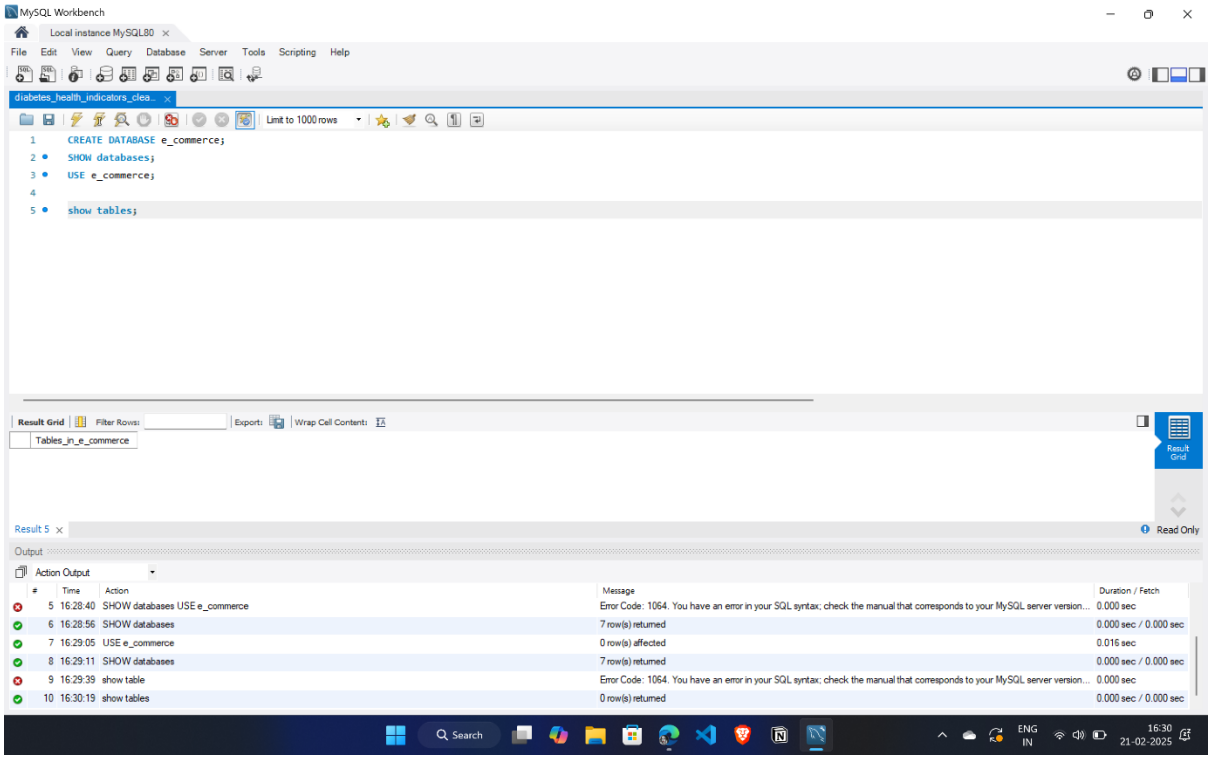
Basic SQL Training Assignment

Github: SQL assignment

Create Database e_commerce:



Create following Tables:

Customers:

a. customer_id - int auto-increment primary key

b. name - varchar(50)

c. email - varchar(50)

d. mobile - varchar(15)

Products:

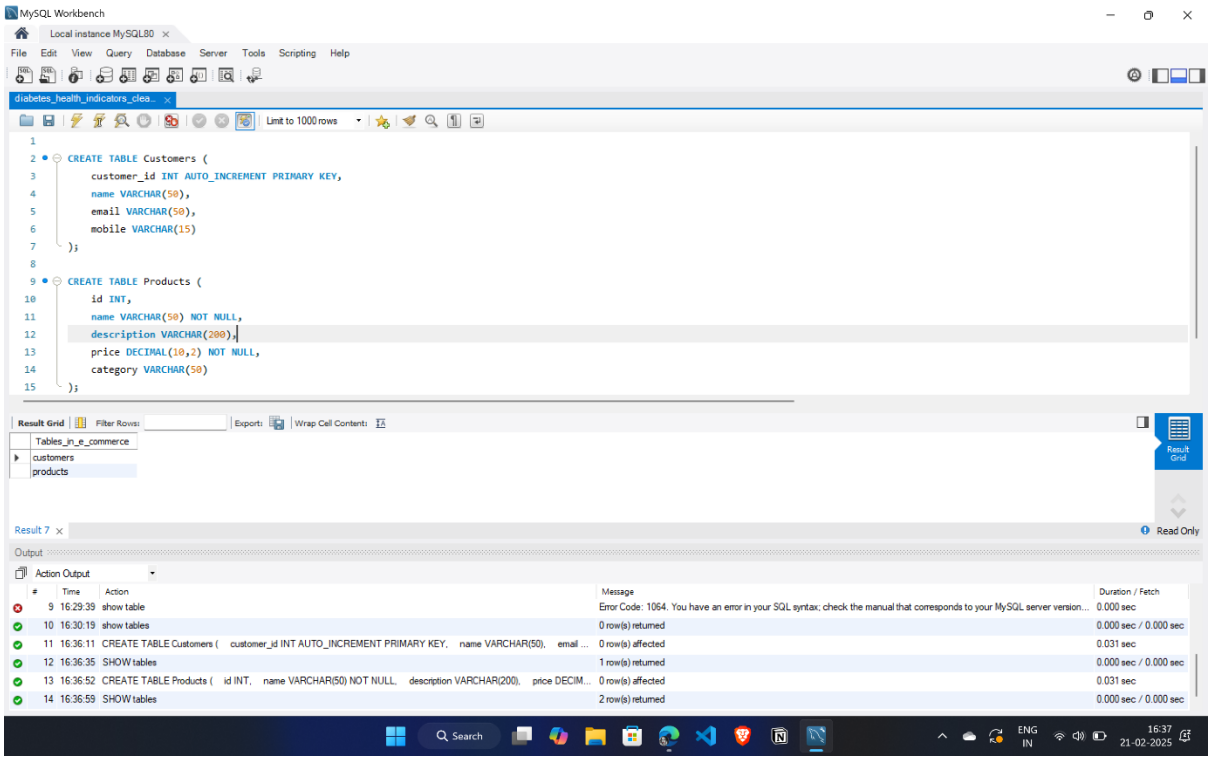
a. id - int

b. name - varchar(50) not null

c. description - varchar(200)

d. price - decimal(10, 2) not null

e. category - varchar(50)



Modify Tables:

a. Add not null on name and email in the Customers table

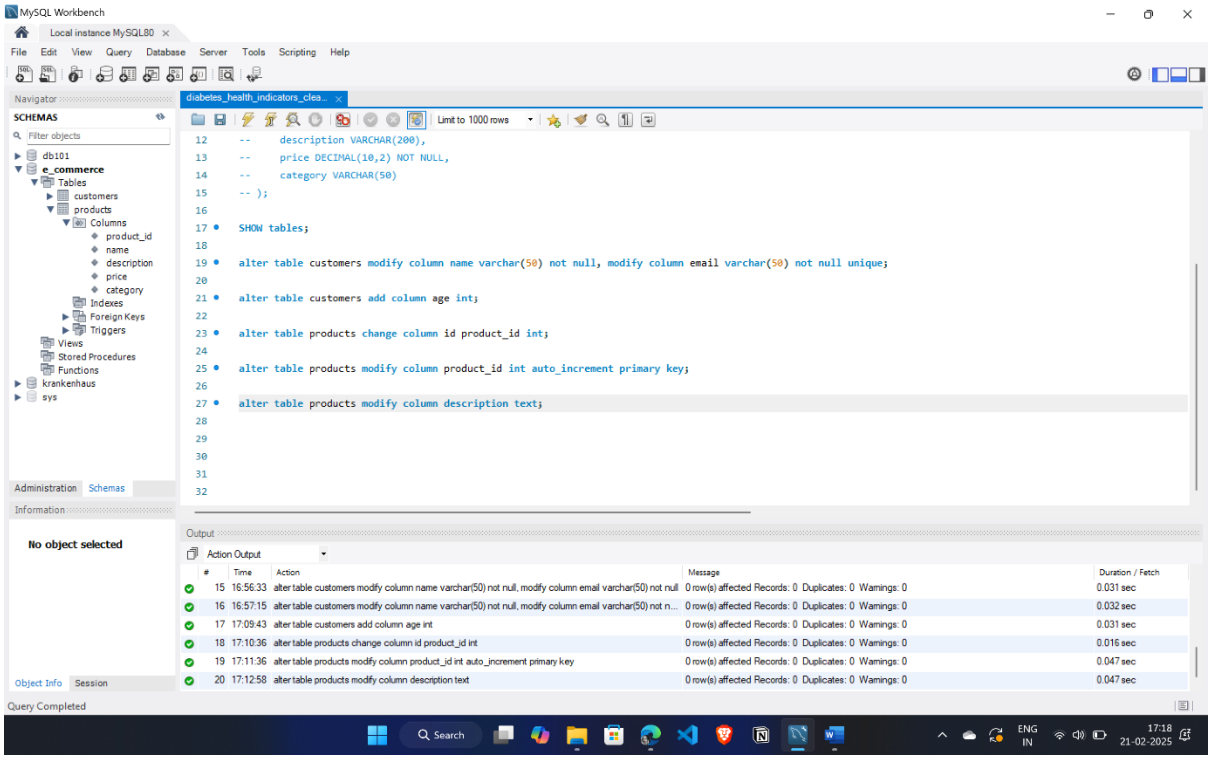
b. Add unique key on email in the Customers table

