

## SQL QUERIES

1. SELECT \* from \_\_\_\_\_

This query just gives the output of all the data stored in that particular table/view.

[illegible]

2. SELECT CONCAT helps us to combine two columns and then give the output.

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following queries:

```
1 • SELECT * FROM product;
2
3 • SELECT CONCAT(FirstName, ' ', LastName) AS FullName from employees;
4
5 • SELECT DISTINCT Name AS product_Name FROM product;
6
7 • SELECT Name AS product_name, price_per_unit FROM product WHERE price_per_unit < 1000;
8
9 • SELECT Name, price_per_unit FROM product ORDER BY Name asc;
10
```

The Results tab shows the output of the first query, displaying a list of employee names:

FullName
Aditya Agrawal
Aditya Gupta
Aditya Bansal
Aditya Kansal
Aditya Jindal
Aditya Singhal
Aditya Tingal
Aditya Goyal
Aditya Goel
Aditya Goerka
Aditya Preet
Aditya Mittal
Aditya Bhandal
Aditya Gang
Aditya Nangal

The Output tab shows the execution log:

#	Time	Action	Message	Duration / Fetch
51	22:32:56	SELECT LastName from employees LIMIT 0, 2000	15 row(s) returned	0.000 sec / 0.000 sec
52	22:33:21	SELECT FirstName from employees LIMIT 0, 2000	15 row(s) returned	0.000 sec / 0.000 sec
53	22:33:21	SELECT LastName from employees LIMIT 0, 2000	15 row(s) returned	0.000 sec / 0.000 sec
54	22:33:21	SELECT FirstName + LastName LIMIT 0, 2000	Error Code: 1054. Unknown column 'FirstName' in 'field list'	0.000 sec
55	22:33:32	SELECT FirstName + LastName from employees LIMIT 0, 2000	15 row(s) returned	0.000 sec / 0.000 sec

3. SELECT \_\_\_\_\_ as \_\_\_\_\_ helps us to get the data of that particular column with the field name that we choose.

The screenshot displays the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows a tree view for the 'dbms\_project' database, including tables like authentication, bill, cart, cartitem, coupon, customer, customers, employees, order\_item, orders, product, reviews, seller, and sellers. The main editor window is titled 'SQL File 3\*' and contains 11 SQL queries. The 'Result Grid' at the bottom shows the output of the first query, which selects all columns from the 'product' table. The results are listed in a table with one column, 'product\_Name'.

```
1 • SELECT * FROM product;
2
3 • SELECT FirstName + ' ' + LastName As "Full Name" FROM employees;
4
5 • SELECT DISTINCT Name AS product_Name FROM product;
6
7 • SELECT Name AS product_name, price_per_unit FROM product WHERE price_per_unit < 1000;
8
9 • SELECT Name, price_per_unit FROM product ORDER BY Name asc;
10
11 • SELECT EmailID FROM employees WHERE FirstName LIKE '%shu%' AND LastName LIKE '%s1%'.
```

product_Name
Toy Car
Toy Bike
Toy game
Mens Shirt
Female Shirt
Mens Shots
Mens shoes
Female shoes

4. SELECT \_\_\_\_\_ as \_\_\_\_\_ , \_\_\_\_\_ as \_\_\_\_\_ where \_\_\_\_\_ >=<= helps us to get the data which is greater than or equal to in that particular column.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

Filter objects

dbms\_project

Tables

- authentication
- bill
- cart
- cartitem
- coupon
- customer
- customers
- employees
- order\_item
- orders
- product
- reviews
- seller
- sellers

Views

- mobile

Stored Procedures

Functions

Administration Schemas

Information: Schema: dbms\_project

SQL Queries x SQL File 3\*

Limit to 2000 rows

```
4
5 • SELECT DISTINCT Name AS product_Name FROM product;
6
7 • SELECT Name AS product_name, price_per_unit FROM product WHERE price_per_unit < 1000;
8
9 • SELECT Name, price_per_unit FROM product ORDER BY Name asc;
10
11 • SELECT EmailID FROM employees WHERE FirstName LIKE '%ahu%' OR LastName LIKE '%sal%';
12
13 • SELECT Name, price_per_unit FROM product ORDER BY Name asc;
14
```

Result Grid

Filter Rows:

Export: Wrap Cell Content:

	product_name	price_per_unit
▶	Toy Car	100
	Toy Bike	120
	Toy game	150
	Toy Bike	160
	Mens Shirt	200
	Female Shirt	220
	Mens Shots	250
	Female Shirt	270
	Mens shoes	300
	Female shoes	320

5. SELECT \_\_\_\_\_ where order by \_\_\_\_ asc helps us to get the data from the field in ascending order.

The screenshot shows the MySQL Workbench interface. On the left, the 'Navigator' pane displays the 'dbms\_project' schema with a list of tables including authentication, bill, cart, cartitem, coupon, customer, customers, employees, order\_item, orders, product, reviews, seller, and sellers. The 'Schemas' tab is selected, showing the 'dbms\_project' schema.

The main window displays a list of SQL queries. The first query is highlighted:

```
SELECT DISTINCT Name AS product_Name FROM product;
```

The 'Result Grid' pane shows the results of the first query, displaying a table with two columns: 'Name' and 'price\_per\_unit'. The results are as follows:

Name	price_per_unit
Female Shirt	220
Female Shirt	270
Female shoes	320
Mens Shirt	200
Mens shoes	300
Mens Shots	250
Toy Bike	120
Toy Bike	160
Toy Car	100
Toy game	150

6. `SELECT lower (_____)` from \_\_\_\_\_ helps us to get the data in lower form from that column.

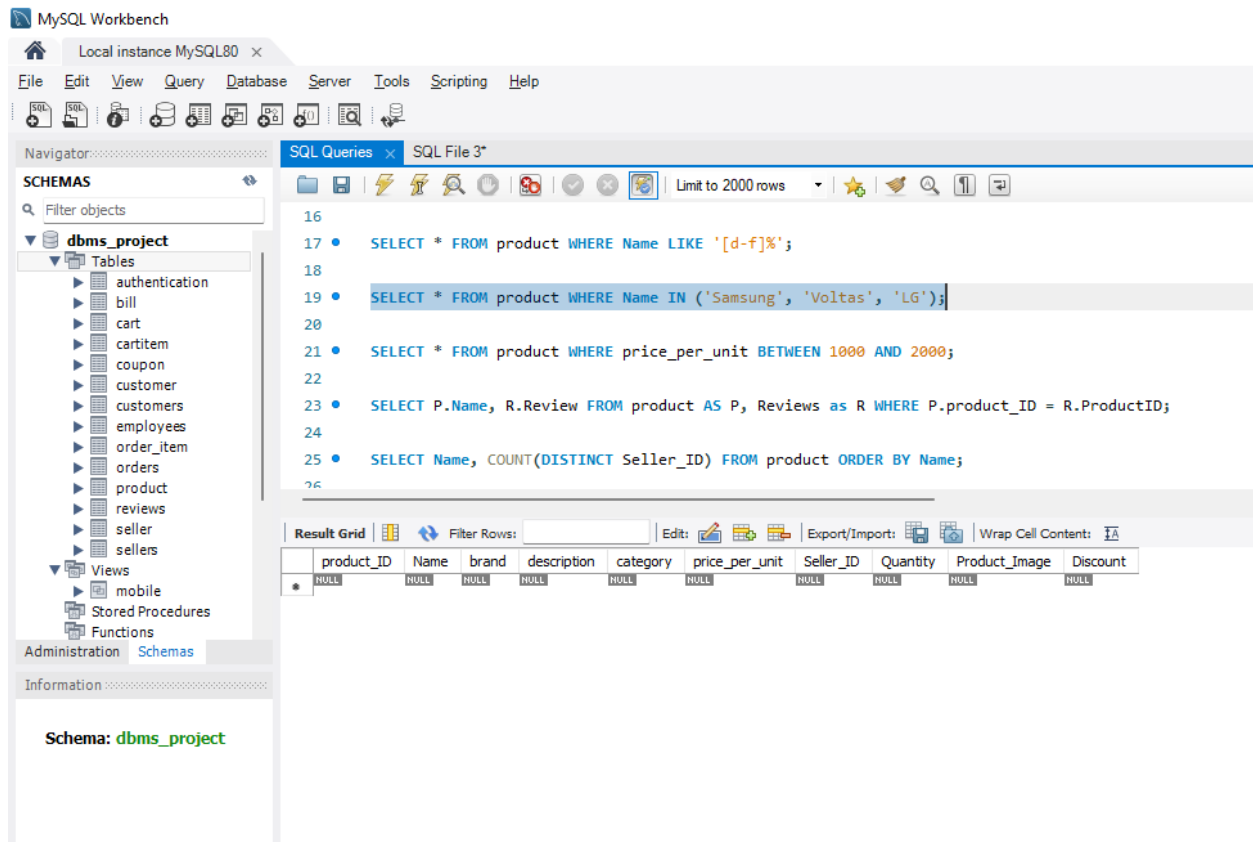
The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays a tree view of the 'dbms\_project' database, including tables like authentication, bill, cart, cartitem, coupon, customer, customers, employees, order\_item, orders, product, reviews, seller, and sellers. The 'authentication' table is selected. The main window shows a SQL query in the 'SQL File 3\*' editor:

```
7 • SELECT Name AS product_name, price_per_unit FROM product WHERE price_per_unit < 1000;
8
9 • SELECT Name, price_per_unit FROM product ORDER BY Name asc;
10
11 • SELECT EmailID FROM employees WHERE FirstName LIKE '%ahu%' OR LastName LIKE '%sal%';
12
13 • SELECT Name, price_per_unit FROM product ORDER BY Name asc;
14
15 • SELECT lower(email) FROM authentication;
16
17 • SELECT * FROM product WHERE Name LIKE 'fd-f1%'.
```

