PHARMACY MANAGEMENT SYSTEM

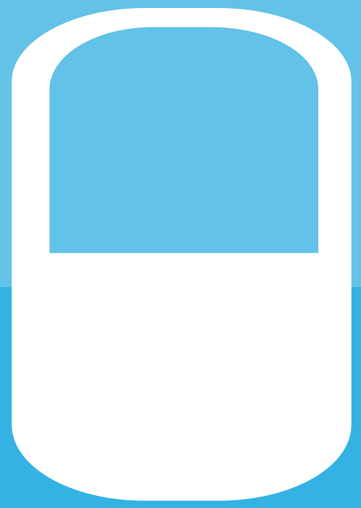
TEAM MEMBERS

Kamran Alam (BSE-23F-061)

Syed Abdul Uzair (BSE-23F-093)

Shahbaz Hanif (BSE-23F-076)

Content:



Overview:

- The Medical Store Management System (MSMS) helps manage all parts of running a medical store.

Main Features

INVENTORY MANAGEMENT
Keep track of medicines, their expiry dates, and stock levels

SUPPLIER MANAGEMENT
Store details of suppliers and their contacts.

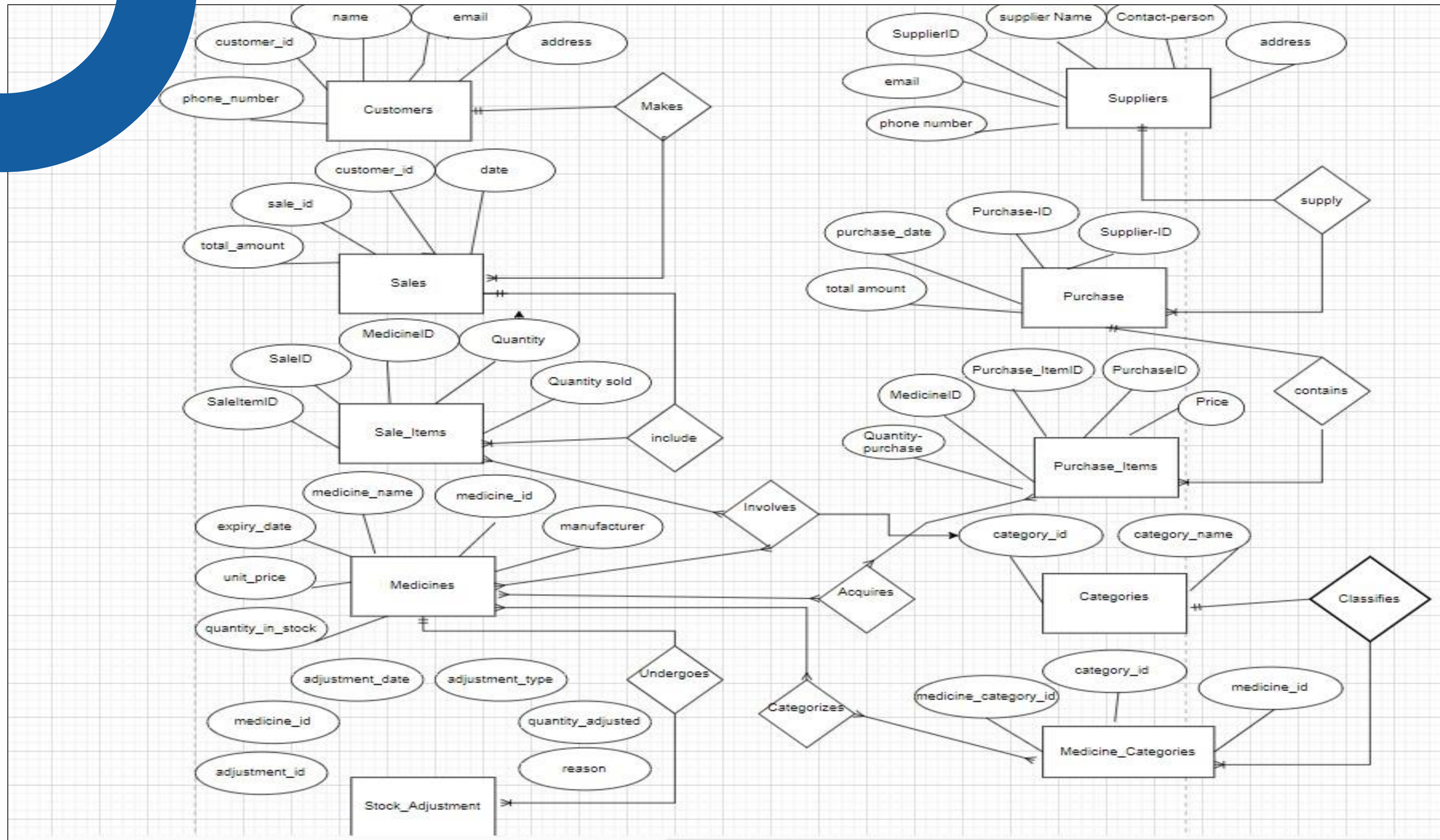
CUSTOMER MANAGEMENT
Organize customer information.

