Women University Press
Jayoti Publication Desk
ISBN No. : 978-93-90892-04-4

A Hand Book On DBMS Manual

- JV'n Ms. Nishu Sharma

JAYOTI VIDYAPEETH WOMEN'S UNIVERSITY, JAIPUR
UGC Approved Under 2(f) & 12(b) | NAAC Accredited | Recognized by Statutory Councils

Printed by :
JAYOTI PUBLICATION DESK

Published by :
Women University Press
Jayoti Vidyapeeth Women's University, Jaipur

Faculty of Education & Methodology

Title: A Hand Book On DBMS Manual

Author Name: Ms. Nishu Sharma

Published By: Women University Press

Publisher's Address: Jayoti Vidyapeeth Women's University, Jaipur
Vedant Gyan Valley,
Village-Jharna, Mahala Jobner Link Road, NH-8
Jaipur Ajmer Express Way,
Jaipur-303122, Rajasthan (India)

Printer's Detail: Jayoti Publication Desk

Edition Detail:

ISBN: 978-93-90892-04-4

Experiment 1

Aim: Installation of Oracle Database 10g Express Edition (Oracle Database XE) Software.

Users should be allowed to choose Proper DBMS software, install it, configure it and start working on it. Users make a sample tables, modify, alteration , updation of a table , execution of queries then user use SQLPLUS features, use PL/SQL features like cursors on sample database.

Exam of becoming a successful DBA are :

I. SQL Fundamental-I

II. OCA

III. OCP

- Login as the DBA.
- Unlock the UserAccount
- Log in as the UserAccount
- Application creation
- Run yourApplication
- Use Menus of Oracle Database XE

1.Log in as the DBA

The first thing you have to do for run your program , is to log in as the Oracle Database XE Administrator. So, these are the basic steps:

1. First Open the Home Page of Database loginwindow then check Start menu, then choose Programs(or All Programs), after then select Oracle Database 10g Express Edition, and then go to Database Home Page.

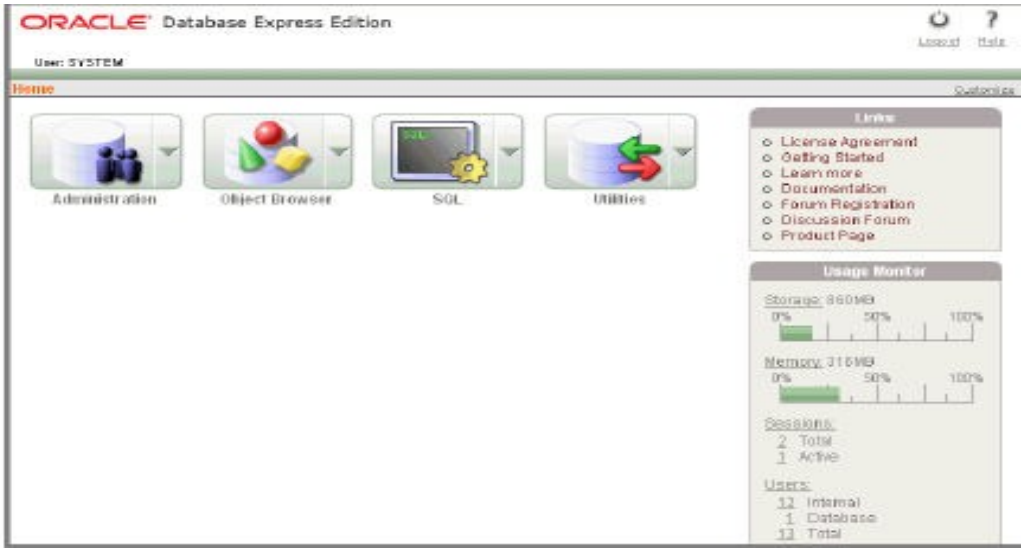
- If you use Linux then you click the Application menu (on Gnome) or the K menu (on KDE), then pointtoOracle Database 10g Express Edition, after then goto Database Home Page.

2. At the Database Home Page login window, enter the following information:

First write the connect on screen .

- Username: Enter system for the username .By default the name of the user is system.
- Password: Enter the password that was specified when Oracle Database XE wasinstalled. You can't Change the password on screen .
- 3. Then Click the Login .

The Oracle Database XE home page appears.



3. Unlock the UserAccount

To make your application, you need to sign in as a database user. Oracle Database XE accompanies with a sampleuser called Management . This user possesses various database tables in a sample that can be utilized to make applications for a fictionalManagement office. Be that as it may, for security reasons, this user record is locked. You need to open this record before you can assemble a sample application.

