Databases

SQL

Relational databases

- A relational database is a collection of data items with pre-defined relationships between them. These items are organized as a set of tables with columns and rows. Tables are used to hold information about the objects to be represented in the database. Each column in a table holds a certain kind of data and a field stores the actual value of an attribute. The rows in the table represent a collection of related values of one object or entity. Each row in a table could be marked with a unique identifier called a primary key, and rows among multiple tables can be made related using foreign keys. This data can be accessed in many different ways without reorganizing the database tables themselves.

Relational database example - internet shop

- products
- orders
- customers

DBMS (database management system)

- MySQL
- PostgreSQL
- Oracle
- SQL Server
- More: https://en.wikipedia.org/wiki/List_of_relational_database_management_systems

SQL - Structured Query Language

- SQL is a standard language for storing, manipulating and retrieving data in databases.

What Can SQL do?

- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert records in a database
- SQL can update records in a database
- SQL can delete records from a database
- SQL can create new databases
- SQL can create new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database
- SQL can set permissions on tables, procedures, and views

Some of The Most Important SQL 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

Tables

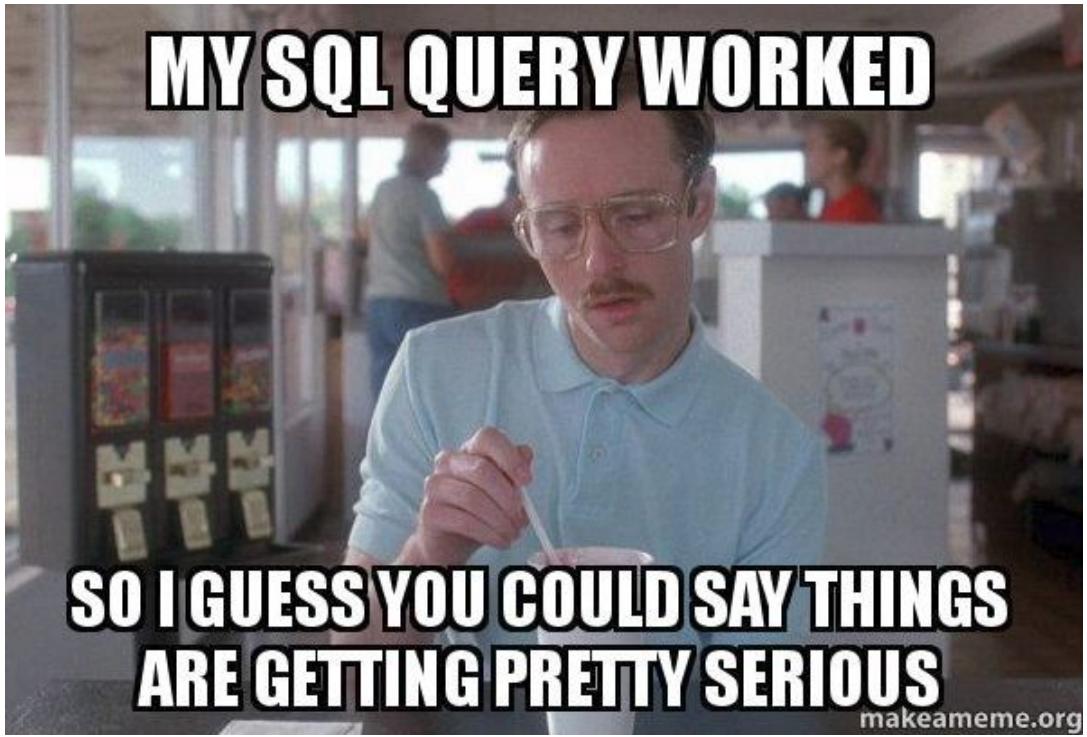
- Creating table: https://www.w3schools.com/sql/sql_create_table.asp
- Dropping table: https://www.w3schools.com/sql/sql_drop_table.asp
- Changing table: https://www.w3schools.com/sql/sql_alter.asp
- MySql data types (for columns):
  - https://dev.mysql.com/doc/refman/8.0/en/data-types.html
  - https://www.tutorialspoint.com/mysql/mysql-data-types.htm

WHERE clause and operators

- https://www.w3schools.com/sql/sql_where.asp
- https://www.w3schools.com/sql/sql_operators.asp

