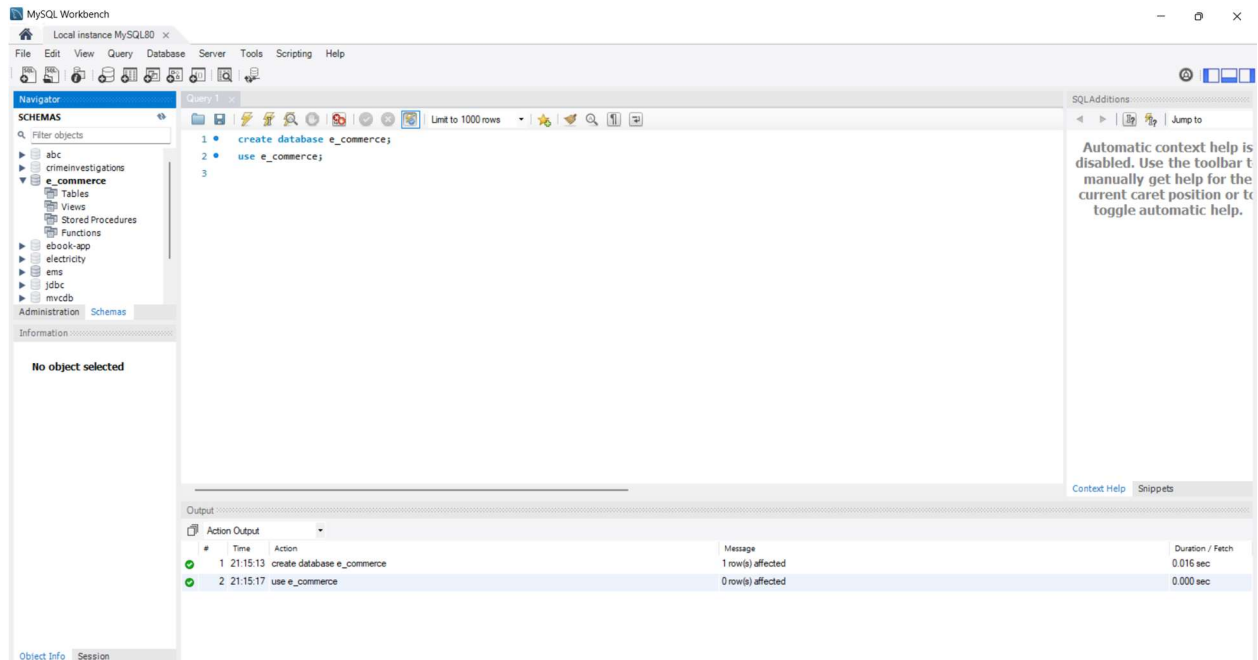


Basic SQL Assignment

1. Create Database e_commerce



2. Create following Tables:

Customers:

- customer_id - int auto-increment primary key
- name - varchar(50)
- email - varchar(50)
- mobile - varchar(15)

Products:

- id - int
- name - varchar(50) not null
- description - varchar(200)
- price - decimal(10, 2) not null
- category - varchar(50)

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- abc
- crimeinvestigations
- e_commerce
 - Tables
 - Views
 - Stored Procedures
 - Functions
- ebisok-400
- electricity
- ems
- jdbc
- mycvdb

Administration Schemas

Information

Schema: e_commerce

Query 1 x

```
1 create database e_commerce;
2 use e_commerce;
3
4 create table Customers(customer_id int primary key auto_increment, name varchar(50), email varchar(50), mobile varchar(15));
5 create table Products(id int, name varchar(50) not null, description varchar(200), price decimal(10,2) not null, category varchar(50));
6 desc Customers;
7
```

Result Grid

Field	Type	Null	Key	Default	Extra
customer_id	int	NO	PK		auto_increment
name	varchar(50)	YES			
email	varchar(50)	YES			
mobile	varchar(15)	YES			

Result 1 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	22:19:59	use e_commerce	0 row(s) affected	0.000 sec
2	22:20:04	create table Customers(customer_id int primary key auto_increment, name varchar(50), email varchar(50), mobile ...	0 row(s) affected	0.032 sec
3	22:20:09	create table Products(id int, name varchar(50) not null, description varchar(200), price decimal(10,2) not null, categ...	0 row(s) affected	0.031 sec
4	22:20:22	desc Customers	4 row(s) returned	0.031 sec / 0.000 sec

