Course name:A level **SUBJECT:DATABASE TECHNOLOGIES**

Topic:Maria db **DATE:17/4/2020**

JOIN

When you need to retrieve data from more than one tables at a go, use MariaDB JOINS. This means that a JOIN works on two or more tables. The following three types of JOINS are supported in MariaDB:

- INNER/SIMPLE JOIN
- LEFT OUTER JOIN/LEFT JOIN
- RIGHT OUTER JOIN/RIGHT JOIN

Let us discuss them one-by-one:

INNER JOIN

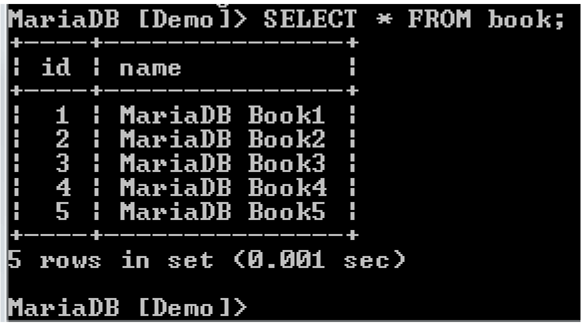
The inner join returns all rows from the tables in which the join condition is true. Its syntax is as follows:

```
SELECT columns
FROM table-1
INNER JOIN table-2
ON table-1.column = table-2.column;
```

For example:

We will use our two tables, books, and book.

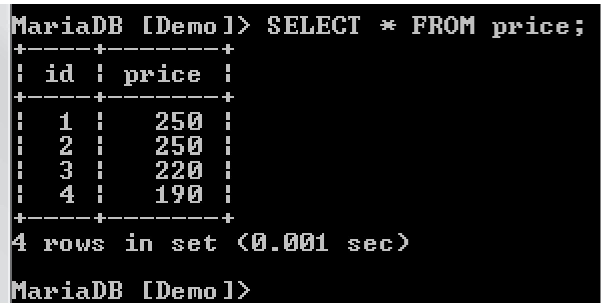
The book table has the following data:



```
MariaDB [Demo]> SELECT * FROM book;
+----+----------------+
| id | name           |
+----+----------------+
|  1 | MariaDB Book1  |
|  2 | MariaDB Book2  |
|  3 | MariaDB Book3  |
|  4 | MariaDB Book4  |
|  5 | MariaDB Book5  |
+----+----------------+
5 rows in set (0.001 sec)

MariaDB [Demo]>
```

The Price table has the following data:



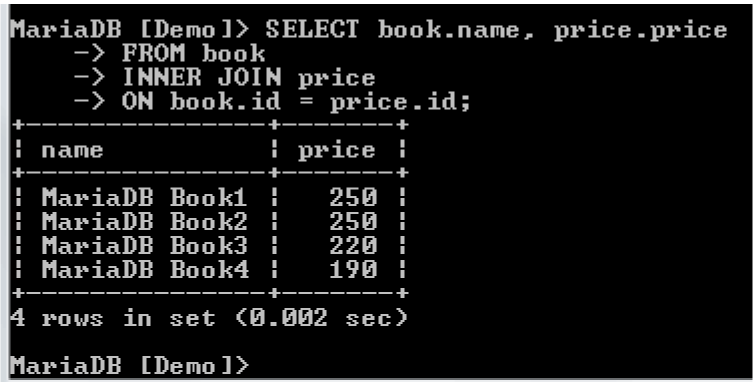
```
MariaDB [Demo]> SELECT * FROM price;
+----+-------+
¦ id ¦ price ¦
+----+-------+
¦  1 ¦   250 ¦
¦  2 ¦   250 ¦
¦  3 ¦   220 ¦
¦  4 ¦   190 ¦
+----+-------+
4 rows in set (0.001 sec)

MariaDB [Demo]>
```

The goal is to join the name column from the Book table and the price column from Price table into a single table. This is possible with an inner join, as demonstrated below:

```
SELECT book.name, price.price
FROM book
INNER JOIN price
ON book.id = price.id;
```

The command returns the following:



```
MariaDB [Demo]> SELECT book.name, price.price
    -> FROM book
    -> INNER JOIN price
    -> ON book.id = price.id;
+---------------+-------+
¦ name          ¦ price ¦
+---------------+-------+
¦ MariaDB Book1 ¦   250 ¦
¦ MariaDB Book2 ¦   250 ¦
¦ MariaDB Book3 ¦   220 ¦
¦ MariaDB Book4 ¦   190 ¦
+---------------+-------+
4 rows in set (0.002 sec)

MariaDB [Demo]>
```

