

CSA0563

Data base Management
systems for
distributed system

B.Bhargavi
192311136.

C.S.E
Assignment - 3

for

ER Diagram Design for Traffic flow management system (TFMS)

- Entities and Attributes (Task 1)
- a) Road
 - Road ID
 - Road Name
 - Length
 - Speedlimit
 - b) Intersection
 - Intersection ID
 - Intersection name
 - Latitude
 - Longitude
 - c) Traffic signal
 - Signal ID
 - Signal status
 - Timer

d) Traffic Data

- Traffic Data ID
- Time stamp
- speed
- congestion level

Relationships (Task 2)

- 1) Road - Intersection (Many-to-many)

Road ID(FK)

IntersectionID(FK)

- 2) Intersection - Traffic (many-to-many)

IntersectionID(FK)

Traffic(FK)

- 3) Road - Traffic Data (One-to-many)

Road(FK)

Traffic CFK

systems

ER Diagram (Task 3)

Road
RoadID(PK)
Road Name
Length
speed limit

Intersection
IntersectionID
Intersection name
Latitude
Longitude

Traffic signal
SignalID(PK)
signal status
Timer
Intersection

Traffic Data
TrafficDataID
Time stamp
speed
congestion level
RoadID(FK)

Road
Intersection
RoadID(FK)
IntersectionID(FK)

Historical
HistoricalID
RoadID
Time stamp
Speed
Congestion level

(Task 4.1)

Normalisation considerations

1) First Normal Form (1NF)

Ensure that all attributes contain atomic values

2) Second Normal Form (2NF)

Ensure that non-key attributes are fully functionally.

3) Third Normal form (3NF)

Ensure that all attributes are dependent only primary key.

Justification Task(4.2)

1 Scalability :-

The design supports the addition of new roads, intersections, traffic signals, and data reloads without structural changes.

2. Real-Time data processing:-

The traffic data entity is designed to accommodate high frequency updates with timestamped records. The one-to-many relationships between Road and traffic

3. Efficient Traffic management:-

The clear separation of entities

Deliverables

ER diagram provided above in plain text format entity definition listed in task Relationship Descriptions

Justification Document

2. Question-1 Top 3 Departments with Highest Average salary

SQL Query

```
SELECT DepartmentID, DepartmentName, Avg(Salary) AS AvgSalary
FROM Employees
LEFT JOIN Departments ON Employees.DepartmentID = Departments.DepartmentID
GROUP BY DepartmentID, DepartmentName
ORDER BY AvgSalary DESC
LIMIT 3;
```

Question 2 Retrieving Hierarchical category paths

SQL Query

```
WITH RECURSIVE CategoryHierarchy AS C
SELECT categoryID, categoryName, categoryName AS VARCHAR(max) AS path
```

```
FROM categories
WHERE parentcategoryID IS NULL
UNION ALL
SELECT c.categoryID, c.categoryName, c.categoryName + ',' + c.categoryName AS VARCHAR(max) AS path
FROM categories c
INNER JOIN
    categoryHierarchy ch ON c.parentcategoryID = ch.categoryID
)
```

```
SELECT categoryID, categoryName, path
FROM categoryHierarchy
ORDER BY Path;
```

Question 3 - Total distinct customers by month

SQL query

```
WITH months AS C
SELECT DATE_FORMAT(Date_ADD(CURDATE(), INTERVAL n Month), '%Y-%m') AS monthYear;
```

```

    FROM
    SELECT @row:=@row+1 AS n FROM (SELECT 1 UNION ALL
    ) t1
    SELECT 2 UNION ALL SELECT 3 AS months
    )
    SELECT
        m.monthyear AS monthname,
        count(DISTINCT o.customer_ID) AS customercount
    FROM
        months.m
    LEFT JOIN
        orders o ON DATE_FORMAT(o.orderdate, '%Y-%m') = m.year
    Group BY
        m.monthyear
    ORDER BY
        m.monthyear;

```

Question 4:- Finding closest locations

SQL Query

```

    SELECT
        LocationID,
        LocationName,
        Latitude,
        Longitude,

```

Credit: Oracle
 $\text{distance} = \sqrt{\cos(\text{lat}_1) \cos(\text{lat}_2) \cos(\text{lon}_1 - \text{lon}_2) + \sin(\text{lat}_1) \sin(\text{lat}_2) \sin(\text{lon}_1 - \text{lon}_2)}$

```

    FROM
        Locations
    ORDER BY
        distance
    LIMIT 5;

```

Questions:- Optimizing Query for orders Table

SQL Query

```

    SELECT *
    FROM Orders
    WHERE
        orderdate >= CURDATE() - INTERVAL 7 DAY
    ORDER BY
        orderdate DESC;

```

```

    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.customerID) AS customercount
    FROM
        months.m
    LEFT JOIN
        orders o ON DATE_FORMAT(o.orderdate,'%Y-%m') = m.year
    GROUP BY
        m.monthyear
    ORDER BY
        m.monthyear;

```

Question 4 - Finding closest locations

SQL Query

```

SELECT
    LocationID,
    LocationName,
    Latitude,
    Longitude,

```

$$\begin{aligned}
 & \text{Coseno} \\
 & \cos(\text{RADIAN}(latitud)) * \cos(\text{RADIAN}(latitud)) + \\
 & \cos(\text{RADIAN}(longitud)) * \cos(\text{RADIAN}(longitud)) + \\
 & \sin(\text{RADIAN}(latitud)) * \sin(\text{RADIAN}(latitud))
 \end{aligned}$$

AS Distance

```

FROM
    Locations
ORDER BY
    DISTANCE
LIMIT 5;

```

Question 5 - Optimizing Query for orders Table

SQL Query

```

SELECT *
FROM Orders
WHERE
    orderdate > CURDATE() - INTERVAL 7 DAY
ORDER BY
    orderdate DESC;

```