CASE STUDY - MANAGING A ONLINE BOOKSTORE DATABASE

INTRODUCTION:

Managing an online bookstore database involves organizing, storing, and retrieving data related to books, customers, orders, and order details in a structured and efficient manner. A well-designed database ensures seamless operations within an online bookstore, facilitating inventory management, sales tracking, customer relationship management, and overall business analysis. This guide provides an overview of the essential components and best practices for managing an online bookstore database using SQL.

Book Store

PROBLEM STATEMENT:

An online bookstore needs an efficient and robust database management system to handle its daily business operations. The bookstore sells a wide variety of books and manages numerous customer accounts and orders every day. To streamline its operations and provide an excellent customer experience, the bookstore requires a well-organized database that efficiently manages inventory, customer information, orders, and order details.

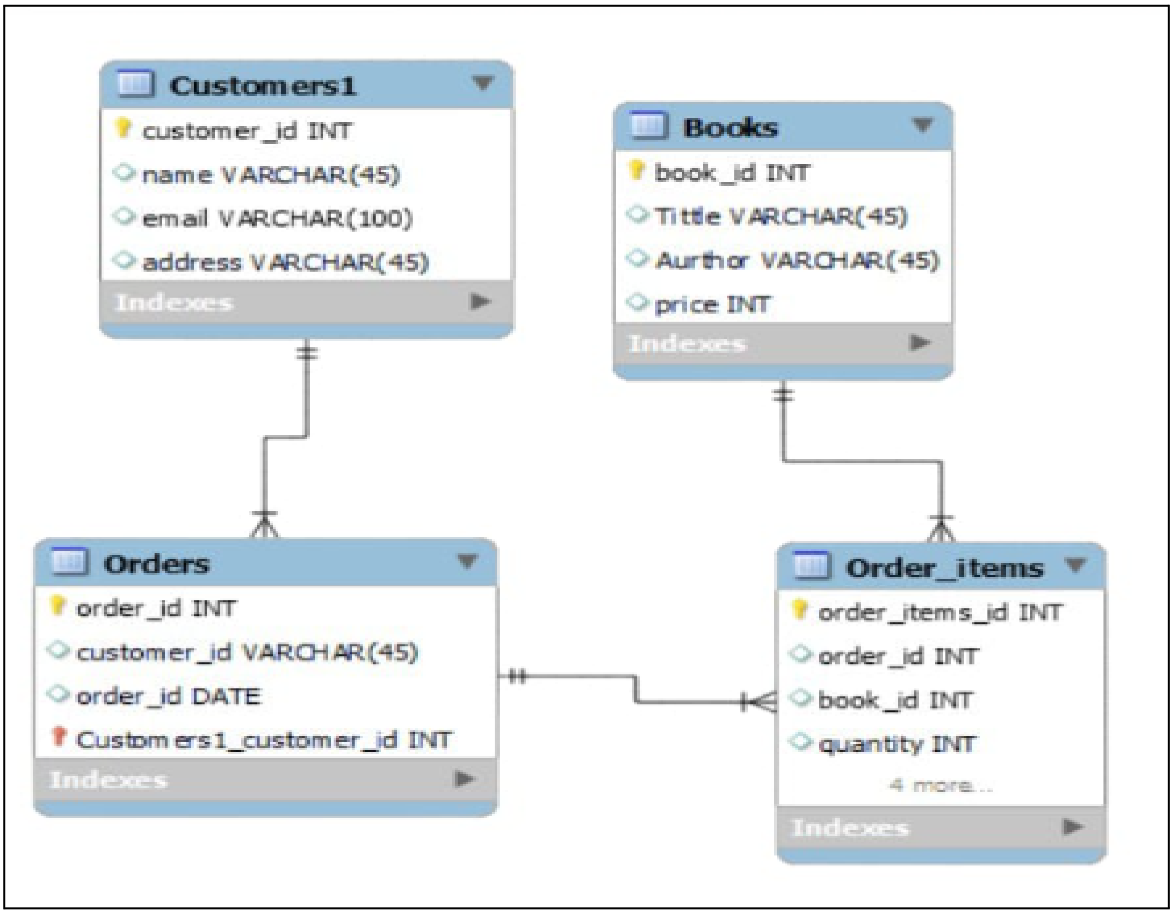
OBJECTIVE:

Develop and manage an online bookstore database using SQL that will organize and streamline the bookstore's operations. The database should include tables for books, customers, orders, and order items, ensuring data integrity and facilitating efficient data retrieval and updates.

