Introduction to Databases MySQL

jan.schulz@devugees.org

Agenda

1. Definition
2. SQL Commands
3. Primary Keys
4. Foreign Keys
5. Table Joins

1. Definition

- **SQL** is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in **relational database.**

- SQL is the standard language for Relation Database System. All relational database management systems like “MySQL, MS Access, Oracle, Sybase, Informix, postgres and SQL Server” use SQL as standard database language.

1. Definition

- What do you think is a „**database**“?
- What do you think means „**relational**“?
- What do you think is a „**relational database**“?

1. Definition

- Database: At its core, a place on your hard-drive or in your memory where data is stored.
    - JSON flatfiles, CSV-files, Textfiles, Binary Files (Images...)

  Usually, a system is wrapped around and gives users access to this data, which is stored in database specific format.

  - MySQL
  - MS-SQL
  - Oracle
  - Postgres
  - ...

1. Definition

- Database: At its core, a place on your hard-drive or in your memory where data is stored.
    - JSON flatfiles, CSV-files, Textfiles, Binary Files (Images...)

  Usually, a system is wrapped around and gives users access to this data, which is stored in database specific format.

  - **MySQL ← we learn this**
  - MS-SQL
  - Oracle
  - Postgres
  - ...

1. Definition

Firstname	Lastname	Age
Max	Müller	30
Sandra	Meier	25

Column Names: Firstname, Lastname, Age
Column Values Row 1: Max, Müller, 30
Column Values Row 2: Sandra, Meier, 25

How would you represent that in JSON ?

1. Definition

- Relational: Concerning the way two or more things (more concrete: tables) are connected with each other

1. Definition

- Relational Database: A database, where one data-entity can be connected to another data-entity
- Example:
  - A customer has one address.
  - What, if he has two addresses?

Max has **one** address

Firstname	Lastname	Street	Postal	City
Max	Müller	Lychener Straße 1	10403	Berlin

Max has two addresses

Firstname	Lastname	Street	Postal	City
Max	Müller	Lychener Straße 1	10403	Berlin
Max	Müller	Karl-Marx-Allee 18	10243	Berlin

Problem?

Max has **two** addresses

Firstname	Lastname	Street	Postal	City
Max	Müller	Lychener Straße 1	10403	Berlin
Max	Müller	Karl-Marx-Allee 18	10243	Berlin

Problem = Redundancy. Max Müller exists twice. Having millions of customers uses too much disk-space.

Max has two addresses

Table: Customers

Id	Firstname	Lastname
1	Max	Müller

Table: Addresses

CustomerId	Street	Postal	City
1	Lychener Straße 1	10403	Berlin
1	Karl-Marx-Allee 18	10243	Berlin

Task

Take a blank piece of paper and think of your own web-project that you would like to implement.

1. What data would you store on your server?
2. How would you name your tables?

If you dont come up with an own idea of a web project, think of an online-shop.

2. SQL

- SQL stands for Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in relational databases.
- It is grouped into four language areas:
  - Data Definition Language (DDL)
  - Data Manipulation Language (DML)
  - Data Query Language (DQL)

2. SQL

- Data Definition Language (DDL)

Command	Description
CREATE	Creates a new table
ALTER	Modifies the structure of a table
DROP	Deletes a table

2. SQL

- Data Manipulation Language (DML)

Command	Description
INSERT	Creates a new row
UPDATE	Updates a row
DELETE	Deletes a row

2. SQL

- Data Query Language (DML)

Command	Description
SELECT	Selects a row

2. SQL

Conditional **Select Queries**

```
SELECT
        *
FROM
   TABLENAME
WHERE
   ATTRIBUTENAME =  ATTRIBUTEVALUE
```

2. SQL

Conditional **Select Queries** with multiple conditions

```
SELECT
        *
FROM
   TABLENAME
WHERE
            ATTRIBUTENAME1 =  ATTRIBUTEVALUE1
   AND  ATTRIBUTENAME2 = ATTRIBUTEVALUE1
```