Object Info Session

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- abc
- crimeinvestigations
- e_commerce
 - Tables
 - customers
 - products
 - Columns
 - Indexes
 - Foreign Keys
 - Views
 - Stored Procedures

Administration Schemas

Information

Table: products

Columns:

- id int
- name varchar(50)
- description varchar(200)
- price decimal(10,2)
- category varchar(50)

Query 1 x

```
1 create database e_commerce;
2 use e_commerce;
3
4 create table Customers(customer_id int primary key auto_increment, name varchar(50), email varchar(50), mobile varchar(15));
5 create table Products(id int, name varchar(50) not null, description varchar(200), price decimal(10,2) not null, category varchar(50));
6 desc Customers;
7 desc Products;
```

Result Grid

Field	Type	Null	Key	Default	Extra
id	int	YES			
name	varchar(50)	NO			
description	varchar(200)	YES			
price	decimal(10,2)	NO			
category	varchar(50)	YES			

Result 4 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	22:25:09	desc Products	5 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

3. Modify Tables(using Alter keyword):

- Add not null on name and email in the Customers table
- Add unique key on email in the Customers table
- Add column age in the Customers table

The screenshot displays the MySQL Workbench interface for a local instance of MySQL 8.0. The 'Query 1' tab contains the following SQL script:

```
9 • alter table Customers modify column name varchar(50) not null;
10 • alter table Customers modify column email varchar(50) not null;
11 • alter table Customers add constraint unique_email unique(email);
12 • alter table Customers add age int;
13
14 • alter table Products change column id product_id int;
15 • alter table Products modify product_id int primary key auto_increment;
16 • alter table Products modify column description text;
17 • desc Customers;
18 • desc Products;
```

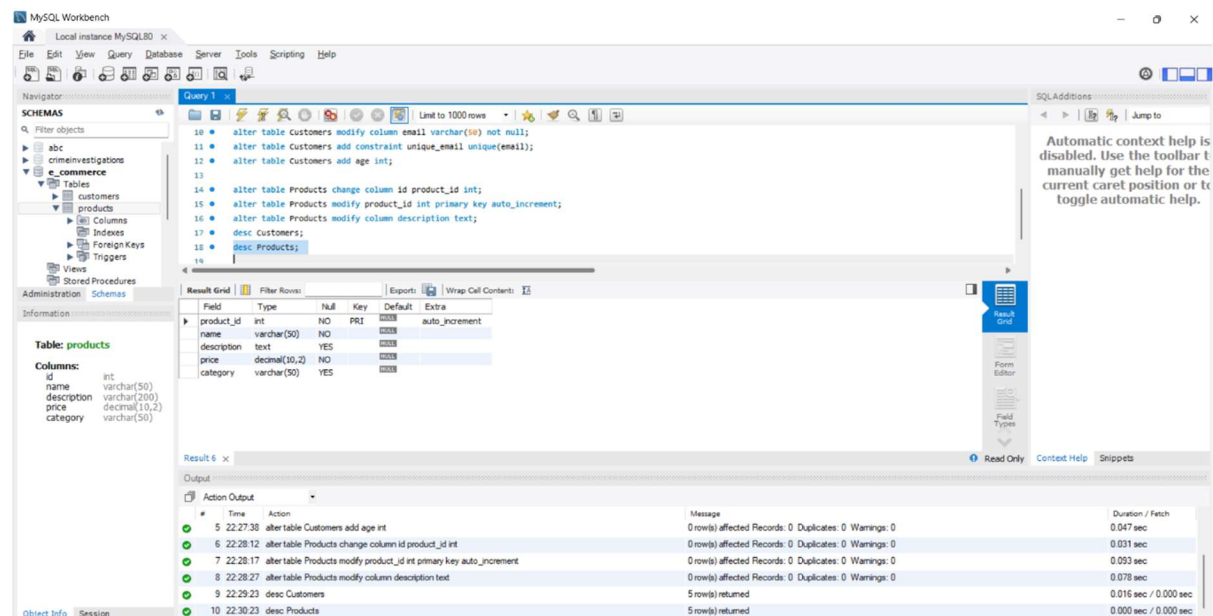
The 'Result Grid' shows the structure of the 'products' table:

Field	Type	Null	Key	Default	Extra
customer_id	int	NO	PRI	BULK	auto_increment
name	varchar(50)	NO		BULK	
email	varchar(50)	NO	UNI	BULK	
mobile	varchar(15)	YES		BULK	
age	int	YES		BULK	

The 'Action Output' log shows the execution of the SQL queries:

#	Time	Action	Message	Duration / Fetch
2	22:26:40	alter table Customers modify column name varchar(50) not null	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.093 sec
3	22:26:48	alter table Customers modify column email varchar(50) not null	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.078 sec
4	22:26:52	alter table Customers add constraint unique_email unique(email)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.032 sec
5	22:27:38	alter table Customers add age int	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec
6	22:28:12	alter table Products change column id product_id int	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
7	22:28:17	alter table Products modify product_id int primary key auto_increment	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.093 sec

- 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



4. 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

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```

20 • create table `Order` (order_id int primary key auto_increment,
21   customer_id int, product_id int, quantity int not null, order_date date not null,
22   status enum('Pending','Success','Cancel'), payment_method enum('Credit','Debit','UPI'),
23   total_amount decimal(10,2) not null,
24   foreign key(customer_id) references customers(customer_id)
25 );
26
27 • desc `Order`;
28

```

Field	Type	Null	Key	Default	Extra
order_id	int	NO	PRI		auto_increment
customer_id	int	YES	MUL		
product_id	int	YES	MUL		
quantity	int	NO			
order_date	date	NO			
status	enum('Pending','Success','Cancel')	YES			
payment_method	enum('Credit','Debit','UPI')	YES			
total_amount	decimal(10,2)	NO			

Result 2 x

Output:

#	Time	Action	Message	Duration / Fetch
1	11:10:03	create table `Order` (order_id int primary key auto_increment, customer_id int, product_id int, quantity int not null, or...	0 row(s) affected	0.109 sec
2	11:10:07	desc `Order`	8 row(s) returned	0.000 sec / 0.000 sec

5. Modify Orders Table(using Alter keyword):

- Change table name Order -> Orders
- Set default value pending in status.
- Modify payment_method ENUM to add one more value: 'COD'
- Make product id as foreign key

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```

28
29 • alter table `Order` rename to Orders;
30 • alter table Orders modify column status enum('Pending','Success','Cancel') default 'Pending';
31 • alter table Orders modify column payment_method enum('Credit','Debit','UPI','COD');
32 • alter table Orders add constraint fkey_product foreign key(product_id) references Products(product_id);
33
34
35

```

Field	Type	Null	Key	Default	Extra
order_id	int	NO	PRI		auto_increment
customer_id	int	YES	MUL		
product_id	int	YES	MUL		
quantity	int	NO			
order_date	date	NO			
status	enum('Pending','Success','Cancel')	YES		Pending	
payment_method	enum('Credit','Debit','UPI','COD')	YES			
total_amount	decimal(10,2)	NO			

Result 3 x

Output:

#	Time	Action	Message	Duration / Fetch
1	11:10:03	create table `Order` (order_id int primary key auto_increment, customer_id int, product_id int, quantity int not null, or...	0 row(s) affected	0.109 sec
2	11:10:07	desc `Order`	8 row(s) returned	0.000 sec / 0.000 sec
3	11:11:22	alter table `Order` rename to Orders	0 row(s) affected	0.062 sec
4	11:15:02	alter table Orders modify column status enum('Pending','Success','Cancel') default 'Pending'	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.046 sec
5	11:16:05	alter table Orders modify column payment_method enum('Credit','Debit','UPI','COD')	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
6	11:19:46	alter table Orders add constraint fkey_product foreign key(product_id) references Products(product_id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.219 sec

6. Insert 20 sample records in all the tables.

101
102 • `select * from Customers;`
103 • `select * from Products;`
104 • `select * from Orders;`
105