The 4 key datasets to be used in the case study are:

- Books
- Customers1
- Orders
- Order_items

ENTITY RELATIONSHIP DIAGRAM



DATASET:

```
CREATE DATABASE Book_store;
USE  Book_store;


SELECT*FROM Books;

SELECT*FROM Customers1;

SELECT*FROM Orders;

SELECT*FROM Order_it;
```

Books Table

```
CREATE TABLE Books(
Book_id int,
Tittle varchar(35),
Author varchar(35),
Price int
);

INSERT INTO Books (Book_id,Tittle, Author, Price)
 VALUES(1, 'To Kill a Mockingbird', 'Harper Lee', 10.99),
(2, '1984', 'George Orwell', 8.99),
(3, 'The Great Gatsby', 'F. Scott Fitzgerald', 9.99),
(4, 'The Catcher in the Rye', 'J.D. Salinger', 10.50),
(5, 'The Hobbit', 'J.R.R. Tolkien', 12.99),
(6, 'Fahrenheit 451', 'Ray Bradbury', 7.99),
(7, 'Pride and Prejudice', 'Jane Austen', 6.99),
(8, 'Moby-Dick', 'Herman Melville', 11.99),
(9, 'The Odyssey', 'Homer', 9.50),
(10, 'War and Peace', 'Leo Tolstoy', 14.99),
(11, 'The Divine Comedy', 'Dante Alighieri', 13.99),
(12, 'Crime and Punishment', 'Fyodor Dostoevsky', 11.50),
(13, 'The Brothers Karamazov', 'Fyodor Dostoevsky', 12.50),
(14, 'Wuthering Heights', 'Emily Brontë', 7.50),
(15, 'Jane Eyre', 'Charlotte Brontë', 8.50),
(16, 'Brave New World', 'Aldous Huxley', 9.99),
(17, 'Animal Farm', 'George Orwell', 6.99),
(18, 'The Picture of Dorian Gray', 'Oscar Wilde', 10.99),
(19, 'A Tale of Two Cities', 'Charles Dickens', 8.99),
(20, 'Les Misérables', 'Victor Hugo', 13.50);
```

Customers1 Table

```
CREATE TABLE Customers1(
Customers_id int,
Customers_Name varchar(35),
Email varchar(35),
Address varchar(35));

INSERT INTO Customers1 (Customers_id, Customers_Name, Email,
Address)
VALUES(1, 'John Doe', 'john.doe@example.com', '123 Maple Street,
Springfield'),
(2, 'Jane Smith', 'jane.smith@example.com', '456 Oak Street, Springfield'),
(3, 'Alice Johnson', 'alice.johnson@example.com', '789 Pine Street,
Springfield'),
(4, 'Bob Brown', 'bob.brown@example.com', '101 Birch Street,
Springfield'),
(5, 'Charlie Davis', 'charlie.davis@example.com', '202 Cedar Street,
Springfield'),
(6, 'Diana Evans', 'diana.evans@example.com', '303 Elm Street,
Springfield'),
(7, 'Ethan Harris', 'ethan.harris@example.com', '404 Fir Street,
Springfield'),
(8, 'Fiona Green', 'fiona.green@example.com', '505 Palm Street, Springfield'),
(9, 'George Hill', 'george.hill@example.com', '606 Cypress Street, Springfield'),
(10, 'Hannah King', 'hannah.king@example.com', '707 Redwood Street, Springfield'),
(11, 'Ian Lewis', 'ian.lewis@example.com', '808 Willow Street, Springfield'),
(12, 'Jessica Moore', 'jessica.moore@example.com', '909 Spruce Street, Springfield'),
(13, 'Kevin Martin', 'kevin.martin@example.com', '1010 Alder Street, Springfield'),
(14, 'Laura Nelson', 'laura.nelson@example.com', '1111 Ash Street, Springfield'),
(15, 'Michael Connor', 'michael.oconnor@example.com', '1212 Beech Street, Springfield'),
(16, 'Nina Parker', 'nina.parker@example.com', '1313 Cherry Street, Springfield'),
(17, 'Oliver Quinn', 'oliver.quinn@example.com', '1414 Poplar Street, Springfield'),
(18, 'Patricia Roberts', 'patricia.roberts@example.com', '1515 Dogwood Street, Springfield'),
(19, 'Quentin Stewart', 'quentin.stewart@example.com', '1616 Magnolia Street, Springfield'),
(20, 'Rachel Thompson', 'rachel.thompson@example.com', '1717 Laurel Street, Springfield');
```