c. Add column age in the Customers table

d. Change column name from id to product_id in the Products table;

e. Add primary key and auto increment on product_id in the Products table

f. Change datatype of description from varchar to text in the Products table



Create table Order:

a. order_id - int auto-increment primary key

b. customer_id - int -foreign key

c. product_id - int

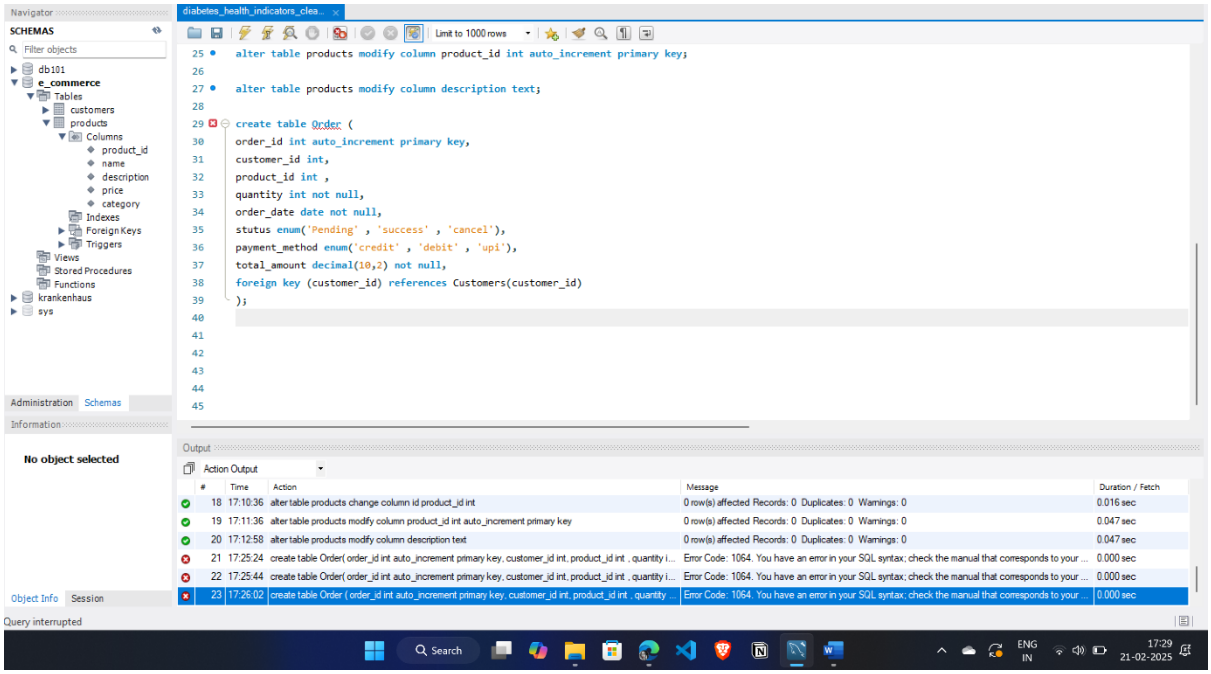
d. quantity - int not null,

e. order_date - date not null,