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

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

customer_id	name	email	mobile	age
1	Amit Sharma	amit.sharma@example.com	9876543210	25
2	Priya Verma	priya.verma@example.com	9876543211	30
3	Rahul Gupta	rahul.gupta@example.com	9876543212	28
4	Sneha Reddy	sneha.reddy@example.com	9876543213	35
5	Vikram Singh	vikram.singh@example.com	9876543214	40
6	Anjali Mehta	anjali.mehta@example.com	9876543215	22
7	Karan Patel	karan.patel@example.com	9876543216	27
8	Pooja Das	pooja.das@example.com	9876543217	32
9	Arjun Nair	arjun.nair@example.com	9876543218	29
10	Riya Malhotra	riya.malhotra@example.com	9876543219	26
11	Manish Joshi	manish.joshi@example.com	9876543220	33
12	Neha Kulkarni	neha.kulkarni@example.com	9876543221	31
13	Siddharth Rao	siddharth.rao@example.com	9876543222	24
14	Meera Iyer	meera.iyer@example.com	9876543223	36
15	Rohan Khanna	rohan.khanna@example.com	9876543224	38
16	Swati Choudhary	swati.choudhary@example.com	9876543225	23
17	Akash Jain	akash.jain@example.com	9876543226	34
18	Divya Bhatt	divya.bhatt@example.com	9876543227	29
19	Nikhil Aggarwal	nikhil.aggarwal@example.com	9876543228	37
20	Sonal Kapoor	sonal.kapoor@example.com	9876541229	28

Customers 1 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	12:35:33	insert into products (product_id, name, description, price, category) values (1, 'Wireless Mouse', 'wireless mouse', 15.00, 'Electronics');	20 row(s) affected Records: 20 Duplicates: 0 Warnings: 0	0.016 sec
2	12:41:35	insert into orders (order_id, customer_id, product_id, quantity, order_date, status, payment_method, total_amount) values (1, 1, 1, 1, '2023-10-26', 'PENDING', 'CREDIT_CARD', 15.00);	20 row(s) affected Records: 20 Duplicates: 0 Warnings: 0	0.016 sec

102 • `select * from Customers;`
103 • `select * from Products;`
104 • `select * from Orders;`
105
106

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

product_id	name	description	price	category
3	Running Shoes	Lightweight running shoes with breathable mesh upper.	75.50	Sportswear
4	Yoga Mat	Non-slip yoga mat made from eco-friendly material.	30.00	Fitness
5	Coffee Maker	Automatic coffee maker with programmable settings.	89.99	Home
6	Smartphone Case	Shockproof and waterproof case for smartphone.	19.99	Accessories
7	Desk Lamp	LED desk lamp with adjustable brightness and color.	35.00	Home
8	Gaming Keyboard	Mechanical keyboard with RGB lighting and custom keys.	79.99	Gaming
9	Electric Kettle	1.5L stainless steel electric kettle with auto shut-off.	40.00	Kitchen
10	Backpack	Water-resistant backpack with multiple compartments.	55.00	Travel
11	Noise-Canceling Headphones	Over-ear headphones with active noise cancellation.	120.00	Electronics
12	Digital Watch	Waterproof digital watch with fitness tracking features.	60.00	Accessories
13	Wireless Charger	Fast wireless charger compatible with most smartphones.	25.00	Electronics
14	Office Chair	Ergonomic office chair with lumbar support and adjustable height.	150.00	Furniture
15	Toothbrush	Rechargeable electric toothbrush with multiple cleaning modes.	35.99	Home
16	Blender	High-speed blender for smoothies and shakes.	70.00	Kitchen
17	Camping Tent	4-person waterproof camping tent with easy setup.	120.00	Outdoor Gear
18	Cookware Set	Non-stick cookware set including pots and pans.	85.50	Kitchen
19	Smart Thermostat	Wi-Fi-enabled thermostat for smart home integration.	199.99	Home
20	Bookshelf	5-tier wooden bookshelf for home or office use.	100.00	Furniture

Limit to 1000 rows

```

100 (20, 20, 4, 7, '2024-06-13', 'Pending', 'Debit', 9858.45);
101
102 • select * from Customers;
103 • select * from Products;
104 • select * from Orders;

