**PRACTICE PAPER**

**Topics for practice**

BA- The topic needs to cover

1. CTE

2. Sub-Queries

3. Query

4. Joins

5. Functions

6. MySQL CONNECTIVITY

7. ER-diagram

8. date and time function

9. Keys

10. Schemas

NOTE:-  Revise all materials

**Questions for practice**

**Question 1: What is a Common Table Expression (CTE) in SQL?**

**Question 2: How do you define a CTE in SQL?**

**Question 3: How is a CTE different from a temporary table in SQL?**

**Question 4: What is the scope of a CTE in SQL?**

**Question 5: How can you reference a CTE within a SQL query?**

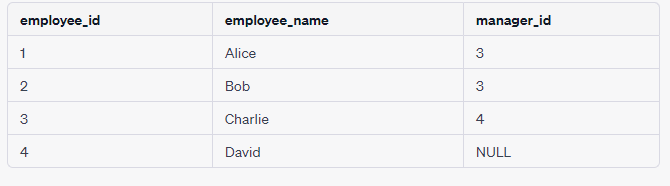
**Question 6: Retrieve a list of employees and their respective managers using a CTE.**

**Use a sample dataset to demonstrate the use of a Common Table Expression (CTE) in SQL.**

Suppose we Created a hypothetical dataset representing a company's employee structure. The dataset contains information about employees and their managers, structured as follows:

**Employee Table:**

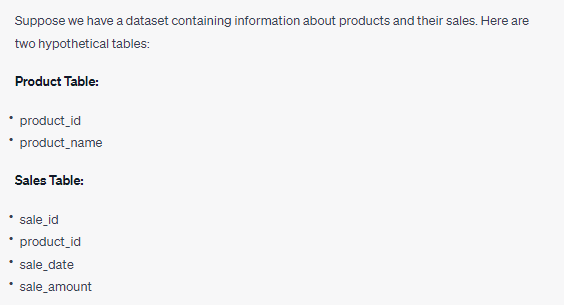
* employee\_id
* employee\_name
* manager\_id

****

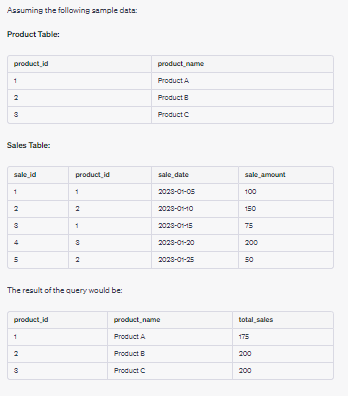
**Sample Answer for Question6:- is given in last page.**

**Question 7: Find the total sales for each product in the "Product" table. And use a CTE to achieve this.**

Certainly, let's work through an example of using a Common Table Expression (CTE) in SQL with a practical dataset.

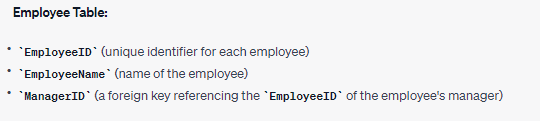


Sample data/ouput:-



**Question 8: Certainly! Let's work through a practical example using a hypothetical dataset to demonstrate the use of Common Table Expressions (CTEs) in SQL. In this scenario, we'll use a simple employee hierarchy dataset.**

Suppose we have an employee table with the following schema:



**(A): Display the names of all employees and their respective managers.**

**(B) : List all employees and their direct reports.**

**(C): Calculate the number of levels in the employee hierarchy for each employee.**

**Question 9: In an inventory database, identify all products that have never been ordered. Write an SQL query based on pairwise and non-pairwise to find the product details.**

Hint:- To identify all products that have never been ordered in an inventory database, you can use both a pairwise comparison (with a subquery) and a non-pairwise approach (with a LEFT JOIN). Here are SQL queries for both methods:

**Question 10: In a student records database, find all students who have a GPA greater than the average GPA of students in their department. Write an SQL query based on scalar, inline, pairwise, non-pairwise, multi-row, single row and correlated sub queries type to extract this information.**

Hint :- You can find all students who have a GPA greater than the average GPA of students in their department using scalar, inline, and correlated sub queries. Here's how you can write SQL queries for each of these sub query types:

**Question 11:In a company database, you want to find all employees who have the same job title as the employee with EmployeeID 101. Write an SQL query based on pair wise , non-pairwise, inline, corelated,multi-row,scalar & single row to accomplish this.**

**Hint:** To find all employees who have the same job title as the employee with EmployeeID 101, you can use various SQL query techniques. Here are SQL queries based on pairwise, non-pairwise, inline, correlated, multi-row, and scalar subqueries:

**Question 12: In a sales database, you need to find the total sales amount for each product category and list only those categories where the total sales amount is greater than the average total sales amount for all categories. Write an SQL query for this by using single row, multi-row, scalar, inline, correlated, pairwise and non-pairwise.**

**Hint:** To find the total sales amount for each product category and list only those categories where the total sales amount is greater than the average total sales amount for all categories, you can use different SQL query techniques. Here are SQL queries based on single row, multi-row, scalar, inline, correlated, pairwise, and non-pairwise subqueries:

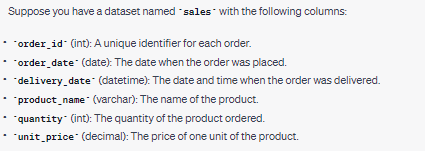
**Question 13: You have a library database and want to find all books that are currently checked out. Write an SQL query pairwise, non-pairwise, inline, correlated, multi-row, and scalar subqueries to retrieve the book titles.**

**Hint:** To find all books that are currently checked out in a library database, you can use different SQL query techniques, including pairwise, non-pairwise, inline, correlated, multi-row, and scalar subqueries. Here are SQL queries for each of these subquery types:

**Question 14:** You are given a list of numbers, and you need to find the sum of all even numbers in the list using Python's built-in functions. Write a Python function to accomplish this task.

numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

**Question 15: Create the given dataset and perform the FUNCTION on questions:-**

****

Q. How many orders were placed in the month of May 2023?

Q. What was the total revenue generated in the first quarter (January to March) of 2023?

Q. Calculate the average delivery time in hours for all orders.

Q. Which product had the highest total sales quantity in 2023?

**Question 16:**

****

**Q.** How many orders were placed in the month of February 2023?

Q. What is the average total amount of orders placed in March 2023?

Q. How many days elapsed between the first and last order placed in the dataset?

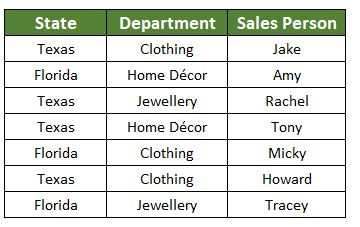
Q. What is the total sales made on weekends (Saturday and Sunday) in the dataset?

**Question 17:** Why ER Model is used in DBMS? Draw the ER Model for Student information with respective attributes

**Question 18:** Create database named as Company.

1. Use Company database.
2. Create a table for the following ER Model named as Emp\_details.
3. Add six records to the above table using INSERT command.
4. View the above table using SELECT command.

**Question 19:** Apply the Sub-query for the following table.



1. Write sub query to display highest state from the above table.
2. Write sub query to display the name of the sales person who work in Clothing departments.
3. Find out the count state from the above table.

**Question 20:** Apply the Sub-query for the following table.

1. Write sub query to display highest salary from the above table.
2. Write sub query to display the name of the employee having maximum salary.
3. Find out the average salary from the above table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Emp\_id** | **Emp\_name** | **Emp\_salary** | **Emp\_add** | **Emp\_mobno** |
| 0101 | Rahul | 10000 | Delhi | 9005689758 |
| 0102 | Ajit | 20000 | Mumbai | 9091659878 |
| 0103 | Nupur | 30000 | MP | 8025947895 |
| 0104 | Swati | 40000 | UP | 9958974635 |

**Question 21:** Convert following table into 1NF, 2NF and 3NF. Please add appropriate columns into it using following queries.

