Assignment :- 3

code :- CSA0563

subject :- DBMS

M. Bala Murali
Krishna

192325015

① ER Diagram Design for traffic flow managament system (TFMS)

Entities and Attributes

a) Road

Road ID
Road Name
Length
Speed Limit

b) Intersection

Insertexation ID
Inseration Name
Latitude
Long tude

c) Traffic single

single
single states,
Timer

d) Traffic Data

Traffic Data ID
Time Stamp
speed
longstion Level

Relationship

1) Road - Inseration
(many - to - many)
Road ID (FK)
Inserseton ID (FK)

2) Insertion - traffic
(many - to - many)
Insertion (FK)
traffic (FK)

3) Road - traffic Data
(one - to many)
Road (FK)
Traffic (FK)

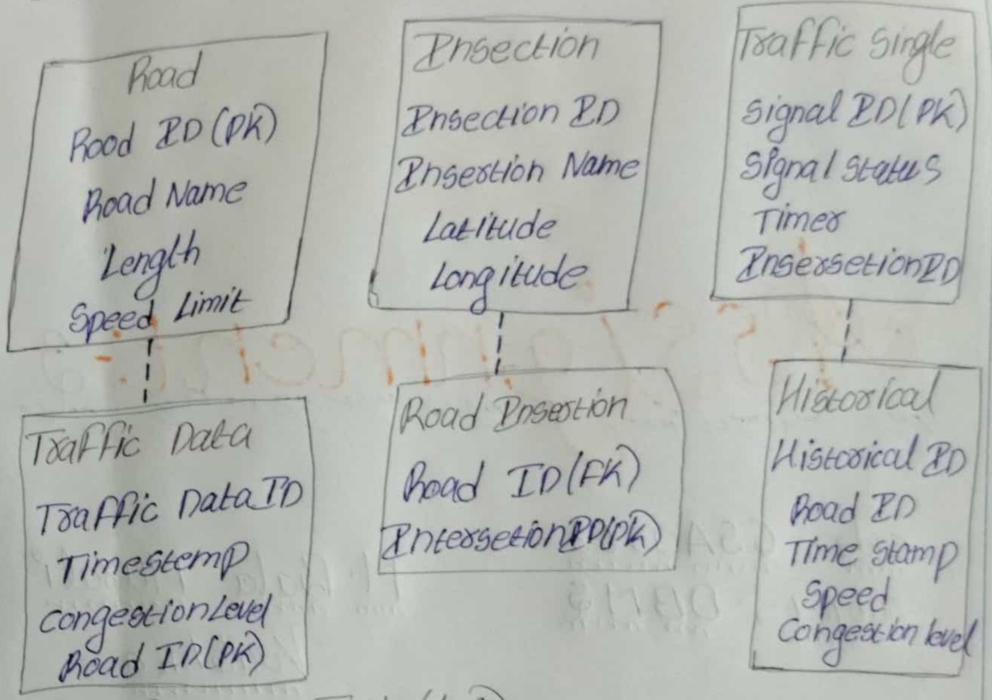
Normatiaalisation conserteations

1) frist normal form (1NF)
Ensure that at attributes contain atomic values

2) second normal from (2NF)
Ensure that non-key attributes are fully atomic values

3) third normal form (3NF)
Ensure that all attributes on depent only primary key

ER Digram (Task3)



Justification Task (4.2)

1) scalability :- The design supports the addtion of new roads, intersetions, traaffic signals and data, record without strutural changes

② Real - Time data processing
The traffic data entity is designed to accom idate high frequency updates with time tam ped record The one tomany relationships.

Question-1 Top 3 Deparement with highest Averge salary

SQL QUERY

```
SELECT
    DeparmentID,
    Deparement Name,
    Ava (salary) As Avg Salary
FROM
    Employees
LEFT JOIN
    Departement on Employees, Deparement ID =
                        Deparment, Deparement ID
GROUP BY
    Department ID, DePartement Name
ORDER By
    Avg salary DESC
LIMIT3;
```

Question2:- Retriveing Hirerarchical catagory Paths

SQL QUERY

```
WITH RECURSIVE category Hiravchy ASC
SELECT
    category ID,
    category Name,
    CAST (castegoryName As VARCHAR(max))
                                As Path
FROM
    categories
WHERE
    Parent Category ID IS NULL
UNION ALL
SELECT
    C. category ID,
    C. category Name,
    CAST (ch. path + 'S' C, categorName As VARCHAR
                 (max )) AS Path
FROM
    categories C
INNER JOIN
    category Hirochy ch. ON C. parent-category ID
                           = CID . categorID
SELECT
    category ID,
    CategoryName,
    Path
```

```
FROM
    category Hireachy
ORDER By
    Path;
```

