

SELECT

Specific Columns `Select Col1, Col2, ..., Column`
Select Distinct `Select Distinct * from tablename`

you can show the results in your desired name by using 'AS' clause like :
`select Count(*) as CountryCount from (select Country from Customers)`

GROUP BY

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
ORDER BY column_name(s);
```

The GROUP BY statement groups rows that have the same values into summary rows, like "find the number of customers in each country".

The GROUP BY statement is often used with aggregate functions (COUNT(), MAX(), MIN(), SUM(), AVG()) to group the result-set by one or more columns.

INSERT INTO

```
INSERT INTO table_name (column1,
column2, column3, ...)
VALUES (value1, value2, value3,
...);
```

SELECT TOP

SQL SERVER `SELECT TOP number |percent column_name(s) FROM table_name WHERE condition;`

SELECT TOP (cont)

MYSQL `SELECT column_name(s) FROM table_name`
ORACLE `SELECT column_name(s) FROM table_name ORDER BY column_name NLY;`

Not all database systems support the SELECT TOP clause. MySQL supports the LIMIT clause to select a limited number of records, while Oracle uses FETCH FIRST n ROWS ONLY and ROWNUM.

BETWEEN

```
SELECT column_name(s)
FROM table_name
WHERE column_name BETWEEN
value1 AND value2;
// you can also use NOT with
BETWEEN
SELECT * FROM Products
WHERE ProductName NOT BETWEEN
value1 AND value2
```

LIKE

```
SELECT column1, column2, ...
FROM table_name
WHERE columnN LIKE pattern;
WHERE CustomerName LIKE
'a%==Finds any values that start
with " a"
```

The percent sign (%) represents zero, one, or multiple characters

The underscore sign (_) represents one, single character

You can also combine any number of conditions using AND or OR operators.

VIEW

CREATE VIEW `CREATE VIEW view_name AS SELECT column_name(s) FROM table_name WHERE condition LIMIT n`
UPDATING VIEW `CREATE OR REPLACE VIEW view_name AS SELECT column_name(s) FROM table_name ORDER BY column_name`
DROPPING VIEW `DROP VIEW view_name;`

CHECK

MYSQL - CREATE TABLE `Persons (column_name(s) CHECK (condition))`
Others - CREATE TABLE `Persons (column_name(s) CHECK (condition))`

CHECK (cont)

Multiple Columns and Naming

```
CREATE TABLE Persons (
    ID int NOT NULL,
    Age int,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int,
    CONSTRAINT CHK_PersonAge CHECK (Age>=18)
);
```

Multiple Columns and naming -

ALTER-TABLE-

DROP CHECK

- MYSQL-

DROP CHECK

-Others-

UNIQUE (cont)

```
CREATE TABLE Persons (
    ID int NOT NULL,
    Age int,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int,
    CONSTRAINT UC_Person UNIQUE
    (ID,LastName)
);
```

```
CREATE TABLE Persons (
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int,
    CONSTRAINT UC_Person UNIQUE
    (ID,LastName)
);
```

ALTER TABLE Persons ADD CONSTRAINT CHK_PersonAge CHECK (Age>=18)

-- Define multiple Unique keys --

```
CREATE TABLE Persons (
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int,
    CONSTRAINT UC_Person UNIQUE
    (ID,LastName)
);
```

ALTER TABLE Persons DROP CONSTRAINT CHK_PersonAge;

ALTER TABLE Persons DROP CHECK CHK_PersonAge;

ALTER TABLE Persons DROP CHECK CHK_PersonAge;

The UNIQUE constraint ensures that all values in a column are different.

Both the UNIQUE and PRIMARY KEY constraints provide a guarantee for uniqueness for a column or set of columns.

A PRIMARY KEY constraint automatically has a UNIQUE constraint.

However, you can have many UNIQUE constraints per table, but only one PRIMARY KEY constraint per table.

Stored procedures (User defined functions)

```
CREATE PROCEDURE procFindName
AS
```

```
sql_statement
GO;
```

```
EXEC procedure_name;
```

```
CREATE PROCEDURE procFindName
AS
```

```
SELECT * FROM Customers WHERE
City = @City AND PostalCode =
@PostalCode
```

```
GO;
```

```
EXEC Select All Customers @City
= 'London', @PostalCode = 'WA1
1DP';
```

```
END EXAMPLE <====
```

```
EXEC Select All Customers @City
= 'London', @PostalCode = 'WA1
1DP';
```

```
END EXAMPLE <====
```

UNIQUE

```
-- SQL SERVER --
```

```
CREATE TABLE Persons (
    ID int NOT NULL UNIQUE,
    LastName varchar(255)
```