Orders Table

```
CREATE TABLE Orders(
Order_id int,
Customers_id varchar(35),
Order_date date);

INSERT INTO Orders (Order_id, Customers_id, Order_date)
```

```
VALUES(1, 1, '2023-01-10'),
(2, 2, '2023-01-15'),
(3, 3, '2023-01-20'),
(4, 4, '2023-01-25'),
(5, 5, '2023-02-01'),
(6, 6, '2023-02-05'),
(7, 7, '2023-02-10'),
(8, 8, '2023-02-15'),
(9, 9, '2023-02-20'),
(10, 10, '2023-02-25'),
(11, 11, '2023-03-01'),
(12, 12, '2023-03-05'),
(13, 13, '2023-03-10'),
(14, 14, '2023-03-15'),
(15, 15, '2023-03-20'),
(16, 16, '2023-03-25'),
(17, 17, '2023-04-01'),
(18, 18, '2023-04-05'),
(19, 19, '2023-04-10'),
(20, 20, '2023-04-15');
```

Order_items Table

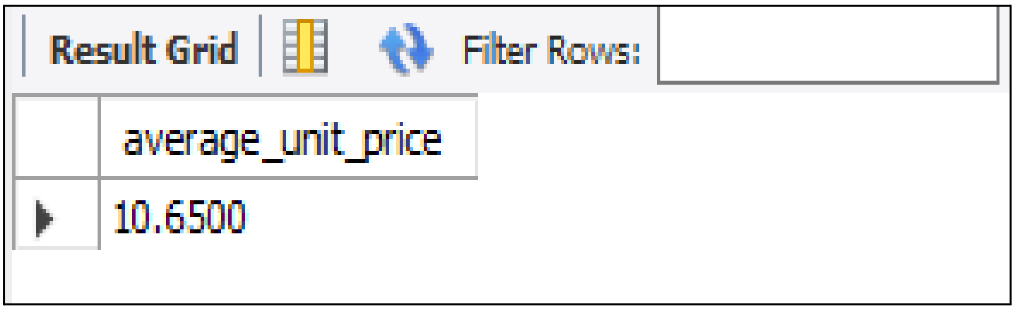
```
CREATE TABLE Order_items(
 Order_items_id int,
 Order_id int,
 Book_id int,
 Quantity int,
 Unit_price int);

 INSERT INTO  Order_items(Order_items_id, Order_id, Book_id,
Quantity, Unit_price)
 VALUES(1, 1, 1, 2, 10.99),
(2, 1, 2, 1, 8.99),
(3, 2, 3, 1, 9.99),
(4, 2, 4, 3, 10.50),
(5, 3, 5, 1, 12.99),
(6, 3, 6, 2, 7.99),
(7, 4, 7, 1, 6.99),
(8, 4, 8, 1, 11.99),
(9, 5, 9, 2, 9.50),
(10, 5, 10, 1, 14.99),
(11, 6, 11, 1, 13.99),
(12, 6, 12, 1, 11.50),
(13, 7, 13, 2, 12.50),
(14, 7, 14, 1, 7.50),
(15, 8, 15, 1, 8.50),
(16, 8, 16, 2, 9.99),
(17, 9, 17, 1, 6.99),
(18, 9, 18, 1, 10.99),
(19, 10, 19, 3, 8.99),
(20, 10, 20, 2, 13.50);
```

CASE STUDY QUESTIONS & ANSWERS

1. **How to Calculate the average unit price of Order_times ?**

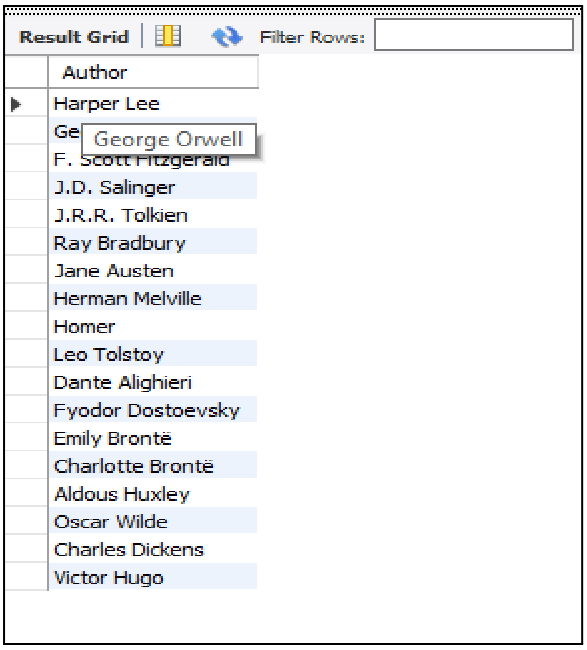
```
SELECT AVG(Unit_price) AS average_unit_price
FROM Order_items;
```



```
average_unit_price
10.6500
```