2. SQL

Conditional **Select Queries** with multiple conditions

```
SELECT
        *
FROM
   TABLENAME
WHERE
            ATTRIBUTENAME1 =  ATTRIBUTEVALUE1
   OR     ATTRIBUTENAME2 = ATTRIBUTEVALUE1
```

2. SQL

Conditional **Delete Queries** with multiple conditions

```
DELETE
FROM
   TABLENAME
WHERE
            ATTRIBUTENAME1 =  ATTRIBUTEVALUE1
   OR     ATTRIBUTENAME2 = ATTRIBUTEVALUE1
```

2. SQL

Conditional **Delete Queries** with multiple conditions

```
DELETE
FROM
   TABLENAME
WHERE
            ATTRIBUTENAME1 =  ATTRIBUTEVALUE1
   OR     ATTRIBUTENAME2 = ATTRIBUTEVALUE1
```

2. SQL

Conditional **Update Queries** with multiple conditions

```
UPDATE
   TABLENAME
SET ATTRIBUTENAME1 =  ATTRIBUTEVALUE1
WHERE  ATTRIBUTENAME2 = ATTRIBUTEVALUE2
```

3. Primary Keys

- A primary keys identifies a table. Common examples for a key is an ID
- Examples:
  - Personalausweis ID
  - Bank Account Number
  - Credit Card Number
  - …

3. Primary Keys

Id	Firstname	Lastname	Age
1	Max	Mueller	32
2	Sandra	Meier	25
3	Robert	Schulz	20
4	Bob	Smith	35
5	Max	Mueller	32

In MySQL, we can create Ids that increment automatically.

4. Foreign Keys

- A foreign key identifies a table whose existence depends on another table
- Examples:
  - A user with one or multiple addresses
    - Address does not exist without a customer
  - A voter with one or multiple votes
    - Votes do not exist without a voter
  - A customer with one or multiple orders
    - An order does not exist without a customer

4. Foreign Keys

Id	Firstname	Lastname	Age
1	Max	Mueller	32
2	Sandra	Meier	25
3	Robert	Schulz	20
4	Bob	Smith	35
5	Max	Mueller	32

UserId	OrderValue	NumberItems
1	150€	3
1	200€	1
2	50€	1
3	300€	2
5	100€	2

5. Table Joins

- Table Joins connect **two or more tables** with each other and results in **one table**

5. Table Joins

Id	Firstname	Lastname	Age
1	Max	Mueller	32
2	Sandra	Meier	25
3	Robert	Schulz	20
4	Bob	Smith	35
5	Max	Mueller	32

