

KIET GROUP OF INSTITUTIONS

DEPARTMENT OF COMPUTER APPLICATIONS



DATABASE MANAGEMENT SYSTEM (DBMS)

(KCA-204) SESSION (2023-25) 2nd SEM

NAME : UTKARSH KUMAR SINGH

CLASS ROLL : 50

UNIVERSITY ROLL : 2300290140198

SEM : II

SEC: C

LAB ASSIGNMENT-1

1)

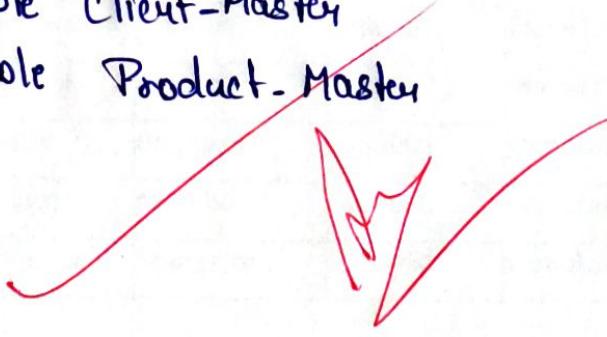
```
CREATE TABLE Client-Master (
    Client-no Varchar(6),
    Name Varchar(20),
    Address1 Varchar(30),
    Address2 Varchar(30),
    City Varchar(15),
    State Varchar(15),
    Pincode Number(10,2),
    Bal-Due Integer
);
```

2)

```
CREATE TABLE Product-MASTER (
    Product-No Varchar(6),
    Description Varchar(20),
    Profit-percentage Number(10,2),
    Unit-Measure Varchar(10),
    Qty-On-Hand Number(10,2),
    Record-Lvl Number(10,2),
    Sell-price Integer,
    Cost-price Integer
);
```

3)

Alter table Client-Master add DOB

- 4) Alter table Client-Master drop column Dob
 - 5) Alter table Product-Master rename column sell-price to sellprice
 - 6) Alter TABLE Product-MASTER rename to ProductMaster
 - 7)
 - a) drop table Client-Master
 - b) drop table Product-Master
- 

ASSIGNMENT - 2

- 1) • Insert into Client-Master (client-no, Name, City, pincode, state, BalDue) values ('C00001', 'Ivan Bayross', 'Mumbai', 400054, 'Maharashtra', 15000)
- Insert into Client-Master Values ('C00002', 'Chhaya Baukar', 'Mumbai', 780001, 'Tamil Nadu', 0)
- Insert into Client-Master values ('C00003', 'Mamta Mazumdar', 'Madras', 400057, 'Maharashtra', 5000)
- Insert into Client-Master Values ('C00004', 'Ashwini Joshi', 'Bangalore', 560001, 'Karnataka', 0)
- Insert into Client-Master Values ('C00005', 'Hansel Galaco', 'Mumbai', 4000060, 'Maharashtra', 2000)
- Insert into Client-Master Values ('C00006', 'Deepak Sharma', 'Manglore', 560050, 'Karnataka', 0)

2)

- a) Select Name from Client-Master
- b) Select * from Client-Master
- c) Select City from Client-Master where City = 'Mumbai'
- d) Select Name, City, State from Client-Master
- e) Select Name from Client-Master where Bal-Due = 15000
- f) Select * from Client-Master Where Name like 'M%'
- g) Select * from Client-Master Where State = 'Maharashtra' OR State = 'Karnataka'

3)

- 0) Update Client-Master
Set City = 'Bangalore'.
Where Client-no = 'C00005'

- b) Update Client-Master
Set bal-Due = 1000
Where Client-no = 'C00001'
- c) Update Client-Master
Set City = 'Pune'
- d)
- a) Delete from Client-Master
b) Delete from Client-Master where state = 'tamil Nadu';

