SQL

Structured Query Language

VIVIFY
ACADEMY

What is SQL? SQL vs MySQL

- SQL is a standard language for accessing and manipulating databases
- MySQL is a database management system, a program that runs on a computer
- MySQL uses the SQL language to manipulate the database itself

Required installations for this week

- MySQL
- PHP extension “php-mysql”
- SQL Client - we will use https://sqlectron.github.io/
- **These things are already installed on your machines**

Connecting to MySQL

Connect to MySQL via SQLECTRON

- Open SQLECTRON
- Add new SQL connection
  - Click on “**Add**” button
  - **Name** should be provided (localhost for example)
  - **Database type** should be Mysql
  - **Server Address** should be 127.0.0.1
  - **Port** is 3306
  - **User** is “root”, **password** is “vivify” - these are credentials for your **mysql connection**
  - Click on “**Save**”
- Open added connection by clicking on “**Connect**”
- Now you are connected to your local MySQL!

SQL statements

SQL statements (queries) rules

- Example of SQL statement: `CREATE DATABASE databasename;`
- SQL keywords are **NOT case sensitive**: create is the same as CREATE
- Every statement ends with a semicolon - ;
- In SQLECTRON you should write statements in textarea

Databases

Databases

- A **database** is a collection of information that is organized so that it can be easily accessed, managed and updated

The CREATE DATABASE Statement

- Usage: `CREATE DATABASE [databasename];`
- Run this example: `CREATE DATABASE vivify_blog;`
- Database and table name convention - **lowercase** and **snake_case**

The SHOW DATABASE Statement

- Show all databases with:

```
SHOW DATABASES;
```

The DROP DATABASE Statement

- Usage: `DROP DATABASE [databasename];`
- Be careful before dropping a database. Deleting a database will result in loss of complete information stored in the database!
- Run example: `DROP DATABASE vivify_blog;`
- Run `SHOW DATABASES;` to see if it is deleted
- Create again the same database

The USE Statement

- This statement is used for selecting database:

```
USE vivify_blog;
```

- After you select a database, it remains the default until you end the session or choose another database with the USE command
- After you select a database, every subsequent statements will be executed for that database (creating tables, inserting records...)

Tables

Tables

- The data in database is stored in objects called **tables**

- A **table** is a collection of related data entries and is identified by a **name**

Blog posts table

id	rating	category	title	teaser	content	created_at	updated_at
1	8	food	Optimized tertiar...	Possimus dolore ...	NEVER come to ...	2015-01-08 10:1..	2017-07-13 05:50:43
2	8	health	Innovative syste...	Molestiae sit fug...	However, I've go...	2009-11-17 04:2..	2017-02-06 09:50:32
3	3	fashion	Versatile intangib..	Dolores est dolor...	And she went ne...	2014-03-06 22:3..	2017-02-21 13:55:35
4	9	music	Fully-configurabl...	Officiis soluta pr...	BEST butter, you...	2010-10-14 10:4..	2017-03-22 13:11:47
5	4	music	Down-sized syst...	Necessitatibus fa...	YET,' she said to ...	2010-04-18 20:2..	2017-04-13 15:50:46
6	4	music	Digitized neutral ...	Dolor voluptatib...	By the time at th...	2012-11-07 09:0..	2017-07-22 05:00:37
7	8	sports	Right-sized 6thge..	At ratione porro ...	Dinah my dear! I ...	2015-11-20 06:2..	2016-08-11 16:52:27
8	7	fashion	Virtual responsiv...	Omnis maiores q...	Queen, stamping...	2012-11-16 19:1..	2016-12-02 16:07:47
9	4	music	User-friendly eco...	Quas et iure sed ...	The jury all brigh...	2008-10-01 19:1..	2017-03-08 11:47:54

Fields (columns) and records (rows)

- Every **table** consists of **fields** (columns) and **records** (rows)

- A field is a column in a table that holds specific information about every record in the table (e.g. ‘title’, ‘rating’, ‘category’)

