

Query 1: Provide a list of customer names, along with the total dollar amount each customer has spent.

Plain English:

Retrieve the names of customers along with the total amount each customer has spent, calculated by summing the total price of the books in their orders.

SQL:

```
SELECT c.Name, SUM(oi.Quantity * oi.Price) AS TotalSpent
FROM customer AS c
JOIN customer_order AS co ON c.CustomerID = co.CustomerID
JOIN orderItem AS oi ON co.OrderID = oi.OrderID
GROUP BY c.CustomerID;
```

Query 2: Provide a list of customer names and e-mail addresses for customers who have spent more than the average customer.

Plain English:

Retrieve the names and email addresses of customers who have spent more than the average amount, calculated by comparing each customer's total spending to the average.

SQL:

```
WITH TotalSpent AS (
SELECT c.CustomerID, SUM(oi.Quantity * oi.Price) AS TotalSpent
FROM customer AS c
JOIN customer_order AS co ON c.CustomerID = co.CustomerID
JOIN orderItem AS oi ON co.OrderID = oi.OrderID
GROUP BY c.CustomerID
),
AvgSpent AS (
SELECT AVG(TotalSpent) AS AvgSpent
FROM TotalSpent
)
SELECT c.Name, c.Email
FROM customer AS c
JOIN TotalSpent AS ts ON c.CustomerID = ts.CustomerID
JOIN AvgSpent AS as_avg ON ts.TotalSpent > as_avg.AvgSpent;
```

Query 3: Provide a list of the titles in the database and associated total copies sold to customers, sorted from the title that has sold the most individual copies to the title that has sold the least.

Plain English:

Retrieve the titles of books along with the total number of copies sold, sorted in descending order by total copies sold.

SQL:

```
SELECT b.Title, SUM(oi.Quantity) AS TotalCopiesSold
FROM book AS b
JOIN orderItem AS oi ON b.ISBN = oi.ISBN
GROUP BY b.ISBN
ORDER BY TotalCopiesSold DESC;
```

Query 4: Provide a list of the titles in the database and associated dollar totals for copies sold to customers, sorted from the title that has sold the highest dollar amount to the title that has sold the smallest.

Plain English:

Retrieve the titles of books along with the total dollar amount generated from their sales, sorted in descending order by total sales amount.

SQL:

```
SELECT b.Title, SUM(oi.Quantity * oi.Price) AS TotalSales
FROM book AS b
JOIN orderItem AS oi ON b.ISBN = oi.ISBN
GROUP BY b.ISBN
ORDER BY TotalSales DESC;
```

Query 5: Find the most popular author in the database (i.e. the one who has sold the most books).

Plain English:

Find the author who has sold the most books by summing up the total quantity of books sold by each author and identifying the author with the highest total.

SQL:

```
WITH TotalBooksSold AS (
  SELECT ba.AuthorID, SUM(oi.Quantity) AS TotalSold
  FROM book_author AS ba
  JOIN orderItem AS oi ON ba.ISBN = oi.ISBN
  GROUP BY ba.AuthorID
)
SELECT a.Name
FROM author AS a
JOIN TotalBooksSold AS tbs ON a.AuthorID = tbs.AuthorID
ORDER BY tbs.TotalSold DESC
```

```
LIMIT 1;
```

Query 6: Find the most profitable author in the database for this store (i.e. the one who has brought in the most money).

Plain English:

Find the author who has generated the most revenue for the store by calculating the total sales amount for each author and identifying the author with the highest total.

SQL:

```
WITH TotalSales AS (  
  SELECT ba.AuthorID, SUM(oi.Quantity * oi.Price) AS TotalRevenue  
  FROM book_author AS ba  
  JOIN orderItem AS oi ON ba.ISBN = oi.ISBN  
  GROUP BY ba.AuthorID  
)  
SELECT a.Name  
FROM author AS a  
JOIN TotalSales AS ts ON a.AuthorID = ts.AuthorID  
ORDER BY ts.TotalRevenue DESC  
LIMIT 1;
```

Query 7: Provide a list of customer information for customers who purchased anything written by the most profitable author in the database.

Plain English:

Retrieve the customer information for customers who have purchased books written by the most profitable author.

SQL:

```
WITH MostProfitableAuthor AS (  
  SELECT ba.AuthorID  
  FROM book_author AS ba  
  JOIN orderItem AS oi ON ba.ISBN = oi.ISBN  
  JOIN profitMargin AS pm ON oi.ISBN = pm.ISBN  
  GROUP BY ba.AuthorID  
  ORDER BY SUM(oi.Quantity * oi.Price) DESC  
  LIMIT 1  
)  
SELECT c.CustomerID, c.Name, c.Email, c.PhoneNumber  
FROM customer AS c  
JOIN customer_order AS co ON c.CustomerID = co.CustomerID  
JOIN orderItem AS oi ON co.OrderID = oi.OrderID
```

```
JOIN book_author AS ba ON oi.ISBN = ba.ISBN
WHERE ba.AuthorID = (SELECT AuthorID FROM MostProfitableAuthor);
```

Query 8: Provide the list of authors who wrote the books purchased by the customers who have spent more than the average customer.

Plain English:

Retrieve the authors who wrote the books purchased by customers who have spent more than the average customer.

SQL:

```
WITH TotalSpent AS (
SELECT c.CustomerID, SUM(oi.Quantity * oi.Price) AS TotalSpent
FROM customer AS c
JOIN customer_order AS co ON c.CustomerID = co.CustomerID
JOIN orderItem AS oi ON co.OrderID = oi.OrderID
GROUP BY c.CustomerID
),
AvgSpent AS (
SELECT AVG(TotalSpent) AS AvgSpent
FROM TotalSpent
),
CustomersAboveAvg AS (
SELECT CustomerID
FROM TotalSpent
WHERE TotalSpent > (SELECT AvgSpent FROM AvgSpent)
)
SELECT DISTINCT a.Name
FROM author AS a
JOIN book_author AS ba ON a.AuthorID = ba.AuthorID
JOIN orderItem AS oi ON ba.ISBN = oi.ISBN
JOIN customer_order AS co ON oi.OrderID = co.OrderID
JOIN CustomersAboveAvg AS cba ON co.CustomerID = cba.CustomerID;
```