```
NOT NULL,
    FirstName varchar(255),
```

```
);
```

```
);
```

```
);
```

```
);
```

```
);
```

```
);
```

```
);
```

```
);
```

```
);
```

```
);
```

```
);
```

ISNULL

```
MYSQL SELECT ProductName, Unit
    ucts;
```

```
SQL SELECT ProductName, Unit
SERVER ucts;
```

```
ORACLE SELECT ProductName, Unit
    ;
```



By Nima (nimakarimian)

Published 18th July, 2022.

Last updated 18th July, 2022.

Page 2 of 7.

Sponsored by CrosswordCheats.com

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>

ISNULL (cont)

Alternative: `SELECT ProductName, UnitPrice * (UnitsInStock + COALESCE(UnitsInStock, 0)) FROM Products;`

COALESCE() Function Works with MYSQL SQLSERVER and ORACLE

ANY & ALL

ANY `SELECT column_name(s) FROM table_name WHERE column_name operator ANY (SELECT column_name FROM table_name);`

ALL `SELECT column_name(s) FROM table_name WHERE column_name operator ALL (SELECT column_name FROM table_name);`

SOME `SELECT * FROM Products WHERE Price > SOME (SELECT Price FROM Products WHERE Price > 20);`

returns a boolean value as a result

The operator must be a standard comparison operator (=, <>, !=, >, >=, <, or <=).

WHERE keyword

`SELECT Col1, Col2 FROM tablename WHERE condition`

WHERE is for when you want to declare a specific condition for the query. In the other hand, you want to filter the records

TIP1 WHERE can be used in UPDATE, SELECT, DELETE, ETC...

UNION

`SELECT column_name(s) FROM table1`

`UNION`

`SELECT column_name(s) FROM table2;`

The UNION operator selects only distinct values by default.

To allow duplicate values, use UNION ALL:

`SELECT column_name(s) FROM table1`

`UNION ALL (SELECT Price FROM Products WHERE Price > 20);`

`UNION`

`SELECT column_name(s) FROM table2;`

The UNION operator is used to combine the result-set of two or more SELECT statements.

Every SELECT statement within UNION

must have the same number of columns

The columns must also have similar data types

The columns in every SELECT statement must also be in the same order

Aliases

Column Aliases `SELECT column_name AS alias_name FROM table_name;`

Table Aliases `SELECT column_name(s) FROM table_name AS alias_name;`

WILDCARDS

% Represents zero or more characters. `bl% finds bl, black, blue, and blob`

_ Represents a single character. `h_t finds hot, hat, and hit`

WILDCARDS (cont)

[o] Represents any single character within the brackets. `h[oa]t finds hot and hat, but not hit`

^ Represents any character not in the brackets. `h[^o]t finds hit, but not hot and hat`

- Represents any single character within the specified range. `c[a-b]t finds cat and cbt`

MIN() & MAX()

MIN() `SELECT MIN(column_name) FROM table_name;`

MAX() `SELECT MAX(column_name) FROM table_name;`

you can use MIN() and MAX() anywhere in any Query you like the examples above are just to show implementation

DELETE

`DELETE FROM table_name EVERYTHING`

`DELETE FROM table_name WITH CONDITION`



By Nima (nimakarimian)

Published 18th July, 2022.

Last updated 18th July, 2022.

Page 3 of 7.

Sponsored by CrosswordCheats.com

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>

NOTNULL

```
CREATE TABLE Persons (
    ID int NOT NULL,
    LastName varchar(255)
NOT NULL,
    FirstName varchar(255)
NOT NULL,
    Age int
);
```

```
ALTER TABLE Persons
MODIFY Age int NOT NULL;
```

To create a NOT NULL constraint on the "Age" column when the "Persons" table is already created, use the above Code

DATE

DATE - format YYYY-MM-DD

DATETIME - format: YYYY-MM-DD

HH:MI:SS

TIMESTAMP - format: YYYY-MM-DD

HH:MI:SS

FOR MYSQL= ==>

YEAR - format YYYY or YY

<===

FOR SQLSERVER ==>

TIMESTAMP - format: a unique number

<===

SELECT INTO (copy)

```
SELECT column1, column2,
column3, ...
INTO newtable [IN externaldb]
FROM oldtable
WHERE condition;
```

```
SELECT * INTO Customers Backup
2017 IN 'Backup.mdb'
FROM Customers;
```

DEFAULT