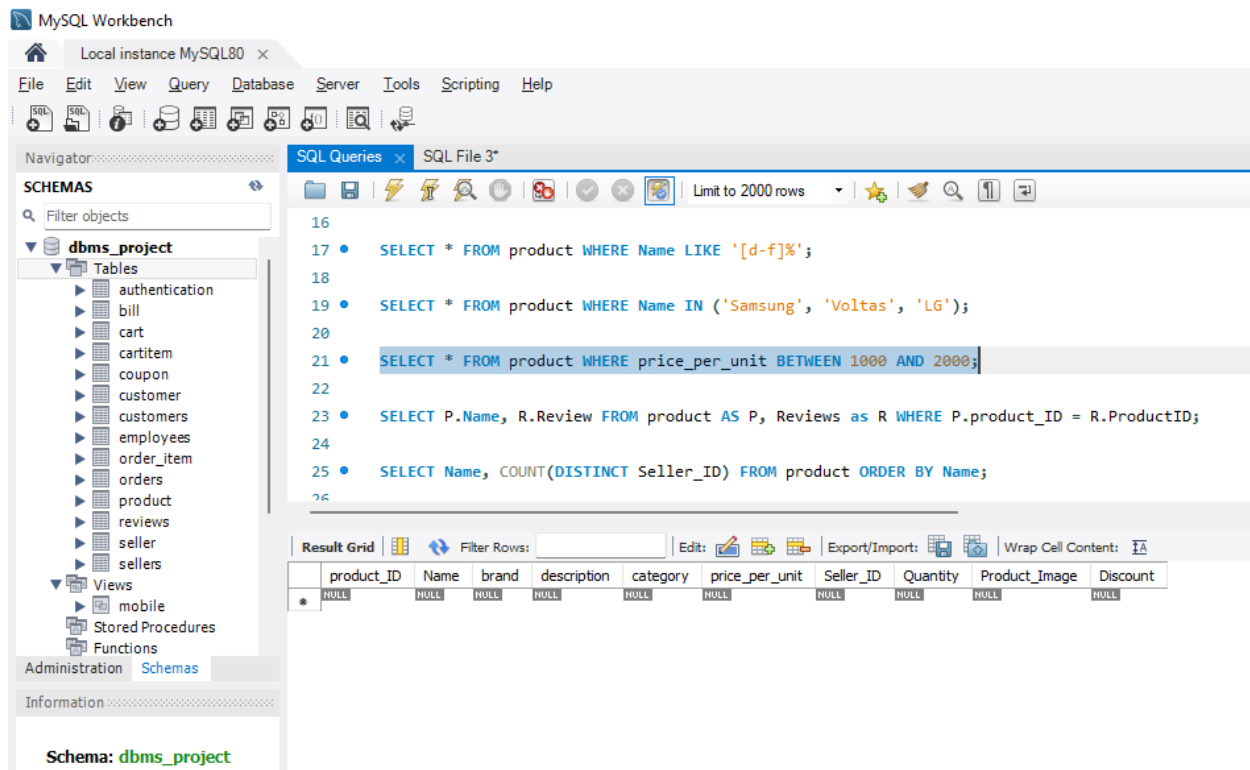
The 'Result Grid' at the bottom shows the results of the query in line 15, displaying the 'lower(email)' column for the 'authentication' table. The results are as follows:

lower(email)
lrsylvjkbpxqbaqbh@kvhrr.com
noj29988@boofx.com
eab83033@boofx.com
noj29988@boofx.com
lrsylvjkbpxqbaqbh@kvhrr.com
six22739@boofx.com
sif85172@boofx.com
brw95713@uooos.com
hth12386@qopow.com
zli19456@mzico.com

7. `SELECT * from _____ where ____ in ('_____', '_____')` helps us to get the data from the table where the data is in the fields provided.



8. `SELECT * from ____ where _____ between 1000&2000` helps to get the data where the values are between the given fields.



The screenshot displays the MySQL Workbench interface. On the left, the 'Navigator' pane shows the 'dbms\_project' schema with a list of tables including authentication, bill, cart, cartitem, coupon, customer, customers, employees, order\_item, orders, product, reviews, seller, and sellers. The 'Schemas' tab is selected. The main editor area shows a SQL query in 'SQL File 3\*':

```
16
17 • SELECT * FROM product WHERE Name LIKE '[d-f]';
18
19 • SELECT * FROM product WHERE Name IN ('Samsung', 'Voltas', 'LG');
20
21 • SELECT * FROM product WHERE price_per_unit BETWEEN 1000 AND 2000;
22
23 • SELECT P.Name, R.Review FROM product AS P, Reviews as R WHERE P.product_ID = R.ProductID;
24
25 • SELECT Name, COUNT(DISTINCT Seller_ID) FROM product ORDER BY Name;
26
```

The 'Result Grid' at the bottom shows the results of the selected query (line 21). The grid has columns: product\_ID, Name, brand, description, category, price\_per\_unit, Seller\_ID, Quantity, Product\_Image, and Discount. The first row shows all NULL values.

product_ID	Name	brand	description	category	price_per_unit	Seller_ID	Quantity	Product_Image	Discount
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

The 'Schema: dbms\_project' is indicated at the bottom left.



9. `SELECT _____ from _____ limit = 5` helps us to get the data from the table till the limit provided.

The screenshot shows the MySQL Workbench interface. The 'Navigator' pane on the left displays the 'dbms\_project' schema with various tables and views. The 'SQL Queries' pane in the center shows a list of queries, with the 37th query selected: `SELECT Name FROM product LIMIT 5;`. The 'Information' pane at the bottom shows the 'Schema: dbms\_project' and the 'Result Grid' with the following data:

Name
Toy Car
Toy Bike
Toy game
Toy Bike
Mens Shirt

## 10. CASE

```
{when Quantity> ____  
Etc.....  
}
```

Helps us to get the data where the quantity matches. Just like if statement in Python, Java.

