ASSIGNMENT-03

NAME :- P.Soma Sekhar Reddy

REGNO :- 192311116

SUB :- DBMS

SUBCODE :- CSA0563

Question 1:-

ER Diagram Question:- Traffic flow management system

Scenario :-

you are treated with designing an Entity, Relationship (or) diagram for a traffic flow management system.

Task 1 :- Entity identification and Attributes.

Roads	Intersection	Traffic signals	Traffic data
Road ID(PK)	intersection id (PK)	signal Id(PK)	Traffic data-ID(PK)
Road Name	intersection name	signal status	Road-ID(FK)
length(m)	Latitude	Timer	TimeStamp
Speed limit(km)	Longitude	Intersection Id	Speed level

Task 2:- Relationship modeling:-

Relationships:-

1. Roads to intersections:-

* one road up can connect to multiple intersection.
* An intersection can be connected by multiple roads.

2. Intersection to Traffic signals:-

* one intersection can host multiple traffic data entities.

Cordinality and optionality:-

1) Roads to Intersection:-

* one road can connect to zero or more intersection.
* one intersection can connect to one or more roads.

2) Intersection to Traffic signals:-

* one road can have zero or more traffic signals.
* one traffic signal must be associated with one intersection.

Roads to traffic data:-

* one road can have zero or more traffic data entities.
* one traffic data entry must be associated with one road.

Task 4:-

Justification and Normalization:-

Scalability:-

* The design allows for easy addition of new roads, intersection traffic signals, and traffic data entities without modifing status.

Real Time data processing:-

* Real-time traffic data integration is facilated by the traffic data.

Efficient Traffic mangement:-

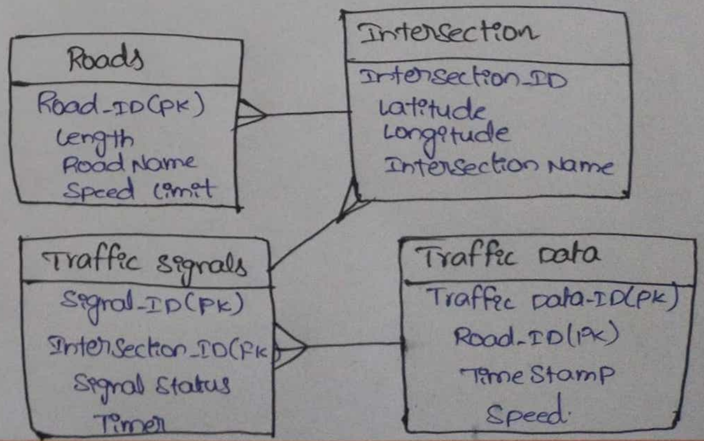
* The clear separation of entities.

Delivarebles:-

ER diagram:- Provided above in plain text format listed in task 1.

Entity defination:- Listed in task 1.

Task 3:- ER Diagram Design:-



Question -2 :-

question-1 :- Top 3 Department with highest average salary

Sql Query

```
WITH AVG Salaries ASC
 SELECT
     d.department ID,
     d. department Name,
     AVG(e.Salary) AS Avg Salary
 FROM
    Departments.d
    LEFT JOIN Employees e IN department ID = e.Department ID
  GROUP BY
     d. department ID,
     d. department Name,
     )
  SELECT
     department ID,
     Department Name,
     )
  SELECT
     Department ID
     Department Name
     Avg salary
   FROM
     Avg Salaries
   ORDER BY
     Avg Salary DESC NULL LAST
```

Question-ii :- Retriving Hierachical category paths

Sql Query

```
WITH RECURSIVE category Path ASC
SELECT
      c.category ID,
      c.category Name,
      c. parent category ID,
  CAST(c. category Name AS VARCHAR (255)) AS path
FROM
      categories c
WHERE
    c. parent category ID IS NULL
SELECT
    c. category ID,
    c. category Name,
    c. parent category ID,
CAST(cp.path||'>'||c. category Name AS VARCHAR(255) AS path.
FROM
    category c
   INNER JOIN category path of ON c.parent category ID = category ID
SELECT
      category ID
      category Name,
      path.
 FROM
    category paths;
```

Question-iii :- Total Distinct customers by month.