Steps for unlocking the user account:

1. First check that you arelogged on as the database administrator, as described earlier.
2. Click the Administration icon, and then click DatabaseUsers.
3. Click the Management schema icon to display the user information forManagement.

Management

4. Manage Database User, then enter the followingsettings:

Password and Confirm Password: Enter Tiger for thepassword.

5. Then check the status of account is checked or unlocked.

6.Roles: To check that both connect and resource areenabled.

5. Then Click AlterUser.

Creation of application is successful.

3. Log in as the User Account

Steps for logging the user account:

1. Log out from the DBA account by clicking Logout in the top rightcorner of the Database HomePage.
2. Then clickthe Login button in the window.
3. Enter system for the user name andTiger for password.
4. Then Click Login.

Then Database Home Page appears for user.

4. Application creation

Create an application is a simple way to view and modify your database content. You create this application based on the STUDENT table, which is part of the Management schema.
To create an application which is based on the STUDENT table.

5. Running Your Application:

For running the application there are some steps:

1. Click the Run ApplicationBUTTON.

In the log in page, enter Management for both the User Name and Password. Your application appears, showing the STUDENT table.

3. Explore application.

You can query the STUDENT table, if necessary . To handle the application, use the Developer toolbar at the bottom on the page.

The Developer toolbar offers aneasy way to modify the current page, create a new page, control, or trigger , view , transaction , or editing the links on and off.

4. To exit from yoursoftware and return to Application Builder, click Edit Page on the Developer toolbar.

5. Use Menus of Oracle Database XE:

Use Oracle Database XE menus to operate the basic functions with Oracle Database XE.

- On Windows, from the Start menu, select Programs (or All Programs) and then Oracle Database 10g ExpressEdition.
- On Linux, click the Application menu (on Gnome) or the K menu (on KDE) and then pointto

Oracle Database 10g Express Edition.

connect username/password

where username is the user name may be sys, system and password are the password(Tiger)which was created at the time of Oracle Database XE installation. For further information you can enter the command help at the SQL prompt, after the connection to the database.

Requirement for creating a table

For creating the table you have:

1)Table name 2)Columnname 3) Datatype

Table Limitation:

•Table name must be unique in database schema.

•The table name should begin with a letter and can be 1-30 characters long.

•Maximum 1000 columns you used.

•Name can contain: A-Z, 0-9.

EXPERIMENT 2

AIM: For Connection

Connect

Enter User name: System

Enter Password: Tiger

For User name, User Id from Memory

Select User name, Uid from dual;

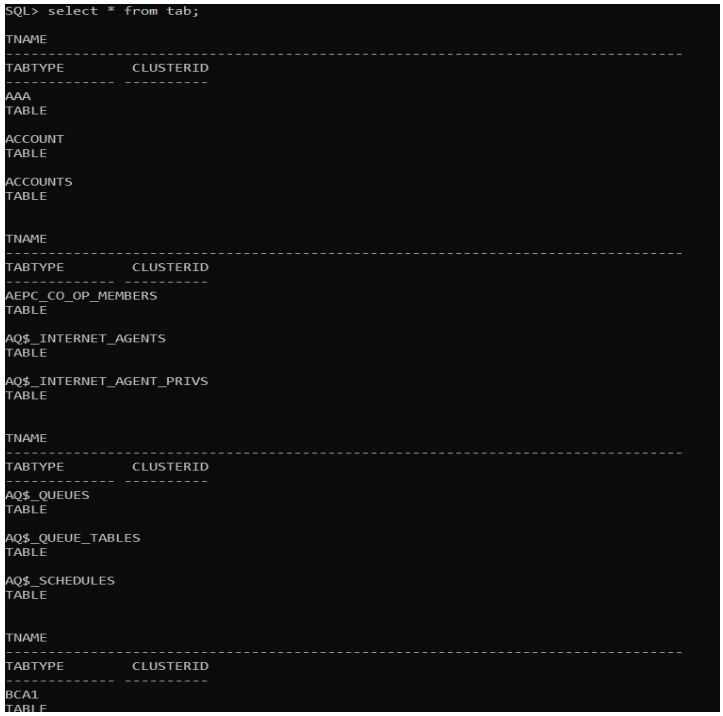
For Time , date from memory

Select Sysdate, Systimestamp from dual;

