DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

LAB MANUAL

COURSE : B.Tech (CSE)
SEMESTER :V
SUBJECT : DBMS LAB
SUBJECT CODE :NCS-552

PREPARED BY:

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

LIST OF EXPERIMENTS

COURSE:B.Tech **SEMESTER:**V **SESSION:** 2015-16(ODD SEM)
BRANCH: CSE
SUBJECT CODE & NAME: NCS-552, DATABASE MANAGEMENT SYSTEM LAB

SI. No.	NAME OF EXPERIMENT
1	Application of Creation, Deletion,Insertion, Updation, Alter , Destroy, Rename Commands a. Create Table CLIENT_MASTER, PRODUCT_MASTER, SALESMAN_MASTER b. Insert relevant data into the tables c. Retrieve data from table CLIENT_MASTER, PRODUCT_MASTER, SALESMAN_MASTER d. Update records in the tables CLIENT_MASTER, PRODUCT_MASTER, SALESMAN_MASTER e. Delete records from tables CLIENT_MASTER, PRODUCT_MASTER, SALESMAN_MASTER f. Create a new table with already existing table g. Insert data into a new table from already existing table h. Alter structure of the tables CLIENT_MASTER, PRODUCT_MASTER, SALESMAN_MASTER i. Destroy a table along with its data j. Rename SALESMAN_MASTER k. Show the structure of the table product_master
2	Application of operators, oracle functions, date conversion functions: a. Application of operator on a column b. Application of operator and renaming of column c. Use of AND operator d. Use of OR e. Use of BETWEEN f. Use of NOT BETWEEN g. USE of LIKE h. USE of LIKE with OR i. Use of IN j. Use of NOT IN k. Use of Oracle functions l. Use of Conversion functions like TO_CHAR, TO_DATE etc.
3	Execute the following queries: a. The NOT NULL b. The UNIQUE Constraint c. The PRIMARY KEY Constraint d. The FOREIGN KEY Constraint e. The CHECK Constraint f. Defining Integrity constraints in ALTER table command

4	Execute queries related to Group By and Having Clause on tables SALES_ORDER.
5	Execute Nested Queries on tables CLIENT_MASTER, PRODUCT_MASTER, SALESMAN_MASTER, SALES_ORDER, SALES_ORDER_DETAILS
6	Execute queries related to Exists, Not Exists, Union, Intersection, Difference, Join on tables CLIENT_MASTER, PRODUCT_MASTER, SALESMAN_MASTER, SALES_ORDER, SALES_ORDER_DETAILS
7	Procedures, Functions & Packages: a. Write a simple procedure to display a message "Good Day to You" b. Code a function to return the Square of a given number. c. Create a package to include the following: i. A named procedure to list the Product_no of products with Quantity_on_hand as 5 in PRODUCT_MASTER table. ii. A function which returns the max maximum Quantity_on_hand for a given product.
8	Write a TRIGGER to ensure that CLIENT_MASTER TABLE does not contain duplicate of null values in CLIENT_NO column.
9	Write a CURSOR to display list of clients in the CLIENT_MASTER table.
10	Installing Oracle
11	Creating Entity-Relationship Diagram using case tools.
12*	PL/SQL programming Write a PL/SQL block code to print the squares of numbers upto 99. Write a PL/SQL block code to insert data into table CUSTOMER

*Innovative Exeriments
Virtual Lab Experiments

Experiment-1

ProgramName: Application of Creation, Deletion, Insertion, Updation, Alter, Destroy, Rename Commands

a. Create Table CLIENT_MASTER, PRODUCT_MASTER, SALESMAN_MASTER
b. Insert relevant data into the tables
c. Retrieve data from table CLIENT_MASTER, PRODUCT_MASTER, SALESMAN_MASTER
d. Update records in the tables CLIENT_MASTER, PRODUCT_MASTER, SALESMAN_MASTER
e. Delete records from tables CLIENT_MASTER, PRODUCT_MASTER, SALESMAN_MASTER
f. Create a new table with already existing table
g. Insert data into a new table from already existing table
h. Alter structure of the tables CLIENT_MASTER, PRODUCT_MASTER, SALESMAN_MASTER
i. Destroy a table along with its data
j. Rename SALESMAN_MASTER
k. Show the structure of the table product_master

Theory Concept: This program intends to demonstrate application of various commands used for data definition and data manipulation language.

Implementation:

Q-a) Create the tables described below

Table Name : **CLIENT_MASTER**
Description : Used to store the client information

Column Name	Data Type	Size
Client_no	Varchar2	6
Name	Varchar2	20
Address1	Varchar2	30
Address2	Varchar2	30
City	Varchar2	15
Pincode	Number	6
State	Varchar2	15

Table Name : **PRODUCT_MASTER**
Description : Used to store the product information

Column Name	Data Type	Size
Product_no	Varchar2	6
Description	Varchar2	20
Quantity_on_hand	Number	8
Reorder_level	Number	8
Cost_price	Number	8,2
Selling_Price	Number	8,2

Table Name : **SALESMAN_MASTER**
Description : Used to store the salesman information working for the Company

Column Name	**Data Type**	**Size**
Salesman_no	Varchar2	6
Name	Varchar2	20
Address1	Varchar2	30
Address2	Varchar2	30
City	Varchar2	15
Pincode	Number	6
State	Varchar2	15
Date_of_joining	Date	
Salary	Number	8,2

Ans:create table client_master(client_no. varchar(6), name varchar(20), city varchar(15), pincode number(6), state varchar(5));
create table product_master(product_no. varchar(6), description varchar(20), quantity_on_headnumber(8), cost price number(8,2), selling price number(8,2));
create table SALESMAN_MASTER(salesman_novarchar(6), name varchar(20),address1 varchar(30), address2 varchar(30), city varchar(15), pincode number(6), state varchar(15), date_of_joining date, salary number(8,2));
Output: Table created

Q-b) Insert data items into the tables created above
Ans : insert into client_master(client_no. , name , city , pincode , state) values('&client_no.' , '&name' , '&city' , '&pincode' , '&state');

Client_no	Name	City	Pincode	State
3	Akshat	Ghaziabad	23456	UP
4	Dhawal	Ghaziabad	24364	UP
5	Akshansh	Dhampur	246761	UP
6	David	Hapur	35498	UP

Output: 4 rows created

insert into product_master
(product_no.,description,quantity_on_hand,cost_price,selling_price)
values('&product_no','&description','&quantity_on_hand','&cost_price','&selling_price');

Product no.	Description	quantity_on_hand	cost_price	selling_price
1	Chair	5	1000	1250
2	Table	5	5000	6000

Output: 2 rows created

Q-c) Retrieve records from the above tables as follows

a) Find out the names of all the clients
b) Retrieve the entire contents of the client_master table
c) Retrieve description, cost_price and selling_price from product master

d) Retrieve clients from client_master table who live in 'Dhampur'

e) Retrieve distinct city from client_master table

f) Retieveproduct_no., description and cost_price from product_masterwhich are ordered by cost_price.

