



D:

- SQL 数据库

SQL 教程

W3School

SQL 教程

SQL 教程

Database

```
CREATE DATABASE <database_name>;
```

```
DROP DATABASE <database_name>;
```

Database

```
SHOW DATABASES;
```

```
BACKUP DATABASE <database_name>  
TO DISK = <filepath>;
```

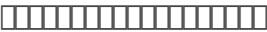
```
BACKUP DATABASE <database_name>  
TO DISK = <filepath>  
WITH DIFFERENTIAL;
```



W3School



...



int

float

double

decimal



- MySQL RDBMS datatype(size) datatype(size, d)



char(size)



int(size)



boolean



Java

-

varchar(size)



varbinary(size)



Kotlin JavaScript



- SQL Server

Unicode

n

nvarchar(max)



SQL Server

datatype



RDBMS

...

-

SQL Server decimal numeric

((,))



CREATE TABLE <table_name>(
 <column1> <datatype>,
 <column2> <datatype>,
 ...
);



DROP TABLE <table_name>;



TRUNCATE TABLE <table_name>;



```
ALTER TABLE <table_name>
ADD <column_name> <datatype>;
```



```
ALTER TABLE <table_name>
DROP COLUMN <column_name>;
```



```
ALTER TABLE <table_name>
RENAME COLUMN <old_column_name> to <new_column_name>;
```

SQL Server:

```
EXEC sp_rename "table_name.old", "new_name", "COLUMN";
```



```
--- MS SQL
ALTER TABLE <table_name>
ALTER COLUMN <column_name> <datatype>;

--- MySQL, PostgreSQL, Oracle
ALTER TABLE <table_name>
MODIFY COLUMN <column_name> <datatype>;

--- Oracle
ALTER TABLE <table_name>
MODIFY <column_name> <datatype>;
```



```
SELECT <column_name>, <column_name2>, ... FROM <table_name>;
```


Column (temp_hi+temp_lo)/2 temp_avg

```
SELECT <column_name>, <column_name2>, ... FROM <table_name>
ORDER BY <column_name>, <column_name2>, ... ASC|DESC;
```

- Column
- Column
- ASC DESC
- ORDER BY <column_name> ASC, <column_name2> DESC

RDBMS specify

- SQL Server/MS Access SELECT TOP PERCENT
- MySQL LIMIT WHERE WHERE
- Oracle 12 ORDER BY <column> FETCH First <number> ROWS ONLY WHERE WHERE ROWNUM <= <number>
- ORDER BY

```
INSERT INTO <table_name>(<column_name>, <column_name2> ...)
VALUES (<value1>, <value2>, ...);
```

- column_name Column
- Column
-

```
INSERT INTO <table_name>(<column_name>, <column_name2> ...)
VALUES
(<value1>, <value2>, ...),
(<value1>, <value2>, ...),
...;
```

NULL

..... NULL

..... IS NULL IS NOT NULL

```
SELECT <column_name>
FROM <table_name>
WHERE <column_name> IS NULL | IS NOT NULL;
```

NULL NULL

UPDATE

```
UPDATE <table_name>
SET <column1> = <value1>, <column2> = <value2>, ...
WHERE <condition>
```

- WHERE WHERE

DELETE

```
DELETE FROM <table_name> WHERE <condition>
```

- UPDATE WHERE
- Bang DROP TABLE <table_name> DELETE FROM

SQL Functions

Aggregation Functions

MAX() MIN()

```
SELECT MAX(<column_name>) | MIN(<column_name>) FROM <table_name> WHERE <condition>;
```

- AS

```
SELECT MAX(<column_name>) AS <CustomName> FROM ...;
```

- `GROUP BY` `/`

```
SELECT MAX(<column1>) AS <CustomName>, <column2>
FROM <table_name>
GROUP BY <column2>;
```



`COUNT()`

```
SELECT COUNT(<column_name>) FROM <table_name> WHERE <condition>;
```

- `(*)`
- `DISTINCT` `SELECT COUNT(DISTINCT <column_name>`
`FROM ...)`
- `WHERE` `AS` `GROUP BY`



`SUM()`

```
SELECT SUM(<column_name>) FROM <table_name> WHERE <condition>;
```

- `AS` `WHERE` `GROUP BY`



`AVG()`

```
SELECT AVG(<column_name>) FROM <table_name> WHERE <condition>;
```

- `AS` `WHERE` `GROUP BY`
-

```
SELECT * FROM <table_name> WHERE <column1> > (SELECT AVG(<column1>) FROM <table_name>)
```




Pattern (LIKE)

LIKE :

```
SELECT <column> FROM <table_name> WHERE <column> LIKE <pattern>;
```

- AND OR NOT LIKE
-



:

%	0
_	1

RDBMS specify):

[]	
^	
-	a-z
{ }	

- PostgreSQL MySQL Oracle DB
- DB System SQL script
- MS Access