f. status - enum(Pending, Success, Cancel),

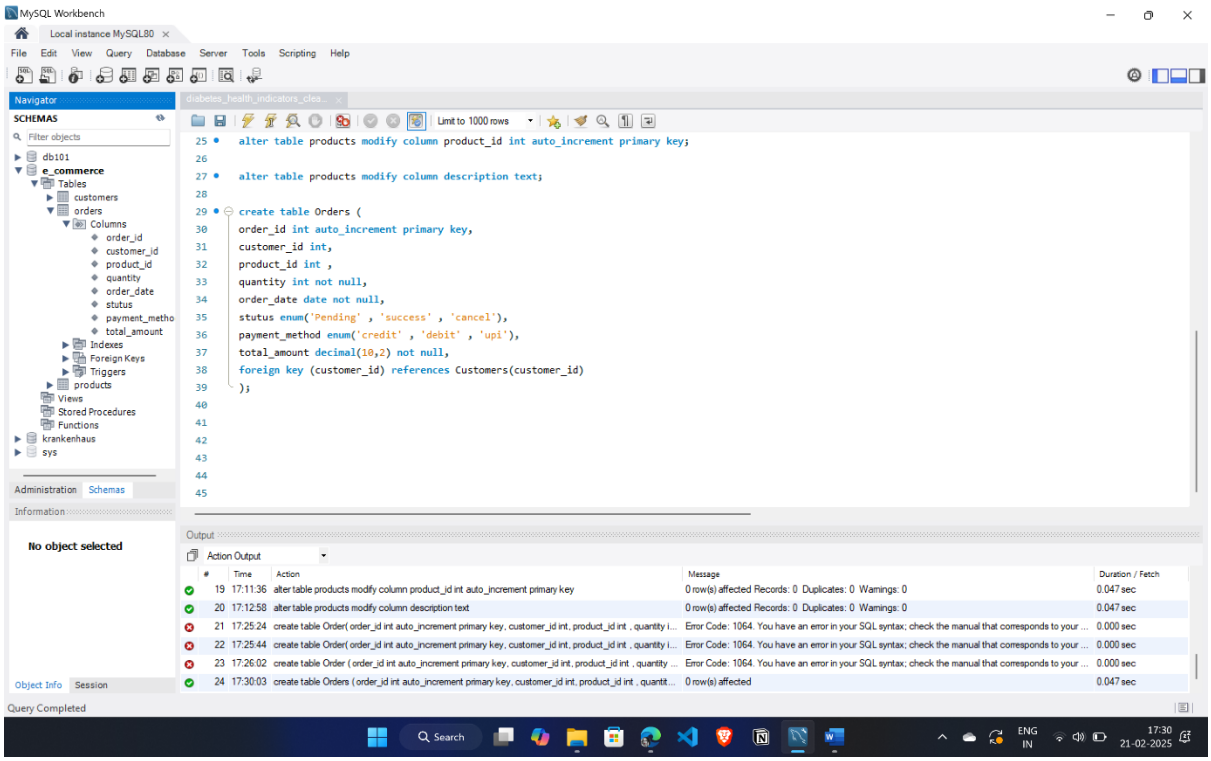
g. payment_method - enum(Credit, Debit, UPI),

h. total_amount - decimal(10, 2) not null



NOTE: SQL showing error because we cannot name your table 'order' or 'ORDER' as it is a reserved keyword in SQL

Thus:



Modify Orders Table:

a. Change table name Order -> Orders (already did in previous query)

b. Set default value pending in status.

c. Modify payment_method ENUM to add one more value: 'COD'

