

Оглавление.

Практика 1.....	3
Практика 2.....	10
Практика 3.....	14
Создание приложения.....	22
Вывод.....	32
Список литературы	33

Практика 1

Задание: создать базу данных и таблицы в ней по выбранной теме, на основе разработанных моделей. Результат работы в виде отчета должен содержать:

- снимки экрана (скриншоты) процесса разработки;
- снимок экрана завершенной базы данных;
- снимки экрана (скриншоты) списка созданных таблиц;
- снимки экрана (скриншоты) структуры созданных таблиц;
- снимки экрана (скриншоты) содержания созданных таблиц;
- итоговый скрипт БД.

Выбранная тема: создание анимационного фильма. Физическая модель представлена на рисунке 1.

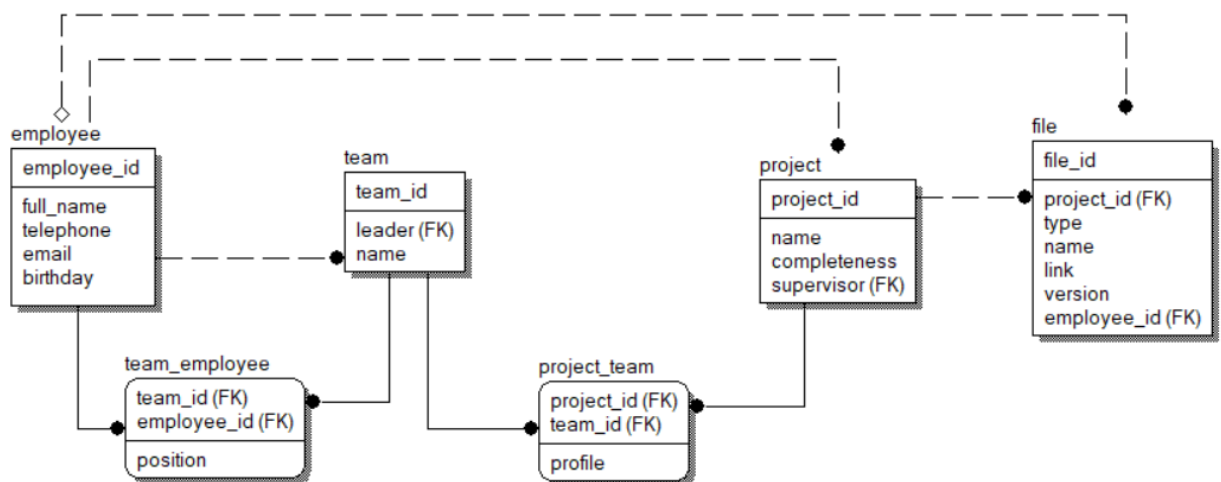


Рисунок 1 – Физическая модель

```
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 8.0.30 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use animation_studio;
ERROR 1049 (42000): Unknown database 'animation_studio'
mysql> create database animation_studio;
Query OK, 1 row affected (0.06 sec)

mysql> use animation_studio;
Database changed
mysql> create table employees
-> (
->   id_employee int(10) AUTO_INCREMENT primary key,
->   full_name   varchar(100) NOT NULL,
->   telephone   varchar(20),
->   email        varchar(50),
->   birthday     date,
->   CHECK (email LIKE '%_@_%._%')
-> );
```