LEFT OUTER JOIN

This join returns all the rows from the left-hand table and only rows in which the join condition is true from the other table. Its syntax is as follows:

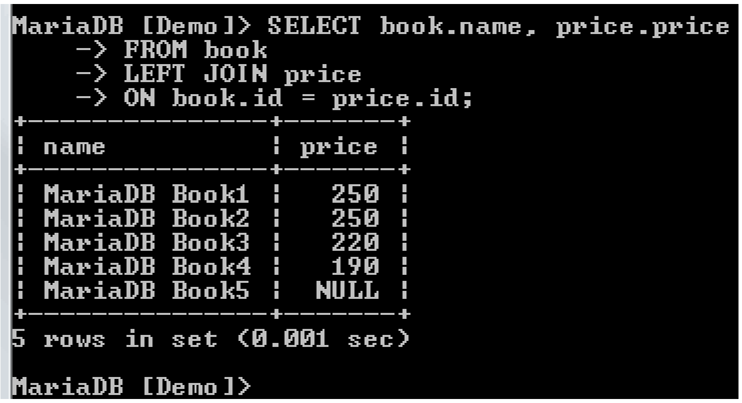
```
SELECT columns
FROM table-1
LEFT [OUTER] JOIN table-2
ON table-1.column = table-2.column;
```

The OUTER keyword has been placed within square brackets because it is optional.

For example:

```
SELECT book.name, price.price
FROM book
LEFT JOIN price
ON book.id = price.id;
```

The command returns the following:



```
MariaDB [Demo]> SELECT book.name, price.price
    -> FROM book
    -> LEFT JOIN price
    -> ON book.id = price.id;
+---------------+-------+
| name          | price |
+---------------+-------+
| MariaDB Book1 |   250 |
| MariaDB Book2 |   250 |
| MariaDB Book3 |   220 |
| MariaDB Book4 |   190 |
| MariaDB Book5 |  NULL |
+---------------+-------+
5 rows in set (0.001 sec)

MariaDB [Demo]>
```

The last record in the above table has no matching value on the left. That is why it has been replaced with NULL.

RIGHT OUTER JOIN

This join returns all the rows from the right-hand table and only rows in which the join condition is true from the other table. Its syntax is as follows:

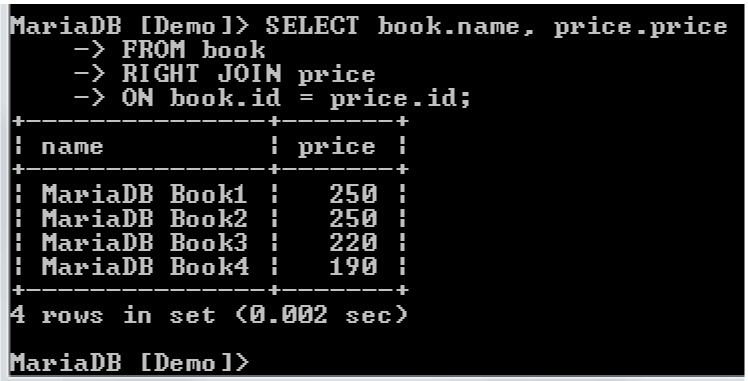
```
SELECT columns
FROM table-1
RIGHT [OUTER] JOIN table-2
ON table-1.column = table-2.column;
```

The OUTER keyword has been placed within square brackets because it is optional.

For example:

```
SELECT book.name, price.price
FROM book
RIGHT JOIN price
ON book.id = price.id;
```

The command returns the following:



```
MariaDB [Demo]> SELECT book.name, price.price
    -> FROM book
    -> RIGHT JOIN price
    -> ON book.id = price.id;
+----------------+-------+
¦ name           ¦ price ¦
+----------------+-------+
¦ MariaDB Book1  ¦   250 ¦
¦ MariaDB Book2  ¦   250 ¦
¦ MariaDB Book3  ¦   220 ¦
¦ MariaDB Book4  ¦   190 ¦
+----------------+-------+
4 rows in set (0.002 sec)

MariaDB [Demo]>
```

The reason is that all rows in the right-hand table were matched to those in the other table. If some of the rows did not match, we would have NULLs in the first column.

Assignment

1. What are different types of Join?

2. How to use different types of join.