Learn with Rabbil

Learn with Rabbil

What is MySQL

- MySQL is a popular open-source relational database management system that is widely used in web applications.
- MySQL supports a wide range of operating systems, including Linux, Windows, and macOS.
- MySQL uses Structured Query Language (SQL) to manage data and is highly scalable, making it suitable for both small and large applications.
- MySQL has several key features, including support for transactions, data security and integrity, and performance optimization tools.
- MySQL is widely used in popular applications such as WordPress, Drupal, and Joomla, as well as in large-scale enterprise systems.

Learn with Rabbil

Important SQL Commands

- SELECT - extracts data from a database
- UPDATE - updates data in a database
- DELETE - deletes data from a database
- INSERT INTO - inserts new data into a database
- CREATE DATABASE - creates a new database
- ALTER DATABASE - modifies a database
- CREATE TABLE - creates a new table
- ALTER TABLE - modifies a table
- DROP TABLE - deletes a table
- CREATE INDEX - creates an index (search key)
- DROP INDEX - deletes an index

Learn with Rabbil

MySQL INSERT INTO Statement

In MySQL, the INSERT INTO statement is used to add new rows to a table

practice.sql

```sql
INSERT INTO employees (first_name, last_name, department, salary)
VALUES ('John', 'Doe', 'Sales', 60000);
```

Learn with Rabbil

MySQL DELETE Statement

In MySQL, the DELETE statement is used to remove one or more rows from a table.

practice.sql

```sql
DELETE FROM employees WHERE department = 'Marketing';
```

Learn with Rabbil

MySQL UPDATE Statement

In MySQL, the UPDATE statement is used to modify one or more rows in a table.

practice.sql

```sql
UPDATE employees SET salary = 65000
WHERE employee_id = 1234;
```

Learn with Rabbil

MySQL SELECT Statement

The SELECT statement in MySQL is used to retrieve data from one or more tables in a database. Here's the basic syntax of the SELECT statement:

practice.sql

```sql
SELECT * FROM employees;
SELECT * FROM employees WHERE department = 'Marketing';
SELECT column1, column2 FROM employees
```

Learn with Rabbil

MySQL SELECT Statement

In addition, you can use various functions and operators within the SELECT statement to manipulate the data, such as:

- COUNT(): counts the number of rows that match a specific condition
- SUM(): calculates the sum of a numeric column
- AVG(): calculates the average of a numeric column
- MAX(): returns the highest value in a column
- MIN(): returns the lowest value in a column
- DISTINCT: selects unique values from a column

Learn with Rabbil

MySQL AND, OR and NOT Operators

In MySQL, the AND, OR, and NOT operators are used to combine multiple conditions in a WHERE clause. Here's a brief overview of each operator:

AND: The AND operator requires that both conditions in the WHERE clause be true for a row to be included in the result set. For example:

practice.sql

```sql
SELECT * FROM employees
WHERE department = 'Sales' AND salary > 50000;
```

Learn with Rabbil

MySQL AND, OR and NOT Operators

OR: The OR operator requires that at least one of the conditions in the WHERE clause be true for a row to be included in the result set. For example:

practice.sql

```sql
SELECT * FROM employees
WHERE department = 'Sales' OR department = 'Marketing';
```

NOT: The NOT operator is used to negate a condition in the WHERE clause. For example:

practice.sql

```sql
SELECT * FROM employees
WHERE NOT department = 'Sales';
```

Learn with Rabbil

MySQL ORDER BY Keyword

In MySQL, the ORDER BY keyword is used to sort the result set of a SELECT statement in either ascending or descending order.

practice.sql

```sql
SELECT first_name, last_name, salary FROM employees
ORDER BY last_name;
```

practice.sql

```sql
SELECT first_name, last_name, salary FROM employees
ORDER BY last_name DESC, salary ASC;
```

Learn with Rabbil

MySQL LIMIT Clause

In MySQL, the LIMIT clause is used to limit the number of rows returned by a query. It is typically used in conjunction with the SELECT statement to retrieve a subset of rows from a table.

practice.sql

```sql
SELECT * FROM employees LIMIT 10;
```

Learn with Rabbil

MySQL MIN() and MAX() Functions

In MySQL, the MIN() and MAX() functions are used to retrieve the minimum and maximum values of a column in a table, respectively

practice.sql

```sql
SELECT MIN(salary) FROM employees;
```

practice.sql

```sql
SELECT MAX(salary) FROM employees;
```

Learn with Rabbil

MySQL COUNT(), AVG() and SUM() Functions

In MySQL, the COUNT(), AVG() and SUM() functions are used to perform aggregate calculations on columns in a table.

practice.sql

```sql
SELECT COUNT(*) FROM employees;
```

practice.sql

```sql
SELECT AVG(salary) FROM employees;
```

practice.sql

```sql
SELECT SUM(salary) FROM employees;
```

Learn with Rabbil

MySQL LIKE Operator

n MySQL, the LIKE operator is used in the WHERE clause of a SELECT statement to search for patterns in a column. The LIKE operator is often used with wildcard characters, which are special characters that represent one or more characters.

practice.sql

```sql
SELECT * FROM employees WHERE first_name LIKE 'J%';
```

Learn with Rabbil

MySQL IN Operator

In MySQL, the IN operator is used to compare a value with a list of values. The IN operator is used in the WHERE clause of a SELECT statement.