```

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

	order_id	customer_id	product_id	quantity	order_date	status	payment_method	total_amount
▶	1	1	5	10	2024-10-22	Pending	COD	6105.72
	2	2	1	5	2024-11-23	Cancel	Debit	8329.56
	3	3	15	9	2024-03-11	Cancel	COD	360.39
	4	4	20	8	2024-10-23	Pending	Credit	10204.08
	5	3	2	9	2024-11-05	Success	COD	11495.58
	6	6	7	3	2024-08-03	Pending	Debit	12175.77
	7	7	12	1	2024-04-30	Cancel	COD	2961.84
	8	18	12	8	2024-12-14	Pending	Debit	6193.05
	9	9	19	6	2024-08-04	Pending	Debit	4982.73
	10	10	3	4	2024-08-24	Success	UPI	10403.34
	11	11	6	8	2024-06-30	Success	UPI	7491.93
	12	12	9	10	2025-02-02	Cancel	UPI	8856.00
	13	13	16	8	2025-01-05	Success	Credit	886.83
	14	1	17	1	2025-01-30	Cancel	Debit	3206.61
	15	15	8	10	2024-10-01	Pending	Debit	7782.21
	16	16	11	8	2024-07-25	Pending	COD	11058.93
	17	17	13	8	2024-08-30	Success	COD	5544.84
	18	18	1	8	2024-10-17	Success	COD	709.71
	19	11	1	6	2025-01-12	Success	Credit	10758.81
	20	20	4	7	2024-06-13	Pending	Debit	9858.45

Orders 3 x

7. Perform following queries:

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

Limit to 1000 rows

```

102 • select * from Customers;
103 • select * from Products;
104 • select * from Orders;
105
106 • select category, count(*) as product_count from Products group by category;

```

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

	category	product_count
▶	Electronics	4
	Sportswear	1
	Fitness	1
	Home	4
	Accessories	2
	Gaming	1
	Kitchen	3
	Travel	1
	Furniture	2
	Outdoor Gear	1

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

```
104 • select * from Orders;
105
106 • select category,count(*) as product_count from Products group by category;
107
108 • select * from Products where category='Electronics' and price between 50 and 500 and name like "%a%";
```

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

	product_id	name	description	price	category
▶	11	Noise-Canceling Headphones	Over-ear headphones with active noise cancella...	120.00	Electronics
*	NULL	NULL	NULL	NULL	NULL

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

```
106 • select category,count(*) as product_count from Products group by category;
107
108 • select * from Products where category='Electronics' and price between 50 and 500 and name like "%a%";
109
110 • select name from Products where category="Electronics" order by price desc limit 5 offset 2;
111
112
113
114
115
```

Limit to 1000 rows | | | | |

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

	name
▶	Wireless Mouse
	Wireless Charger

d. Retrieve customers who have not placed any orders.

```
111
112 • select c.customer_id,name from customers c left join orders o on c.customer_id=o.customer_id where o.customer_id is null;
```

Result Grid

customer_id	name
5	Vikram Singh
8	Pooja Das
14	Meera Iyer
19	Nikhil Aggarwal

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

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- Filter objects
- abc
- crimeinvestigations
- e-commerce
 - customers
 - orders
 - products
 - Views
 - Stored Procedures
 - Functions
- ebook-app
- electricity
- Administration
- Schemas

Information

No object selected

Query 1 x

```
114 • select c.customer_id , c.name, avg(o.total_amount) as avg_spent
115 from customers c
116 left join orders o on c.customer_id=o.customer_id
117 group by c.customer_id;
```

Result Grid

customer_id	name	avg_spent
1	Amit Sharma	4656.165000
2	Priya Verma	8329.560000
3	Rahul Gupta	5927.985000
4	Sheha Reddy	10204.080000
5	Vikram Singh	0.000000
6	Anjali Mehta	12175.770000
7	Karan Patel	2961.840000
8	Pooja Das	0.000000
9	Arjun Nair	4982.730000
10	Riya Malhotra	10403.340000
11	Harsh Joshi	9125.370000
12	Neha Kulkarni	8856.000000
13	Siddharth Rao	886.830000
14	Meera Iyer	0.000000
15	Rohan Khanna	7782.210000
16	Swati Choudhary	11058.930000
17	Akash Jain	5544.840000
18	Divya Shetty	3451.380000
19	Nikhil Aggarwal	0.000000
20	Sonal Kapoor	9858.450000