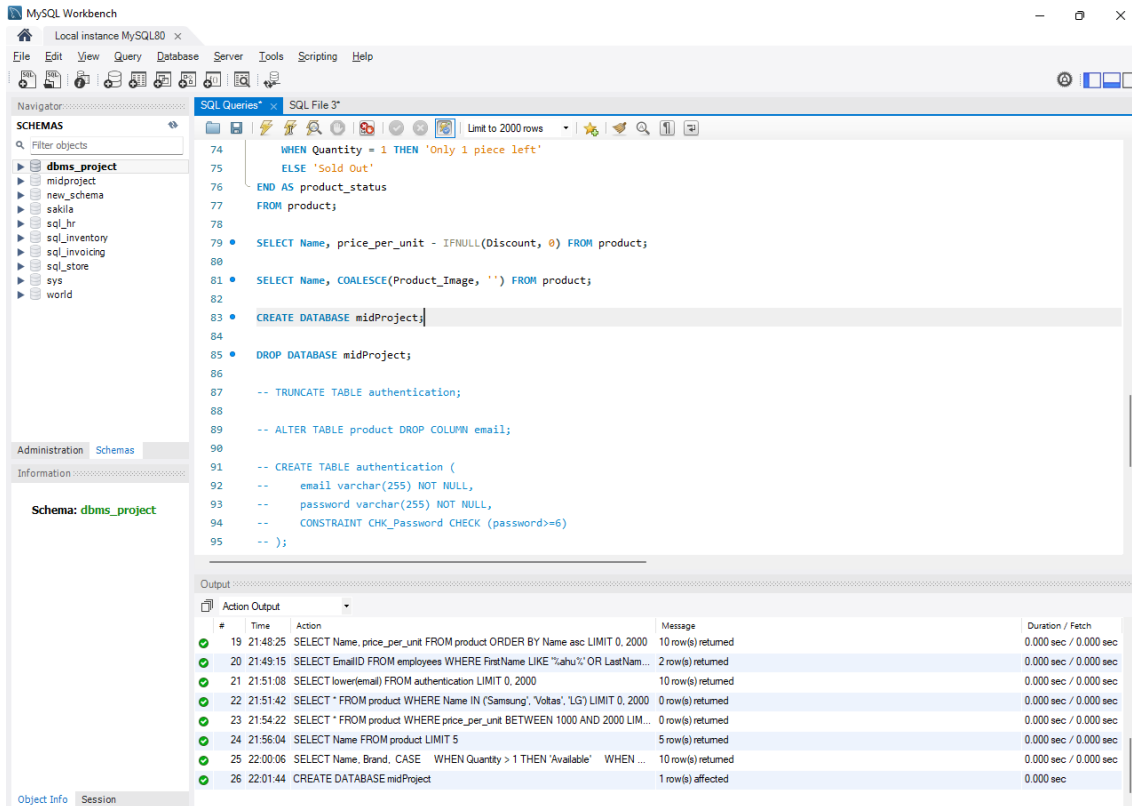
The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays the 'dbms\_project' database with various tables and views. The main editor shows a SQL query using a CASE statement to filter products based on their quantity in the 'cartItem' table. The query is as follows:

```
68  
69 • SELECT Name FROM product WHERE product_ID = ANY (SELECT ProductID FROM cartItem WHERE Quantity > 10);  
70  
71 • SELECT Name, Brand,  
72 CASE  
73 WHEN Quantity > 1 THEN 'Available'  
74 WHEN Quantity = 1 THEN 'Only 1 piece left'  
75 ELSE 'Sold Out'  
76 END AS product_status  
77 FROM product;  
78  
79 • SELECT Name, price_per_unit - IFNULL(Discount, 0) FROM product;  
80  
81 • SELECT Name, COALESCE(Product_Image, '') FROM product;  
82  
83 -- CREATE PROCEDURE FetchAllProducts AS SELECT * FROM product GO;  
84  
85 -- EXEC FetchAllProducts;  
86
```

Below the query editor, the 'Result Grid' shows the output of the query. It displays a table with three columns: Name, Brand, and product\_status. The data is as follows:

Name	Brand	product_status
Toy Car	Toyota	Available
Toy Bike	Suzuki	Available
Toy game	gameCompany	Available
Toy Bike	honda	Available
Mens Shirt	HnM	Available
Female Shirt	HnM	Available
Mens Shots	Addidas	Available
Female Shirt	HnM	Available
Mens shoes	HnM	Available
Female shoes	HnM	Available

## 11. CREATE DATABASE helps us creating a whole new database.



The screenshot displays the MySQL Workbench interface. The left sidebar shows the 'SCHEMAS' panel with a tree view of databases including 'dbms\_project', 'midproject', 'new\_schema', 'sakila', 'sql\_hr', 'sql\_inventory', 'sql\_invoicing', 'sql\_store', 'sys', and 'world'. The 'Administration' tab is selected, and the 'Schemas' section is active, showing 'Schema: dbms\_project'.

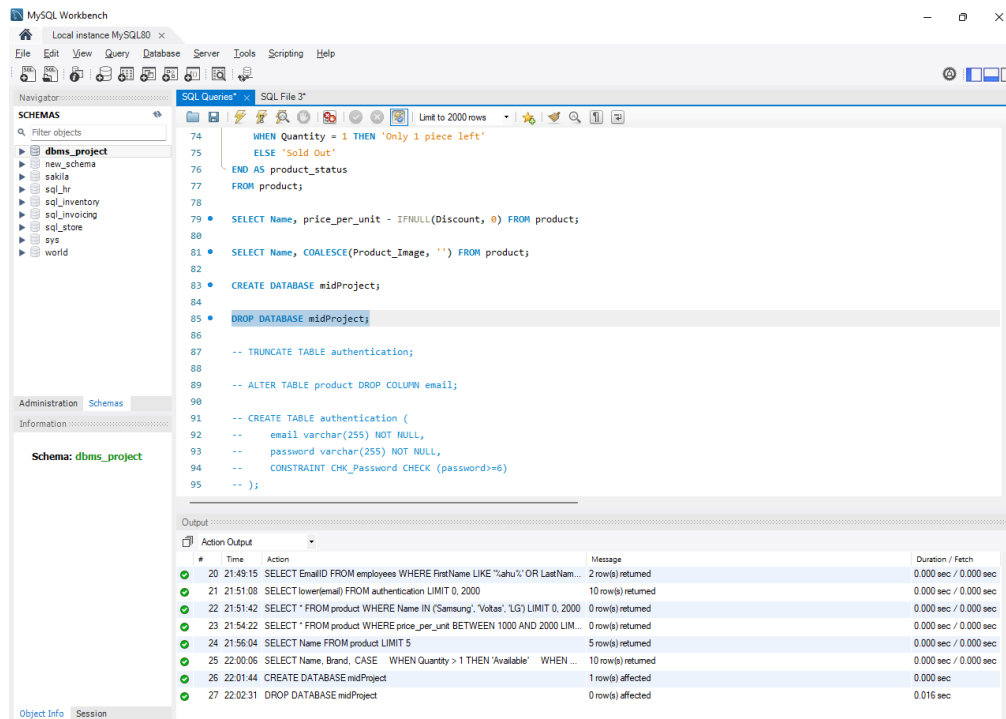
The main editor window shows a SQL script with the following queries:

```
74 WHEN Quantity = 1 THEN 'Only 1 piece left'
75 ELSE 'Sold Out'
76 END AS product_status
77 FROM product;
78
79 • SELECT Name, price_per_unit - IFNULL(Discount, 0) FROM product;
80
81 • SELECT Name, COALESCE(Product_Image, '') FROM product;
82
83 • CREATE DATABASE midProject;
84
85 • DROP DATABASE midProject;
86
87 -- TRUNCATE TABLE authentication;
88
89 -- ALTER TABLE product DROP COLUMN email;
90
91 -- CREATE TABLE authentication (
92 --   email varchar(255) NOT NULL,
93 --   password varchar(255) NOT NULL,
94 --   CONSTRAINT CHK_Password CHECK (password >= 6)
95 -- );
```

