

Contents 🔄 ⚙️

- Lesson Goals
- Lesson Vocabulary
- What Are the Elements of the SQL Select Statement?
- So What Does Each Element Do?
- ▼ Now What?
 - Basic `SELECT` Statement
 - `SELECT` Statement
 - `DISTINCT` Statement
 - ▼ The `WHERE` Clause
 - Use More Operators
 - `WHERE` Clause
 - Create Alias Using `AS`

SQL Basic Statements

Lesson Goals

- Use the `SELECT` statement to read data.
- Introduce the `DISTINCT` keyword and `WHERE` clause for filtering.
- Understand MySQL operators.
- Create aliases using `AS`.

Lesson Vocabulary

- **A Query** is a request that returns information or records. A query can contain multiple clauses and subclauses, each containing one or more SQL keywords, but it can also be as simple as a `SELECT *` statement that returns all fields followed by a `FROM table_name` clause indicating the table you want the data to come from.
- **A Statement** is any valid piece of code that is executable by a RDBMS.
 - `SELECT`, `SELECT DISTINCT`
- **A Clause** is a subsection of a query that contains at least one keyword and the accompanying information, like fields or tables, to be used with the keyword.
 - `FROM`, `WHERE`
- **A Keyword** is a reserved word that performs an operation.
 - `SELECT`, `DISTINCT`, `FROM`, `WHERE`, `LIKE`
- **Filtering** can narrow the result set returned by your query.

What Are the Elements of the SQL Select Statement?

Contents ↻ ⚙

- Lesson Goals
- Lesson Vocabulary
- What Are the Element
- So What Does Each E
- ▼ Now What?
 - Basic SELECT Statement
- DISTINCT Statement
 - ▼ The WHERE Clause
 - Use More Operat
 - WHERE Clause
 - Create Alias Using AS

Faith Kane
2021

THE BIG 6 ELEMENTS OF A SQL SELECT STATEMENT →

Necessary

Optional

- 1 **SELECT** COLUMN_NAME
- 2 **FROM** TABLE_NAME
- 3 **WHERE** LOGICAL_CONDITION
- 4 **GROUP BY** COLUMN_NAME
- 5 **HAVING** LOGICAL_CONDITION
- 6 **ORDER BY** COLUMN_NAME

So What Does Each Element Do?

Faith Kane
2021

THE BIG 6 ELEMENTS OF A SQL SELECT STATEMENT →

Necessary

Optional

- 1 **SELECT** IDENTIFIES COLUMN(S)
- 2 **FROM** IDENTIFIES TABLE
- 3 **WHERE** RECORD-FILTERING CRITERIA
- 4 **GROUP BY** SPECIFIES HOW TO GROUP DATA
- 5 **HAVING** GROUP-FILTERING CRITERIA
- 6 **ORDER BY** SPECIFIES ORDER OF RESULTS

Contents 🔄 ⚙️

- Lesson Goals
- Lesson Vocabulary
- What Are the Elements of a SELECT Statement?
- So What Does Each Element Do?
- ▼ Now What?
 - Basic SELECT Statement
 - DISTINCT Statement
 - ▼ The WHERE Clause
 - Use More Operators
 - WHERE Clause
 - Create Alias Using AS

Now What?

Basic SELECT Statement

```
-- Select the database.
USE fruits_db;

-- Inspect the columns and data types from a table.
DESCRIBE fruits;

-- Another way to Inspect the columns and data types from a table.
SHOW COLUMNS
FROM fruits;

-- Return all of the rows and columns from a table.
SELECT *
FROM fruits;

-- Select specific column(s) and all of the rows from those columns.
SELECT name
FROM fruits;

SELECT name, quantity
FROM fruits;
```

Contents 🔄 ⚙️

- Lesson Goals
- Lesson Vocabulary
- What Are the Elements?
- So What Does Each Element Do?
- ▼ Now What?
 - Basic SELECT Statement
 - DISTINCT Statement
 - ▼ The WHERE Clause
 - Use More Operators
 - WHERE Clause
 - Create Alias Using AS

SELECT DISTINCT Statement

```
-- Use chipotle database to demo a db with duplicates.
USE chipotle;

-- Inspect the columns and data types from a table.
DESCRIBE orders;

-- Return all of the rows and columns from a table. (4622 records returned)
SELECT *
FROM orders;

/*
Select specific column(s) and all of the rows from those columns.
(4622 records returned)
*/

SELECT item_name
FROM orders;

SELECT
    item_name,
    item_price
FROM orders;

/*
Return only the unique values from a column using the DISTINCT keyword
(50 records returned)
*/

SELECT DISTINCT item_name
FROM orders;
```

The WHERE Clause

Contents 🔄 ⚙️

- Lesson Goals
- Lesson Vocabulary
- What Are the Elements?
- So What Does Each Element Do?
- ▼ Now What?
 - Basic SELECT Statement
 - SELECT
 - DISTINCT Statement
 - ▼ The WHERE Clause
 - Use More Operators
 - WHERE Clause
 - Create Alias Using AS

```
/*  
Filter so that only records with the value 'Chicken Bowl' in item_name  
are returned.  
(726 records returned)  
*/
```

```
SELECT *  
FROM orders  
WHERE item_name = 'Chicken Bowl';
```

-- Why doesn't the query below run? Never forget this lesson!

```
SELECT *  
FROM orders  
WHERE item_price = $4.45;
```

```
/*  
Filter using the primary key column; only one record will be returned  
because the value must be unique.  
*/
```

```
SELECT *  
FROM orders  
WHERE id = 15;
```

Use More Operators with WHERE Clause

Contents 🔄 ⚙️

- Lesson Goals
- Lesson Vocabulary
- What Are the Element
- So What Does Each E
- ▼ Now What?
 - Basic SELECT
- Statement
 - SELECT
- DISTINCT Statement
 - ▼ The WHERE Clause
 - Use More Operat
 - WHERE Clause
 - Create Alias Using
- AS

```
-- Filter using a WHERE clause with the BETWEEN & AND operators.  
(Returns 39 records)
```

```
SELECT *  
FROM orders  
WHERE quantity BETWEEN 3 AND 5;
```

```
-- Filter using a WHERE statement >, <, <> operators.
```

```
-- (returns 870 records)
```

```
SELECT *  
FROM orders  
WHERE order_id > 1500;
```

```
-- (returns 267 records)
```

```
SELECT *  
FROM orders  
WHERE quantity <> 1;
```

Create Alias Using AS

```
-- Create an alias for a column using the AS keyword. (Returns 2  
67 records)
```

```
SELECT  
    item_name AS 'Multiple Item Order',  
    quantity AS Number  
FROM orders  
WHERE quantity >= 2;
```

```
/*
```

```
Notice that if I have spaces in my column alias, I have to put i  
t in single quotes.
```

```
If I do not have a space in my colum alias, I do not have to put  
it in quotes.
```

```
*/
```