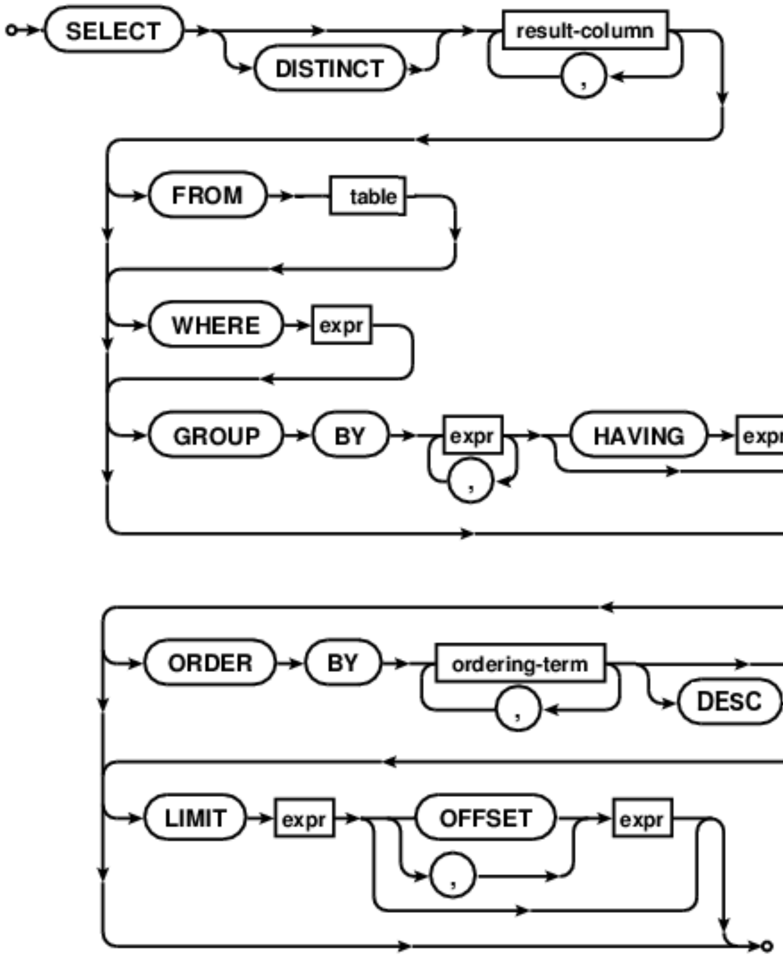
SQLite basic SELECT syntax



The `FROM` clause can specify a single table or it can join several tables:
`table1 JOIN table2 ON table1.col1 = table2.col2`

This creates a virtual table where rows from both tables are combined if the specified columns have the same value.

result-column can list one or more of:

- The name of a column
- An aggregation function applied to a column, eg. `SUM(col)`
    - `SUM, COUNT, MIN, MAX, AVG`
- A math expression with column names and numbers, eg. `col1 * col2 / 2`
- `*` to return all columns
- `COUNT(*)` to count the rows
- … `||` … concatenates text

`WHERE` clauses are row filters built from the following operators:

- … `=` …
- comparisons also include `!=   >   <   >=   <=`
- … `BETWEEN` … `AND` …
- `NOT` …
- (…) parenthesis are used for expressions within expressions
- … `IS NULL`
- … `IS NOT NULL`
- … `AND` …
- … `OR` …
- … `IN (…,…,…)` tests whether a value is in a set
- … `LIKE "…"`
  where the quoted pattern can have **%** to match any sequence of characters and **_** to match any single character.

Remember that column names don't have quotes but text constants do.
For example: `WHERE CustomerName = "Jane"`

`GROUP BY X` will combine rows in the result if they have the same value for **`X`** (usually the name of a column).

`HAVING` is a filter like `WHERE`, but it applies after grouping.

Subqueries can be used to fill in a *single value* or a *set of values*.
This allows you to use data from more than one table.
Enclose the subquery with parenthesis `(SELECT …)`

SQL Examples

Recipes.sqlite

Print an alphabetically sorted list of ingredients

```
SELECT IngredientName FROM Ingredients
  ORDER BY IngredientName;
```

How many times is butter used as an ingredient?

```
SELECT COUNT(*) FROM Recipe_Ingredients
  WHERE IngredientID=47;
```

How many ingredients are in the Yorkshire Pudding recipe?

```
SELECT COUNT(*) FROM Recipe_Ingredients
  WHERE RecipeID=10;
```

What percentage of ingredients are vegetarian?

```
SELECT 100.0 * COUNT(*)/
  (SELECT COUNT(*) FROM Ingredients) FROM Ingredients
  WHERE IngredientClassID NOT IN (2, 10);
```

How many recipes have multi-word names? Nine-letter names?

```
SELECT COUNT(*) FROM Recipes
  WHERE RecipeTitle LIKE "% %";

SELECT COUNT(*) FROM Recipes
  WHERE RecipeTitle LIKE "_________";
SELECT COUNT(*) FROM Recipes
  WHERE LENGTH(RecipeTitle) = 9;
```

What is the name of the recipe with the most ingredients?

```
SELECT RecipeTitle, COUNT(*) AS numIngredients
FROM
  Recipe_Ingredients JOIN Recipes
    ON Recipes.RecipeID
       = Recipe_Ingredients.RecipeID
ORDER BY numIngredients DESC
LIMIT 1
GROUP BY Recipes.RecipeID
```

SalesOrders.sqlite

List all customers in a west coast state (CA, OR, WA).

```
SELECT * FROM Customers
  WHERE CustState IN ("CA","OR", "WA");
```

Count the unique customer area codes in California (CA).

```
SELECT COUNT(DISTINCT CustAreaCode) FROM Customers
  WHERE CustState = "CA";
```

What is the full address of customer John Viescas?

```
SELECT CustStreetAddress || " " || CustCity || " "
  || CustState || " " || CustZipCode AS FullAddress
  FROM Customers WHERE
  CustFirstName = "John" AND CustLastName = "Viescas";
```

What is the value of the product inventory on hand? Bike inventory?

```
SELECT SUM(RetailPrice * QuantityOnHand) FROM Products;
SELECT SUM(RetailPrice * QuantityOnHand) FROM Products
  WHERE CategoryID=2;
```

SchoolScheduling.sqlite

What is the mean average classroom capacity? Median?

```
SELECT AVG(Capacity) FROM Class_Rooms;
SELECT Capacity FROM Class_Rooms ORDER BY Capacity
  LIMIT 1 OFFSET (SELECT COUNT(*)/2 FROM Class_Rooms);
```

How much classroom capacity is there in each building?

```
SELECT BuildingCode, SUM(Capacity) FROM Class_Rooms
  GROUP BY BuildingCode;
```

How many classes does each instructor teach on average?

```
SELECT AVG(NumClasses) FROM
  (SELECT COUNT(*) AS NumClasses FROM Faculty_Classes
  GROUP BY StaffID);
```

What is the average grade earned by students?

```
SELECT AVG(Grade) FROM Student_Schedules
  WHERE Grade > 0;
```