ASSIGNMENT-3

- 1) • Insert into Product-Master values ('P00001', '1.44 floppies', 5, 'piece',
200, 50, 350, 250)
• Insert into Product-Master values ('P03453', 'Monitors', 6, 'piece',
150, 50, 500, 350)
• Insert into product-Master values ('P07865', 'Mouse', 5, 'piece',
100, 20, 600, 450)
• Insert into product-Master values ('Keyboards', 2, 'pieces', 180,
50, 850, 550)
• Insert into Product-Master values ('840 HDD', 205, 'piece', 80,
30, 700, 450)
• Insert into Product-Master values ('1.22 Drive', 5, 'piece', 75,
30, 450, 300)

- 1) SELECT Client-Name from Client-master
- 2) SELECT Client-Name, city From Client-master
- 3) SELECT * From product-master
- 4) SELECT * from Client-master WHERE City = 'Bombay'
- 5) SELECT * from Client-master WHERE Client-no IN ('C00001',
'C00002')
- 6) SELECT * FROM Product-Master Where description IN ('1.44
Drive', '1.22 Drive')
- 7) Select * FROM Product-Master where sell price > 5000
- 8) Select * FROM Client-Master WHERE City IN ('Bombay', 'Delhi',
'Mumbai')

- 9) Select * from product master where sell-price > 2000
And sell-price <= 5000
- 10) SELECT Client-name, city, state FROM Client-master WHERE State J = 'Maharashtra'
- 11) UPDATE product-master SET sell price = 1150 WHERE description = '1.400 floppy disk'.
- 12) DELETE FROM Client-master WHERE Client no = '0001'.
- 13) SELECT description, sell-price, sell-price * 15 AS new-sell-price FROM product-Master WHERE sell price > 1500
- 14) SELECT * FROM Client-master WHERE city Like '%a%'
- 15) SELECT Client-name FROM Client-master WHERE Client-name Like '%a%'
- 16) SELECT * FROM Product-master ORDER By description
- 17) SELECT COUNT(*) FROM Product-Master
- 18) Select Count from product-master
- 19) Select MIN(sell-price) from Product-Master
- 20) Select Max(sell-price) As Max-price, Min(sell-price) As min-price
FROM Product-Master
- 21) SELECT COUNT(*) FROM Product-Master WHERE sell Price >= 1500
- 22) SELECT * FROM product-Master ORDER By sell-price ASC
- 23) SELECT * FROM product-Master ORDER By sell-price DESC

LAB-ASSIGNMENT-4

Table -

- Insert into Employee values (1, 'Vinay', 'Kumar', MCA, 25000, '2001-Sep-27', 'Ghaziabad', 'Y')
- Insert into Employee values (2, 'Suman', 'Verma', MCA, 15000, '2006-Oct-17', 'Meerut', 'Y')
- Insert into Employee values (3, 'Akash', 'Singh', CS, 2000, '2005-Jan-15', 'Kanpur', 'Y')
- Insert into Employee values (4, 'Ragav', 'Kumar', IT, 2000, '2019-Feb-12', 'Ghaziabad', 'N')
- Insert into Employee values (5, 'Rohan', 'Sharma', CS, 21000, '2008-Sep-18', 'Ghaziabad', NULL)
- Insert into Employee values (7, 'Rajan', 'CS', 22000, '2019-Nov-14', 'Ghaziabad', NULL)

- 1) SELECT * from employee where address = 'Ghaziabad'.
- 2) Select distinct dept from employee
- 3) Select ID, Concat(F-name, ' ', L-Name) As full-name, DEPT from employee
- 4) Select Concat(F-Name, ' is working in ', 'Dept', ' Department') As employee from employee
- 5) Select F-name, L-name from employee where DOJ > '2005-Jan-15'
- 6) Update employee set Salary = 25000 where ID = 2
- 7) SELECT ID, CONCAT(F-Name, ' ', L-Name) As Name, salary * 12 As 'annual salary' from employee
- 8) SELECT * ; salary + 300 As gross_salary from employee

- 9) SELECT * from salary WHERE salary Between 15000 And 30000
- 10) SELECT * from employee where DOJ Between '2001-Sep-27' And
'2019-Feb-12'
- 11) SELECT * from Employee WHERE DEPT IN ('MCA', 'CS')
- 12) SELECT * FROM Employee Where DEPT != 'MCA'
- 13) SELECT * FROM Employee where F-Name Like 'R%'
- 14) SELECT * From employee where F-Name Like 'Ro%'
- 15) SELECT * From employee where F-name Like 'R_'
- 16) SELECT * From employee where F-Name Like 'D%'
- 17) Select * from employee where Married = 'y'
- 18) Select * from employee where Married = 'N'
- 19) Select * from employee where married is NULL
- 20) Select * From employee where DEpt = 'MCA' And married = 'Y'