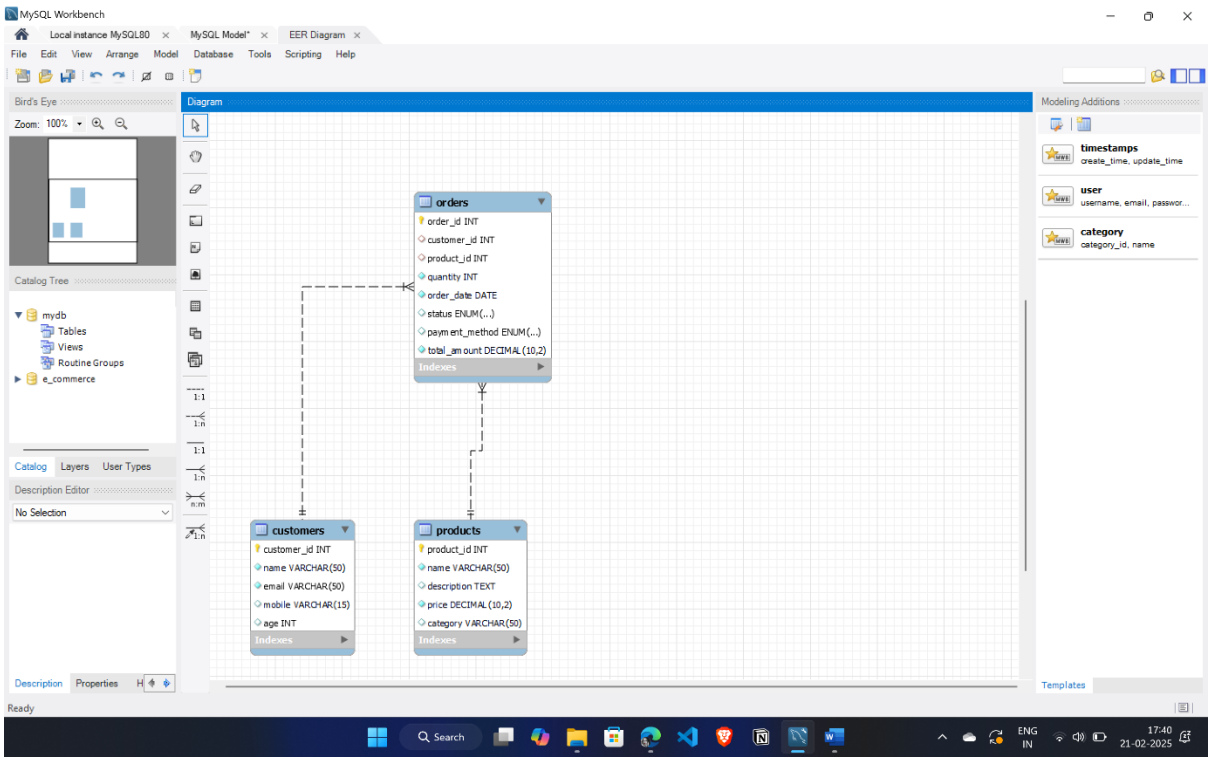
d. Make product id as foreign key

```
40
41 •  alter table  Orders rename to Orders;
42 •  alter table orders change column stutus status  enum('Pending' , 'success' , 'cancel') default 'Pending' ;
43
44 •  alter table Orders modify column payment_method enum('Credit', 'Debit', 'UPI', 'COD');
45
46 •  alter table Orders add constraint fk_product foreign key (product_id) references Products(product_id);
47
48
49
```