Aggregate functions

- https://www.w3schools.com/sql/sql_count_avg_sum.asp


MY SQL QUERY WORKED
SO I GUESS YOU COULD SAY THINGS ARE GETTING PRETTY SERIOUS
makeameme.org

Constraints - Primary key

```
CREATE TABLE Persons (
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int,
    PRIMARY KEY (ID)
);
```

- https://www.w3schools.com/sql/sql_primarykey.ASP
- https://www.mysqltutorial.org/mysql-primary-key/
- Not the same as Unique constraint (https://www.w3schools.com/sql/sql_unique.asp)
- PK auto increment: https://www.w3schools.com/sql/sql_autoincrement.asp

Constraints - Foreign key

```
CREATE TABLE Orders (
    OrderID int NOT NULL,
    OrderNumber int NOT NULL,
    PersonID int,
    PRIMARY KEY (OrderID),
    CONSTRAINT FK_PersonOrder FOREIGN KEY (PersonID)
    REFERENCES Persons(PersonID)
);

ALTER TABLE Orders
ADD FOREIGN KEY (PersonID) REFERENCES Persons(PersonID);
```

- https://www.w3schools.com/sql/sql_foreignkey.asp

Table relationships

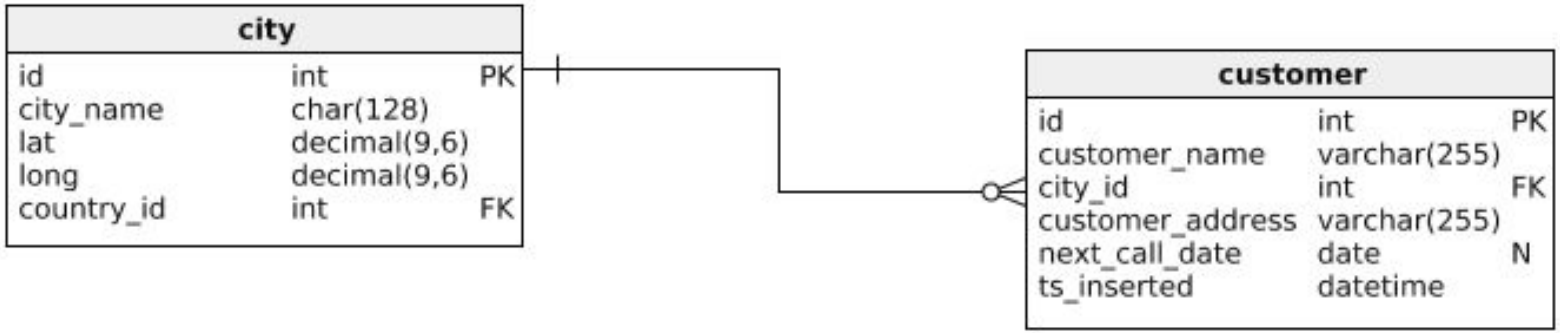
- In a relational database, a relationship is formed by correlating rows belonging to different tables. A table relationship is established when a child table defines a Foreign Key column that references the Primary Key column of its parent table.

Relationships

- Every database table relationship is, therefore, built on top of Foreign Key columns, and there can be three table relationship types:
  - **one-to-many** is the most common relationship, and it associates a row from a parent table to multiple rows in a child table.
  - **many-to-many** requires a link table containing two Foreign Key columns that reference the two different parent tables.
  - **one-to-one** requires the child table Primary Key to be associated via a Foreign Key with the parent table Primary Key column.

