

MY SQL PROJECT

Introduction:

In today's digital world, managing data efficiently is essential for all organizations. MySQL, a popular open-source relational database system, uses SQL to store, retrieve, and manage information reliably. This project builds a database system to handle key operations smoothly.

RISHU HRITHIK

MY SQL PROJECT

Objectives:

- To design and implement a fully normalized database schema (up to 3NF) with related tables, ensuring data integrity through primary/foreign keys and constraints.
- To develop comprehensive CRUD operations via SQL DML statements, enabling efficient Create (INSERT), Read (SELECT with JOINs), Update (UPDATE), and Delete (DELETE) functionalities across complex queries.
- To produce actionable analytical reports using aggregate functions (COUNT, SUM, AVG), GROUP BY, subqueries, and views for data-driven insights.

RISHU HRITHIK

MY SQL PROJECT

Scope :

This project covers core database design, data operations, and basic reporting using sample data. It excludes advanced features like user authentication, real-time updates, or external API integrations.

Methodology :

The project follows standard database development steps—creating ER diagrams for planning, using DDL statements for table structure, DML for data insertion, and DQL queries for retrieval and analysis. Tools include MySQL Workbench for design and MySQL command line for execution.

RISHU HRITHIK

MY SQL PROJECT

TABLES CREATION :

- `use mini_project;`

```
• CREATE TABLE customers (
    customer_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE,
    phone VARCHAR(15),
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP
);
```

```
• CREATE TABLE products (
    product_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    category VARCHAR(50) NOT NULL,
    price DECIMAL(10,2) NOT NULL,
    stock_quantity INT NOT NULL DEFAULT 0,
    added_on DATETIME DEFAULT CURRENT_TIMESTAMP
);
```

```
• CREATE TABLE orders (
    order_id INT PRIMARY KEY AUTO_INCREMENT,
    customer_id INT,
    order_date DATETIME DEFAULT CURRENT_TIMESTAMP,
    status VARCHAR(20) DEFAULT 'Pending',
    total_amount DECIMAL(10,2),
    FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
);
```

```
• CREATE TABLE order_items (
    order_item_id INT PRIMARY KEY AUTO_INCREMENT,
    order_id INT,
    product_id INT,
    quantity INT NOT NULL CHECK (quantity > 0),
    item_price DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (order_id) REFERENCES orders(order_id),
    FOREIGN KEY (product_id) REFERENCES products(product_id)
);
```

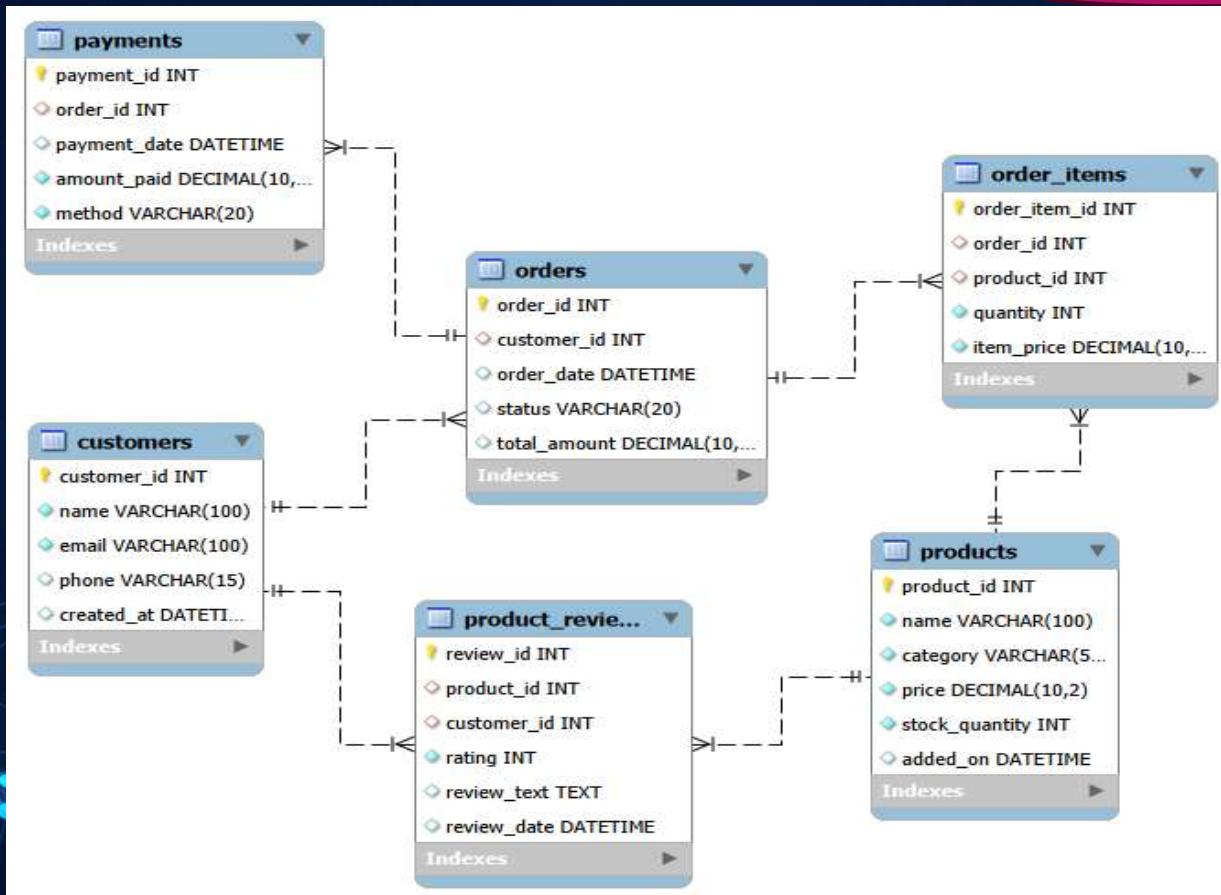
```
• CREATE TABLE payments (
    payment_id INT PRIMARY KEY AUTO_INCREMENT,
    order_id INT,
    payment_date DATETIME DEFAULT CURRENT_TIMESTAMP,
    amount_paid DECIMAL(10,2) NOT NULL CHECK (amount_paid > 0),
    method VARCHAR(20) NOT NULL,
    FOREIGN KEY (order_id) REFERENCES orders(order_id)
);
```

```
• CREATE TABLE product_reviews (
    review_id INT PRIMARY KEY AUTO_INCREMENT,
    product_id INT,
    customer_id INT,
    rating INT NOT NULL,
    review_text TEXT,
    review_date DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (product_id) REFERENCES products(product_id),
    FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
);
```