Assignment 5

1)

- a) Select Roll, F-Name, M-Name, L-Name from student
- b) Select Roll, F-Name, city, Area, House-No, from student
- c) Select roll, concat(left(F-name,1), '.', Left(M-Name,1), '.', (L-Name)) as formatted-Name from student
- d) Select * from student order by Sec ASC
- e) Select * from student order by Sec DESC
- f) Select * from student order by Sec, F-Name
- g) Select Roll, concat('****', F-Name, ',', L-Name) as Padded-Name
concat('***', city) as padded-City from student
- h) Select * from student where Div is Null
- i) Select distinct City from student
- j) Select Roll, F-Name Case
 - When Div = 1 Then 'first'
 - When Div = 2 Then 'second'
 - When Div = 3 then 'third'
 - When Div = 0 then 'fail'
 - else 'Not Awarded'
- k) End As Division from student

k) Select 2011; F_Name , Case . Sec When 'A'. Then 1 When 'B' Then
& End As section from student

3) Select EmpCode, Substring(Empname, 4, LEN(Empname)) As
Empname from student

4) Select Concat(Left([First Name], 1), '.', Left([Middle Name], 1),
'.', [Last Name]) As Name from student;

B) Select Sname, CASE When Marks >= 35 Then 'Pass' Else 'Fail'
END As result from student_marks



LAB ASSIGNMENT - 6

- 1) • Insert into Client-Master values ('C00001', 'Pankaj Sharma', 'Delhi', 400054, 'Delhi', 15000, 'PAK@gmail.com')
- Insert into Client-Master values ('C00002', 'Yogesh Sharma', 'Delhi', 700001, 'Delhi', 0, 'YS@gmail.com')
- Insert into Client-Master values ('C00004', 'Ashwini Joshi', 'Chennai', 560001, 'Maharashtra', 5000, 'AS@gmail.com')
- Insert into Client-Master values ('C00005', 'Neha Sharma', 'Mumbai', 400601, 'Maharashtra', 2000, 'NS@gmail.com')
- Insert into Client-Master values ('C00006', 'Dirya Gupta', 'Chennai', 560050, 'Tamil Nadu', 0, NULL);

- 2) • Insert into productsNo values ('P00001', 'T-shirt', 'Piece', 200, 50, 350, 250)
- Insert into products values ('P03453', 'Shirt', 'Piece', 150, 30, 800, 350)
- Insert into products values ('P07865', 'Jeans', 'Piece', 100, 20, 750, 500)
- Insert into products values ('P07985', 'Denim Shirt', 'Piece', 100, 40, 300)
- Insert into products values ('P07975', 'Lyra Tops', 'Piece', 70, 30, 300)
- Insert into products values ('P08065', 'Skirts', 'Piece', 75, 30, 450, 300);

- 3) • Insert into salesforce-Master values ('S00001', 'Kirat', 'A/H.Worli', 'Mumbai', 400002, 'Mah', 3000, 100, 50, 'Grand')
- Insert into SalesMan-Master values ('S00002', 'B/S.Nasiman', 'Mumbai', 3000, 'Mah', 3000, 200, 100, 'Grand')
- Insert into SalesMan-Master values ('S00003', 'Ravi', 'P-Z.Bandra', 'Mumbai', 4000, 'Mah', 3000, 200, 100, 'Grand')
- Insert into SalesMan-Master values ('S00004', 'Ashish', 'A/S.Juhu.Bombay', 'Mumbai', 400044, 'Mah', 3500, 200, 150, 'Grand')

