

# Advanced SQL

## Q1. Common Table Expression (CTE)

A CTE is defined using WITH clause. It creates a temporary result set that improves readability and modularizes complex queries.

## Q2. Updatable vs Read-only Views

A view is updatable if based on a single table without GROUP BY, DISTINCT, joins, or aggregates. Otherwise, it becomes read-only.

## Q3. Advantages of Stored Procedures

Stored procedures are precompiled, improve performance, enhance security, reduce redundancy, and centralize business logic.

## Q4. Purpose of Triggers

Triggers execute automatically on INSERT, UPDATE, or DELETE events. Example: auditing deleted records.

## Q5. Data Modelling & Normalization

Data modelling structures database design. Normalization reduces redundancy and maintains data integrity.

## Products Table (Used for Q6–Q10)

```
CREATE TABLE Products ( ProductID INT PRIMARY KEY, ProductName VARCHAR(100),  
Category VARCHAR(50), Price DECIMAL(10,2), Quantity INT );
```

## Q6. CTE for Revenue > 3000

```
WITH ProductRevenue AS ( SELECT ProductID, ProductName, (Price * Quantity) AS  
Revenue FROM Products ) SELECT * FROM ProductRevenue WHERE Revenue > 3000;
```

## Q7. Create View vw\_CategorySummary

```
CREATE VIEW vw_CategorySummary AS SELECT Category, COUNT(*) AS TotalProducts,  
AVG(Price) AS AveragePrice FROM Products GROUP BY Category;
```

## Q8. Updatable View & Update

```
CREATE VIEW vw_ProductSimple AS SELECT ProductID, ProductName, Price FROM Products;  
UPDATE vw_ProductSimple SET Price = 5000 WHERE ProductID = 1;
```

### **Q9. Stored Procedure**

```
DELIMITER // CREATE PROCEDURE GetProductsByCategory(IN cat_name VARCHAR(50)) BEGIN  
SELECT * FROM Products WHERE Category = cat_name; END // DELIMITER ;
```

### **Q10. AFTER DELETE Trigger**

```
CREATE TABLE ProductArchive ( ProductID INT, ProductName VARCHAR(100), Category  
VARCHAR(50), Price DECIMAL(10,2), DeletedAt TIMESTAMP ); DELIMITER // CREATE TRIGGER  
trg_AfterDeleteProduct AFTER DELETE ON Products FOR EACH ROW BEGIN INSERT INTO  
ProductArchive (ProductID, ProductName, Category, Price, DeletedAt) VALUES  
(OLD.ProductID, OLD.ProductName, OLD.Category, OLD.Price, NOW()); END // DELIMITER ;
```