Question 3:- Total Distinct customers By month

SQL Query!-

```
WITH months ASC
SELCT Date-format(Date-ADD(CURDATE(),
                                Interval 1-1'
    n MONTH), '%y-%m')As month year
FROM
(SELECT@row! =@row+1 As n From (SELECT 1 Union
    SELECT 2 WhINON 2 SELECT) As monTHS
)SELET
    m.month Year As MonTHName
count (DISTINCT o.customer_ID) As customercount
FROM
  months m
LEF IOIN
    order o ON DATE-format (orderdate'% Y-m)
Group By m.month Year
ORDER By
    m.month year!
```

Question 4:- find clasest Locations

SQL QUERY

```
SELECT
  LocationID,
  Location Name,
  Latitude
  Longitude
  (L371* ACOSC
  COSC RADIANS ( Latitude))* COS(RADIANS) Latites
  COS (RADIANS (Longitude) -RADIANS (@Longitude
  SIN (RADIANS (@ Latitude))*SIN(RADANS(Lati)
   As Distance
  From
    Locations
  ORDER BY
     Distance
  LIMIT 5;
```

Question 5:- optimizing Query for ordeeTables

SQL QUERY

```
SELECT
  *
FROM
  orders
WHERE
  order Date >=CURDTE) -INTERval
ORDER By
  order Date DESC;
```

estion 1:- Handling Division Operation

```
QL QUERY
ECLARE
V_numerator NumBeR := 100;
 V_divisor NUMBER;
 V_result NUMBER;
 BEGIN
   V-divisor := &user-divisor
   V- result := v-number /v-divisor
   DBMS-output.put-LINE ("Result of division
  " result ExcePTION
 WHEN ZERO-DIVIDE THEN
   DBMS-output. put-LINE ("Error, Divison By
   zero notalru)
   WHEN OTHERS THEN
   OBMS-output put-LINE ("An unexpected errors
   || SQLERRM);
   END;
```

Question 2:- Updating Rows with FORALL

```
SQL QUERY
DECLARE
   Type emp-Id-array IS TABLE OFF NUMBER;
   TYPE salary-array IS TABLE OFF NUMBER;
V -emp- ids emp-id -array :- emp-id (101,102,103);
V -salary -array := sasaly - array (500,600,700);
 BEGIN
   FORALL IN 1.V-emp-ids-count
    UPDATE Employess
    sey salary = salaray +V-salaries (i);
    WHERE Employee ID =Y- emp-ids (i)
 ComIT;
  DBMS-output, Put_Line ('salaries is updaled
  Exeption
  WHEN OTHERS THEN
  DBMS -output.put-Line ("An erros occsed "
   Roll back;
  END;
```

Question 3:- Implementing Nested Table procedure

SQL Query

```
CREATE OR REPLACE proce Get_Employees
                              _By-Dept(
    P_dept-id IN NUMBER
    P_emp-list out SYS-REFCURSOR
)As
 BEGIN
   OPEN P_emp_list FOR
   SELECT EmployeeID, First Name, Last Name
    FROM Employess
 WHERE Department ID = P_dept_id
 END;
```

Question 4:- using cursor variables and Dynamic SQL

SQL QUERY

```
DECLARE
    TYPE emp-cursor IS REA CURSOR!
      V- emp - cursor emp_cursor
      V- salary - threshod NUMBER := 50000;
      V- Employee-id employee, Employee% TYPE
BEGIN
 OPEN V-emp-cursor For
 SELECT Employee ID / First Name, Last name
 FROM: Employees
 WHERE salary % >V-salary - threshort;
 Loop
 FETCH V-emp- cursor IN, V- employee -id, V- FirsName
                  , V-last Name;
 Exit WHEN V- emp -cursor , % NoT foun D;
 DBMS- output- _LINE ("ID"/ he enploy fullname
 || V- firstname ||" "|| V-last name)
 END LOOP;
 ClosE V-emp-cursor;
   Exception
      WHEN OTHERS THEN
   DBMS-output. put_LINE ("In error occured"
   || SQLERm))
   END;
```

Questions5:- Designing Pipelined Function for sales Data

SQL QUERY

```
~~Cre~~ CREATE OR REPLACE TYPE sales-Record object
    Order ID NumBER
    CustomerID NUMBER
    OrderAmout NUMBER
 );
 CREATE OR REPLACE TYPE sales-Table
  IS tables sales-Record
CREATOR REPLACE Function get-sales, data(
P-month IN Number P-year IN NUMBER)
RETURN Sales-table PIPELINED
 AS
BEGIN
WHERE EXTRACT (monTH from orderdate)
                = P-month
 AND EXTRACT(Year FROM orderdate=
                P-year
)
Loop
  PIPE Row(Sales-Record (order ID
                 .r v-rcv-customer ID)):
  END LOOP;
 END;
```