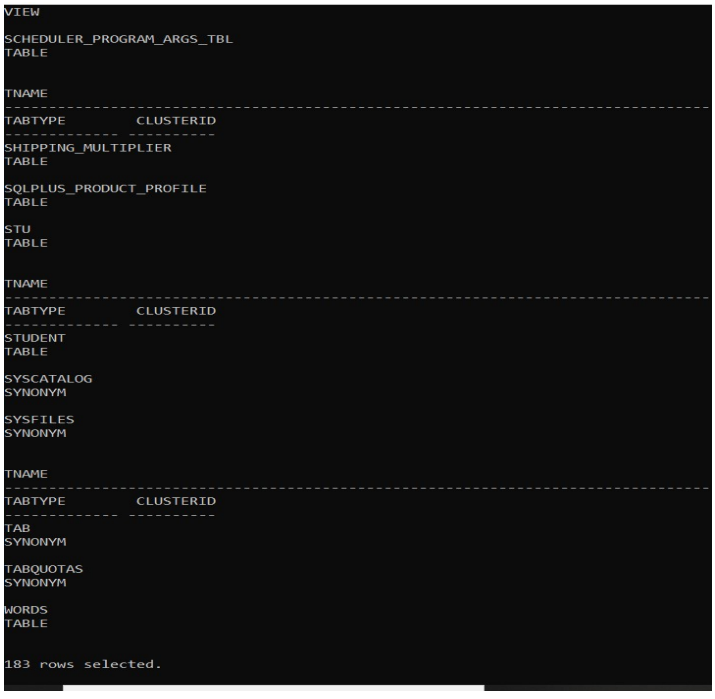
```
SQL> select user name ,uid from dual;

NAME
--------------------------------------------------------------------------------
       UID
----------
SYSTEM
         9


SQL> select sysdate,systimestamp from dual;

SYSDATE
---------
SYSTIMESTAMP
---------------------------------------------------------------------------
01-JAN-21
01-JAN-21 10.40.45.782000 AM +05:30
```

To check How many tables are stored at the time of installation

Select * from tab;



```
SQL> select * from tab;

TNAME
--------------------------------------------------------------------------------
TABTYPE        CLUSTERID
------------- ----------
AAA
TABLE

ACCOUNT
TABLE

ACCOUNTS
TABLE


TNAME
--------------------------------------------------------------------------------
TABTYPE        CLUSTERID
------------- ----------
AEPC_CO_OP_MEMBERS
TABLE

AQ$_INTERNET_AGENTS
TABLE

AQ$_INTERNET_AGENT_PRIVS
TABLE


TNAME
--------------------------------------------------------------------------------
TABTYPE        CLUSTERID
------------- ----------
AQ$_QUEUES
TABLE

AQ$_QUEUE_TABLES
TABLE

AQ$_SCHEDULES
TABLE


TNAME
--------------------------------------------------------------------------------
TABTYPE        CLUSTERID
------------- ----------
BCA1
TABLE
```



```
VIEW

SCHEDULER_PROGRAM_ARGS_TBL
TABLE


TNAME
--------------------------------------------------------------------------------
TABTYPE        CLUSTERID
------------- ----------
SHIPPING_MULTIPLIER
TABLE

SQLPLUS_PRODUCT_PROFILE
TABLE

STU
TABLE


TNAME
--------------------------------------------------------------------------------
TABTYPE        CLUSTERID
------------- ----------
STUDENT
TABLE

SYSCATALOG
SYNONYM

SYSFILES
SYNONYM


TNAME
--------------------------------------------------------------------------------
TABTYPE        CLUSTERID
------------- ----------
TAB
SYNONYM

TABQUOTAS
SYNONYM

WORDS
TABLE


183 rows selected.
```

EXPERIMENT 3

AIM: TO CREATE TABLE AND CHECK THE DESCRIPTION OF TABLE.

SYNTAX:

create table<table name> (<column1><Data type>,<column2><Data type>, <column3><Data type>,<column4><Data type>);

SQL>create table employee (empno number(10),ename varchar2(10), job varchar2(10), deptno number(10)) ;

Tablecreated.

In sql*plus, we use **clscr** for clear the screen.

DESCRIBE: Command will give us with what columns we created table and their data type.

Syntax:

SQL>create table <table name>(attributename 1 datatype(size) ,attributename2 datatype(size),attributename3 datatype(size));

SQL> create table Employee(empno number ,ename varchar2(10), job varchar2(10) ,deptno number(10));

Expected Outcome:

Table created.

SQL>Desc tablename;

For describe the table

SQL>DESC employee;

```
SQL> create table Employee (empno number, ename varchar2(10), job varchar2 (10) , deptno number(10)) ;

Table created.

SQL> desc Employee;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPNO                                              NUMBER
 ENAME                                              VARCHAR2(10)
 JOB                                                VARCHAR2(10)
 DEPTNO                                             NUMBER(10)
```

EXPERIMENT 4

AIM: INSERTING VALUES IN TO TABLE.