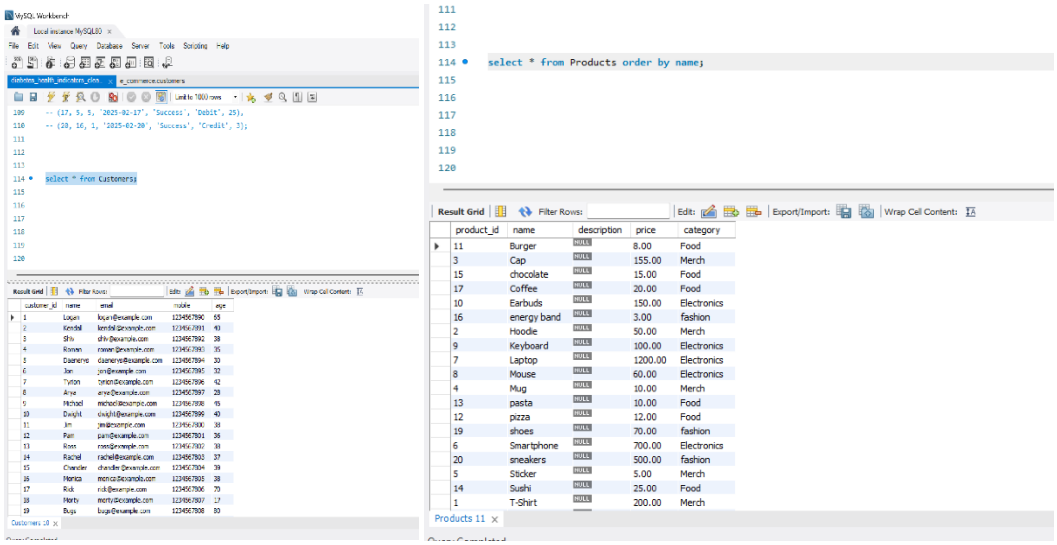
Output

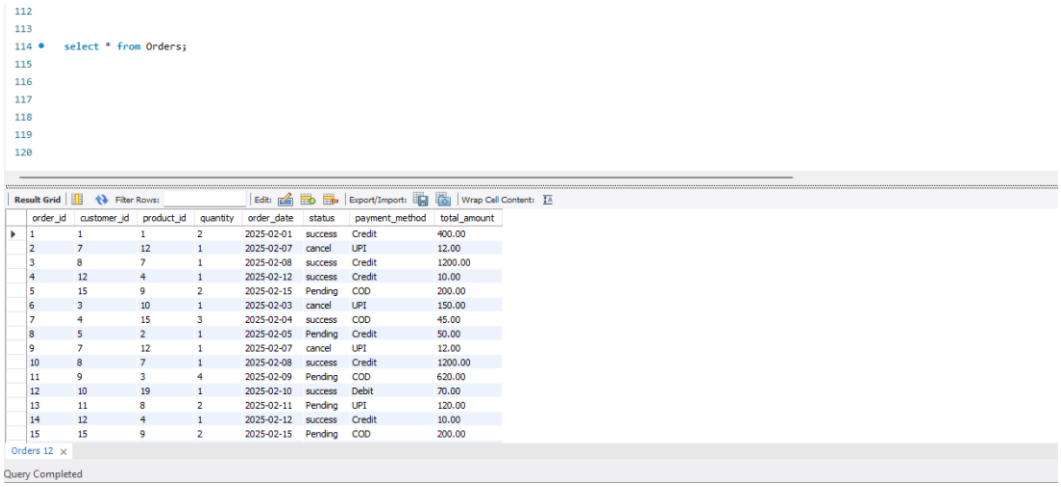
Action Output

#	Time	Action	Message	Duration / Fetch
24	17:30:03	create table Orders (order_id int auto_increment primary key, customer_id int, product_id int , quantit...	0 row(s) affected	0.047 sec
25	17:33:41	alter table Orders rename to Orders	0 row(s) affected	0.016 sec
26	17:35:40	alter table orders change column stutus status enum('Pending' , 'success' , 'cancel') default 'Pending'	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
27	17:36:50	alter table Order modify column payment_method enum('Credit', 'Debit', 'UPI', 'COD')	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your ...	0.000 sec
28	17:37:01	alter table Orders modify column payment_method enum('Credit', 'Debit', 'UPI', 'COD')	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.016 sec
29	17:38:15	alter table Orders add constraint fk_product foreign key (product_id) references Products(product_id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.063 sec



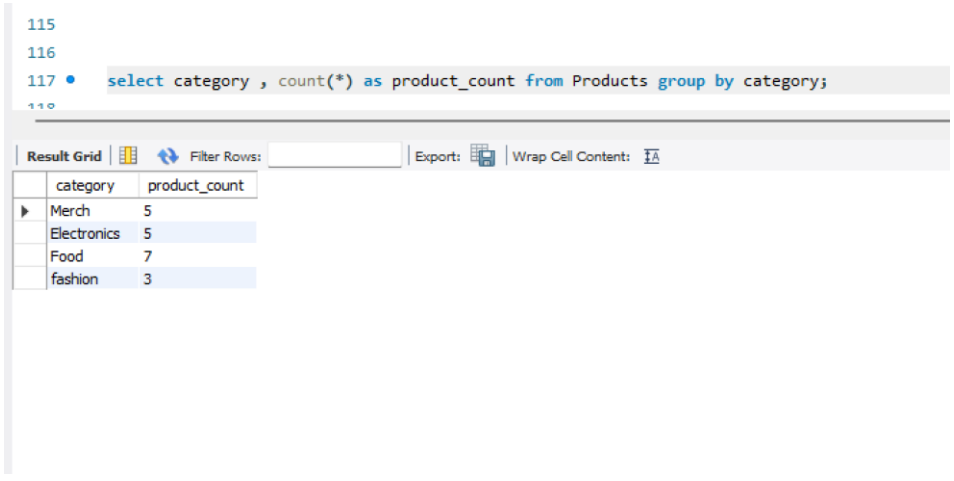
Insert 20 sample records in all the tables.





