

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 tableNo JOINNo GROUP BYNo aggregate functions (SUM, AVG, COUNT)No DISTINCTNo 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 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

MySQL Workbench

pw_skills_DA_Batch x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- college
- company_db
- company_md
- db_name
- employee
- Tables
- Views
- Stored Procedures
- Functions
- mitsuh
- orders
- product
 - Tables
 - products
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
 - sales
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers

Administration Schemas

Information

Table: products

Columns:

- product_id int PK
- product_name varchar(100)
- category varchar(50)
- price decimal(10,2)

SQL File 3*

Don't Limit

```

60 GROUP BY s.product_id
61 )
62 SELECT
63     product_id,
64     total_revenue
65 FROM product_revenue
66 WHERE total_revenue > 3000;
67
68
69
70
  
```

Result Grid

product_id	total_revenue
1	4800.00
2	8000.00
3	5000.00
4	5500.00

Result 11 x

Read Only

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(p.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

Object Info Session

QUESTION 7

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

ANSWER

MySQL Workbench

pw_skills_DA_Batch x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

Stored Procedures

Functions

mitesh

orders

product

Tables

products

Columns

product_id

product_name

category

price

Indexes

Foreign Keys

Triggers

sales

Views

vw_category_summary

Category

TotalProducts

AveragePrice

Stored Procedures

Functions

Administration Schemas

Information

View: vw_category_summary

Columns:

Category varchar(50)

TotalProducts bigint

AveragePrice decimal(14,6)

SQL File 3*

Don't Limit

```

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`;
96
97
98
99

```

Output

Action Output

#	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

Object Info Session

QUESTION 8

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

ANSWER

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHMANS

Views Objects

Stored Procedures

Functions

Views

product

Tables

Columns

product_id

product_name

category

price

Indexes

Foreign Keys

Triggers

Views

vw_category_summary

Category

TotalProducts

AveragePrice

vw_productinfo

Stored Procedures

Administration

Schemas

Information

View: vw_category_summary

Columns:

Category varchar(50)

TotalProducts bigint

AveragePrice decimal(14,6)

vw_CATEGORY_SUMMARY SQL

```

180 SELECT
181     Product_ID,
182     Product_Name,
183     Price
184 FROM Products
185
186 * UPDATE VW_ProductInfo
187 SET Price = 25.00 -- example new price
188 WHERE Product_ID = 1
189 -- To see the category summary view
190
191 * SELECT *
192 FROM vw_category_summary

```

Result Grid

Category	TotalProducts	AveragePrice
Electronics	2	412.500000
Furniture	2	4000.000000

Output

Action Output

Time	Action	Message	Duration / Feat
182 17:12:31	UPDATE VW_ProductInfo SET Price = 25.00 -- example new price WHERE ProductID = 1	Error Code: 1054 Unknown column 'ProductID' in 'where clause'	0.000 sec
183 17:12:47	UPDATE VW_ProductInfo SET Price = 25.00 -- example new price WHERE Product_ID = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.016 sec
184 17:13:47	SELECT * FROM vw_CATEGORY_SUMMARY	2 row(s) returned	0.000 sec / 0.000 sec

Object Info

Session

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHMANS

Views Objects

Stored Procedures

Functions

Views

product

Tables

Columns

product_id

product_name

category

price

Indexes

Foreign Keys

Triggers

Views

vw_category_summary

Category

TotalProducts

AveragePrice

vw_productinfo

Stored Procedures

Administration

Schemas

Information

View: vw_category_summary

Columns:

Category varchar(50)

TotalProducts bigint

AveragePrice decimal(14,6)

vw_CATEGORY_SUMMARY SQL

```

185
186 * UPDATE VW_ProductInfo
187 SET Price = 25.00 -- example new price
188 WHERE Product_ID = 1
189 -- To see the category summary view
190
191 * SELECT *
192 FROM vw_CATEGORY_SUMMARY

```

Result Grid

Product_ID	Product_Name	Price
1	keyboard	25.00
2	mouse	800.00
3	chair	2000.00
4	desk	5000.00

Output

Action Output

Time	Action	Message	Duration / Feat
182 17:12:31	UPDATE VW_ProductInfo SET Price = 25.00 -- example new price WHERE Product_ID = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.016 sec
183 17:12:47	SELECT * FROM vw_CATEGORY_SUMMARY	2 row(s) returned	0.000 sec / 0.000 sec
184 17:13:47	SELECT * FROM vw_ProductInfo	4 row(s) returned	0.000 sec / 0.000 sec

Object Info

Session

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 table 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						