SYNTAX:

INSERT into<table name> [list of columns] values (list of values);

QUERIES:

SQL>insert into Employee Values(1,'Nishu','Faculty',81);

1RowInserted.

SQL>insert into Employee Values(2,'Bhawana','Faculty',81);

1 RowInserted.

SQL> insert into Employee Values(3,'Swarnima','Faculty',81);

1 row created.

SQL> insert into Employee Values(4,'Indu','Faculty',81);

1 row created.

SQL> insert into Employee Values(5,'Paridhi','Faculty',81);

1 row created.

After insert we check our table data by using select command.

Syntax:Select * fromtablename;

SQL>**Select * from Employee;**

Output:

```
SQL> insert into  Employee Values(1,'Nishu','Faculty',81);

1 row created.

SQL> insert into  Employee Values(2,'Bhawana','Faculty',81);

1 row created.

SQL> insert into  Employee Values(3,'Swarnima','Faculty',81);

1 row created.

SQL> insert into  Employee Values(4,'Indu','Faculty',81);

1 row created.

SQL> insert into  Employee Values(5,'Paridhi','Faculty',81);

1 row created.

SQL> select * from Employee;

     EMPNO ENAME      JOB            DEPTNO
---------- ---------- ---------- ----------
         1 Nishu      Faculty            81
         2 Bhawana    Faculty            81
         3 Swarnima   Faculty            81
         4 Indu       Faculty            81
         5 Paridhi    Faculty            81
```

INSERTING DATA INTO REQURIED COLUMNS:

SQL> insert into Employee (empno,ename) values(6,'Ayushi');

1 row created.

```
SQL> insert into Employee (empno,ename) values(6,'Ayushi');

1 row created.

SQL> insert into Employee values(&empno,'&ename','&job',&deptno);
Enter value for empno: 7
Enter value for ename: muskaan
Enter value for job: faculty
Enter value for deptno: 81
old   1: insert into Employee values(&empno,'&ename','&job',&deptno)
new   1: insert into Employee values(7,'muskaan','faculty',81)
```

EXPERIMENT 5

AIM:SELECTING VALUES FROM A TABLE

Syntax: Select column name from tablename;

Queries:

Select ename from Employee;

```
SQL> select ename from Employee ;

ENAME
----------
Nishu
Bhawana
Swarnima
Indu
Paridhi
```

HERE INSTEAD OF TYPING ALL THE COLUMN NAMES WE TYPE "*"

SQL> Select * from Employee;

```
SQL> select * from Employee;

     EMPNO ENAME      JOB            DEPTNO
---------- ---------- ---------- ----------
         1 Nishu      Faculty            81
         2 Bhawana    Faculty            81
         3 Swarnima   Faculty            81
         4 Indu       Faculty            81
         5 Paridhi    Faculty            81
```

SELECTING WITH WHERE CLAUSE:

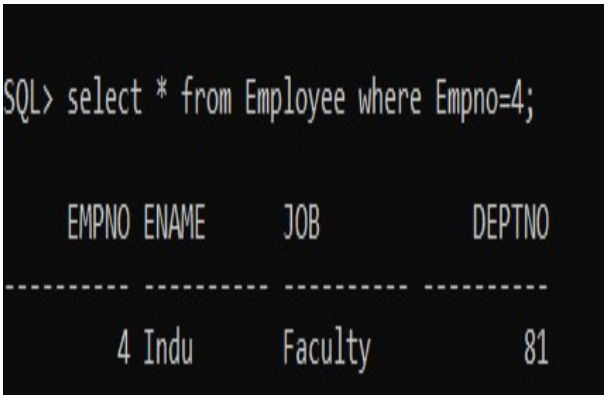
Whenever we select using where clause we get particular information depends on the column you specify in whereclause.

Syntax:

SQL>Select * from tablename where condition(given);

Query:

SQL>Select * from Employee where Empno=4;



EXPERIMENT 6

AIM: UPDATE STATEMENT

While updating a table if you don't give where clause whole table will be updated.

SQL> updatetablenameset column name = 'values ' where column name = values;

Queries:

SQL> update Employee set job = 'faculty' where empno=6;

1 row updated

```
SQL> update Employee set job = 'faculty' where empno=6;

1 row updated.

SQL> SELECT * FROM EMPLOYEE;

     EMPNO ENAME      JOB            DEPTNO
---------- ---------- ---------- ----------
         1 Nishu      Faculty            81
         2 Bhawana    Faculty            81
         3 Swarnima   Faculty            81
         4 Indu       Faculty            81
         5 Paridhi    Faculty            81
         6 Ayushi     faculty

6 rows selected.
```

UPDATE TABLE USING WHERE CLAUSE:

Whenever we give where clause in updating only that column corresponding rows will be updated.

