DATABASE LABORATORY

Subject Code: 2TMCAL1 I. A. Marks: 50
Hours/Week: 02 Exam Marks: 50
Total Hours: 24 Exam Hours: 03

Practical – 2 :

Consider the following relations for an order processing database application in a company.

CUSTOMER (Cust#: Number, CustomerName: String, City: String)
CORDER (Order#: int, OrderDate: date, Cust#: int, OrderAmount:Number)
ORDER_ITEM (Order#: int, Item#: Int, Qty: int)
ITEM (Item#: int, ItemName string,Unitprice: int)
SHIPMENT (Order# Number, Warehouse#: Number, ShipDate: date)
WAREHOUSE (WareHouse#: Number, WareHouseName: string, City: String)

Create the above tables by properly specifying the primary keys and the foreign keys.

Enter at least five tuples for each relation.

Following are the Queries :

Query-1 : List out the details of orders, i.e. orderno, warehouse name, shipdate for a particular city :
Query-2 : List out customer name, number of orders they made, their total ordered amount and their average ordered amount for all the customers:
Query-3 : Find out the warehouse and number of orders they obtained from different customers.
Query-4 : Find out the warehouse, which has got number of orders more than 2.
Query-5 : Find out the customer who has ordered the number of items more than 3.

Create the tables with the following source codes :

Table : Customer

```
create table CUSTOMER (
customer_no        varchar2(4) primary key,
customer_name      varchar2(20),
customer_city      varchar2(20)
);
```

Table : Item

```
create table ITEM (
item_no            varchar2(6) primary key,
item_name          varchar2(20),
unit_price         number(7,2)
);
```

Table : Warehouse

```
create table WAREHOUSE (
warehouse_no       varchar2(6) primary key,
warehouse_name     varchar2(25),
warehouse_city     varchar2(20)
);
```

Table : Corder

```
create table CORDER (
order_no           number(4) primary key,
order_date         date,
customer_no        references CUSTOMER(customer_no),
order_amount       number(8,2)
);
```

Table : Order_item

```
create table ORDER_ITEM (
order_no           references CORDER(order_no),
item_no            references ITEM(item_no),
quantity           number(4)
);
```

Table : Shipment

```
create table SHIPMENT (
order_no        references CORDER(order_no),
warehouse_no    references WAREHOUSE(warehouse_no),
ship_date       date
);
```

Insert the records in to the tables with the following codes :

To the table Customer :

```
insert into customer values ('CU01','Amar','Tumkur');
insert into customer values ('CU02','Bharath','Bangalore');
insert into customer values ('CU03','Jeevan','Kolar');
insert into customer values ('CU04','Kiran','Bangalore');
insert into customer values ('CU05','Varun','Tumkur');
insert into customer values ('CU06','Vijay','Bangalore');
```

To the table Item :

```
insert into item values ('ITM001','Soap',15);
insert into item values ('ITM002','WashingPowder',35)
insert into item values ('ITM003','Sugar',45)
insert into item values ('ITM004','ToothPaste',30)
insert into item values ('ITM005','ToothBrush',20)
```

To the table Warehouse :

```
insert into warehouse values ('WNO1','SSS Enterprises','Bangalore');
insert into warehouse values ('WNO2','SIT Enterprises','Tumkur');
insert into warehouse values ('WNO3','TTK Enterprises','Salem');
insert into warehouse values ('WNO4','VGP Enterprises','Chennai');
insert into warehouse values ('WNO5','KGF Enterprises','Kolar');
```

To the table Corder :

```
insert into corder values (1,'12-jan-2010','CU01',5000);
insert into corder values (2,'15-dec-2009','CU02',7500);
insert into corder values (3,'1-jan-2010','CU03',6500);
insert into corder values (4,'15-feb-2010','CU04',4500);
insert into corder values (5,'11-feb-2010','CU05',3500);
insert into corder values (6,'21-feb-2010','CU01',4500);
insert into corder values (7,'22-Jan-2010','CU02',2500);
insert into corder values (8,'25-Jan-2010','CU01',1500);
insert into corder values (9,'12-Jan-2010','CU02',500);
insert into corder values (10,'28-Jan-2010','CU01',2500);
```