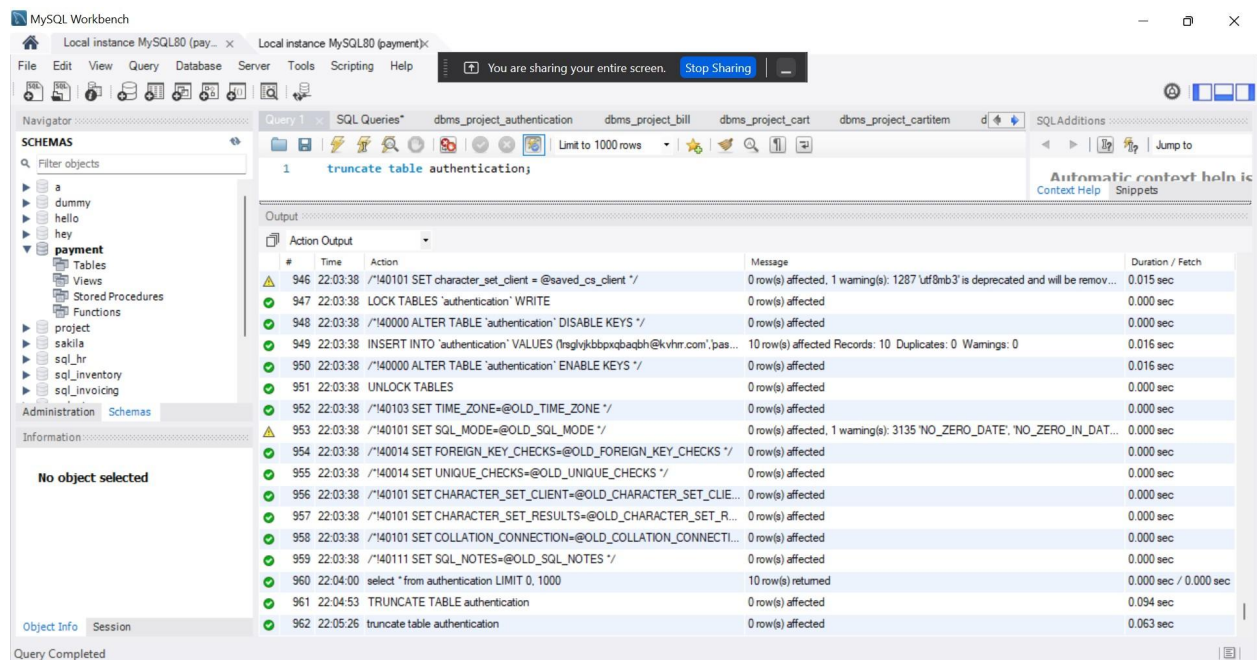
The 'Output' panel at the bottom shows the execution results of the queries. The table below summarizes the output:

#	Time	Action	Message	Duration / Fetch
19	21:48:25	SELECT Name, price_per_unit FROM product ORDER BY Name asc LIMIT 0, 2000	10 row(s) returned	0.000 sec / 0.000 sec
20	21:49:15	SELECT EmailID FROM employees WHERE FirstName LIKE 'John%'; OR LastNam...	2 row(s) returned	0.000 sec / 0.000 sec
21	21:51:08	SELECT lower(email) FROM authentication LIMIT 0, 2000	10 row(s) returned	0.000 sec / 0.000 sec
22	21:51:42	SELECT * FROM product WHERE Name IN ('Samsung', 'Volvo', 'LG') LIMIT 0, 2000	0 row(s) returned	0.000 sec / 0.000 sec
23	21:54:22	SELECT * FROM product WHERE price_per_unit BETWEEN 1000 AND 2000 LIM...	0 row(s) returned	0.000 sec / 0.000 sec
24	21:56:04	SELECT Name FROM product LIMIT 5	5 row(s) returned	0.000 sec / 0.000 sec
25	22:00:06	SELECT Name, Brand, CASE WHEN Quantity > 1 THEN 'Available' WHEN ...	10 row(s) returned	0.000 sec / 0.000 sec
26	22:01:44	CREATE DATABASE midProject	1 row(s) affected	0.000 sec

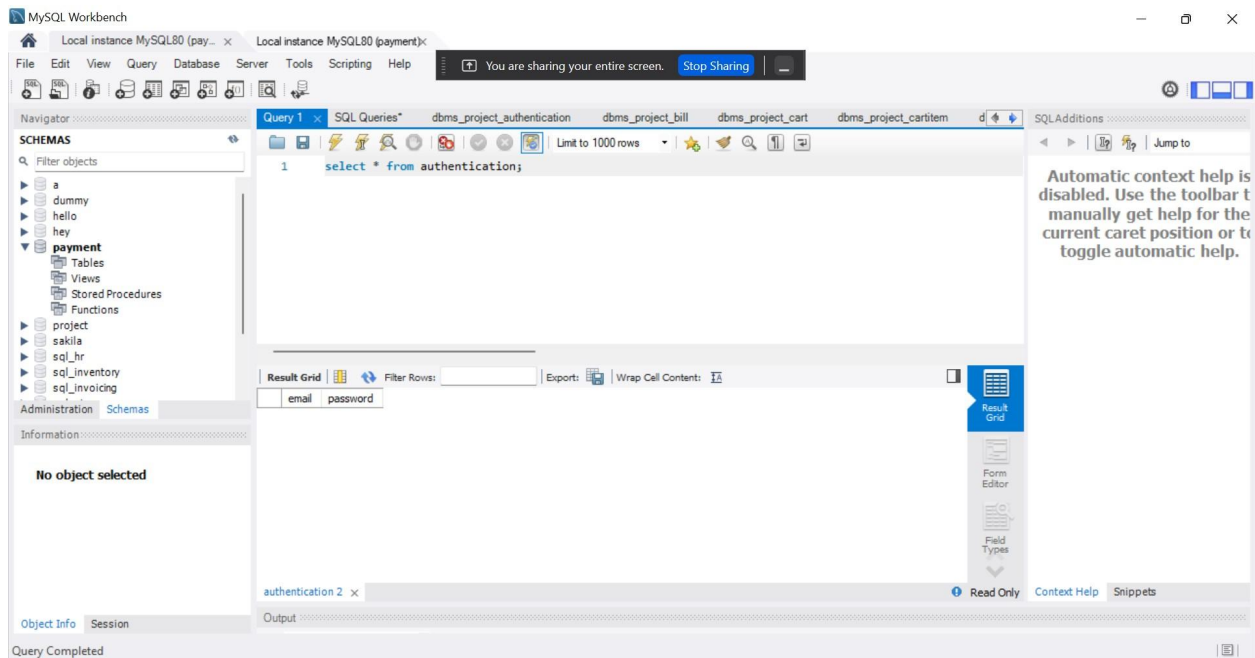
## 12. DROP DATABASE helps us delete the database.



## 13. TRUNCATE TABLE helps us to delete the data of that particular table.



After TRUNCATE :



14. ALTER TABLE \_\_\_\_\_ DROP COLUMN \_\_\_\_\_ helps us to delete that particular column from that particular table.

The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying the following SQL query:

```
ALTER TABLE product DROP COLUMN description;
```

The 'Output' pane shows the execution results. The table below represents the data shown in the 'Action Output' pane:

#	Time	Action	Message	Duration / Fetch
954	22:03:38	/140014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS ~/	0 row(s) affected	0.000 sec
955	22:03:38	/140014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS ~/	0 row(s) affected	0.000 sec
956	22:03:38	/140101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT	0 row(s) affected	0.000 sec
957	22:03:38	/140101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS	0 row(s) affected	0.000 sec
958	22:03:38	/140101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION	0 row(s) affected	0.000 sec
959	22:03:38	/140111 SET SQL_NOTES=@OLD_SQL_NOTES ~/	0 row(s) affected	0.000 sec
960	22:04:00	select * from authentication LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
961	22:04:53	TRUNCATE TABLE authentication	0 row(s) affected	0.094 sec
962	22:05:26	truncate table authentication	0 row(s) affected	0.063 sec
963	22:06:35	select * from authentication LIMIT 0, 1000	select * from authentication LIMIT 0, 1000	0.000 sec / 0.000 sec
964	22:07:34	SELECT e.FirstName + " " + e.LastName As "Full Name" FROM employee	0 row(s) affected	0.000 sec / 0.000 sec
965	22:08:04	ALTER TABLE product DROP COLUMN email	Error Code: 1091. Can't DROP 'email'; check that column/key exists	0.016 sec
966	22:08:34	ALTER TABLE product DROP COLUMN description	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.250 sec

The error message for query 965 is: "Error Code: 1091. Can't DROP 'email'; check that column/key exists".