Рисунок 2 – Скриншот процесса разработки
(подключение к серверу), создание базы данных и создание таблицы
сотрудников

```
mysql> create table teams
-> (
->   id_team int(10) AUTO_INCREMENT primary key,
->   leader  int NOT NULL,
->   name    varchar(100),
->   FOREIGN KEY (leader) REFERENCES employees (id_employee) ON UPDATE CASCADE ON DELETE CASCADE
-> );
Query OK, 0 rows affected, 1 warning (0.09 sec)

mysql> create table projects
-> (
->   id_project  int(10) AUTO_INCREMENT primary key,
->   name        varchar(100) NOT NULL,
->   completeness bool default (false),
->   supervisor  int NOT NULL,
->   FOREIGN KEY (supervisor) REFERENCES employees (id_employee) ON UPDATE CASCADE ON DELETE CASCADE
-> );
Query OK, 0 rows affected, 1 warning (0.09 sec)

mysql> create table files
-> (
->   id_file      int(10) AUTO_INCREMENT primary key,
->   id_project   int,
->   type         enum ('audio','animation','video','text') default ('text'),
->   name         varchar(500),
->   version      varchar(35),
->   id_employee  int,
->   FOREIGN KEY (id_project) REFERENCES projects (id_project) ON UPDATE CASCADE ON DELETE CASCADE,
->   FOREIGN KEY (id_employee) REFERENCES employees (id_employee) ON UPDATE CASCADE ON DELETE CASCADE
-> );
```

Рисунок 3 – Скриншот создания таблиц

```
mysql> create table teams_employees
-> (
->   id_team      int(10),
->   id_employee  int(10),
->   position     varchar(100) NOT NULL,
->   FOREIGN KEY (id_team) REFERENCES teams (id_team) ON UPDATE CASCADE ON DELETE CASCADE,
->   FOREIGN KEY (id_employee) REFERENCES employees (id_employee) ON UPDATE CASCADE ON DELETE CASCADE,
->   primary key (id_team, id_employee)
-> );
Query OK, 0 rows affected, 2 warnings (0.09 sec)

mysql> create table projects_teams
-> (
->   id_project int(10),
->   id_team    int(10),
->   profile    varchar(100) NOT NULL,
->   FOREIGN KEY (id_team) REFERENCES teams (id_team) ON UPDATE CASCADE ON DELETE CASCADE,
->   FOREIGN KEY (id_project) REFERENCES projects (id_project) ON UPDATE CASCADE ON DELETE CASCADE,
->   primary key (id_team, id_project)
-> );
Query OK, 0 rows affected, 2 warnings (0.08 sec)
```

Рисунок 4 – Скриншот создания таблиц

```
mysql> show tables;
+-----+
| Tables_in_animation_studio |
+-----+
| employees                  |
| files                      |
| projects                   |
| projects_teams             |
| teams                      |
| teams_employees            |
+-----+
6 rows in set (0.00 sec)

mysql> INSERT INTO employees (full_name, telephone, email) VALUES ('Bespalov Mecheslav Protasyevich', '8005553535', 'sad@mail.com');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO employees (full_name, telephone, email, birthday) VALUES ('Isakov Martyn Yaroslavovich', '8006663535', 'sadam@mail.com', '1986-10-01');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO employees (full_name, telephone, email) VALUES ('Schukin Emelyan Valentinovich', '8005513655', 'mamam@mail.com');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO employees (full_name, telephone, email, birthday) VALUES ('Tikhonova Dominika Artemovna', '8006663535', 'adolk@mail.com', '2000-01-10');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO employees (full_name, telephone, birthday) VALUES ('Rozhkova Vladlena Artemovna', '8005553535', '1976-07-19');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO employees (full_name) VALUES ('Mamontova Vesta Frolovna'),
->                                             ('Frolova Luiza Fedoseevna'),
->                                             ('Kudryavtsev Azariy Serapionovich');
Query OK, 3 rows affected (0.05 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

Рисунок 5 – Скриншот вывода всех созданных таблиц и заполнения таблицы