RISHU HRITHIK

MY SQL PROJECT

ER Diagram



RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:1

1. Retrieve customer names and emails for email marketing

QUERY

```
select name, email from customers;
```

OUTPUT

	name	email
▶	Thomas Owens	user1@example.com
	Charles Grant	user2@example.com
	Kaitlin Richards	user3@example.com
	Christina Williams	user4@example.com
	David Allen	user5@example.com
	Mark Duke	user6@example.com
	Briana Wright	user7@example.com
	John Bryan	user8@example.com
	Jason Thompson	user9@example.com
	Shawn Hill	user10@example.com
	Melton Jenkins	user11@example.com

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:1

2. View complete product catalog with all available details

QUERY

```
select * from products;
```

OUTPUT

product_id	name	category	price	stock_quantity	added_on
1	Plant No	Home	639.43	152	2024-01-30 06:30:53
2	Population Social	Clothing	4813.68	84	2025-05-30 10:02:50
3	Available Answer	Electronics	2529.51	101	2025-04-13 01:11:46
4	Any Question	Clothing	4759.28	179	2025-06-03 13:34:03
5	Natural Network	Toys	4722.66	75	2023-11-06 00:47:37
6	If Whatever	Electronics	177.40	64	2024-12-19 10:37:14
7	Response Indeed	Clothing	4897.36	36	2025-03-29 02:43:08
8	Every Amount	Home	4173.60	156	2025-04-30 03:11:10
9	Common Study	Toys	985.19	171	2023-07-20 13:06:42
10	Development System	Electronics	4801.78	153	2025-03-12 08:22:57

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:1

3. List all unique product categories

QUERY

```
select distinct(category) from  
products;
```

OUTPUT

	category
▶	Home
	Clothing
	Electronics
	Toys
	Books

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:1

4. Show all products priced above ₹1,000

QUERY

```
select * from products where price >  
1000;
```

OUTPUT

product_id	name	category	price	stock_quantity	added_on
2	Population Social	Clothing	4813.68	84	2025-05-30 10:02:50
3	Available Answer	Electronics	2529.51	101	2025-04-13 01:11:46
4	Any Question	Clothing	4759.28	179	2025-06-03 13:34:03
5	Natural Network	Toys	4722.66	75	2023-11-06 00:47:37
7	Response Indeed	Clothing	4897.36	36	2025-03-29 02:43:08
8	Every Amount	Home	4173.60	156	2025-04-30 03:11:10
10	Development System	Electronics	4801.78	153	2025-03-12 08:22:57
11	Build Her	Books	1852.64	150	2024-09-08 01:09:15
12	Action Ask	Electronics	4017.01	19	2025-02-14 03:38:06
13	Full West	Books	2112.33	172	2023-09-15 03:13:38

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:1

5. Display products within a mid-range price bracket (₹2,000 to ₹5,000)

QUERY

```
select * from products where  
price>2000 and price<5000;
```

-- we can also use price BETWEEN
2000 AND 5000.