Syntax:

SQL> Update tablename set column name=values where column name=values;

Queries:

SQL> update Employee set job='faculty' where empno=7;

1 row updated.

```
SQL> update Employee set job='faculty' where empno=7;

1 row updated.
```

EXPERIMENT 7

AIM: DELETING DATA FROM A TABLE

If you want to delete data from a table, then the query will be:

SYNTAX: Delete from table name ;

SQL>Delete from Employee**;**

If you want to delete particular data from a table, then the query will be:

SQL>Delete from Employee WHERE EMPNO=7;

```
SQL> Delete from  Employee  WHERE EMPNO=7;

1 row deleted.

SQL> select * from employee;

     EMPNO ENAME      JOB             DEPTNO
---------- ---------- ---------- ----------
         1 Nishu      Faculty            81
         2 Bhawana    Faculty            81
         3 Swarnima   Faculty            81
         4 Indu       Faculty            81
         5 Paridhi    Faculty            81
         6 Ayushi     faculty

6 rows selected.
```

PRACTICE

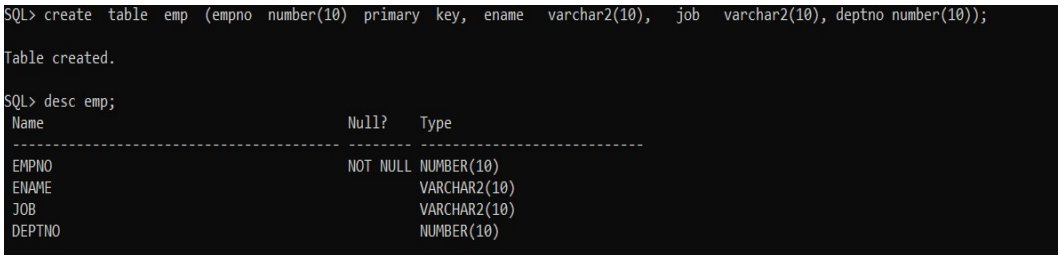
1. Create tables department and employee with requiredconstraints. QUERY:

create table emp (empno number(10) primary key, ename varchar2(10), job varchar2(10), deptno number(10));

EXPECTED OUTPUT:

Table Created

OUTPUT:



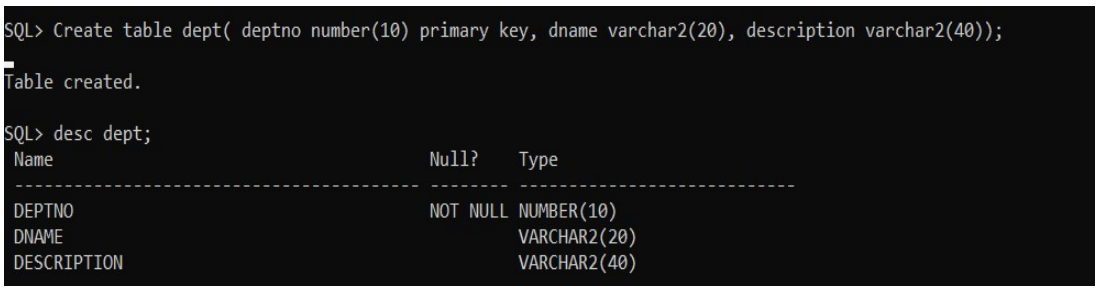
QUERY:

Create table dept(deptno number(10) primary key, dname varchar2(20), description varchar2(40));

EXPECTED OUTPUT:

Table Created

OUTPUT:



2. Add the remaining columns in the existing separately by using SQLcommand QUERY

Alter table emp (add basics number(12));

EXPECTED OUTPUT:

Table Altered

OUTPUT:

```
SQL> Alter table emp add basics number(12);

Table altered.

SQL> desc emp;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPNO                                     NOT NULL NUMBER(10)
 ENAME                                              VARCHAR2(10)
 JOB                                                VARCHAR2(10)
 DEPTNO                                             NUMBER(10)
 BASIC                                              NUMBER(10)
 BASICS                                             NUMBER(12)
```

3. Basic column should not be null QUERY:

Create table employee1 (empno number(10) not null, basic number(10) not null, ename varchar2(10));

```
SQL> Create table employee1 (empno number(10) not null, basic number(10) not null, ename varchar2(10));

Table created.

SQL> desc employee1;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPNO                                     NOT NULL NUMBER(10)
 BASIC                                     NOT NULL NUMBER(10)
 ENAME                                              VARCHAR2(10)
```

Insert into employee1 values (null,null,'akash');

Note: if you tried to insert null values in those two columns , but it has not accepted and provide error.