SALES AND PURCHASES
Record all sales and purchase transactions

STOCK ADJUSTMENTS
Log changes in stock with reasons.

CATEGORIZATION
Group medicines into different categories

ER-MODEL


name
email
customer_id
address
phone_number
Customers
Makes
customer_id
date
sale_id
total_amount
Sales
MedicineID
Quantity
SaleID
Quantity sold
SaleItemID
Sale_Items
include
medicine_name
medicine_id
expiry_date
manufacturer
unit_price
Medicines
quantity_in_stock
Involves
Acquires
adjustment_date
adjustment_type
Undergoes
medicine_id
quantity_adjusted
reason
adjustment_id
Stock_Adjustment
Categorizes
SupplierID
supplier Name
Contact-person
address
email
phone number
Suppliers
supply
Purchase-ID
purchase_date
Supplier-ID
total amount
Purchase
contains
Purchase_ItemID
PurchaseID
MedicineID
Price
Quantity-purchase
Purchase_Items
category_id
category_name
Categories
Classifies
category_id
medicine_category_id
medicine_id
Medicine_Categories

CONTRIBUTION OF EACH PARTNER

01

- **KAMRAN ALAM**
  - Tables & Queries + Project Lead + PPT

02

- **SYED ABDUL UZAIR**
  - Tables & Queries + Editing & Review

03

- **SHAHBAZ HANIF**
  - Tables & Queries + Entity-Relationship Model+ Upload On GitHub

QUERIES !

KAMRAN

```sql
SELECT COUNT(*) FROM Medicines;
SELECT SUM(quantity_in_stock) AS total_quantity FROM Medicines;
SELECT AVG(unit_price) AS average_price FROM Medicines;
SELECT MAX(unit_price) AS max_price FROM Medicines;
SELECT * FROM Customers WHERE customer_name LIKE 'A%';
SELECT MIN(unit_price) AS min_price FROM Medicines;
SELECT * FROM Medicines WHERE manufacturer = 'Bayer';
SELECT * FROM Sales WHERE sale_date BETWEEN '2024-05-01' AND '2024-05-31';
SELECT customer_id, SUM(total_amount) AS total_sales FROM Sales GROUP BY customer_id;
SELECT customer_id, COUNT(*) AS num_sales FROM Sales GROUP BY customer_id;
SELECT medicine_id, SUM(quantity_sold) AS total_quantity_sold FROM Sale_Items GROUP BY medicine_id;

SELECT m.medicine_id, m.medicine_name FROM Medicines m
LEFT JOIN Sale_Items si ON m.medicine_id = si.medicine_id
WHERE si.sale_item_id IS NULL;

SELECT p.supplier_id FROM Purchase_Items pi
JOIN Purchase p ON pi.purchase_id = p.purchase_id
GROUP BY p.supplier_id
HAVING SUM(pi.quantity_purchased) > 10;

SELECT * FROM Medicines ORDER BY unit_price DESC LIMIT 1;
SELECT * FROM Medicines ORDER BY unit_price ASC LIMIT 1;
SELECT DISTINCT SUBSTRING_INDEX(address, ',', -2) AS city FROM Customers;
```

QUERIES !

UZARI

```
204
205 •  SELECT AVG(num_purchases) AS avg_sales_per_customer
206    FROM (
207        SELECT c.customer_name, COUNT(s.sale_id) AS num_purchases
208        FROM Customers c
209        LEFT JOIN Sales s ON c.customer_id = s.customer_id
210        GROUP BY c.customer_name
211    ) AS customer_sales;
212
213 •  SELECT c.category_name, COUNT(mc.medicine_id) AS num_medicines
214    FROM Categories c
215    LEFT JOIN Medicine_Categories mc ON c.category_id = mc.category_id
216    GROUP BY c.category_name;
217
218 •  SELECT sa.medicine_id, m.medicine_name, COUNT(sa.adjustment_id) AS num_adjustments
219    FROM Stock_Adjustment sa
220    JOIN Medicines m ON sa.medicine_id = m.medicine_id
221    GROUP BY sa.medicine_id, m.medicine_name;
222
223 •  SELECT COUNT(DISTINCT category_id) AS num_distinct_categories FROM Categories;
224
```

```
180 •  SELECT pi.purchase_item_id, pi.purchase_id, pi.medicine_id, pi.quantity_purchased, pi.unit_price,
181           m.medicine_name, m.manufacturer, m.expiry_date, m.unit_price AS medicine_unit_price,
182           s.supplier_name, s.contact_person, s.phone_number, s.email
183    FROM Purchase_Items pi
184    RIGHT JOIN Medicines m ON pi.medicine_id = m.medicine_id
185    RIGHT JOIN Purchase p ON pi.purchase_id = p.purchase_id
186    RIGHT JOIN Suppliers s ON p.supplier_id = s.supplier_id;
187
188 •  SELECT m.medicine_name, m.manufacturer, m.expiry_date, m.unit_price,
189    pi.quantity_purchased, pi.unit_price AS purchase_unit_price
190    FROM Medicines m
191    LEFT JOIN Purchase_Items pi ON m.medicine_id = pi.medicine_id;
192
193 •  SELECT s.supplier_name, SUM(p.total_amount) AS total_purchased
194    FROM Suppliers s
195    JOIN Purchase p ON s.supplier_id = p.supplier_id
196    GROUP BY s.supplier_name
197    ORDER BY total_purchased DESC
198    LIMIT 3;
199
200 •  SELECT c.customer_name, COUNT(s.sale_id) AS num_purchases, SUM(s.total_amount) AS total_spent
201    FROM Customers c
202    LEFT JOIN Sales s ON c.customer_id = s.customer_id
203    GROUP BY c.customer_name;
```

