Theory: Results ordering

19 minutes 0 / 5 problems solved

Let's imagine that today you are responsible for a hotel booking service database. You already know how to select a set of rows that contains information about hotels according to criteria specified by the client. Still, there are many hotels in the selection, so you need to sort them by price, rating, other attribute or even expression requested by a client. Let's see how this can be done in SQL.

§1. ORDER BY

When you query data, SQL does not provide any default order of rows in the query evaluation result. To specify order of the resulting rows, you should use **ORDER BY** clause in the query.

Let's consider an example query:

```
SELECT
    hotel_id,
    hotel_name,
    price_per_night,
    price_for_early_check_in,
    rating,
    stars
FROM
    hotels

ORDER BY

    price_per_night

;
```

We have specified at the end of SELECT statement that the resulting rows should be sorted by attribute *price_per_night*.

You may sort the rows by expressions as well. For example, in the query below, we order hotels by price for two nights with early check-in:

```
SELECT
    hotel_id,
    hotel_name,
    price_per_night,
    price_for_early_check_in,
    rating,
    stars
FROM
    hotels

ORDER BY

    price_per_night*2 + price_for_early_check_in

;
```

§2. Ascending and descending

The sorting is based on a definition of the comparison operator (<) for expression type. It can be specified whether greater or smaller values should be higher. Let's consider an example:

Current topic:

Results ordering ...

Topic depends on:

✕ SELECT FROM statement ...

Table of contents:

↑ Results ordering

§1. ORDER BY

§2. Ascending and descending

§3. Multiple expressions

§4. Syntax

§5. Conclusion

Feedback & Comments

```
SELECT
    hotel_id,
    hotel_name,
    price_per_night,
    price_for_early_check_in,
    rating,
    stars
FROM
    hotels

ORDER BY

    rating DESC

;
```

Here, the ORDER BY clause is `ORDER BY expr [ASC, DESC]`: ORDER BY keywords, correct expression, and optional **keywords ASC or DESC** to specify whether the order is ascending or descending. By default, the order is assumed to be ascending, which is why we omitted the keyword ASC in the previous example queries.

§3. Multiple expressions

Let's write a query that sorts hotels by both price and rating:

```
SELECT
    hotel_id,
    hotel_name,
    price_per_night,
    price_for_early_check_in,
    rating,
    stars
FROM
    hotels

ORDER BY

    rating DESC,

    price_per_night*2 + price_for_early_check_in

;
```

When more than one expression is specified in ORDER BY clause, the later values are used to sort rows that are equal according to the earlier values. Each expression can be followed by an optional **`ASC`** or **`DESC`** keyword. In our example, hotels should be sorted by rating (from greater to smaller), and those with equal ratings should be ordered by price.

§4. Syntax

If you sort the result rows by an expression that defines a result attribute, you can address it in ORDER BY clause by a column alias or number. For example, in the query below, we sort rows by total price and rating:

```
SELECT
    hotel_name,
    price_per_night*2 + price_for_early_check_in AS total_price,
    rating,
    stars
FROM
    hotels
ORDER BY
    total_price, 3 DESC

;
```

§5. Conclusion

The ORDER BY clause allows you to sort the rows returned from the SELECT statement in ascending or descending order based on the specified expressions.

The ORDER BY clause pattern is the following: `ORDER BY expr1 [ASC, DESC], ..., exprN [ASC, DESC]`; ORDER BY keywords, list of expressions (or corresponding aliases or numbers from SELECT part of a query) with optional ASC or DESC keywords.

Report a typo

107 users liked this theory. **1** didn't like it. **What about you?**

Start practicing

Comments (4) Hints (0) Useful links (0) Show discussion