EXPECTED OUTPUT:

Cannot insert null value in basics column(not null constraint violated)

OUTPUT:

```
SQL> Insert into employee1 values (null,null,'akash');
Insert into employee1 values (null,null,'akash')
                             *

ERROR at line 1:
ORA-01400: cannot insert NULL into ("SYSTEM"."EMPLOYEE1"."EMPNO")
```

1. Add constraint in which basics should not be less than 500.

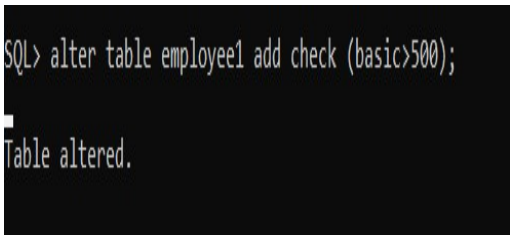
QUERY:

alter table employee1 add check (basic>500)

EXPECTED OUTPUT:

Table Altered

OUTPUT:



2. Empno should be unique and has to be generatedautomatically. QUERY:

Create or replace sequence s1 Increment by 1

Start with 501

End with 590 Nocache

Nocycle;

Insert into employee values (s1.nextval,'siva', 9000, 900,4500);

Insert into employee values (s1.nextval,'siva', 9000, 900,4500);

EXPECTED OUTPUT:

Sequence created.

6. The default value for basic is 0

QUERY:

Alter table employee1 modify basic DEFAULT 0;

EXPECTED OUTPUT:

Table Altered

OUTPUT:

```
SQL> Alter table employee1 modify basic DEFAULT 0;

Table altered.
```

7. when the employees dailywagers are to be added the constraint then their amount should be greater than or equal to 500 should bedropped.

QUERY:

Alter table employee1 Add dailywagers number (10);

Alter table employee1 add check (dailywagers>=5000);

EXPECTED OUTPUT:

Table Altered

OUTPUT:

```
SQL> Alter table employee1 Add dailywagers number (10);

Table altered.

SQL> Alter table employee1 add check (dailywagers>=5000);

Table altered.
```

5. Display the information of the departments with description of thefields.

QUERY:

desc dept;

EXPECTED OUTPUT

```
SQL> desc dept;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DEPTNO                                    NOT NULL NUMBER(10)
 DNAME                                              VARCHAR2(20)
 DESCRIPTION                                        VARCHAR2(40)
```

6. Display the average deptno of all the departments. QUERY:

first you have to insert all the values after then insert all the command

```
SQL> insert into dept values(&deptno ,'&dname','&description');
Enter value for deptno: 1
Enter value for dname: cs
Enter value for description: computer science
old   1: insert into dept values(&deptno ,'&dname','&description')
new   1: insert into dept values(1 ,'cs','computer science')

1 row created.

SQL> insert into dept values(&deptno ,'&dname','&description');
Enter value for deptno: 2
Enter value for dname: civil
Enter value for description: civil engineering
old   1: insert into dept values(&deptno ,'&dname','&description')
new   1: insert into dept values(2 ,'civil','civil engineering')

1 row created.

SQL> insert into dept values(&deptno ,'&dname','&description');
Enter value for deptno: 3
Enter value for dname: ec
Enter value for description: electronics engineering
old   1: insert into dept values(&deptno ,'&dname','&description')
new   1: insert into dept values(3 ,'ec','electronics engineering')

1 row created.
```

7. Select avg(deptno) fromdept;

EXPECTEDOUTPUT:

deptno

-------- 2

OUTPUT:

```
SQL> Select avg(deptno) from dept;

AVG(DEPTNO)
-----------
          2
```

8. Display the average deptno description wise. QUERY:

Select avg(deptno) from emp group by description;

OUTPUT:

```
SQL> Select avg(deptno) from dept group by description;

AVG(DEPTNO)
-----------
          5
          2
          1
          3
```

9. Display the maximum deptno.

QUERY:

Select max(deptno) from dept;

OUTPUT:

```
SQL> Select * from dept;

    DEPTNO DNAME              DESCRIPTION
---------- ------------------ ---------------------------------------
         1 cs                 computer science
         2 civil              civil engineering
         3 ec                 electronics engineering
         4 me                 mechanical Engineer
         6 me                 mechanical Engineer
```

