# Лабораторная работа 6

### Задача:

1. Скачать сервер MySQL - <https://downloads.mysql.com/archives/get/p/25/file/mysql-installer-community-5.7.29.0.msi>
   1. Подключаться к БД и взаимодействовать с нею можно через MySQL Workbench - https://www.mysql.com/products/workbench/
2. Создайте базу данных blog
3. Завести в базе данных табличку post, содержащую информацию о посте со следующими колонками:
   1. title VARCHAR(255)
   2. subtitle VARCHAR(255)
   3. author VARCHAR(255)
   4. author\_url VARCHAR(255)
   5. publish\_date VARCHAR(255)
   6. image\_url VARCHAR(255)
   7. featured TINYINT(1) DEFAULT 0
4. SQL-запрос для создания таблички сложить в data/sql/table
5. Заготовить данные о постах в виде sql-файлов с конструкцией INSERT INTO, где прописать вставку информации о каждом из постов(см. блок примеры), SQL-файл с информацией о статьях сложить в data/sql/posts
6. Реализовать чтение данных о статьях из базы данных MySQL через Go приложение на бекенде
   1. В go.mod описать добавление новых зависимостей для работы с базой (см. Пример проекта и пункт “Добавление новых зависимостей”)
   2. Посты из секций featured-posts и most-recent-post хранятся в одной таблице. Разница только в том, что для featured-posts в колонке featured проставлена 1(true), а для most-recent-posts 0(false). Учитывайте это при составлении SQL-запроса

### Соглашение по форматированию SQL

<https://docs.google.com/document/d/1hCIbR1FmQbJ5Fx_LR9xiTV6wS4v1L6pV_-Vgz56HVZc/edit?usp=sharing>

Обратите внимание, что в GO так же нужно будет форматировать в несколько строк. Чтобы в GO писать SQL в несколько строк нужно использовать обратные кавычки `

### Примеры

#### SQL для создания таблицы

CREATE TABLE post

(

`post\_id` INT NOT NULL AUTO\_INCREMENT,

`title` VARCHAR(255) NOT NULL,

`subtitle` VARCHAR(255) NOT NULL,

`publish\_date` TIMESTAMP,

`featured` TINYINT(1),

PRIMARY KEY (`post\_id`)

) ENGINE = InnoDB

CHARACTER SET = utf8mb4

COLLATE utf8mb4\_unicode\_ci

;

#### SQL для формирования данных о постах

INSERT INTO post (title, subtitle, publish\_date, featured) VALUES ('Post about traveling', 'My best post', '9/25/2015', 1);

INSERT INTO post (title, subtitle, publish\_date, featured) VALUES ('Post number 2 about traveling', 'My best number 2 post', '9/25/2015', 0);

#### Подключение к базе данных

В Golang для подключения к базе данных используется стандартная ф-ия sql.Open, куда нужно передать тип базы данных(mysql) и url для подключения.

В этом url защиты следующие параметры:

USER:PASS@tcp(HOST:PORT)/DATABASE\_NAME?charset=utf8mb4&collation=utf8mb4\_unicode\_ci&parseTime=true

Параметры после ? это параметры отвечающие за установление кодировки и прочее. Просто их скопируйте.

USER и PASS - это ваши доступы для базы данных, например root 1234

HOST и PORT - хост и порт базы, если база локально развёрнута, то localhost:3306

#### Добавление новых зависимостей

Если есть необходимость добавить новые зависимости в проект(например для работы с БД) нужно прописать из в go.mod

module blog

go 1.20

require (

github.com/go-sql-driver/mysql v1.7.0

github.com/jmoiron/sqlx v1.3.5

)

И выполнить в консоли - данная команда установит нужные зависимости для проекта

go mod tidy

После чего можно как обычно скомпилировать проект

#### Примерочный проект:

https://drive.google.com/file/d/1-DHNh1giL2b7TXMZKmZucgzStxDLvMdS/view?usp=drivesdk