1. Create table, Update table and Alter table.
2. Candidate Key, Unique Key, Primary Key and Foreign Key.

|  |  |  |  |
| --- | --- | --- | --- |
| **Movie\_details** | **Member\_details** | **Rental\_detalis** | **Rental\_discount** |
| MV001 Pirates of the Caribbean Action | MB100 Janet Jones Express | 15-Aug-17 50 | 0 |
| MV002 Clash of the Titans Action | MB100 Janet Jones Express | 17-Aug-17 50 | 0 |
| MV003 Cinderella Children | MB101 Rakesh N Gold | 15-Aug-17 25 | 5 |
| MV002 Clash of the Titans Action | MB102 Anita Gupta Platinum | 20-Jul-17 50 | 10 |
| MV005 Annabelle Horror | MB103 Samir M Gold | 23-Jul-17 45 | 5 |

**Question 22:** Convert following table into 1NF, 2NF and 3NF. Please add appropriate columns into it using following queries.

1. Create table, Update table and Alter table.
2. Candidate Key, Unique Key, Primary Key and Foreign Key.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Emp\_id** | **Emp\_name** | **Emp\_salary** | **Emp\_add** | **Emp\_mobno** |
| 0101 | Rahul | 10000 | Delhi | 9005689758 |
| 0102 | Ajit | 20000 | Mumbai | 9091659878 |
| 0103 | Nupur | 30000 | MP | 8025947895 |
| 0104 | Swati | 40000 | UP | 9958974635 |

**Question 23:** Convert following table into 1NF, 2NF and 3NF. Please add appropriate columns into it using following queries.

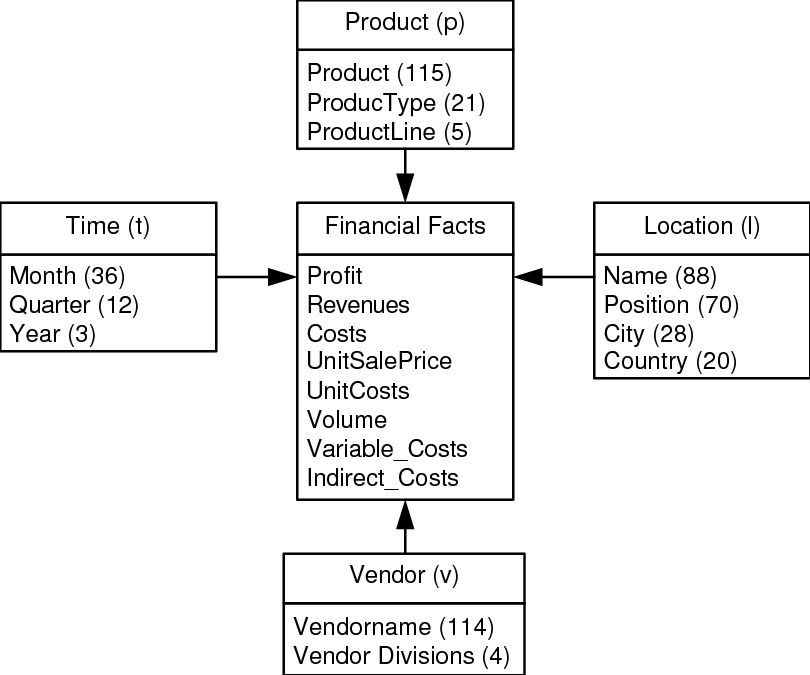
1. Create table, Update table and Alter table.
2. Candidate Key, Unique Key, Primary Key and Foreign Key.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **product \_id** | **product\_name** | **product \_salary** | **product \_add** | **product \_mobno** |
| 0101 | Nil | 10000 | Punjab | 9005689758 |
| 0102 | Siya | 60000 | AP | 9091659878 |
| 0103 | Nitu | 60000 | MP | 8025947895 |
| 0104 | Sita | 80000 | UP | 9958974635 |

**Question 24: Create the given dataset on WORKBENCH with given table reference and perform required keys & show the table on workbench.**

**After creating a dataset connect sql on jupyter notebook by using any connection method and show any one table similar to workbench from given figure.**

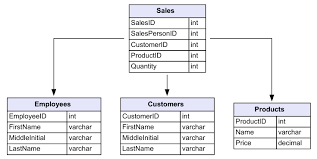
**At the end Draw ER-diagram as per the created tables related to your dataset(as per given figure)**



**Question 25: Create the given dataset on WORKBENCH with given table reference and perform required keys & show the table on workbench.**