Ans:

a) select name from client_master ;

Output:

NAME

Akshat
Dhawal
Akshansh
David

4 rows selected

b) select * from client_master ;

Output:

Client_no	Name	City	Pincode	State
3	Akshat	Ghaziabad	23456	UP
4	Dhawal	Ghaziabad	24364	UP
5	Akshansh	Dhampur	246761	UP
6	David	Hapur	35498	UP

4 rows selected

c) select description, cost_price, selling_price from product master ;

Output:

DESCRIPTION COST_PRICE SELLING_PRICE

Chair 1000 1250
Table 5000 6000

2 rows selected

d) select name from client_master where city = 'Dhampur';

Output:

NAME

Akshansh

1 row selected

e) select distinct city from client_master ;

Output:

CITY

Ghaziabad

Dhampur

Hapur

3 row selected

f) select product_no., description, cost_price from product_master order by cost_price ;

Output:

Product No.	Description	Cost Price
4	Mirror	250
1	Chair	1000

2 rows selected

Q-d) Update the records in the tables above as follows

a) Change the city of client_no 'C1' to dhampur
b) Change the cost price from 250 to 500 in product_master;

Ans :

a) Update client-master set city='Noida' where city='Dhampur';

Output:1 row selected

b) Update product_master
set cost _price=500 where cost _price=250;

Output: 1 row updated

Q-e) Delete records in the table above as follows

Delete all records for the product_master table

Ans:

Delete from product_master;

Output: 5 rows deleted

Q-f) Create table New_client from client_master with the fields Client_no, name

Ans :

create table new-client(client_no,name) as (select client_no,name from client_master);

Output: Table created

Q-g) Insert into table new_client data from table client_master where city = 'Hapur'

Ans: insert into new_client select client_no.,name from client-master where city='Hapur';

Output:

Client_no.	Name
6	David

1 row created

Q-h) Alter the table structures as instructed

a) Add a column called Telephone_no of data type number and size = 10 to the client_master table
b) Change the size of the description column in product_master to 25
c) Drop the column Telephone_no from the table client_master

Ans:

a) alter table client_master add(telephone_number(10));

Output: Table created

b) alter table product_master modify(description varchar(25));

Output: Table created

c) alter table client_master drop column telephone_no;

Output: Table created

Q-i) Destroy the table new_client along with its data

Ans:

Drop table new_client;

Output: Table dropped

Q-j) Change the name of the product_master to products

Ans:

renameproduct_master to products;

Output: Table renamed

Q-k) Show the structure of the table product_master

Ans :

Describe product1

Output:

Column	Description
Product_no.	Varchar(6)
Description	Varchar(20)

Experiment- 2

Program Name:

Application of operators, oracle functions, date conversion functions:

a. Application of operator on a column
b. Application of operator and renaming of column
c. Use of AND operator
d. Use of OR
e. Use of BETWEEN
f. Use of NOT BETWEEN
g. USE of LIKE
h. USE of LIKE with OR
i. Use of IN
j. Use of NOT IN
k. Use of Oracle functions
l. Use of Conversion functions like TO_CHAR, TO_DATE etc.

Theory Concepts:

This experiment deals with commands of SQL which are used to print data from a table with various conditions. It also deals with various in built commands like max(), min(), sqrt(), round(), trim(), etc. The program would print the current system date and time using the different commands.

Implementation:

Computation on Tables:

a)Select product_no, description, selling_price * 0.05 from Product_master;

Output:

PRODUCT_NO	DESCRIPTION	SELLING_PRICE*0.05
1	Chair	62.5
2	Table	300

b)Select product_no, description, selling_price * 0.05 new_price from Product_master;

Output:

PRODUCT_NO	DESCRIPTION	NEW_PRICE
1	Chair	62.5
2	Table	300

c)Select product_no, description, selling_price fromProduct_master where cost_price> 500 **AND**cost_price<700;

Output:

No rows selected

d)Select client_no, name, city from client_master where pincode = 201001 **OR**pincode = 201009;

Output:

CLIENT_NO	NAME	CITY
3	Akshat	Ghaziabad

e)Select product_no, description, selling_price fromProduct_master where cost_price**BETWEEN** 500 AND 700;

Output:

PRODUCT_NO	DESCRIPTION	SELL_PRICE
4	mirror	300

f)Select product_no, description, selling_price fromProduct_master where cost_price**NOTBETWEEN** 500 AND 700;

Output:

PRODUCT_NO	DESCRIPTION	SELLING_PRICE
1	Chair	1250
2	Table	6000

g)Select * from client_master where name **like** 'ja**%**';

Output:

No rows selected

h)Select * from client_master where name **like '_r%**' OR name **like '_h%';**

Output:

CLIENT_NO	NAME	CITY	PINCODE	STATE
4	Dhawal	Ghaziabad	24364	UP

i)Select from client_masterclient_no, name, city where name **IN**('Ajay', 'Vijay', 'Amit');

Output:

No rows selected

j)Select from client_masterclient_no, name, city where name **NOT IN**('Ajay', 'Vijay', 'Amit');

Output:

CLIENT_NO	NAME	CITY
3	Aksita	Ghaziabad
4	Dhawal	Ghaziabad
5	AkshanshDharampur	

k)**Oracle Functions:**

1. Select **avg(**selling_price) from product_master;

Output:

AVG(SELLING_PRICE)
5160

2. Select **avg**(selling_price) "Average" from product_master;

Output:

AVERAGE
5160

3. Select **min**(selling_price) from product_master;

Output:

MIN(SELLING_PRICE)
300

4. Select **max**(selling_price) from product_master;

Output:

MAX(SELLING_PRICE)
10250

5. Select**count**(client_no) from client_master;

COUNT(CLIENT_NO)
5

6. Select **count(***) from client_master;

Output:

COUNT(*)
5

7. Select **sum**(selling_price) from product_master;

Output:

SUM(SELLING_PRICE)
25800

8. Select **abs**(-15) from dual;

Output:

ABS(-15)
15

9.Select **power**(3,2) from dual;

Output:

POWER(3,2)
9

10.Select **round**(15.19,1) from dual;

Output:

ROUND(15.19,1)
15.2

11.Select **round(**15.19) from dual;

Output:

ROUND(15.19)
16

12.Select **round**(15.55) from dual;

Output:

ROUND(15.55)
16

13.Select **sqrt**(25) from dual;

Output:

SQRT(25)

14.Select **lower**('IVAN BAYRASS') from dual;

Output:

LOWER(IVAN BAYRASS)
ivanbayrass

15.Select **lower**('ivanbayross') from dual;

Output:

LOWER(IVAN BAYRASS)
Ivan bayrass

16.Select **initcap**('ivanbayross') from dual;
INIT(IVAN BAYRASS)
Ivan Bayrass

17.Select **substr**('SECURE',3,4) from dual;

Output:

SUBSTR('SECURE',3,4)
CURE

18.Select **substr(**'SECURE',3,3) from dual;

Output:

SUBSTR('SECURE',3,3)
CUR

19.Select **length**('ELEPHANT') from dual;

Output:

LENGTH('ELEPHANT')
8

20.Select **ltrim**('NISHA','NI') from dual;

Output:

LTRIM('NISHA','NI')
SHA

21.Select **rtrim**('SUNILA','LA') from dual;

Output:

RTRIM('SUNILA','LA')
SUNI

22.Select **lpad**('Page 1',10,'*') from dual;

Output:

LPAD('Page 1',10,'*')
****PAGE1

23.select **rpad**('Anita',10,'*') from dual;

Output:

RPAD('Anita',10,'*')
ANITA*****

l)**CONVERSION FUNCTIONS:**

1.Select **to_date**('30-Sep-1966') from dual;

Output:

TO_DATE('30-SEP-1966)
30-Sep-66

2.Select**add**_months(sysdate,4) from dual;

Output:

ADD_MONTHS(SYSDATE,4)
13-Jun-13

3.Select **last_day**(sysdate) from dual;

Output:

LAST_DAY(SYSDATE)
28-Feb-13

4.Select **months_between**('15-Feb-10','15-Jun-10') from dual;

Output:

MONTHS_BETWEEN('15-Feb-10','15-Jun-10')
-4

5.Select **next_day**('15-Feb-10', 'Wednesday') from dual;

Output:

NEXT_DAY('15-Feb-10', 'Wednesday')
17-Feb-10

Experiment- 3

ProgramName : Execute the following queries:

a. The NOT NULL
b. The UNIQUE Constraint

c. The PRIMARY KEY Constraint
d. The FOREIGN KEY Constraint
e. The CHECK Constraint
f. Defining Integrity constraints in ALTER table command

Theory Concept:

This program intends to explore various constraints enforced on the database like NOT NULL, UNIQUE constraint etc. Primary kay is an attribute of table which is used to identify each row of the table uniquely. Foreign key is used to reference other tables.