SQL Query

```
SELECT
        Date_Format (order date, (%.y-%m)
  AS month name 1
        COUNT (Distinct customer ID) AS
        CUSTOMER COUNT
FROM
    orders
WHERE
    order date ≥ Date_sub(curdate(), internal year)
GROUP By
    MONTH Name
```

question-iv:- Finding closest location

```
SELECT
        Location ID,
        location Name,
        latitude,
        longitude
        (637* AS OS(Radius(37.7747)* AS
        (Radius (latitude)) -AIOS( Radius(-122.4194)
          Radius (longitude) is in (Radius (latitude))
          AS distance.
```

question v:- optimizing query for order table.

```
 SELECT * FROM orders
 WHERE order date ≥ Date sub (curdate(), Internal day)
  ORDER By
    order Date Base;
```

Question-3 :-

Task 1 :- Handling Division operation.

SQL query

```
Declare
      dividend number =100;
        divisor number;
        result number;
BEGIN
      divisor =&divisor;
BEGIN
    result  = dividend/divisor;
    DBMS_output_line(Result::// result);
EXCEPTION
        is NOT allowed);
    End;
    End;
```

Task-2 :- updating rows with for All

SQL Query

```
 DECLARE
        Emp-ids DBMS-SQL number_Table :=
        DBMS_SQL. number_Table (101, 102, 103);
        Salary-line DBMS-SQL number.Table :=
        DBMS-SQL number_Table(1000, 2000, 3000);
BEGIN
      For All in Emp-ids . First----Emp-ids... LAST
UPDATED EMPLOYEES
      Set salary = salary* salary-in cc(i)
 WHERE
        Employee ID = Emp-ids(i);
  END;
```

TASK-3 :- Implementing Nested-Table procedure

SQL Query

```
CREATE OR Replace PROCEDURE Get-Employees -By-dept {
            P.dept-id IN Number
            P.Emp-list OUT SYS-RefucrSor
        )AS
      OPEN P_emp_list FOR
SELECT
    Employee ID, First Name ,Last Name
FROM
    Employees
WHERE
    Department_ID = p.dept-id,
   END;
```

Task-4 :- using cursor variables and Dynamic SQL

SQL QUERY

```
 DECLARE
        Type Emp-cursor IS REF CURSOR;
        V_Emp_ cursor Emp-cursor;
        V_Salary_threshold Number : = 50000;
        V-employee-id Employee ID% Type;
        v_first_name Employees. First-Name% Type;
        v-Last_name Employees. LAST-Name %. Type;
BEGIN
    OPEN V_Emp_cursor FOR
SELECT
   Employee ID, First Name, Last Name
FROM
   Employees
WHERE
   Salary > V_Salary. threshold;
LOOP
   FETCH V_Emp_cursor INTO V_Employee-id; V-First name; V_Last_name;
EXIT WHEN V_Emp_cursor %. NOT FOUND;
DBMS_OUTPUT_ LINE ('ID': //v_employee-id // 'name' // v_first name// " "//
                                                       V_last name)

 END LOOP;
 CLOSE V_Emp_cursor;
EXCEPTION
 WHEN others THEN
 DBMS_output . Put_LINE ('AN Error occured : '// SQL ERM);
```

Task-5 :- Desiging Pipelined Function for Sales Data

SQL Query :-

```
CREATE OR Replace Type Sales_Record object (
          Order ID Number,
          CustomerID Number,
          Order-Amount Number,
        );
CREATE OR Replace Type Sales_Table IS Table of Sales_Record
CREATE OR Replace Function get-Sales_date (P.month IN Number, P_year
                                                IN Number)

 RETURN Sales-table pipelened.
  AS
 BEGIN
 WHERE Extract (month from orderdate) = P.month.
 AND   extract (year from order date) = P. year
 )
  LOOP
    Pipe Row (Sales_ Record (Order ID, V-Customer ID))
 END Loop;
 END;
```