**After creating a dataset connect sql on jupyter notebook by using any connection method and show any one table similar to workbench from given figure.**

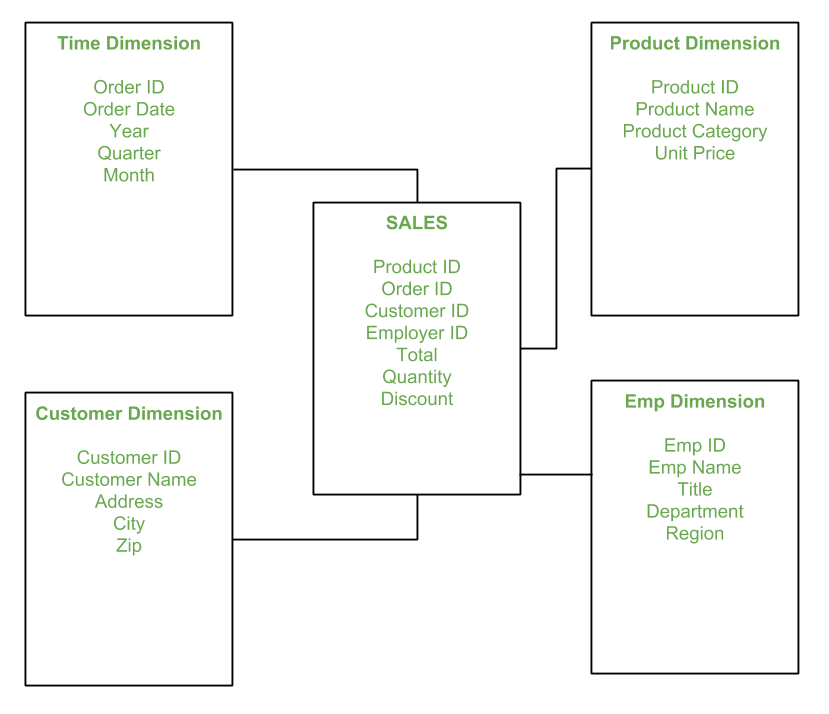
**At the end Draw ER-diagram as per the created tables related to your dataset(as per given figure)**

****

**Question 26: Create the given dataset on WORKBENCH with given table reference and perform required keys & show the table on workbench.**

**After creating a dataset connect sql on jupyter notebook by using any connection method and show any one table similar to workbench from given figure.**

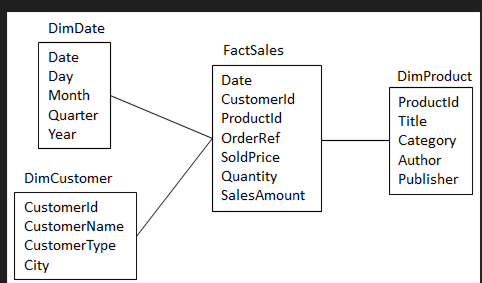
**At the end Draw ER-diagram as per the created tables related to your dataset(as per given figure)**



**Question 27: Create the given dataset on WORKBENCH with given table reference and perform required keys & show the table on workbench.**

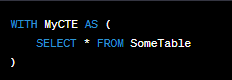
**After creating a dataset connect sql on jupyter notebook by using any connection method and show any one table similar to workbench from given figure.**

**At the end Draw ER-diagram as per the created tables related to your dataset(as per given figure)**

****

**ANSWERS**

**ANSWER-Q2:-**

****

**ANSWER-Q6:-**

****

**Code:-**

WITH EmployeeCTE AS (

SELECT e.employee\_id, e.employee\_name, e.manager\_id, m.employee\_name AS manager\_name

FROM employees e

LEFT JOIN employees m ON e.manager\_id = m.employee\_id

)

SELECT employee\_name, manager\_name

FROM EmployeeCTE;

Explanation of above code:-

* The **EmployeeCTE** is defined, joining the **employees** table to itself to match each employee's **manager\_id** with their manager's **employee\_id**.
* Then, the main query selects the **employee\_name** and **manager\_name** from the CTE, providing a list of employees and their corresponding managers.

By Running the query with the provided data would result in a table displaying employee names alongside their manager names:

**Output:-**

****