Implementation:

a.The NOT NULL Constraint

Q-1) Create table Employee with attributes eno, name, salary and the constraints that eno and name cannot be NULL

Ans: create table employee(eno number NOT NULL, name varchar(20) NOT NULL,salary number);

Output:

Table created

Q-2) Insert data into the table Employee.Then Test the NOT NULL constraint with appropriate data.

Ans:insert into employee (eno,name,salary) values(2,'Amit',20000);

Output:1 row inserted

insert into employee (eno,name,salary) values(2,'Sumit',null);

Output:1 row inserted

insert into employee (eno,name,salary) values(null,'Amit',20000);

Output: ***** This query will give an error for violation of NOT NULL constraint***

Q-3) Select all records from Employee where salary is NULL

Ans: select * from employee where salary is null;

Output:

eno	name
2	Sumit

1 row selected

b.The UNIQUE Constraint

Q-4) Create table Department with attributes dno, dname and no_of_employees and the constraint that dno and dname must be unique

Ans:create table department (dno number UNIQUE,dnamevarchar(20) UNIQUE, No_of_employees number);

OR

create table department (dnonumber,dnamevarchar(20) UNIQUE, No_of_employees number) UNIQUE(dno);

Output: Table created

Q-5) Insert records into the above table. Test the UNIQUE constraint with appropriate data

Ans: insert into department (dno,dname,no_of_employees) values(1,'EC',20);

Output: 1 row inserted

insert into department (dno,name,no_of_employees) values(1,'EC',20)

Output: ** ***This query will give an error for violation of UNIQUE constraint***

c.<u>The PRIMARY KEY Constraint</u>

Q6) Create table Project with attributes pno, pname, location and pno as the primary key

Ans: create table Project (pnonumber,pnamevarchar(20),location varchar(20),PRIMARY KEY(pno));

OR

create table Project (pno number PRIMARY KEY ,pnamevarchar(20),location varchar(20));

Output: Table created

Q7) Insert into project appropriate records

Ans: insert into project (pno,pname,location) values (1,'Web Designing', 'Lab);

Output:1 row inserted

insert into project (pno,pname,location) values (1,'Web Designing', 'Lab);

Output: ** ***This query will give an error for violation of UNIQUE constraint***

d.<u>The FOREIGN KEY Constraint</u>

Q8) Create table works_on with attributes eno, pno and hours where eno is foreign key referencing eno from Employee table and pno is the foreign key referencing pno from Project Table. Let {eno,pno} be the primary key for table Works_on

Ans: create table works_on(eno number REFERENCES Employee, pno number REFERENCES Project, Hours number ,PRIMARY KEY(eno,pno));

Output: Table created

Q9) Insert appropriate records into the Works_on Table

Ans: Insert into Works_onvalues(1,2,30);

delete from employee where eno = 1;

Output: ** Error **ORA-02292: integrity constraint (SCOTT.SYS_C003014) violated - child record found**

Q-10) Select appropriate records from the table Works_on by referencing the table Employee and Project

Ans:select employee.eno,employee.name,project.pno,project.pname,works_on.hours from works_on,employee,project
where (works_on.eno = employee.eno and works_on.pno = project.pno);

Output:

ENO	NAME	PNO	PNAME	HOURS
2	Amit	1	asd	8
2	Amit	2	sjxks	9
1	SAmit	1	asd	23
1	SAmit	2	sjxks	30

4 rows selected

e.The CHECK Constraint

Q-11) Create table person with attributes ssno, name city such that all ssno have 'C' as the first character, all names are entered in uppercase and city is either Delhi or Mumbai or Bangalore

Ans: Create table person(ssnovarchar(3) CHECK(ssno LIKE 'C%'),name varchar(20) CHECK (name = upper(name)), city varchar(20) CHECK (city IN('Delhi','Mumbai','Bangalore')));

insert into person values('C12','MAMTA','Mumbai');

Output:1 row created

insert into person values('C12','mamta','Mumbai');

Output:ERROR at line 1:
ORA-02290: check constraint (SCOTT.SYS_C003017) violated

f)Defining Integrity Constraints in the Alter Table Command

Q-12) Alter table client_master to make client_no the primary key

Ans: alter table client_master ADD Primary key(client_no);

Output:

Table Altered

Q-13) Alter table client_master to drop the primary key

Ans: alter table client_master drop primary key;

Output:

Table Alterted

Experiment- 4

Program Name:

Execute queries related to Group By and Having Clause on tablesSALES_ORDER.

Theory Concept:

The program aims to familiarize the user with grouping of data based on conditions to ensure better usability of data.

Implementation:

GROUP BY

Q1) Create table sales_order with attributes product_no and Qty. Insert records into the table and find the total qty ordered for each product_no.

Ans: Create table sales_order (product_novarchar(10), Qty numbe(4));

Output: Table created.

insert into sales_order values(&product_no, &qty);

select* from sales_order;

Output:

```
PRODUCT_NO       QTY
---------- ----------
p1                12
p2               112
p1                 9
p2                23
p3                23
p3                23

6 rows selected.
```

selectproduct_no, sum(qty) from sales_order group by product_no;

Output:

```
PRODUCT_NO   SUM(QTY)
---------- ----------
p1                21
p2               135
p3                46
3 rows selected.
```

HAVING clause

Q2) Find the total Qty for product_no 'p1' and 'p2' from the table sales_order

Ans:select product_no, sum(qty) from sales_order group by product_no havingproduct_no = 'p1' OR product_no = 'p2';

Output:

```
PRODUCT_NO   SUM(QTY)
---------- ----------
p1                21
p3                46
```

2 rows selected

Experiment- 5

Program Name:

Execute Nested Queries on tables CLIENT_MASTER, PRODUCT_MASTER, SALESMAN_MASTER, SALES_ORDER, SALES_ORDER_DETAILS

Theory Concept:

The program intends to familiarize nested queries so as to retrieve data from a record by using filtered data from another record.

Implementation:

Q1) Retrieve the order numbers, client names and their order dates from client_master and sales_order tables.

Ans: Select order_no, order_date, name from sales_order, client_master where client_master.client_no = sales_order.client_no order by order_date;

OUTPUT :

Order_no	order_date	name
1	1999/12/05	akshansh
2	1999/12/12	david

Q2) Retrieve the product numbers, description and total quantity ordered for each product

Ans:Selectsales_order_details.product_no, description, sum(qty_ordered) from sales_order_details, product_master where product_master.product_no = sales_order_details.product_no group by sales_order_details.product_no, description;

OUTPUT :

product_no	description	sums(qty_ordered)
1	chair	2
2	pen	5

Q3) Retrieve the names of employees and names of their respective managers from the employee table.