```
mysql> INSERT INTO teams (leader, name) VALUES (1, 'animators'),
->                                             (4, 'designers'),
->                                             (6, 'sound engineers');
Query OK, 3 rows affected (0.06 sec)
Records: 3 Duplicates: 0 Warnings: 0

mysql>
mysql> INSERT INTO projects (name, supervisor) VALUES ('The owl house', 1),
->                                             ('Wander Over Yonder', 5);
Query OK, 2 rows affected (0.01 sec)
Records: 2 Duplicates: 0 Warnings: 0

mysql> INSERT INTO projects (name, supervisor, completeness) VALUES ('Samurai Jack', 1, true);
Query OK, 1 row affected (0.00 sec)

mysql>
mysql> INSERT INTO files (id_project, type, name, version, id_employee) VALUES (1, 'animation', 'start gif', 'v1', 2),
->                                     (1, 'animation', 'end gif', 'v1', 2),
->                                     (1, 'audio', 'outro', 'v1', 6),
->                                     (1, 'audio', 'outro', 'v2', 7),
->                                     (1, 'text', 'scene', 'v1', 1),
->                                     (2, 'animation', 'start gif', 'v1', 3),
->                                     (3, 'video', 'final project', 'final', 2),
->                                     (3, 'audio', 'outro', 'v1', 8),
->                                     (2, 'audio', 'intro', 'v1', 8),
->                                     (2, 'text', 'scenel', 'v1', 4),
->                                     (2, 'text', 'scenel', 'v2', 1);
Query OK, 11 rows affected (0.01 sec)
Records: 11 Duplicates: 0 Warnings: 0
```

Рисунок 6 – Скриншот заполнения таблиц

```
mysql> INSERT INTO teams_employees (id_team, id_employee, position) VALUES (1, 2, 'storyboarder'),
->                                     (1, 3, 'animator'),
->                                     (1, 4, 'animator'),
->                                     (2, 5, 'art designer'),
->                                     (3, 7, 'sound engineer'),
->                                     (3, 8, 'singer');
Query OK, 6 rows affected (0.01 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql>
mysql> INSERT INTO projects_teams (id_project, id_team, profile) VALUES (1, 1, 'animation'),
->                                     (1, 3, 'sound'),
->                                     (2, 1, 'animation'),
->                                     (2, 3, 'music'),
->                                     (3, 2, 'design'),
->                                     (3,1, 'animation');
Query OK, 6 rows affected (0.00 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql> SELECT * FROM employees;
+-----+-----+-----+-----+-----+
| id_employee | full_name                               | telephone | email           | birthday |
+-----+-----+-----+-----+-----+
| 1 | Bespalov Mecheslav Protasyevich | 8005553535 | sad@mail.com    | NULL     |
| 2 | Isakov Martyn Yaroslavovich      | 8006663535 | sadad@mail.com  | 1986-10-01 |
| 3 | Schukin Emelyan Valentinovich    | 8005513655 | mamam@mail.com  | NULL     |
| 4 | Tikhonova Dominika Artemovna     | 8006663535 | adolk@mail.com  | 2000-01-10 |
| 5 | Rozhkova Vladlena Artemovna     | 8005553535 | NULL            | 1976-07-19 |
| 6 | Mamontova Vesta Frolovna         | NULL       | NULL            | NULL     |
| 7 | Frolova Luiza Fedoseevna         | NULL       | NULL            | NULL     |
| 8 | Kudryavtsev Azariy Serapionovich | NULL       | NULL            | NULL     |
+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

Рисунок 7 – Скриншот создания таблиц и вывода данных

```
mysql> SELECT * FROM teams;
+-----+-----+-----+
| id_team | leader | name           |
+-----+-----+-----+
| 1 | 1 | animators      |
| 2 | 4 | designers      |
| 3 | 6 | sound engineers |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> SELECT * FROM projects;
+-----+-----+-----+-----+-----+
| id_project | name           | completeness | supervisor |
+-----+-----+-----+-----+-----+
| 1 | The owl house | 0 | 1 |
| 2 | Wander Over Yonder | 0 | 5 |
| 3 | Samurai Jack | 1 | 1 |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Рисунок 8 – Скриншот заполненных таблиц

