Databases & SQL

Hello!

I am Eslam Ahmed

I am a software engineer.

You can find me at jeksogsa@gmail.com

Hello!

I am Eman Ehab

I am a ML research engineer.

You can find me at
emanehab.ieee@gmail.com

Agenda

- What is RDBMS.
- Tables, Columns and Data types
- How to design a database.
- One-To-Many & Many-To-Many Relationships.
- Examples: FB, Youtube, Souq.com
- SQL
- CRUD
- Selecting data
- Filtering data
- Ordering data
- Limiting data
- Aggregate Functions
- Joining tables
- Grouping data
- Subqueries
- Inserting new data
- Updating data
- Deleting data
- Python and SQLite
- DB Browser for SQLite

Agenda

- **What is RDBMS.**
- Tables, Columns and Data types
- How to design a database.
- One-To-Many & Many-To-Many Relationships.
- Examples: FB, Youtube, Souq.com
- SQL
- CRUD
- Selecting data
- Filtering data
- Ordering data
- Limiting data
- Aggregate Functions
- Joining tables
- Grouping data
- Subqueries
- Inserting new data
- Updating data
- Deleting data
- Python and SQLite
- DB Browser for SQLite

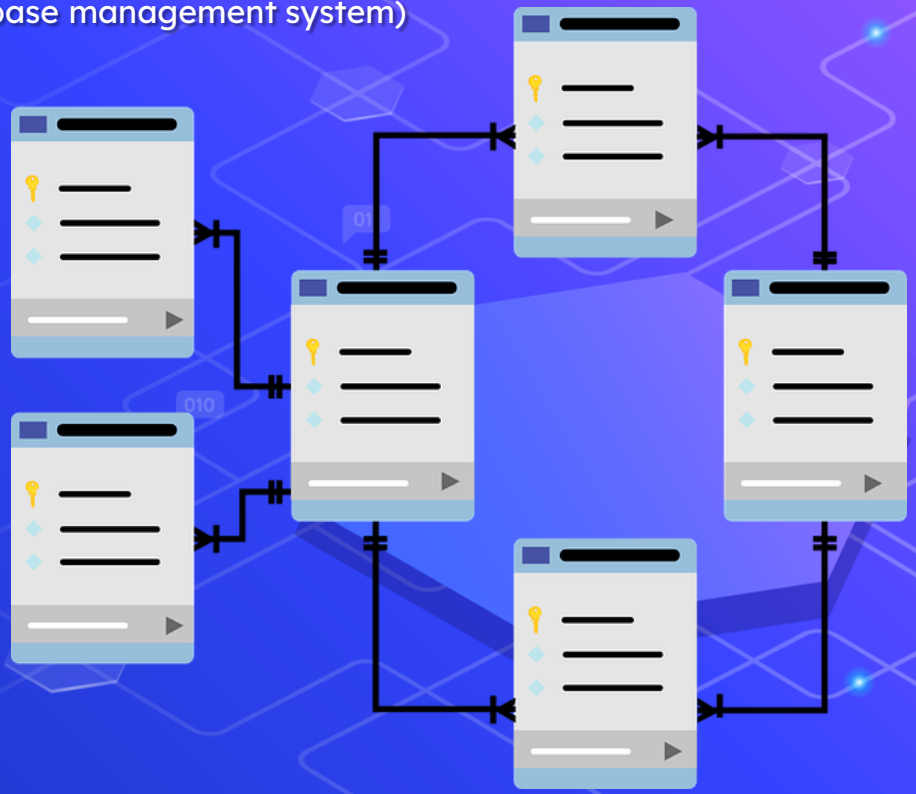
What is RDBMS. (relational database management system)

File systems made for storing and manipulating relational data.

Data organized in tabular form each table represents an entity like users, products, categories, etc…

Each table consists of columns that represent entity properties, each column can be any data type like numbers, text, etc…

SQL is an easy language to connect with database and query the data from it easily.



Agenda

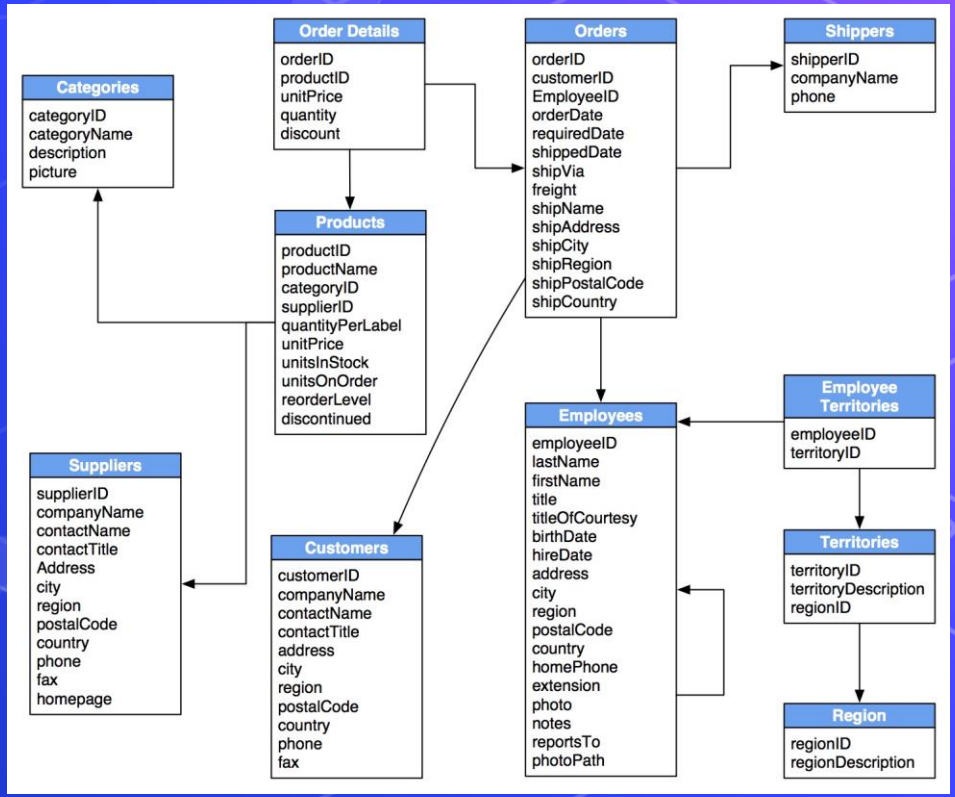
- What is RDBMS.
- **Tables, Columns and Data types**
- How to design a database.
- One-To-Many & Many-To-Many Relationships.
- Examples: FB, Youtube, Souq.com
- SQL
- CRUD
- Selecting data
- Filtering data
- Ordering data
- Limiting data
- Aggregate Functions
- Joining tables
- Grouping data
- Subqueries
- Inserting new data
- Updating data
- Deleting data
- Python and SQLite
- DB Browser for SQLite

Tables, Columns and Data types

Tables
Users, posts, comments, products, ...

