERT INTO <TABLE_NAME>
VALUES ('val-1', 'val-2',.... );
```

Snapshot:

The screenshot shows a MySQL command-line interface. The title bar says "User: 19DCE109". Below it, the path "Home > SQL > SQL Commands" is visible. A dropdown menu shows "Autocommit" checked and "Display" set to 10. The main area contains the following SQL code:

```
INSERT INTO BORROW
VALUES (481, 'KRANTI', 'NEHRU PLACE',3000.00);
```

1 row(s) inserted.

0.05 seconds

From the above given tables perform the following queries:

- (1) Describe deposit, branch.

Snapshot:

User: 19DCE109
Home > SQL > SQL Commands

```
 Autocommit Display 10
  (ACTNO, CNAME, BNAME, AMOUNT, ADATE)
  Values
  (109, 'MINU', 'POWAI', 7000.00, '10-AUG-95')
  SELECT * FROM DUAL;

  SELECT * FROM DEPOSIT

  SELECT MAX(AMOUNT) FROM DEPOSIT
  WHERE BNAME='VRCE';

  SELECT COUNT(CNAME) FROM DEPOSIT;

  DELETE FROM DEPOSIT;

  DESC DEPOSIT;
```

Results Explain Describe Saved SQL History

Object Type TABLE Object DEPOSIT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DEPOSIT	ACTNO	Varchar2	5	-	-	-	✓	-	-
	CNAME	Varchar2	18	-	-	-	✓	-	-
	BNAME	Varchar2	18	-	-	-	✓	-	-
	AMOUNT	Number	-	8	2	-	✓	-	-
	ADATE	Date	7	-	-	-	✓	-	-

1 - 5

User: 19DCE109
Home > SQL > SQL Commands

```
 Autocommit Display 10
  INSERT ALL
  INTO BRANCH(BNAME, CITY) VALUES('VRCE', 'NAGPUR')
  INTO BRANCH(BNAME, CITY) VALUES('AJNI', 'NAGPUR')
  INTO BRANCH(BNAME, CITY) VALUES('KAROLBAGH', 'DELHI')
  INTO BRANCH(BNAME, CITY) VALUES('CHANDI', 'DELHI')
  INTO BRANCH(BNAME, CITY) VALUES('DHARAMPETH', 'NAGPUR')
  INTO BRANCH(BNAME, CITY) VALUES('M.G.ROAD', 'BANGLORE')
  INTO BRANCH(BNAME, CITY) VALUES('ANDHERI', 'BOMBAY')
  INTO BRANCH(BNAME, CITY) VALUES('VIRAR', 'BOMBAY')
  INTO BRANCH(BNAME, CITY) VALUES('NEHRU PLACE', 'DELHI')
  INTO BRANCH(BNAME, CITY) VALUES('POWAI', 'BOMBAY')
  SELECT * FROM DUAL;
  SELECT * FROM BRANCH;
  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	18	-	-	-	✓	-	-
	CITY	Varchar2	18	-	-	-	✓	-	-

1 - 2

(2) Describe borrow, customers.

Snapshot:

User: 19DCE109
Home > SQL > SQL Commands

Autocommit Display 10 ▾
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	-	-	-	✓	-	-
	CNAME	Varchar2	18	-	-	-	✓	-	-
	BNAME	Varchar2	18	-	-	-	✓	-	-
	AMOUNT	Number	-	8	2	-	✓	-	-

1 - 4

User: 19DCE109
Home > SQL > SQL Commands

Autocommit Display 10 ▾
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	19	-	-	-	✓	-	-
	CITY	Varchar2	18	-	-	-	✓	-	-

1 - 2

(3) List all data from table DEPOSIT.

Snapshot:

```

User: 19DCE109
Home > SQL > SQL Commands
 Autocommit Display 10
INSERT INTO DEPOSIT (ACTNO, CNAME, BNAME, AMOUNT, ADATE)
VALUES (109, 'INTNU', 'POVAI', 7000.00, '10-AUG-95')
SELECT * FROM DEPOSIT
SELECT MAX(AMOUNT) FROM DEPOSIT
WHERE BNAME='VRCE';
SELECT COUNT(CNAME) FROM DEPOSIT;
DELETE 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	KAROBAGH	3500	17-NOV-95
104	MADHURI	CHANDI	1200	17-DEC-95
105	PRAMOD	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	POVAI	7000	10-AUG-95

9 rows returned in 0.08 seconds [CSV Export](#)

(4) List all data from table BORROW.

Snapshot:

```

User: 19DCE109
Home > SQL > SQL Commands
 Autocommit Display 10
INSERT INTO BORROW(LOANNO,CNAME,BNAME,AMOUNT)VALUES(206,'MEHUL','AJNI',5000.00)
INSERT INTO BORROW(LOANNO,CNAME,BNAME,AMOUNT)VALUES(311,'SUNIL','DHARAMPETH',3000.00)
INSERT INTO BORROW(LOANNO,CNAME,BNAME,AMOUNT)VALUES(321,'MADHURI','ANDHERI',2000.00)
INSERT INTO BORROW(LOANNO,CNAME,BNAME,AMOUNT)VALUES(375,'PRAMOD','VRAR',8000.00)
INSERT INTO BORROW(LOANNO,CNAME,BNAME,AMOUNT)VALUES(481,'KRANTI','NEHRU PLACE',3000.00)

SELECT * FROM DUAL;
SELECT * FROM BORROW;

```

Results Explain Describe Saved SQL History

LOANNO	CNAME	BNAME	AMOUNT
481	KRANTI	NEHRU PLACE	3000
201	ANIL	VRCE	1000
206	MEHUL	AJNI	5000
311	SUNIL	DHARAMPETH	3000
321	MADHURI	ANDHERI	2000
375	PRAMOD	VRAR	8000
481	KRANTI	NEHRU PLACE	3000

7 rows returned in 0.03 seconds [CSV Export](#)

Language: en-us

(5) List all data from table CUSTOMERS.

Snapshot:

User: 19DCE109
 Home > SQL > SQL Commands
 Autocommit Display 10 ▾

```
INTO CUSTOMERS(CNAME,CITY)VALUES('MANDHURI','NAGPUR')
INTO CUSTOMERS(CNAME,CITY)VALUES('PRAHOD','NAGPUR')
INTO CUSTOMERS(CNAME,CITY)VALUES('SANDIP','SURAT')
INTO CUSTOMERS(CNAME,CITY)VALUES('SHIVANI','BOMBAY')
INTO CUSTOMERS(CNAME,CITY)VALUES('KRANTI','BOMBAY')
INTO CUSTOMERS(CNAME,CITY)VALUES('NAREN','BOMBAY')
SELECT * FROM DUAL;
SELECT * FROM CUSTOMERS;
```


 Results Explain Describe Saved SQL History

CNAME	CITY
ANIL	CALCUTTA
SUNIL	DELHI
MEHUL	BARODA
MANDAR	PATNA
MANDHURI	NAGPUR
PRAMOD	NAGPUR
SANDIP	SURAT
SHIVANI	BOMBAY
KRANTI	BOMBAY
NAREN	BOMBAY

 10 rows returned in 0.07 seconds CSV Export

(6) List all data from table BRANCH.

Snapshot:

User: 19DCE109
 Home > SQL > SQL Commands
 Autocommit Display 10 ▾

```
INSERT ALL
INTO BRANCH(BNAME, CITY) VALUES('VRCE', 'NAGPUR')
INTO BRANCH(BNAME, CITY) VALUES('AJNI', 'NAGPUR')
INTO BRANCH(BNAME, CITY) VALUES('KAROLBAUGH', 'DELHI')
INTO BRANCH(BNAME, CITY) VALUES('CHANDI', 'DELHI')
INTO BRANCH(BNAME, CITY) VALUES('DHARAMPETH', 'NAGPUR')
INTO BRANCH(BNAME, CITY) VALUES('M.G.ROAD', 'BANGLORE')
INTO BRANCH(BNAME, CITY) VALUES('ANDHERI', 'BOMBAY')
INTO BRANCH(BNAME, CITY) VALUES('VIRAR', 'BOMBAY')
INTO BRANCH(BNAME, CITY) VALUES('NEHRU PLACE', 'DELHI')
INTO BRANCH(BNAME, CITY) VALUES('POWAI', 'BOMBAY')
SELECT * FROM DUAL;
SELECT * FROM BRANCH;
```


 Results Explain Describe Saved SQL History

BNAME	CITY
VRCE	NAGPUR
AJNI	NAGPUR
KAROLBAUGH	DELHI
CHANDI	DELHI
DHARAMPETH	NAGPUR
M.G.ROAD	BANGLORE
ANDHERI	BOMBAY
VIRAR	BOMBAY
NEHRU PLACE	DELHI
POWAI	BOMBAY

 10 rows returned in 0.03 seconds CSV Export

(7) Give account no and amount of depositors.

Snapshot:

User: 19DCE109
 Home > SQL > SQL Commands

```

 Autocommit Display 10
INTO DEPOSIT
  (ACTNO, CNAME, BNAME, AMOUNT, ADATE)
Values
  (109, 'MINU', 'POWAI', 7000.00, '10-AUG-95')
SELECT * FROM DUAL;

SELECT * FROM DEPOSIT

SELECT MAX(AMOUNT) FROM DEPOSIT
WHERE BNAME='VRCE';

SELECT COUNT(CNAME) FROM DEPOSIT;

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

Results Explain Describe Saved SQL History

ACTNO	AMOUNT
100	1000
101	5000
102	3500
104	1200
105	3000
106	2000
107	1000
108	5000
109	7000

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

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

Snapshot:

User: 19DCE109
 Home > SQL > SQL Commands

```

 Autocommit Display 10
  (109, MINU, POWAI, 7000.00, 10-AUG-95 )
SELECT * FROM DUAL;

SELECT * FROM DEPOSIT

SELECT MAX(AMOUNT) FROM DEPOSIT
WHERE BNAME='VRCE';

SELECT COUNT(CNAME) FROM DEPOSIT;

DELETE FROM DEPOSIT;
SELECT ACTNO,AMOUNT FROM DEPOSIT;

SELECT CNAME FROM DEPOSIT
WHERE AMOUNT > 4000.00
  
```

Results Explain Describe Saved SQL History

CNAME
SUNIL
KRANTI
MINU

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

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

Snapshot:

User: 19DCE109

Home > SQL > SQL Commands

Autocommit Display 10

```
SELECT * FROM DEPOSIT
SELECT MAX(AMOUNT) FROM DEPOSIT
WHERE BNAME= 'VRCE';

SELECT COUNT(CNAME) FROM DEPOSIT;

DELETE FROM DEPOSIT;
SELECT ACTNO,AMOUNT FROM DEPOSIT;

SELECT CNAME FROM DEPOSIT
WHERE AMOUNT > 4000.00

SELECT CNAME FROM DEPOSIT
WHERE ADATE >'1-DEC-96'
```

Results Explain Describe Saved SQL History

no data found

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

Snapshot:

User: 19DCE109

Home > SQL > SQL Commands

Autocommit Display 10

```
INTO BRANCH(BNAME, CITY) VALUES('VRCE', 'NAGPUR')
INTO BRANCH(BNAME, CITY) VALUES('AJNI', 'NAGPUR')
INTO BRANCH(BNAME, CITY) VALUES('KAROLBAGH', 'DELHI')
INTO BRANCH(BNAME, CITY) VALUES('CHANDI', 'DELHI')
INTO BRANCH(BNAME, CITY) VALUES('DHARAMPETH', 'NAGPUR')
INTO BRANCH(BNAME, CITY) VALUES('M.G.ROAD', 'BANGLORE')
INTO BRANCH(BNAME, CITY) VALUES('ANDHERI', 'BOMBAY')
INTO BRANCH(BNAME, CITY) VALUES('VIRAR', 'BOMBAY')
INTO BRANCH(BNAME, CITY) VALUES('NEHRU PLACE', 'DELHI')
INTO BRANCH(BNAME, CITY) VALUES('POWAI', 'BOMBAY')
SELECT * FROM DUAL;
SELECT * FROM BRANCH;
```

**SELECT CITY FROM BRANCH
WHERE BNAME = 'KAROLBAGH'**

Results Explain Describe Saved SQL History

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.

Snapshot:

User: 19DCE109
 Home > SQL > SQL Commands

```

 Autocommit Display 10
  values
  (109, 'MINU', 'POWAI', 7000.00, '10-AUG-95')
  SELECT * FROM DUAL;

  SELECT * FROM DEPOSIT

  SELECT MAX(AMOUNT) FROM DEPOSIT
  WHERE BNAME='VRCE';

  SELECT COUNT(CNAME) FROM DEPOSIT;

  DELETE FROM DEPOSIT;

  SELECT ACTNO FROM DEPOSIT
  WHERE ADATE BETWEEN '1-DEC-96' AND '1-JUN-96';

```

Results Explain Describe Saved SQL History

no data found

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

Snapshot:

User: 19DCE109
 Home > SQL > SQL Commands

```

 Autocommit Display 10
  SELECT * FROM DEPOSIT

  SELECT MAX(AMOUNT) FROM DEPOSIT
  WHERE BNAME='VRCE';

  SELECT COUNT(CNAME) FROM DEPOSIT;

  DELETE FROM DEPOSIT;

  SELECT ACTNO FROM DEPOSIT
  WHERE ADATE BETWEEN '1-DEC-96' AND '1-JUN-96';

  SELECT CNAME FROM DEPOSIT
  WHERE BNAME = 'VRCE';

```

Results Explain Describe Saved SQL History

CNAME
ANIL

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

Conclusion: We Learned how to create tables, insert data into tables & read from the table as any required query.

Practical -4

Perform following queries

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

Snap-Shot:

User: 19DCE109
Home > SQL > SQL Commands

Autocommit Display [10]

```

    (a_no, cname, bname, amount, a_date)
    Values
    (104, 'vijay', 'andheri', 8000, '17-sep-06')
INTO DEPOSIT4
    (a_no, cname, bname, amount, a_date)
    Values
    (105, 'keyur', 'dadar', 7500, '19-nov-06')
INTO DEPOSIT4
    (a_no, cname, bname, amount, a_date)
    Values
    (106, 'mayur', 'borvali', 5500, '21-dec-06')
SELECT * FROM DUAL;

SELECT * FROM DEPOSIT4
  
```

Results Explain Describe Saved SQL History

A_NO	CNAME	BNAME	AMOUNT	A_DATE
101	Anil	andheri	7000	01-JAN-06
102	sunil	virar	5000	15-JUL-06
103	jay	villeparle	6500	12-MAR-06
104	vijay	andheri	8000	17-SEP-06
105	keyur	dadar	7500	19-NOV-06
106	mayur	borvali	5500	21-DEC-06

6 rows returned in 0.04 seconds [CSV Export](#)

User: 19DCE109
Home > SQL > SQL Commands

Autocommit Display [10]

```

Create Table Job (job_id Varchar(15), job_title Varchar(30), min_sal Number(7,2), max_sal Number(7,2))
INSERT INTO Job(job_id, job_title, min_sal, max_sal) VALUES('it_prog', 'Programmer', 4000, 10000)
INTO Job(job_id, job_title, min_sal, max_sal) VALUES('mk_mngr', 'Marketing manager', 9000, 15000)
INTO Job(job_id, job_title, min_sal, max_sal) VALUES('fin_mngr', 'Finance manager', 8200, 12000)
INTO Job(job_id, job_title, min_sal, max_sal) VALUES('fi_acc', 'Account', 4200, 9000)
INTO Job(job_id, job_title, min_sal, max_sal) VALUES('lec', 'Lecturer', 6000, 17000)
INTO Job(job_id, job_title, min_sal, max_sal) VALUES('comp_op', 'Computer Operator', 1500, 3000)
SELECT * FROM DUAL;
  
```

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
fin_mngr	Finance manager	8200	12000
fi_acc	Account	4200	9000
lec	Lecturer	6000	17000
comp_op	Computer Operator	1500	3000

6 rows returned in 0.19 seconds [CSV Export](#)

User: 19DCE109
Home > SQL > SQL Commands

Autocommit Display [10]

```

INSERT Employee(emp_no, emp_name, emp_sal, emp_com, dept_no, l_name, dept_name, job_id, location, manager_id, hiredate)VALUES(103,'Adena',1100,0,20,'shel','Machine Learning','it_prog', 'boston', 105, '29-AUG-96')
INSERT Employee(emp_no, emp_name, emp_sal, emp_com, dept_no, l_name, dept_name, job_id, location, manager_id, hiredate)VALUES(104,'Shenka',1500,0,20,'shab','Machine Learning','it_prog', 'boston', 105, '30-NOV-95')
INSERT Employee(emp_no, emp_name, emp_sal, emp_com, dept_no, l_name, dept_name, job_id, location, manager_id, hiredate)VALUES(105,'Aman',3000,0,15,'shama','Virtual Reality','comp-op', 'mumbai', 106, '02-OCT-07')
INSERT Employee(emp_no, emp_name, emp_sal, emp_com, dept_no, l_name, dept_name, job_id, location, manager_id, hiredate)VALUES(106,'Anita',2450,0,10,'anita','Big Data Analytics','comp-op', 'mumbai', 107, '14-JUN-08')
INSERT Employee(emp_no, emp_name, emp_sal, emp_com, dept_no, l_name, dept_name, job_id, location, manager_id, hiredate)VALUES(107,'Aneka',2975,0,30,'jha','Artificial Intelligence','it_prog', 'new york', '15-Jul-97')
SELECT * FROM DUAL;
  
```

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	Smith	600	20	20	shel	Machine Learning	it_prog	boston	105	29-AUG-96
102	Snela	1000	300	25	gupta	Data Science	it_prog	boston	-	144600.00
103	Adena	1100	500	20	shab	Machine Learning	it_prog	ontario	105	30-NOV-95
104	Aman	3000	-	15	shama	Virtual Reality	comp-op	mumbai	106	02-OCT-07
105	Anita	2450	5000	10	anita	Big Data Analytics	comp-op	mumbai	107	14-JUN-08
106	Sneha	2450	24500	10	jha	Big Data Analytics	it_prog	mumbai	105	24-SEP-07
107	Ananya	2975	-	30	jha	Artificial Intelligence	it_prog	new york	-	15-JUL-97

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

Snap-Shot:

User: 19DCE109
Home > SQL > SQL Commands

```

 Autocommit Display 10 

    INSERT INTO Employee(emp_no, emp_name, emp_sal, emp_comm, dept_no, l_name, dept_name, job_id, location, manager_id, hiredate) VALUES(106, 'Sneha', 2450, 24500, 20, 'joseph', 'big data analytics', 'fi_acc', 'melbourne', 105, '26-sep-97')
    INSERT INTO Employee(emp_no, emp_name, emp_sal, dept_no, l_name, dept_name, job_id, location, hiredate) VALUES(107, 'Anamika', 2975, 30, 'jha', 'artificial intelligence', 'it_prog', 'new york', '15-jul-97')

    SELECT * FROM DUAL;
    SELECT * FROM EMPLOYEE;

    SELECT A.NO,AMOUNT
    FROM DEPOSITA4
    WHERE A.DATE BETWEEN '01-JAN-06' AND '25-JUN-06';
  

```

Results Explain Describe Saved SQL History

A_NO	AMOUNT
101	7000
103	6500

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

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

Snap-Shot:

General (4CE-2 : AY-2020-21(EVE)) | 23 Arrays vs Linked List | Data < SQL Commands

User: 19DCE109
Home > SQL > SQL Commands

```

 Autocommit Display 10  

    SELECT * FROM JOB WHERE MIN_SAL >4000
  

```

Results Explain Describe Saved SQL History

JOB_ID	JOB_TITLE	MIN_SAL	MAX_SAL
mk_mngr	Marketing manager	9000	15000
fi_mngr	Finance manager	8200	12000
fi_acc	Account	4200	9000
lec	Lecturer	6000	17000

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

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2008, Oracle. All rights reserved.

Type here to search 21:54 ENG 16-04-2021

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

Snap-Shot:

```

SELECT * FROM JOB WHERE MIN_SAL >4000
SELECT EMP_NAME AS NAME,EMP_SAL FROM EMPLOYEE WHERE DEPT_NO = 26
  
```

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).

Snap-Shot:

```

SELECT EMP_NO,EMP_NAME,DEPT_NAME FROM EMPLOYEE WHERE DEPT_NO BETWEEN 10 AND 20
  
```

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

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

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

Snap-Shot:

```
SELECT * FROM EMPLOYEE WHERE MANAGER_ID IS NOT NULL AND EMP_COMM IS NOT NULL
```

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	LN_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE
101	Smith	800	500	20	shah	machine learning	ft_mngr	toronto	105	09-AUG-96
103	Adama	1100	500	20	wales	machine learning	mk_mngr	ontario	105	30-NOV-95
105	Anita	5000	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-98
106	Shreya	2450	24500	10	joseph	big data analytics	ft_acc	melbourne	105	26-SEP-97

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

(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.