QUERIES !

UZARI

```sql
-- syed abdul uzair

SELECT c.customer_name, SUM(s.total_amount) AS total_spent
FROM Customers c
JOIN Sales s ON c.customer_id = s.customer_id
GROUP BY c.customer_name;

SELECT COUNT(DISTINCT supplier_id) AS num_distinct_suppliers FROM Purchase;

SELECT medicine_id, COUNT(*) AS num_adjustments FROM Stock_Adjustment
GROUP BY medicine_id HAVING COUNT(*) > 1;

SELECT m.medicine_name, SUM(pi.quantity_purchased) AS total_quantity_purchased
FROM Medicines m
JOIN Purchase_Items pi ON m.medicine_id = pi.medicine_id
GROUP BY m.medicine_name;

SELECT s.sale_id, c.customer_name, m.medicine_name, si.quantity_sold, si.unit_price
FROM Sales s
JOIN Customers c ON s.customer_id = c.customer_id
JOIN Sale_Items si ON s.sale_id = si.sale_id
JOIN Medicines m ON si.medicine_id = m.medicine_id;
```

QUERIES !

SHAHBAZ

```sql
-- Shabaz Hanif

SELECT SUBSTRING_INDEX(address, ',', -2) AS city, COUNT(*) AS num_suppliers
FROM Suppliers
GROUP BY city;

SELECT s.supplier_name, m.medicine_name FROM Suppliers s
JOIN Purchase p ON s.supplier_id = p.supplier_id
JOIN Purchase_Items pi ON p.purchase_id = pi.purchase_id
JOIN Medicines m ON pi.medicine_id = m.medicine_id;

SELECT medicine_id, COUNT(*) AS num_adjustments FROM Stock_Adjustment GROUP BY medicine_id;

SELECT medicine_id, reason FROM Stock_Adjustment ORDER BY adjustment_date DESC LIMIT 1;

SELECT m.medicine_name, c.category_name FROM Medicines m
JOIN Medicine_Categories mc ON m.medicine_id = mc.medicine_id
JOIN Categories c ON mc.category_id = c.category_id;

SELECT medicine_id, SUM(CASE WHEN adjustment_type = 'Addition' THEN quantity_adjusted ELSE 0 END) AS total_added,
SUM(CASE WHEN adjustment_type = 'Subtraction' THEN quantity_adjusted ELSE 0 END) AS total_subtracted
FROM Stock_Adjustment GROUP BY medicine_id;

```

QUERIES !

SHAHBAZ

```sql
SELECT DISTINCT c.customer_name FROM Customers c
JOIN Sales s ON c.customer_id = s.customer_id
JOIN Sale_Items si ON s.sale_id = si.sale_id
JOIN Medicines m ON si.medicine_id = m.medicine_id
WHERE m.medicine_name = 'Aspirin';

SELECT s.sale_id, c.customer_name, s.sale_date, s.total_amount
FROM Sales s
JOIN Customers c ON s.customer_id = c.customer_id;

SELECT AVG(quantity_purchased) AS avg_quantity FROM Purchase_Items;

SELECT DISTINCT s.supplier_name FROM Suppliers s
JOIN Purchase p ON s.supplier_id = p.supplier_id
JOIN Purchase_Items pi ON p.purchase_id = pi.purchase_id
JOIN Medicines m ON pi.medicine_id = m.medicine_id
WHERE m.medicine_name = 'Ibuprofen';

SELECT p.purchase_id, s.supplier_name, p.purchase_date, p.total_amount
FROM Purchase p
JOIN Suppliers s ON p.supplier_id = s.supplier_id;

SELECT COUNT(DISTINCT medicine_id) AS num_distinct_medicines_sold FROM Sale_Items;

SELECT s.sale_id, SUM(si.quantity_sold * si.unit_price) AS total_sale_amount
FROM Sales s
JOIN Sale_Items si ON s.sale_id = si.sale_id
GROUP BY s.sale_id;
```

```sql
SELECT m.medicine_name, SUM(si.quantity_sold) AS total_quantity_sold
FROM Medicines m
JOIN Sale_Items si ON m.medicine_id = si.medicine_id
GROUP BY m.medicine_name
ORDER BY total_quantity_sold DESC
LIMIT 3;

SELECT COUNT(DISTINCT customer_id) AS num_distinct_customers FROM Sales;

SELECT m.medicine_name FROM Medicines m
JOIN Medicine_Categories mc ON m.medicine_id = mc.medicine_id
JOIN Categories c ON mc.category_id = c.category_id
WHERE c.category_name = 'Pain Relief';
```