Columns
Users table have columns like age, gender, username, email, is_online,...

Data Types
Age is Numeric, username is TEXT
Gender is CHAR, is_online is BOOL



Agenda

- What is RDBMS.
- Tables, Columns and Data types
- **How to design a database.**
- One-To-Many & Many-To-Many Relationships.
- Examples: FB, Youtube, Souq.com
- SQL
- CRUD
- Selecting data
- Filtering data
- Ordering data
- Limiting data
- Aggregate Functions
- Joining tables
- Grouping data
- Subqueries
- Inserting new data
- Updating data
- Deleting data
- Python and SQLite
- DB Browser for SQLite

How to design a database.

1) Analyze the system and divide it to its main entities (tables).

---

for example we make an ecommerce system so the main entities tables are Categories, Products, Orders, Customers, etc...

2) For each table define what its properties (columns) are and what is their data types.

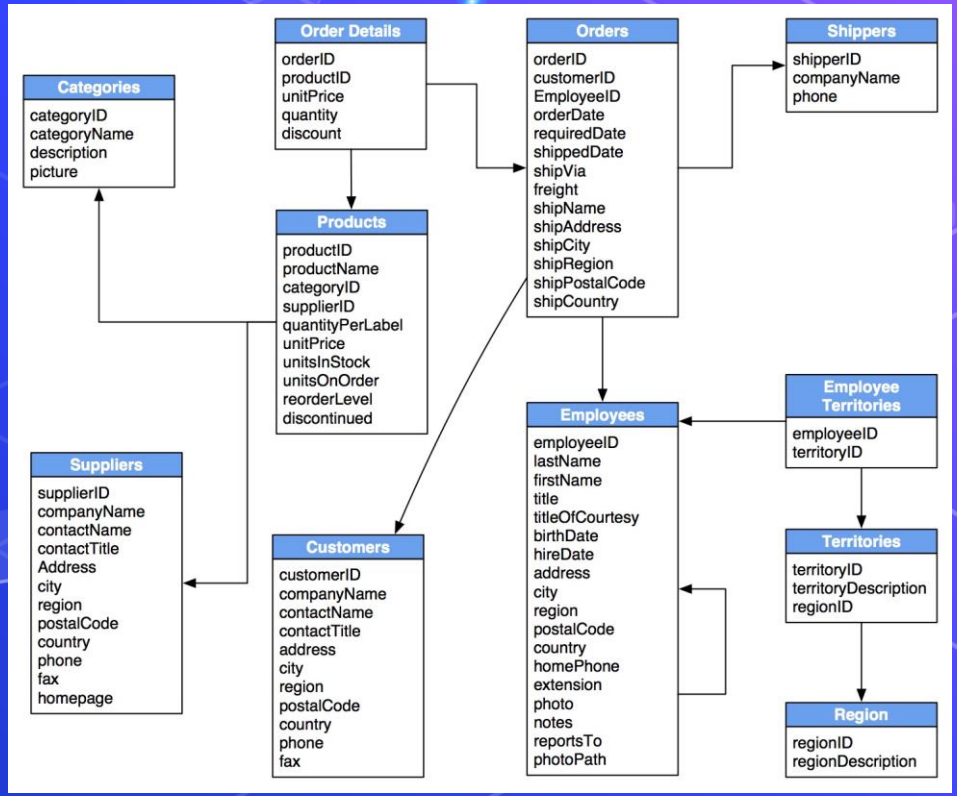
---

for example customers table have name, city ,phone and products table have price, name, etc.., the most important thing don't forget to add the unique auto increment **id** column (Primary Key) because it's the row identifier.

3) Define tables relationships and based on the relationship define the foreign keys.

---

for example one-to-many relationship between the products and categories tables so we define the **CategoryID** column (Foreign Key) in the products table, another relation is many-to-many relation between orders and products tables so we make Order Details table that contain the foreign keys



Agenda

- What is RDBMS.
- Tables, Columns and Data types
- How to design a database.
- One-To-Many & Many-To-Many Relationships.
- Examples: FB, Youtube, Souq.com
- SQL
- CRUD
- Selecting data
- Filtering data
- Ordering data
- Limiting data
- Aggregate Functions
- Joining tables
- Grouping data
- Subqueries
- Inserting new data
- Updating data
- Deleting data
- Python and SQLite
- DB Browser for SQLite

One-To-Many & Many-To-Many Relationships

Products --->

CategoryID is a foreign key that point to the CategoryID primary key in categories table

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18
2	Chang	1	1	24 - 12 oz bottles	19
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10
4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	22
5	Chef Anton's Gumbo Mix	2	2	36 boxes	21.35
6	Grandma's Boysenberry Spread	3	2	12 - 8 oz jars	25
7	Uncle Bob's Organic Dried Pears	3	7	12 - 1 lb pkgs.	30
8	Northwoods Cranberry Sauce	3	2	12 - 12 oz jars	40
9	Mishi Kobe Niku	4	6	18 - 500 g pkgs.	97
10	Ikura	4	8	12 - 200 ml jars	31

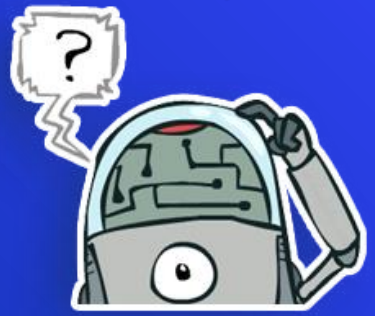
CategoryID	CategoryName	Description
1	Beverages	Soft drinks, coffees, teas, beers, and ales
2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings
3	Confections	Desserts, candies, and sweet breads
4	Dairy Products	Cheeses
5	Grains/Cereals	Breads, crackers, pasta, and cereal
6	Meat/Poultry	Prepared meats
7	Produce	Dried fruit and bean curd
8	Seafood	Seaweed and fish

<--- Categories

CategoryID is a primary key

One-To-Many & **Many-To-Many** Relationships

Products --->



How can we represent the relationship ?!