```
mysql> SELECT * FROM files;
```

id_file	id_project	type	name	version	id_employee
1	1	animation	start gif	v1	2
2	1	animation	end gif	v1	2
3	1	audio	outro	v1	6
4	1	audio	outro	v2	7
5	1	text	scene	v1	1
6	2	animation	start gif	v1	3
7	3	video	final project	final	2
8	3	audio	outro	v1	8
9	2	audio	intro	v1	8
10	2	text	scene1	v1	4
11	2	text	scene1	v2	1

```
11 rows in set (0.00 sec)
```



```
mysql> SELECT * FROM teams_employees;
```

id_team	id_employee	position
1	2	storyboarder
1	3	animator
1	4	animator
2	5	art designer
3	7	sound engineer
3	8	singer

```
6 rows in set (0.00 sec)
```

Рисунок 9 – Скриншот заполненных таблиц

```
mysql> SELECT * FROM projects_teams;
```

id_project	id_team	profile
1	1	animation
2	1	animation
3	1	animation
3	2	design
1	3	sound
2	3	music

```
6 rows in set (0.00 sec)
```



```
mysql> describe employees;
```

Field	Type	Null	Key	Default	Extra
id_employee	int	NO	PRI	NULL	auto_increment
full_name	varchar(100)	NO		NULL	
telephone	varchar(20)	YES		NULL	
email	varchar(50)	YES		NULL	
birthday	date	YES		NULL	

```
5 rows in set (0.01 sec)
```



```
mysql> describe teams;
```

Field	Type	Null	Key	Default	Extra
id_team	int	NO	PRI	NULL	auto_increment
leader	int	NO	MUL	NULL	
name	varchar(100)	YES		NULL	

```
3 rows in set (0.00 sec)
```

Рисунок 10 – Скриншот заполненных таблиц и описание столбцов таблицы

```
mysql> describe projects;
```

Field	Type	Null	Key	Default	Extra
id_project	int	NO	PRI	NULL	auto_increment
name	varchar(100)	NO		NULL	
completeness	tinyint(1)	YES		false	DEFAULT_GENERATED
supervisor	int	NO	MUL	NULL	

```
4 rows in set (0.00 sec)
```



```
mysql> describe files;
```

Field	Type	Null	Key	Default	Extra
id_file	int	NO	PRI	NULL	auto_increment
id_project	int	YES	MUL	NULL	
type	enum('audio','animation','video','text')	YES		_cp866\'text\'	DEFAULT_GENERATED
name	varchar(500)	YES		NULL	
version	varchar(35)	YES		NULL	
id_employee	int	YES	MUL	NULL	

```
6 rows in set (0.00 sec)
```



```
mysql> describe teams_employees;
```

Field	Type	Null	Key	Default	Extra
id_team	int	NO	PRI	NULL	
id_employee	int	NO	PRI	NULL	
position	varchar(100)	NO		NULL	

```
3 rows in set (0.00 sec)
```

Рисунок 11 – Скриншот описания столбцов таблиц

```
mysql> describe projects_teams;
```

Field	Type	Null	Key	Default	Extra
id_project	int	NO	PRI	NULL	
id_team	int	NO	PRI	NULL	
profile	varchar(100)	NO		NULL	

```
3 rows in set (0.00 sec)
```



```
mysql>
mysql> show tables;
```

Tables_in_animation_studio
employees
files
projects
projects_teams
teams
teams_employees

```
6 rows in set (0.00 sec)
```

Рисунок 12 – Скриншот описания столбцов таблицы

Практика 2

Задание: изучить и создать выборку и сортировку данных. Изучить и применить операторы для изменения данных в таблицах. Результат работы в виде отчета должен содержать:

- снимки экрана (скриншоты) выборки данных по различным параметрам (по каждому оператору);
- снимок экрана (скриншоты) сортировки данных;
- снимки экрана (скриншоты) применения операторов изменения данных в таблицах Вашей базы данных.