10. Commit the changes whenever required and rollback ifnecessary. QUERY:

Commit ;

EXPECTED OUTPUT:

Commit complete.

roll back;

EXPECTED OUTPUT:

Rollback complete.

select * from employee1 ;

EXPECTED OUTPUT:

No rows selected

```
SQL> commit;

Commit complete.

SQL> roll back
Rollback complete.
SQL>
```

11. Use substitution variables to insert valuesrepeatedly. QUERY:

CREATE TABLE departments (department_id number(10) not null, department_name varchar2(50) not null);

Insert into departments values (&department_id,'&department_name');

EXPECTED OUTPUT:

1 row inserted

OUTPUT:

```
SQL> INSERT INTO departments VALUES(&department_id,'&department_name');
Enter value for department_id: 1
Enter value for department_name: cs
old   1: INSERT INTO departments VALUES(&department_id,'&department_name')
new   1: INSERT INTO departments VALUES(1,'cs')

1 row created.
```

12. Find the dept whose deptno is between 1 and 3 but not exactly 7.

QUERY:

select dname from dept where deptno between 1 and 3 and not deptno=7;

OUTPUT:

```
SQL> select dname from dept where deptno between 1 and 3 and not deptno=7;

DNAME
--------------------
cs
civil
ec
```

13. Find the employee1 whose name contains 'e'. QUERY:

Select empno,ename from employee1 Where ename like '%e%';

OUTPUT:

```
SQL> Select deptno,dname from dept Where dname like '%e%';

    DEPTNO DNAME
---------- --------------------
         3 ec
         4 me
         6 me
```

14. Try to delete a particular deptno. What happens if there are department in it and if there are nodepartment.

QUERY:

SQL> delete from dept where deptno=1;

1 row deleted.

SQL> select * from dept;

DEPTNO	DNAME	DESCRIPTION
2	civil	civil engineering
3	ec	electronics engineering
4	me	mechanical Engineer

6 me mechanical Engineer

EXPECTED OUTPUT:

1 row deleted.

OUTPUT:

```
SQL> select * from dept;

    DEPTNO DNAME                DESCRIPTION
---------- -------------------- ----------------------------------------
         1 cs                   computer science
         2 civil                civil engineering
         3 ec                   electronics engineering
         4 me                   mechanical Engineer
         6 me                   mechanical Engineer

SQL> delete from dept where deptno=1;

1 row deleted.

SQL> select * from dept;

    DEPTNO DNAME                DESCRIPTION
---------- -------------------- ----------------------------------------
         2 civil                civil engineering
         3 ec                   electronics engineering
         4 me                   mechanical Engineer
         6 me                   mechanical Engineer
```

SQL> delete from dept where deptno=7;

0 row deleted.

SQL> select * from dept;

```
SQL> delete from dept where deptno=7;

0 rows deleted.
```

15. Create alias for columns inqueries.

QUERY:

select deptno**as** deptnumber,dnamedepartmentname from dept;

EXPECTED OUTPUT:

DEPTNUMBER	DEPARTMENTNAME
2	civil
3	ec
4	me
6	me

OUTPUT:

```
SQL> select deptno as deptnumber,dname departmentname from dept;

DEPTNUMBER DEPARTMENTNAME
---------- --------------------
         2 civil
         3 ec
         4 me
         6 me
```

16. List the department according to ascending order ofdname.

QUERY:

a) Select * from dept Order by dnameasc;

DEPTNO	DNAME	DESCRIPTION
2	civil	civil engineering
3	ecelectronics	engineering
4	me	mechanical Engineering
6	me	mechanical Engineer

b) Select * from dept Order by dname ;

EXPECTED OUTPUT:

DEPTNO DNAME DESCRIPTION

---------- ---------- ----------

2 civil civil engineering

3ecelectronicsengineering

4 me mechanical Engineering

6 me mechanical Engineer

```
SQL> Select * from dept Order by dname asc;

    DEPTNO DNAME              DESCRIPTION
---------- ------------------ ----------------------------------------
         2 civil              civil engineering
         3 ec                 electronics engineering
         4 me                 mechanical Engineer
         6 me                 mechanical Engineer

SQL> Select * from dept Order by dname ;

    DEPTNO DNAME              DESCRIPTION
---------- ------------------ ----------------------------------------
         2 civil              civil engineering
         3 ec                 electronics engineering
         4 me                 mechanical Engineer
         6 me                 mechanical Engineer
```

