

		Advance SQL assingment no 7					
QUESTTION 1		What is a Common Table Expression (CTE), and how does it improve SQL query readability?					
ANSWER		A Common Table Expression (CTE) is a temporary named result set in SQL that you can use inside a SELECT, INSERT, UPDATE, or DELETE statement.					
		It is created using the WITH keyword.					
		How CTE Improves SQL Query Readability :-					
		Makes Complex Queries Easy to Understand					
		Breaks big queries into logical parts					
		Each CTE has a meaningful name					
		Replaces Nested Subqueries					
		Avoids deeply nested SELECT statements					
		Makes queries clean and structured					
		Improves Maintainability					
		Easy to modify or debug					
		Change logic in one place instead of many					

QUESTION 2	Why are some views updatable while others are read-only? Explain with an example.				
ANSWER	<p>A view is a virtual table based on a SELECT query.</p> <p>Whether a view is updatable or read-only depends on how clearly SQL can map changes in the view back to the base table(s).</p> <p>Conditions for an Updatable View</p> <p>A view is usually updatable if:</p> <ul style="list-style-type: none"> It is based on a single table No JOIN No GROUP BY No aggregate functions (SUM, AVG, COUNT) No DISTINCT No subquery in SELECT <p>If one row in the view maps to exactly one row in one base table, the view is updatable.</p> <p>If not, the view is read-only.</p>				

QUESTION 3	What advantages do stored procedures offer compared to writing raw SQL queries repeatedly?				
ANSWER	What Are Stored Procedures? A stored procedure is a precompiled set of SQL statements stored inside the database and executed using a single call.				
	Reusability				
	Write SQL logic once				
	Reuse it many times				
	Better Performance				
	Stored procedures are precompiled				
	Execution plan is stored in cache				
	Improved Security				
	Users can be given permission to execute procedures, not access tables directly				
	Helps prevent SQL Injection				

QUESTION 4	What is the purpose of triggers in a database? Mention one use case where a trigger is essential.						
ANSWER	A trigger is a special database object that automatically executes when a specific event occurs on a table or view.						
	Events can be:						
	INSERT						
	UPDATE						
	DELETE						
	Timing can be:						
	BEFORE						
	AFTER						
	Main Purposes of Triggers						
	Enforce Business Rules Automatically						
	Ensures rules are followed no matter which application accesses the database						
	Example:						
	Prevent negative salary						
	Prevent deleting critical records						
	Maintain Data Integrity						
	Keeps data consistent and accurate						
	Especially when related tables must stay in sync						

QUESTION 5	Explain the need for data modelling and normalization when designing a database.					
ANSWER	When designing a database, our goal is to store data in a way that is:					
	Accurate					
	Consistent					
	Efficient					
	Easy to maintain					
	Data modelling helps us plan the structure.					
	Normalization helps us remove problems in that structure.					
	Data Modelling – Design Before Building					
	What Is Data Modelling?					
	Data modelling is the process of defining how data is structured, stored, and related in a database.					
	What entities exist? (Employee, Customer, Order)					
	What attributes do they have?					
	How are they related?					
	Normalization – Clean and Organize Data					
	What Is Normalization?					
	Normalization is the process of organizing data to:					
	Remove redundancy					

		Avoid anomalies					
		Improve data integrity					

Dataset (Use for Q6–Q9)

QUESTION 6

Write a CTE to calculate the total revenue for each product (Revenues = Price × Quantity), and return only products where revenue > 3000.

ANSWER

The screenshot shows the MySQL Workbench interface with a query editor and a results grid.

Query Editor:

```
MySQL Workbench - pw_skills_DA_Batch
File Edit View Query Database Server Tools Scripting Help
Navigator: joins* SQL File * x
Schemas: college, company_db, company_md, db_name, employee, mitesh, orders, product
Tables: products, sales
Administration Schemas Information
Table: products
Columns:
product_id int PK
product_name varchar(100)
category varchar(50)
price decimal(10,2)

SELECT
    product_id,
    total_revenue
FROM product_revenue
WHERE total_revenue > 3000;
```

Result Grid:

product_id	total_revenue
1	4800.00
2	8000.00
3	5000.00
4	5500.00

Action Output:

#	Time	Action	Message	Duration / Fetch
41	12:31:46	WITH product_revenue AS (SELECT products . SUM(price * quantity) AS total_revenue FR... Error Code: 1054. Unknown column 'products' in field list'	0.000 sec
42	12:32:27	WITH product_revenue AS (SELECT product_id , SUM(price * quantity) AS total_revenue F... Error Code: 1054. Unknown column 'price' in field list'	0.000 sec
43	12:34:18	WITH product_revenue AS (SELECT s.product_id , SUM(p.price * s.quantity) AS total_revenue... 4 row(s) returned	0.000 sec / 0.000 sec

QUESTION 7

Create a view named VW-CATEGORY SUMMARY that shows Category, TotalProducts, AveragePrice.

ANSWER

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database schema. Under the "product" schema, there is a "Views" folder containing a view named "vw_category_summary".
- SQL Editor:** Displays the SQL code for creating the view:

```
75      c.category_name AS Category,
76      COUNT(p.product_id) AS TotalProducts,
77      AVG(p.price) AS AveragePrice
78  FROM product.c
79  JOIN product p
80    ON c.category_id = p.category_id
81  GROUP BY c.category_name;
82
83
84
85
86 • CREATE VIEW VW_CATEGORY_SUMMARY AS
87 SELECT
88   Category,
89   COUNT(*) AS TotalProducts,
90   AVG(Price) AS AveragePrice
91 FROM Products
92 GROUP BY Category;
93
94
95 • SELECT * FROM `VW-CATEGORY SUMMARY`;
```
- Output:** Shows the execution log with three entries:

#	Time	Action	Message	Duration / Fetch
156	17:01:01	CREATE VIEW 'VW-CATEGORY SUMMARY' AS SELECT c.category_name AS Category, COUNT(p....	Error Code: 1146. Table 'product.product' doesn't exist	0.000 sec
157	17:03:49	CREATE VIEW VW_CATEGORY_SUMMARY AS SELECT Category, COUNT(*) AS TotalProducts,...	0 row(s) affected	0.016 sec
158	17:07:45	SELECT * FROM 'VW-CATEGORY SUMMARY'	Error Code: 1146. Table 'product.vw-category summary' doesn't exist	0.016 sec

QUESTION 8

Create an updatable view containing ProductID, ProductName, and Price. Then update the price of ProductID = 1 using the view

ANSWER

```

100  SELECT
101    Product_ID,
102    Product_Name,
103    Price
104   FROM Products;
105
106  • UPDATE vw_ProductInfo
107    SET Price = 25.00 -- example new price
108   WHERE Product_ID = 1;
109
110  -- To see the category summary view
111
112  • SELECT *
113    FROM vw_CATEGORY_SUMMARY;
114
115  • SELECT *
116    FROM vw_ProductInfo;
  
```

Category	TotalProducts	AveragePrice
electronics	2	412.500000
furniture	2	4000.000000

Product_ID	Product_name	Price
1	keyboard	25.00
2	mouse	80.00
3	chair	150.00
4	desk	1000.00

QUESTION 9	Create a stored procedure that accepts a category name and returns all products belonging to that category.							
ANSWER	sorry i didnt get the answer this question please help me for that particular reason							

QUESTION 10		Create an AFTER DELETE trigger on the product able that archives deleted product rows into a new table product archive .The archive should store ProductID, ProductName, Category, Price, and DeletedAt						
ANSWER		sorry i didnt get the answer this question please help me for that particular reason						