2. How can we retrieve a list of distinct authors from the Books table?

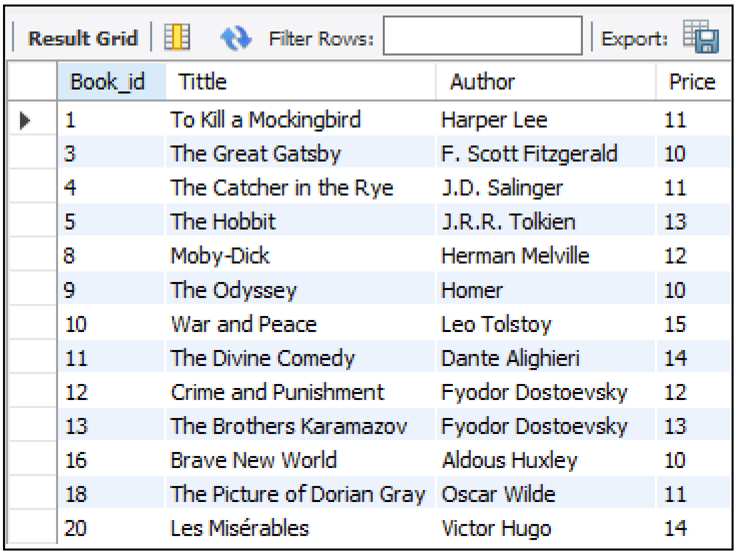
```
SELECT DISTINCT Author
FROM Books;
```



Author
Harper Lee
George Orwell
F. Scott Fitzgerald
J.D. Salinger
J.R.R. Tolkien
Ray Bradbury
Jane Austen
Herman Melville
Homer
Leo Tolstoy
Dante Alighieri
Fyodor Dostoevsky
Emily Brontë
Charlotte Brontë
Aldous Huxley
Oscar Wilde
Charles Dickens
Victor Hugo

3. How to Select all products with a price between 10 and 20 ?

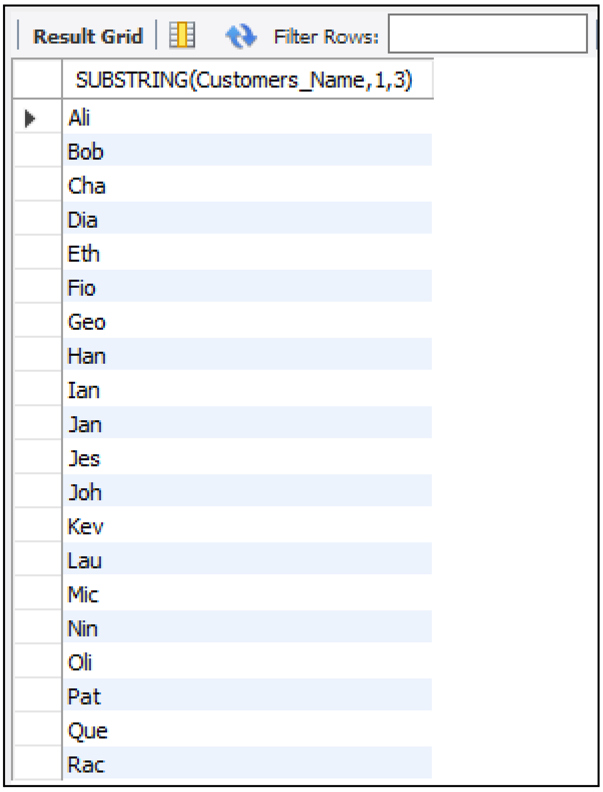
```
SELECT * FROM  Books
WHERE  Price  BETWEEN  10  AND  20
```



Book_id	Tittle	Author	Price
1	To Kill a Mockingbird	Harper Lee	11
3	The Great Gatsby	F. Scott Fitzgerald	10
4	The Catcher in the Rye	J.D. Salinger	11
5	The Hobbit	J.R.R. Tolkien	13
8	Moby-Dick	Herman Melville	12
9	The Odyssey	Homer	10
10	War and Peace	Leo Tolstoy	15
11	The Divine Comedy	Dante Alighieri	14
12	Crime and Punishment	Fyodor Dostoevsky	12
13	The Brothers Karamazov	Fyodor Dostoevsky	13
16	Brave New World	Aldous Huxley	10
18	The Picture of Dorian Gray	Oscar Wilde	11
20	Les Misérables	Victor Hugo	14