QUERIES !

SHAHBAZ

QUERIES - 076

1st Query with Output

```
SELECT SUBSTRING_INDEX(address, ',', -2) AS city, COUNT(*) AS num_suppliers
FROM Suppliers
GROUP BY city;
```

city	num_suppliers
123 Main St, CityA	1
456 Elm St, CityB	1

2st Query with Output

```
SELECT medicine_id, COUNT(*) AS num_adjustments FROM Stock_Adjustment GROUP BY medicine_id;
```

medicine_id	num_adjustments
1	1
3	1

3 Query with Output

```
SELECT s.supplier_name, m.medicine_name FROM Suppliers s
JOIN Purchase p ON s.supplier_id = p.supplier_id
JOIN Purchase_Items pi ON p.purchase_id = pi.purchase_id
JOIN Medicines m ON pi.medicine_id = m.medicine_id;
```

supplier_name	medicine_name
Supplier A	Paracetamol
Supplier A	Betadine
Supplier B	Amoxicillin
Supplier B	Vitamin C

4 Query with Output

```
SELECT m.medicine_name, c.category_name FROM Medicines m
JOIN Medicine_Categories mc ON m.medicine_id = mc.medicine_id
JOIN Categories c ON mc.category_id = c.category_id;
```

medicine_name	category_name
Paracetamol	Analgesics
Amoxicillin	Antibiotics
Betadine	Antiseptics
Vitamin C	Vitamins
Ibuprofen	Analgesics

QUERIES - 076

5 Query with Output

```
SELECT medicine_id, SUM(CASE WHEN adjustment_type = 'Addition' THEN quantity_adjusted ELSE 0 END) AS total_added,
SUM(CASE WHEN adjustment_type = 'Subtraction' THEN quantity_adjusted ELSE 0 END) AS total_subtracted
FROM Stock_Adjustment GROUP BY medicine_id;
```

medicine_id	total_added	total_subtracted
1	50	0
3	0	10

6 Query with Output

```
SELECT s.sale_id, c.customer_name, s.sale_date, s.total_amount
FROM Sales s
JOIN Customers c ON s.customer_id = c.customer_id;
```

sale_id	customer_name	sale_date	total_amount
1	Customer One	2024-05-01 10:00:00	15.00
2	Customer Two	2024-05-02 14:30:00	20.00

7 Query with Output

```
SELECT DISTINCT c.customer_name FROM Customers c
JOIN Sales s ON c.customer_id = s.customer_id
JOIN Sale_Items si ON s.sale_id = si.sale_id
JOIN Medicines m ON si.medicine_id = m.medicine_id
WHERE m.medicine_name = 'Aspirin';
```

customer_name

8 Query with Output

```
SELECT medicine_id, reason FROM Stock_Adjustment ORDER BY adjustment_date DESC LIMIT 1;
```

medicine_id	reason
3	Expired stock

QUERIES - 076

9 Query with Output

```
SELECT AVG(quantity_purchased) AS avg_quantity FROM Purchase_Items;
```

avg_quantity
175.0000

10 Query with Output

```
SELECT p.purchase_id, s.supplier_name, p.purchase_date, p.total_amount
FROM Purchase p
JOIN Suppliers s ON p.supplier_id = s.supplier_id;
```

purchase_id	supplier_name	purchase_date	total_amount
1	Supplier A	2024-04-20 09:00:00	500.00
2	Supplier B	2024-04-25 11:15:00	300.00