Perform following queries:

a. Count the number of products as product_count in each category.



category	product_count
Merch	5
Electronics	5
Food	7
fashion	3

b. Retrieve all products that belong to the 'Electronics' category, have a price between $50 and $500, and whose name contains the letter 'a'.

```
118
119     -- Retrieve all products that belong to the 'Electronics' category, have a price between $50 and $500, and whose name contains the letter 'a'.
120 •   select * from Products where category ='Electronics' and  name like '%a%' and price>=50 and price<=500;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

product_id	name	description	price	category
9	Keyboard	NULL	100.00	Electronics
10	Earbuds	NULL	150.00	Electronics
NULL	NULL	NULL	NULL	NULL

c. Get the top 5 most expensive products in the 'Electronics' category, skipping the first 2.

```
122     -- c.   Get the top 5 most expensive products in the 'Electronics' category, skipping the first 2.
123 •   select * from Products where category='Electronics' order by price desc limit 2,5;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rows:

product_id	name	description	price	category
10	Earbuds	NULL	150.00	Electronics
9	Keyboard	NULL	100.00	Electronics
8	Mouse	NULL	60.00	Electronics
NULL	NULL	NULL	NULL	NULL

Products 18 | Apply | Revert

Output

Action Output

#	Time	Action	Message	Duration / Fetch
47	18:30:49	select * from Products where category ='Electronics' and name like '%a%' and price>=50 and price<=500 LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
48	18:34:29	select * from Products where category='Electronics' order by price desc limit 5	5 row(s) returned	0.000 sec / 0.000 sec
49	18:34:48	select * from Products where category='Electronics' order by price desc limit (5,2)	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version...	0.000 sec
50	18:35:01	select * from Products where category='Electronics' order by price desc limit(5,2)	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version...	0.000 sec
51	18:35:16	select * from Products where category='Electronics' order by price desc limit 5 2 -- d.Retrieve customers who have not plac...	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version...	0.000 sec
52	18:36:37	select * from Products where category='Electronics' order by price desc limit 2,5	3 row(s) returned	0.000 sec / 0.000 sec

uery Completed

d. Retrieve customers who have not placed any orders.

```
23 •    select * from Products where category='Electronics' order by price desc limit 2,5;
24      -- d.   Retrieve customers who have not placed any orders.
25 •    select * from Customers
26      left join Orders
27      using(customer_id)
28      where product_id is null
29
30
31
32
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	name	email	mobile	age	order_id	product_id	quantity	order_date	status	payment_method	total_amount
2	Kendall	kendall@example.com	1234567891	40	NULL	NULL	NULL	NULL	NULL	NULL	NULL
6	Jon	jon@example.com	1234567895	32	NULL	NULL	NULL	NULL	NULL	NULL	NULL
13	Ross	ross@example.com	1234567802	38	NULL	NULL	NULL	NULL	NULL	NULL	NULL
14	Rachel	rachel@example.com	1234567803	37	NULL	NULL	NULL	NULL	NULL	NULL	NULL
18	Morty	morty@example.com	1234567807	17	NULL	NULL	NULL	NULL	NULL	NULL	NULL
19	Bugs	bugs@example.com	1234567808	80	NULL	NULL	NULL	NULL	NULL	NULL	NULL

e. Find the average total amount spent by each customer.

```
130
131     -- e.   Find the average total amount spent by each customer.
132 •   select c.name , ifnull(avg(o.total_amount),0) as average_amount_spent from Customers c left join  Orders o  using(customer_id)
133     group by c.name;
134
135
136
137
138
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	average_amount_spent
Logan	400.000000
Kendall	0.000000
Shiv	150.000000
Roman	45.000000
Daenerys	50.000000
Jon	0.000000
Tyrion	12.000000
Arya	1200.000000
Michael	620.000000
Dwight	70.000000
Jim	120.000000
Pam	10.000000
Ross	0.000000
Rachel	0.000000
Chandler	200.000000
Monica	15.000000
Rick	25.000000
Morty	0.000000
Bugs	0.000000
Daffy	3.000000