Snap-Shot:

```
SELECT (A_NO || ' ' || CNAME) AS REQUIRED_TABLE FROM DEPOSIT4 WHERE AMOUNT != 8000
```

REQUIRED_TABLE
101 Anil
102 sunil
103 jay
105 keyur
106 mayur

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

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

Snap-Shot:

The screenshot shows a browser window for Oracle Database Express Edition. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:936863846923326::NO:::. The page title is "SQL Commands". The SQL query entered is:

```
SELECT * FROM JOB WHERE MIN_SAL = 2000 OR MIN_SAL = 4000
```

The results show one row:

JOB_ID	JOB_TITLE	MIN_SAL	MAX_SAL
it_prog	Programmer	4000	10000

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

At the bottom, it says Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

To study various options of LIKE predicate

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

Snap-Shot:

The screenshot shows a browser window for Oracle Database Express Edition. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:936863846923326::NO:::. The page title is "SQL Commands". The SQL query entered is:

```
SELECT * FROM EMPLOYEE WHERE EMP_NAME LIKE 'A_a%'
```

The results show three rows:

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE
103	Adama	1100	500	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
107	Anamika	2975	-	30	jha	artificial intelligence	it_prog	new york	-	15-JUL-97

3 rows returned In 0.00 seconds [CSV Export](#)

At the bottom, it says Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

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

Snap-Shot:

User: 19DCE109
Home > SQL > SQL Commands
SELECT EMP_NAME, EMP_NO,EMP_SAL FROM EMPLOYEE WHERE EMP_NAME LIKE 'Ani__'.

EMP_NAME	EMP_NO	EMP_SAL
Anita	105	5000

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

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2008, Oracle. All rights reserved.

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

Snap-Shot:

User: 19DCE109
Home > SQL > SQL Commands
SELECT * FROM EMPLOYEE WHERE EMP_NAME LIKE '_m%' OR EMP_NAME LIKE '_n%'

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE
101	Smith	800	500	20	shah	machine learning	ft_mgr	Toronto	105	09-AUG-96
102	Snehal	1600	300	25	gupta	data science	lec	las vegas	-	14-MAR-96
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	ft_acc	melbourne	105	26-SEP-97
107	Anamika	2975	-	30	jha	artificial intelligence	it_prog	new york	-	15-JUL-97

6 rows returned in 0.03 seconds [CSV Export](#)

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2008, Oracle. All rights reserved.

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

Snap-Shot:

General (4CE-2 : AY2020-21(EVE) x 2.3 Arrays vs Linked List | Data st: x SQL Commands x +

127.0.0.1:8080/apex/f?p=4500:1003:936863846923326::NO::

ORACLE® Database Express Edition

User: 19DCE109

Home > SQL > SQL Commands

Autocommit

```
SELECT CNAME FROM DEPOSITO WHERE BNAME = 'andheri' OR BNAME = 'dadar' OR BNAME = 'vircat'
```

Results Explain Describe Saved SQL History

CNAME
Anil
sunil
vijay
keyur

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

Application Express 2.1.0.00.39
Copyright © 1999, 2009, Oracle. All rights reserved.

Type here to search 22:13 ENG 16-04-2021

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

Snap-Shot:

General (4CE-2 : AY2020-21(EVE) x 2.3 Arrays vs Linked List | Data st: x SQL Commands x +

127.0.0.1:8080/apex/f?p=4500:1003:936863846923326::NO::

ORACLE® Database Express Edition

User: 19DCE109

Home > SQL > SQL Commands

Autocommit

```
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](#)

Application Express 2.1.0.00.39
Copyright © 1999, 2008, Oracle. All rights reserved.

Type here to search 22:43 ENG 16-04-2021

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

Snap-Shot:

The screenshot shows a browser window for Oracle Database Express Edition. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:936863846923326::NO:::. The page title is "SQL Commands". A SQL query is entered in the command field:

```
SELECT JOB_TITLE FROM JOB WHERE JOB_ID LIKE '%mgr' AND MAX_SAL > 12000
```

The results show one row:

JOB_TITLE
Marketing manager

1 rows returned in 0.00 seconds. There is a CSV Export link.

At the bottom, it says "Language: en-us" and "Application Express 2.1.0.00.39 Copyright © 1999, 2008, Oracle. All rights reserved."

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

Snap-Shot:

The screenshot shows a browser window for Oracle Database Express Edition. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:936863846923326::NO:::. The page title is "SQL Commands". A SQL query is entered in the command field:

```
SELECT * FROM EMPLOYEE WHERE EMP_COMM IS NOT NULL AND MANAGER_ID IS NOT NULL AND EMP_NAME LIKE '_n___'
```

The results show two rows:

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
106	Sneha	2450	24500	10	joseph	big data analytics	fi_acc	melbourne	105	26-SEP-97

2 rows returned in 0.00 seconds. There is a CSV Export link.

At the bottom, it says "Language: en-us" and "Application Express 2.1.0.00.39 Copyright © 1999, 2008, Oracle. All rights reserved."

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

Snap-Shot:

The screenshot shows a browser window for Oracle Database Express Edition. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:936863846923326::NO:::. The page displays a SQL command in the SQL Commands tab:

```
SELECT * FROM EMPLOYEE WHERE EMP_COMM IS NULL AND MANAGER_ID IS NULL AND EMP_NAME LIKE '__a%'
```

The results show one row:

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE
107	Anamika	2975	-	30	jha	artificial intelligence	it_prog	newyork	-	15-JUL-97

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

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

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

Snap-Shot:

The screenshot shows a browser window for Oracle Database Express Edition. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:936863846923326::NO:::. The page displays a SQL command in the SQL Commands tab:

```
SELECT * FROM JOB WHERE JOB_ID LIKE '%\_%' ESCAPE '\';
```

The results show five rows:

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 0.00 seconds [CSV Export](#)

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

Conclusion: We Learned how to use Like Predicate & perform specific queries.

Practical -5

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

- (1) List total deposit from deposit.

Snap-Shot:

The screenshot shows a browser window for Oracle Database Express Edition. The URL is 127.0.0.1:8080/apex/?p=4500:1003:936863846923326:N0:::.

The SQL Commands page displays the following SQL query:

```
SELECT SUM(AMOUNT) FROM DEPOSIT
```

The results section shows the output:

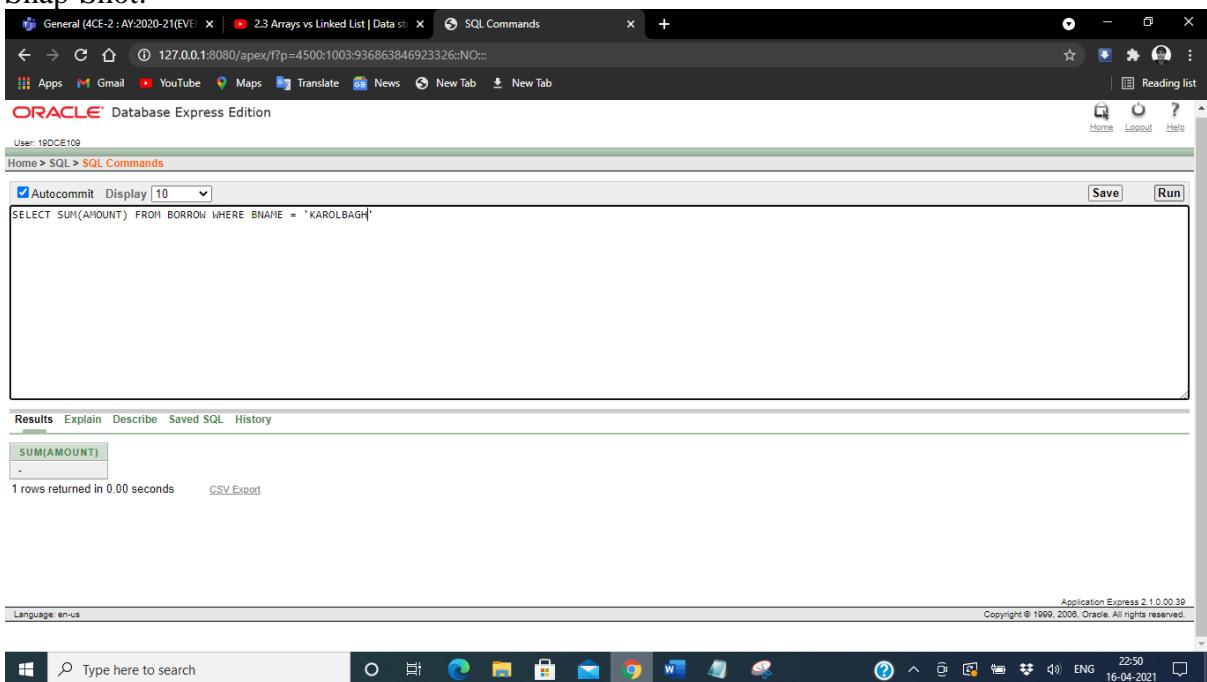
SUM(AMOUNT)
28700

1 rows returned in 0.00 seconds

At the bottom, the status bar indicates Application Express 2.1.0.00.39 and Copyright © 1999, 2008, Oracle. All rights reserved.

(2) List total loan from karolbagh branch

Snap-Shot:



The screenshot shows a browser window for Oracle Database Express Edition. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:936863846923326::NO:: . The page title is "General (4CE-2 : AY2020-21(EVE)) | 2.3 Arrays vs Linked List | Data st | SQL Commands". The SQL command entered is:

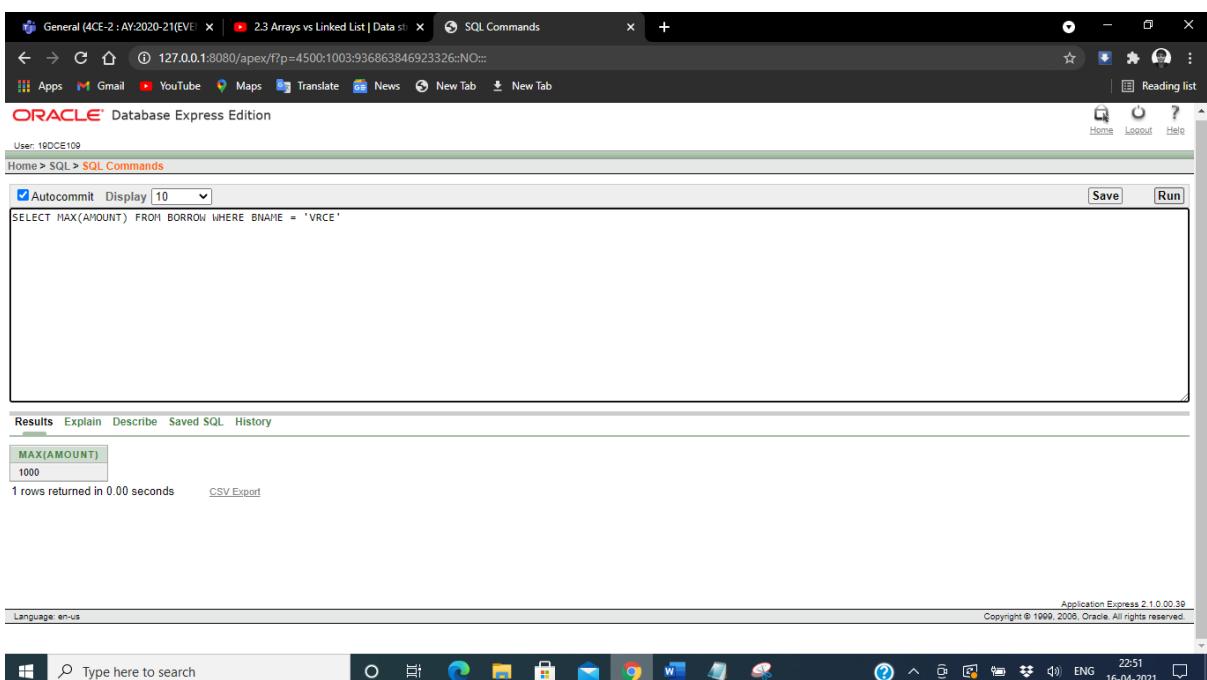
```
SELECT SUM(AMOUNT) FROM BORROW WHERE BNAME = 'KAROLBAGH';
```

The results section shows a single row with the value 0.00 under the column "SUM(AMOUNT)". Below the results, it says "1 rows returned in 0.00 seconds".

At the bottom, the Application Express version is listed as "Application Express 2.1.0.00.39 Copyright © 1999, 2008, Oracle. All rights reserved."

(3) Give maximum loan from branch vrce.

Snap-Shot:



The screenshot shows a browser window for Oracle Database Express Edition. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:936863846923326::NO:: . The page title is "General (4CE-2 : AY2020-21(EVE)) | 2.3 Arrays vs Linked List | Data st | SQL Commands". The SQL command entered is:

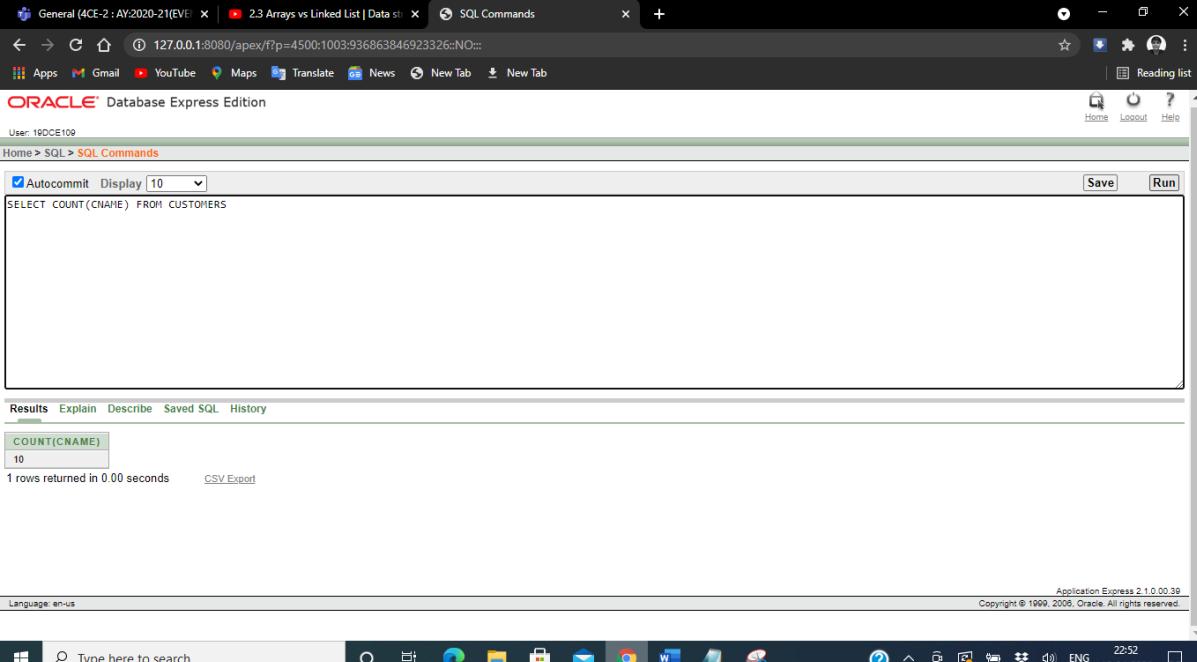
```
SELECT MAX(AMOUNT) FROM BORROW WHERE BNAME = 'VRCE';
```

The results section shows a single row with the value 1000 under the column "MAX(AMOUNT)". Below the results, it says "1 rows returned in 0.00 seconds".

At the bottom, the Application Express version is listed as "Application Express 2.1.0.00.39 Copyright © 1999, 2008, Oracle. All rights reserved."

(4) Count total number of customers

Snap-Shot:



The screenshot shows the Oracle Database Express Edition interface. A SQL command is entered in the query editor:

```
SELECT COUNT(CNAME) FROM CUSTOMERS
```

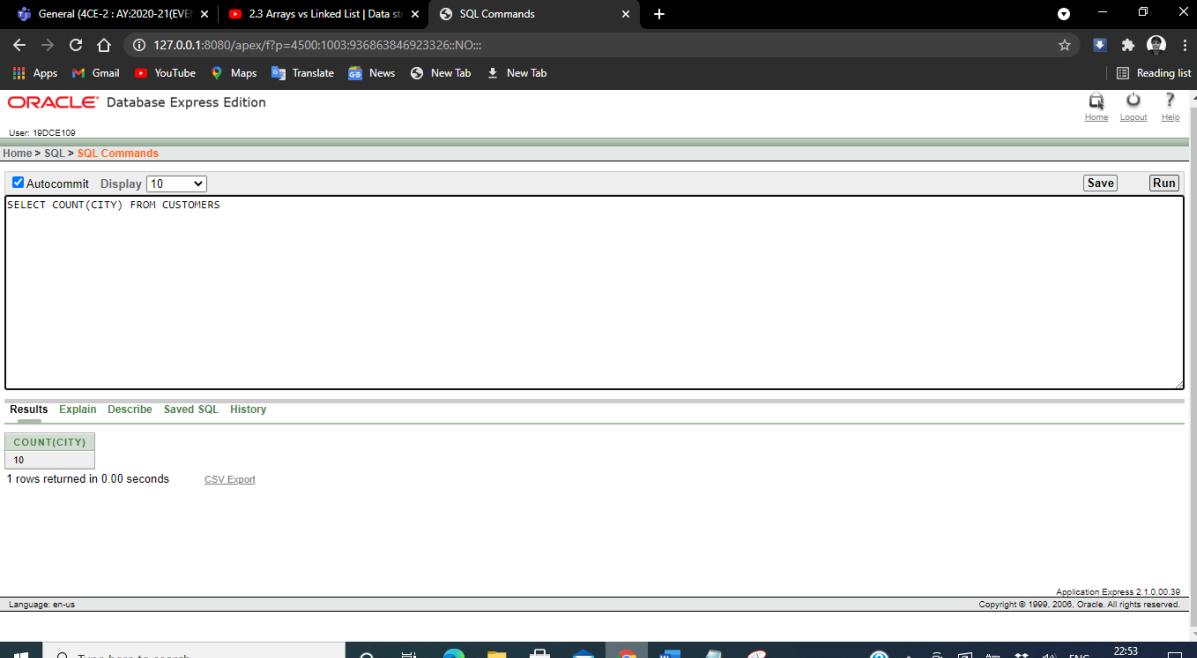
The results pane displays the output:

COUNT(CNAME)
10

Below the results, it says "1 rows returned in 0.00 seconds". The status bar at the bottom right shows "Application Express 2.1.0.00.39 Copyright © 1999, 2008, Oracle. All rights reserved." and the date "16-04-2021".

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

Snap-Shot:



The screenshot shows the Oracle Database Express Edition interface. A SQL command is entered in the query editor:

```
SELECT COUNT(CITY) FROM CUSTOMERS
```

The results pane displays the output:

COUNT(CITY)
10

Below the results, it says "1 rows returned in 0.00 seconds". The status bar at the bottom right shows "Application Express 2.1.0.00.39 Copyright © 1999, 2008, Oracle. All rights reserved." and the date "16-04-2021".

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

Snap-Shot:

```

General (4CE-2 : AY2020-21(EVE) | 2.3 Arrays vs Linked List | Data st | SQL Commands
User: 19DCE109
Home > SQL > SQL Commands
Autocommit Display 10 Save Run
CREATE TABLE SUPPLIER AS (SELECT * FROM EMPLOYEE)
SELECT * FROM SUPPLIER

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 800 500 20 shah machine learning ft_mgr toronto 105 09-AUG-96
102 Snehal 1600 300 25 gupta data science lec las vegas - 14-MAR-96
103 Adama 1100 500 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 ft_aer melbourne 105 26-SEP-97
107 Anamika 2975 - 30 jha artificial intelligence ft_prog new york - 15-JUL-97
7 rows returned in 0.06 seconds CSV Export
Application Express 2.1.0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.
Language: en-us
Type here to search

```

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

Snap-Shot:

```

General (4CE-2 : AY2020-21(EVE) | 2.3 Arrays vs Linked List | Data st | SQL Commands
User: 19DCE109
Home > SQL > SQL Commands
Autocommit Display 10 Save Run
CREATE TABLE SUP1 AS (SELECT EMP_NO,EMP_NAME FROM EMPLOYEE)
SELECT * FROM SUP1

Results Explain Describe Saved SQL History
EMP_NO EMP_NAME
101 Smith
102 Snehal
103 Adama
104 Aman
105 Anita
106 Sneha
107 Anamika
7 rows returned in 0.02 seconds CSV Export
Application Express 2.1.0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.
Language: en-us
Type here to search

```

(8) Create table sup2 from employee with no data

Snap-Shot:

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The user is executing the following SQL code:

```
CREATE TABLE SUP2 AS (SELECT * FROM EMPLOYEE WHERE EMP_NO = 10000);
DESC SUP2;
SELECT * FROM EMPLOYEE;
```

The results pane shows the message "Table created." and a execution time of "0.05 seconds". The status bar at the bottom indicates "Application Express 2.1.0.00.39" and "Copyright © 1999, 2000, Oracle. All rights reserved."

(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.

Snap-Shot :

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The user is executing the following SQL code:

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

The results pane displays the data inserted into SUP2:

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
106	Sneha	2450	24500	10	joseph	big data analytics	fi_acc	melbourne	105	28-SEP-97

The status bar at the bottom indicates "Application Express 2.1.0.00.39" and "Copyright © 1999, 2000, Oracle. All rights reserved."

(10) Delete all the rows from sup1.

Snap-Shot:

The screenshot shows a browser window for Oracle Database Express Edition. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:936863846923326::NO:::. The page title is "SQL Commands". The SQL query entered is "DELETE FROM SUP1". The results section shows "7 row(s) deleted." and "0.04 seconds". The status bar at the bottom right indicates "Application Express 2.1.0.00.39" and "Copyright © 1999, 2008, Oracle. All rights reserved.". The taskbar at the bottom shows various application icons.