- 4) • Insert into Sales-Order values ('019001', To-Date ('12-Jan-21'), 'dd-mon-yy', '001', 'F', 'N', 'S00001', To-Date ('20-Jan-21'), 'dd-mon-yy'), 'In process'
• Insert into Sales-Order values ('019002', To-Date ('25-Jan-21'), 'dd-mon-yy'), '002', 'P', 'N', 'S00002', To-Date ('27-Jan-21'), 'dd-mon-yy'), 'Cancelled'
• Insert into Sales-Order values ('016865', To-Date ('18-Feb-21'), 'dd-mon-yy'), '003', 'F', 'Y', '50003', To-Date ('20-Feb-21'), 'dd-mon-yy'), 'Fulfilled'
• Insert into Sales-Order values ('010008', To-Date ('24-May-21'), 'dd-mon-yy'), '000B', 'F', 'N', 'S00004', To-Date ('26-May-21'), 'dd-mon-yy'), 'In process').

- 5) • Insert into SALES-Order-Details ('019001', 'P0001', 4, 4)
• Insert into Sales-Order-details ('019001', 'P07968', 2, 1)
• Insert into Sales-Order-details ('019001', 'P07885', 2, 1)
• Insert into Sales-Order-details ('046865', 'P0786', 3, 3)
• Insert into Sales-Order-details ('019003', 'P03453', 2, 2)
• Insert into Sales-Order-details ('046866', 'P06724', 1, 1)
• Insert into Sales-Order-details ('046866', 'P07968', 1, 0)
• Insert into Sales-Order-details ('010008', 'P07975', 1, 0)
• Insert into Sales-Order-details ('010008', 'P00001', 10, 5)



LAB ASSIGNMENT - 7

1) Create table Employees (

Empld Varchar(30) Primary key check (Empld, Like 'E%'),

Ename Varchar(25) NOT NULL,

Dob DATE NOT NULL

DOJ date NOT NULL Check (DOJ > Dob),

Salary Number(7) NOT NULL Check (Salary >= 0)

Adhar Number(12) Unique Check (Length(Adhar) = 12)

);

Q2)

a) Select * from Employees where DOJ = '20-Jan-2021'.

b) Select * from Employees where

c) Select * from employees where Extract(2020) < 2018

d) Select * from employees where extract(May) = 1;

e) Select * from employees where ORDER BY DOJ Desc.

f) Select * from employees where extract(2028) = 2019

g) Select Empld, Ename, DOJ, TRUNCATE(MONTHS_Between, ('10-June-2024', '7-Jan-2020')) / 12) AS Experience Years from Employees

h) Select * from Employee where MOD(Salary, 2) = 1;

i) select * from Employee where Empld('E001', 'E10') AND extract(2020) = 1991

j) select * from Employee Extract('20-Jan-2020') < 22

k) select Empld, Ename, To_char(Dob, 'DD - Jan - 2024') AS DOB from Employee

LAB ASSIGNMENT-8

- 1) Select S.name As "Salesman", C.custname As "Customer Name",
C.city from Salesman S InnerJoin Customer C ON S.city = C.city.
- 2) Select O.order-no, O.purchase-amt, C.cust-name As "Customer Name",
City from Order O InnerJoin Customer C ON O.customer-ID
Where O.purchase-amt > 500 AND < 2000.
- 3) Select C.cust-name As "Customer Name", S.name As "Salesman" ->
InnerJoin Salesman S ON C.salesman-ID = S.salesman-ID.
- 4) Select C.cust As "Customer", C.city, S.name "salesman", S.comm
from Customer C InnerJoin Salesman S ON C.salesman-ID = S.salesman-ID
Where S.comm > 0.12,
- 5) Select C.cust As "Customer", C.city, S.name "salesman", S.city,
S.comm from Cust C InnerJoin Salesman S ON C.salesman-ID = S.salesman-ID
Where S.comm > 0.12 And C.city > S.city
- 6) Select O.order-no, O.order-date, O.purchase-amt, C.cust-name As "Customer Name",
S.name As "Salesman", S.comm from Order O InnerJoin
Customer C ON O.order-ID = C.customer-ID
InnerJoin Salesman S ON C.salesman-ID = S.salesman-ID
Where Order.no = "specific order no".
- 7) Select a.cust-name, a.city, name As Salesman a left Join
Salesman b ON Salesman b.salesman-ID = a.salesman-ID
Order by cust-name.
- 8) Select C.cust-name, b.name As Salesman from Customer a left Join
Salesman b ON Salesman b.salesman-ID = a.salesman-ID
Where a.grade < 300 Order by a.cust-name.
- 9) Select C.cust-name, C.city, O.order-no, O.order-date, O.purchase-amt
from Customer C left Join Order O ON Customer ID = O.customer-ID
Order by O.order-date.