Ans: Select employee.name, employee.name from employee where employee.manager_no = employee.employee_no;

OUTPUT :

Name	Name
akshansh	David
Akshat	David

UNION , INTERSECT and MINUS CLAUSE

Q1) Retrieve the names of all clients and salesmen in the city of Mumbai from the tablesclient_master and salesman_master.

Ans: Selectsalesman_no from salesman_master where city = ‘Mumbai’
UNION
Select client_no from client_master where city = ‘Mumbai’;

OUTPUT :

Name

Akshansh
Akshat
David

Q2) Retrieve the salesman name in Mumbai whose efforts have resulted into atleast one sales transaction

Ans: Selectsalesman_no, name from salesman_master where city = 'Mumbai'
INTERSECT
Select salesman_master.salesman_no, name from salesman_master, sales_order where salesman_master.salesman_no = sales_order.salesman_no;

OUTPUT :

Saleman_noName
---
1 akshansh
2 David

Q3) Retrieve all the product numbers of non-moving items from the product_master table

Ans:

OUTPUT :

product_no

3
4

VIEWS

Q1) Create a view on salesman_master table for the sales department

Ans: Create view vw_sales as select * from salesman_master;

OUTPUT:

View created

Q2) Create a view on client_master table

Ans: Create view vw_client as select name, address1, address2 , city,pincode , state, bal_due from client_master;

OUTPUT:

View created

Q3) Perform insert, modify and delete operations on the view created in Q2

Ans:

a) Insert into vw_client values('C001', 'Robert', 'AAAAAA', 'BBB', 'Delhi', 2000000, 'MMM');

OUTPUT:

1rows created

b)Updatevw_client set bal_due = 10000 where client_no = 'C001';

OUTPUT:

1 row updated

c)Delete from vw_client where client_no = ‘C001’;

OUTPUT:

1 row deleted

Experiment-6

ProgramName:Execute queries related to Exists, Not Exists, Union, Intersection, Difference, Join on tables CLIENT_MASTER, PRODUCT_MASTER, SALESMAN_MASTER, SALES_ORDER, SALES_ORDER_DETAILS

Theory Concept:

The program retrieves data from records by defining relation between two tables so as to retrieve filtered records.

Implementation:

Correlated queries with EXISTS/NOT EXISTS clause

1)Select all products and order_no where order_status is 'in Process'
Ans: Select order_no.,product_no. from sales_order_details where exists(select * from sales_order , order_no = sales_order_details,order_no and order_status='in process');

Output:

Order_no	Product_no
0003	3

2)Selectorder_no and order_date for all orders which include product_no 'P001' and quantity_ordered>10
Ans: Select order_no,order_data from sales_order where exists(select * from sales_order_details where sales_order_details,order_no = sales_order.Order_no and product-no='p001' and quantity-ordered>10;

Output :

Order_no	Product_no
0002	05/feb/13

3)Find all order_no for salesman rashmi.
Ans: Select order_no from sales_order where exists(select * from salesman_master where salesman_master.saleman-no= sales_order-salesman_no and name='rashmi');

Output :

Order_no
0003

4)Select all clients who have not placed any orders.
Ans: Select * from client_master where not exists(select * from sales_order.client_no=client_master.client_no);

Output :

Client_no	Name	City	Pincode	State
6	David	Hamirpur	35498	H.P.
7	Dorothy	Noida	32547	U.P.

5)Select all orders with order_date for 'acrylic colors'
Ans: Select order_no,order_date from sales_order where exists(select * from sales_order_details.oder_no=sales_order.order_no AND exists(select * from product 1 where sales_order_details.product_no=product_no AND description='acrylic colors');

Output :

Order_no	Order_date
0001	23/jan/13

Union,Intersect and minus clause:

1)List all the clients and salesman and their names
Ans: Select client_no, name from client_master UNION select salesman_no,name from salesman_master;

Output :

Client_no	Name
3	Akshat
4	Dhawal

2)List all the clients and their names who are also salesman.
Ans: Select name from client_master INTERSECT,select name from salesman_master;

Output :

No rows selected

3)List all the clients who are not salesman.
Ans: Select name from client_master MINUS select name from salesman_master;

Output :

Name
Akshat
Dhawal
Akshansh
David
Dorothy

4)List all the clients who have placed orders
Ans: Select client_no from client_masterINTERSECT select client_no from sales_order;

Output :

Client_no
6
7

5)List all the clients who have not placed any order.

Ans: Select client_no from client_masterMINUS select client_no from sales_order;

Output :

Client_no
3
4
5

6)List all the clients in UP who have placed orders

Ans: Select client_no from client_master where state='UP' INTERSECT select client_no from sales_order;

Output :

Client_no
3
4
5

7)Find all the clients and their names from city Ghaziabad who have delivery date of their orders as today.

Ans: Select client_no from client_master where city='Ghaziazbad' INTERSECT select client_no from sales_order where delivery_date='09-MAR-13'

Output :

Client_no
5

Queries on Joins

1)List the product_no and description of products sold.

Ans: Select product_no, description from (product1 natural join sales_order_details)

Output :

Product_no	Description
1	Chair
1	Chair
2	Table
3	Sofa

2) Find the products which have been sold to 'akshansh'

Ans: Select product_no, description from (product1 natural join sales_order details natural join sales_order natural join client_master) where name='akshansh';

Output :

Product_no	Description
3	Sofa

3)Find the products and their quantities that will have to be delivered in the current month.

Ans: Select sales_order_detailsproduct_no, product1 ,description, sum(sales_order_details, quantity_ordered) from sales_order_details, sales_order, product1 where product1, product_no=sales_order_details, product_no and sales_order, order_no=sales_order_details, order_no and to_char (delivery_date,'mon-yy') = to_char(sysdate,'mon-yy')group by sales_order_details, product_no, product1, description ;

Output :

no rows selected

4)Find the names of client who have purchased 'chair'

Ans: Select name from(client_master natural join sales_order natural join sales_order_details natural join product1) where description= 'chair';

Output :

Name
Akshat
Akshansh

5)List the orders for less than 5 units of sale of 'chair'

Ans: Select product_no, order_no from (sales_order_details natural join product1) where (description='chair' and qty_ordered<5);

Output :

Product_no	Order_no
1	0001
1	0001

6)Find the products and their quantities placed by 'akshansh' or 'Akshat'.

Ans: Select product_no, description, qty_ordered from (product1 natural join sales_order_details natural join sales_order_natural join client_master) where (name='akshansh' or name='Akshat');

Output :

Product_no	Description	Qty_ordered
1	Chair	4
1	Chair	3
2	Sofa	2

7)Find the products and their quantities for the orders placed by the client_no '3' and '5'

Ans: Select product_no, description, qty_orderedfrom(product1 natural join sales_order_details natural join sales_order natural join client_master) where (client_no=3 OR client_no=5);

Output :

PRODUCT_NO	DESCRIPTION	QTY_ORDERED
1	Chair	4

1	Chair	3
3	Sofa	2

Experiment- 7

Program Name: Procedures, Functions & Packages:

a. Write a simple procedure to display a message "Good Day to You"
b. Code a function to return the Square of a given number.
c. Create a package to include the following:

A named procedure to list the Product_no of products with Quantity_on_hand as 5 in PRODUCT_MASTER table.