After ALTER TABLE:

The screenshot shows the MySQL Workbench interface. The 'Query 1' window displays the SQL query: `select * from product;`. The 'Result Grid' shows the output of this query, which is a table with 10 columns: `product_ID`, `Name`, `brand`, `category`, `price_per_unit`, `Seller_ID`, `Quantity`, `Product_Image`, and `Discount`. The data is as follows:

product_ID	Name	brand	category	price_per_unit	Seller_ID	Quantity	Product_Image	Discount
1	Toy Car	Toyota	Toys	100	1	10	Toy_Car.jpg	10
2	Toy Bike	Suzuki	Toys	120	2	10	Toy_Bike.jpg	14
3	Toy game	gameCompany	Toys	150	3	10	Toy_game.jpg	10
4	Toy Bike	honda	Toys	160	4	10	Toy_Bike.jpg	20
5	Mens Shirt	HrM	Clothes	200	5	10	Mens_Shirt.jpg	15
6	Female Shirt	HrM	Clothes	220	6	10	Female_Shirt.jpg	18
7	Mens Shorts	Adidas	Clothes	230	7	10	Mens_Shorts.jpg	20
8	Female Shirt	HrM	Clothes	270	250	10	Female_Shirt.jpg	16
9	Mens shoes	HrM	Footwear	300	9	10	Mens_shoes.jpg	21
10	Female shoes	HrM	Footwear	320	10	10	Female_shoes.jpg	20

The 'Schemas' panel on the left shows the database structure, including the 'payment' database and the 'product' table. The 'Output' panel at the bottom shows the query execution status: 'Query Completed'.

## 15. CREATE TABLE helps us create a new table with the given fields and values.

The screenshot displays the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows a tree view of the 'dbms\_project' database, including tables like 'authentication', 'authentication2', 'bill', 'cart', 'cartitem', 'coupon', 'customer', 'customers', 'employees', 'order\_item', 'orders', 'product', 'reviews', 'seller', and 'sellers'. The main editor window shows a series of SQL queries. The 'Output' pane at the bottom displays the execution results of these queries.

```
83 • CREATE DATABASE midProject;
84
85 • DROP DATABASE midProject;
86
87 -- TRUNCATE TABLE authentication;
88
89 -- ALTER TABLE product DROP COLUMN email;
90
91 • CREATE TABLE authentication2 (
92     email varchar(255) NOT NULL,
93     password varchar(255) NOT NULL,
94     CONSTRAINT CHK_Password CHECK (password >= 6)
95 );
96
97 • INSERT INTO reviews (ProductID) SELECT product_ID FROM product;
98
99 • CREATE VIEW Mobile AS SELECT * FROM product WHERE category = 'mobile';
100
101 • CREATE OR REPLACE VIEW Delhiite AS SELECT FirstName, LastName, State FROM customers WHERE State = 'delhi';
102
103 • SELECT * FROM Delhiite;
104
```

The 'Output' pane shows the following results:

#	Time	Action	Message	Duration / Fetch
23	21:54:22	SELECT * FROM product WHERE price_per_unit BETWEEN 1000 AND 2000 LIM...	0 row(s) returned	0.000 sec / 0.000 sec
24	21:56:04	SELECT Name FROM product LIMIT 5	5 row(s) returned	0.000 sec / 0.000 sec
25	22:00:06	SELECT Name, Brand, CASE WHEN Quantity > 1 THEN 'Available' WHEN ...	10 row(s) returned	0.000 sec / 0.000 sec
26	22:01:44	CREATE DATABASE midProject	1 row(s) affected	0.000 sec
27	22:02:31	DROP DATABASE midProject	0 row(s) affected	0.016 sec
28	22:02:48	DROP DATABASE midProject	Error Code: 1008. Can't drop database 'midproject'; database doesn't exist	0.000 sec
29	22:08:24	desc product	10 row(s) returned	0.000 sec / 0.000 sec
30	22:09:35	CREATE TABLE authentication2 ( email varchar(255) NOT NULL, password v...	0 row(s) affected	0.046 sec



## 17. DROP TABLE helps us to delete the whole table.

The screenshot displays the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows the 'dbms\_project' database with various tables and views. The main editor shows a series of SQL queries. The 'Output' pane at the bottom shows the execution results of these queries.

```
83 • CREATE DATABASE midProject;
84
85 • DROP DATABASE midProject;
86
87 -- TRUNCATE TABLE authentication;
88
89 -- ALTER TABLE product DROP COLUMN email;
90
91 -- CREATE TABLE authentication2 (
92 --     email varchar(255) NOT NULL,
93 --     password varchar(255) NOT NULL,
94 --     CONSTRAINT CHK_Password CHECK (password >= 6)
95 -- );
96
97 • DROP TABLE authentication2;
98
99 • INSERT INTO reviews (ProductID) SELECT product_ID FROM product;
100
101 • CREATE VIEW Mobile AS SELECT * FROM product WHERE category = 'mobile';
102
103 • CREATE OR REPLACE VIEW Delhiite AS SELECT FirstName, LastName, State FROM customers WHERE State = 'delhi';
104
```

The 'Output' pane shows the following results:

#	Time	Action	Message	Duration / Fetch
24	21:56:04	SELECT Name FROM product LIMIT 5	5 row(s) returned	0.000 sec / 0.000 sec
25	22:00:06	SELECT Name, Brand, CASE WHEN Quantity > 1 THEN 'Available' WHEN ...	10 row(s) returned	0.000 sec / 0.000 sec
26	22:01:44	CREATE DATABASE midProject	1 row(s) affected	0.000 sec
27	22:02:31	DROP DATABASE midProject	0 row(s) affected	0.016 sec
28	22:02:48	DROP DATABASE midProject	Error Code: 1008. Can't drop database 'midproject'; database doesn't exist	0.000 sec
29	22:08:24	desc product	10 row(s) returned	0.000 sec / 0.000 sec
30	22:09:35	CREATE TABLE authentication2 ( email varchar(255) NOT NULL, password v...	0 row(s) affected	0.046 sec
31	22:10:18	DROP TABLE authentication2	0 row(s) affected	0.015 sec

18. CREATE VIEW helps us creating a new view with the name we have provided.

The screenshot displays the MySQL Workbench interface. On the left, the 'Navigator' pane shows the 'dbms\_project' schema with various tables and views. The 'Views' section is expanded, showing 'mobile' and 'Delhiite'. The main editor pane contains the following SQL script:

```
89 -- ALTER TABLE product DROP COLUMN email;
90
91 -- CREATE TABLE authentication2 (
92 --     email varchar(255) NOT NULL,
93 --     password varchar(255) NOT NULL,
94 --     CONSTRAINT CHK_Password CHECK (password=6)
95 -- );
96
97 DROP TABLE authentication2;
98
99 INSERT INTO reviews (ProductID) SELECT product_ID FROM product;
100
101 CREATE VIEW Mobile AS SELECT * FROM product WHERE category = 'mobile';
102
103 CREATE OR REPLACE VIEW Delhiite AS SELECT FirstName, LastName, State FROM customers WHERE State = 'delhi';
104
105 SELECT * FROM Delhiite;
106
107 DROP VIEW Delhiite;
108
109 DROP VIEW mobile;
110
```

The 'Output' pane at the bottom shows the execution results of the queries:

#	Time	Action	Message	Duration / Fetch
29	22:08:24	desc product	10 row(s) returned	0.000 sec / 0.000 sec
30	22:09:35	CREATE TABLE authentication2 ( email varchar(255) NOT NULL, password v...	0 row(s) affected	0.046 sec
31	22:10:18	DROP TABLE authentication2	0 row(s) affected	0.015 sec
32	22:10:42	INSERT INTO reviews (ProductID) SELECT product_ID FROM product	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.016 sec
33	22:10:43	INSERT INTO reviews (ProductID) SELECT product_ID FROM product	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
34	22:10:43	INSERT INTO reviews (ProductID) SELECT product_ID FROM product	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
35	22:10:43	INSERT INTO reviews (ProductID) SELECT product_ID FROM product	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
36	22:10:43	INSERT INTO reviews (ProductID) SELECT product_ID FROM product	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.015 sec