4. How to print the first 3 characters of Customer_Name from Customers1 table ?

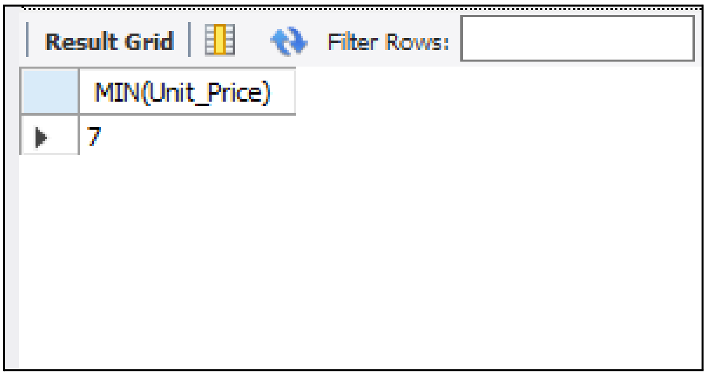
```
SELECT SUBSTRING(Customers_Name,1,3)
FROM Customers1;
```



SUBSTRING(Customers_Name,1,3)
Ali
Bob
Cha
Dia
Eth
Fio
Geo
Han
Ian
Jan
Jes
Joh
Kev
Lau
Mic
Nin
Oli
Pat
Que
Rac

5. How to find the lowest price in the unit_price ?

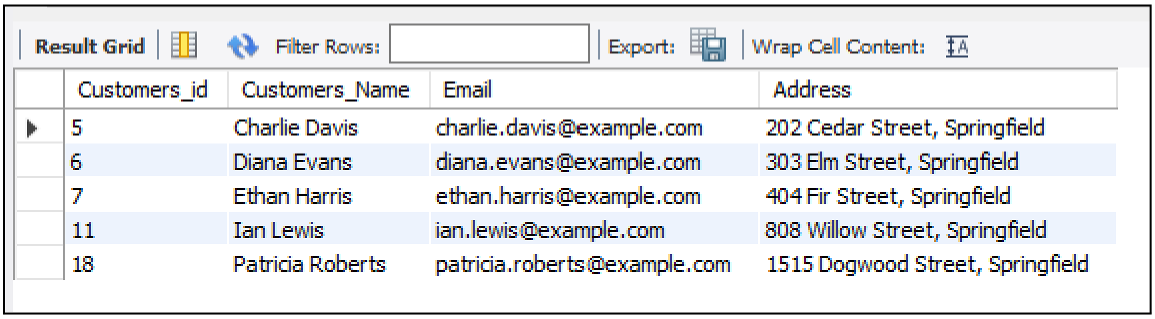
```
SELECT MIN(Unit_Price)
FROM order_items;
```



MIN(Unit_Price)
7

6. How to print details of the Customers1 whose Customers_Name ends with 's' ?

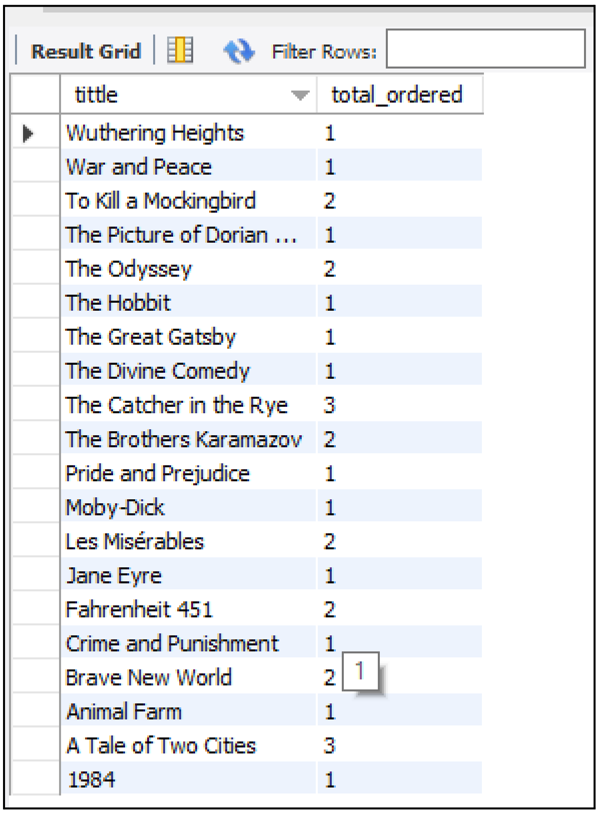
```
SELECT * FROM Customers1
WHERE Customers_Name  LIKE '%s';
```



Customers_id	Customers_Name	Email	Address
5	Charlie Davis	charlie.davis@example.com	202 Cedar Street, Springfield
6	Diana Evans	diana.evans@example.com	303 Elm Street, Springfield
7	Ethan Harris	ethan.harris@example.com	404 Fir Street, Springfield
11	Ian Lewis	ian.lewis@example.com	808 Willow Street, Springfield
18	Patricia Roberts	patricia.roberts@example.com	1515 Dogwood Street, Springfield