A function which returns the max maximum Quantity_on_hand for a given product.

Theory Concept:

The program would print the message using a procedure in Oracle.

Implementation:

Ans(a):

```
SQL> create or replace procedure goodday is
  2    begin
  3      dbms_output.put_line ('Good Day to You');
  4    end;
  5  /

Procedure created.

SQL> execute goodday;
Good Day to You

PL/SQL procedure successfully completed.

SQL> drop procedure goodday;

Procedure dropped.

SQL>

End listing
```

Ans(b):

```
CREATE OR REPLACE FUNCTION square(original NUMBER)
RETURN NUMBER
AS
  original_squared NUMBER;
BEGIN
  original_squared := original * original;
  RETURN original_squared;
END;
```

Ans(c):

```
Create procedure Product_PNO
(in  Quantity_on_had integer
Out  P_Product_No integer)
Begin
Select product_NO into P_product_No from Product_Master
Where Product_Master.
Quantity_on_hand=Product_PNO.Quantity_On_hand.
```

Deaclare P_product_No integer;
Call product_PNo('5'P_Product_No)

Second part

C2: Create function max _Quanity(Product_No integer)
Return table(Quantity_on_Hand integer)
Return table(
Select max(Quantity_on_hand) from Product_Master wher
Product_master.Product_no=max.quantity.product_no

Experiment- 9

Program Name: Write a CURSOR to display list of clients in the CLIENT_MASTER table.

Theory Concept: The following example would illustrate the concept of CURSORS. We will be using the CLIENT_MASTER table and display records.

Implementation:

```
DECLARE
  CURSOR client_cur is
    SELECT id, name, address
    FROM client_master;
  client_rec client_cur%rowtype;
BEGIN
  OPEN client_cur;
  LOOP
    FETCH client_cur into client_rec;
    EXIT WHEN client_cur%notfound;
    DBMS_OUTPUT.put_line(client_rec.id || ' ' || client_rec.name);
  END LOOP;
END;
/
```

Output: When the above code is executed at SQL prompt, it produces the following result:

```
1 Ramesh
2 Khilan
3 kaushik
4 Chaitali
5 Hardik
6 Komal

PL/SQL procedure successfully completed.
```

Experiment- 10

Program Name: Installing Oracle Database

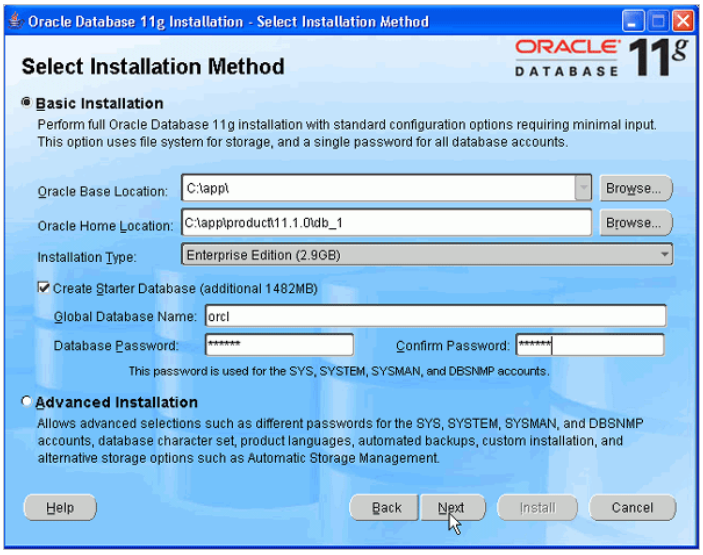
Theory Concept: To install the Oracle software, you must use the Oracle Universal installer.

Implementation:

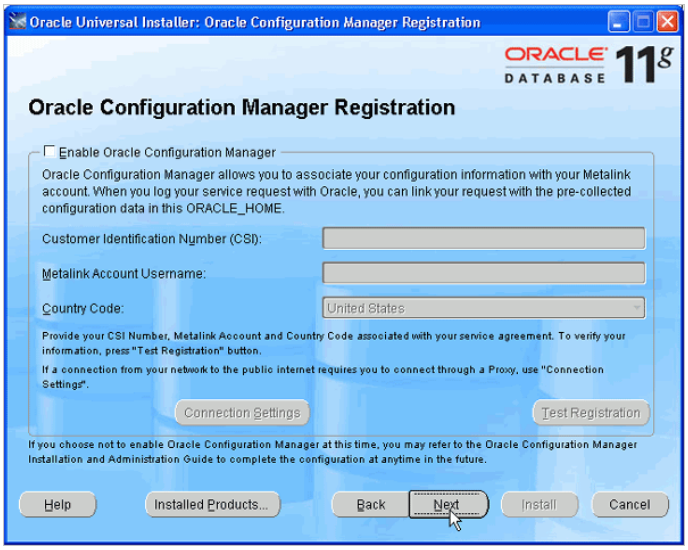
1. For this installation, you need either the DVDs or a downloaded version of the DVDs. In this tutorial, you install from the downloaded version. From the directory where the DVD files were unzipped, open Windows Explorer and double-click on **setup.exe** from the \db\Disk1 directory.

2. The product you want to install is **Oracle Database 11g**. Make sure the product is selected and click **Next**.

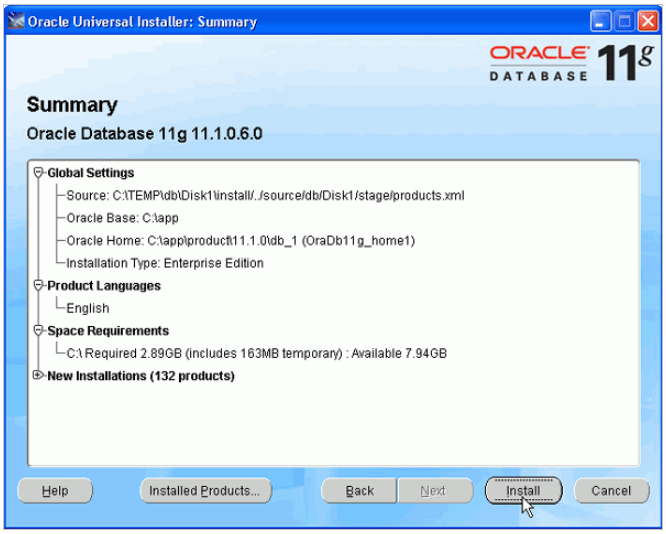
3. You will perform a basic installation with a starter database. Enter **orcl** for the Global Database Name and **oracle**for Database Password and Confirm Password. Then, click **Next**.