```
DEFAULT - CREATE TABLE Orders (
CREATE TABLE-
```

```
ALERTABLE ALTER TABLE Persons ALTER
-MYSQL-
```

```
ALERTABLE ALTER TABLE Persons ADD
-SQLSE-
```

```
RVER-
```

```
ALERTABLE ALTER TABLE Persons MODIFY
E-Oracle-
```

```
DROPTABLE ALTER TABLE Persons ALTER
BLE-MYSQL-
```

```
DROPTABLE ALTER TABLE Persons ALTER
-Others-
```

CASE

```
CASE
    WHEN condition1 THEN
result1
    WHEN condition2 THEN
result2
    WHEN conditionN THEN
resultN
    ELSE result
END;
```

Database

```
CREATE CREATE DATABASE testDB;
SHOW SHOW DATABASES
DROP DROP DATABASE databasename;
```

Database (cont)

```
BACKUP BACKUP DATABASE databasename;
Differential;
BACKUP SET DEFAULT 'Sandnes';
```

ALTER table

```
ADD Column ALTER TABLE table_name
DROP ALTER TABLE table_name
Column;
```

```
ALTER/ ALTER TABLE table_name
MODIFY DROP DEFAULT;
```

```
COLUMN
SQLSERVER ALTER TABLE table_name
DROP DEFAULT;
```

```
ALTER/ ALTER TABLE table_name
MODIFY ;
```

```
COLUMN
MYSQL
```

```
ALTER/ ALTER TABLE table_name
MODIFY
```

```
COLUMN
ORACLE
```

PRIMARY KEY

```
--MYSQL--
CREATE TABLE Persons (
    ID int NOT NULL,
    LastName varchar(255)
NOT NULL,
    FirstName varchar(255),
    Age int,
    PRIMARY KEY (ID)
```



By Nima (nimakarimian)

Published 18th July, 2022.

Last updated 18th July, 2022.

Page 4 of 7.

Sponsored by CrosswordCheats.com

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>

PRIMARY KEY (cont)

```
> );
--SQLSERVER ORACLE--
CREATE TABLE Persons (
  ID int NOT NULL PRIMARY KEY,
  LastName varchar(255) NOT NULL,
  FirstName varchar(255),
  Age int
);
--Defining multiple Columns as PK--
CREATE TABLE Persons (
  ID int NOT NULL,
  LastName varchar(255) NOT NULL,
  FirstName varchar(255),
  Age int,
  CONSTRAINT PK_Person PRIMARY
KEY (ID,LastName)
);
In the example above there is only ONE
PRIMARY KEY (PK_Person).
However, the VALUE of the primary key is
made up of
TWO COLUMNS (ID + LastName).
```

SUM() & AVG() & COUNT ()

```
AVG() SELECT AVG(column_name) FROM table_name WHERE condition;
SUM() SELECT SUM(column_name) FROM table_name WHERE condition;
COUNT() SELECT COUNT(column_name) FROM table_name WHERE condition;
```

IN

```
SELECT column_name(s)
FROM table_name
WHERE column_name IN (value1,
value2, ...);
OR
SELECT column_name(s)
FROM table_name
WHERE column_name IN (SELECT
STATEMENT);
```

The IN operator allows you to specify multiple values in a WHERE clause.

The IN operator is a shorthand for multiple OR conditions.

UPDATE

```
UPDATE table_name SET column1 = value1, column2 = value2, ...
WITH
CONDITION
```

Be careful when updating records. If you omit the WHERE clause, ALL records will be updated!

JOINS

```
INNER JOIN table1 INNER JOIN table2 ON table1.column_name = table2.column_name;
LEFT JOIN table1 LEFT JOIN table2 ON table1.column_name = table2.column_name;
RIGHT JOIN table1 RIGHT JOIN table2 ON table1.column_name = table2.column_name;
FULL JOIN table1 FULL OUTER JOIN table2 ON table1.column_name = table2.column_name;
```

JOINS (cont)

```
SELF SELECT column_name(s) FROM
JOIN n;
```

ORDER BY

```
SELECT column1, column2, ...
FROM table_name
ORDER BY column1, column2, ...
ASC|DESC;
```

Constraints

```
CREATE TABLE table_name (
  column1 datatype
  constraint,
  column2 datatype
  constraint,
  column3 datatype
  constraint,
  ...
);
```

Constraints can be specified when the table is created with the CREATE TABLE statement, or after the table is created with the ALTER TABLE statement.

NOT NULL
UNIQUE
PRIMARY KEY
FOREIGN KEY
CHECK
DEFAULT
CREATE INDEX

PRIMARY KEY on ALTER TABLE