19. CREATE OR REPLACE view helps us creating if not there and if there then replace with the given information provided.

The screenshot displays the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows a tree view for the 'dbms\_project' database, including tables like authentication, bill, cart, cartitem, coupon, customer, customers, employees, order\_item, orders, product, reviews, seller, and sellers, as well as views like mobile. The main editor window shows a SQL script with the following queries:

```
89 -- ALTER TABLE product DROP COLUMN email;
90
91 -- CREATE TABLE authentication2 (
92 --     email varchar(255) NOT NULL,
93 --     password varchar(255) NOT NULL,
94 --     CONSTRAINT CHK_Password CHECK (password >= 6)
95 -- );
96
97 DROP TABLE authentication2;
98
99 INSERT INTO reviews (ProductID) SELECT product_ID FROM product;
100
101 CREATE VIEW Mobile AS SELECT * FROM product WHERE category = 'mobile';
102
103 CREATE OR REPLACE VIEW Delhiite AS SELECT FirstName, LastName, State FROM customers WHERE State = 'delhi';
104
105 SELECT * FROM Delhiite;
106
107 DROP VIEW Delhiite;
108
109 DROP VIEW mobile;
110
```

The 'Output' pane at the bottom shows the execution results of these queries:

#	Time	Action	Message	Duration / Fetch
30	22:09:35	CREATE TABLE authentication2 ( email varchar(255) NOT NULL, password v...	0 row(s) affected	0.046 sec
31	22:10:18	DROP TABLE authentication2	0 row(s) affected	0.015 sec
32	22:10:42	INSERT INTO reviews (ProductID) SELECT product_ID FROM product	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.016 sec
33	22:10:43	INSERT INTO reviews (ProductID) SELECT product_ID FROM product	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
34	22:10:43	INSERT INTO reviews (ProductID) SELECT product_ID FROM product	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
35	22:10:43	INSERT INTO reviews (ProductID) SELECT product_ID FROM product	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
36	22:10:43	INSERT INTO reviews (ProductID) SELECT product_ID FROM product	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.015 sec
37	22:11:50	CREATE OR REPLACE VIEW Delhiite AS SELECT FirstName, LastName, State F...	0 row(s) affected	0.000 sec

## 20. DROP view helps us delete that view.

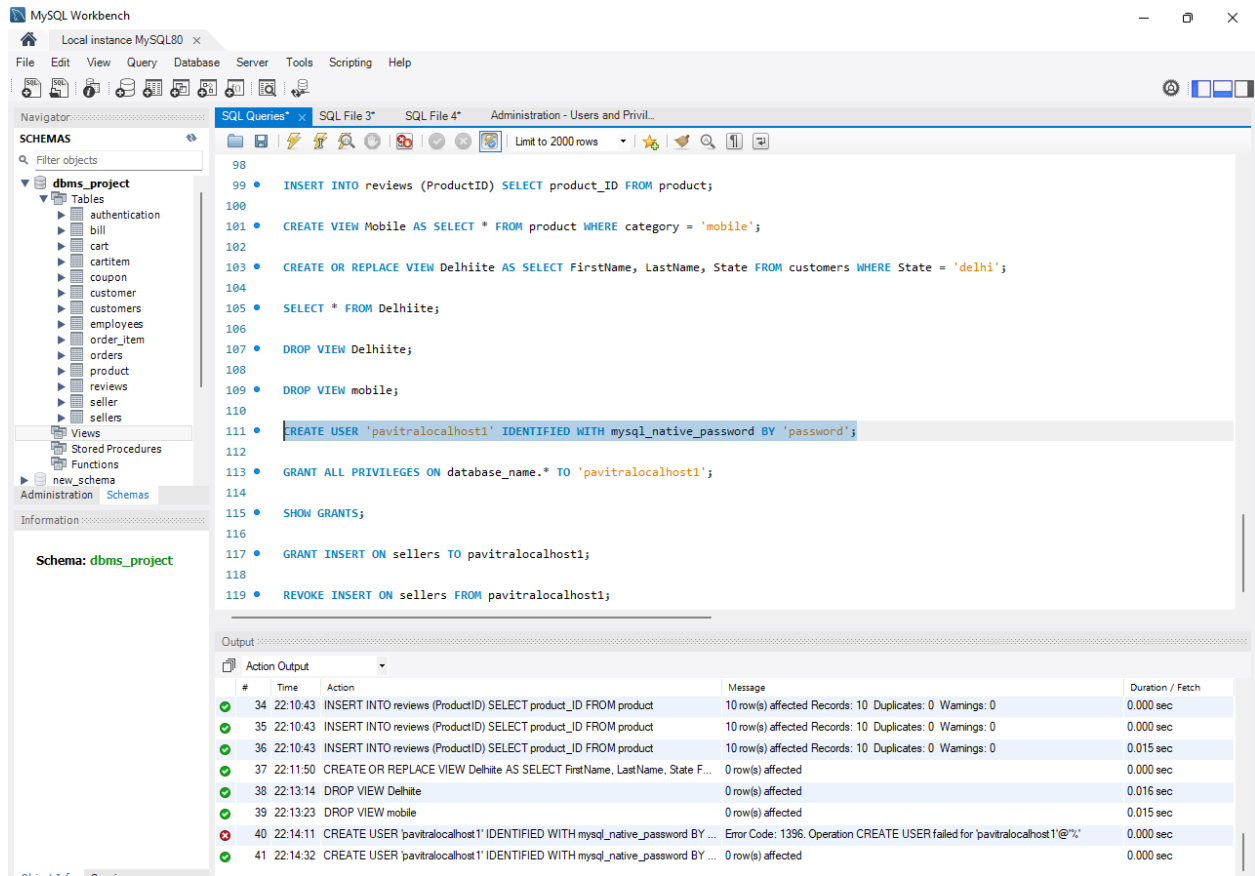
The screenshot shows the MySQL Workbench interface with a SQL query editor and an output window. The query editor contains the following SQL code:

```
89 -- ALTER TABLE product DROP COLUMN email;
90
91 -- CREATE TABLE authentication2 (
92 --     email varchar(255) NOT NULL,
93 --     password varchar(255) NOT NULL,
94 --     CONSTRAINT CHK_Password CHECK (password>=6)
95 -- );
96
97 • DROP TABLE authentication2;
98
99 • INSERT INTO reviews (ProductID) SELECT product_ID FROM product;
100
101 • CREATE VIEW Mobile AS SELECT * FROM product WHERE category = 'mobile';
102
103 • CREATE OR REPLACE VIEW Delhiite AS SELECT FirstName, LastName, State FROM customers WHERE State = 'delhi';
104
105 • SELECT * FROM Delhiite;
106
107 • DROP VIEW Delhiite;
108
109 • DROP VIEW mobile;
110
```

The output window shows the execution results of these queries:

#	Time	Action	Message	Duration / Fetch
✓ 32	22:10:42	INSERT INTO reviews (ProductID) SELECT product_ID FROM product	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.016 sec
✓ 33	22:10:43	INSERT INTO reviews (ProductID) SELECT product_ID FROM product	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
✓ 34	22:10:43	INSERT INTO reviews (ProductID) SELECT product_ID FROM product	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
✓ 35	22:10:43	INSERT INTO reviews (ProductID) SELECT product_ID FROM product	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
✓ 36	22:10:43	INSERT INTO reviews (ProductID) SELECT product_ID FROM product	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.015 sec
✓ 37	22:11:50	CREATE OR REPLACE VIEW Delhiite AS SELECT FirstName, LastName, State F...	0 row(s) affected	0.000 sec
✓ 38	22:13:14	DROP VIEW Delhiite	0 row(s) affected	0.016 sec
✓ 39	22:13:23	DROP VIEW mobile	0 row(s) affected	0.015 sec

21. CREATE USER \_\_\_\_\_ helps us create a new user.



The screenshot displays the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows a tree view for the 'dbms\_project' database, including tables like authentication, bill, cart, cartitem, coupon, customer, customers, employees, order\_item, orders, product, reviews, seller, and sellers. The 'Information' pane at the bottom left indicates the current schema is 'dbms\_project'.

