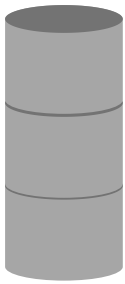


# Crash Course



# SQL

Antonio Scapellato

# Introduction

SQL (Structured Query Language) is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS). It is particularly useful in handling structured data, i.e. data incorporating relations among entities and variables.

# Syntax and commands

- SELECT - extracts data from a database
- UPDATE - updates data in a database
- DELETE - deletes data from a database
- INSERT INTO - inserts new data into a database
- CREATE DATABASE - creates a new database
- ALTER DATABASE - modifies a database
- CREATE TABLE - creates a new table
- ALTER TABLE - modifies a table
- DROP TABLE - deletes a table
- CREATE INDEX - creates an index (search key)
- DROP INDEX - deletes an index

# SELECT

## Syntax

```
SELECT column1, column2, ...  
FROM table_name;
```

## Example

```
SELECT CustomerName, City  
FROM Customers;
```

# SELECT WHERE

## Syntax

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

## Example

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

**Note: \* == Everything**

# SELECT AND, OR, NOT

## Syntax

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition1 AND condition2 AND condition3 ...;
```

## Example

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

**Note: \* == Everything**

# SELECT ORDER BY

## Syntax

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

## Example

```
SELECT * FROM Customers  
ORDER BY Country;
```

# SELECT NULL

## Syntax

```
SELECT column_names  
FROM table_name  
WHERE column_name IS NULL;
```

## Example

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



# UPDATE

## Syntax

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

## Example

```
UPDATE Customers  
SET ContactName = 'Alfred  
Schmidt', City = 'Frankfurt'  
WHERE CustomerID = 1;
```

# DELETE

## Syntax

```
DELETE FROM table_name  
WHERE condition;
```

## Example

```
DELETE FROM Customers  
WHERE CustomerName='Alfreds Futterkiste';
```

# DELETE

## Syntax

```
DELETE FROM table_name  
WHERE condition;
```

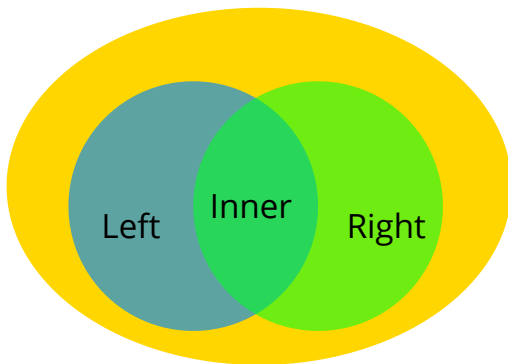
## Example

```
DELETE FROM Customers  
WHERE CustomerName='Alfreds Futterkiste';
```

# JOINS

## Intro

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



FULL

# JOINS INNER

## Syntax

```
SELECT column_name(s)FROM table1  
INNER JOIN table2  
ON table1.column_name = table2.column_name;
```

## Example

```
SELECT Orders.OrderID, Customers.CustomerName  
FROM Orders  
INNER JOIN Customers  
ON Orders.CustomerID = Customers.CustomerID;
```

# JOINS LEFT

## Syntax

```
SELECT column_name(s)  
FROM table1  
LEFT JOIN table2  
ON table1.column_name = table2.column_name;
```

## Example

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

# JOINS RIGHT

## Syntax

```
SELECT column_name(s)FROM table1  
RIGHT JOIN table2  
ON table1.column_name = table2.column_name;
```

## Example

```
SELECT Orders.OrderID, Employees.LastName, Employees.FirstName  
FROM Orders  
RIGHT JOIN Employees  
ON Orders.EmployeeID = Employees.EmployeeIDORDER BY Orders.OrderID;
```

# JOINS FULL

## Syntax

```
SELECT column_name(s)
FROM table1
FULL OUTER JOIN table2
ON table1.column_name = table2.column_name
WHERE condition;
```

## Example

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



# SQL