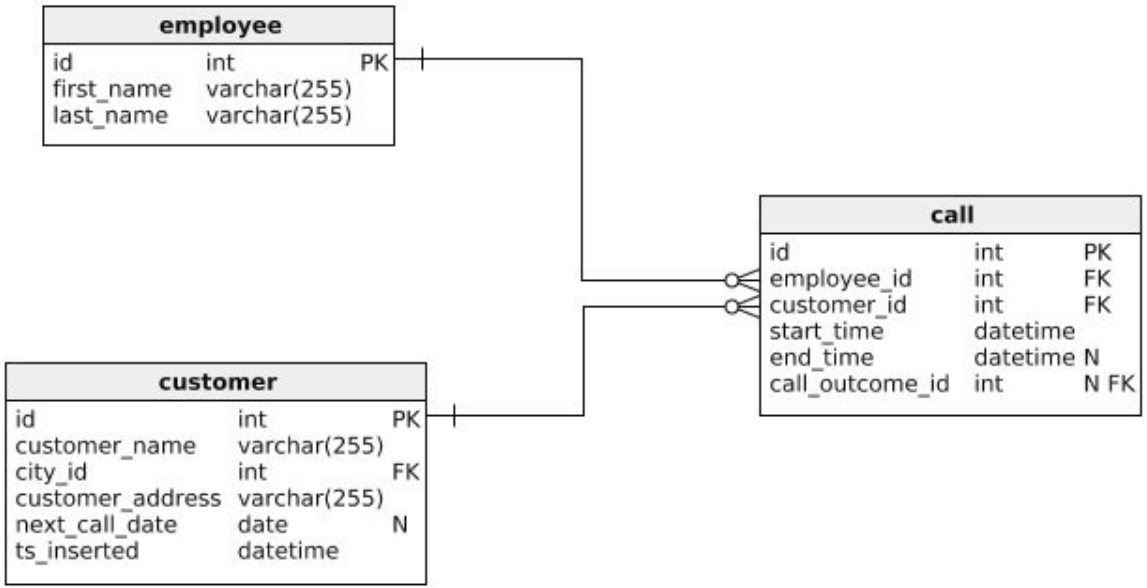
One to many

- **one-to-many** is the most common relationship, and it associates a row from a parent table to multiple rows in a child table.



Many to many (model)

- **many-to-many** requires a link table containing two Foreign Key columns that reference the two different parent tables.



Many to many - SQL (creating tables)

```sql
create table employees (
    id int auto_increment,
    first_name varchar(255) not null,
    PRIMARY KEY (id)
);

create table customers (
   id int not null auto_increment,
   email varchar(255) not null,
   PRIMARY KEY (id)
);
```

```sql
create table employe_customer_calls (
   id int not null auto_increment,
   employee_id int not null,
   customer_id int not null,
   call_time datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
   PRIMARY KEY (id),
   FOREIGN KEY (employee_id) REFERENCES employees(id),
   FOREIGN KEY (customer_id) REFERENCES customers(id)
);
```

Many to many - SQL (inserting data)

```sql
-- insert customers
insert into customers values
  (null, 'peters@mail.com'),
  (null, 'sanita@mail.com');

-- insert employees
 insert into employees values
  (1, 'Janis employee'),
  (2, 'Juris employee');

-- add calls
insert into employe_customer_calls (customer_id, employee_id) values
   -- Peteris called Janis
   (1, 1),
   -- Peteris called Juris
   (1, 2);
```

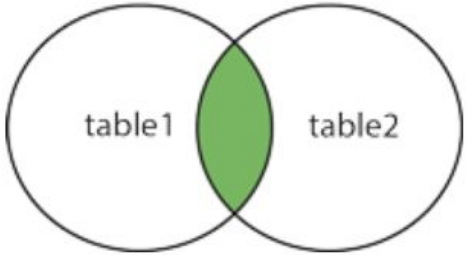
JOINS

- A JOIN clause is used to combine rows from two or more tables, based on a related column between them.
  - INNER JOIN
  - LEFT JOIN
  - RIGHT JOIN
  - FULL OUTER JOIN
  - CROSS JOIN

INNER JOIN

- The INNER JOIN keyword selects records that have matching values in both tables.

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



example: https://www.w3schools.com/sql/sql_join_inner.asp

JOINS

- LEFT JOIN - https://www.w3schools.com/sql/sql_join_left.asp
- RIGHT JOIN - https://www.w3schools.com/sql/sql_join_right.asp
- FULL OUTER JOIN - https://www.w3schools.com/sql/sql_join_full.asp

Other useful SQL language

- Aliases: https://www.w3schools.com/sql/sql_alias.asp
- ORDER BY: https://www.w3schools.com/sql/sql_orderby.asp
- HAVING: https://www.w3schools.com/sql/sql_having.asp
- LIMIT: https://www.mysqltutorial.org/mysql-limit.aspx

Optional topics which you should be aware of

- **Transactions** - https://www.mysqltutorial.org/mysql-transaction.aspx/
- **ACID:**
  - https://database.guide/what-is-acid-in-databases/
  - https://stackoverflow.com/questions/999394/whats-a-real-world-example-of-acid/34347332
- Normal forms:
  - https://www.geeksforgeeks.org/normal-forms-in-dbms/
  - https://www.guru99.com/database-normalization.html
- DB storage engines http://zetcode.com/mysql/storageengines/
  - (We use InnoDB, because it's the default in MySQL)
- Procedures, triggers and views