OUTPUT

	product_id	name	category	price	stock_quantity	added_on
▶	2	Population Social	Clothing	4813.68	84	2025-05-30 10:02:50
	3	Available Answer	Electronics	2529.51	101	2025-04-13 01:11:46
	4	Any Question	Clothing	4759.28	179	2025-06-03 13:34:03
	5	Natural Network	Toys	4722.66	75	2023-11-06 00:47:37
	7	Response Indeed	Clothing	4897.36	36	2025-03-29 02:43:08
	8	Every Amount	Home	4173.60	156	2025-04-30 03:11:10
	10	Development System	Electronics	4801.78	153	2025-03-12 08:22:57
	12	Action Ask	Electronics	4017.01	19	2025-02-14 03:38:06
	13	Full West	Books	2112.33	172	2023-09-15 03:13:38
	17	Everything Plant	Books	2496.68	120	2023-10-08 20:11:55

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:1

6. Fetch data for specific customer IDs (e.g., from loyalty program list)

QUERY

```
select * from customers where  
customer_id in (2,4,5,7,25);
```

OUTPUT

	customer_id	name	email	phone	created_at
▶	2	Charles Grant	user2@example.com	9153947511	2023-11-25 15:45:24
	4	Christina Williams	user4@example.com	586-605-5061x06	2024-10-27 17:19:38
	5	David Allen	user5@example.com	(751)456-8289x1	2023-10-29 02:43:00
	7	Briana Wright	user7@example.com	223-833-9635	2023-06-25 00:35:43
	25	Cindy Hart	user25@example.com	617.574.8421x41	2025-05-27 18:02:54
	NULL	NULL	NULL	NULL	NULL
	8				

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:1

7. Identify customers whose names start with the letter 'A'

QUERY

```
select * from customers  
where substring(name,1,1) in ("a",  
"A");
```

OUTPUT

	customer_id	name	email	phone	created_at
▶	15	Austin Flores	user15@example.com	329.901.1576x66	2024-06-13 09:03:42
	16	Amy Landry	user16@example.com	+1-278-019-3748	2024-02-28 17:51:50
	19	Amanda Bright	user19@example.com	380.981.9798x69	2024-12-20 22:58:15
	27	Adrienne Green	user27@example.com	530.644.8455x93	2023-08-22 01:55:29
	HULL	HULL	HULL	HULL	HULL

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:1

8. List electronics products priced under ₹3,000

QUERY

```
select * from products where  
category = "Electronics" and price <  
3000;
```

OUTPUT

	product_id	name	category	price	stock_quantity	added_on
▶	3	Available Answer	Electronics	2529.51	101	2025-04-13 01:11:46
	6	If Whatever	Electronics	177.40	64	2024-12-19 10:37:14
	15	Place Low	Electronics	723.97	46	2023-07-05 14:36:07
	31	Series Page	Electronics	2070.37	83	2024-04-01 00:24:06
	34	Despite Win	Electronics	1340.34	64	2024-11-27 06:55:45
	44	Actually Term	Electronics	396.11	85	2023-11-02 13:09:20
	47	Southern Thing	Electronics	512.46	40	2024-02-28 17:57:38
	NULL	NULL	NULL	NULL	NULL	NULL

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:1

9. Display product names and prices in descending order of price

QUERY

```
select name, price from products  
order by price desc;
```

OUTPUT

	name	price
▶	Response Indeed	4897.36
	Population Social	4813.68
	Development System	4801.78
	Any Question	4759.28
	Fire Often	4734.89
	Natural Network	4722.66
	Build High	4707.14
	Serious Recognize	4523.10
	Study Total	4413.68
	Real Source	4398.66

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:1

10. Display product names and prices, sorted by price and then by name

QUERY

```
select name, price from products  
order by price desc, name;
```

OUTPUT

	name	price
►	Response Indeed	4897.36
	Population Social	4813.68
	Development System	4801.78
	Any Question	4759.28
	Fire Often	4734.89
	Natural Network	4722.66
	Build High	4707.14
	Serious Recognize	4523.10
	Study Total	4413.68
	Real Source	4398.66