```
mysql> select * from employees;
```

id_employee	full_name	telephone	email	birthday
1	Bespalov Mecheslav Protasyevich	8005553535	sad@mail.com	NULL
2	Isakov Martyn Yaroslavovich	8006663535	sadad@mail.com	1986-10-01
3	Schukin Emelyan Valentinovich	8005513655	mamam@mail.com	NULL
4	Tikhonova Dominika Artemovna	8006663535	adolk@mail.com	2000-01-10
5	Rozhkova Vladlena Artemovna	8005553535	NULL	1976-07-19
6	Mamontova Vesta Frolovna	NULL	NULL	NULL
7	Frolova Luiza Fedoseevna	NULL	NULL	NULL
8	Kudryavtsev Azariy Serapionovich	NULL	NULL	NULL

```
8 rows in set (0.00 sec)
```



```
mysql> select id_employee, full_name from employees where telephone like '____35';
```

id_employee	full_name
1	Bespalov Mecheslav Protasyevich
2	Isakov Martyn Yaroslavovich
4	Tikhonova Dominika Artemovna
5	Rozhkova Vladlena Artemovna

```
4 rows in set (0.00 sec)
```

Рисунок 12 – Выполнение запроса: выбрать id и ФИО сотрудников, у которых телефон заканчивается на 35.

```
mysql> select * from projects;
```

id_project	name	completeness	supervisor
1	The owl house	0	1
2	Wander Over Yonder	0	5
3	Samurai Jack	1	1

```
3 rows in set (0.00 sec)
```

Рисунок 13 – Выполнение запроса: выбрать все записи из таблицы проектов

```
mysql> select * from projects where id_project in (1,2) and supervisor not in (5) and completeness in (0);
+-----+-----+-----+-----+
| id_project | name          | completeness | supervisor |
+-----+-----+-----+-----+
|          1 | The owl house |           0 |          1 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Рисунок 14 – Выполнение запроса: выбрать все записи, где id 1 или 2, управляющий не с 5 id и проект не завершен

```
mysql> select * from employees where (id_employee in (1,3,5) and email like 's%') or birthday < '1988-01-01';
+-----+-----+-----+-----+-----+
| id_employee | full_name                | telephone | email          | birthday |
+-----+-----+-----+-----+-----+
|          1 | Bepalov Mecheslav Protasyevich | 8005553535 | sad@mail.com   | NULL     |
|          2 | Isakov Martyn Yaroslavovich  | 8006663535 | sadad@mail.com | 1986-10-01 |
|          5 | Rozhkova Vladlena Artemovna  | 8005553535 | NULL           | 1976-07-19 |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Рисунок 15 – Выполнение запроса: выбрать все записи, где id 1,3,5 и email начинается на s или день рождения раньше 1988 года

```
mysql> select * from employees where id_employee in (1,3,5) and (email like 's%' or birthday < '1988-01-01');
+-----+-----+-----+-----+-----+
| id_employee | full_name                | telephone | email          | birthday |
+-----+-----+-----+-----+-----+
|          1 | Bepalov Mecheslav Protasyevich | 8005553535 | sad@mail.com   | NULL     |
|          5 | Rozhkova Vladlena Artemovna  | 8005553535 | NULL           | 1976-07-19 |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Рисунок 16 – Выполнение запроса: выбрать все записи, где id 1,3,5 и email начинается на s или день рождения раньше 1988 года

```
mysql> select full_name, birthday from employees order by full_name asc;
```

full_name	birthday
Bespalov Mecheslav Protasyevich	NULL
Frolova Luiza Fedoseevna	NULL
Isakov Martyn Yaroslavovich	1986-10-01
Kudryavtsev Azariy Serapionovich	NULL
Mamontova Vesta Frolovna	NULL
Rozhkova Vladlena Artemovna	1976-07-19
Schukin Emelyan Valentinovich	NULL
Tikhonova Dominika Artemovna	2000-01-10

```
8 rows in set (0.00 sec)
```



```
mysql> select full_name, birthday from employees order by birthday desc;
```