7. How to list all books along with their order quantities ?

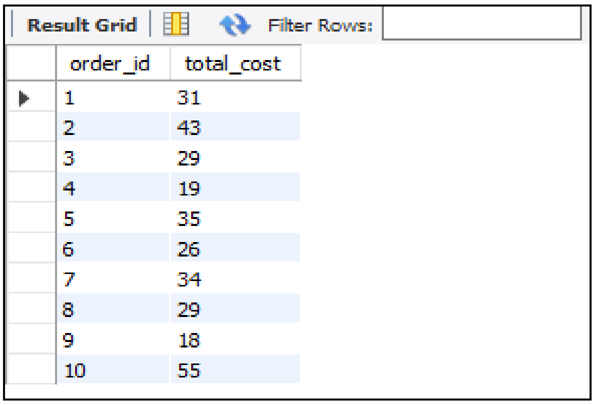
```
 SELECT b.tittle,SUM(o.quantity) AS total_ordered
 FROM Books b
JOIN Order_items o ON b.Book_id = o.Book_id
GROUP BY b.tittle;
```



tittle	total_ordered
Wuthering Heights	1
War and Peace	1
To Kill a Mockingbird	2
The Picture of Dorian ...	1
The Odyssey	2
The Hobbit	1
The Great Gatsby	1
The Divine Comedy	1
The Catcher in the Rye	3
The Brothers Karamazov	2
Pride and Prejudice	1
Moby-Dick	1
Les Misérables	2
Jane Eyre	1
Fahrenheit 451	2
Crime and Punishment	1
Brave New World	2
Animal Farm	1
A Tale of Two Cities	3
1984	1

8. How to calculate the total cost of each order ?

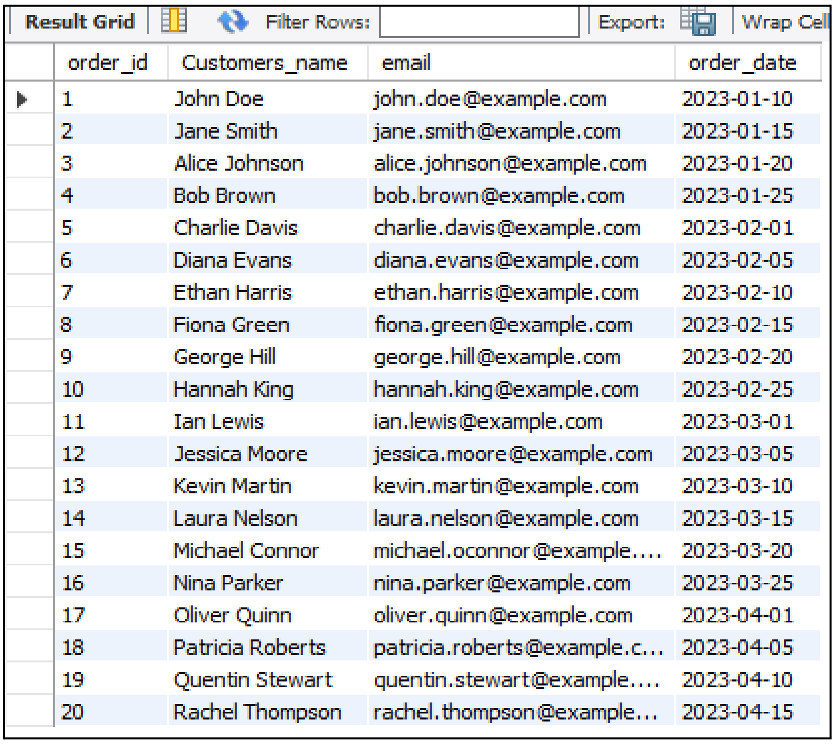
```
SELECT o.order_id, SUM(o.quantity * o.unit_price) AS total_cost
FROM Order_items o
GROUP BY o.order_id;
```