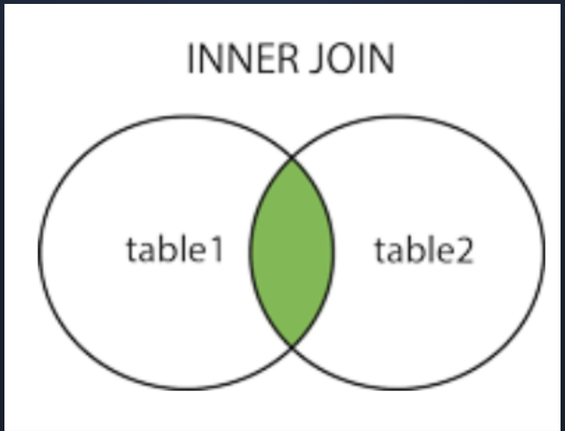
practice.sql

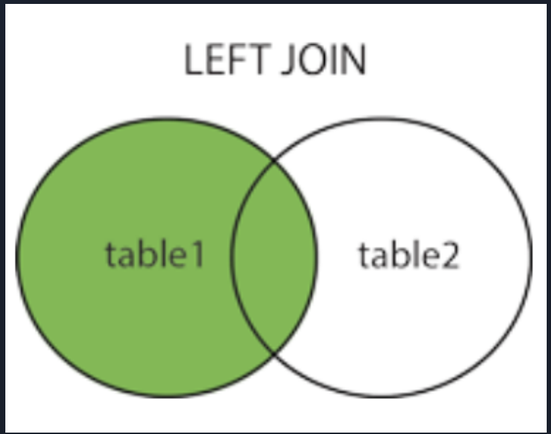
```sql
SELECT * FROM employees
WHERE department IN ('Sales', 'Marketing');
```

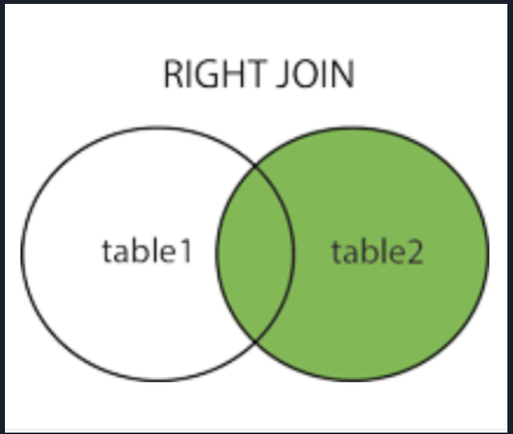
Learn with Rabbil

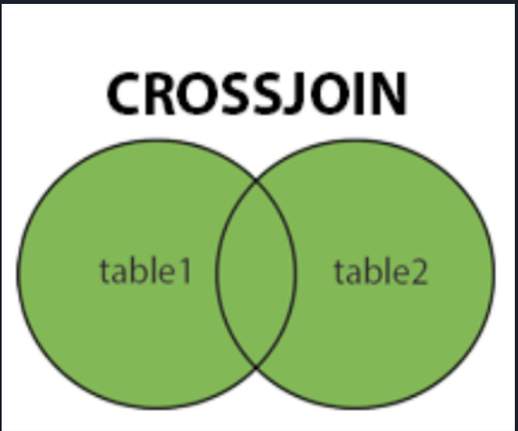
Types of Joins in MySQL

- INNER JOIN: Returns records that have matching values in both tables
- LEFT JOIN: Returns all records from the left table, and the matched records from the right table
- RIGHT JOIN: Returns all records from the right table, and the matched records from the left table
- CROSS JOIN: Returns all records from both tables









Learn with Rabbil

MySQL INNER JOIN Keyword

In MySQL, the INNER JOIN keyword is used to combine rows from two or more tables based on a related column between them. The INNER JOIN returns only the matching rows between the tables.

practice.sql

```sql
SELECT employees.name, departments.department_name FROM employees
INNER JOIN departments ON employees.department_id = departments.id;

```

Learn with Rabbil

MySQL LEFT JOIN Keyword

In MySQL, the LEFT JOIN keyword is used to combine all the rows from the left table with matching rows from the right table. If there are no matching rows in the right table, the result will contain NULL values for the right table columns.

practice.sql

```sql
SELECT employees.name, departments.department_name FROM employees
LEFT JOIN departments ON employees.department_id = departments.id;
```

Learn with Rabbil

MySQL RIGHT JOIN Keyword

In MySQL, the RIGHT JOIN keyword is used to combine all the rows from the right table with matching rows from the left table. If there are no matching rows in the left table, the result will contain NULL values for the left table columns.

practice.sql

```sql
SELECT employees.name, departments.department_name FROM employees
RIGHT JOIN departments ON employees.department_id = departments.id;
```

Learn with Rabbil

CROSS JOIN Keyword

The CROSS JOIN keyword in MySQL is used to combine every row of one table with every row of another table. It produces a Cartesian product of the two tables involved in the join, meaning it will combine all rows from the first table with all rows from the second table.

practice.sql

```sql
SELECT * FROM employees CROSS JOIN departments;

```

Learn with Rabbil

MySQL UNION Operator

The UNION operator in MySQL is used to combine the results of two or more SELECT statements into a single result set. It is used to merge the data from two or more tables with the same structure, and it removes duplicate rows from the result set.

practice.sql

```sql
SELECT City FROM Customers
UNION
SELECT City FROM Suppliers
ORDER BY City;

```

Learn with Rabbil

MySQL GROUP BY Statement

The GROUP BY statement in MySQL is used to group rows in a table based on one or more columns. It is often used in conjunction with aggregate functions, such as SUM, COUNT, AVG, MAX, or MIN, to summarize data.

practice.sql

```sql
SELECT region, year, SUM(sales_amount) FROM sales GROUP BY region, year;
```

practice.sql

```sql
SELECT customer_name, COUNT(order_id) FROM orders
GROUP BY customer_name;
```