11 Query with Output

```
SELECT DISTINCT s.supplier_name FROM Suppliers s
JOIN Purchase p ON s.supplier_id = p.supplier_id
JOIN Purchase_Items pi ON p.purchase_id = pi.purchase_id
JOIN Medicines m ON pi.medicine_id = m.medicine_id
WHERE m.medicine_name = 'Ibuprofen';
```

supplier_name

12 Query with Output

```
SELECT COUNT(DISTINCT medicine_id) AS num_distinct_medicines_sold FROM Sale_Items;
```

num_distinct_medicines_sold
4

QUERIES - 076

13 Query with Output

```
SELECT s.sale_id, SUM(si.quantity_sold * si.unit_price) AS total_sale_amount
FROM Sales s
JOIN Sale_Items si ON s.sale_id = si.sale_id
GROUP BY s.sale_id;
```

sale_id	total_sale_amount
1	20.00
2	21.00

14 Query with Output

```
SELECT m.medicine_name, SUM(si.quantity_sold) AS total_quantity_sold
FROM Medicines m
JOIN Sale_Items si ON m.medicine_id = si.medicine_id
GROUP BY m.medicine_name
ORDER BY total_quantity_sold DESC
LIMIT 3;
```

medicine_name	total_quantity_sold
Vitamin C	20
Paracetamol	10
Ibuprofen	2

15 Query with Output

```
SELECT COUNT(DISTINCT customer_id) AS num_distinct_customers FROM Sales;
```

num_distinct_customers
2

16 Query with Output

```
SELECT m.medicine_name FROM Medicines m
JOIN Medicine_Categories mc ON m.medicine_id = mc.medicine_id
JOIN Categories c ON mc.category_id = c.category_id
WHERE c.category_name = 'Pain Relief';
```

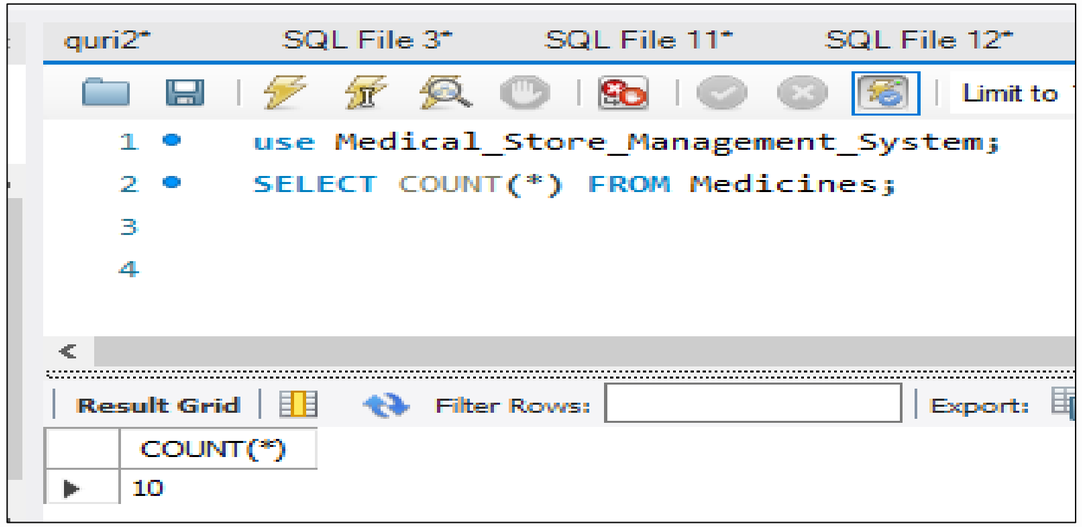
medicine_name

QUERIES !