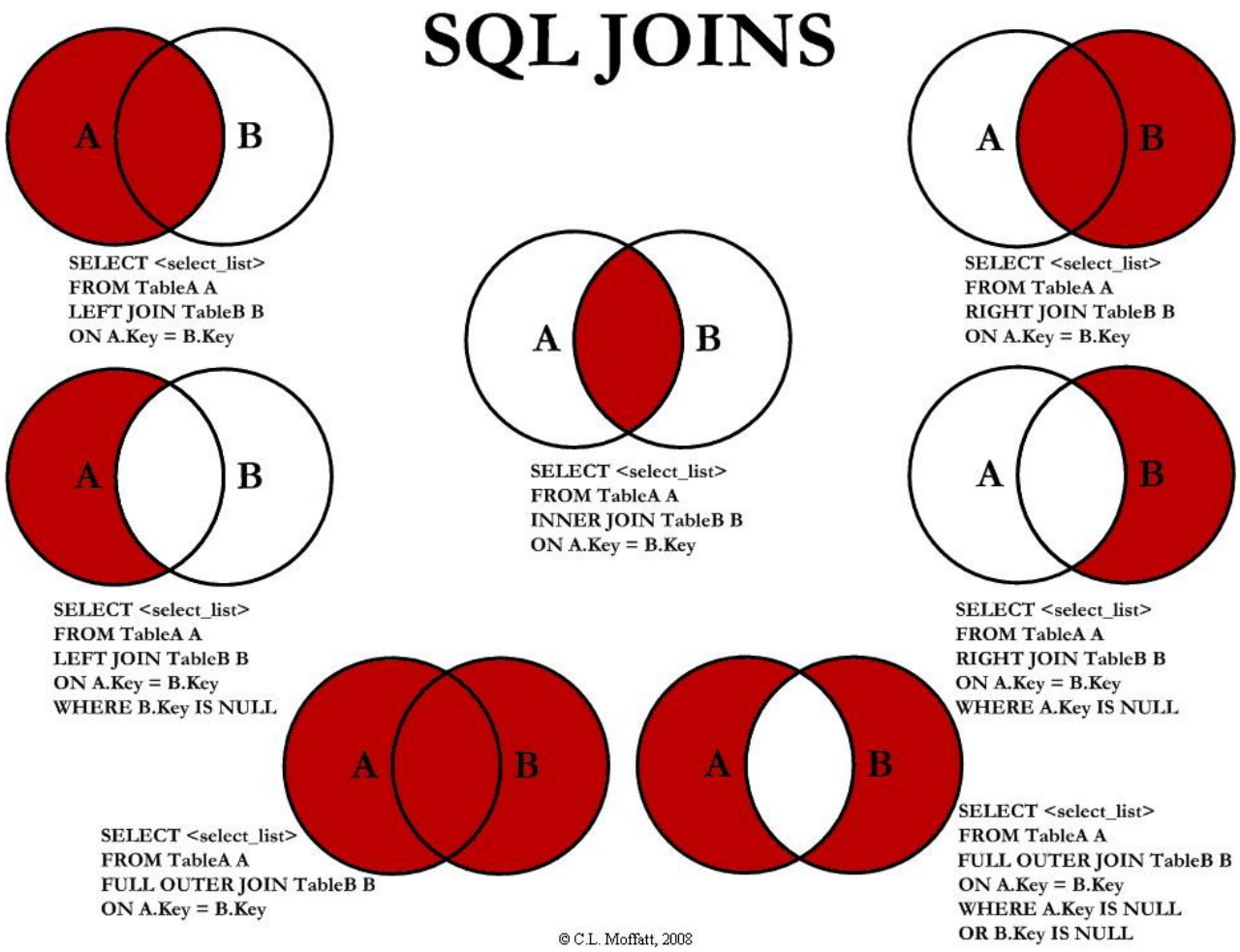
+

UserId	OrderValue	NumberItems
1	150€	3
1	200€	1
2	50€	1
3	300€	2
5	100€	2

=

Id	Firstname	Lastname	Age	OrderValue	NumberItems
1	Max	Mueller	32	150€	3
1	Max	Mueller	32	200€	1
2	Sandra	Meier	25	50€	1
3	Robert	Schulz	20	300€	2
4	Bob	Smith	35	NULL	NULL
5	Max	Mueller	32	100€	2

5. Table Joins



5. Table Joins

```
SELECT
   a.*, b.*
FROM
   Table1 a
JOIN
   Table2 b
ON
   a.id = b.Table1Id
```

Task

Create a MySQL version of your JSON-flatfiles database.

1) Create a database „productsserver“.
2) Create two tables „products“ and „product_types“. Think of what attributes these tables should have.
3) Introduce three product_types: „Movie“, „Book“ and „Music“
4) Fill the products-table with example data.

Task

Change the backend code of your Products Server to use MySQL database instead of JSON flatfiles.

5) Rename server.js to server_json.js. Make a copy of it and name it server_sql.js
6) Change the GET, POST and DELETE methods to use your MySQL database.
7) Test your Implementation!