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:2

1. Retrieve orders where customer information is missing (possibly due to data migration or deletion)

QUERY

```
select * from orders where  
customer_id is not null;
```

OUTPUT

	order_id	customer_id	order_date	status	total_amount
▶	1	20	2025-03-02 07:20:11	Delivered	9414.28
	2	18	2024-10-09 18:08:21	Shipped	532.20
	3	15	2025-05-08 00:08:27	Cancelled	5164.56
	4	11	2024-09-19 22:16:13	Delivered	9469.78
	5	12	2025-04-08 18:02:06	Pending	14501.86
	6	29	2024-10-25 07:33:59	Cancelled	31050.17
	7	22	2024-07-29 11:58:47	Shipped	3043.67
	8	19	2024-07-30 22:49:49	Cancelled	32714.06
	9	6	2025-06-10 17:00:25	Pending	24219.20
	10	28	2025-02-16 12:45:59	Delivered	24342.52

-- all orders customers detail is linked

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:2

2. Display customer names and emails using column aliases for frontend readability

QUERY

```
SELECT NAME AS CUST_NAME, EMAIL  
AS CUST_EMAIL FROM CUSTOMERS;
```

OUTPUT

	CUST_NAME	CUST_EMAIL
▶	Thomas Owens	user1@example.com
	Charles Grant	user2@example.com
	Kaitlin Richards	user3@example.com
	Christina Williams	user4@example.com
	David Allen	user5@example.com
	Mark Duke	user6@example.com
	Briana Wright	user7@example.com
	John Bryan	user8@example.com
	Jason Thompson	user9@example.com
	Shawn Hill	user10@example.com

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:2

3. Calculate total value per item ordered by multiplying quantity and item price

QUERY

```
SELECT * , QUANTITY*ITEM_PRICE AS  
TOTAL_ITEM_AMOUNT FROM  
order_items;
```

OUTPUT

	order_item_id	order_id	product_id	quantity	item_price	TOTAL_ITEM_AMOUNT
▶	1	1	19	2	4707.14	9414.28
	2	2	6	3	177.40	532.20
	3	3	9	3	985.19	2955.57
	4	3	23	1	2208.99	2208.99
	5	4	35	2	4734.89	9469.78
	6	5	19	1	4707.14	4707.14
	7	5	7	2	4897.36	9794.72
	8	6	7	3	4897.36	14692.08
	9	6	37	1	1429.45	1429.45
	10	6	15	1	723.97	723.97

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:2

4. Combine customer name and phone number in a single column

QUERY

```
SELECT concat(NAME, ' : ', PHONE) AS  
'NAME+PHONE' FROM CUSTOMERS;
```

OUTPUT

	NAME +PHONE
▶	Thomas Owens :142-479-1945
	Charles Grant :9153947511
	Kaitlin Richards :2073473421
	Christina Williams :586-605-5061x06
	David Allen :(751)456-8289x1
	Mark Duke :(144)957-2811
	Briana Wright :223-833-9635
	John Bryan :045.568.0798x27
	Jason Thompson :1862659420
	Shawn Hill :(268) 113-3152x7

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:2

5. Extract only the date part from order timestamps for date-wise reporting

QUERY

```
SELECT ORDER_ID, CUSTOMER_ID,  
DATE(ORDER_DATE) AS OrderDate,  
TIME(ORDER_DATE) AS OrderTime,  
STATUS, TOTAL_AMOUNT FROM  
ORDERS;
```

OUTPUT

	ORDER_ID	CUSTOMER_ID	OrderDate	OrderTime	STATUS	TOTAL_AMOUNT
▶	1	20	2025-03-02	07:20:11	Delivered	9414.28
	2	18	2024-10-09	18:08:21	Shipped	532.20
	3	15	2025-05-08	00:08:27	Cancelled	5164.56
	4	11	2024-09-19	22:16:13	Delivered	9469.78
	5	12	2025-04-08	18:02:06	Pending	14501.86
	6	29	2024-10-25	07:33:59	Cancelled	31050.17
	7	22	2024-07-29	11:58:47	Shipped	3043.67
	8	19	2024-07-30	22:49:49	Cancelled	32714.06
	9	6	2025-06-10	17:00:25	Pending	24219.20
	10	28	2025-02-16	12:45:59	Delivered	24342.52

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:2

6. List products that do not have any stock left

QUERY

```
SELECT * FROM PRODUCTS WHERE  
STOCK_QUANTITY = 0 OR  
STOCK_QUANTITY IS NULL;
```

OUTPUT

	product_id	name	category	price	stock_quantity	added_on
1	NULL	NULL	NULL	NULL	NULL	NULL