Orders --->

ProductID	ProductName	SupplierID	CategoryID	Unit
1	Chais	1	1	10 box
2	Chang	1	1	24 - 12
3	Aniseed Syrup	1	2	12 - 55
4	Chef Anton's Cajun Seasoning	2	2	48 - 6
5	Chef Anton's Gumbo Mix	2	2	36 box
6	Grandma's Boysenberry Spread	3	2	12 - 8
7	Uncle Bob's Organic Dried Pears	3	7	12 - 1
8	Northwoods Cranberry Sauce	3	2	12 - 12
9	Mishi Kobe Niku	4	6	18 - 50
10	Ikura	4	8	12 - 20

OrderID	CustomerID	EmployeeID	OrderDate
10248	90	5	1996-07-04
10249	81	6	1996-07-05
10250	34	4	1996-07-08
10251	84	3	1996-07-08
10252	76	4	1996-07-09
10253	34	3	1996-07-10
10254	14	5	1996-07-11
10255	68	9	1996-07-12

One-To-Many & **Many-To-Many** Relationships

Order Details

OrderDetailID	OrderID	ProductID	Quantity
1	10248	1	12
2	10248	42	10
3	10248	72	5
4	10249	14	9
5	10249	51	40
6	10250	41	10
7	10250	1	35
8	10250	65	15
9	10251	1	6
10	10251	57	15

Products

ProductID	ProductName
1	Chais
2	Chang
3	Aniseed Syrup
4	Chef Anton's Cajun Seasoning
5	Chef Anton's Gumbo Mix
6	Grandma's Boysenberry Spread
7	Uncle Bob's Organic Dried Pea
8	Northwoods Cranberry Sauce
9	Mishi Kobe Niku
10	Ikura

Orders

OrderID	CustomerID
10248	90
10249	81
10250	34
10251	84
10252	76
10253	34
10254	14

We make a new table that have both foreign keys

Agenda

- What is RDBMS.
- Tables, Columns and Data types
- How to design a database.
- One-To-Many & Many-To-Many Relationships.
- **Examples: FB, Youtube, Souq.com**
- SQL
- CRUD
- Selecting data
- Filtering data
- Ordering data
- Limiting data
- Aggregate Functions
- Joining tables
- Grouping data
- Subqueries
- Inserting new data
- Updating data
- Deleting data
- Python and SQLite
- DB Browser for SQLite

Examples: FB, Youtube, Souq.com

Agenda

- What is RDBMS.
- Tables, Columns and Data types
- How to design a database.
- One-To-Many & Many-To-Many Relationships.
- Examples: FB, Youtube, Souq.com
- **SQL**
- CRUD
- Selecting data
- Filtering data
- Ordering data
- Limiting data
- Aggregate Functions
- Joining tables
- Grouping data
- Subqueries
- Inserting new data
- Updating data
- Deleting data
- Python and SQLite
- DB Browser for SQLite

SQL

SQL or Structured Query Language used to communicate with the database by executing commands for creating tables and columns and also reading, filtering, ordering, inserting, updating, deleting data and more.

You can try SQL on bunch of examples from this link

https://www.w3schools.com/sql/default.asp

https://www.sqlitetutorial.net

Agenda

- What is RDBMS.
- Tables, Columns and Data types
- How to design a database.
- One-To-Many & Many-To-Many Relationships.
- Examples: FB, Youtube, Souq.com
- SQL
- **CRUD**
- Selecting data
- Filtering data
- Ordering data
- Limiting data
- Aggregate Functions
- Joining tables
- Grouping data
- Subqueries
- Inserting new data
- Updating data
- Deleting data
- Python and SQLite
- DB Browser for SQLite

CRUD

C R U D

Create Read Update Delete

Agenda

- What is RDBMS.
- Tables, Columns and Data types
- How to design a database.
- One-To-Many & Many-To-Many Relationships.
- Examples: FB, Youtube, Souq.com
- SQL
- CRUD
- **Selecting data**
- Filtering data
- Ordering data
- Limiting data
- Aggregate Functions
- Joining tables
- Grouping data
- Subqueries
- Inserting new data
- Updating data
- Deleting data
- Python and SQLite
- DB Browser for SQLite

Selecting data

- Select all columns from a table
- Select specific columns from a table

```
1 SELECT * FROM Customers
```

Selecting data

- Select all columns from a table
- Select specific columns from a table

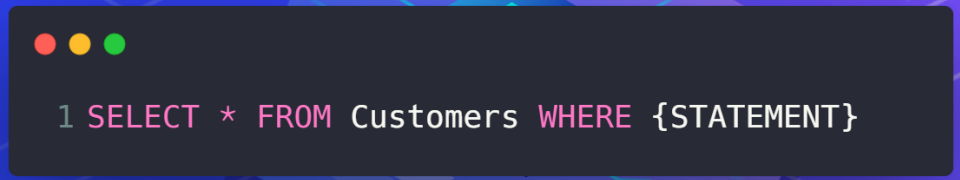
```sql
SELECT CustomerName, City FROM Customers
```

Agenda

- What is RDBMS.
- Tables, Columns and Data types
- How to design a database.
- One-To-Many & Many-To-Many Relationships.
- Examples: FB, Youtube, Souq.com
- SQL
- CRUD
- Selecting data
- **Filtering data**
- Ordering data
- Limiting data
- Aggregate Functions
- Joining tables
- Grouping data
- Subqueries
- Inserting new data
- Updating data
- Deleting data
- Python and SQLite
- DB Browser for SQLite