```
DELETE FROM SUP1
```

7 row(s) deleted.
0.04 seconds

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

Snap-Shot:

The screenshot shows a browser window for Oracle Database Express Edition. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:936863846923326::NO:::. The page title is "SQL Commands". The SQL query entered is "DELETE FROM SUPPLIER WHERE EMP_NO = 103". The results section shows "1 row(s) deleted." and "0.01 seconds". The status bar at the bottom right indicates "Application Express 2.1.0.00.39" and "Copyright © 1999, 2008, Oracle. All rights reserved.". The taskbar at the bottom shows various application icons.

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

1 row(s) deleted.
0.01 seconds

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The user is connected as '19DCE109'. In the SQL editor, the following commands are entered:

```
DELETE FROM SUPPLIER WHERE EMP_NO = 103
SELECT * FROM SUPPLIER
```

The results section displays a table with 6 rows of data from the SUPPLIER table:

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE
101	Smith	800	500	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
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

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

Application Express 2.1.0.0.39
Copyright © 1999, 2008, Oracle. All rights reserved.

(12) Rename the table sup2.

Snap-Shot:

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The user is connected as '19DCE109'. In the SQL editor, the following command is entered:

```
ALTER TABLE SUP2 RENAME TO SUPPLIER
```

The results section shows the message 'Table altered.' and a timestamp '0.11 seconds'.

Application Express 2.1.0.0.39
Copyright © 1999, 2008, Oracle. All rights reserved.

(13) Destroy table sup1 with all the data.

Snap-Shot:

The screenshot shows the Oracle Database Express Edition SQL Commands interface. A user named '19DCE109' is connected. In the SQL editor, the command `DROP TABLE SUP1;` is entered. After executing the command, the output shows 'Table dropped.' and a execution time of '0.23 seconds'. The interface includes tabs for Results, Explain, Describe, Saved SQL, and History.

```
DROP TABLE SUP1;
Table dropped.
0.23 seconds
```

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

Snap-Shot:

The screenshot shows the Oracle Database Express Edition SQL Commands interface. A user named '19DCE109' is connected. In the SQL editor, the command `UPDATE EMPLOYEE SET DEPT_NO = 10 WHERE EMP_NAME LIKE '_m%'` is entered. After executing the command, the output shows '2 row(s) updated.' and an execution time of '0.00 seconds'. The interface includes tabs for Results, Explain, Describe, Saved SQL, and History.

```
UPDATE EMPLOYEE SET DEPT_NO = 10 WHERE EMP_NAME LIKE '_m%'
2 row(s) updated.
0.00 seconds
```

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

Snap-Shot:

The screenshot shows a browser window for Oracle Database Express Edition. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:936863846923326::NO:::. The page displays an SQL command window with the following content:

```
UPDATE EMPLOYEE SET EMP_NAME = 'RAMESH' WHERE EMP_NO = 103
SELECT * FROM EMPLOYEE
```

Below the command window, the results of the SELECT query are shown in a table:

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE
101	Smith	800	500	10	shah	machine learning	ft_mgr	Toronto	105	09-AUG-96
102	Snehal	1600	300	10	gupta	data science	lec	las vegas	-	14-MAR-96
103	RAMESH	1100	500	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95
104	Aman	3000	-	10	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	ft_acc	melbourne	105	26-SEP-97
107	Anamika	2975	-	10	jha	artificial intelligence	it_prog	new york	-	15-JUL-97

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

Application Express 2.1.0.0.39
Copyright © 1999, 2008, Oracle. All rights reserved.

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

Snap-Shot:

The screenshot shows a browser window for Oracle Database Express Edition. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:936863846923326::NO::. The page displays an SQL command window with the following content:

```
ALTER TABLE EMPLOYEE ADD PHONE NUMBER(10)
SELECT * FROM EMPLOYEE
```

Below the command window, the results of the SELECT query are shown in a table:

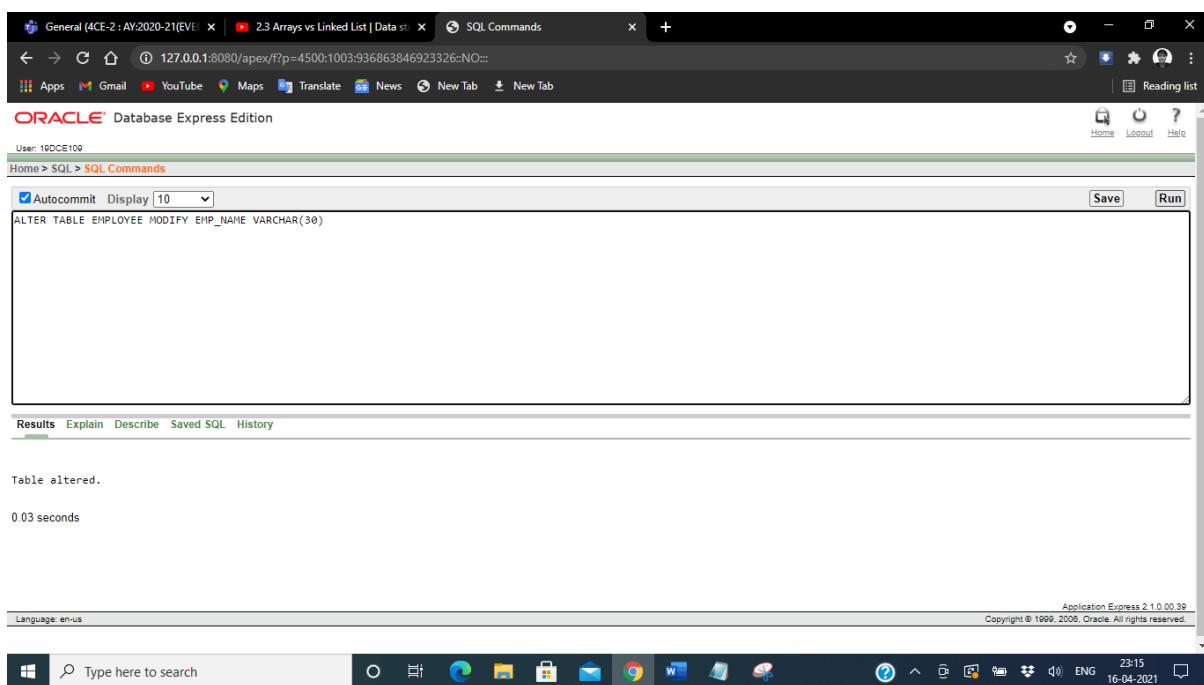
EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE	PHONE
101	Smith	800	500	10	shah	machine learning	ft_mgr	Toronto	105	09-AUG-96	-
102	Snehal	1600	300	10	gupta	data science	lec	las vegas	-	14-MAR-96	-
103	RAMESH	1100	500	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95	-
104	Aman	3000	-	10	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	ft_acc	melbourne	105	26-SEP-97	-
107	Anamika	2975	-	10	jha	artificial intelligence	it_prog	new york	-	15-JUL-97	-

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

Application Express 2.1.0.0.39
Copyright © 1999, 2008, Oracle. All rights reserved.

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

Snap-Shot:



The screenshot shows the Oracle Database Express Edition SQL Commands interface. A user named '19DCE109' is logged in. In the SQL editor, the following command is entered:

```
ALTER TABLE EMPLOYEE MODIFY EMP_NAME VARCHAR(30)
```

The 'Run' button is clicked, and the output shows:

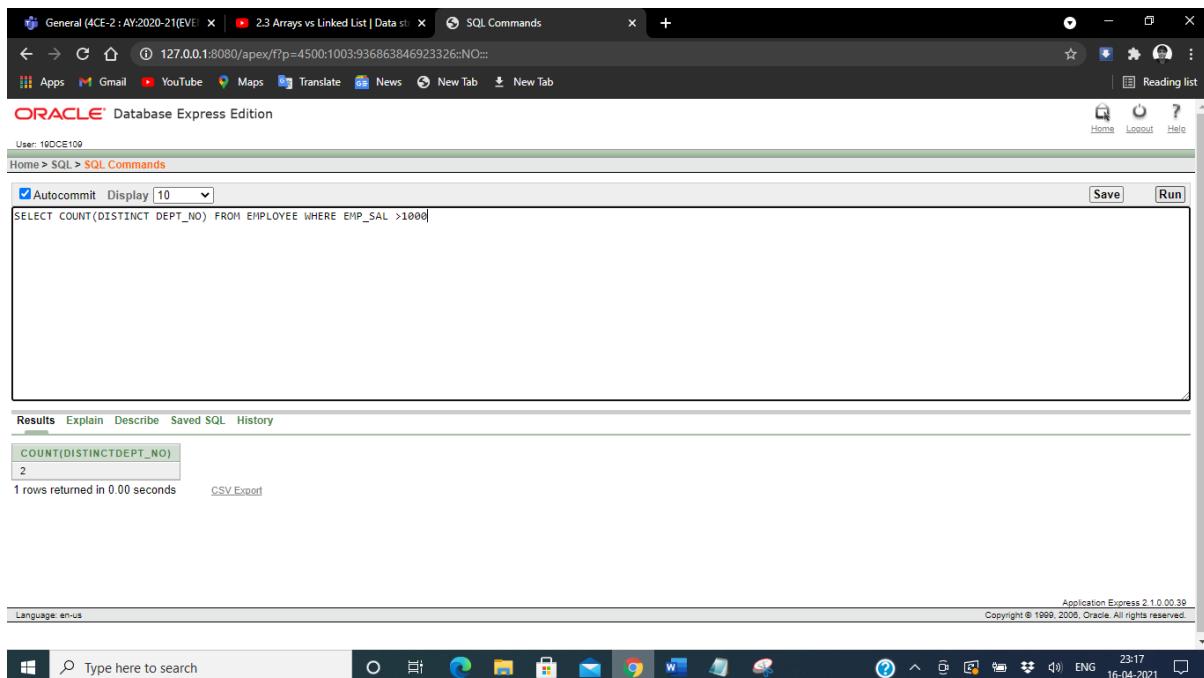
```
Table altered.
```

Execution time: 0.03 seconds.

At the bottom, the status bar indicates 'Application Express 2.1.0.00.39' and 'Copyright © 1999, 2008, Oracle. All rights reserved.'

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

Snap-Shot:



The screenshot shows the Oracle Database Express Edition SQL Commands interface. A user named '19DCE109' is logged in. In the SQL editor, the following query is entered:

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

The 'Run' button is clicked, and the output shows:

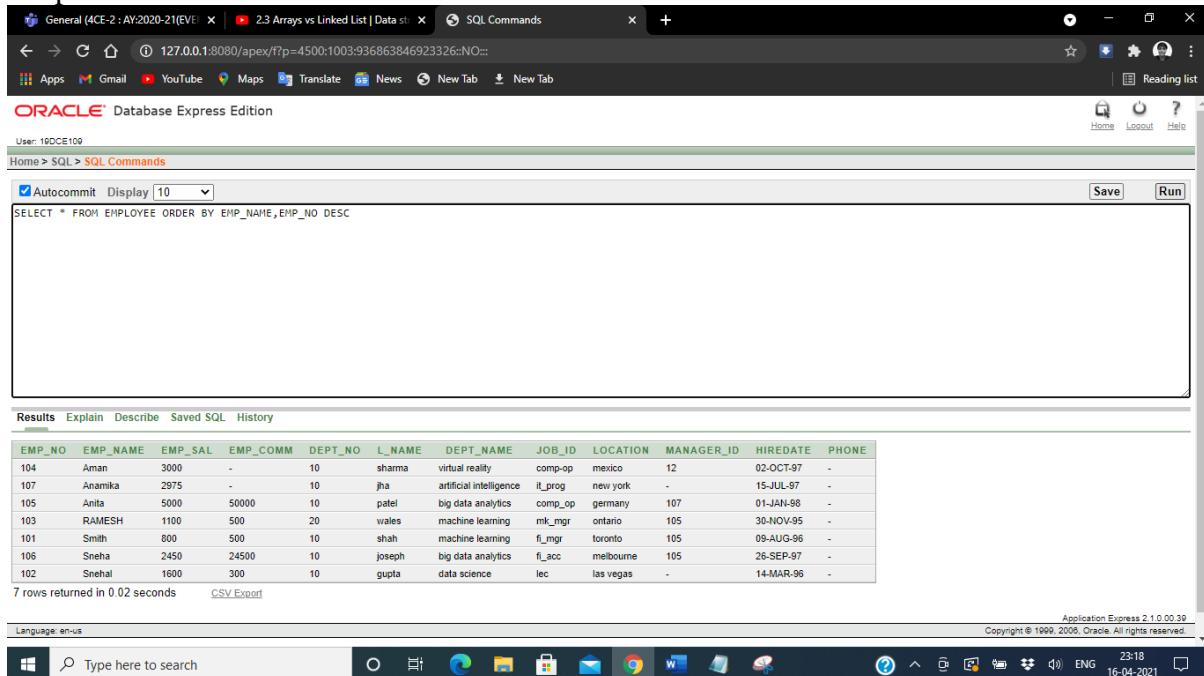
COUNT(DISTINCTDEPT_NO)
2

1 rows returned in 0.00 seconds.

At the bottom, the status bar indicates 'Application Express 2.1.0.00.39' and 'Copyright © 1999, 2008, Oracle. All rights reserved.'

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

Snap-Shot:



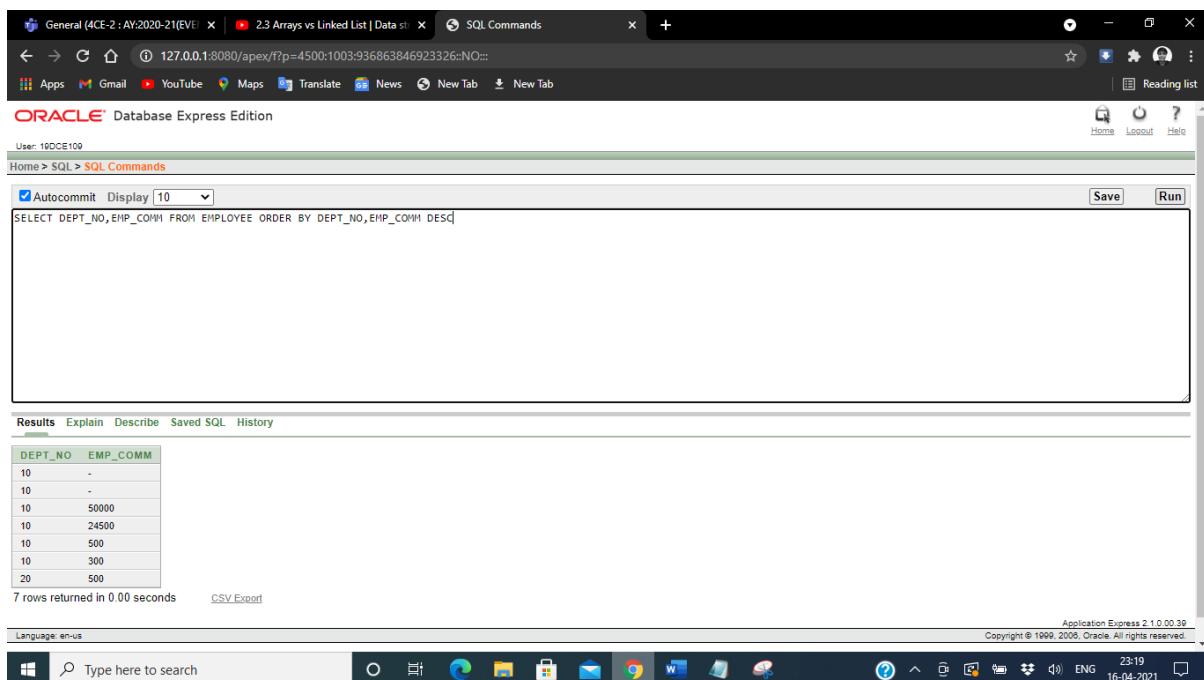
```
SELECT * FROM EMPLOYEE ORDER BY EMP_NAME,EMP_NO DESC
```

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	-	10	jha	artificial intelligence	it_prog	new york	-	15-JUL-97	-
105	Anita	5000	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-98	-
103	RAMESH	1100	500	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95	-
101	Smith	600	500	10	shah	machine learning	ft_mgr	Toronto	105	09-AUG-96	-
106	Sneha	2450	24500	10	joseph	big data analytics	ft_ae	melbourne	105	26-SEP-97	-
102	Snehal	1600	300	10	gupta	data science	iec	las vegas	-	14-MAR-96	-

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

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

Snap-Shot:



```
SELECT DEPT_NO,EMP_COMM FROM EMPLOYEE ORDER BY DEPT_NO,EMP_COMM DESC
```

DEPT_NO	EMP_COMM
10	-
10	-
10	50000
10	24500
10	500
10	300
20	500

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

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

Snap-Shot:

User: 19DCE109
Home > SQL > SQL Commands
Autocommit Display 10 Save Run
UPDATE EMPLOYEE SET EMP_COMM = 500 WHERE DEPT_NO = 20
SELECT * FROM EMPLOYEE

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE	PHONE
101	Smith	800	500	10	shah	machine learning	fi_mgr	Toronto	105	09-AUG-96	-
102	Snehal	1600	300	10	gupta	data science	lec	Las Vegas	-	14-MAR-96	-
103	RAMESH	1100	500	20	wales	machine learning	mk_mgr	Ontario	105	30-NOV-95	-
104	Aman	3000	-	10	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	-	10	jha	artificial intelligence	it_prog	New York	-	15-JUL-97	-

7 rows returned in 0.00 seconds CSV Export

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

Snap-Shot:

User: 19DCE109
Home > SQL > SQL Commands
Autocommit Display 10 Save Run
SELECT * FROM EMPLOYEE ORDER BY EMP_COMM NULLS FIRST, EMP_SAL DESC

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	-	10	jha	artificial intelligence	it_prog	New York	-	15-JUL-97	-
102	Snehal	1600	300	10	gupta	data science	lec	Las Vegas	-	14-MAR-96	-
103	RAMESH	1100	500	20	wales	machine learning	mk_mgr	Ontario	105	30-NOV-95	-
101	Smith	800	500	10	shah	machine learning	fi_mgr	Toronto	105	09-AUG-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	-

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.

Snap-Shot:

The screenshot shows the Oracle Database Express Edition interface. The SQL command entered is:

```
SELECT * FROM EMPLOYEE ORDER BY EMP_COMM NULLS FIRST, EMP_NO DESC
```

The results table displays the following data:

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE	PHONE
107	Anamika	2975	-	10	jha	artificial intelligence	it_prog	new york	-	15-JUL-97	-
104	Aman	3000	-	10	sharma	virtual reality	comp-op	mexico	12	02-OCT-97	-
102	Snehal	1600	300	10	gupta	data science	lec	las vegas	-	14-MAR-96	-
103	RAMESH	1100	500	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95	-
101	Smith	800	500	10	shah	machine learning	fi_mgr	toronto	105	09-AUG-96	-
106	Sreha	2450	24500	10	joseph	big data analytics	ft_acc	melbourne	105	26-SEP-97	-
105	Anita	5000	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-98	-

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

Conclusion: We Learned how to use aggregate functions, DML queries & Sorting.

Practical -6

To study Single-row functions.

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

Snap-Shot:

The screenshot shows the Oracle Database Express Edition interface. The SQL command entered is:

```
SELECT CURRENT_DATE "DATE" FROM DUAL
```

The results table displays the following data:

DATE
16-APR-21

1 rows returned in 0.00 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.

Snap-Shot:

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands section, the following SQL code is entered:

```
ALTER TABLE EMPLOYEE ADD NEW_SALARY NUMBER(8,2)
UPDATE EMPLOYEE SET NEW_SALARY = EMP_SAL *1.15
SELECT EMP_NO,EMP_SAL,ROUND(NEW_SALARY,0) "NEW SALARY" FROM EMPLOYEE
```

In the Results section, the output is displayed as a table:

EMP_NO	EMP_SAL	NEW SALARY
101	800	920
102	1600	1840
103	1100	1265
104	3000	3450
105	5000	5750
106	2450	2818
107	2975	3421

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.

Snap-Shot:

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands section, the following SQL code is entered:

```
ALTER TABLE EMPLOYEE ADD INCREASE NUMBER
UPDATE EMPLOYEE SET INCREASE = NEW_SALARY - EMP_SAL
SELECT EMP_NO,EMP_SAL,ROUND(NEW_SALARY,0) "NEW SALARY",INCREASE FROM EMPLOYEE
```

In the Results section, the output is displayed as a table:

EMP_NO	EMP_SAL	NEW SALARY	INCREASE
101	800	920	120
102	1600	1840	240
103	1100	1265	165
104	3000	3450	450
105	5000	5750	750
106	2450	2818	367.5
107	2975	3421	446.25

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.

+-

Snap-Shot:

```
SELECT INITCAP(EMP_NAME) AS "EMPLOYEE NAME", LENGTH(EMP_NAME) AS "LENGTH" FROM EMPLOYEE WHERE EMP_NAME LIKE 'J%' OR EMP_NAME LIKE 'A%' OR EMP_NAME LIKE 'M%' ORDER BY L_NAME DESC
```

EMPLOYEE NAME	LENGTH
Aman	4
Anita	5
Anamika	7

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

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

(5) Write a query that produces the following for each employee:
 <employee last name> earns <salary> monthly

Snap-Shot:

```
SELECT EMP_NAME || ' EARNs ' || EMP_SAL || ' MONTHLY ' || "MONTHLY SALARY" FROM EMPLOYEE
```

MONTHLY SALARY
Smith EARNs 800 MONTHLY
Snehal EARNs 1600 MONTHLY
RAMESH EARNs 1100 MONTHLY
Aman EARNs 3000 MONTHLY
Anita EARNs 5000 MONTHLY
Sneha EARNs 2450 MONTHLY
Anamika EARNs 2975 MONTHLY

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

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

(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.

Snap-Shot:

The screenshot shows a browser window with multiple tabs open. The active tab is titled "SQL Commands". The URL is "127.0.0.1:8080/apex/f?p=4500:1003:936863846923326::NO::". The SQL command entered is:

```
SELECT EMP_NAME, HIREDATE, ROUND(MONTHS_BETWEEN((SELECT CURRENT_DATE FROM DUAL), HIREDATE), 0) "MONTHS EMPLOYED", TO_CHAR(HIREDATE, 'DAY') "STARTING_DAY" FROM EMPLOYEE ORDER BY STARTING_DAY DESC
```

The results are displayed in a table:

EMP_NAME	HIREDATE	MONTHS EMPLOYED	STARTING_DAY
Anamika	15-JUL-97	285	TUESDAY
RAMESH	30-NOV-95	305	THURSDAY
Snehal	14-MAR-96	301	THURSDAY
Aman	02-OCT-97	282	THURSDAY
Anita	01-JAN-98	280	THURSDAY
Sneha	26-SEP-97	283	FRIDAY
Smith	09-AUG-96	296	FRIDAY

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

Application Express 2.1.0.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.

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

Snap-Shot:

The screenshot shows a browser window with multiple tabs open. The active tab is titled "SQL Commands". The URL is "127.0.0.1:8080/apex/f?p=4500:1003:936863846923326::NO::". The SQL command entered is:

```
SELECT TO_CHAR(HIREDATE, 'DDSPTH')||' OF '||TO_CHAR(HIREDATE, 'MONTH')||' '|EXTRACT(YEAR FROM HIREDATE)||' '|TO_CHAR(HIREDATE, 'HH:MM:SS AM') "DATE FORMAT" FROM EMPLOYEE
```

The results are displayed in a table:

DATE FORMAT
NINTH OF AUGUST 1996 12:08:00 AM
FOURTEENTH OF MARCH 1996 12:03:00 AM
THIRTEETH OF NOVEMBER 1995 12:11:00 AM
SECOND OF OCTOBER 1997 12:10:00 AM
FIRST OF JANUARY 1998 12:01:00 AM
TWENTY-SIXTH OF SEPTEMBER 1997 12:09:00 AM
FIFTEENTH OF JULY 1997 12:07:00 AM

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

Application Express 2.1.0.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.

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

Snap-Shot:

The screenshot shows a browser window with multiple tabs open. The active tab is titled "SQL Commands". The URL is 127.0.0.1:8080/apex/f?p=4500:1003:936863846923326::NO:::. The page displays the results of a SQL query:

```
SELECT EMP_SAL + EMP_COMM "ANNUAL COMPENSATION" FROM EMPLOYEE
```

The results are shown in a table:

ANNUAL COMPENSATION
1300
1900
1600
-
55000
26950
-

Below the table, it says "7 rows returned in 0.00 seconds".

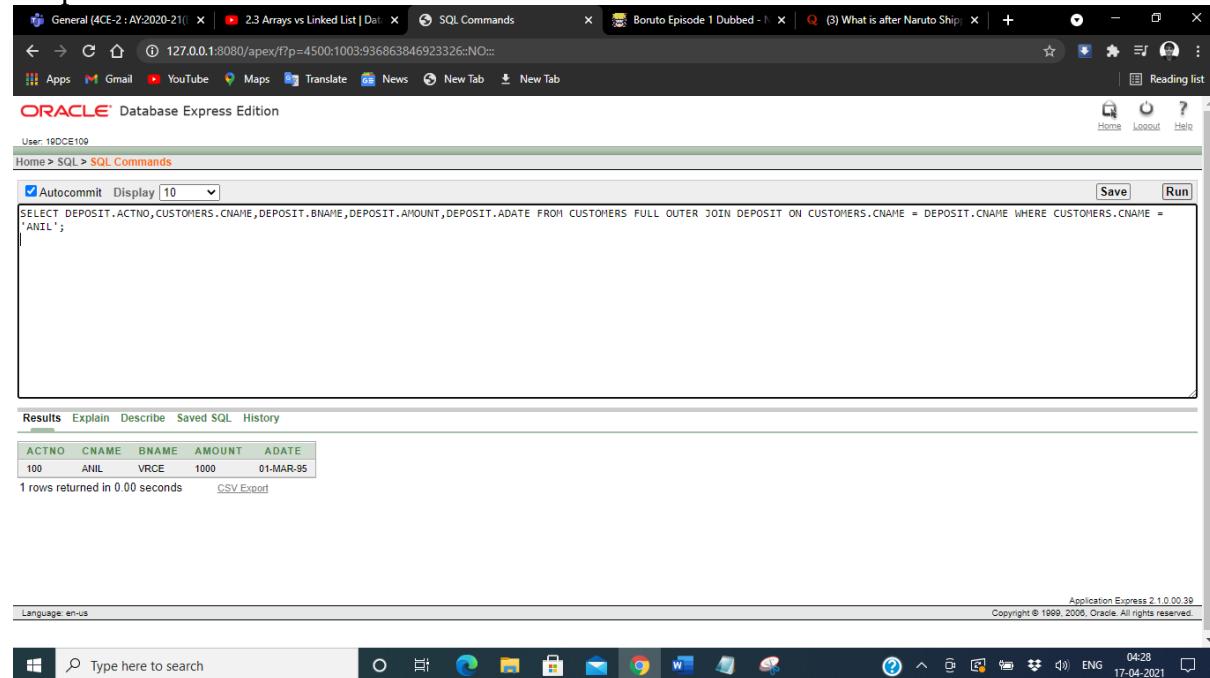
Conclusion: We Learned how to use Single-row functions.

Practical -7

Displaying data from Multiple Tables (join)

(1) Give details of customers ANIL.

Snap-Shot:



The screenshot shows a browser window with multiple tabs open. The active tab is 'SQL Commands' under 'Home > SQL > SQL Commands'. The SQL query entered is:

```
SELECT ACTNO, CUSTOMERS.CNAME, DEPOSIT.BNAME, DEPOSIT.AMOUNT, DEPOSIT.ADATE FROM CUSTOMERS FULL OUTER JOIN DEPOSIT ON CUSTOMERS.CNAME = DEPOSIT.CNAME WHERE CUSTOMERS.CNAME = 'ANIL';
```

The results pane shows a single row of data:

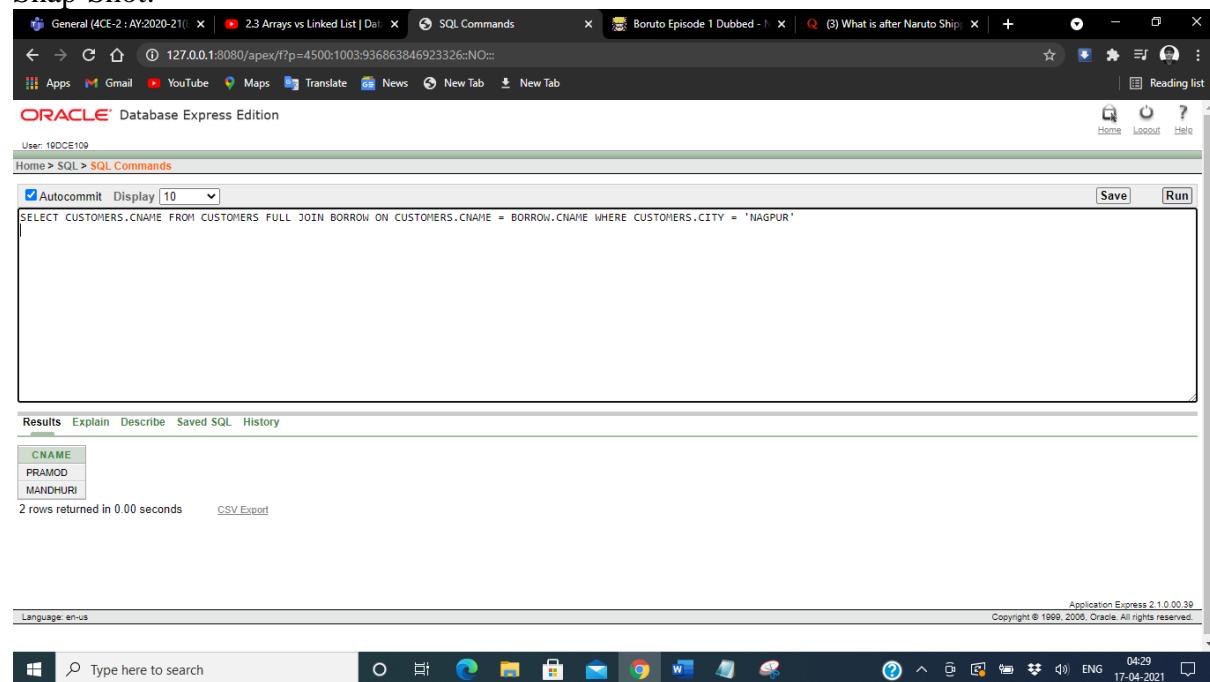
ACTNO	CNAME	BNAME	AMOUNT	ADATE
100	ANIL	VRCE	1000	01-MAR-95

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

At the bottom of the interface, it says 'Language: en-us' and 'Application Express 2.1.0.00.39 Copyright © 1999, 2008, Oracle. All rights reserved.'

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

Snap-Shot:



The screenshot shows a browser window with multiple tabs open. The active tab is 'SQL Commands' under 'Home > SQL > SQL Commands'. The SQL query entered is:

```
SELECT CUSTOMERS.CNAME FROM CUSTOMERS FULL JOIN BORROW ON CUSTOMERS.CNAME = BORROW.CNAME WHERE CUSTOMERS.CITY = 'NAGPUR'
```

The results pane shows two rows of data:

CNAME
PRAMOD
MANDHURI

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

At the bottom of the interface, it says 'Language: en-us' and 'Application Express 2.1.0.00.39 Copyright © 1999, 2008, Oracle. All rights reserved.'

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

Snap-Shot:

The screenshot shows the Oracle Database Express Edition SQL Commands interface. A SQL query is entered in the command window:

```
SELECT DEPOSIT.CNAME,BRANCH.CITY,CUSTOMERS.CITY FROM DEPOSIT JOIN BRANCH ON DEPOSIT.BNAME = BRANCH.BNAME JOIN CUSTOMERS ON DEPOSIT.CNAME = CUSTOMERS.CNAME WHERE BRANCH.CITY = CUSTOMERS.CITY
```

The results are displayed in a table:

CNAME	CITY	CITY
SHIVANI	BOMBAY	BOMBAY

1 rows returned in 0.00 seconds

CSV Export

Application Express 2.1.0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.

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

Snap-Shot:

```

SELECT DEPOSIT.CNAME,BRANCH.CITY,CUSTOMERS.CITY FROM DEPOSIT JOIN BRANCH ON DEPOSIT.BNAME = BRANCH.BNAME JOIN CUSTOMERS ON DEPOSIT.CNAME = CUSTOMERS.CNAME WHERE BRANCH.CITY = CUSTOMERS.CITY
  
```

CNAME	CITY	CITY
SHIVANI	BOMBAY	BOMBAY

1 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

Snap-Shot:

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

Snap-Shot:

```

SELECT JOB.JOB_TITLE, EMPLOYEE.LOCATION FROM EMPLOYEE INNER JOIN JOB ON EMPLOYEE.JOB_ID = JOB.JOB_ID WHERE EMPLOYEE.DEPT_NO = 30
  
```

no data found

The screenshot shows a browser window with multiple tabs open. The active tab is titled "SQL Commands". The URL is 127.0.0.1:8080/apex/r?p=4500:1003:936863846923326::NO::.

The SQL command entered is:

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

The results table shows one row:

EMP_NAME	DEPT_NO	DEPT_NAME
Anamika	10	artificial intelligence

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

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999, 2008, Oracle. All rights reserved.

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

Snap-Shot:

The screenshot shows a browser window with multiple tabs open. The active tab is titled "SQL Commands". The URL is 127.0.0.1:8080/apex/r?p=4500:1003:936863846923326::NO::.

The SQL command entered is:

```
SELECT EMP_NAME "EMPLOYEE",EMP_NO "Emp#",MANAGER_ID "Mgr#" FROM EMPLOYEE
```

The results table shows seven rows:

EMPLOYEE	Emp#	Mgr#
Smith	101	105
Snehal	102	-
RAMESH	103	105
Aman	104	12
Anita	105	107
Sneha	106	105
Anamika	107	-

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

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999, 2008, Oracle. All rights reserved.

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

Snap-Shot:

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "SQL Commands" and displays a SQL query being run against an Oracle database. The query is:

```
SELECT EMP_NAME,HIREDATE FROM EMPLOYEE WHERE HIREDATE >(SELECT HIREDATE FROM EMPLOYEE WHERE EMP_NO = 101)
```

The results of the query are displayed in a table:

EMP_NAME	HIREDATE
Aman	02-OCT-97
Anita	01-JAN-98
Sneha	26-SEP-97
Anamika	15-JUL-97

Below the table, it says "4 rows returned in 0.00 seconds" and there is a "CSV Export" link.

At the bottom of the browser window, the status bar shows "Language: en-us" and "Application Express 2.1.0.0.39 Copyright © 1999, 2008, Oracle. All rights reserved." The system tray at the bottom of the screen shows various icons and the date/time "17-04-2021 04:34".

Conclusion: We Learned how to display data using join from multiple tables.

Practical -8

To apply the concept of Aggregating Data using Group functions.

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

Snap-Shot:

The screenshot shows a browser window with multiple tabs open. The active tab is titled "SQL Commands". The URL is 127.0.0.1:8080/apex/r?p=4500:1003:936863846923326::NO::. The SQL query entered is:

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

The results section shows a single row:

SUM(AMOUNT)
10000

Below the table, it says "1 rows returned in 0.00 seconds".

At the bottom of the page, it says "Language: en-us" and "Application Express 2.1.0.0.39 Copyright © 1999, 2008, Oracle. All rights reserved."

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

Snap-Shot:

The screenshot shows a browser window with multiple tabs open. The active tab is titled "SQL Commands". The URL is 127.0.0.1:8080/apex/r?p=4500:1003:936863846923326::NO::. The SQL query entered is:

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

The results section shows a single row:

SUM(DEPOSIT.AMOUNT)
-

Below the table, it says "1 rows returned in 0.00 seconds".

At the bottom of the page, it says "Language: en-us" and "Application Express 2.1.0.0.39 Copyright © 1999, 2008, Oracle. All rights reserved."

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

Snap-Shot:

The screenshot shows a browser window with multiple tabs open. The active tab is titled "SQL Commands". The URL is "127.0.0.1:8080/apex/f?p=4500:1003:936863846923326::NO::". The page displays an Oracle Database Express Edition interface. A SQL command is entered in the editor:

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

The results pane shows a single row:

MAX(DEPOSIT.AMOUNT)
5000

Below the table, it says "1 rows returned in 0.00 seconds". The bottom status bar indicates "Application Express 2.1.0.0.39 Copyright © 1999, 2008, Oracle. All rights reserved." and the system date "17-04-2021".

(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.

Snap-Shot:

The screenshot shows a browser window with multiple tabs open. The active tab is titled "SQL Commands". The URL is "127.0.0.1:8080/apex/f?p=4500:1003:936863846923326::NO::". The page displays an Oracle Database Express Edition interface. A SQL command is entered in the editor:

```
SELECT ROUND(MAX(EMP_SAL),0) "MAXIMUM", ROUND(MIN(EMP_SAL),0)"MINIMUM", ROUND(SUM(EMP_SAL),0)"SUM",ROUND(AVG(EMP_SAL),0)"AVERAGE" FROM EMPLOYEE
```

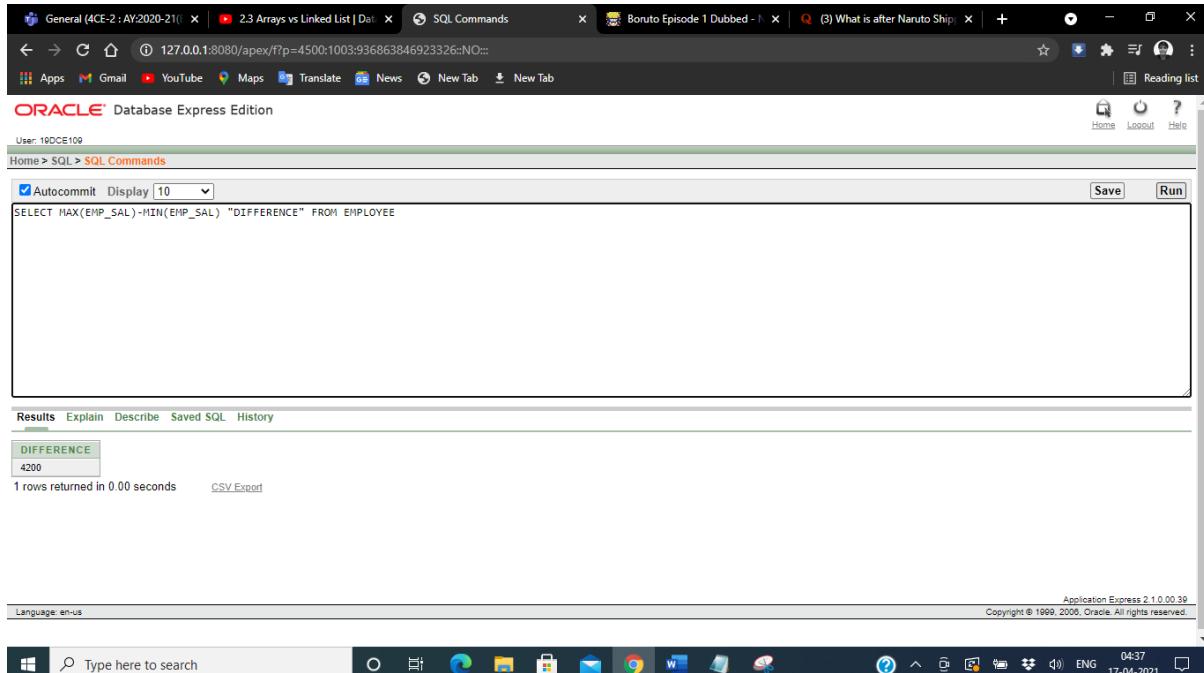
The results pane shows a single row:

MAXIMUM	MINIMUM	SUM	AVERAGE
5000	800	16925	2418

Below the table, it says "1 rows returned in 0.00 seconds". The bottom status bar indicates "Application Express 2.1.0.0.39 Copyright © 1999, 2008, Oracle. All rights reserved." and the system date "17-04-2021".

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

Snap-Shot:



The screenshot shows a browser window with multiple tabs open. The active tab is titled "SQL Commands". The URL is 127.0.0.1:8080/apex/r?p=4500:1003:936863846923326::NO::. The page displays an Oracle Database Express Edition interface. A SQL query is entered in the editor:

```
SELECT MAX(EMP_SAL)-MIN(EMP_SAL) "DIFFERENCE" FROM EMPLOYEE
```

The results section shows a single row:

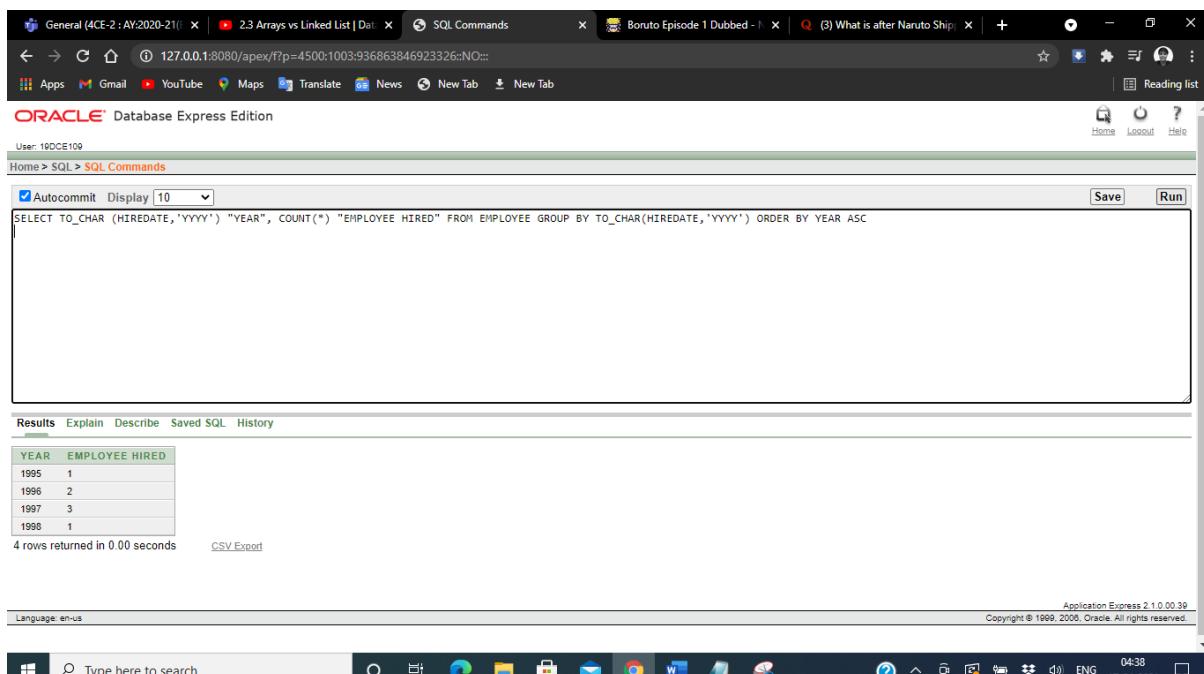
DIFFERENCE
4200

Below the table, it says "1 rows returned in 0.00 seconds".

At the bottom of the page, it says "Language: en-us" and "Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved."

(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

Snap-Shot:



The screenshot shows a browser window with multiple tabs open. The active tab is titled "SQL Commands". The URL is 127.0.0.1:8080/apex/r?p=4500:1003:936863846923326::NO::. The page displays an Oracle Database Express Edition interface. A SQL query is entered in the editor:

```
SELECT TO_CHAR (HIREDATE,'YYYY') "YEAR", COUNT(*) "EMPLOYEE_HIRED" FROM EMPLOYEE GROUP BY TO_CHAR(HIREDATE,'YYYY') ORDER BY YEAR ASC
```

The results section shows a table:

YEAR	EMPLOYEE_HIRED
1995	1
1996	2
1997	3
1998	1

Below the table, it says "4 rows returned in 0.00 seconds".

At the bottom of the page, it says "Language: en-us" and "Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved."

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

Snap-Shot:

The screenshot shows a browser window with multiple tabs open. The active tab is titled "SQL Commands". The URL is 127.0.0.1:8080/apex/r?p=4500:1003:936863846923326::NO::.

The SQL command entered is:

```
SELECT DISTINCT DEPT_NAME, AVG(EMP_SAL) "SALARY" FROM EMPLOYEE GROUP BY DEPT_NAME ORDER BY DEPT_NAME ASC
```

The results table shows the following data:

DEPT_NAME	SALARY
artificial intelligence	2975
big data analytics	3725
data science	1600
machine learning	950
virtual reality	3000

5 rows returned in 0.02 seconds

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2000, Oracle. All rights reserved.

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

Snap-Shot:

The screenshot shows a browser window with multiple tabs open. The active tab is titled "SQL Commands". The URL is 127.0.0.1:8080/apex/r?p=4500:1003:936863846923326::NO::.

The SQL command entered is:

```
SELECT DISTINCT JOB.JOB_TITLE, SUM(EMPLOYEE.EMP_SAL) "TOTAL SALARY" FROM EMPLOYEE JOIN JOB ON EMPLOYEE.JOB_ID = JOB.JOB_ID GROUP BY JOB.JOB_TITLE
```

The results table shows the following data:

JOB_TITLE	TOTAL SALARY
Programmer	2975
Lecturer	1600
Computer Operator	5000
Account	2450
Finance manager	800
Marketing manager	1100

6 rows returned in 0.02 seconds

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2000, Oracle. All rights reserved.

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

Snap-Shot:

The screenshot shows the Oracle Database Express Edition interface. A SQL command is entered in the SQL Commands window:

```
SELECT DISTINCT DEPT_NAME, AVG(EMP_SAL) "SALARY" FROM EMPLOYEE WHERE EMP_SAL > 2000 GROUP BY DEPT_NAME
```

The results are displayed in a table:

DEPT_NAME	SALARY
big data analytics	3725
artificial intelligence	2975
virtual reality	3000

3 rows returned in 0.00 seconds

CSV Export

Application Express 2.1.0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.

(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.

Snap-Shot:

The screenshot shows the Oracle Database Express Edition interface. A SQL command is entered in the SQL Commands window:

```
SELECT DISTINCT JOB.JOB_TITLE, SUM(EMPLOYEE.EMP_SAL) "TOTAL SALARY" FROM EMPLOYEE JOIN JOB ON EMPLOYEE.EMP_SAL >= 3000 GROUP BY JOB.JOB_TITLE ORDER BY SUM(EMPLOYEE.EMP_SAL)
```

The results are displayed in a table:

JOB_TITLE	TOTAL SALARY
Programmer	8000
Account	8000
Marketing manager	8000
Computer Operator	8000
Finance manager	8000
Lecturer	8000

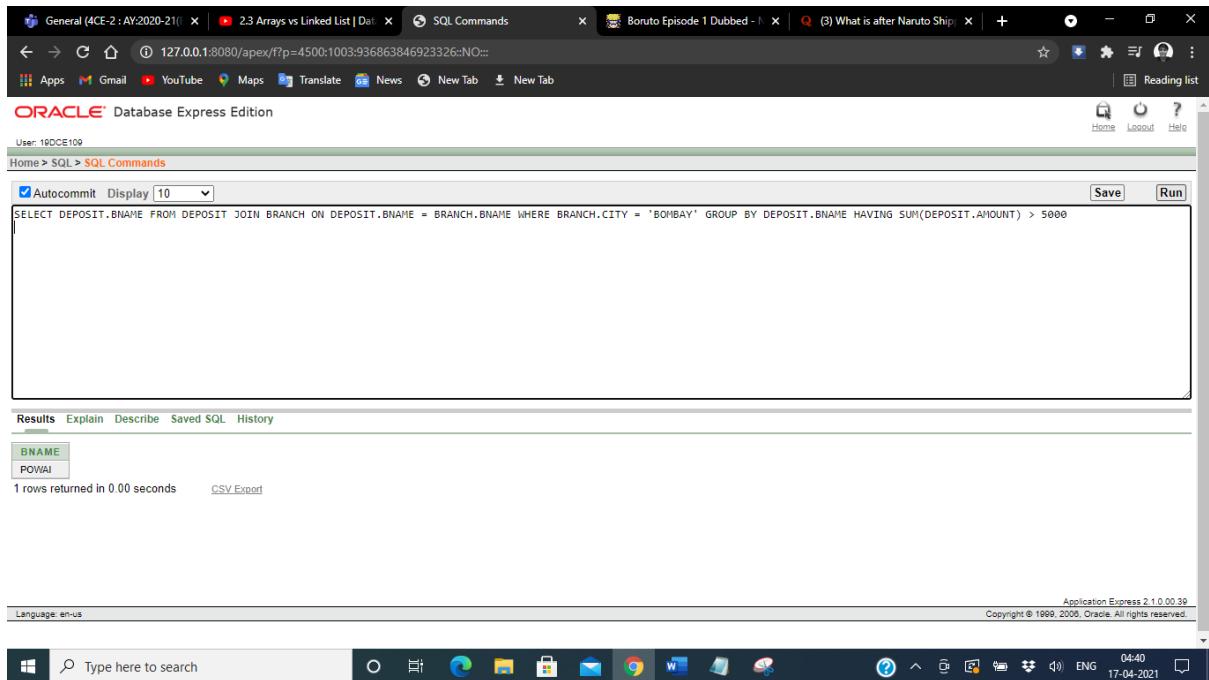
6 rows returned in 0.00 seconds

CSV Export

Application Express 2.1.0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.

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

Snap-Shot:



The screenshot shows a Windows desktop environment with a browser window open to the Oracle Database Express Edition SQL Commands interface. The browser's address bar shows the URL: 127.0.0.1:8080/apex/f?p=4500:1003:936863846923326::NO:::. The main window displays a SQL query:

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

The results pane shows one row:

BNAME
POWAI

Below the results, it says "1 rows returned in 0.00 seconds". The bottom status bar indicates "Language: en-us", "Application Express 2.1.0.00.39", and "Copyright © 1999, 2006, Oracle. All rights reserved."

Conclusion: We Learned about Aggregating Data using Group functions.

Practical -9

To solve queries using the concept of sub query.

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

Snap-Shot:

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

L_NAME	HIREDATE
wales	30-NOV-95

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

Application Express 2.1.0.0.39
Copyright © 1999, 2008, Oracle. All rights reserved.

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

Snap-Shot:

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

Snap-Shot:

The screenshot shows a browser window with multiple tabs open. The active tab is titled "SQL Commands". The URL is "127.0.0.1:8080/apex/f?p=4500:1003:936863846923326::NO::". The SQL query is:

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

The results show two rows:

CNAME
ANIL
SUNIL

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

The screenshot shows a browser window with multiple tabs open. The active tab is titled "SQL Commands". The URL is "127.0.0.1:8080/apex/f?p=4500:1003:936863846923326::NO::". The SQL query is:

```
SELECT DEPOSIT.ACNO,DEPOSIT.AMOUNT AS DEPOSIT_AMOUNT,BORROW.LOANNO,BORROW.AMOUNT AS BORROW_AMOUNT FROM DEPOSIT JOIN BORROW ON DEPOSIT.CNAME = BORROW.CNAME JOIN CUSTOMERS ON DEPOSIT.CNAME = CUSTOMERS.CNAME WHERE CUSTOMERS.CITY = ( SELECT CITY FROM CUSTOMERS WHERE CNAME = 'PRAMOD')
```

The results show no data found.

(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.

Snap-Shot:

User: 19DCE109

Home > SQL > SQL Commands

Autocommit Display 10

```
SELECT EMP_NO,L_NAME FROM EMPLOYEE WHERE EMP_SAL > (SELECT AVG(EMP_SAL) FROM EMPLOYEE) ORDER BY EMP_SAL ASC
```

Results Explain Describe Saved SQL History

EMP_NO	L_NAME
106	joseph
107	jha
104	sharma
105	patel

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

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999, 2008, Oracle. All rights reserved.

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

Snap-Shot:

User: 19DCE109

Home > SQL > SQL Commands

Autocommit Display 10

```
SELECT DEPOSIT.CNAME FROM DEPOSIT JOIN BRANCH ON DEPOSIT.BNAME = BRANCH.BNAME WHERE DEPOSIT.AMOUNT > 2000 AND BRANCH.CITY = (SELECT CITY FROM BRANCH WHERE BNAME = (SELECT BNAME FROM DEPOSIT WHERE CNAME = 'ANIL'))
```

Results Explain Describe Saved SQL History

CNAME
SUNIL

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

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999, 2008, Oracle. All rights reserved.

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

Snap-Shot:

The screenshot shows a browser window with multiple tabs open. The active tab is 'SQL Commands' under 'Home > SQL'. A SQL query is entered in the command field:

```
SELECT L_NAME,EMP_SAL FROM EMPLOYEE WHERE MANAGER_ID = (SELECT EMP_NO FROM EMPLOYEE WHERE EMP_NAME = 'FROD')
```

The results section below shows the message: "no data found".

At the bottom of the page, it says "Language: en-us" and "Application Express 2.1.0.0.39 Copyright © 1999, 2008, Oracle. All rights reserved."

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

Snap-Shot:

The screenshot shows a browser window with multiple tabs open. The active tab is 'SQL Commands' under 'Home > SQL'. A SQL query is entered in the command field:

```
SELECT EMPLOYEE.DEPT_NO,EMPLOYEE.DEPT_NAME,JOB.JOB_TITLE FROM EMPLOYEE JOIN JOB ON EMPLOYEE.JOB_ID = JOB.JOB_ID WHERE JOB.JOB_TITLE = 'Account'
```

The results section shows a single row of data:

DEPT_NO	DEPT_NAME	JOB_TITLE
10	big data analytics	Account

Below the table, it says "1 rows returned in 0.00 seconds" and "CSV Export".

At the bottom of the page, it says "Language: en-us" and "Application Express 2.1.0.0.39 Copyright © 1999, 2008, Oracle. All rights reserved."

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

Snap-Shot:

The screenshot shows the Oracle Database Express Edition interface. A SQL command is entered in the query editor:

```
SELECT D1.BNAME FROM DEPOSIT D1 GROUP BY D1.BNAME HAVING COUNT(D1.CNAME)>= ALL (SELECT COUNT(D2.CNAME) FROM DEPOSIT D2 GROUP BY D2.BNAME);
```

The results table shows the following data:

BNAME
VRCE
AJNI
M.G ROAD
VIRAR
POVVAI
CHANDI
ANDHERI
KAROBAGH
NEHRU PLACE

9 rows returned in 0.00 seconds

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

Snap-Shot:

The screenshot shows the Oracle Database Express Edition interface. A SQL command is entered in the query editor:

```
SELECT CITY FROM BRANCH GROUP BY CITY HAVING COUNT(BRANCH.BNAME) >= ALL (SELECT COUNT(BNAME) FROM BRANCH GROUP BY CITY)
```

The results table shows the following data:

CITY
NAGPUR
DELHI
BOMBAY

3 rows returned in 0.01 seconds

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2000, Oracle. All rights reserved.

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

Snap-Shot:

The screenshot shows a browser window with multiple tabs open. The active tab is titled "SQL Commands". The URL is 127.0.0.1:8080/apex/r?p=4500:1003:936863846923326::NO::.

The page displays an Oracle Database Express Edition interface. A SQL query is entered in the command window:

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

The results are shown in a table:

CNAME
SHIVANI
KRANTI
NAREN

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

At the bottom, the status bar shows: Language: en-us Application Express 2.1.0.0.39 Copyright © 1999, 2008, Oracle. All rights reserved. 04:47 17-04-2021

Conclusion: From This practical I learned how to perform sub-queries.

Practical -10

Manipulating Data

(1) Give 10% interest to all depositors.

Snap-Shot:

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL command entered is:

```
UPDATE DEPOSIT SET AMOUNT = AMOUNT*1.10
SELECT * FROM DEPOSIT;
```

The results section displays the following table:

ACTNO	CNAME	BNAME	AMOUNT	ADATE
100	ANIL	VRCE	1100	01-MAR-95
101	SUNIL	AJNI	5500	04-JAN-96
102	MEHUL	KAROBAGH	3850	17-NOV-95
104	MADHURI	CHANDI	1320	17-DEC-95
105	PRMOD	M.G.ROAD	3300	27-MAR-96
106	SANDIP	ANDHERI	2200	31-MAR-96
107	SHIVANI	VIRAR	1100	05-SEP-95
108	KRANTI	NEHRU PLACE	5500	02-JUL-95
109	MINU	POWAI	7700	10-AUG-95

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

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

Snap-Shot:

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL command entered is:

```
UPDATE DEPOSIT SET AMOUNT = AMOUNT*1.10 WHERE BNAME = 'VRCE'
SELECT * FROM DEPOSIT;
```

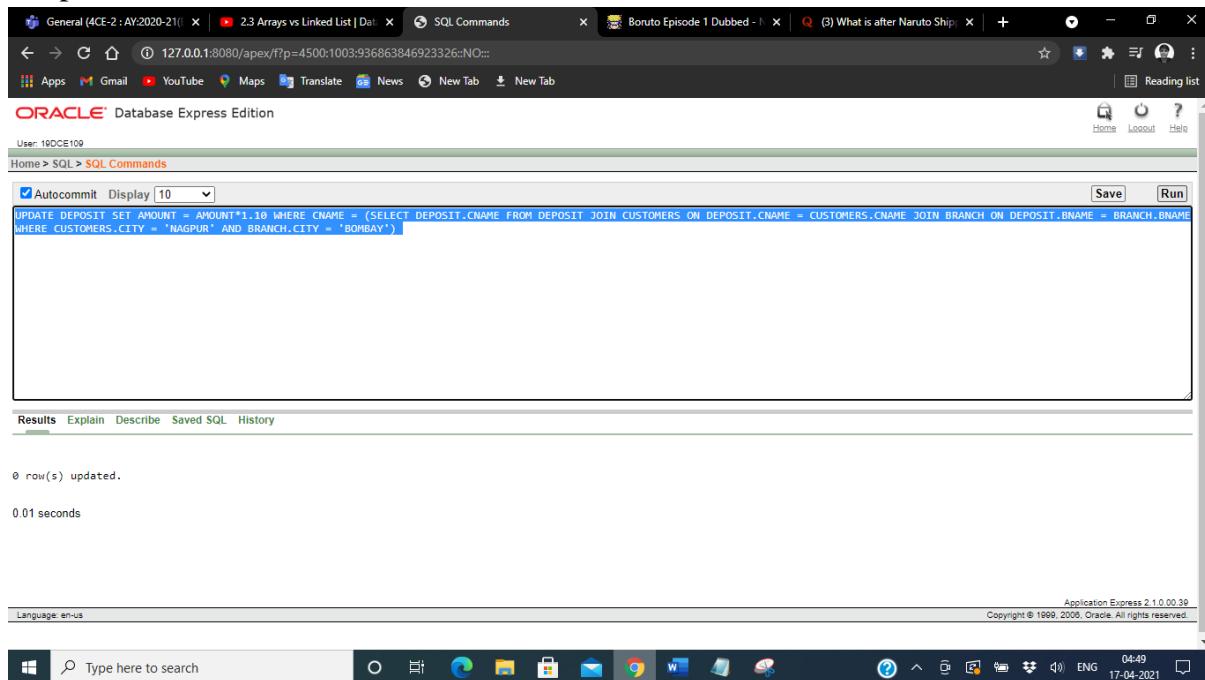
The results section displays the following table:

ACTNO	CNAME	BNAME	AMOUNT	ADATE
100	ANIL	VRCE	1210	01-MAR-95
101	SUNIL	AJNI	5500	04-JAN-96
102	MEHUL	KAROBAGH	3850	17-NOV-95
104	MADHURI	CHANDI	1320	17-DEC-95
105	PRMOD	M.G.ROAD	3300	27-MAR-96
106	SANDIP	ANDHERI	2200	31-MAR-96
107	SHIVANI	VIRAR	1100	05-SEP-95
108	KRANTI	NEHRU PLACE	5500	02-JUL-95
109	MINU	POWAI	7700	10-AUG-95

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

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

Snap-Shot:



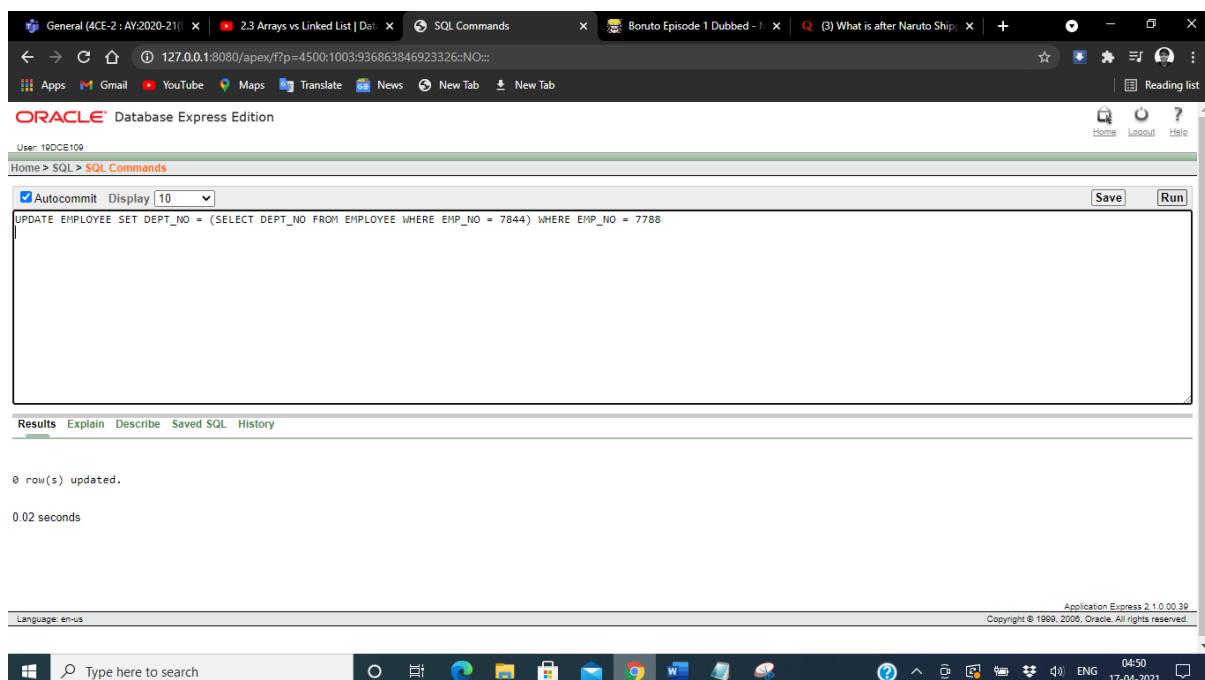
The screenshot shows a browser window with the Oracle Database Express Edition interface. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:936863846923326::NO:::. The page title is "SQL Commands". The SQL query entered is:

```
UPDATE DEPOSIT SET AMOUNT = AMOUNT*1.10 WHERE CNAME = (SELECT DEPOSIT.CNAME FROM DEPOSIT JOIN CUSTOMERS ON DEPOSIT.CNAME = CUSTOMERS.CNAME JOIN BRANCH ON DEPOSIT.BNAME = BRANCH.BNAME WHERE CUSTOMERS.CITY = 'NAGPUR' AND BRANCH.CITY = 'BOMBAY');
```

Below the query, the results show "0 row(s) updated." and "0.01 seconds". The status bar at the bottom right indicates "Application Express 2.1.0.00.39 Copyright © 1999, 2000, Oracle. All rights reserved." and the date "17-04-2021".

(4) Write a query which changes the department number of all employees with empno 7788's job to employee 7844's current department number.

Snap-Shot:



The screenshot shows a browser window with the Oracle Database Express Edition interface. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:936863846923326::NO:::. The page title is "SQL Commands". The SQL query entered is:

```
UPDATE EMPLOYEE SET DEPT_NO = (SELECT DEPT_NO FROM EMPLOYEE WHERE EMP_NO = 7844) WHERE EMP_NO = 7788;
```

Below the query, the results show "0 row(s) updated." and "0.02 seconds". The status bar at the bottom right indicates "Application Express 2.1.0.00.39 Copyright © 1999, 2000, Oracle. All rights reserved." and the date "17-04-2021".

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

Snap-Shot:

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands tab, the following SQL code is entered:

```

SELECT * FROM DEPOSIT
UPDATE DEPOSIT SET BNAME = 'VRCE' WHERE ACTNO = 101
UPDATE DEPOSIT SET AMOUNT = AMOUNT - 10 WHERE CNAME = 'ANIL' AND BNAME = (SELECT BNAME FROM DEPOSIT WHERE CNAME = 'SUNIL')
UPDATE DEPOSIT SET AMOUNT = AMOUNT + 10 WHERE CNAME = 'SUNIL' AND BNAME = (SELECT BNAME FROM DEPOSIT WHERE CNAME = 'ANIL')

```

The Results tab displays the updated deposit records:

ACTNO	CNAME	BNAME	AMOUNT	ADATE
100	ANIL	VRCE	1200	01-MAR-95
101	SUNIL	VRCE	5510	04-JAN-96
102	MEHUL	KAROBAGH	3850	17-NOV-95
104	MADHURI	CHANDI	1320	17-DEC-95
105	PRMOD	M.G.ROAD	3300	27-MAR-96
106	SANDIP	ANDHERI	2200	31-MAR-96
107	SHIVANI	VIRAR	1100	05-SEP-95
108	KRANTI	NEHRU PLACE	5500	02-JUL-95
109	MINU	POWAI	7700	10-AUG-95

9 rows returned in 0.00 seconds

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

Snap-Shot:

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands tab, the following SQL code is entered:

```

UPDATE DEPOSIT SET AMOUNT = AMOUNT + 100 WHERE CNAME = ANY(SELECT CNAME FROM DEPOSIT WHERE AMOUNT IN (SELECT MAX(AMOUNT) FROM DEPOSIT GROUP BY BNAME))
SELECT * FROM DEPOSIT

```

The Results tab displays the updated deposit records:

ACTNO	CNAME	BNAME	AMOUNT	ADATE
100	ANIL	VRCE	1200	01-MAR-95
101	SUNIL	VRCE	5610	04-JAN-96
102	MEHUL	KAROBAGH	3950	17-NOV-95
104	MADHURI	CHANDI	1420	17-DEC-95
105	PRMOD	M.G.ROAD	3400	27-MAR-96
106	SANDIP	ANDHERI	2300	31-MAR-96
107	SHIVANI	VIRAR	1200	05-SEP-95
108	KRANTI	NEHRU PLACE	5600	02-JUL-95
109	MINU	POWAI	7800	10-AUG-95

9 rows returned in 0.00 seconds

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

Snap-Shot:

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

ACTNO	CNAME	BNAME	AMOUNT	ADATE
102	MEHUL	KAROBAGH	3950	17-NOV-95
104	MADHURI	CHANDI	1420	17-DEC-95
105	PRMOD	M.G.ROAD	3400	27-MAR-96
106	SANDIP	ANDHERI	2300	31-MAR-96
107	SHIVANI	VIRAR	1200	05-SEP-95
108	KRANTI	NEHRU PLACE	5600	02-JUL-95
109	MINU	POWAI	7800	10-AUG-95

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

(8) Delete deposit of vijay.

Snap-Shot:

```
DELETE FROM DEPOSIT WHERE CNAME = 'VIJAY'
```

0 row(s) deleted.

0.00 seconds

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

Snap-Shot:

The screenshot shows a browser window with multiple tabs open. The active tab is titled "SQL Commands". The URL is 127.0.0.1:8080/apex/f?p=4500:1003:936863846923326::NO:::. The page header indicates "ORACLE Database Express Edition" and "User: 19DCE109". Below the header, the navigation bar shows "Home > SQL > SQL Commands". The main area contains a SQL editor with the following code:

```
DELETE FROM BORROW WHERE AMOUNT < 1000
```

Below the editor, there are buttons for "Save" and "Run". The status bar at the bottom shows "0 row(s) deleted." and "0.00 seconds". The footer of the application window displays "Application Express 2.1.0.0.39" and "Copyright © 1999, 2008, Oracle. All rights reserved.". The taskbar at the bottom of the screen shows various icons for Windows applications like File Explorer, Mail, and Google Chrome.

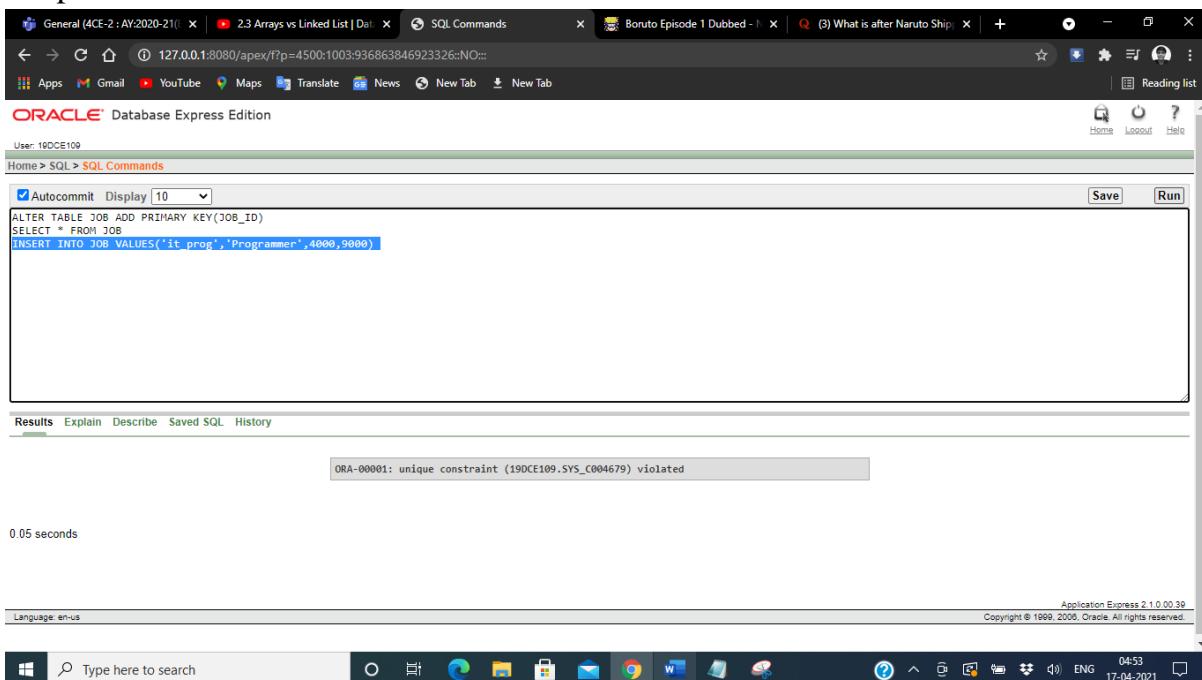
Conclusion: From this practical I learned how to manipulate data.

Practical -11

Add and Remove constraint

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

Snap-Shot:



The screenshot shows a browser window with multiple tabs open. The active tab is titled "SQL Commands". The SQL code entered is:

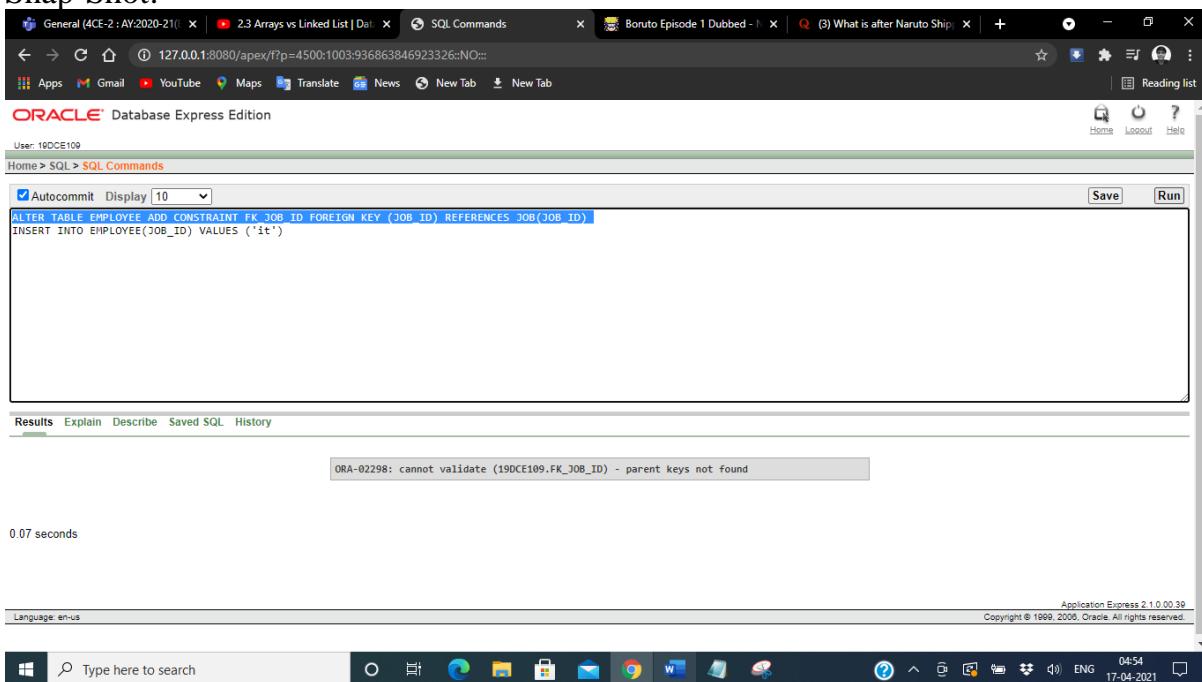
```
ALTER TABLE JOB ADD PRIMARY KEY(JOB_ID)
SELECT * FROM JOB
INSERT INTO JOB VALUES('it_prog','Programmer',4000,9000)
```

After running the command, an error message is displayed: "ORA-00001: unique constraint (19DCE109.SYS_C004679) violated".

At the bottom of the screen, the Windows taskbar is visible with various icons and the date/time: "04:53 17-04-2021".

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

Snap-Shot:



The screenshot shows a browser window with multiple tabs open. The active tab is titled "SQL Commands". The SQL code entered is:

```
ALTER TABLE EMPLOYEE ADD CONSTRAINT FK_JOB_ID FOREIGN KEY (JOB_ID) REFERENCES JOB(JOB_ID)
INSERT INTO EMPLOYEE(JOB_ID) VALUES ('it')
```

After running the command, an error message is displayed: "ORA-02298: cannot validate (19DCE109.FK_JOB_ID) - parent keys not found".

At the bottom of the screen, the Windows taskbar is visible with various icons and the date/time: "04:54 17-04-2021".

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

Snap-Shot:

```

CREATE TABLE LOCK11(COMPANY VARCHAR2(30), MODEL VARCHAR2(30), PRIMARY KEY(COMPANY,MODEL))
SELECT * FROM LOCK11
INSERT INTO LOCK11(COMPANY,MODEL) VALUES('Euro','654-5259x')
INSERT INTO LOCK11(COMPANY,MODEL) VALUES('Euro','654-5259x')

Results Explain Describe Saved SQL History
COMPANY MODEL
Euro 654-5259x
1 rows returned in 0.01 seconds CSV Export

```

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

(4) Remove primary key constraint on job_id

Snap-Shot:

```

ALTER TABLE EMPLOYEE DROP CONSTRAINT FK_JOB_ID
ALTER TABLE JOB DROP PRIMARY KEY

Results Explain Describe Saved SQL History
Table dropped.
0.03 seconds

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

```

(5) Remove foreign key constraint on employee table

Snap-Shot:

The screenshot shows a browser window for Oracle Database Express Edition. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:936863846923326::NO::: Home > SQL > SQL Commands. A SQL command is entered in the text area: `ALTER TABLE EMPLOYEE DROP CONSTRAINT FK_JOB_ID;`. The results section shows an error message: `ORA-02443: Cannot drop constraint - nonexistent constraint`. The operating system taskbar at the bottom indicates it's running on Windows 10.

Conclusion: From this practical I learned how to remove and add constraints on a table.

Practical -12

(12.1) Data Dictionary

Snap-shot:

Field ID	Data type	Field Name	Examples	Constraints
2 emp_id	number	6 Employee Unique Id	101	PK
3 emp_name	text	20 Employee Name (F_name + L_name)	Het Patel	Non Null
4 F_name	text	10 First Name	Het	Non Null
5 L_name	text	10 Last Name	Patel	Non Null
6 Designation	text	10 Designation of an employee	Cleaner	Non Null
7 joining_date	date	8 Date of Joining	28-01-2002	Non Null
8 year_of_experience	number	2 Current year - joining year	18	Non Null
9 Cleaning_assig_Id	text	16 Format :- Date(6) + Emp_id(6)+Room_id(4)	200120021011234	PK
10 Room_id	number	4 Unique Room Number	1234	PK
11 Room_Type	text	It can be either SR(Single Room) or DR (Double Room) or TR(Triple Room)		
		Also it is Derived Attribute	SR	Non Null
12 Occupation	Date / Time	10 Room will be occupied (Departure Time-Arrival Time)	48	Non Null
13 Credit_Card	number	16 Credit Card Number and its details	1234 4567 8901	Unique And Not Null
14 Documents	text	50 More details about guest like their address proof, driving licence or adhar card	GJ02 20200024042	Not Null

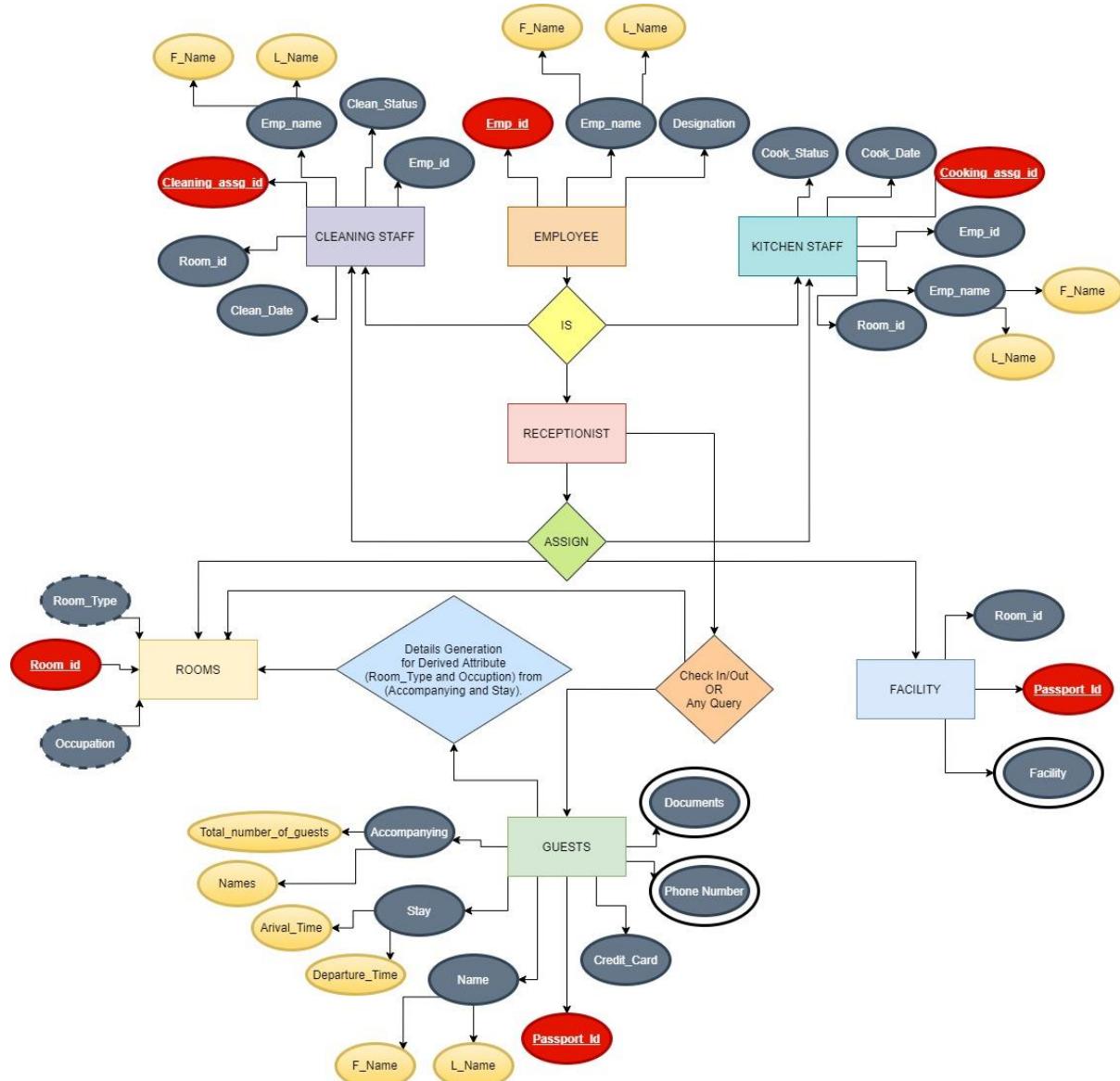
A	B	C	D	E	F	G	H	I	J	K	L	M	N
13 Credit_Card	number	16 Credit Card Number and its details.	1234 4567 8901	Unique And Not Null									
14 Documents	Text	More details about guest like their 50 address proof, driving licence or adhar card.	Driving licence :- kadjakcd1243	Non Null									
15 Passport_id	number	9 PassPort Number of guest.	123456789	PK									
16 PhoneNumber	number	10 Phone number of guest. There can more than one number	623456789	Unique And Not Null									
17 Arrival_Time	Date/Time	10 Arrival date with time.	01/01/2007 12:30pm	Non Null									
18 Departure_Time	Date/Time	10 Departure date with time.	02/01/2007 12:30pm	Non Null									
19 Name	Text	20 Guest Name (F_Name + L_Name).	Meet	Non Null									
20 Accompanying	Text	Other people accompanying guest.it is contains of total number of guest and names of them.if guest is single it should be filled with "None".	total guest :- 12 ;Other Guests Names:- Yug,Tanmay,Shrey,Jay,Deep ,Vaibhav,Darshan,Priyanshu ,Yash,Jaimin Don	Non Null									
21 Clean_Date	Date/Time	10 Cleaning Date and time	01/01/2007 12:00pm	Non Null									
22 Clean_Status	Text	10 Its describes current cleaning status of an assigned room.	Cleaning Done.	Non Null									
23 Facility	Text	It can have multiple activities like(Gym and Swimming) according to guest package.	Gym,GameZone and Swimming	Non Null									
		Format :- Date(6)+Emp_id(6)+Room_id(4)											

Excel Screenshot showing Data Dictionary - Excel

22	Clean_Status	Text	10 Its describes current cleaning status of an assigned room.	Cleaning Done.	Non Null								
23	Facility	Text	It can have multiple activities like(Gym 20 and Swimming) according to guest package.	Gym,GameZone and Swimming	Non Null								
24	Cooking_assign_id	Text	Format :- Date(6)+Emp_Id(6)+Room_Id(4) 16 This is Unique assigned id work by cooking staff.	12032006100120101	PK								
25	Cook_Date	Date/Time	10 Cooking Date and Time	01/01/2007 2:00pm	Non Null								
26	Cook_Status	Text	10 Its describes current cooking order status given for the guest for assigned room.	Food is ready.	Non Null								
27	Stay	Date/Time	10 Guest Stay (Departure Time - Arrival Time). And it is also derived attribute.	48 Hours	Non Null								
28													
29													
30													
31													
32													
33													
34													
35													
36													
37													

(12.2) E-R Diagram.

Snap-shot:



Conclusion: From This Practical I Learned how to make an E-R Diagram.

Practical -13

To perform basic PL/SQL blocks

Write a PL-SQL block to find Sum and average of three numbers.

Code:

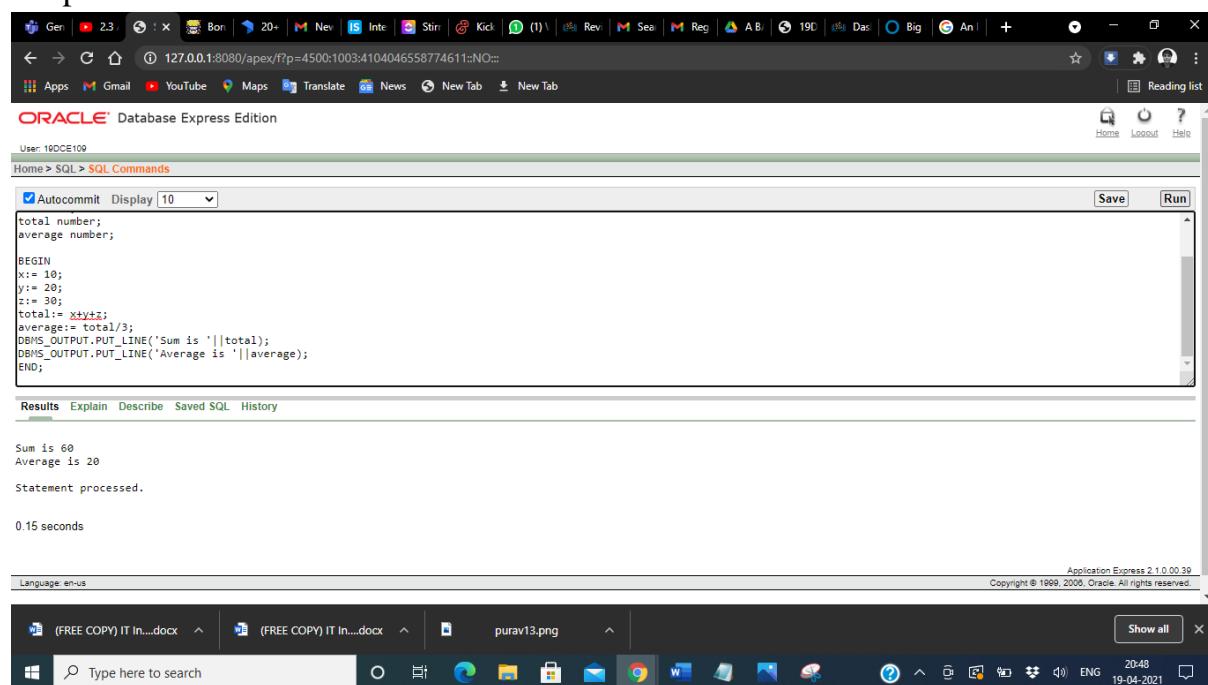
```

DECLARE
    x number;
    y number;
    z number;
    total number;
    average number;

BEGIN
    x:= 10;
    y:= 20;
    z:= 30;
    total:= x+y+z;
    average:= total/3;
    DBMS_OUTPUT.PUT_LINE('Sum is'||total);
    DBMS_OUTPUT.PUT_LINE('Average is'||average);
END;

```

Snap-Shot:



Conclusion: From this practical I learned How to write code in PL/SQL.

Practical -14

To perform the concept of loop

Find the factorial of a number in pl/sql using for, While and Simple Loop.

Code:

Basic Loop

```
DECLARE
    x number := 5;
    fact number :=1;

BEGIN
    LOOP
        fact:= fact * x;
        x:= x-1;
        IF x<2 THEN
            exit;
        END IF;
    END LOOP;
    dbms_output.put_line('Factorial is: ' || fact);
END;
```

While Loop

```
DECLARE
    x number := 5;
    fact number :=1;

BEGIN
    WHILE x>=2 LOOP
        fact:= fact * x;
        x:= x-1;
    END LOOP;
```

```
dbms_output.put_line('Factorial is: ' || fact);
END;
```

Snap-Shot:

Basic Loop

The screenshot shows the Oracle Application Express interface. In the SQL Commands editor, there is a syntax error in the PL/SQL code. The code attempts to declare variables and use a WHILE loop to calculate a factorial, but it includes a redundant 'END;' at the end. The error message displayed is:

```
ORA-06550: line 16, column 1:
PLS-00103: Encountered the symbol "WHILE"
ORA-06550: line 28, column 4:
PLS-00103: Encountered the symbol "end-of-file" when expecting one of the following:
begin case declare end exit for goto if loop mod null pragma
raise return select update while with

```

The code in the editor is:

```
DECLARE
    x number := 5;
    fact number :=1;
BEGIN
    WHILE x>=2 LOOP
        fact:= fact * x;
        x:= x-1;
    END LOOP;
    dbms_output.put_line('Factorial is: ' || fact);
END;
```

Below the editor, a Windows taskbar is visible with several open documents and a system clock showing 20:51 on 19-04-2021.

While Loop

The screenshot shows the Oracle Application Express interface. The same PL/SQL code as before is run, but this time it executes successfully. The output shows the factorial of 5 is 120, and the statement is processed in 0.02 seconds.

```
Factorial is: 120
Statement processed.

0.02 seconds
```

At the bottom of the page, a footer note indicates the application version: Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

Below the application window, a Windows taskbar is visible with several open documents and a system clock showing 20:52 on 19-04-2021.

Conclusion: From this practical I learned How to write simple loop & while loop in PL/SQL.

Practical -15

To understand the concept of “select into” and “% type” attribute.

Create an EMPLOYEES table that is a replica of the EMP table. Add a new column, STARS, of VARCHAR2 data type and length of 50 to the EMPLOYEES table for storing asterisk (*).

Create a PL/SQL block that rewards an employee by appending an asterisk in the STARS column for every Rs1000/- of the employee's salary. For example, if the employee has a salary amount of Rs8000/-, the string of asterisks should contain eight asterisks. If the employee has a salary amount of Rs12500/-, the string of asterisks should contain 13 asterisks.

Update the STARS column for the employee with the string of asterisks.

Code:

```

DECLARE
    EMP1 EMP%ROWTYPE;
    CNT NUMBER:=1;
    CNT_TOTAL NUMBER;
    STARSX EMP1.STARS%TYPE;

BEGIN
    CNT_TOTAL:= 7;
    WHILE CNT<=CNT_TOTAL LOOP
        SELECT * INTO EMP1 FROM EMP WHERE EMP_NO = 100+CNT;
        IF EMP1.EMP_SAL > 3000 THEN
            STARSX := '* * * *';
        ELSIF EMP1.EMP_SAL >2000 THEN
            STARSX := '* * *';
    END LOOP;
END;

```

```
ELSIF EMP1.EMP_SAL >1000 THEN
  STARSX := '* *';
```

```
ELSE
  STARSX := '*';
```

```
END IF;
```

```
UPDATE EMP SET STARS = STARSX
WHERE EMP_NO = EMP1.EMP_NO;
```

```
CNT:=CNT+1;
```

```
END LOOP;
```

```
END;
```

Snap-Shot:

The screenshot shows a browser window for Oracle Database Express Edition. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:4104046558774611::NO:::. The page title is "Home > SQL > SQL Commands". A SQL command "SELECT * FROM EMP" is entered in the query editor. The results section displays the following data:

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE	PHONE	NEW_SALARY	INCREASE	STARS
101	Smith	800	500	10	shah	machine learning	ft_mgr	Toronto	105	09-AUG-96	-	920	120	*
102	Snehal	1600	300	10	gupta	data science	lec	Las Vegas	-	14-MAR-96	-	1840	240	**
103	RAMESH	1100	500	20	wales	Machine Learning	mk_mgr	Ontario	105	30-NOV-95	-	1285	165	**
104	Aman	3000	-	10	sharma	Virtual Reality	comp_op	Mexico	12	02-OCT-97	-	3450	450	***
105	Anita	5000	50000	10	patel	Big Data Analytics	comp_op	Germany	107	01-JAN-98	-	5750	750	****
106	Sneha	2450	24500	10	joseph	Big Data Analytics	ft_acc	Melbourne	105	26-SEP-97	-	2817.5	367.5	***
107	Anamika	2975	-	10	jha	Artificial Intelligence	it_prog	New York	-	15-JUL-97	-	3421.25	446.25	***

7 rows returned in 0.03 seconds [CSV Export](#)

Application Express 2.1.0.0.39

The taskbar at the bottom shows several open documents and a search bar.

Conclusion: From this practical I learned how to update columns through PL/SQL.

Practical -16

To perform the concept of cursor

(a) Display all the information of EMP table using %ROWTYPE.

Code:

DECLARE

EMP1 EMP%ROWTYPE;
CNT NUMBER:=1;
CNT1 NUMBER:=7;

BEGIN

DBMS_OUTPUT.PUT_LINE('EMP_NO' || 'EMP_NAME' || 'EMP_SAL' ||
'EMP_COMM' || 'DEPT_NO' || 'L_NAME' || 'DEPT_NAME' || 'JOB_ID' || 'LOCATION'
|| 'MANAGER_ID' || 'HIREDATE');

WHILE CNT < CNT1 LOOP

SELECT * INTO EMP1 FROM EMP WHERE EMP_NO = 100+CNT;

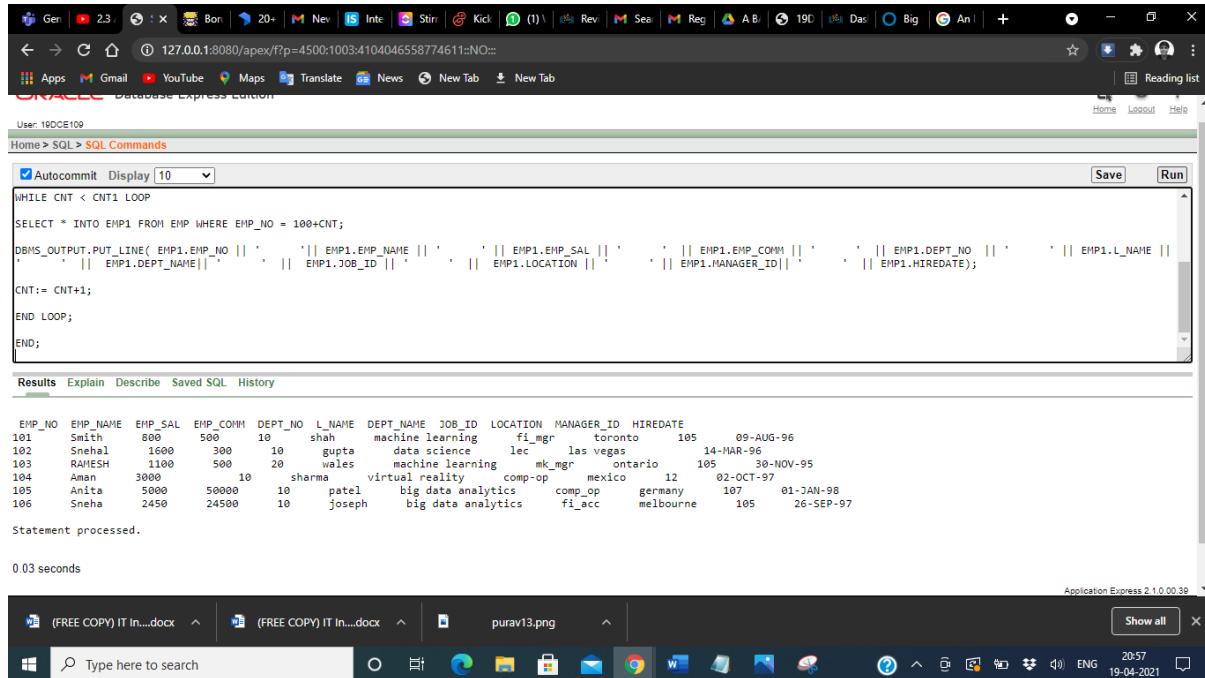
DBMS_OUTPUT.PUT_LINE(EMP1.EMP_NO || ' ' || EMP1.EMP_NAME || ' ' ||
EMP1.EMP_SAL || ' ' || EMP1.EMP_COMM || ' ' || EMP1.DEPT_NO || ' ' ||
EMP1.L_NAME || ' ' || EMP1.DEPT_NAME || ' ' || EMP1.JOB_ID || ' ' ||
EMP1.LOCATION || ' ' || EMP1.MANAGER_ID || ' ' || EMP1.HIREDATE);

CNT:= CNT+1;

END LOOP;

END;

Snap-shot:



```

User: 19DCE109
Home > SQL > SQL Commands
Autocommit: Display: 10 | Save | Run
WHILE CNT < CNT1 LOOP
  SELECT * INTO EMP1 FROM EMP WHERE EMP_NO = 100+CNT;
  DBMS_OUTPUT.PUT_LINE(EMP1.EMP_NO || '||' || EMP1.EMP_NAME || '||' || EMP1.EMP_SAL || '||' || EMP1.EMP_COMM || '||' || EMP1.DEPT_NO || '||' || EMP1.L_NAME || '||' || EMP1.DEPT_NAME || '||' || EMP1.JOB_ID || '||' || EMP1.LOCATION || '||' || EMP1.MANAGER_ID || '||' || EMP1.HIREDATE);
  CNT:= CNT+1;
END LOOP;
END;

```

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	500	10	shah	machine learning	fi_mgr	toronto	105	09-AUG-96
102	Snehal	1600	300	10	gupta	data science	lec	las vegas	105	14-MAR-96
103	RAMESH	1100	500	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95
104	Aman	3000	10	sharma	virtual reality	comp-op	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

Statement processed.

0.03 seconds

Application Express 2.1.0.00.39

(b) Create a PL/SQL block that does the following:

Code:

```

DECLARE
  EMP1 EMP%ROWTYPE;
  X EMP.DEPT_NO%TYPE:= :X;

  CURSOR C1 IS SELECT EMP_NAME,EMP_SAL,MANAGER_ID,L_NAME FROM EMP
  WHERE DEPT_NO = X;

  BEGIN
    DBMS_OUTPUT.PUT_LINE( 'EMP_NAME ' || 'EMP_SAL ' || 'MANAGER_ID ' );
    FOR EMP1 IN C1 LOOP
      DBMS_OUTPUT.PUT_LINE( EMP1.EMP_NAME || '||' || EMP1.EMP_SAL || '||' || EMP1.MANAGER_ID );
    END LOOP;
  END;

```

```

END LOOP;

DBMS_OUTPUT.PUT_LINE('-----');

FOR EMP1 IN C1 LOOP

IF EMP1.EMP_SAL < 1000 THEN
DBMS_OUTPUT.PUT_LINE(EMP1.L_NAME||' Is due for a Raise.);

ELSE
DBMS_OUTPUT.PUT_LINE(EMP1.L_NAME||' Is not due for a Raise.);

END IF;

END LOOP;

END;

```

Snap-shot:

The screenshot shows the Oracle Application Express interface with the following details:

- SQL Commands:** The tab is selected.
- URL:** 127.0.0.1:8080/apex/f?p=4500:1003:141103333850459::NO:1003::
- Autocommit:** Checked.
- Display:** Set to 10.
- Code:**

```

DBMS_OUTPUT.PUT_LINE('EMP_NAME ' || 'EMP_SAL ' || 'MAANGER_ID ');
FOR EMP1 IN C1 LOOP
DBMS_OUTPUT.PUT_LINE(EMP1.EMP_NAME || ' ' || EMP1.EMP_SAL || ' ' || EMP1.MANAGER_ID );
END LOOP;
DBMS_OUTPUT.PUT_LINE('-----');
FOR EMP1 IN C1 LOOP
IF EMP1.EMP_SAL < 1000 THEN

```
- Results:** The results section displays the following data and output:

EMP_NAME	EMP_SAL	MAANGER_ID
Smith	800	105
Aman	3000	12
Anita	5000	107
Sneha	2450	105

shah Is due for a Raise.
sharma Is not due for a Raise.
patel Is not due for a Raise.
joseph Is not due for a Raise.
Statement processed.
- Timing:** 0.02 seconds.
- Version:** Application Express 2.1.0.0 | close
Copyright © 1999, 2006, Oracle. All rights reserved.
- System:** Language: en-us | 03:29 PM | ENG | 30-03-2021

Conclusion: From this Practical I learned how to perform the concept of cursor.

Practical -17

To perform the concept of trigger

Write a PL/SQL block to update the salary where deptno is 10. Generate trigger that will store the original record in other table before updation take place.

Code:

STEP 1 : CREATE BACKUP TABLE