```
ALTERTABLE ALTER TABLE Persons ADD
SIMPLE
```

```
ALTERTABLE ALTER TABLE Persons ADD
multiple or );
naming
```

```
DROP PK - ALTER TABLE Persons DI
MYSQL-
```

```
DROP PK - ALTER TABLE Persons DI
Others-
```



By Nima (nimakarimian)

Published 18th July, 2022.

Last updated 18th July, 2022.

Page 5 of 7.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>

FOREIGN KEY	FOREIGN KEY (cont)	TABLE (cont)
<p>MYSQL - CREATE TABLE -</p> <pre>CREATE TABLE Orders (OrderID int NOT NULL PRIMARY KEY, OrderNumber int NOT NULL, PersonID int NOT NULL, OrderDate datetime);</pre>	<p>FOREIGN KEY - ALTER TABLE -</p> <pre>ALTER TABLE Orders ADD CONSTRAINT FK_Person_Order FOREIGN KEY (PersonID) REFERENCES Persons (PersonID);</pre>	<p>TRUNCATE TABLE</p> <pre>TRUNCATE TABLE table_name;</pre> <p>The TRUNCATE TABLE statement is used to delete the data inside a table, but not the table itself.</p>
<p>Others - CREATE TABLE -</p> <pre>CREATE TABLE Orders (OrderID int NOT NULL PRIMARY KEY, OrderNumber int NOT NULL, PersonID int NOT NULL, OrderDate datetime);</pre>	<p>Multiple Rows or FK with Name - ALTER TABLE -</p> <pre>ALTER TABLE Orders ADD CONSTRAINT FK_Person_Order FOREIGN KEY (PersonID) REFERENCES Persons (PersonID);</pre>	<p>INSERT INTO</p> <pre>INSERT INTO table2 SELECT * FROM table1 WHERE condition;</pre> <p>The INSERT INTO SELECT statement copies data from one table and inserts it into another table.</p> <p>The INSERT INTO SELECT statement requires that the data types in source and target tables match.</p>
<p>Multiple Rows or FK with Name - DROP FK -MYSQL-</p> <pre>ALTER TABLE Orders DROP FOREIGN KEY FK_Person_Order;</pre>	<p>Drop FK -Others-</p> <pre>ALTER TABLE Orders DROP CONSTRAINT FK_Person_Order;</pre>	<p>INDEX</p> <pre>CREATE INDEX index_name ON table_name (column1, column2);</pre> <p>CREATE INDEX - CREATE UNIQUE INDEX - UNIQUE</p> <pre>CREATE UNIQUE INDEX index_name ON table_name (column1);</pre> <p>DROP - SQLSE-RVER-</p> <pre>DROP INDEX table_name.</pre> <p>DROP - MYSQL-</p> <pre>ALTER TABLE table_name DROP INDEX index_name;</pre> <p>DROP - ORACLE-</p> <pre>DROP INDEX index_name;</pre>
<p>TABLE</p> <pre>CREATE TABLE table_name (column1 datatype, column2 datatype);</pre> <p>DROP</p> <pre>DROP TABLE table_name;</pre>	<p>CREATE TABLE new_table_name AS</p> <pre>CREATE TABLE new_table_name AS SELECT * FROM table_name;</pre> <p>from another Table</p>	



By Nima (nimakarimian)

Published 18th July, 2022.

Last updated 18th July, 2022.

Page 6 of 7.

Sponsored by [CrosswordCheats.com](https://crosswordcheats.com)

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>

AUTO INCREMENT

```
CREATE TABLE Persons (
  FirstName varchar(10),
  LastName varchar(255)
);
```

```
ALTER TABLE Persons AUTO_INCREMENT = 10;
```

```
CREATE TABLE Persons (
  FirstName varchar(10),
  LastName varchar(255)
);
```

```
CREATE SEQUENCE seq_persons
  INCREMENT BY 1
  CACHE 10;
```

```
INSERT INTO Persons (FirstName, LastName)
VALUES (seq_persons.nextval, 'Larson');
```

use sequence in oracle

EXISTS

```
SELECT column_name
FROM table_name
WHERE EXISTS
```

```
(SELECT column_name
FROM table_name
WHERE condition);
```

The EXISTS operator is used to test for the existence of any record in a subquery.

The EXISTS operator returns TRUE if the

subquery returns one or more records.

HAVING

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
HAVING condition
ORDER BY column_name(s);
```

The HAVING clause was added to SQL because the WHERE keyword cannot be used with aggregate functions.



By Nima (nimakarimian)

Published 18th July, 2022.

Last updated 18th July, 2022.

Page 7 of 7.

Sponsored by [CrosswordCheats.com](https://crosswordcheats.com)

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>