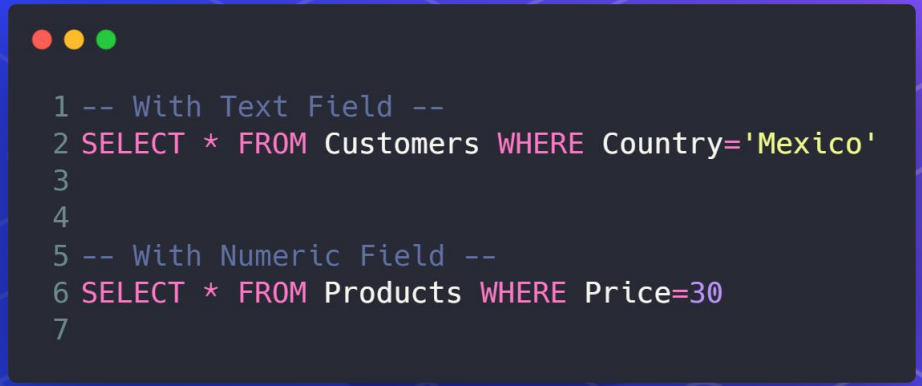
Filtering data

- WHERE Clause
- Filtering with Text vs Numeric Fields
- Comparison operators (>, <, >=, <=, =, <>)
- BETWEEN
- IN
- LIKE
- Logic operators (AND, OR, NOT)
- IS NULL, IS NOT NULL



Filtering data

- WHERE Clause
- Filtering with Text vs Numeric Fields
- Comparison operators (>, <, >=, <=, =, <>)
- BETWEEN
- IN
- LIKE
- Logic operators (AND, OR, NOT)
- IS NULL, IS NOT NULL



```sql
-- With Text Field --
SELECT * FROM Customers WHERE Country='Mexico'


-- With Numeric Field --
SELECT * FROM Products WHERE Price=30

```

Filtering data

- WHERE Clause
- Filtering with Text vs Numeric Fields
- Comparison operators (=, !=, >, <, >=, <=)
- BETWEEN
- IN
- LIKE
- Logic operators (AND, OR, NOT)
- IS NULL, IS NOT NULL

```sql
SELECT * FROM Customers WHERE Country = 'Mexico'

SELECT * FROM Customers WHERE Age != 25

SELECT * FROM Products WHERE Price > 30

SELECT * FROM Books WHERE Rate >= 3

SELECT * FROM Products WHERE Price < 30

SELECT * FROM Books WHERE Rate <= 3
```

Filtering data

- WHERE Clause
- Filtering with Text vs Numeric Fields
- Comparison operators (=, !=, >, <, >=, <=)
- BETWEEN
- IN
- LIKE
- Logic operators (AND, OR, NOT)
- IS NULL, IS NOT NULL

```sql
SELECT * FROM Products
WHERE Price BETWEEN 10 AND 20


SELECT * FROM Products
WHERE Price NOT BETWEEN 10 AND 20
```

Filtering data

- WHERE Clause
- Filtering with Text vs Numeric Fields
- Comparison operators (=, !=, >, <, >=, <=)
- BETWEEN
- IN
- LIKE
- Logic operators (AND, OR, NOT)
- IS NULL, IS NOT NULL

```sql
SELECT * FROM Customers
WHERE Country IN ('Germany', 'France', 'UK')


SELECT * FROM Products
WHERE Price NOT IN (10, 30, 50)
```

Filtering data

- WHERE Clause
- Filtering with Text vs Numeric Fields
- Comparison operators (=, !=, >, <, >=, <=)
- BETWEEN
- IN
- **LIKE**
- Logic operators (AND, OR, NOT)
- IS NULL, IS NOT NULL

```sql
-- Select rows that start with 'a'
SELECT * FROM Customers WHERE CustomerName LIKE 'a%'


-- Select rows that start with any letter except 'a'
SELECT * FROM Customers WHERE CustomerName NOT LIKE 'a%'


-- Select rows that end with 'a'
SELECT * FROM Customers WHERE CustomerName LIKE '%a'


-- Select rows that have 'or' in any position
SELECT * FROM Customers WHERE CustomerName LIKE '%or%'


-- Select rows that start with 'a'and end with 'o'
SELECT * FROM Customers WHERE CustomerName LIKE 'a%o'
```

Filtering data

- WHERE Clause
- Filtering with Text vs Numeric Fields
- Comparison operators (=, !=, >, <, >=, <=)
- BETWEEN
- IN
- LIKE
- Logic operators (AND, OR, NOT)
- IS NULL, IS NOT NULL

```sql
SELECT * FROM Customers
WHERE Country='Germany' AND City='Berlin'


SELECT * FROM Customers
WHERE City='Berlin' OR City='München'


SELECT * FROM Products
WHERE Price NOT BETWEEN 10 AND 30


-- combine multi logic operator in one statment
SELECT * FROM Customers
WHERE Country='Germany' AND (City='Berlin' OR City='München')

```

Filtering data

- WHERE Clause
- Filtering with Text vs Numeric Fields
- Comparison operators (=, !=, >, <, >=, <=)
- BETWEEN
- IN
- LIKE
- Logic operators (AND, OR, NOT)
- IS NULL, IS NOT NULL

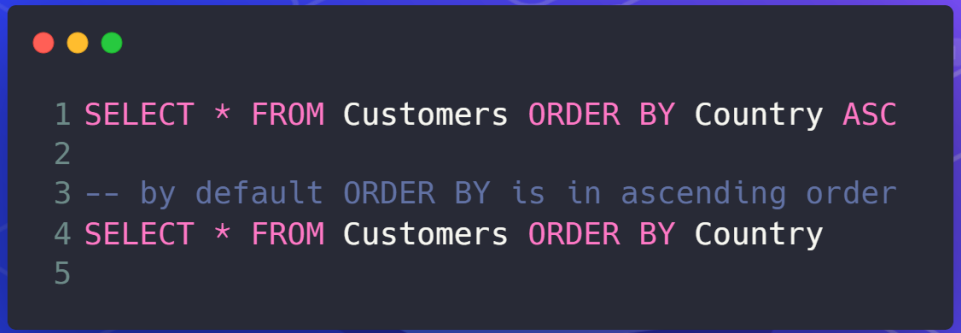
```sql
SELECT CustomerName, ContactName, Address
FROM Customers
WHERE Address IS NULL


SELECT CustomerName, ContactName, Address
FROM Customers
WHERE Address IS NOT NULL
```

Agenda

- What is RDBMS.
- Tables, Columns and Data types
- How to design a database.
- One-To-Many & Many-To-Many Relationships.
- Examples: FB, Youtube, Souq.com
- SQL
- CRUD
- Selecting data
- Filtering data
- **Ordering data**
- Limiting data
- Aggregate Functions
- Joining tables
- Grouping data
- Subqueries
- Inserting new data
- Updating data
- Deleting data
- Python and SQLite
- DB Browser for SQLite

