DATABASE ASSIGNMENT

GROUP MEMBERS

NISMAH AKHLAS (BAI-23S-029)

AMMARA ASIF (BAI-23S-078)

ABDULLAH YAQOOB (BAI-23S-044)

QUERIES:

1- SELECT o.order\_id, p.product\_name, p.price

FROM orders o

JOIN order\_details od ON o.order\_id = od.order\_id

JOIN products p ON od.product\_id = p.product\_id

WHERE o.customer\_id = 'specific\_customer\_id';

2- SELECT product\_name, category\_name

FROM Products

JOIN Categories ON Products.category\_id = Categories.category\_id

WHERE category\_name = 'Vegetables';

3-SELECT SUM(quantity) AS total\_items\_in\_stock

FROM your\_table\_name;

4-SELECT \*

FROM orders

WHERE order\_date >= DATE\_SUB(CURDATE(), INTERVAL 1 WEEK);

5-SELECT o.order\_id, p.product\_name, p.price

FROM orders o

JOIN order\_details od ON o.order\_id = od.order\_id

JOIN products p ON od.product\_id = p.product\_id

WHERE o.customer\_id = 'specific\_customer\_id';

6- ALTER TABLE Products

ADD COLUMN weight DECIMAL(10,2);

7-SELECT Employees.employee\_name, SUM(Orders.total\_amount) AS total\_sales

FROM Employees

JOIN Orders ON Employees.employee\_id = Orders.employee\_id

GROUP BY Employees.employee\_id;

8-SELECT Products.product\_name, Categories.category\_name

FROM Products

JOIN Categories ON Products.category\_id = Categories.category\_id

WHERE Products.expiration\_date < CURRENT\_DATE;

9-SELECT SUM(total\_amount) AS total\_sales

FROM Orders;

10-UPDATE Categories

SET category\_name = 'Fresh Vegetables'

WHERE category\_id = 'C002';

11-UPDATE Employees

SET salary = 55000

WHERE employee\_id = 'EMP001';

12- SELECT \*

FROM products

WHERE price = (SELECT MAX(price) FROM products);

13- Alter table order\_details

Add constraints

Foreign key(product\_id)

References products (product\_id)

14- SELECT customer\_id, customer\_name

FROM Customers

WHERE customer\_id IN (SELECT customer\_id FROM Orders GROUP BY customer\_id HAVING COUNT(\*) > 1);

15-SELECT product\_id, product\_name

FROM Products

WHERE category\_id IN (SELECT category\_id FROM Categories WHERE stock\_quantity < 10);

16-SELECT customer\_id, customer\_name, total\_amount

FROM Orders

WHERE total\_amount > (SELECT AVG(total\_amount) FROM Orders);

17-SELECT product\_name, unit\_price

FROM Products

WHERE unit\_price > (SELECT AVG(unit\_price) FROM Products WHERE category\_id = 'CATEGORY\_ID');

18- ALTER TABLE Products

MODIFY COLUMN product\_name VARCHAR(50) NOT NULL;

19- SELECT Suppliers.supplier\_id, Suppliers.supplier\_name, SUM(Products.unit\_price \* Products.stock\_quantity) AS total\_revenue

FROM Suppliers

LEFT JOIN Products ON Suppliers.supplier\_id = Products.supplier\_id

GROUP BY Suppliers.supplier\_id;

20- SELECT Categories.category\_name, AVG(Products.unit\_price) AS avg\_price

FROM Products

JOIN Categories ON Products.category\_id = Categories.category\_id

GROUP BY Categories.category\_name;

21-SELECT \*

FROM Customers

WHERE customer\_id IN (SELECT customer\_id

FROM Orders

GROUP BY customer\_id

HAVING COUNT(order\_id) > 2);

22- ALTER TABLE Products

MODIFY COLUMN expiration\_date DATE NOT NULL;

23- SELECT category\_id, category\_name

FROM Categories

WHERE category\_id = (

SELECT category\_id

FROM Products

GROUP BY category\_id

ORDER BY SUM(stock\_quantity) ASC

LIMIT 1);

24-SELECT \*

FROM Customers

WHERE customer\_id IN (

SELECT DISTINCT customer\_id

FROM Orders

WHERE order\_date >= DATE\_SUB(CURDATE(), INTERVAL 1 MONTH) );

25-SELECT product\_id, product\_name

FROM Products

WHERE unit\_price > (SELECT AVG(unit\_price) FROM Products);

26- SELECT customer\_id, MAX(order\_date) AS latest\_order\_date

FROM Orders

GROUP BY customer\_id;

27-SELECT \*

FROM Products

WHERE product\_id IN (

SELECT product\_id

FROM Order\_details

GROUP BY product\_id

HAVING COUNT(\*) > 1);

28-UPDATE Customers

SET loyalty\_points = (

SELECT SUM(total\_amount) / 10

FROM Orders

WHERE Orders.customer\_id = Customers.customer\_id);

29- UPDATE Products

SET expiration\_date = '2024-12-31'

WHERE product\_id = 'P002';

30- DELETE FROM Products WHERE product\_id = 'P001';

31- SELECT Suppliers.supplier\_id, Suppliers.supplier\_name, COUNT(Products.product\_id) AS out\_of\_stock\_products

FROM Suppliers

LEFT JOIN Products ON Suppliers.supplier\_id = Products.supplier\_id

WHERE Products.stock\_quantity = 0

GROUP BY Suppliers.supplier\_id;

32- SELECT Categories.category\_name, SUM(Products.stock\_quantity) AS total\_stock

FROM Categories

LEFT JOIN Products ON Categories.category\_id = Products.category\_id

GROUP BY Categories.category\_name;

33- SELECT supplier\_id, supplier\_name,

(SELECT SUM(stock\_quantity)

FROM Products

WHERE supplier\_id = Suppliers.supplier\_id) AS total\_stock

FROM Suppliers;

34- SELECT \*

FROM Products

WHERE stock\_quantity >

(SELECT AVG(stock\_quantity)

FROM Products);

35- SELECT Products.product\_name, Categories.category\_name

FROM Products

JOIN Categories ON Products.category\_id = Categories.category\_id;

36- UPDATE Products

SET unit\_price = (

SELECT AVG(unit\_price)

FROM Products

WHERE category\_id = Products.category\_id)

WHERE category\_id = 'CAT001';

37- SELECT customer\_id, COUNT(\*) AS order\_count

FROM Orders

GROUP BY customer\_id;

38- ALTER TABLE Customers

CHANGE COLUMN email customer\_email VARCHAR(100);

39- ALTER TABLE Products

MODIFY COLUMN expiration\_date DATE NOT NULL;

40- SELECT product\_id, product\_name, stock\_quantity

FROM Products

WHERE stock\_quantity < 20;