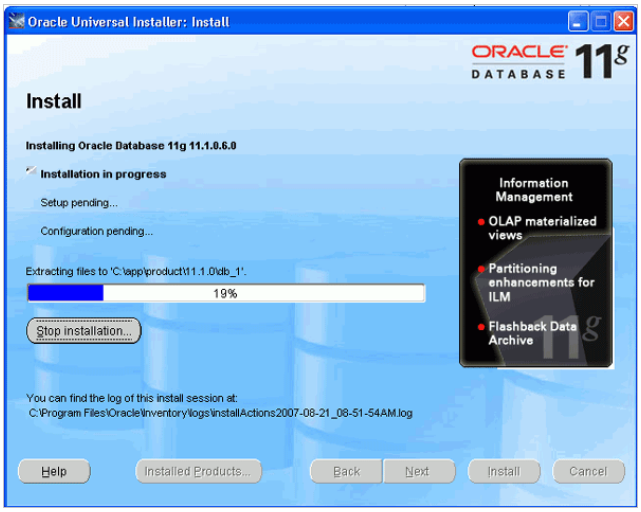
4. Oracle Configuration Manager allows you to associate your configuration information with your Metalink account. You can choose to enable it on this window. Then, click **Next.**



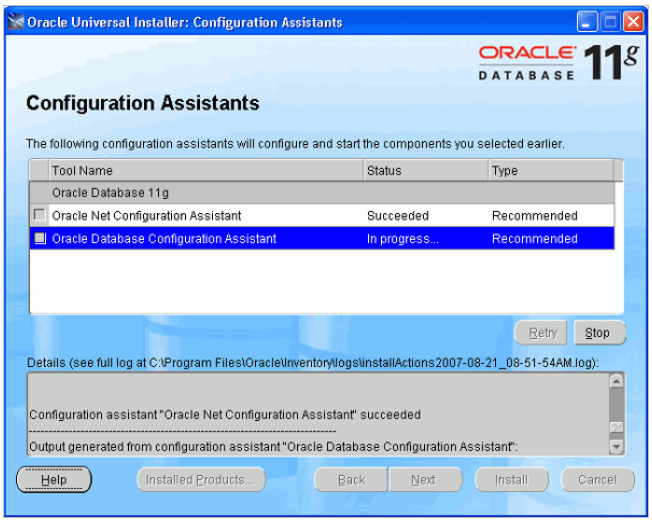
5. Review the Summary window to verify what is to be installed. Then, click **Install**.



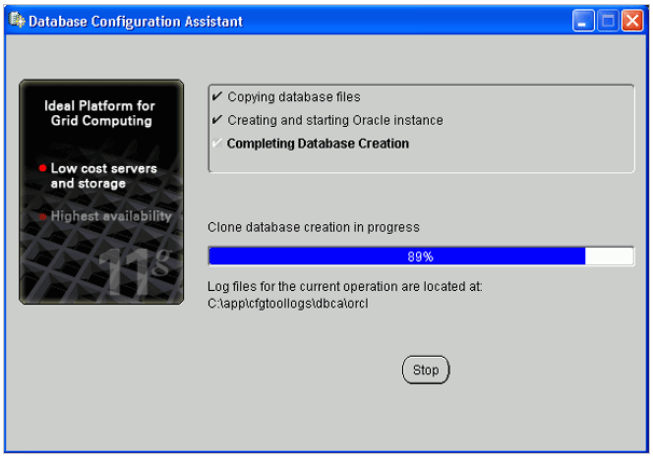
6. The progress window appears.



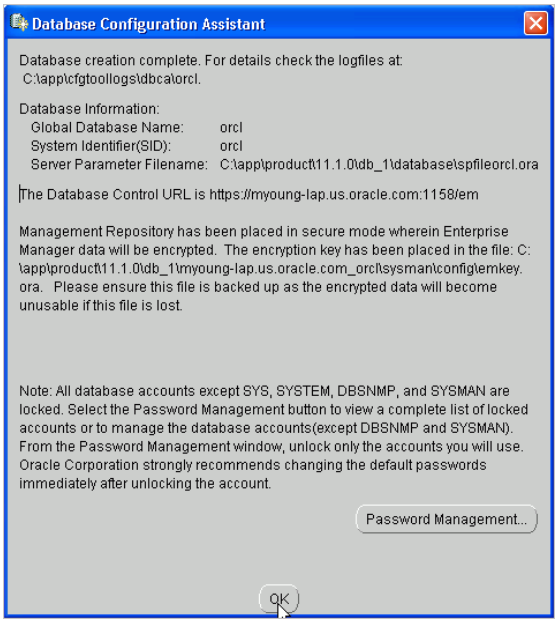
7. The Configuration Assistants window appears.



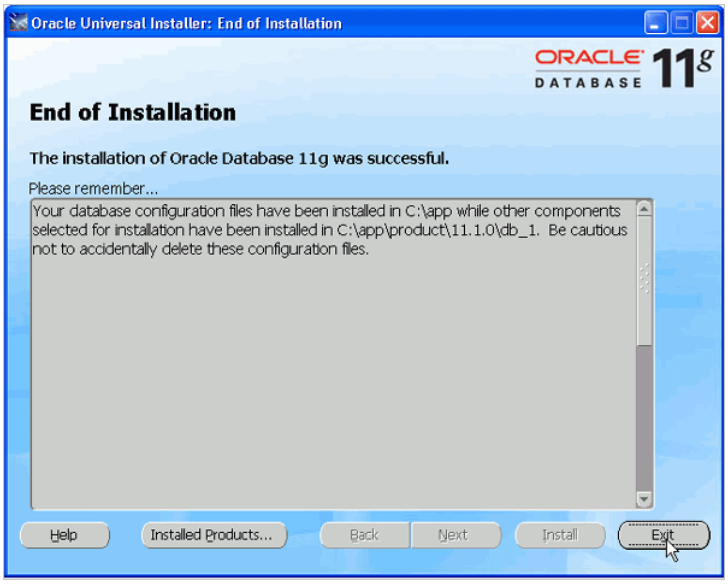
8. Your database is now being created.



9. When the database has been created, you can unlock the users you want to use. Click **OK**.



10. Click **Exit**. Click **Yes** to confirm exit.



Experiment- 11

Program Name: Creating Entity-Relationship Diagram using case tools.

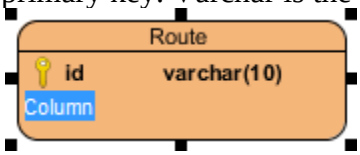
Theory Concept: Entity relationship diagram (ERD) is a kind of diagram for presenting visually the structure of relational database. In this experiment we will make use of ERD to model the database structure of a simple bus route management system.

Implementation:

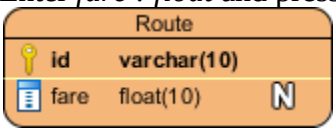
1. Start Visual Paradigm. Select a new workspace folder for this tutorial.
2. Select **Project > New** from the toolbar to create a project. Name the project as *Bus Route Management* and confirm.
3. To create an ERD, select **Diagram > New** from the toolbar. In the **New Diagram** window, select **Entity Relationship Diagram** and click**Next**. Enter *Bus Route Management* as diagram name and click **OK**.
4. Let's start by creating the first entity *Route*. Select **Entity** in diagram toolbar and click on the diagram to create an entity. Name the entity*Route* and press **Enter** to confirm.

5. Create columns in *Route*. Let's start with a primary key. Right click on entity *Route* and select **New Column** from popup menu.

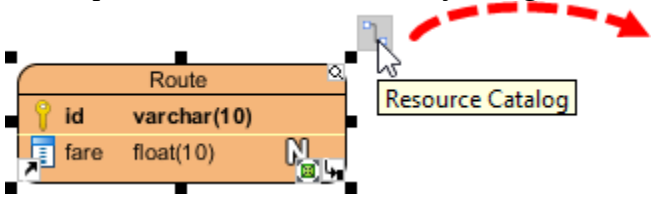
6. Enter *+id : varchar(10)* and press **Enter**. Note that the + sign means that the column is a primary key. Varchar is the column type and 10 is the length.



