

```
CREATE TABLE Books (
    book_id INT PRIMARY KEY,
    title VARCHAR(100),
    author VARCHAR(100),
    genre VARCHAR(50),
    price DECIMAL(6,2),
    stock INT
);

INSERT INTO Books (book_id, title, author, genre, price, stock)
VALUES
(1, 'The Silent Forest', 'Laura Green', 'Fiction', 14.99, 120),
(2, 'Data Science 101', 'Sam Curtis', 'Education', 34.50, 60),
(3, 'Journey to Mars', 'Alan Peters', 'Sci-Fi', 19.75, 80),
(4, 'Mastering SQL', 'Janet Cole', 'Education', 29.99, 40),
(5, 'Healthy Living', 'Mia Stone', 'Lifestyle', 22.50, 100);

CREATE TABLE Sales (
    sale_id INT PRIMARY KEY,
    book_id INT,
    quantity INT,
    sale_date DATE,
    FOREIGN KEY (book_id) REFERENCES Books(book_id)
);

INSERT INTO Sales (sale_id, book_id, quantity, sale_date) VALUES
(1, 1, 3, '2025-01-03'),
(2, 2, 1, '2025-01-04'),
(3, 3, 4, '2025-01-04'),
(4, 1, 2, '2025-01-05'),
(5, 4, 1, '2025-01-06'),
(6, 3, 2, '2025-01-06'),
(7, 5, 5, '2025-01-07'),
(8, 2, 2, '2025-01-08');
```

### 1) Total revenue by Genre

```
SELECT b.genre,
       SUM(s.quantity * b.price) AS total_revenue
  FROM Sales s
 JOIN Books b ON s.book_id = b.book_id
 GROUP BY b.genre
 ORDER BY total_revenue DESC;
```

The screenshot shows a SQL query results window. At the top, there are tabs for 'Results' and 'Messages'. The 'Results' tab is selected, displaying a table with two columns: 'genre' and 'total\_revenue'. The data is as follows:

|   | genre     | total_revenue |
|---|-----------|---------------|
| 1 | Education | 133.49        |
| 2 | Sci-Fi    | 118.50        |
| 3 | Lifestyle | 112.50        |
| 4 | Fiction   | 74.95         |

### 2) Bestselling books

```
SELECT b.title,
       SUM(s.quantity) AS units_sold
  FROM Sales s
 JOIN Books b ON s.book_id = b.book_id
 GROUP BY b.title
 ORDER BY units_sold DESC;
```

The screenshot shows a SQL query results window. At the top, there are tabs for 'Results' and 'Messages'. The 'Results' tab is selected, displaying a table with two columns: 'title' and 'units\_sold'. The data is as follows:

|   | title             | units_sold |
|---|-------------------|------------|
| 1 | Journey to Mars   | 6          |
| 2 | Healthy Living    | 5          |
| 3 | The Silent Forest | 5          |
| 4 | Data Science 101  | 3          |
| 5 | Mastering SQL     | 1          |

### 3) Total revenue per book

```
SELECT b.title,
       SUM(s.quantity * b.price) AS revenue
  FROM Sales s
 JOIN Books b ON s.book_id = b.book_id
 GROUP BY b.title
 ORDER BY revenue DESC;
```

The screenshot shows a software interface for running SQL queries. At the top, there are two tabs: "Results" (which is selected, indicated by a blue background) and "Messages". Below the tabs is a table with five rows of data. The table has three columns: "title" (containing the book titles), "revenue" (containing the total revenue for each book), and a row number column (containing values 1 through 5). The data is sorted by revenue in descending order, with "Journey to Mars" at the top.

|   | title             | revenue |
|---|-------------------|---------|
| 1 | Journey to Mars   | 118.50  |
| 2 | Healthy Living    | 112.50  |
| 3 | Data Science 101  | 103.50  |
| 4 | The Silent Forest | 74.95   |
| 5 | Mastering SQL     | 29.99   |

#### 4) Books by Revenue

```
SELECT title, revenue,
       RANK() OVER (ORDER BY revenue DESC) AS ranking
FROM (
    SELECT b.title,
           SUM(s.quantity * b.price) AS revenue
    FROM Sales s
   JOIN Books b ON s.book_id = b.book_id
   GROUP BY b.title
) t;
```

The screenshot shows a database query results window. At the top, there are two tabs: "Results" (which is selected) and "Messages". Below the tabs is a table with four columns: "title", "revenue", and "ranking". The "title" column contains the book names, and the "revenue" column contains their total sales amount. The "ranking" column shows the rank of each book based on its revenue, with 1 being the highest. The table has 5 rows, corresponding to the books listed in the query results.

|   | title             | revenue | ranking |
|---|-------------------|---------|---------|
| 1 | Journey to Mars   | 118.50  | 1       |
| 2 | Healthy Living    | 112.50  | 2       |
| 3 | Data Science 101  | 103.50  | 3       |
| 4 | The Silent Forest | 74.95   | 4       |
| 5 | Mastering SQL     | 29.99   | 5       |