Ordering data

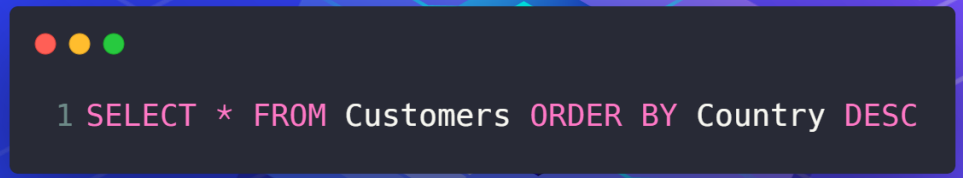
- Order rows by single column ascending
- Order rows by single column descending
- Filter then Order
- Order rows by several columns



```sql
SELECT * FROM Customers ORDER BY Country ASC

-- by default ORDER BY is in ascending order
SELECT * FROM Customers ORDER BY Country

```

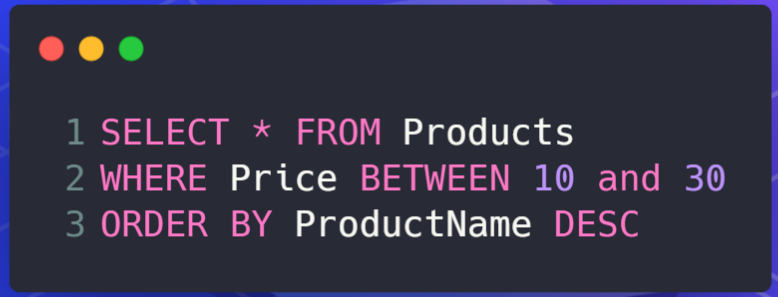
Ordering data

- Order rows by single column ascending
- Order rows by single column descending
- Filter then Order
- Order rows by several columns



Ordering data

- Order rows by single column ascending
- Order rows by single column descending
- Filter then Order
- Order rows by several columns



Ordering data

- Order rows by single column ascending
- Order rows by single column descending
- Filter then Order
- Order rows by several columns

```sql
/*
Order first with Country ASC
and when there are many rows that have same countries
order them by CustomerName ASC
*/
SELECT * FROM Customers
ORDER BY Country, CustomerName


/*
Order first with Country ASC
and when there are many rows that have same countries
order them by CustomerName DESC
*/
SELECT * FROM Customers
ORDER BY Country ASC, CustomerName DESC


/*
Order first with Country DESC
and when there are many rows that have same countries
order them by CustomerName DESC
*/
SELECT * FROM Customers
ORDER BY Country DESC, CustomerName DESC
```

Agenda

- What is RDBMS.
- Tables, Columns and Data types
- How to design a database.
- One-To-Many & Many-To-Many Relationships.
- Examples: FB, Youtube, Souq.com
- SQL
- CRUD
- Selecting data
- Filtering data
- Ordering data
- Limiting data
- Aggregate Functions
- Joining tables
- Grouping data
- Subqueries
- Inserting new data
- Updating data
- Deleting data
- Python and SQLite
- DB Browser for SQLite

Limiting data

- Limit resulted rows to a specific number from first row
- Limit resulted rows to a specific number from a specific offset row

```sql
-- return only the first 3 rows
SELECT * FROM Customers LIMIT 3
```

Limiting data

- Limit resulted rows to a specific number from first row
- Limit resulted rows to a specific number from a specific offset row

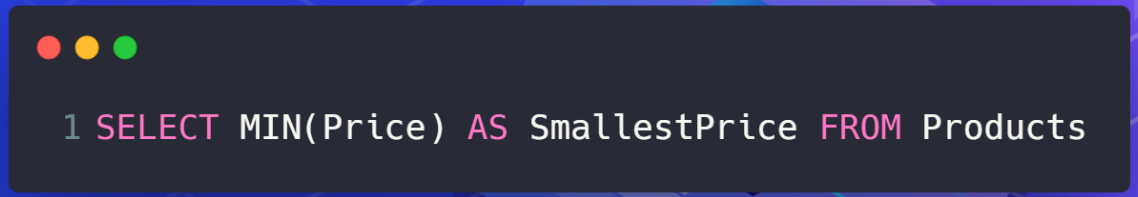
```sql
-- return only 3 rows after the second row
SELECT * FROM Customers LIMIT 2, 3

-- or you can use OFFSET keyword
SELECT * FROM Customers LIMIT 3 OFFSET 2
```

Agenda

- What is RDBMS.
- Tables, Columns and Data types
- How to design a database.
- One-To-Many & Many-To-Many Relationships.
- Examples: FB, Youtube, Souq.com
- SQL
- CRUD
- Selecting data
- Filtering data
- Ordering data
- Limiting data
- **Aggregate Functions**
- Joining tables
- Grouping data
- Subqueries
- Inserting new data
- Updating data
- Deleting data
- Python and SQLite
- DB Browser for SQLite

Aggregate Functions

- MIN
- MAX
- COUNT
- AVG
- SUM



```sql
SELECT MIN(Price) AS SmallestPrice FROM Products
```

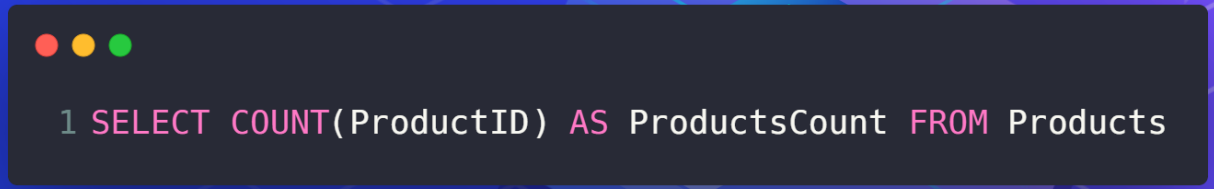
Aggregate Functions

- MIN
- MAX
- COUNT
- AVG
- SUM

```
1 SELECT MAX(Price) AS LargestPrice FROM Products
```