All products are in stock

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:3

1. Count the total number of orders placed

QUERY

```
SELECT COUNT(ORDER_ID) AS  
TOTAL_ORDER_PLACED FROM ORDERS;
```

OUTPUT

TOTAL_ORDER_PLACED
400

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:3

2. Calculate the total revenue collected from all orders

QUERY

```
SELECT SUM(TOTAL_AMOUNT) AS  
TOTAL_REVENUE FROM ORDERS;
```

-- ALL THE ORDERS(CANCELLED ALSO
INCLUDED)

OUTPUT

	TOTAL_REVENUE
▶	6960973.66

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:3

3. Calculate the average order value

QUERY

```
SELECT  
ROUND(AVG(TOTAL_AMOUNT),2) AS  
AVG_ORDER_VALUE FROM ORDERS;
```

OUTPUT

	AVG_ORDER_VALUE
▶	17402.43

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:3

4. Count the number of customers who have placed at least one order

QUERY

```
SELECT COUNT(CUSTOMER_ID) AS  
CUST_WITH_MIN_1_ORDER FROM  
CUSTOMERS WHERE CUSTOMER_ID IN  
(SELECT CUSTOMER_ID FROM ORDERS);
```

OUTPUT

	CUST_WITH_MIN_1_ORDER
▶	30

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:3

5. Find the number of orders placed by each customer

QUERY

```
SELECT CUSTOMER_ID,  
COUNT(ORDER_ID) AS 'NO. OF ORDER  
PLACED' FROM ORDERS GROUP BY  
CUSTOMER_ID;
```

OUTPUT

	CUSTOMER_ID	NO. OF ORDER PLACED
►	1	12
	2	17
	3	17
	4	9
	5	14
	6	16
	7	13
	8	10
	9	10

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:3

6. Find total sales amount made by each customer

QUERY

```
SELECT CUSTOMERS.NAME,  
ORDERS.CUSTOMER_ID AS  
CUSTOMER_ID,  
SUM(ORDERS.TOTAL_AMOUNT) AS  
AMOUNT_SPEND FROM ORDERS INNER  
JOIN CUSTOMERS ON  
CUSTOMERS.CUSTOMER_ID =  
ORDERS.CUSTOMER_ID GROUP BY  
CUSTOMER_ID;
```

OUTPUT

	NAME	CUSTOMER_ID	AMOUNT_SPEND
▶	Thomas Owens	1	183747.44
	Charles Grant	2	284420.07
	Kaitlin Richards	3	253783.31
	Christina Williams	4	137562.22
	David Allen	5	262504.19
	Mark Duke	6	212173.58
	Briana Wright	7	167960.11
	John Bryan	8	164701.78
	Jason Thompson	9	106226.03
	Shawn Hill	10	252722.75

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:3

7. List the number of products sold per category

QUERY

```
SELECT products.CATEGORY AS  
PRODUCT_CATEGORY,  
SUM(order_items.QUANTITY) AS  
SOLD_QUANTITYFROM products INNER  
JOIN order_items ON  
products.product_id =  
order_items.product_id GROUP BY  
CATEGORY;
```

OUTPUT

	PRODUCT_CATEGORY	SOLD_QUANTITY
▶	Home	443
	Clothing	559
	Electronics	687
	Toys	405
	Books	350

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:3

8. Find the average item price per category

QUERY

```
SELECT CATEGORY, avg(PRICE) AS  
AVG_PRICE FROM products group by  
category order by AVG_PRICE desc;
```

OUTPUT

	CATEGORY	AVG_PRICE
▶	Clothing	3434.581538
	Books	3167.084286
	Electronics	2653.388571
	Toys	2516.526250
	Home	2146.367500

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:3

9. Show number of orders placed per day

QUERY

```
SELECT dayname(order_date) AS DAY  
, count(order_id) AS "NO. OF ORDERS"  
FROM ORDERS group by DAY order by  
count(order_id) DESC;
```

OUTPUT

	DAY	NO. OF ORDERS
▶	Friday	65
	Wednesday	63
	Monday	61
	Tuesday	59
	Thursday	51
	Saturday	51
	Sunday	50

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:3

10. List total payments received per payment method

QUERY

```
SELECT METHOD AS  
PAYMENT_METHOD,  
COUNT(PAYMENT_ID) AS "NO. OF  
TIMES" FROM PAYMENTS GROUP BY  
METHOD ORDER BY  
COUNT(PAYMENT_ID) DESC;
```

OUTPUT

	PAYMENT_METHOD	NO. OF TIMES
►	Debit Card	113
	Net Banking	98
	Credit Card	96
	UPI	93