- Record, also called a row, is each individual entry that exists in a table

Fields (columns)

Records (rows)

id	rating	category	title	teaser	content	created_at	updated_at
1	8	food	Optimized tertiar...	Possimus dolore ...	NEVER come to ...	2015-01-08 10:1..	2017-07-13 05:50:43
2	8	health	Innovative syste...	Molestiae sit fug...	However, I've go...	2009-11-17 04:2..	2017-02-06 09:50:32
3	3	fashion	Versatile intangib..	Dolores est dolor...	And she went ne...	2014-03-06 22:3..	2017-02-21 13:55:35
4	9	music	Fully-configurabl...	Officiis soluta pr...	BEST butter, you...	2010-10-14 10:4..	2017-03-22 13:11:47
5	4	music	Down-sized syst...	Necessitatibus fa...	YET,' she said to ...	2010-04-18 20:2..	2017-04-13 15:50:46
6	4	music	Digitized neutral ...	Dolor voluptatib...	By the time at th...	2012-11-07 09:0..	2017-07-22 05:00:37
7	8	sports	Right-sized 6thge..	At ratione porro ...	Dinah my dear! I ...	2015-11-20 06:2..	2016-08-11 16:52:27
8	7	fashion	Virtual responsiv...	Omnis maiores q...	Queen, stamping...	2012-11-16 19:1..	2016-12-02 16:07:47
9	4	music	User-friendly eco...	Quas et iure sed ...	The jury all brigh...	2008-10-01 19:1..	2017-03-08 11:47:54

The CREATE TABLE Statement

```
CREATE TABLE table_name ( column1 datatype, column2 datatype, .... );
```

- The column parameters specify the names of the columns of the table.

- The datatype parameter specifies the type of data the column can hold (e.g. varchar, integer, date, etc.).

SQL datatypes

data type	description
CHARACTER(n)	Character string. Fixed-length n
VARCHAR(n)	Character string. Variable length. Maximum length n
BOOLEAN	Stores TRUE or FALSE values
INTEGER	Integer numerical (no decimal)
DECIMAL(l,s)	Exact numerical, length (l), scale (s); e.g. (5,2)
DATE	Stores year, month, and day values
TIME	Stores hour, minute, and second values
TIMESTAMP	Stores year, month, day, hour, minute, and second values

The CREATE TABLE Statement (example)

```
CREATE TABLE posts (
    id int,
    title varchar(100),
    content text,
    created_at datetime,
    updated_at datetime
);
```

The DESCRIBE TABLE Statement

```
DESCRIBE [table_name];

DESCRIBE posts;
```

Rows 6

Field	Type	Null	Key	Default	Extra
id	int(10) unsigned	NO	PRI	NULL	auto_increment
title	varchar(100)	NO		NULL	
content	text	NO		NULL	
created_at	timestamp	YES		NULL	
updated_at	timestamp	YES		NULL	
teaser	varchar(300)	YES		NULL	

The DROP TABLE and TRUNCATE TABLE statements

```
DROP TABLE [table_name];
```

- Be careful before dropping a table. Deleting a table will result in loss of complete information stored in the table!
- Drop the **'posts'** table and then run the **describe** statement
- Create the '**posts**' table again

```
TRUNCATE TABLE [table_name];
```

- The TRUNCATE TABLE statement is used to delete the data inside a table, but not the table itself.

The ALTER TABLE statement

```
ALTER TABLE [table_name] ADD [column_name] [datatype];
```

```
ALTER TABLE [table_name] DROP COLUMN [column_name];
```

```
ALTER TABLE [table_name] MODIFY COLUMN [column_name] [datatype];
```

The ALTER TABLE statement (example)

```
ALTER TABLE posts ADD teaser varchar(300);

ALTER TABLE posts MODIFY COLUMN teaser varchar(200);

ALTER TABLE posts DROP COLUMN teaser;

* test after each statement with: DESCRIBE table_name
```