Aggregate Functions

- MIN
- MAX
- COUNT
- AVG
- SUM



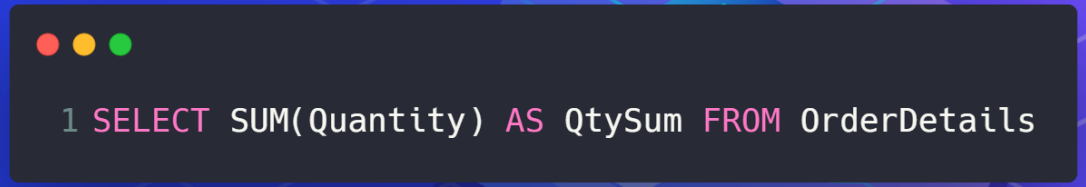
```sql
SELECT COUNT(ProductID) AS ProductsCount FROM Products
```

Aggregate Functions

- MIN
- MAX
- COUNT
- AVG
- SUM

Aggregate Functions

- MIN
- MAX
- COUNT
- AVG
- SUM



```sql
SELECT SUM(Quantity) AS QtySum FROM OrderDetails
```

Agenda

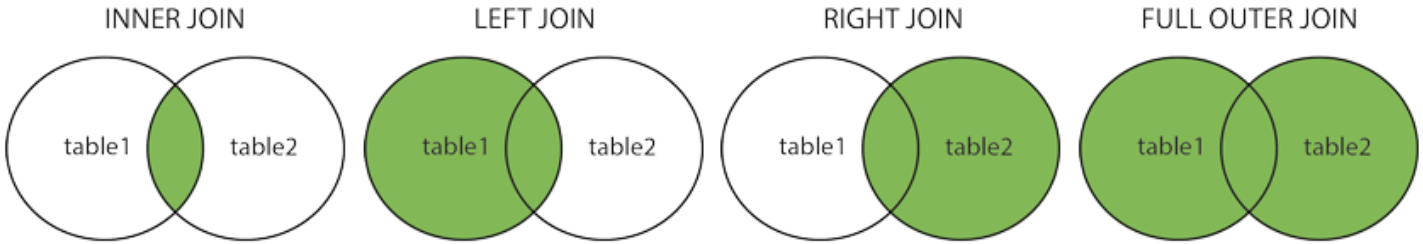
- What is RDBMS.
- Tables, Columns and Data types
- How to design a database.
- One-To-Many & Many-To-Many Relationships.
- Examples: FB, Youtube, Souq.com
- SQL
- CRUD
- Selecting data
- Filtering data
- Ordering data
- Limiting data
- Aggregate Functions
- **Joining tables**
- Grouping data
- Subqueries
- Inserting new data
- Updating data
- Deleting data
- Python and SQLite
- DB Browser for SQLite

Joining tables

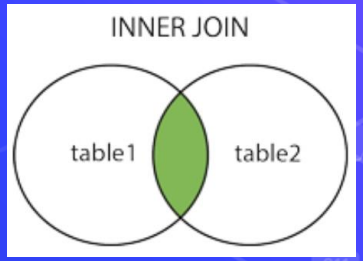
Different Types of SQL JOINs

Here are the different types of the JOINs in SQL:

- **(INNER) JOIN**: Returns records that have matching values in both tables
- **LEFT (OUTER) JOIN**: Return all records from the left table, and the matched records from the right table
- **RIGHT (OUTER) JOIN**: Return all records from the right table, and the matched records from the left table
- **FULL (OUTER) JOIN**: Return all records when there is a match in either left or right table



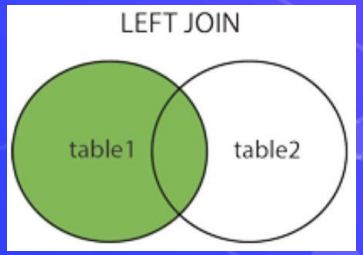
Joining tables (INNER)



The INNER JOIN keyword selects all rows from both tables as long as there is a match between the columns. If there are records in the "Orders" table that do not have matches in "Customers“, these orders will not be shown!

```sql
-- This SQL statement selects all orders with customer information
SELECT Orders.OrderID, Customers.CustomerName
FROM Orders INNER JOIN Customers
ON Orders.CustomerID = Customers.CustomerID
```

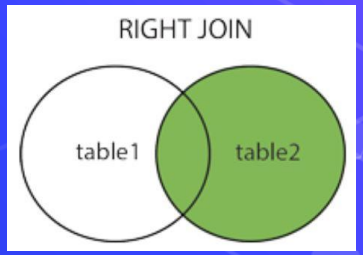
Joining tables (LEFT)



The LEFT JOIN keyword returns all records from the left table (Customers), and the matched records from the right table (Orders). The result is NULL from the right side, if there is no match.

```sql
-- This SQL statement select all customers
-- and any orders they might have
SELECT Customers.CustomerName, Orders.OrderID
FROM Customers LEFT JOIN Orders
ON Customers.CustomerID = Orders.CustomerID
ORDER BY Customers.CustomerName
```

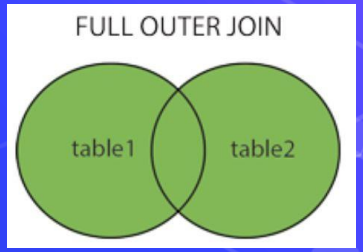
Joining tables (RIGHT)



The RIGHT JOIN keyword returns all records from the right table (Employees), and the matched records from the left table (Orders). The result is NULL from the left side, when there is no match.

```sql
-- This SQL statement will return all employees
-- and any orders they might have placed
SELECT Orders.OrderID, Employees.LastName, Employees.FirstName
FROM Orders RIGHT JOIN Employees
ON Orders.EmployeeID = Employees.EmployeeID
ORDER BY Orders.OrderID
```

Joining tables (FULL)



The FULL OUTER JOIN keyword returns all matching records from both tables whether the other table matches or not. So, if there are rows in "Customers" that do not have matches in "Orders", or if there are rows in "Orders" that do not have matches in "Customers", those rows will be listed as well with NULL

```sql
SELECT Customers.CustomerName, Orders.OrderID
FROM Customers FULL OUTER JOIN Orders
ON Customers.CustomerID=Orders.CustomerID
ORDER BY Customers.CustomerName;
```

Agenda

- What is RDBMS.
- Tables, Columns and Data types
- How to design a database.
- One-To-Many & Many-To-Many Relationships.
- Examples: FB, Youtube, Souq.com
- SQL
- CRUD
- Selecting data
- Filtering data
- Ordering data
- Limiting data
- Aggregate Functions
- Joining tables
- **Grouping data**
- Subqueries
- Inserting new data
- Updating data
- Deleting data
- Python and SQLite
- DB Browser for SQLite