Output

Action Output

#	Time	Action	Message	Duration / Fetch
4	17:40:29	select name from Products where category="Electronics" order by price desc limit 5 offset 2	2 row(s) returned	0.000 sec / 0.000 sec
5	17:46:57	select c.customer_id , c.name, avg(o.total_amount) as avg_spent from customers c left join orders o on c.customer_id=o.customer_id group by c.customer_id;	20 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

SQL Additions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Read Only Context Help Snippets

ENG IN 5:47 PM 2/18/2025

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

```
119 • select product_id, name, price from products where price < (select avg(price) from products);
```

product_id	name	price
1	Wireless Mouse	25.99
2	Bluetooth Speaker	49.99
4	Yoga Mat	30.00
6	Smartphone Case	19.99
7	Desk Lamp	35.00
9	Electric Kettle	40.00
10	Backpack	55.00
12	Digital Watch	60.00
13	Wireless Charger	25.00
15	Toothbrush	35.99
16	Blender	70.00
NULL	NULL	NULL



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

```
121 • select c.customer_id, c.name, sum(o.quantity) as total_qunatity
122 from customers c
123 join orders o on c.customer_id=o.customer_id
124 group by c.customer_id;
```

customer_id	name	total_qunatity
1	Amit Sharma	11
2	Priya Verma	5
3	Rahul Gupta	18
4	Sneha Reddy	8
6	Anjali Mehta	3
7	Karan Patel	1
9	Arjun Nair	6
10	Riya Malhotra	4
11	Manish Joshi	14
12	Neha Kulkarni	10
13	Siddharth Rao	8
15	Rohan Khanna	10
16	Swati Choud...	8
17	Akash Jain	8
18	Divya Bhatt	16
20	Sonal Kapoor	7

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



```
127 • select o.order_id, c.name as customer_name, p.name as product_name, o.quantity, o.total_amount
128 from Orders o
129 join Customers c on o.customer_id = c.customer_id
130 join Products p on o.product_id = p.product_id;
131
```

Result Grid					
Filter Rows: <input type="text"/>					
Export:  Wrap Cell Content: 					
	order_id	customer_name	product_name	quantity	total_amount
▶	1	Amit Sharma	Coffee Maker	10	6105.72
	2	Priya Verma	Wireless Mouse	5	8329.56
	3	Rahul Gupta	Toothbrush	9	360.39
	4	Sneha Reddy	Bookshelf	8	10204.08
	5	Rahul Gupta	Bluetooth Speaker	9	11495.58
	6	Anjali Mehta	Desk Lamp	3	12175.77
	7	Karan Patel	Digital Watch	1	2961.84
	8	Divya Bhatt	Digital Watch	8	6193.05
	9	Arjun Nair	Smart Thermostat	6	4982.73
	10	Riya Malhotra	Running Shoes	4	10403.34
	11	Manish Joshi	Smartphone Case	8	7491.93
	12	Neha Kulkarni	Electric Kettle	10	8856.00
	13	Siddharth Rao	Blender	8	886.83
	14	Amit Sharma	Camping Tent	1	3206.61
	15	Rohan Khanna	Gaming Keyboard	10	7782.21
	16	Swati Choudhary	Noise-Canceling ...	8	11058.93
	17	Akash Jain	Wireless Charger	8	5544.84
	18	Divya Bhatt	Wireless Mouse	8	709.71
	19	Manish Joshi	Wireless Mouse	6	10758.81
	20	Sonal Kanour	Yoga Mat	7	9858.45

Result 19 ×

i. Find products that have never been ordered.

```
132
133 • select p.* from products p
134 left join orders o on p.product_id = o.product_id
135 where o.order_id is null;
```

Result Grid					
Filter Rows: <input type="text"/>					
Export:  Wrap Cell Content: 					
	product_id	name	description	price	category
▶	10	Backpack	Water-resistant backpack with multiple compart...	55.00	Travel
	14	Office Chair	Ergonomic office chair with lumbar support and ...	150.00	Furniture
	18	Cookware Set	Non-stick cookware set including pots and pans.	85.50	Kitchen