SQL

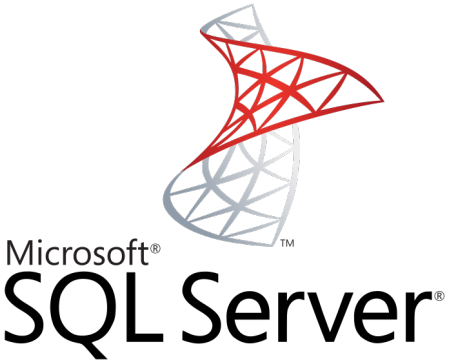
Egor Kulbachka

Relational database

- Digital database whose organization is based on the relational model of data

RDBMS

- Relational Database Management System



Installing MySQL

- http://dev.mysql.com/downloads/mysql/
- http://dev.mysql.com/downloads/workbench/
- http://sqlfiddle.com

Creating a database

- CREATE DATABASE learn_sql;
- USE learn_sql;

What is relation?

Attribute

Name	Age	Position	Level
Sam Smith	22	QA	Junior
John Black	23	Dev	Senior

Tuple

Integer types

- TINYINT – 1 Byte
- SMALLINT – 2 Bytes
- MEDIUMINT – 3 Bytes
- INTEGER – 4 Bytes
- BIGINT – 8 Bytes

Real number types

- DECIMAL(M, N) – 1234,42
- FLOAT – 4 Bytes
- DOUBLE – 8 Bytes

Date/time types

- DATE
- DATETIME
- TIMESTAMP

String types

- CHAR(N)
- VARCHAR(N)
- TINYTEXT
- TEXT
- MEDIUMTEXT
- LONGTEXT

DDL

Data definition language

- Creating first table:

```
CREATE TABLE person (
    name VARCHAR(255) NOT NULL,
    age INTEGER,
    skills TEXT
) Engine=InnoDB;
```

DML

Data Manipulation Language

- Inserting a record

```
INSERT INTO person (name, age, skills)
    VALUES
    (‘Egor Kulbachka’, 27, ‘Java,SQL,MS Word’);
```

DML

- Select data from the table

```
SELECT name, skills FROM person;
```

Primary Key

```
CREATE TABLE person (
    name VARCHAR(255) NOT NULL,
    age INTEGER,
    skills TEXT,
    PRIMARY KEY(name)
) Engine=InnoDB;
```

Foreign key

```
CREATE TABLE experience (
    employer VARCHAR(255) NOT NULL,
    employee VARCHAR(255) NOT NULL,
    project VARCHAR(255),
    foundation_date DATE,
    PRIMARY KEY(employer, employee),
    FOREIGN KEY(employee) REFERENCES
    person(name)
) Engine=InnoDB;
```

Foreign key

```
CREATE TABLE experience (
    employer VARCHAR(255) NOT NULL,
    employee VARCHAR(255) NOT NULL,
    project VARCHAR(255),
    foundation_date DATE,
    PRIMARY KEY(employer, employee),
    FOREIGN KEY(employee) REFERENCES
    person(name)
) Engine=InnoDB;
```

Filtering queries

- SELECT * FROM person WHERE age >= 30;
- SELECT * FROM person WHERE age < 30 OR name = ‘Robert Martin’;
- SELECT * FROM person WHERE skills like ‘%JavaScript%’