Grouping data

- GROUP BY
- HAVING

```sql
SELECT COUNT(CustomerID) AS CustomersCount, City
FROM Customers WHERE Country = 'France'
GROUP BY City
ORDER BY CustomersCount DESC
```

Grouping data

- GROUP BY
- HAVING

```sql
SELECT COUNT(CustomerID) AS CustomersCount, City
FROM Customers WHERE Country = 'France'
GROUP BY City
HAVING CustomersCount > 4
ORDER BY CustomersCount DESC, City ASC
```

Agenda

- What is RDBMS.
- Tables, Columns and Data types
- How to design a database.
- One-To-Many & Many-To-Many Relationships.
- Examples: FB, Youtube, Souq.com
- SQL
- CRUD
- Selecting data
- Filtering data
- Ordering data
- Limiting data
- Aggregate Functions
- Joining tables
- Grouping data
- **Subqueries**
- Inserting new data
- Updating data
- Deleting data
- Python and SQLite
- DB Browser for SQLite

Subqueries

```sql
SELECT * FROM Products
WHERE SupplierID =
(SELECT SupplierID FROM Suppliers WHERE SupplierName='Exotic Liquid')
```

```sql
SELECT * FROM Orders
WHERE EmployeeID IN
(SELECT EmployeeID FROM Employees WHERE Country='Canada')
```

Agenda

- What is RDBMS.
- Tables, Columns and Data types
- How to design a database.
- One-To-Many & Many-To-Many Relationships.
- Examples: FB, Youtube, Souq.com
- SQL
- CRUD
- Selecting data
- Filtering data
- Ordering data
- Limiting data
- Aggregate Functions
- Joining tables
- Grouping data
- Subqueries
- **Inserting new data**
- Updating data
- Deleting data
- Python and SQLite
- DB Browser for SQLite

Inserting new data

```sql
INSERT INTO Customers
(CustomerName, ContactName, Address, City, PostalCode, Country)
VALUES
('Ahmed', 'Mr. Ahmed', 'Maadi 21', 'Cairo', '4006', 'Egypt')
```

Agenda

- What is RDBMS.
- Tables, Columns and Data types
- How to design a database.
- One-To-Many & Many-To-Many Relationships.
- Examples: FB, Youtube, Souq.com
- SQL
- CRUD
- Selecting data
- Filtering data
- Ordering data
- Limiting data
- Aggregate Functions
- Joining tables
- Grouping data
- Subqueries
- Inserting new data
- **Updating data**
- Deleting data
- Python and SQLite
- DB Browser for SQLite

Updating data

```sql
/*
The following SQL statement updates the first customer
(CustomerID = 1) with a new contact person and a new city.
*/
UPDATE Customers
SET ContactName = 'Alfred Schmidt', City= 'Frankfurt'
WHERE CustomerID = 1


/*
The following SQL statement will update the contactname
to "Juan" for all records where country is "Mexico"
*/
UPDATE Customers
SET ContactName='Juan'
WHERE Country='Mexico'
```

Updating data

```sql
UPDATE Products SET Stock = Stock - 2 WHERE ProductID = 200

UPDATE Products SET Stock = Stock + 2 WHERE ProductID = 200
```

Agenda

- What is RDBMS.
- Tables, Columns and Data types
- How to design a database.
- One-To-Many & Many-To-Many Relationships.
- Examples: FB, Youtube, Souq.com
- SQL
- CRUD
- Selecting data
- Filtering data
- Ordering data
- Limiting data
- Aggregate Functions
- Joining tables
- Grouping data
- Subqueries
- Inserting new data
- Updating data
- **Deleting data**
- Python and SQLite
- DB Browser for SQLite

Deleting data

```sql
/*
The following SQL statement deletes the customer
"Alfreds Futterkiste" from the "Customers" table
*/
DELETE FROM Customers
WHERE CustomerName='Alfreds Futterkiste'


/*
The following SQL statement deletes all rows
in the "Customers" table, without deleting the table
*/
DELETE FROM Customers
```

Check Also

- Create / Drop / Backup Database
- Create / Drop / Alter Table
- Constraints (Not Null, Unique, Primary and Foreign Key)
- Indexes
- Database normalization
- Transactions
- Triggers & Stored Procedures
- Views
- SQL Injection
- NoSQL
- Replication
- Shards

Agenda

- What is RDBMS.
- Tables, Columns and Data types
- How to design a database.
- One-To-Many & Many-To-Many Relationships.
- Examples: FB, Youtube, Souq.com
- SQL
- CRUD
- Selecting data
- Filtering data
- Ordering data
- Limiting data
- Aggregate Functions
- Joining tables
- Grouping data
- Subqueries
- Inserting new data
- Updating data
- Deleting data
- **Python and SQLite**
- DB Browser for SQLite

Python and SQLite

There is a lot of RDBMS out there

- SQLite
- MySQL
- PostgreSQL
- ...

We will use SQLite because it’s simple and easy to work with.

Python and SQLite

- Read
- Create
- Update
- Delete

```python
import sqlite3

connection = sqlite3.connect("weather.db")

sql = "SELECT * FROM users WHERE age >= ? AND name LIKE ?"
data = [20, '%hmed']

rows = connection.execute().fetchall()
connection.close()

print(rows)
"""
[(3, 'eman', 'Giza', 20),
 (12, 'Eslam Ahmed', 'Cairo', 20),
 (15, 'hamada', 'Cairo', 22)
]
"""
```

Python and SQLite

- Read
- Create
- Update
- Delete

```python
import sqlite3

connection = sqlite3.connect("weather.db")

sql = "INSERT INTO users (name,city,age) VALUES (?, ?, ?)"
data = ['Ebrahem', 'Giza', '35']

cursor = connection.execute(sql, data)

connection.commit()
connection.close()

print(cursor.lastrowid)
"""
10
"""
```

Python and SQLite

- Read
- Create
- Update
- Delete

```python
import sqlite3

connection = sqlite3.connect("weather.db")

sql = "UPDATE users SET name=?, age=? WHERE id=?"
data = ['Ezz', 22, 31]

cursor = connection.execute(sql, data)

connection.commit()
connection.close()
```