To the table Order_item :

```
insert into order_item values (1,'ITM001',50);
insert into order_item values (2,'ITM002',75);
insert into order_item values (3,'ITM003',25);
insert into order_item values (4,'ITM004',85);
insert into order_item values (5,'ITM005',35);
insert into order_item values (6,'ITM001',55);
insert into order_item values (7,'ITM002',50);
insert into order_item values (8,'ITM001',15);
insert into order_item values (9,'ITM003',45);
insert into order_item values (10,'ITM004',25);
```

To the table Shipment :

```
insert into shipment values (1,'WNO1','15-Jan-2010');
insert into shipment values (2,'WNO2','20-Dec-2009');
insert into shipment values (3,'WNO3','7-Jan-2010');
insert into shipment values (4,'WNO4','20-feb-2010');
insert into shipment values (5,'WNO5','15-feb-2010');
insert into shipment values (6,'WNO1','25-feb-2010');
insert into shipment values (7,'WNO1','25-jan-2010');
insert into shipment values (8,'WNO2','1-feb-2010');
insert into shipment values (9,'WNO3','16-Jan-2010');
insert into shipment values (10,'WNO1','1-Jan-2010');
```

Execute the queries in the following steps :

Query-1 : List out the details of orders, i.e. orderno, warehouse name, shipdate for a particular city :

Step-1 : Try to select details of orders from shipment and warehouse tables by using join concept :

```
SQL> select order_no, warehouse_name,ship_date,warehouse_city
     from shipment, warehouse
     where shipment.warehouse_no = warehouse.warehouse_no
```

Step-2 : Then, Try to select the same for a particular city :

```
SQL> select order_no, warehouse_name,ship_date,warehouse_city
     from shipment, warehouse
     where shipment.warehouse_no = warehouse.warehouse_no
     and warehouse_city='Bangalore'
```

By using Sub Query concept :

```
SQL >  select * from shipment
        where warehouse_no in ( select warehouse_no
                                from warehouse
                                where warehouse_city='Bangalore')
```

Query - 2 : List out customer name, number of orders they made, their total ordered amount and their average ordered amount for all the customers:

Step – 1 : First try to select customer name, their number of orders, their total amount, their average amount for all the customers by using aggregate functions and group by clause

```
SQL > select customer_name, count(customer_name) as NoofOrders,
        sum(order_amount) as Total_amount, Avg(order_amount) as Average_amount
        from customer, corder
        where customer.customer_no = corder.customer_no
        group by customer_name
```

Query - 3 : Find out the warehouse and number of orders they obtained from different customers.

Step – 1 : First try to select warehouse name, their number of orders by using GROUP BY clause

```
SQL > select warehouse_name, count(warehouse_name) as NoofOrders
        from shipment, warehouse
        where shipment.warehouse_no = warehouse.warehouse_no
        group by warehouse_name
```

Query - 4 : Find out the warehouse, which has got number of orders more than 2.

Step – 1 : First try to select warehouse name, their number of orders by using GROUP BY clause

```
SQL > select warehouse_name, count(warehouse_name) as NoofOrders
        from shipment, warehouse
        where shipment.warehouse_no = warehouse.warehouse_no
        group by warehouse_name
```

Step – 2 : Then try to select warehouse name, their number of orders more than 3 by using additional HAVING clause

```
SQL > select warehouse_name, count(warehouse_name) as NoofOrders
        from shipment, warehouse
        where shipment.warehouse_no = warehouse.warehouse_no
        group by warehouse_name
        having count(warehouse_name) > 2
```

Query - 5 : Find out the customer who has ordered the number of items more than 3.

Step-1 : First Try to display customer name and their number of ordereing the items by using GROUP BY clause.

```
SQL> select customer_name, count(customer_name)
     from corder, customer
     where corder.customer_no = customer.customer_no
     group by customer_name
```

Step-2 : Then Try to display customer name whose number of ordered items are more than 3 by using additional clause HAVING clause.

```
SQL> select customer_name, count(customer_name)
     from corder, customer
     where corder.customer_no = customer.customer_no
     group by customer_name
     having count(customer_name) > 3
```