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:4

1. Retrieve order details along with the customer name (INNER JOIN)

QUERY

```
select customers.name , orders.* from  
customers inner join orders on  
customers.customer_id =  
orders.customer_id order by order_id;
```

OUTPUT

	name	order_id	customer_id	order_date	status	total_amount
▶	Megan Lee	1	20	2025-03-02 07:20:11	Delivered	9414.28
	Jeffrey Bray	2	18	2024-10-09 18:08:21	Shipped	532.20
	Austin Flores	3	15	2025-05-08 00:08:27	Cancelled	5164.56
	Walter Jenkins	4	11	2024-09-19 22:16:13	Delivered	9469.78
	Mary Knight	5	12	2025-04-08 18:02:06	Pending	14501.86
	Kara Zavala	6	29	2024-10-25 07:33:59	Cancelled	31050.17
	Peter Phillips	7	22	2024-07-29 11:58:47	Shipped	3043.67
	Amanda Bright	8	19	2024-07-30 22:49:49	Cancelled	32714.06
	Mark Duke	9	6	2025-06-10 17:00:25	Pending	24219.20
	Joseph Stuart	10	28	2025-02-16 12:45:59	Delivered	24342.52

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:4

2. Get list of products that have been sold (INNER JOIN with order_items)

QUERY

```
select distinct  
products.product_id,products.name,  
products.categoryfrom products  
inner join order_items on  
products.product_id =  
order_items.product_id;
```

OUTPUT

	product_id	name	category
▶	1	Plant No	Home
	2	Population Social	Clothing
	3	Available Answer	Electronics
	4	Any Question	Clothing
	5	Natural Network	Toys
	6	If Whatever	Electronics
	7	Response Indeed	Clothing
	8	Every Amount	Home
	9	Common Study	Toys
	10	Development System	Electronics

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:4

3. List all orders with their payment method (INNER JOIN)

QUERY

```
select orders.* , payments.method  
from orders inner join payments on  
orders.order_id = payments.order_id;
```

OUTPUT

	order_id	customer_id	order_date	status	total_amount	method
▶	1	20	2025-03-02 07:20:11	Delivered	9414.28	Credit Card
	2	18	2024-10-09 18:08:21	Shipped	532.20	Net Banking
	3	15	2025-05-08 00:08:27	Cancelled	5164.56	Credit Card
	4	11	2024-09-19 22:16:13	Delivered	9469.78	UPI
	5	12	2025-04-08 18:02:06	Pending	14501.86	UPI
	6	29	2024-10-25 07:33:59	Cancelled	31050.17	UPI
	7	22	2024-07-29 11:58:47	Shipped	3043.67	UPI
	8	19	2024-07-30 22:49:49	Cancelled	32714.06	Net Banking
	9	6	2025-06-10 17:00:25	Pending	24219.20	Net Banking
	10	28	2025-02-16 12:45:59	Delivered	24342.52	Debit Card

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:4

4. Get list of customers and their orders (LEFT JOIN)

QUERY

```
select customers.customer_id,  
customers.name, [REDACTED]  
group_concat(orders.order_id order  
by orders.order_id asc separator ',')  
as Order_Id_Listfrom customers left  
join orders on customers.customer_id  
= orders.customer_id group by  
customers.customer_id,  
customers.name;
```

OUTPUT

customer_id	name	Order_Id_List
1	Thomas Owens	14,17,61,76,92,109,127,135,144,221,278,379
2	Charles Grant	56,169,209,223,230,232,248,283,315,319,329,334,335,377,378,381,394
3	Kaitlin Richards	42,63,134,151,154,202,234,240,281,294,302,306,318,338,342,369,391
4	Christina Williams	78,87,149,188,274,290,328,352,386
5	David Allen	16,83,99,126,131,145,148,178,179,213,227,249,266,280
6	Mark Duke	9,24,50,96,106,172,180,194,261,271,307,330,333,361,375,383
7	Briana Wright	29,72,95,128,158,176,208,255,288,310,372,390,400
8	John Bryan	33,60,67,89,102,118,239,246,322,341
9	Jason Thompson	59,101,114,182,211,220,296,348,356,395
10	Shawn Hill	62,69,113,132,205,215,238,251,314,355,359,365,397

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:4

5. List all products along with order item quantity (LEFT JOIN)

QUERY

```
select p.product_id, p.name,  
sum(o.quantity) as Quantity_Sold  
from order_items as o left join  
products as p on o.product_id =  
p.product_id group by p.product_id,  
p.name order by Quantity_Sold desc;
```