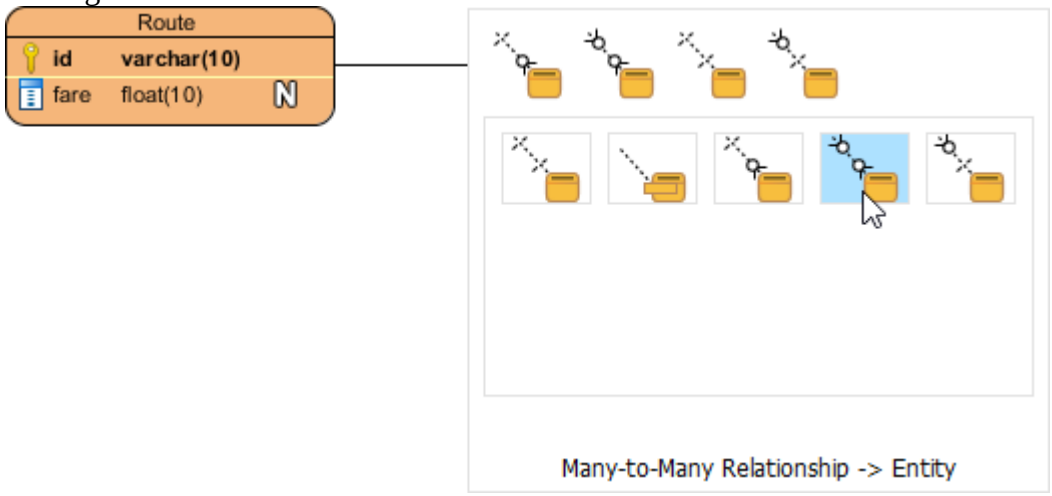
7. Enter *fare : float* and press **Enter**, then **Esc** to create another column.



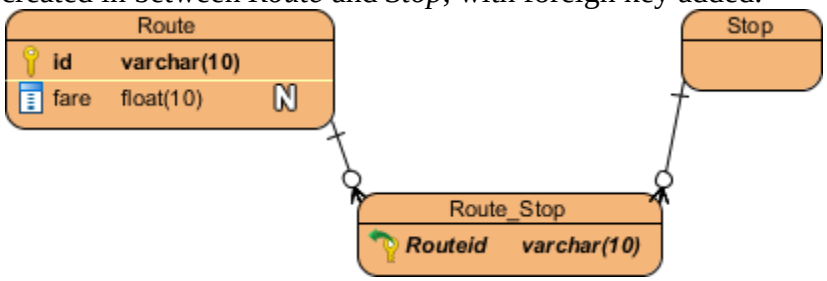
8. Create entity *Stop*. A bus route has many bus stops, while a stop can be shared by many routes. Therefore, there is a many-to-many relationship between *Route* and *Stop*. Place the mouse pointer over the *Route* entity. Drag out the **Resource Catalog** icon at top right.



9. Release the mouse button and select **Many-to-Many Relationship -> Entity** from Resource Catalog.



Name the new entity *Stop*, You can see that a linked entity *Route_Stop* is automatically created in between *Route* and *Stop*, with foreign key added.

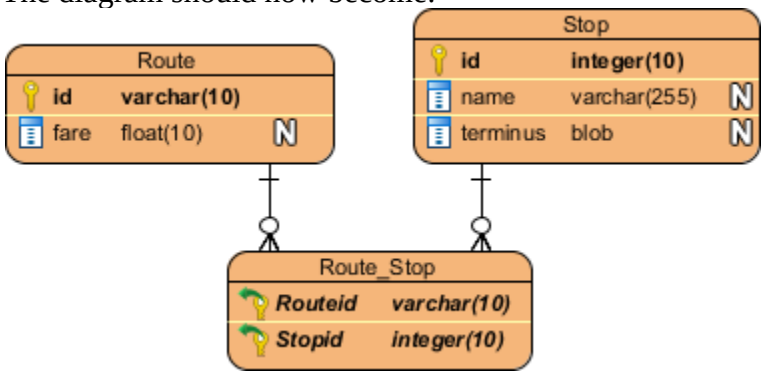


10. Create the following columns in *Stop*:

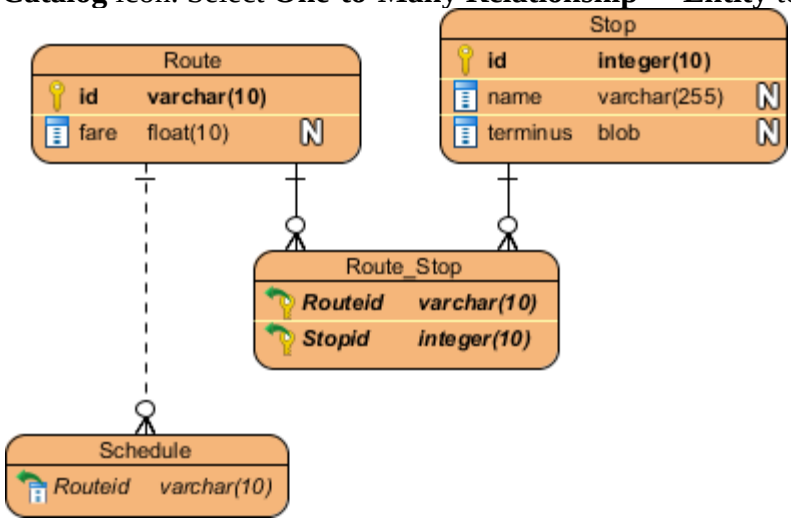
Key	Name	Type
PK	id	int(10)
	name	varchar(255)
	terminus	blob

11.

The diagram should now become:



12. A route has multiple bus schedules. Create an entity *Schedule* from *Route* with a one-to-many relationship. Move the mouse pointer to*Route*. Press and drag out the **Resource Catalog** icon. Select **One-to-Many Relationship -> Entity** to create entity *Schedule*.



13. Create the following columns in *Schedule*:

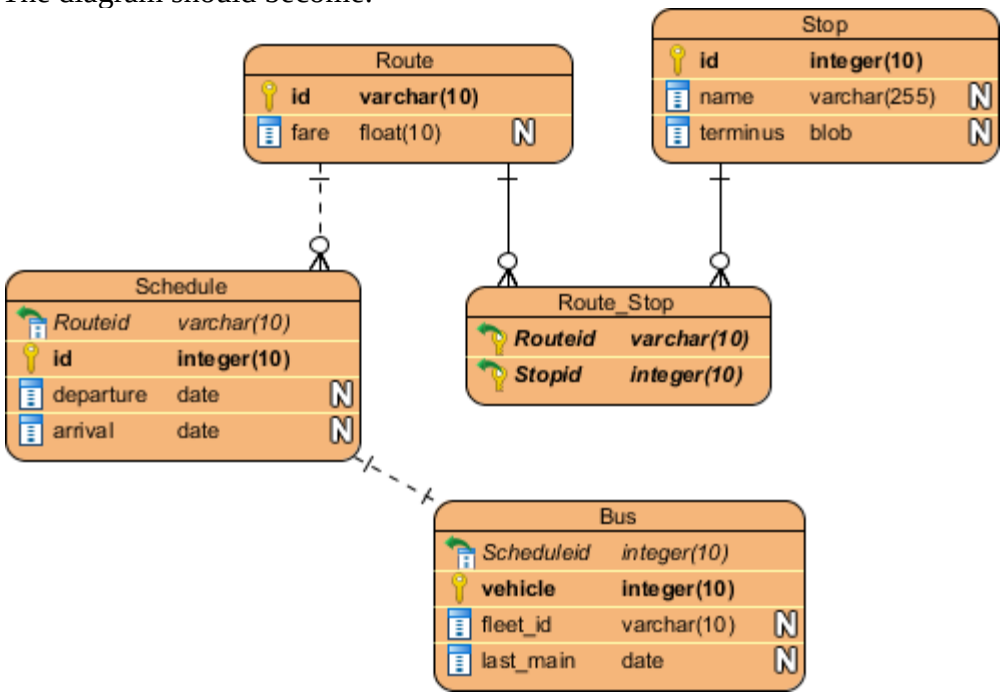
Key	**Name**	**Type**
PK	id	int(10)
	departure	date
	arrive	date

