

CSA0563

Data Base Management Systems
for Distributed system.

P. Hema Sanjana Reddy

192371037

CS - Biosciences.

Assignment - 2

1. ER Diagram Design for Traffic flow management system (TFMS)

Task 1 (Entities and Attributes)

a) Road b) Intersection c) Traffic signal

RoadID	IntersectionID	SignalID
RoadName	IntersectionName	SignalStatus
Length	Latitude	Timer
SpeedLimit	Longitude	

d) Traffic data

TrafficDataID
Timestamp
Speed
Congestion level

Relationships (Task 2)

① Road-Intersection (many-to-many)

RoadID (FK)
IntersectionID (FK)

② Intersection-Traffic (many-to-many)

Intersection (FK)
Traffic (FK)

③ Road-Traffic data (one-to-many)

Road (FK)
Traffic (FK)

Task 4 (Normalisation)

1) First Normal Form (1NF)

Ensure that all attributes contain atomic values

2) Second Normal Form (2NF)

Ensure that non-key attributes are fully functional.

3) Third Normal Form (3NF)

Ensure that all attributes are on dependent only primary key.

4.2

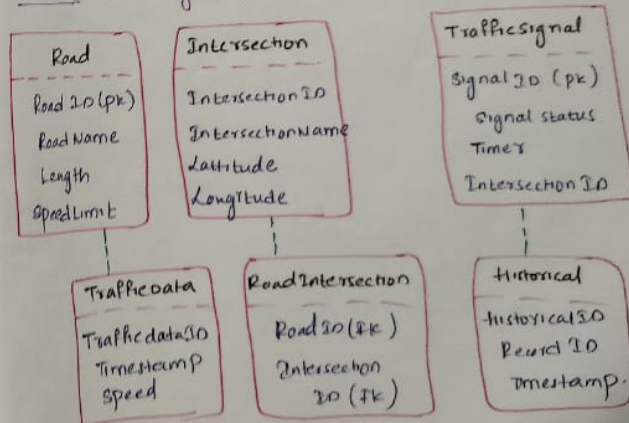
Justification :-

1. Scalability The design supports the addition of new roads, intersection, traffic signals, and data, records without structural changes.

2. Real-time data processing :-

The traffic data entity is designed to accommodate high frequency updates with time records. The one-to-many relationships b/w road & traffic.

Task 3 (ER-diagram).



(1) Top 3 Departments with highest Average Salary.

SQL Query :-

```
SELECT
    DepartmentID,
    DepartmentName,
    Ave(salary) As AvgSalary
FROM
    Employees
LEFT JOIN
    Departments ON Employees.DepartmentID =
    Departments.DepartmentID.
GROUP BY
    DepartmentID, DepartmentName.
ORDER BY
    AvgSalary DESC
LIMIT 3;
```

(11) Retrieving hierarchical category paths

SQL query :-

WITH RECURSIVE category hierarchy ASC,

```
SELECT
    category ID
    Category Name
    CAST Category Name AS VARCHAR(max) AS Path
FROM
    Categories
```

WHERE

Parentcategory ID is NULL

UNION ALL

```
SELECT
    C.categoryID,
    C.CategoryName,
    CAST (ch.Path + '>' + C.CategoryName AS VARCHAR) AS Path
FROM
    Categories C
```

INNER JOIN

CategoryHierarchy ch ON C.parent category ID
= ch.category ID

```
)
SELECT CategoryID, CategoryName, Path
FROM Category Hierarchy
ORDER BY Path;
```

iii) Total distinct customers by month

SQL query :-

WITH months ASC

```
SELECT DATE-Format (Date-ADD (CONVERT(DATE), Interval in
month); 'Y. Y - 'M. M') AS month Year
```

FROM

(SELECT @row = @row+1 ASn FROM (SELECT 1 UNION
ALL SELECT 2 UNION 2 SELECT 3) AS months
)

SELECT

m.monthyear AS monthName

COUNT (DISTINCT o.customer_ID) AS CustomerCount

FROM

months m

LEFT JOIN

orders ON DATE_FORMAT (o.orderdate, 'Y-' + m) = m.monthyear

GROUP BY

m.monthyear

ORDER BY

m.monthyear

(IV) Finding Closest locations SQL Query
SQL Query

SELECT

Location ID,

Location Name,

Latitude,

Longitude,

(1371 * COS (

COS (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 ;

V) Optimising Query for orders Table SQL Query

SQL Query :-

SELECT *

FROM orders

WHERE

Orderdate = (GETDATE() - INTERVAL DAY

ORDER BY

Orderdate DESC;