- 10) Select C.cust_name, C.city, O.order_no, O.orddate, purch_amt from Customer C left JOIN Order O ON C.customer_id = O.customer_id
left JOIN Salesman S ON S.salesman_id = C.salesman_id Order by O.ord_date
- 11) Select S.name AS "salesman", COUNT(C.customer_id) AS "Number of Customer" from Salesman S left JOIN Customer C ON S.salesman_id = C.salesman_id Group By S.name Order By S.name;
- (12) Select S.name AS "Salesman", C.cust_name AS "Customer Name", C.city, O.order_no from Salesman S left JOIN Customer C ON S.salesman_id = C.salesman_id left JOIN Order O ON C.customer_id = O.customer_id.
- (13) Select S.name AS "salesman", C.cust_name AS "Customer Name", C.city, O.order_no from Salesman S left JOIN Customer C ON S.salesman_id = C.customer_id left JOIN O ON C.customer_id = O.customer_id
- 14) Select S.name AS "salesman", C.cust_name AS "Customer Name" from Salesman S cross JOIN Customer C.
- 15) Select * from item_mast
- 16) Select I.item_name AS "item Name", I.pro_price AS "price", C.com_name AS "Company Name" from Item-mast I INNER JOIN Company Mast C ON I.pro_com = C.com_id
- 17) Select C.com_name AS "Company Name" from Item-mast I
Inner JOIN Company-mast C.com_id group By com_name
Having AVG(I.pro_price) >= 350

Q1) Select * from emp-details

Q2) Select emp details from "first Name", emp details .emp Income
As Last name from emp details INNER JOIN emp department
ON emp details = emp department .dpt Code where emp -
department .dpt - allotment > 5000

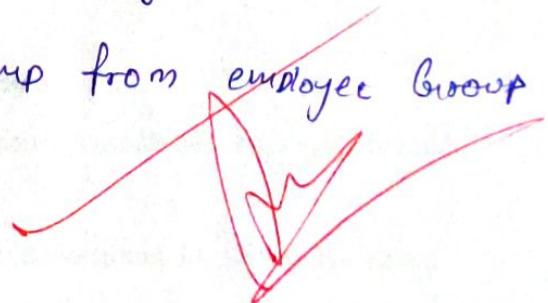
Q3) Select emp dept .dptname from emp details INNER JOIN emp
department ON emp-details .emp = emp department .dpt
code group by emp-department .dpt name Having Count(*) >

✓ ✓ ✓ .

LAB-ASSIGNMENT-9

- 1) Select sum(purch-amt) as TotalPurchaseAmount from Order;
- 2) Select Avg(purch-amt) as AveragePurchaseAmount from Order;
- 3) Select Count(Distinct Salesman-Id) as NumberofSalespeople from Order;
- 4) Select Max(purch-amt) as maxamount from Order;
- 5) Select Min(purch-amt) as minamount from Order;
- 6) Select Cust-Id, MAX(purch-amt) as HighestPurchase from Order
Group By Customer-Id;
- 7) Select Cust-Id, MAX(Ord-date), MAX(Purc-amt) as HighestPurchase
from Order, Where Ord-date = '2012-Aug-17' Group by Cust-Id,
Ord-date
- 8) Select Salesman-Id, MAX(purch-amt) as HighestPurchase from Order from
Order, Where Ord-date = '2012-Aug-17' Group By Salesman-Id;
- 9) Select Cust-Id, Ord-date, MAX(purch-amt) as HighestPurchase from Order
Where Ord-date = '2012-Aug-17' Group By Customer-Id,
Ord-date Having Max(purch-amt) > 2000)
- 10) Select CustomerId, Ord-date, MAX(purch-amt) as MaxPurchase from
Order Where purch-amt Between 2000 And 6000 By CustId, Orddate
- 11) Select CustomerId, Max(purc-amt) as maxamount from Order Where
CustId Between 3002 Group By CustomerID
- 12) Select Count(*) As NumofOrders from Order, Where Ord-date = '2012-Aug-17'
- 13) Select Count(*) As NumofCust from Customer;
- 14) Select Count(*) As CustomerId from Customer Where grade is NOT
NULL