The main editor shows a series of SQL queries. The query at line 111, `CREATE USER 'pavitrlocalhost1' IDENTIFIED WITH mysql_native_password BY 'password';`, is highlighted. Below the editor, the 'Output' pane shows the execution results of the queries. The results table has columns for line number, time, action, message, and duration. The execution of the `CREATE USER` query at line 41 failed with error code 1396.

#	Time	Action	Message	Duration / Fetch
34	22:10:43	INSERT INTO reviews (ProductID) SELECT product_ID FROM product	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
35	22:10:43	INSERT INTO reviews (ProductID) SELECT product_ID FROM product	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
36	22:10:43	INSERT INTO reviews (ProductID) SELECT product_ID FROM product	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.015 sec
37	22:11:50	CREATE OR REPLACE VIEW Delhiite AS SELECT FirstName, LastName, State F...	0 row(s) affected	0.000 sec
38	22:13:14	DROP VIEW Delhiite	0 row(s) affected	0.016 sec
39	22:13:23	DROP VIEW mobile	0 row(s) affected	0.015 sec
40	22:14:11	CREATE USER pavitrlocalhost1 IDENTIFIED WITH mysql_native_password BY ...	Error Code: 1396. Operation CREATE USER failed for 'pavitrlocalhost1'@'%'	0.000 sec
41	22:14:32	CREATE USER pavitrlocalhost1 IDENTIFIED WITH mysql_native_password BY ...	0 row(s) affected	0.000 sec

## 22. GRANT ALL PRIVILEGES helps us set and give all the privileges to the user created.

The screenshot shows the MySQL Workbench interface with the 'Administration - Users and Privileges' tab selected. The SQL editor contains the following queries:

```
98
99 • INSERT INTO reviews (ProductID) SELECT product_ID FROM product;
100
101 • CREATE VIEW Mobile AS SELECT * FROM product WHERE category = 'mobile';
102
103 • CREATE OR REPLACE VIEW Delhiite AS SELECT FirstName, LastName, State FROM customers WHERE State = 'delhi';
104
105 • SELECT * FROM Delhiite;
106
107 • DROP VIEW Delhiite;
108
109 • DROP VIEW mobile;
110
111 • CREATE USER 'pavitrlocalhost1' IDENTIFIED WITH mysql_native_password BY 'password';
112
113 • GRANT ALL PRIVILEGES ON database_name.* TO 'pavitrlocalhost1';
114
115 • SHOW GRANTS;
116
117 • GRANT INSERT ON sellers TO pavitrlocalhost1;
118
119 • REVOKE INSERT ON sellers FROM pavitrlocalhost1;
```

The Output pane shows the execution results:

#	Time	Action	Message	Duration / Fetch
35	22:10:43	INSERT INTO reviews (ProductID) SELECT product_ID FROM product	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
36	22:10:43	INSERT INTO reviews (ProductID) SELECT product_ID FROM product	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.015 sec
37	22:11:50	CREATE OR REPLACE VIEW Delhiite AS SELECT FirstName, LastName, State F...	0 row(s) affected	0.000 sec
38	22:13:14	DROP VIEW Delhiite	0 row(s) affected	0.016 sec
39	22:13:23	DROP VIEW mobile	0 row(s) affected	0.015 sec
40	22:14:11	CREATE USER 'pavitrlocalhost1' IDENTIFIED WITH mysql_native_password BY ...	Error Code: 1396. Operation CREATE USER failed for 'pavitrlocalhost1'@'%'	0.000 sec
41	22:14:32	CREATE USER 'pavitrlocalhost1' IDENTIFIED WITH mysql_native_password BY ...	0 row(s) affected	0.000 sec
42	22:15:08	GRANT ALL PRIVILEGES ON database_name.* TO 'pavitrlocalhost1'	0 row(s) affected	0.016 sec

The screenshot shows the 'Users and Privileges' window in MySQL Workbench. The 'User Accounts' tab is selected, showing a list of users:

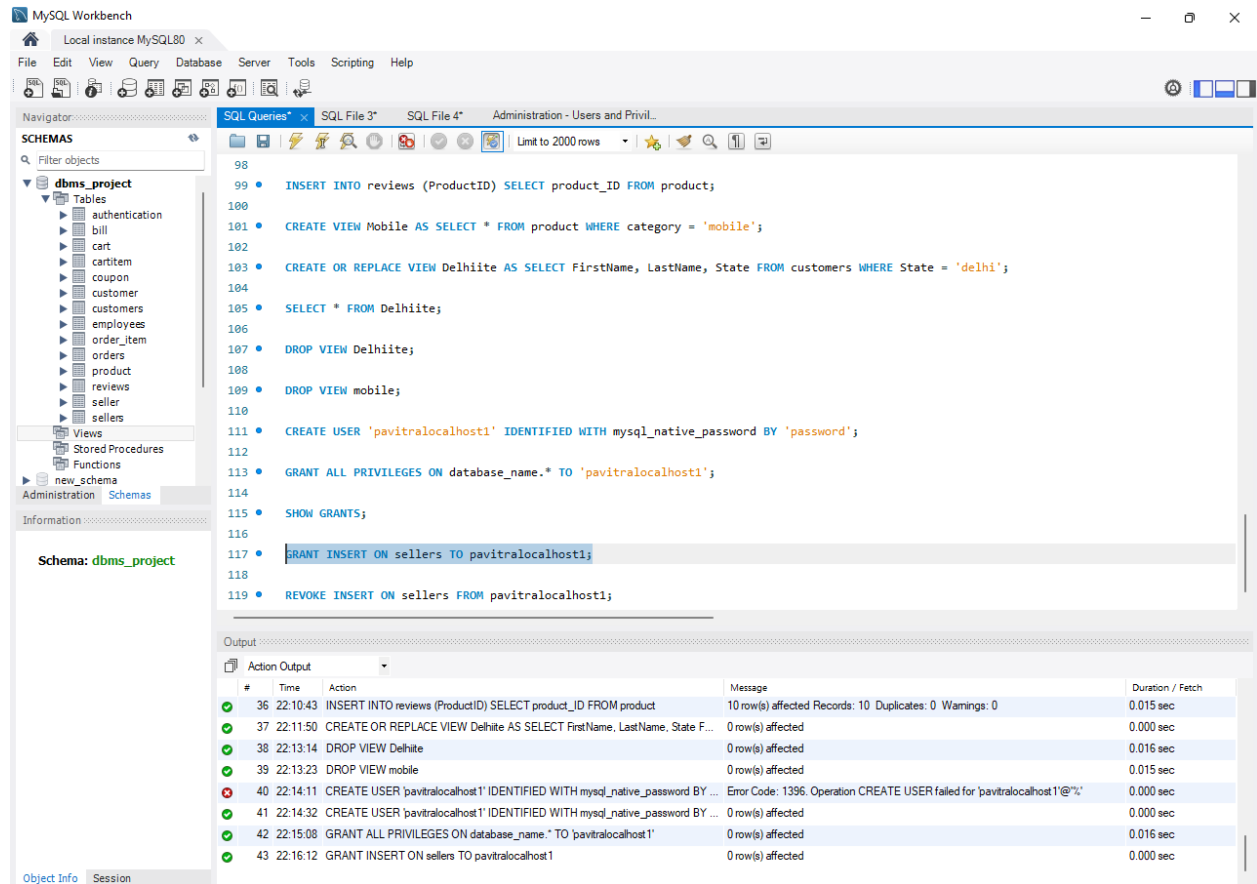
User	From Host
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
pavitrlocalhost1	%
root	%
root	localhost
root@%	%

The 'Details for account pavitrlocalhost1@%' window is open, showing the 'Global Privileges' tab. The privileges are listed as follows:

Privilege	Description
<input checked="" type="checkbox"/> ALTER	grants the rights to perform all tasks
<input checked="" type="checkbox"/> ALTER ROUTINE	grants rights needed to maintain server
<input checked="" type="checkbox"/> CREATE	rights needed to assess, monitor, and kill any user proce...
<input checked="" type="checkbox"/> CREATE ROUTINE	grants rights to create users logins and reset passwords
<input checked="" type="checkbox"/> CREATE TABLESPACE	rights to manage logins and grant and revoke server an...
<input checked="" type="checkbox"/> CREATE TEMPORARY TABLES	minimum set of rights needed to monitor server
<input checked="" type="checkbox"/> CREATE USER	grants full rights on all databases
<input checked="" type="checkbox"/> CREATE VIEW	rights to create and reverse engineer any database sche...
<input checked="" type="checkbox"/> DELETE	rights needed to setup and manage replication
<input checked="" type="checkbox"/> DROP	minimal rights needed to backup any database
<input checked="" type="checkbox"/> EVENT	
<input checked="" type="checkbox"/> EXECUTE	
<input checked="" type="checkbox"/> FILE	
<input checked="" type="checkbox"/> GRANT OPTION	
<input checked="" type="checkbox"/> INDEX	
<input checked="" type="checkbox"/> INSERT	
<input checked="" type="checkbox"/> LOCK TABLES	
<input checked="" type="checkbox"/> PROCESS	

Buttons at the bottom include 'Add Account', 'Delete', 'Refresh', 'Revoke All Privileges', 'Revert', and 'Apply'.

