

## How to write scripts?

- Split big query into small steps & introduce structure using **With result AS (SELECT ...)** construction
- Syntax:
  - SELECT, UPDATE, INSERT, DELETE ... FROM table1 ...;
  - CREATE TABLE t\_name (a\_name a\_type, ...);
  - ALTER TABLE t\_name ADD (c\_name datatype, ...);  
DROP  
MODIFY
  - Arithmetic Operators (+, -, \*, /, %); Logical (AND, OR, ...)
  - Scalar Functions: (LEN(), ROUND(), ...)
  - Aggregate Functions: [AVG(), SUM(), ...]
  - GROUP BY, ORDER BY, HAVING
  - others

## Relational Database

TABLE 1 = Entity Type

ID	Attribute 1	Attribute 2	← Entity Attributes
1	value1-1	value2-1	← Entity 1
2	value1-2	value2-2	← Entity 2
3			← Entity 3

Foreign key

same type of data

same type of data

ID	Att3	Att4
1	val3.1	val4.1
2	val3.2	val4.2

TABLE 2 - Entity Type 2

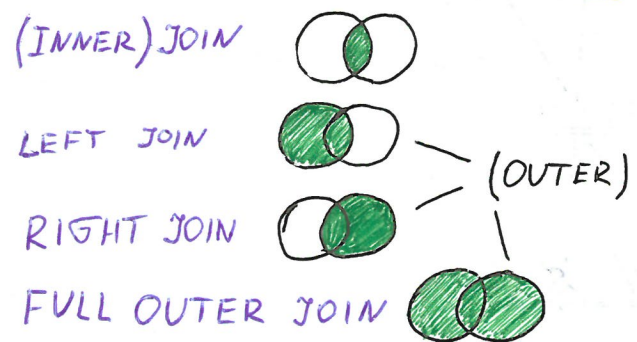
**SQL**  
Structured query language

## Core Definitions & Rules

- Primary Key (PK) - unique id. of entity
  - Foreign Key (FK) - unique id of another (linked) table
  - Normalization - actions to make DB in a normal form (correct & efficient)
  - Anomaly - situation of inconsistency in DB
    - Update
    - Insert
    - Delete
  - Normal Forms:
    - Unnormalized (UNF) - no duplicates, unique ID
    - 1NF - no multiple values in cells  
Single values
    - 2NF - full dependency on PK (no partial dependency)
    - 3NF - no transitive dependencies a1 → a2 → a3
    - 4NF
    - 5NF
    - 6NF and others

in between are too difficult to explain :)
- The idea to decompose to the most trivial dependencies.

## Tables Connections (JOIN)



Additionally:

select ...  
from TableA  
LEFT JOIN TableB  
ON A.key = B.key  
WHERE B.key is NULL

select ...  
FROM TableA  
FULL OUTER JOIN TableB  
ON A.key = B.key  
WHERE A.key is NULL OR  
B.key is NULL