OUTPUT

	product_id	name	Quantity_Sold
▶	24	Assume Serve	83
	46	Age Treatment	67
	18	Some Them	63
	47	Southern Thing	62
	8	Every Amount	61
	20	Real Source	60
	22	Move Small	60
	37	Between Up	58
	35	Fire Often	58
	17	Everything Plant	58

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:4

6. List all payments including those with no matching orders (RIGHT JOIN)

QUERY

```
select payments.* from orders right  
join payments on orders.order_id =  
payments.order_id;
```

OUTPUT

payment_id	order_id	payment_date	amount_paid	method
1	1	2025-03-02 08:16:11	9414.28	Credit Card
2	2	2024-10-09 19:00:21	532.20	Net Banking
3	3	2025-05-08 00:55:27	5164.56	Credit Card
4	4	2024-09-19 22:28:13	9469.78	UPI
5	5	2025-04-08 18:26:06	14501.86	UPI
6	6	2024-10-25 08:16:59	31050.17	UPI
7	7	2024-07-29 12:45:47	3043.67	UPI
8	8	2024-07-30 23:42:49	32714.06	Net Banking
9	9	2025-06-10 17:36:25	24219.20	Net Banking
10	10	2025-02-16 13:43:59	24342.52	Debit Card

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:4

7. Combine data from three tables: customer, order, and payment

QUERY

```
select c.customer_id, c.name,
o.order_id, o.order_date, 
o.total_amount, p.payment_id,
p.payment_date, p.amount_paid,
p.methodfrom customers as c inner
join orders as o on c.customer_id =
o.customer_id inner join payments as
p on o.order_id = p.order_id order by
o.order_id asc;
```

OUTPUT

customer_id	name	order_id	order_date	total_amount	payment_id	payment_date	amount_paid	method
20	Megan Lee	1	2025-03-02 07:20:11	9414.28	1	2025-03-02 08:16:11	9414.28	Credit Card
18	Jeffrey Bray	2	2024-10-09 18:08:21	532.20	2	2024-10-09 19:00:21	532.20	Net Banking
15	Austin Flores	3	2025-05-08 00:08:27	5164.56	3	2025-05-08 00:55:27	5164.56	Credit Card
11	Walter Jenkins	4	2024-09-19 22:16:13	9469.78	4	2024-09-19 22:28:13	9469.78	UPI
12	Mary Knight	5	2025-04-08 18:02:06	14501.86	5	2025-04-08 18:26:06	14501.86	UPI
29	Kara Zavala	6	2024-10-25 07:33:59	31050.17	6	2024-10-25 08:16:59	31050.17	UPI
22	Peter Phillips	7	2024-07-29 11:58:47	3043.67	7	2024-07-29 12:45:47	3043.67	UPI
19	Amanda Bright	8	2024-07-30 22:49:49	32714.06	8	2024-07-30 23:42:49	32714.06	Net Banking
6	Mark Duke	9	2025-06-10 17:00:25	24219.20	9	2025-06-10 17:36:25	24219.20	Net Banking
28	Joseph Stuart	10	2025-02-16 12:45:59	24342.52	10	2025-02-16 13:43:59	24342.52	Debit Card

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:5

1. List all products priced above the average product price

QUERY

```
select * from products where price >  
(select avg(price) from products);
```

OUTPUT

	product_id	name	category	price	stock_quantity	added_on
▶	2	Population Social	Clothing	4813.68	84	2025-05-30 10:02:50
	4	Any Question	Clothing	4759.28	179	2025-06-03 13:34:03
	5	Natural Network	Toys	4722.66	75	2023-11-06 00:47:37
	7	Response Indeed	Clothing	4897.36	36	2025-03-29 02:43:08
	8	Every Amount	Home	4173.60	156	2025-04-30 03:11:10
	10	Development System	Electronics	4801.78	153	2025-03-12 08:22:57
	12	Action Ask	Electronics	4017.01	19	2025-02-14 03:38:06
	18	Some Them	Toys	3673.86	110	2024-07-25 00:33:53
	19	Build High	Clothing	4707.14	47	2023-09-01 02:50:01
	20	Real Source	Books	4398.66	197	2025-02-08 10:28:27

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:5

2. Find customers who have placed at least one order

QUERY

```
select * from customers where  
customer_id in (select customer_id  
from orders group by customer_id  
having count(order_id) >=1);
```