17. List the department according to ascending order of dname in eachdepartment.

QUERY:

select deptno ,dname from dept order by deptno ;

EXPECTED OUTPUT:

SQL> select deptno ,dname from dept order by deptno ;

DEPTNO DNAME

---------- ----------

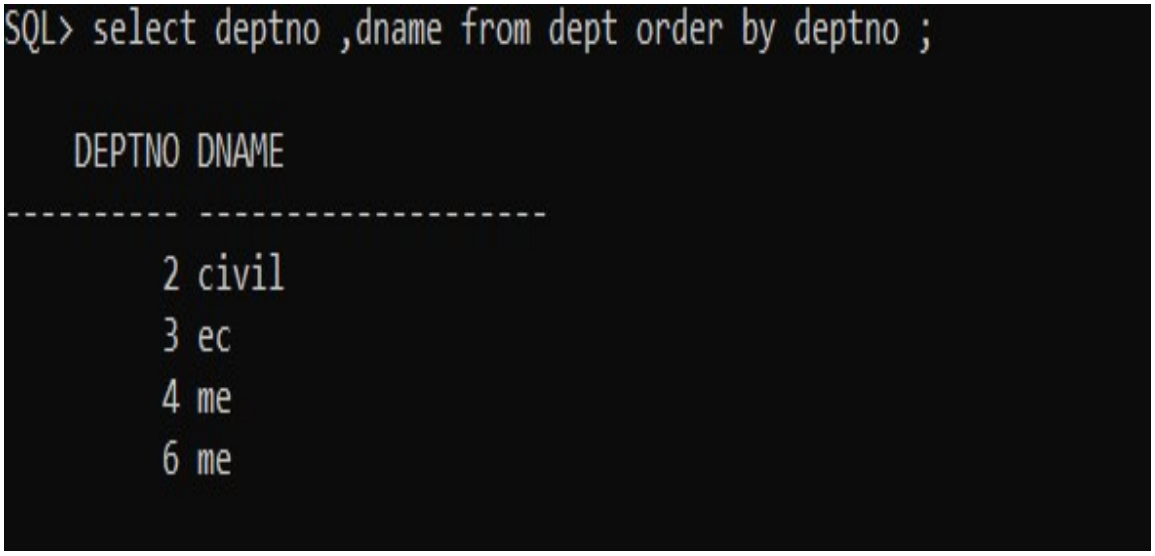
2 civil

3ec

4 me

6 me

OUTPUT:



18. Use of **'&' in insert command**

QUERY:

SQL> Insert into departments VALUES(&department_id,'&department_name');

Enter value for department_id: 9

Enter value for department_name: civil

old 1: Insert into departments VALUES(&department_id,'&department_name')

new 1: Insert into departments VALUES(9,'civil')

1 row created.

Note: using & in inserting time is asking for data at the compile time.

EXPECTED OUTPUT:

1 row inserted

OUTPUT:

```
SQL> INSERT INTO departments VALUES(&department_id,'&department_name');
Enter value for department_id: 9
Enter value for department_name: civil
old   1: INSERT INTO departments VALUES(&department_id,'&department_name')
new   1: INSERT INTO departments VALUES(9,'civil')

1 row created.
```

REFERENCES:

1. C. J. Date, A. Kannan and S. Swamynathan, *An Introduction to Database Systems*, Pearson Education, Eighth Edition, 2009.

2. Abraham Silberschatz, Henry F. Korth and S. Sudarshan, *Database System Concepts*, McGraw-Hill Education (Asia), Fifth Edition, 2006.

3. Shio Kumar Singh, *Database Systems Concepts, Designs and Application*, Pearson Education, Second Edition, 2011.

4. Peter Rob and Carlos Coronel, *Database Systems Design, Implementation and Management*, Thomson Learning-Course Technology, Seventh Edition, 2007.

5. Patrick O'Neil and Elizabeth O'Neil, *Database Principles, Programming and Performance*, Harcourt Asia Pte. Ltd., First Edition, 2001.

6. Atul Kahate, *Introduction to Database Management Systems*, Pearson

Contact Us:

University Campus Address:

Jayoti Vidyapeeth Women's University

Vadaant Gyan Valley, Village-Jharna, Mahala Jobner Link Road,

Jaipur Ajmer Express Way, NH-8, Jaipur- 303122, Rajasthan (INDIA)

(Only Speed Post is Received at University Campus Address, No. any Courier Facility is available at Campus Address)

Pages	: 30
Book Price	: ₹ 150/-

Women University Press

Jayoti Publication Desk

Year & Month of Publication- 3/3/2021