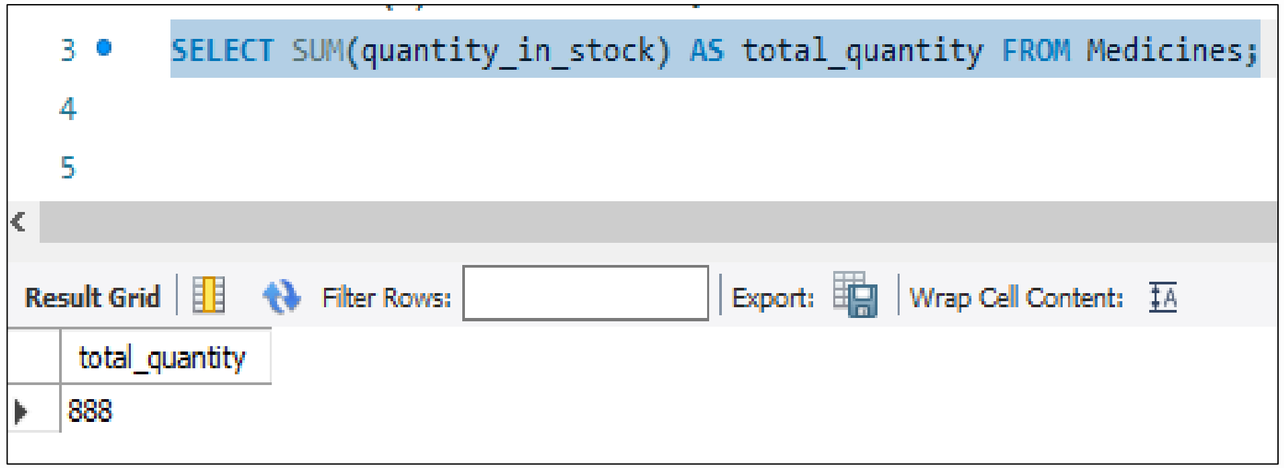
KAMRAN

QUERIES - 061

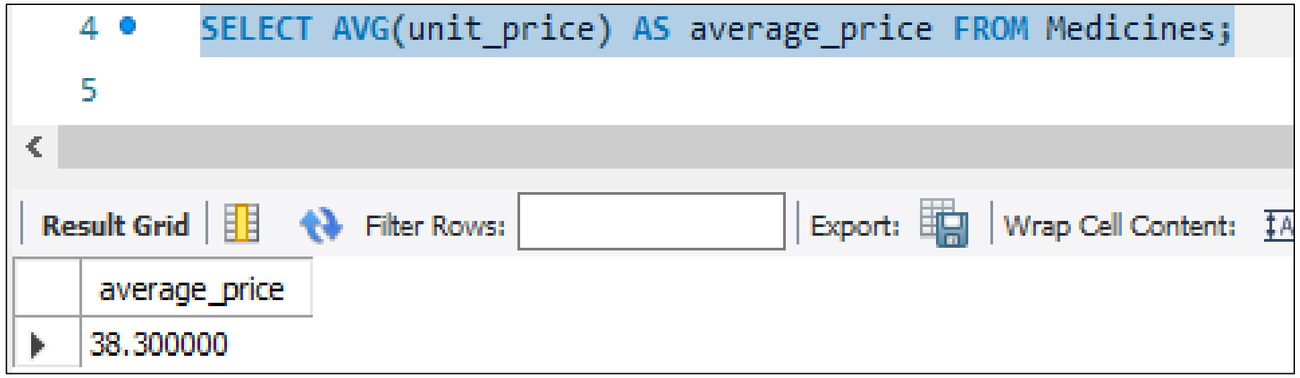
1 Query with Output



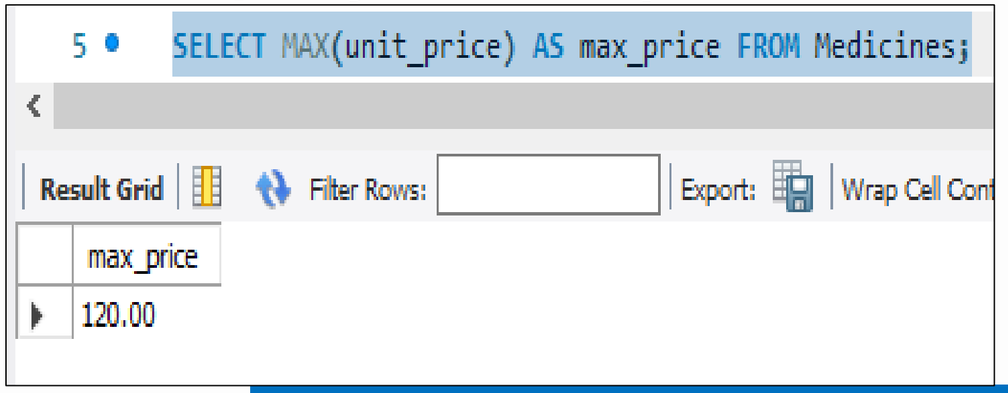
2 Query with Output



3 Query with Output



4 Query with Output



QUERIES - 061

5 Query with Output

```
6 • SELECT * FROM Customers WHERE customer_name LIKE 'A%';
```

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

customer_id	customer_name	phone_number	email	address
4	Ayesha Khan	03232678901	ayeshakhan300@gmail.com	456 Oak Avenue, LAHORE, Pakistan
NULL	NULL	NULL	NULL	NULL

6 Query with Output

```
7 • SELECT MIN(unit_price) AS min_price FROM Medicines;
```

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

min_price
0.75

7 Query with Output

```
7 • SELECT * FROM Medicines WHERE manufacturer = 'Bayer';
```

Result Grid | Filter Rows: | Edit: | Export/Import:

medicine_id	medicine_name	manufacturer	expiry_date	unit_price	quantity_in_sto
6	Aspirin	Bayer	2024-06-30	50.00	20
7	pandol	Bayer	2026-04-25	50.00	20
NULL	NULL	NULL	NULL	NULL	NULL

8 Query with Output

```
9 • SELECT * FROM Sales WHERE sale_date BETWEEN '2024-05-01' AND '2024-05-31';
```

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

sale_id	customer_id	sale_date	total_amount
1	1	2024-05-01 10:00:00	15.00
2	2	2024-05-02 14:30:00	20.00
NULL	NULL	NULL	NULL

