ASSIGNMENT -3

P. Rohit Reddy

192372033

1. ER Diagram design for Traffic Flow managment system (TFMS)

Entities and Attributes (Task 1)

ⓐ Road
- Road ID
- Road Name
- Length
- Speed limit

ⓑ Intersection
- Intersection ID
- Intersection Name
- Latitude
- Longitude

ⓒ Traffic Signal
- Signal ID
- Signal Status
- Timer

ⓓ Traffic data
- Traffic Data ID
- Time Stamp
- Speed
- congestion level

Relationship (Task 2)

1) Road - Intersection
(many-to-many)
Road ID (FK)

2) Intersection - Traffic
Intersection (FK)
Traffic (FK)

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

Normalisation considerations

1) First Normal Form (1NF)
ensure that all attributes contain atomic values

2) Second Normal Form (2NF)
ensure that non-key attribute

3) third Normal Form (3NF)
ensure that all attributes depend only prime key

ER Diagram (Task 3)

Road	Intersection	Traffic signal
Road ID (PK)	Intersection ID	Signal ID
Road Name	Intersection name	Signal Status
Speed limit	Latitude	Timer
	Longitude	

Traffic Data	Road Intersection	Historical
Traffic Data ID	Road ID	Historical
Timestamp	Intersection ID	Road ID
Speed		Timestamp
Road ID		Speed

Justification

Scalability: The design support the addition of new roads, intersection traffic signals and data ifle ds without structural changes.

2) Real Time data Processing
Traffic Data entity is designed to accomodate high frequency updates with Time records The one to-many relationship b/w Road and traffic.

(2) question 1 : top 3 dept with highest Avg salary

SQL query

```
Select
    DeptID,
    DeptName,
    AVG(Salary) AS AvgSalary

From
    Employees
Left Join
    Dept on employee.DeptID = Dept.DeptID

GroupBy
    DeptID, DeptName.

Order by
    Avg Salary DESC.
Limit 3;
```

Question 2: Retrieving Hierarchical Paths

SQL Query

```
With RecursiveCategories AS (
SELECT
    CategoryID,
    CategoryName
    CAST ( CategoryName AS VARCHAR(MAX)) AS Path
From
    Categories
WHERE
UNION ALL
 SELECT
    C.CategoryID,
    C.CategoryName,
    Cast(ch.Path +'>'+C.CategoryName AS VARCHER)

From
    CategoriesC

  INNER JOIN
  CategoryHierarchy CH ON C.ParentCategoryID
                        = CD.CategoryID
  )
  SELECT
      CategoryID,
      CategoryName,
      Path.
  From
      Category
```

QUST 3 total distinct customer By month

SQL QUERY:

```
WITH months AS (
SLECT DATE - Format(DateADD(CURRDATE(), IN MONTH-1*
```

```
h months; %Y-%m) AS MONTH-Year

FROM
 (SELECT @row := @row + 1 AS n FROM (SELECT 1 UNION ALL
      SELECT 2 UNION 2 SELECT ) AS months
 )
 SELECT
      m.month_year AS monthName
  COUNT (DISTINCT o.CUSTOMER_ID)
  FROM
      monthsm
  LEFT JOIN
      ORDERS o ON DATE_FORMAT(o.orderdue, '%Y-%m')
  GROUP BY
       m.month_year
ORDER BY
      m.month_year;
```

Question: Finding closed Locations

```
SELECT
     locationID,
     locationname,
     latitude
     longitude,
     (6371 *
     acos (RADIANS (@latitude) * cos(RADIANS(latitude) *
      cos (RADIANS (longitude - RADIANS (@longitude))
```

```
SIN(RADIANS(@latitude)) * SIN(RADIANS(latitude))
  ) AS Distance
FROM
     Locations
  ORDER BY
      Distance
  Limit 5;
```

Question: Optimizing query for orders table

```
SQL QUERY
SELECT
     *
FROM
     address
WHERE
     orderDue >= CURDATE() - INTERVAL 7 DAY
ORDER BY
     orderDue DESC;
```

Question 4: Using cursor variable 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 employees.EID%type;
    v_first_name employees.first_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 INT _employee_id, v_Pfname
        EXT_OUTPUT_LINE ('ID ||v_employee_id||name'

      END LOOP;

      close v_emp_cursor;
```

Exception
  WHEN OTHERS THEN
    DBms_output.put_LINES ('An Error occurred'||SQL_ERRM)

END;

Question: Designing Pipelined Function for sales data

SQL query

```
CREAT OR REPLACE TYPE Sales_Record object
      OrderID NUMBER
      );
CREATE OR REPLACE TYPE sales_Table. SalesRecord

CREAT REPLACE function get_sales_data (P_month IN)

  Begin
  WHERE Extract (month from order date) = P_month
  AND EXTRACT (YEAR FROM order date ) = P_year
  )
  LOOP
    PIPE ROW (Sales_Record (order ID, Customer ID
    END LOOP
    END;
```