14. A schedule is handled by a bus. Create an entity Bus from *Schedule*, with an one-to-one relationship. Create the following columns in*Bus*:

Key	**Name**	**Type**
PK	vehicle_id	int(10)

	fleet_id	varchar(10)
	last_main	date

15. The diagram should become:

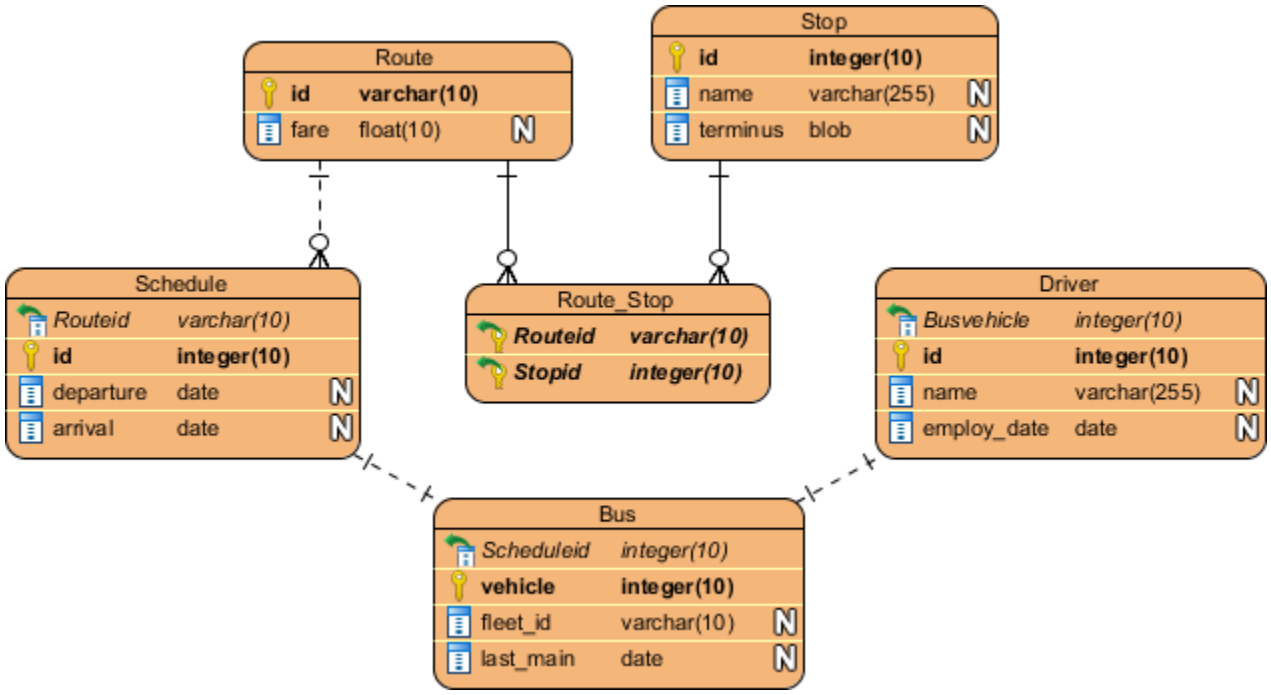


16. A bus is driven by a bus driver. Create entity *Driver* from *Bus* with a one-to-one relationship.

    Add the following columns to *Driver*:

Key	**Name**	**Type**
PK	id	int(10)
	name	varchar(255)
	employ_date	date

17. This is the final ERD.



Experiment- 12

Program Name:

PL/SQL programming

a. Write a PL/SQL block code to print the squares of numbers upto 99.
b. Write a PL/SQL block code to insert data into table CUSTOMER

Theory Concept:

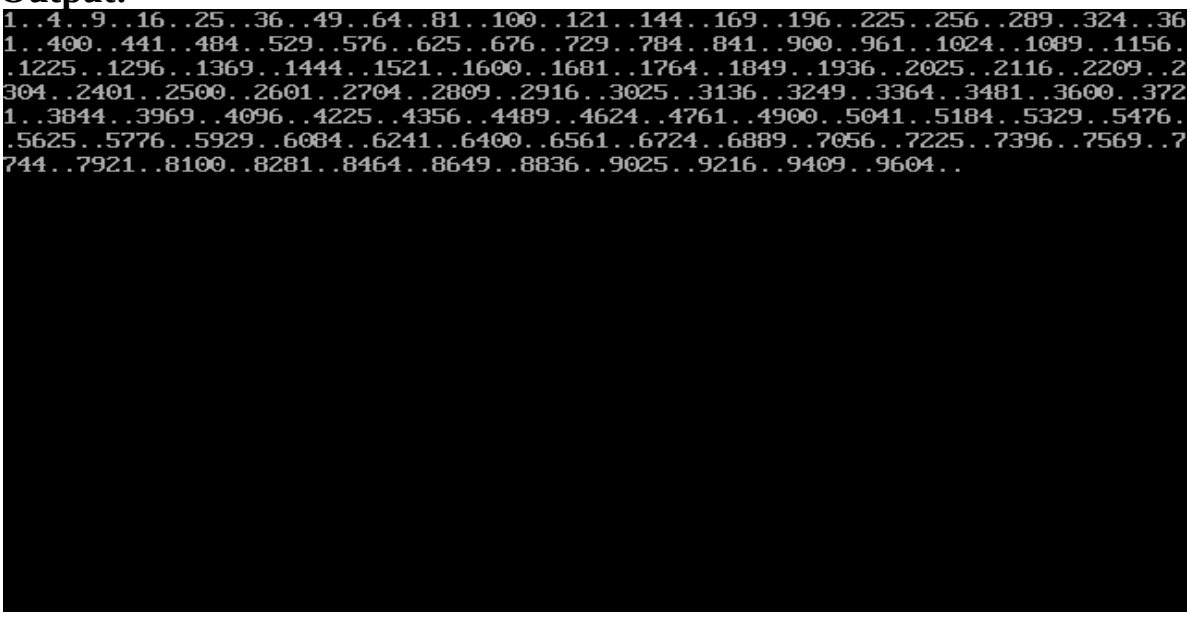
The program would print the squares of numbers upto 99 using for loop and data into table CUSTOMER in pl/sql.

Implementation:

Ans (a):

```
setserveroutput on
DECLARE

BEGIN

    for x in 1..99

    loop

        dbms_output.put_line(x * x);
    end loop;
end;
```

Output:



Implementation:

Ans:(b)

desc customer;

Name	Null?	Type
C_IDVARCHAR2(2)		
CNAME		VARCHAR2(10)
SALARY		NUMBER

select * from customer;

Output:

C_	CNAME	SALARY
c1jkjkjkj		7888