order_id	total_cost
1	31
2	43
3	29
4	19
5	35
6	26
7	34
8	29
9	18
10	55

9. How to list all orders with the customer name, email, and order date ?

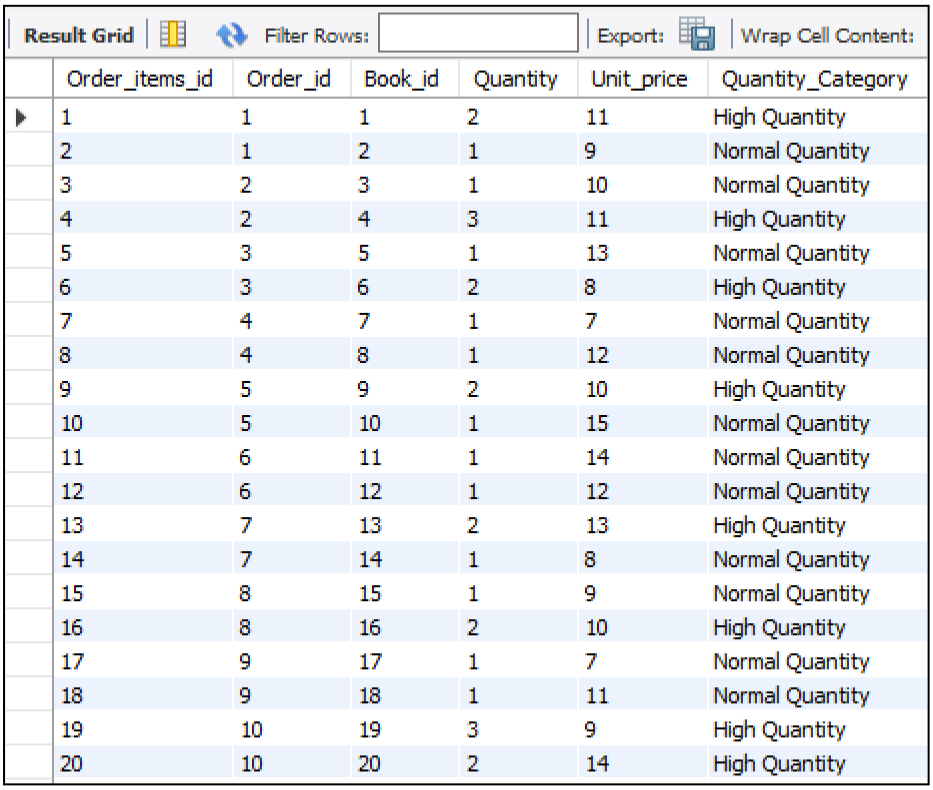
```
SELECT o.order_id, C.Customers_name, C.email, o.order_date
FROM Orders o
INNER JOIN Customers1 c
ON o.Customers_id = c.customers_id;
```



order_id	Customers_name	email	order_date
1	John Doe	john.doe@example.com	2023-01-10
2	Jane Smith	jane.smith@example.com	2023-01-15
3	Alice Johnson	alice.johnson@example.com	2023-01-20
4	Bob Brown	bob.brown@example.com	2023-01-25
5	Charlie Davis	charlie.davis@example.com	2023-02-01
6	Diana Evans	diana.evans@example.com	2023-02-05
7	Ethan Harris	ethan.harris@example.com	2023-02-10
8	Fiona Green	fiona.green@example.com	2023-02-15
9	George Hill	george.hill@example.com	2023-02-20
10	Hannah King	hannah.king@example.com	2023-02-25
11	Ian Lewis	ian.lewis@example.com	2023-03-01
12	Jessica Moore	jessica.moore@example.com	2023-03-05
13	Kevin Martin	kevin.martin@example.com	2023-03-10
14	Laura Nelson	laura.nelson@example.com	2023-03-15
15	Michael Connor	michael.oconnor@example....	2023-03-20
16	Nina Parker	nina.parker@example.com	2023-03-25
17	Oliver Quinn	oliver.quinn@example.com	2023-04-01
18	Patricia Roberts	patricia.roberts@example.c...	2023-04-05
19	Quentin Stewart	quentin.stewart@example....	2023-04-10
20	Rachel Thompson	rachel.thompson@example...	2023-04-15

10. .How to a sql query identify orders with high quantities ?