- 15) Select City, Max(grade) As maximum grade from customer group By
City
- 16) Select City , Max(grade) As Numofsalespeople from salesman Group By
City
- 17) Select Count(*) " No of product' from item-mast where Pro-Price
 ≥ 350 ;
- 18) Select Max (DOJ) as latestjoin from employee
- 19) Select min (DOJ) As firstjoin from employee
- 20) Select Count(Distinct Dept) As numofdept from employee
- 21) Select Count (*) As noofemployee from employee where
Dept = 'MCA';
- 22) Select , dept , Max (Salary) As MaxAve . from employee Group By
dept
- 23) Select Count(*) As Total employee from employee
- 24) Select dept, Count (*) As Numberofemp from employee Group
By dept.



LAB ASSIGNMENT - 10

- 1) Select Name From Employee Where Salary > (Select Salary From Employee Where EmpId = 104);
 - 2) Select Name From employee Where Salary > 5000;
 - 3) Select Name, Salary from employee where Salary = (Select Min(Salary) from employee)
 - 4) Select * from Employee Where DOJ = (Select DOJ from employee Where EmpId = 106);
 - 5) Select * from Employee Where DOJ from employee where EmpId = 102)
- ✓ ✓ ✓ ✓ ✓

Experiment - 11

Write down the statement for trigger -

Create table Suppliers (

Suppliers - id numbers;

Supplies - name varchar 2 (4000);

Supplier - address varchar 2 (4000);

);

Create table Audits (

table - name Varchar 2 (255);

transaction - name varchar 2 (10);

transaction - date Date

);

Insert into Suppliers (supplier-id, supplies-name, supplier add.)
values (1, 'ABC', America);

Insert into SuppliersValues (2, 'xyz', 'Australia');

Create or replace Trigger customer - audit - tag After

Insert or update or delete

ON Suppliers

for each row

Declare

l - transaction Varchar 2 (20);

BEGIN

L - transaction := Case

When Updating Then 'update'

When Deleting Then 'Delete'

When Inserting Then 'Insert'

END ;

Declare

Total - rows number (2);

Begin

Update supplier

Set Suppliers_id = supplier_id + 10;

If SQL % Netfound Then

DBMS - Output 1Line ("No customers selected");

Else If SQL % found Then

Total_rows := SQL %. Row Count;

DBMS - Output.. Put - Line (Total - Rows || "customers
selected");

End if;

End;

Procedures -

Create Procedure student - list

As

Begin

Select Name, age, salary

from student

Order By salary;

End;

Alter Procedure student - list

AS

Begin

Set NO Count ON;

Select name, salary

from student;

Order By salary;

END;

Begin

functions :-

Create or Replace function find Customers

Return number IS

total number (2) : = 0;

Begin

Select count(+) into total

from Customers;

Return total

END;

Declare

c number (2);

Begin

c := total customers();

dbms_output.put_line ('total no of customers is');

END;

Declare

a number;

b number;

c number;

function find max (x IN number, y IN number)

Return number

IS

2 number;

Begin

if $x > y$ then

z := x

else

z := y;

End if;

return z

END;

Begin

a := 23

b := 45

c = find max(a,b);

dbms_output.put_line ('maximum of (23,45): ' || c);

END;

Cursor:

Declare

v_emp_id employees.employee_id % Type;

v_emp_name employee.first_name% Type;

cursor emp_cursor IS

Select employee_id , first_name
from employee

Where department_id = 30;

Begin.

Open emp_cursor;

loop

Fetch emp_cursor.JNO v_emp_id;

v_emp_name;

Exit When emp_cursor%Not found;

dbms_output.put_line ('employee id : ' || v_emp_id ||
' Name' : ' || v_emp_name);

END Loop;

ENDs

✓ 16/24