```
CREATE TABLE SALARY(
    EMP_NO NUMBER,
    EMP_SAL    NUMBER
);
```

STEP 2: ADD TRIGGER

```
CREATE OR REPLACE TRIGGER COPY_SALARY
BEFORE UPDATE
ON EMPLOYEE4
FOR EACH ROW
```

DECLARE

```
EMP_SAL NUMBER;
EMP_NO NUMBER;
```

BEGIN

```
EMP_SAL := :OLD.EMP_SAL;
EMP_NO := :OLD.EMP_NO;
```

```
INSERT INTO SALARY
VALUES (EMP_NO,EMP_SAL);
```

END;

STEP 3: UPDATE VALUE IN EMP WHERE DEPT_NO = 10

```
UPDATE EMPLOYEE4 SET EMP_SAL = 3500 WHERE DEPT_NO = 10
```

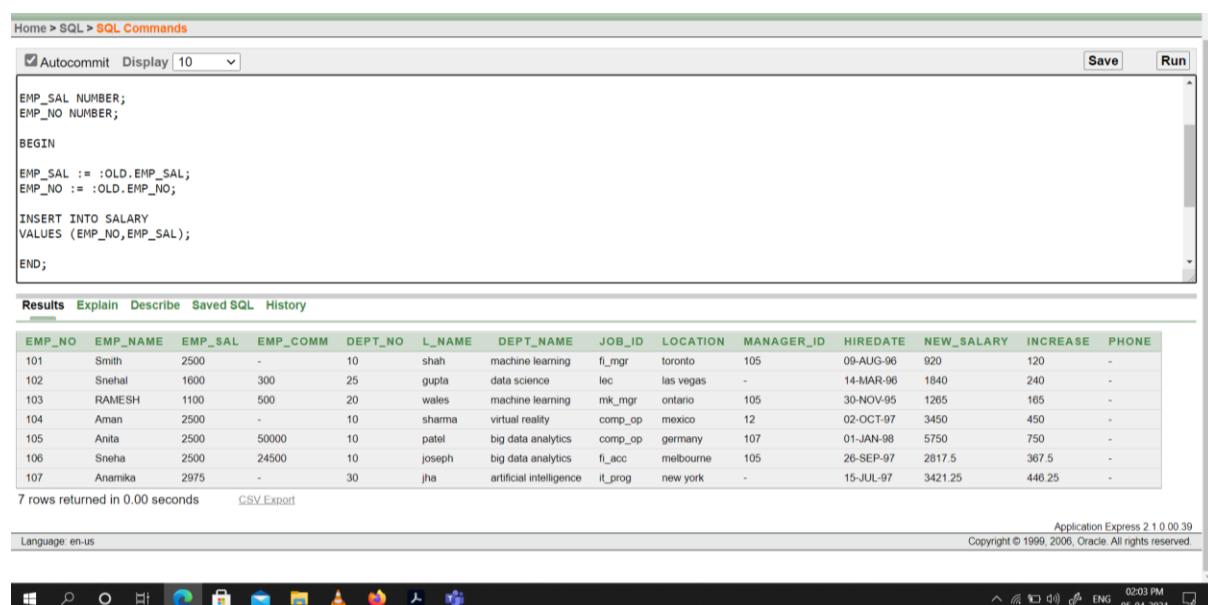
STEP 4: AFTER UPDATE CHECK VALUES IN BOTH TABLES

```
SELECT * FROM EMPLOYEE4
```

```
SELECT * FROM SALARY ORDER BY EMP_NO
```

Snap-shots:

Employee Table Before Command Execution:



The screenshot shows the Oracle Application Express interface. The top navigation bar includes 'Home > SQL > SQL Commands'. Below it is a toolbar with 'Autocommit' checked, 'Display' set to 10, and buttons for 'Save' and 'Run'. The main area contains the following PL/SQL code:

```
EMP_SAL NUMBER;
EMP_NO NUMBER;

BEGIN
  EMP_SAL := :OLD.EMP_SAL;
  EMP_NO := :OLD.EMP_NO;

  INSERT INTO SALARY
  VALUES (EMP_NO,EMP_SAL);

END;
```

Below the code, there's a results grid titled 'Results' with columns: EMP_NO, EMP_NAME, EMP_SAL, EMP_COMM, DEPT_NO, L_NAME, DEPT_NAME, JOB_ID, LOCATION, MANAGER_ID, HIREDATE, NEW_SALARY, INCREASE, and PHONE. The data is as follows:

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE	NEW_SALARY	INCREASE	PHONE
101	Smith	2500	-	10	shah	machine learning	fi_mgr	toronto	105	09-AUG-96	920	120	-
102	Snehal	1600	300	25	gupta	data science	fec	las vegas	-	14-MAR-96	1840	240	-
103	RAMESH	1100	500	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95	1265	165	-
104	Aman	2500	-	10	sharma	virtual reality	comp_op	mexico	12	02-OCT-97	3450	450	-
105	Anita	2500	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-98	5750	750	-
106	Sneha	2500	24500	10	joseph	big data analytics	fi_acc	melbourne	105	28-SEP-97	2817.5	367.5	-
107	Anamika	2975	-	30	jha	artificial intelligence	it_prog	new york	-	15-JUL-97	3421.25	446.25	-

Below the grid, it says '7 rows returned in 0.00 seconds' and has a 'CSV Export' link. The bottom status bar shows 'Application Express 2.1.0.0.39', 'Copyright © 1999, 2006, Oracle. All rights reserved.', 'Language: en-us', and a timestamp '02:03 PM 05-04-2021'.

Employee Table After Command Execution:

The screenshot shows the Oracle Application Express interface. The SQL Commands page displays the following SQL code:

```

Home > SQL > SQL Commands
 Autocommit Display 10  
INSERT INTO SALARY
VALUES (EMP_NO,EMP_SAL);
END;
DELETE SALARY
SELECT * FROM EMPLOYEE4
UPDATE EMPLOYEE4 SET EMP_SAL = 3500 WHERE DEPT_NO = 10
SELECT * FROM SALARY ORDER BY EMP_NO
    
```

The Results section shows the Employee table with 7 rows returned in 0.00 seconds. The table data is:

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE	NEW_SALARY	INCREASE	PHONE
101	Smith	3500	-	10	shah	machine learning	fi_mgr	Toronto	105	09-AUG-96	920	120	-
102	Snehal	1600	300	25	gupta	data science	le_c	las vegas	-	14-MAR-96	1840	240	-
103	RAMESH	1100	500	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95	1265	165	-
104	Aman	3500	-	10	sharma	virtual reality	comp_op	mexico	12	02-OCT-97	3450	450	-
105	Anita	3500	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-96	5750	750	-
106	Sneha	3500	24500	10	joseph	big data analytics	fi_acc	melbourne	105	28-SEP-97	2817.5	367.5	-
107	Anamika	2975	-	30	jha	artificial intelligence	it_prog	new york	-	15-JUL-97	3421.25	446.25	-

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

Application Express 2.1.0.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.
Language: en-us
Windows 10 Taskbar: 02:04 PM 05-04-2021

Salary Table After Command Execution having old values of Emp_Sal:

The screenshot shows the Oracle Application Express interface. The SQL Commands page displays the same SQL code as the previous screenshot.

The Results section shows the Salary table with 4 rows returned in 0.00 seconds. The table data is:

EMP_NO	EMP_SAL
101	2500
104	2500
105	2500
106	2500

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

Application Express 2.1.0.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.
Language: en-us
Windows 10 Taskbar: 02:04 PM 05-04-2021

Conclusion: From this Practical I learned how to use trigger to perform certain tasks.

To solve queries using the concept of View.

- (1) Write a query to create a view for that employee belongs to the location New York.

Code:

```
--create view
CREATE OR REPLACE VIEW NEW_YORK_EMP AS
SELECT * FROM EMPLOYEE4 WHERE LOCATION = 'new york'
--display view
SELECT * FROM NEW_YORK_EMP;
```

Snap-shot:

The screenshot shows the Oracle SQL Developer interface. In the top SQL tab, the following code is entered:

```
CREATE OR REPLACE VIEW NEW_YORK_EMP AS
SELECT * FROM EMPLOYEE4 WHERE LOCATION = 'new york'
SELECT * FROM NEW_YORK_EMP;
```

In the bottom Results tab, the output is displayed as a table:

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE	NEW_SALARY	INCREASE	PHONE
107	Anamika	2975	-	30	jha	artificial intelligence	it_prog	new york	-	15-JUL-97	3421.25	446.25	-

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

At the bottom of the window, the operating system taskbar is visible, showing icons for various applications like File Explorer, Task Manager, and a browser. The system tray shows the date and time as 13-04-2021 04:20 PM.

- (2) Write a query to create a view for all employee with columns emp_id, emp_name, and job_id.

Code:

--create view

```
CREATE OR REPLACE VIEW EMP1 AS  
SELECT EMP_NO,EMP_NAME,JOB_ID FROM EMPLOYEE4  
--display view  
SELECT * FROM EMP1;
```

Snap-shot:

The screenshot shows the Oracle Application Express interface. At the top, there is a toolbar with various icons. Below the toolbar, the main area has tabs for 'Home', 'SQL', and 'SQL Commands'. The 'SQL Commands' tab is active, showing the SQL code for creating a view and executing a select statement. The results section displays a table with 7 rows of data from the EMPLOYEE4 table, mapping it to the EMP1 view. The table has columns: EMP_NO, EMP_NAME, and JOB_ID. The data is as follows:

EMP_NO	EMP_NAME	JOB_ID
101	Smith	fi_mgr
102	Snehal	lec
103	RAMESH	mk_mgr
104	Aman	comp_op
105	Anita	comp_op
106	Sneha	fi_acc
107	Anamika	it_prog

At the bottom of the results page, it says '7 rows returned in 0.02 seconds' and has a 'CSV Export' link. The status bar at the bottom right indicates the application version 'Application Express 2.1.0.00.39', the copyright notice 'Copyright © 1999-2006, Oracle. All rights reserved.', the date '13-04-2021', and the time '04:22 PM'.

(3) Write a query to find the salesmen of the location New York who having salary more than 3000.

Code:

```
--create view  
CREATE OR REPLACE VIEW EMP1 AS
```

```
SELECT EMP_NO,EMP_NAME,JOB_ID FROM EMPLOYEE4  
--display view  
SELECT * FROM EMP1;
```

Snap-shot:

The screenshot shows the Oracle Application Express SQL Commands interface. The SQL editor contains the following code:

```
CREATE OR REPLACE VIEW NEW_YORK_EMP AS  
SELECT * FROM EMPLOYEE4 WHERE EMP_SAL > 3000;  
SELECT * FROM NEW_YORK_EMP ;
```

The results section below the editor shows the message "no data found". At the bottom of the page, the status bar indicates "Language: en-us", "Application Express 2.1.0.00.39", and "Copyright © 1999, 2006, Oracle. All rights reserved". The system tray at the bottom right shows the date and time as "13-04-2021 04:23 PM".

(4) Write a query to create a view to getting a count of how many employee we have at each department.

Code:

```
--create view  
CREATE OR REPLACE VIEW DEPT_COUNTER AS  
SELECT DEPT_NO,COUNT(*)"COUNT" FROM EMPLOYEE4 GROUP BY DEPT_NO;  
--display view  
SELECT * FROM DEPT_COUNTER;
```

Snap-shot:

The screenshot shows the Oracle Application Express SQL Commands interface. At the top, there is a toolbar with 'Autocommit' checked, 'Display' set to 10, and buttons for 'Save' and 'Run'. The SQL code entered is:

```
CREATE OR REPLACE VIEW DEPT_COUNTER AS
SELECT DEPT_NO,COUNT(*)"COUNT" FROM EMPLOYEE4 GROUP BY DEPT_NO;
SELECT * FROM DEPT_COUNTER;
```

Below the code, the results are displayed in a table:

DEPT_NO	COUNT
25	1
30	1
20	1
10	4

Text at the bottom of the results pane says "4 rows returned in 0.01 seconds" and "CSV Export". At the bottom of the window, it says "Language: en-us" and "Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved." The system tray at the bottom right shows icons for network, battery, volume, and date/time (04:26 PM, 13-04-2021).

Conclusion: From This Practical I Learned how to create a view in oracle PL/SQL.

Practical -19

To perform the concept of function and procedure

Write a PL/SQL block to update the salary of employee specified by empid. If record exist, then update the salary otherwise display appropriate message. Write a function as well as procedure for updating salary.

Code:

Using Procedure:

STEP 1: CREATE A PROCEDURE

```
CREATE OR REPLACE PROCEDURE UPDATE_SAL(DEPT_ID IN  
NUMBER,NEW_EMP_SAL IN NUMBER)
```

```
IS
```

```
BEGIN
```

```
UPDATE EMPLOYEE4 SET EMP_SAL = NEW_EMP_SAL WHERE DEPT_NO =  
DEPT_ID;
```

```
DBMS_OUTPUT.PUT_LINE('Updated Salary Is: '|| NEW_EMP_SAL);
```

```
END;
```

STEP 2: EXECUTE THE COMMAND

```
BEGIN
```

```
UPDATE_SAL(20,1250);
```

```
END;
```

Snap-shot:

```

Home > SQL > SQL Commands
 Autocommit Display 10
Save Run
BEGIN
UPDATE EMPLOYEE4 SET EMP_SAL = NEW_EMP_SAL WHERE DEPT_NO = DEPT_ID;
DBMS_OUTPUT.PUT_LINE('Updated Salary Is: '|| NEW_EMP_SAL);
END;
BEGIN
UPDATE_SAL(20,1250);
END;

Results Explain Describe Saved SQL History
Updated Salary Is: 1250
Statement processed.

0.00 seconds

Language: en-us Application Express 2.1 0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.

Windows taskbar: 03:26 PM 06-04-2021

```

Using Function:

STEP 1: CREATE A FUNCTION

```

CREATE OR REPLACE FUNCTION UPDATE_SALARY(DEPT_ID NUMBER,
EMP_NEW_SAL NUMBER)
RETURN NUMBER IS

```

```
BEGIN
```

```
UPDATE EMPLOYEE4 SET EMP_SAL = EMP_NEW_SAL WHERE DEPT_NO =
DEPT_ID;
```

```
RETURN EMP_NEW_SAL;
```

```
END;
```

STEP 2: USE THAT FUCNTION IN OTHER PL/SQL BLOCKS

```
DECLARE
```

```

NEW_EMP_SAL EMPLOYEE4.EMP_SAL%TYPE;
EMP_NO EMPLOYEE4.EMP_NO%TYPE;
UPDATED_SAL NUMBER;

```

```
BEGIN
```

```
    NEW_EMP_SAL := 3079;  
    EMP_NO := 20;  
    UPDATED_SAL:= 0;  
    UPDATED_SAL := UPDATE_SALARY(EMP_NO,NEW_EMP_SAL);
```

```
    DBMS_OUTPUT.PUT_LINE('Updated Salary Is : ' || UPDATED_SAL);
```

```
END;
```

Snap-shot:

The screenshot shows the Oracle Application Express 2.1 interface. In the top window, a PL/SQL block is written:

```
Home > SQL > SQL Commands  
Autocommit Display 10 Save Run  
DECLARE  
    NEW_EMP_SAL EMPLOYEE4.EMP_SAL%TYPE;  
    EMP_NO EMPLOYEE4.EMP_NO%TYPE;  
    UPDATED_SAL NUMBER;  
BEGIN  
    NEW_EMP_SAL := 3089;  
    EMP_NO := 20;  
    UPDATED_SAL:= 0;  
    UPDATED_SAL := UPDATE_SALARY(EMP_NO,NEW_EMP_SAL);
```

Below the code, the results of the execution are displayed:

```
Results Explain Describe Saved SQL History  
Updated Salary Is :3089  
Statement processed.  
0.00 seconds
```

In the bottom status bar, it says "Application Express 2.1 0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved".

Conclusion: From this Practical I learned how to use Procedure & Functions which are pl/sql objects to perform tasks, making code reusable.

Practical -20

To perform the concept of exception handler

Write a PL/SQL block that will accept the employee code, amount and operation. Based on specified operation amount is added or deducted from salary of said employee. Use user defined exception handler for handling the exception.

Code:

```
declare
```

```
OPERATION VARCHAR(10);  
AMOUNT NUMBER;  
EMP_CODE EMPLOYEE4.EMP_NO%TYPE;  
EMP1 EMPLOYEE4%ROWTYPE;
```

```
my_exp EXCEPTION;
```

```
begin
```

```
AMOUNT:= :AMOUNT;  
OPERATION := :OPERATION;  
EMP_CODE := :EMP_CODE;
```

```
SELECT * INTO EMP1 FROM EMPLOYEE4 WHERE EMP_NO = EMP_CODE;
```

```
IF EMP1.EMP_SAL <= 0 AND OPERATION = 'SUB' OR EMP1.EMP_SAL < AMOUNT  
AND OPERATION = 'SUB' THEN  
    RAISE my_exp;  
END IF;
```

```
IF OPERATION = 'ADD' THEN  
    UPDATE EMPLOYEE4 SET EMP_SAL = EMP_SAL+ AMOUNT WHERE EMP_NO =  
    EMP_CODE;  
    DBMS_OUTPUT.PUT_LINE('Added '|| AMOUNT || ' to the salary of employee.');
```

```
ELSIF OPERATION = 'SUB' THEN  
    UPDATE EMPLOYEE4 SET EMP_SAL = EMP_SAL - AMOUNT WHERE EMP_NO =  
    EMP_CODE;  
    DBMS_OUTPUT.PUT_LINE('Deducted '|| AMOUNT || ' from the salary of employee.');
```

```
END IF;
```

```
EXCEPTION
```

```
WHEN NO_DATA_FOUND THEN
```

```
DBMS_OUTPUT.PUT_LINE('No Records Found');
```

```
WHEN my_exp THEN
DBMS_OUTPUT.PUT_LINE('We Can not Deduct Salary');
```

```
end;
```

Snap-shot:

The screenshot shows the Oracle Application Express SQL Commands interface. The code entered is:

```
IF EMP1.EMP_SAL <= 0 AND OPERATION = 'SUB' OR EMP1.EMP_SAL < AMOUNT AND OPERATION = 'SUB' THEN
    RAISE my_exp;
END IF;

IF OPERATION = 'ADD' THEN
    UPDATE EMPLOYEE4 SET EMP_SAL = EMP_SAL + AMOUNT WHERE EMP_NO = EMP_CODE;
    DBMS_OUTPUT.PUT_LINE('Added '|| AMOUNT || ' to the salary of employee.');
ELSIF OPERATION = 'SUB' THEN
    UPDATE EMPLOYEE4 SET EMP_SAL = EMP_SAL - AMOUNT WHERE EMP_NO = EMP_CODE;
    DBMS_OUTPUT.PUT_LINE('Deducted '|| AMOUNT || ' from the salary of employee.');
```

The results section shows the output of the query:

```
Added 1000 to the salary of employee.
1 row(s) updated.

0.02 seconds
```

At the bottom, it says "Language: en-us" and "Copyright © 1999, 2006, Oracle. All rights reserved". The system status bar indicates "04:08 PM 13-04-2021".

Conclusion: From This Practical I Learned how to handle exceptions in oracle.

Practical -21

To perform the concept of package

Code:

STEP 1: CREATE A PACKAGE DECLARATION.

```
CREATE OR REPLACE PACKAGE getEmployeeData AS  
  -- public  
  FUNCTION getRow(Emp_id number) return EMPLOYEE4%ROWTYPE;  
  
END getEmployeeData;
```

STEP 2: CREATE A PACKAGE BODY.

```
CREATE OR REPLACE PACKAGE BODY getEmployeeData AS  
  FUNCTION getRow(Emp_id number) return EMPLOYEE4%ROWTYPE AS  
    -- THIS VARIABLE WILL BE PRIVATE.  
    EMPX EMPLOYEE4%ROWTYPE;  
  
  BEGIN  
  
    SELECT * INTO EMPX FROM EMPLOYEE4 WHERE EMP_NO = Emp_id;  
  
    RETURN EMPX;  
  
  END getRow;  
  
END getEmployeeData;
```

STEP 3: Use the Package In a PL/SQL Block.

```
DECLARE  
  EMPX EMPLOYEE4%ROWTYPE;  
  
BEGIN  
  
  EMPX:= getEmployeeData.getRow(102);  
  
  DBMS_OUTPUT.PUT_LINE('EMP_SAL: '|| EMPX.EMP_SAL);  
  
  -- Similarly all rows and values can be printed.
```

END;

Snap-shot:

The screenshot shows the Oracle Application Express SQL Commands interface. The top navigation bar includes 'Home > SQL > SQL Commands'. The main area contains the following PL/SQL code:

```
DECLARE
  EMPX EMPLOYEE4%ROWTYPE;
BEGIN
  EMPX:= getEmployeeData.getRow(102);
  DBMS_OUTPUT.PUT_LINE('EMP_SAL: '|| EMPX.EMP_SAL);
  -- Similarly all rows and values can be printed.
END;
```

Below the code, the 'Results' tab is selected, showing the output:

```
EMP_SAL: 1600
Statement processed.
```

At the bottom, it shows '0.00 seconds' and the footer includes 'Language: en-us', 'close', 'Application Express 2.1.0.0.39', and 'Copyright © 1999, 2006, Oracle. All rights reserved.'

Conclusion: From This Practical I Learned how to create & use a package in oracle.