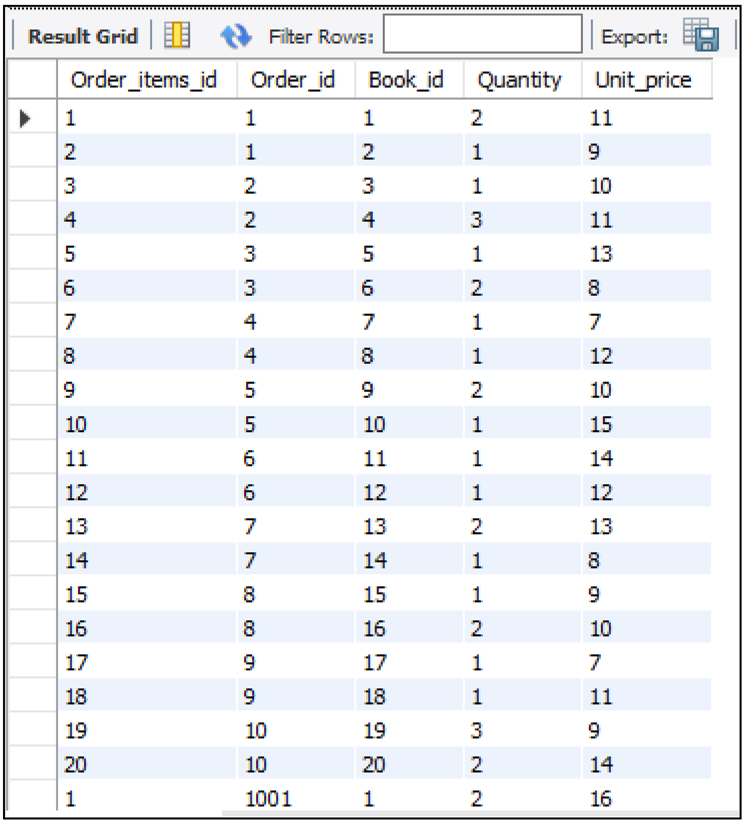
```
 SELECT Order_items_id, Order_id, Book_id, Quantity, Unit_price,
CASE
WHEN Quantity > 1 THEN 'High Quantity'
ELSE 'Normal Quantity'
END AS Quantity_Category
```



Order_items_id	Order_id	Book_id	Quantity	Unit_price	Quantity_Category
1	1	1	2	11	High Quantity
2	1	2	1	9	Normal Quantity
3	2	3	1	10	Normal Quantity
4	2	4	3	11	High Quantity
5	3	5	1	13	Normal Quantity
6	3	6	2	8	High Quantity
7	4	7	1	7	Normal Quantity
8	4	8	1	12	Normal Quantity
9	5	9	2	10	High Quantity
10	5	10	1	15	Normal Quantity
11	6	11	1	14	Normal Quantity
12	6	12	1	12	Normal Quantity
13	7	13	2	13	High Quantity
14	7	14	1	8	Normal Quantity
15	8	15	1	9	Normal Quantity
16	8	16	2	10	High Quantity
17	9	17	1	7	Normal Quantity
18	9	18	1	11	Normal Quantity
19	10	19	3	9	High Quantity
20	10	20	2	14	High Quantity

11. How to using a CASE statement in the INSERT for the Order_times table?

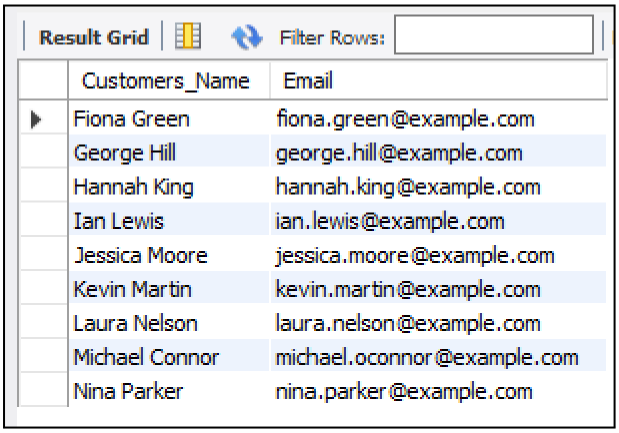
```
  INSERT INTO Order_times (Order_items_id, Order_id, Book_id,
Quantity,                 Unit_price)
   VALUES (1, 1001, 1, 2,
    CASE
    WHEN 2 > 1 THEN 15.99
    ELSE 12.99
     END);
```



Order_items_id	Order_id	Book_id	Quantity	Unit_price
1	1	1	2	11
2	1	2	1	9
3	2	3	1	10
4	2	4	3	11
5	3	5	1	13
6	3	6	2	8
7	4	7	1	7
8	4	8	1	12
9	5	9	2	10
10	5	10	1	15
11	6	11	1	14
12	6	12	1	12
13	7	13	2	13
14	7	14	1	8
15	8	15	1	9
16	8	16	2	10
17	9	17	1	7
18	9	18	1	11
19	10	19	3	9
20	10	20	2	14
1	1001	1	2	16

12. Which customers placed orders in July 2023?

```
SELECT Customers_Name, Email
FROM Customers1
WHERE Customers_id IN (
SELECT Customers_id
FROM Orders
WHERE Order_date BETWEEN '2023-02-15' AND '2023-03-25'
 );
```



Customers_Name	Email
Fiona Green	fiona.green@example.com
George Hill	george.hill@example.com
Hannah King	hannah.king@example.com
Ian Lewis	ian.lewis@example.com
Jessica Moore	jessica.moore@example.com
Kevin Martin	kevin.martin@example.com
Laura Nelson	laura.nelson@example.com
Michael Connor	michael.oconnor@example.com
Nina Parker	nina.parker@example.com