23. GRANT INSERT ON helps us to give the inserting power to the user created.



The screenshot displays the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows the 'dbms\_project' database with various tables and views. The main editor window contains a series of SQL queries. The 'Output' pane at the bottom shows the execution results of these queries.

**SQL Queries:**

```
98 • INSERT INTO reviews (ProductID) SELECT product_ID FROM product;
99 •
100 •
101 • CREATE VIEW Mobile AS SELECT * FROM product WHERE category = 'mobile';
102 •
103 • CREATE OR REPLACE VIEW Delhiite AS SELECT FirstName, LastName, State FROM customers WHERE State = 'delhi';
104 •
105 • SELECT * FROM Delhiite;
106 •
107 • DROP VIEW Delhiite;
108 •
109 • DROP VIEW mobile;
110 •
111 • CREATE USER 'pavitrlocalhost1' IDENTIFIED WITH mysql_native_password BY 'password';
112 •
113 • GRANT ALL PRIVILEGES ON database_name.* TO 'pavitrlocalhost1';
114 •
115 • SHOW GRANTS;
116 •
117 • GRANT INSERT ON sellers TO pavitrlocalhost1;
118 •
119 • REVOKE INSERT ON sellers FROM pavitrlocalhost1;
```

**Output:**

#	Time	Action	Message	Duration / Fetch
36	22:10:43	INSERT INTO reviews (ProductID) SELECT product_ID FROM product	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.015 sec
37	22:11:50	CREATE OR REPLACE VIEW Delhiite AS SELECT FirstName, LastName, State F...	0 row(s) affected	0.000 sec
38	22:13:14	DROP VIEW Delhiite	0 row(s) affected	0.016 sec
39	22:13:23	DROP VIEW mobile	0 row(s) affected	0.015 sec
40	22:14:11	CREATE USER 'pavitrlocalhost1' IDENTIFIED WITH mysql_native_password BY ...	Error Code: 1396. Operation CREATE USER failed for 'pavitrlocalhost1'@'%'	0.000 sec
41	22:14:32	CREATE USER 'pavitrlocalhost1' IDENTIFIED WITH mysql_native_password BY ...	0 row(s) affected	0.000 sec
42	22:15:08	GRANT ALL PRIVILEGES ON database_name.* TO 'pavitrlocalhost1'	0 row(s) affected	0.016 sec
43	22:16:12	GRANT INSERT ON sellers TO pavitrlocalhost1	0 row(s) affected	0.000 sec

24. SHOW GRANTS helps us to see the grants given to the users.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'dbms\_project' schema with various tables and views. The main editor window contains the following SQL queries:

```
108
109 • DROP VIEW Delhite;
110
111 • CREATE USER 'pavitrlocalhost1' IDENTIFIED WITH mysql_native_password BY 'password';
112
113 • GRANT ALL PRIVILEGES ON database_name.* TO 'pavitrlocalhost1';
114
115 • SHOW GRANTS;
```

The 'Result Grid' shows the output of the 'SHOW GRANTS;' query:

Grants for root@localhost
GRANT SELECT, INSERT, UPDATE, DELETE, CR...
GRANT APPLICATION_PASSWORD_ADMIN, AU...
GRANT PROXY ON ''@'' TO 'root'@'localho...

The 'Output' window shows the 'Action Output' for the executed queries:

#	Time	Action	Message	Duration / Fetch
37	22:11:50	CREATE OR REPLACE VIEW Delhite AS SELECT FirstName, LastName, State F...	0 row(s) affected	0.000 sec
38	22:13:14	DROP VIEW Delhite	0 row(s) affected	0.016 sec
39	22:13:23	DROP VIEW Delhite	0 row(s) affected	0.015 sec
40	22:14:11	CREATE USER 'pavitrlocalhost1' IDENTIFIED WITH mysql_native_password BY ...	Error Code: 1396. Operation CREATE USER failed for 'pavitrlocalhost1'@'%'	0.000 sec
41	22:14:32	CREATE USER 'pavitrlocalhost1' IDENTIFIED WITH mysql_native_password BY ...	0 row(s) affected	0.000 sec
42	22:15:08	GRANT ALL PRIVILEGES ON database_name.* TO 'pavitrlocalhost1'	0 row(s) affected	0.016 sec
43	22:16:12	GRANT INSERT ON sellers TO pavitrlocalhost1	0 row(s) affected	0.000 sec
44	22:17:43	SHOW GRANTS	3 row(s) returned	0.000 sec / 0.000 sec



25. REVOKE INSERT ON \_\_\_\_\_ helps us to revoke the insert access from the user.

The screenshot displays the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows the 'dbms\_project' schema with various tables and views. The main editor window contains a series of SQL queries, with the final query being 'REVOKE INSERT ON sellers FROM pavitrlocalhost1;'. Below the editor, the 'Output' pane shows the execution results of these queries.

SQL Queries:

```
98
99 • INSERT INTO reviews (ProductID) SELECT product_ID FROM product;
100
101 • CREATE VIEW Mobile AS SELECT * FROM product WHERE category = 'mobile';
102
103 • CREATE OR REPLACE VIEW Delhiite AS SELECT FirstName, LastName, State FROM customers WHERE State = 'delhi';
104
105 • SELECT * FROM Delhiite;
106
107 • DROP VIEW Delhiite;
108
109 • DROP VIEW mobile;
110
111 • CREATE USER 'pavitrlocalhost1' IDENTIFIED WITH mysql_native_password BY 'password';
112
113 • GRANT ALL PRIVILEGES ON database_name.* TO 'pavitrlocalhost1';
114
115 • SHOW GRANTS;
116
117 • GRANT INSERT ON sellers TO pavitrlocalhost1;
118
119 • REVOKE INSERT ON sellers FROM pavitrlocalhost1;
```

Output:

#	Time	Action	Message	Duration / Fetch
✓ 38	22:13:14	DROP VIEW Delhiite	0 row(s) affected	0.016 sec
✓ 39	22:13:23	DROP VIEW mobile	0 row(s) affected	0.015 sec
✗ 40	22:14:11	CREATE USER 'pavitrlocalhost1' IDENTIFIED WITH mysql_native_password BY ...	Error Code: 1396. Operation CREATE USER failed for 'pavitrlocalhost1'@'%'	0.000 sec
✓ 41	22:14:32	CREATE USER 'pavitrlocalhost1' IDENTIFIED WITH mysql_native_password BY ...	0 row(s) affected	0.000 sec
✓ 42	22:15:08	GRANT ALL PRIVILEGES ON database_name.* TO 'pavitrlocalhost1'	0 row(s) affected	0.016 sec
✓ 43	22:16:12	GRANT INSERT ON sellers TO pavitrlocalhost1	0 row(s) affected	0.000 sec
✓ 44	22:17:43	SHOW GRANTS	3 row(s) returned	0.000 sec / 0.000 sec
✓ 45	22:18:22	REVOKE INSERT ON sellers FROM pavitrlocalhost1	0 row(s) affected	0.016 sec