Python and SQLite

- Read
- Create
- Update
- Delete

```python
import sqlite3

connection = sqlite3.connect("weather.db")

sql = "DELETE FROM users WHERE age >= ?"
data = [31]

cursor = connection.execute(sql, data)

connection.commit()
connection.close()
```

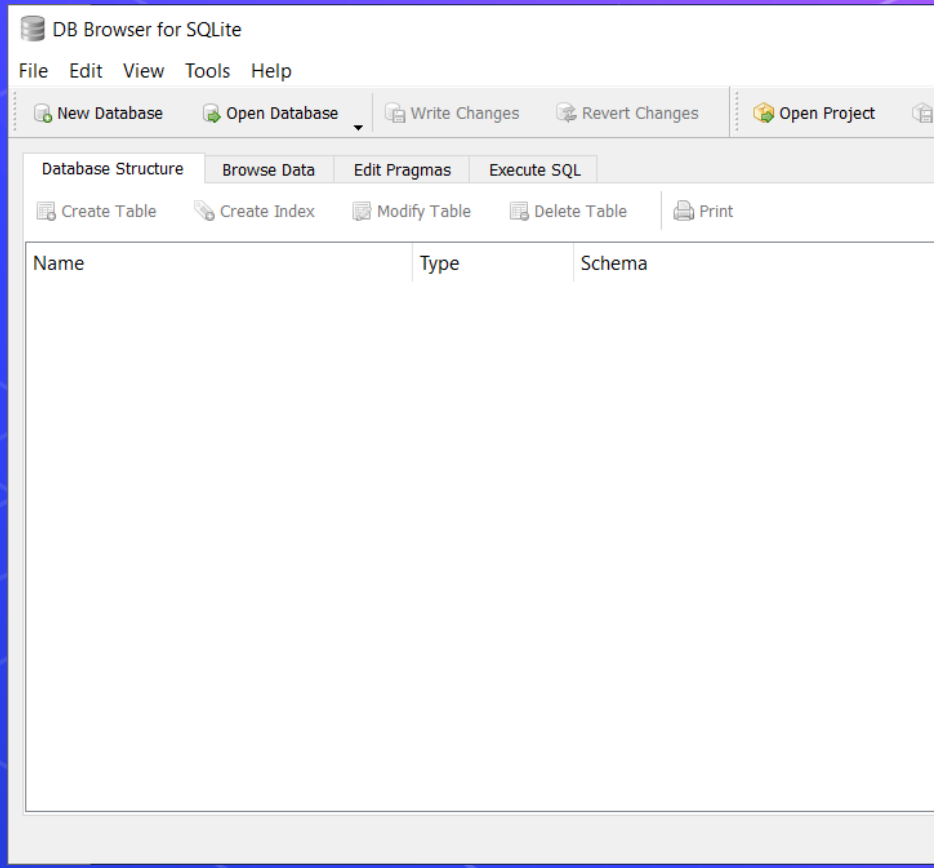
Agenda

- What is RDBMS.
- Tables, Columns and Data types
- How to design a database.
- One-To-Many & Many-To-Many Relationships.
- Examples: FB, Youtube, Souq.com
- SQL
- CRUD
- Selecting data
- Filtering data
- Ordering data
- Limiting data
- Aggregate Functions
- Joining tables
- Grouping data
- Subqueries
- Inserting new data
- Updating data
- Deleting data
- Python and SQLite
- **DB Browser for SQLite**

DB Browser for SQLite

GUI to work with SQLite

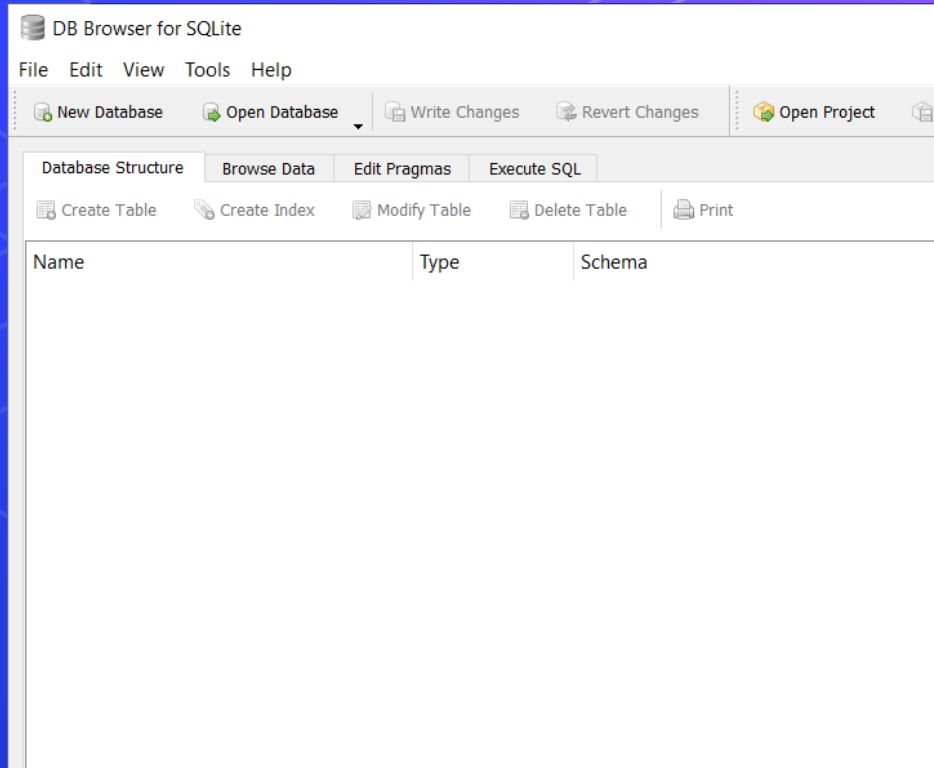
Download
https://sqlitebrowser.org



Tutorial
https://www.youtube.com/playlist?list=PLU70qqWW4frGdwNh4czgTrCrHvPhyt2aI

DB Browser for SQLite

- Open Database
- Create Database and tables
- Create columns
- Create Foreign Keys
- Modify tables
- CRUD with GUI
- Execute SQL
- Import from SQL file
- Export table to CSV



Questions ?!

Thanks!

>_ Live long and prosper