Filters

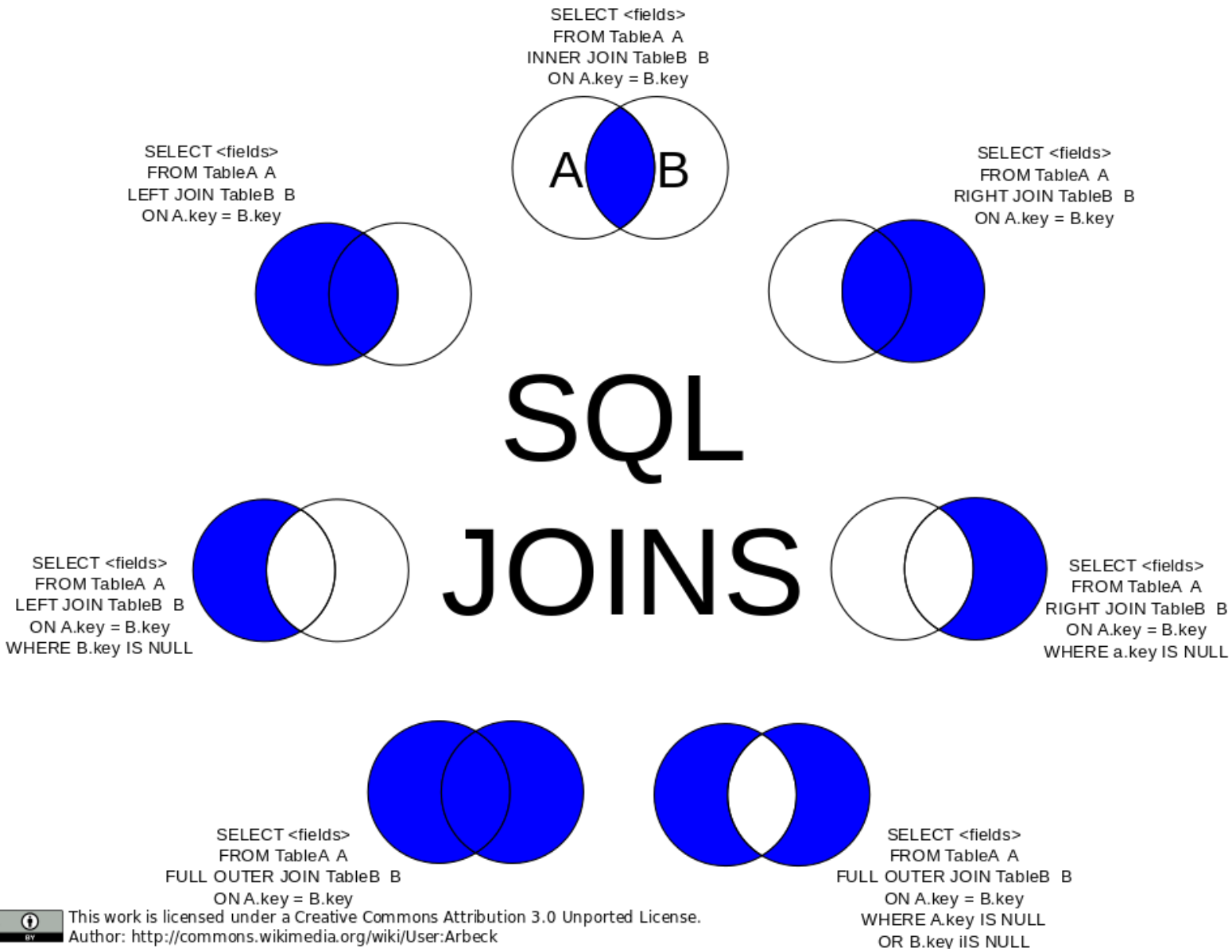
- Equals → =
- Not equals → <>
- Greater than → >
- Less than → <
- Greater/Less than or equals → >= / <=
- Null check → IS NULL / IS NOT NULL

Sorting

- SELECT * FROM person ORDER BY age;

- SELECT * FROM experience ORDER BY foundation_date DESC, project;

Joining



1st Normal Form

- A relation is in first normal form if and only if the domain of each attribute contains only atomic (indivisible) values, and the value of each attribute contains only a single value from that domain

2nd Normal Form

- a table is in 2NF if it is in 1NF and every non-prime attribute of the table is dependent on the whole of every candidate key

3rd Normal Form

3NF if and only if both of the following conditions hold:

- The relation R is in second normal form (2NF)
- Every non-prime attribute of R is non-transitively dependent on every key of R.

Aggregation

- COUNT()
- AVG()
- MIN()
- MAX()

Group result

- SELECT country, COUNT(*) FROM location GROUP BY country;

Having

- SELECT country, COUNT(*) FROM location GROUP BY country HAVING COUNT(*)>1;

Distinct

- SELECT distinct country FROM location;

Indexes

- CREATE INDEX age_index ON person(age);

Other sql operations

DML:

- DELETE FROM table WHERE <condition>;
- UPDATE table SET column=value WHERE <condition>;

DDL:

- DROP TABLE table;
- ALTER TABLE ADD column <type>;
- ALTER TABLE DROP column;
- ALTER TABLE MODIFY column <type>;

Thank you!