QUERIES - 061

9 Query with Output

```
14  SELECT m.medicine_id, m.medicine_name FROM Medicines m
15  LEFT JOIN Sale_Items si ON m.medicine_id = si.medicine_id
16  WHERE si.sale_item_id IS NULL;
```

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

medicine_id	medicine_name
3	Betadine
6	Aspirin
7	pandol
8	Ibuprofen
9	Amoxicillin
10	Paracetamol

10 Query with Output

```
10  SELECT customer_id, SUM(total_amount) AS total_sales FROM Sales GROUP BY customer_id;
```

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

customer_id	total_sales
1	15.00
2	20.00

11 Query with Output

```
12  SELECT medicine_id, SUM(quantity_sold) AS total_quantity_sold FROM Sale_Items GROUP BY medicine_id;
```

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

medicine_id	total_quantity_sold
1	10
2	1
4	20
5	2

12 Query with Output

```
11  SELECT customer_id, COUNT(*) AS num_sales FROM Sales GROUP BY customer_id;
```

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

customer_id	num_sales
1	1
2	1

QUERIES - 061

13 Query with Output

```
25 •  SELECT DISTINCT SUBSTRING_INDEX(address, ',', -2) AS city FROM Customers;
26
```

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

city
789 Pine St, CityC
101 Maple St, CityD
BALOCHISTAN, Pakistan
LAHORE, Pakistan
KARACHI, Pakistan
KPK, Pakistan

14 Query with Output

```
18 •  SELECT p.supplier_id FROM Purchase_Items pi
19    JOIN Purchase p ON pi.purchase_id = p.purchase_id
20    GROUP BY p.supplier_id
21    HAVING SUM(pi.quantity_purchased) > 10;
22
```

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

supplier_id
1
2

15 Query with Output

```
24 •  SELECT * FROM Medicines ORDER BY unit_price ASC LIMIT 1;
25
```

Result Grid | Filter Rows: | Edit: | Export/Import:

medicine_id	medicine_name	manufacturer	expiry_date	unit_price	quantity_in_stock
9	Amoxicillin	GSK	2025-03-31	120.00	8
NULL	NULL	NULL	NULL	NULL	NULL

16 Query with Output

```
23 •  SELECT * FROM Medicines ORDER BY unit_price DESC LIMIT 1;
24
```

Result Grid | Filter Rows: | Edit: | Export/Import:

medicine_id	medicine_name	manufacturer	expiry_date	unit_price	quantity_in_stock
9	Amoxicillin	GSK	2025-03-31	120.00	8
NULL	NULL	NULL	NULL	NULL	NULL

QUERIES !

UZAIR

QUERIES - 093

1 Query with Output

```
170 •  SELECT c.customer_name, SUM(s.total_amount) AS total_spent FROM Customers c
171    JOIN Sales s ON c.customer_id = s.customer_id GROUP BY c.customer_name;
```

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

customer_name	total_spent
Shabu Baloch	120.00
Ayesha Khan	150.00
Uzair	180.00
Kamran Alam	200.00
Sara Khan	220.00

2 Query with Output

```
174 •  SELECT COUNT(DISTINCT supplier_id) AS num_distinct_suppliers FROM Purchase;
```

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

num_distinct_suppliers
5

3 Query with Output

```
180 •  SELECT m.medicine_name, SUM(pi.quantity_purchased) AS total_quantity_purchased
181    FROM Medicines m JOIN Purchase_Items pi ON m.medicine_id = pi.medicine_id
182    GROUP BY m.medicine_name;
```

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

medicine_name	total_quantity_purchased
Aspirin	10
pandol	5
Ibuprofen	8
Amoxicillin	3
Paracetamol	12

4 Query with Output

```
186 •  SELECT s.sale_id, c.customer_name, m.medicine_name, si.quantity_sold, si.unit_price
187    FROM Sales s JOIN Customers c ON s.customer_id = c.customer_id
188    JOIN Sale_Items si ON s.sale_id = si.sale_id
189    JOIN Medicines m ON si.medicine_id = m.medicine_id;
```

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

sale_id	customer_name	medicine_name	quantity_sold	unit_price
1	Shabu Baloch	Aspirin	2	25.00
1	Shabu Baloch	pandol	1	40.00
2	Ayesha Khan	Ibuprofen	3	30.00
2	Ayesha Khan	Amoxicillin	2	50.00
3	Uzair	Paracetamol	4	35.00

QUERIES - 093

5 Query with Output

```
195 • SELECT pi.purchase_item_id, pi.purchase_id, pi.medicine_id, pi.quantity_purchased, pi.unit_price,
196       m.medicine_name, m.manufacturer, m.expiry_date, m.unit_price AS medicine_unit_price,
197       s.supplier_name, s.contact_person, s.phone_number, s.email
198   FROM Purchase_Items pi
199   RIGHT JOIN Medicines m ON pi.medicine_id = m.medicine_id
200   RIGHT JOIN Purchase p ON pi.purchase_id = p.purchase_id
201   RIGHT JOIN Suppliers s ON p.supplier_id = s.supplier_id;
202
```