f. Get the products that have a price less than the average price of all products.

```
135
136     -- f.   Get the products that have a price less than the average price of all products
137 •   select * from Products where price<(select avg(price) from Products);
138
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

product_id	name	description	price	category
2	Hoodie	NULL	50.00	Merch
3	Cap	NULL	155.00	Merch
4	Mug	NULL	10.00	Merch
5	Sticker	NULL	5.00	Merch
8	Mouse	NULL	60.00	Electronics
9	Keyboard	NULL	100.00	Electronics
10	Earbuds	NULL	150.00	Electronics
11	Burger	NULL	8.00	Food
12	pizza	NULL	12.00	Food
13	pasta	NULL	10.00	Food
14	Sushi	NULL	25.00	Food
15	chocolate	NULL	15.00	Food
16	energy b...	NULL	3.00	fashion
17	Coffee	NULL	20.00	Food
18	tea	NULL	10.00	Food
19	shoes	NULL	70.00	fashion
NULL	NULL	NULL	NULL	NULL

g. Calculate the total quantity of products ordered by each customer:

```
139
140     -- g.   Calculate the total quantity of products ordered by each customer:
141 •   select c.name , ifnull(sum(o.quantity),0) as total_quantity from Customers c
142     left join Orders o
143     using(customer_id)
144     group by c.customer_id;
145
146
147
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	total_quantity
Logan	2
Kendall	0
Shiv	1
Roman	3
Daenerys	1
Jon	0
Tyrion	2
Arya	2
Michael	4
Dwight	1
Jim	2
Pam	2
Ross	0
Rachel	0
Chandler	4
Monica	1
Rick	5
Morty	0
Bugs	0
Daffy	1

h. List all orders along with customer name and product name.

```
147
148     -- h.   List all orders along with customer name and product name.
149 •   select c.name  Customer_name, p.name Product_name, o.* from Orders o
150     inner join Customers c using(customer_id)
151     inner join Products p using(product_id)
152
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Customer_name	Product_name	order_id	customer_id	product_id	quantity	order_date	status	payment_method	total_amount
Logan	T-Shirt	1	1	1	2	2025-02-01	success	Credit	400.00
Tyrion	pizza	2	7	12	1	2025-02-07	cancel	UPI	12.00
Arya	Laptop	3	8	7	1	2025-02-08	success	Credit	1200.00
Pam	Mug	4	12	4	1	2025-02-12	success	Credit	10.00
Chandler	Keyboard	5	15	9	2	2025-02-15	Pending	COD	200.00
Shiv	Earbuds	6	3	10	1	2025-02-03	cancel	UPI	150.00
Roman	chocolate	7	4	15	3	2025-02-04	success	COD	45.00
Daenerys	Hoodie	8	5	2	1	2025-02-05	Pending	Credit	50.00
Tyrion	pizza	9	7	12	1	2025-02-07	cancel	UPI	12.00
Arya	Laptop	10	8	7	1	2025-02-08	success	Credit	1200.00
Michael	Cap	11	9	3	4	2025-02-09	Pending	COD	620.00
Dwight	shoes	12	10	19	1	2025-02-10	success	Debit	70.00
Jim	Mouse	13	11	8	2	2025-02-11	Pending	UPI	120.00
Pam	Mug	14	12	4	1	2025-02-12	success	Credit	10.00
Chandler	Keyboard	15	15	9	2	2025-02-15	Pending	COD	200.00
Monica	Sushi	16	16	14	1	2025-02-16	success	Credit	15.00
Rick	Sticker	17	17	5	5	2025-02-17	success	Debit	25.00
Daffy	energy band	18	20	16	1	2025-02-20	success	Credit	3.00

i. Find products that have never been ordered.

```
155
156     -- i.   Find products that have never been ordered.
157
158 •   select p.* from Products p
159     left join Orders o
160     using(product_id)
161     where o.order_id is null
162
163
164
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

product_id	name	description	price	category
6	Smartphone	NULL	700.00	Electronics
11	Burger	NULL	8.00	Food
13	pasta	NULL	10.00	Food
17	Coffee	NULL	20.00	Food
18	tea	NULL	10.00	Food
20	sneakers	NULL	500.00	fashion