Group Summary Statistics with SQL: Takeaways

Syntax

- Counting rows by group

```
SELECT group_column, COUNT(*) AS num_row
   FROM table
  GROUP BY group_column;
```

- Summing a computed column by group

```
SELECT group_column, SUM(column_1 * column_2) AS total
   FROM table
  GROUP BY group_column;
```

- Computing statistics with grouped data under conditions

```
SELECT group_column, COUNT(*) AS num_row, AVG(column) AS avg_column
   FROM table
  WHERE conditions
  GROUP BY group_column;
```

- Grouping rows and ordering the result

```
SELECT group_column,
        COUNT(*) AS num_row,
        SUM(column) AS sum_column
   FROM table
  GROUP BY group_column
  ORDER BY sum_column DESC,
        num_row DESC
  LIMIT n;
```

- Writing a comprehensive query

```
SELECT billing_city,
        COUNT(*) AS num_row,
        SUM(total) AS overall_sale,
        MIN(total) AS min_sale,
        AVG(total) AS avg_sale,
        MAX(total) AS max_sale
   FROM invoice
  WHERE billing_country = 'Canada'
     OR billing_country = 'France'
  GROUP BY billing_city
  ORDER BY overall_sale DESC, num_row DESC
  LIMIT 3;
```

Concepts

- Aggregate functions allow us to make operations combining several rows over groups.
- With the new clause, the new SQL order of clauses is as follows:

```
SELECT > FROM > WHERE > GROUP BY > ORDER BY > LIMIT
```

- With the new clause, the new SQL execution order is as follows:

```
FROM > WHERE > GROUP BY > SELECT > ORDER BY > LIMIT
```

Resources

- SQL aggregate functions