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

purchase_item_id	purchase_id	medicine_id	quantity_purchased	unit_price	medicine_name	manufacturer	expiry_date	medicine_unit_price
1	1	1	10	20.00	Aspirin	Bayer	2024-06-30	50.00
2	2	2	5	30.00	pandol	Bayer	2026-04-25	50.00
3	3	3	8	25.00	Ibuprofen	Pfizer	2024-05-25	80.00
4	4	4	3	35.00	Amoxicillin	GSK	2025-03-31	120.00
5	5	5	12	15.00	Paracetamol	Johnson & Johnson	2025-12-31	70.00

supplier_name	contact_person	phone_number	email
XYZ Pharmaceuticals	subhan	03134543675	subhan034@gmail.com
MediCorp	haris	0332432543	haris567@gmail.com
HealthCare Suppliers	haseeb	03342353257	haseeb353@gmail.com
Pharma Solutions Ltd	hamza	03443322110	hamza532@gmail.com
Global Generics	haseeb	03777888999	haseeb334@gmail.com

6 Query with Output

```
223 • SELECT s.supplier_name, SUM(p.total_amount) AS total_purchased
224   FROM Suppliers s
225   JOIN Purchase p ON s.supplier_id = p.supplier_id
226   GROUP BY s.supplier_name
227   ORDER BY total_purchased DESC
228   LIMIT 3;
```

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

supplier_name	total_purchased
Global Generics	900.00
Pharma Solutions Ltd	800.00
HealthCare Suppliers	700.00

7 Query with Output

```
212 • SELECT m.medicine_name, m.manufacturer, m.expiry_date, m.unit_price,
213   pi.quantity_purchased, pi.unit_price AS purchase_unit_price
214   FROM Medicines m
215   LEFT JOIN Purchase_Items pi ON m.medicine_id = pi.medicine_id;
```

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

medicine_name	manufacturer	expiry_date	unit_price	quantity_purchased	purchase_unit_price
Aspirin	Bayer	2024-06-30	50.00	10	20.00
pandol	Bayer	2026-04-25	50.00	5	30.00
Ibuprofen	Pfizer	2024-05-25	80.00	8	25.00
Amoxicillin	GSK	2025-03-31	120.00	3	35.00
Paracetamol	Johnson & Johnson	2025-12-31	70.00	12	15.00

QUERIES - 093

8 Query with Output

```
236 •  SELECT c.customer_name, COUNT(s.sale_id) AS num_purchases, SUM(s.total_amount) AS total_spent
237    FROM Customers c
238    LEFT JOIN Sales s ON c.customer_id = s.customer_id
239    GROUP BY c.customer_name;
```

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

customer_name	num_purchases	total_spent
Shabu Baloch	1	120.00
Ayesha Khan	1	150.00
Uzair	1	180.00
Kamran Alam	1	200.00
Sara Khan	1	220.00

9 Query with Output

```
243 •  SELECT AVG(num_purchases) AS avg_sales_per_customer
244    FROM (
245        SELECT c.customer_name, COUNT(s.sale_id) AS num_purchases
246        FROM Customers c
247        LEFT JOIN Sales s ON c.customer_id = s.customer_id
248        GROUP BY c.customer_name
249    ) AS customer_sales;
```

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

avg_sales_per_customer
1.0000

10 Query with Output

```
251 •  SELECT c.category_name, COUNT(mc.medicine_id) AS num_medicines
252    FROM Categories c
253    LEFT JOIN Medicine_Categories mc ON c.category_id = mc.category_id
254    GROUP BY c.category_name;
```

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

category_name	num_medicines
Pain Relief	1
Antibiotics	1
Digestive Health	1
Cardiovascular	1
Respiratory	1

11 Query with Output

```
260 •  SELECT sa.medicine_id, m.medicine_name, COUNT(sa.adjustment_id) AS num_adjustments
261    FROM Stock_Adjustment sa
262    JOIN Medicines m ON sa.medicine_id = m.medicine_id
263    GROUP BY sa.medicine_id, m.medicine_name;
```

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

medicine_id	medicine_name	num_adjustments
1	Aspirin	1
2	pandol	1
3	Ibuprofen	1

QUERIES - 093

12 Query with Output

```
265 •  SELECT COUNT(DISTINCT category_id) AS num_distinct_categories FROM Categories;
```

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

num_distinct_categories
5

13 Query with Output

```
177 •  SELECT medicine_id, COUNT(*) AS num_adjustments FROM Stock_Adjustment
178    GROUP BY medicine_id HAVING COUNT(*) > 1;
```

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

medicine_id	num_adjustments

THANK YOU!