OUTPUT

	customer_id	name	email	phone	created_at
▶	1	Thomas Owens	user1@example.com	142-479-1945	2024-10-14 16:01:12
	2	Charles Grant	user2@example.com	9153947511	2023-11-25 15:45:24
	3	Kaitlin Richards	user3@example.com	2073473421	2024-06-23 09:55:22
	4	Christina Williams	user4@example.com	586-605-5061x06	2024-10-27 17:19:38
	5	David Allen	user5@example.com	(751)456-8289x1	2023-10-29 02:43:00
	6	Mark Duke	user6@example.com	(144)957-2811	2024-06-24 03:22:59
	7	Briana Wright	user7@example.com	223-833-9635	2023-06-25 00:35:43
	8	John Bryan	user8@example.com	045.568.0798x27	2025-02-15 17:57:04
	9	Jason Thompson	user9@example.com	1862659420	2024-08-31 08:18:51
	10	Shawn Hill	user10@example.com	(268)113-3152x7	2023-12-14 20:46:43

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:5

3. Show orders whose total amount is above the average for that customer

QUERY

```
with t as (select customer_id ,  
avg(total_amount) as avg_amount  
from orders group by customer_id)  
select o.customer_id, o.order_id ,  
o.total_amount from orders as o inner  
join t on o.customer_id =  
t.customer_id where o.total_amount  
> t.avg_amount ;
```

OUTPUT

	customer_id	order_id	total_amount
▶	1	14	15803.34
	1	76	23506.81
	1	127	20160.64
	1	278	32015.16
	1	379	21586.89
	2	223	41020.75
	2	230	22655.32
	2	315	42056.04
	2	329	31387.01
	2	334	27857.86

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:5

4. Display customers who haven't placed any orders

QUERY

```
SELECT * FROM CUSTOMERS WHERE  
CUSTOMER_ID NOT IN (SELECT  
CUSTOMER_ID FROM ORDERS );
```

OUTPUT

	customer_id	name	email	phone	created_at
1	NULL	NULL	NULL	NULL	NULL

ALL CUSTOMERS HAVE PLACED ORDER

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:5

5. Show products that were never ordered

QUERY

```
select * from products where  
product_id not in (select product_id  
from order_items);
```

OUTPUT

	product_id	name	category	price	stock_quantity	added_on
1	NULL	NULL	NULL	NULL	NULL	NULL

ALL PRODUCTS HAS BEEN ORDERED

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:5

6. Show highest value order per customer

QUERY

```
select customer_id ,order_id,  
total_amount from orders as o1 where  
total_amount = (select  
max(total_amount) from orders as o2  
where o1.customer_id =  
o2.customer_id) order by  
customer_id;
```

OUTPUT

	customer_id	order_id	total_amount
▶	1	278	32015.16
	2	315	42056.04
	3	281	41679.11
	4	386	25747.34
	5	266	39921.78
	6	330	39003.19
	7	128	28589.04
	8	33	36147.09
	9	348	21414.44
	10	62	35723.17

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:5

7. Highest Order Per Customer (Including Names)

QUERY

```
with t as ( select customer_id,  
max(total_amount) as Highest_Order  
from orders as o2 group by  
customer_id )  
select c.customer_id, c.name,  
t.Highest_Order from customers as c  
join t on t.customer_id =  
c.customer_id;
```

OUTPUT

	customer_id	order_id	total_amount
▶	1	278	32015.16
	2	315	42056.04
	3	281	41679.11
	4	386	25747.34
	5	266	39921.78
	6	330	39003.19
	7	128	28589.04
	8	33	36147.09
	9	348	21414.44
	10	62	35723.17

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:6

1. List all customers who have either placed an order or written a product review

QUERY

```
select customer_id, name from  
customers where customer_id in  
(select customer_id from customers  
union select customer_id from  
product_reviews);
```

OUTPUT

	customer_id	name
▶	1	Thomas Owens
	2	Charles Grant
	3	Kaitlin Richards
	4	Christina Williams
	5	David Allen
	6	Mark Duke
	7	Briana Wright
	8	John Bryan
	9	Jason Thompson
	10	Shawn Hill

RISHU HRITHIK

MY SQL PROJECT

QUESTION

LEVEL:6

2. List all customers who have placed an order as well as reviewed a product [intersect not supported]

QUERY

```
select customer_id, name from  
customers where customer_id in  
(select customer_id from orders) and  
customer_id in (select customer_id  
from product_reviews);
```

OUTPUT

	customer_id	name
▶	1	Thomas Owens
	2	Charles Grant
	4	Christina Williams
	6	Mark Duke
	7	Briana Wright
	9	Jason Thompson
	10	Shawn Hill
	11	Walter Jenkins
	13	Leslie Wilson
	14	Deborah Arias

RISHU HRITHIK