full_name	birthday
Tikhonova Dominika Artemovna	2000-01-10
Isakov Martyn Yaroslavovich	1986-10-01
Rozhkova Vladlena Artemovna	1976-07-19
Bespalov Mecheslav Protasyevich	NULL
Schukin Emelyan Valentinovich	NULL
Mamontova Vesta Frolovna	NULL
Frolova Luiza Fedoseevna	NULL
Kudryavtsev Azariy Serapionovich	NULL

```
8 rows in set (0.00 sec)
```

Рисунок 17 – Выполнение запросов: выбрать ФИО и день рождения человека, отсортированные по алфавиту по ФИО и по уменьшению дня рождения

```
mysql> ALTER TABLE projects ADD COLUMN link varchar(500) after name;
Query OK, 0 rows affected (0.09 sec)
Records: 0 Duplicates: 0 Warnings: 0
```



```
mysql> describe projects;
```

Field	Type	Null	Key	Default	Extra
id_project	int	NO	PRI	NULL	auto_increment
name	varchar(100)	NO		NULL	
link	varchar(500)	YES		NULL	
completeness	tinyint(1)	YES		false	DEFAULT_GENERATED
supervisor	int	NO	MUL	NULL	

```
5 rows in set (0.00 sec)
```

Рисунок 18 – Изменение таблицы проектов (добавление колонки)

```
mysql> alter table projects modify name varchar(100) not null;
Query OK, 0 rows affected (0.14 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> describe projects;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra           |
+-----+-----+-----+-----+-----+-----+
| id_project | int           | NO   | PRI | NULL    | auto_increment |
| name       | varchar(100)  | NO   |     | NULL    |                 |
| link       | varchar(500)  | YES  |     | NULL    |                 |
| completeness | tinyint(1)   | YES  |     | false   | DEFAULT_GENERATED |
| supervisor | int           | NO   | MUL | NULL    |                 |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Рисунок 19 – Изменение таблицы проектов (изменение типа данных)

```
mysql> select * from employees;
+-----+-----+-----+-----+-----+
| id_employee | full_name          | telephone | email          | birthday |
+-----+-----+-----+-----+-----+
| 1 | Bepalov Mecheslav Protasyevich | 8005553535 | sad@mail.com | NULL     |
| 2 | Isakov Martyn Yaroslavovich    | 8006663535 | sadad@mail.com | 1986-10-01 |
| 3 | Schukin Emelyan Valentinovich  | 8005513655 | mamam@mail.com | NULL     |
| 4 | Tikhonova Dominika Artemovna   | 8006663535 | adolk@mail.com | 2000-01-10 |
| 5 | Rozhkova Vladlena Artemovna    | 8005553535 | NULL          | 1976-07-19 |
| 6 | Mamontova Vesta Frolovna       | NULL       | NULL          | NULL     |
| 7 | Frolova Luiza Fedoseevna       | NULL       | NULL          | NULL     |
| 8 | Kudryavtsev Azariy Serapionovich | NULL       | NULL          | NULL     |
+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql> delete from employees where id_employee = 7;
Query OK, 1 row affected (0.06 sec)

mysql> select * from employees;
+-----+-----+-----+-----+-----+
| id_employee | full_name          | telephone | email          | birthday |
+-----+-----+-----+-----+-----+
| 1 | Bepalov Mecheslav Protasyevich | 8005553535 | sad@mail.com | NULL     |
| 2 | Isakov Martyn Yaroslavovich    | 8006663535 | sadad@mail.com | 1986-10-01 |
| 3 | Schukin Emelyan Valentinovich  | 8005513655 | mamam@mail.com | NULL     |
| 4 | Tikhonova Dominika Artemovna   | 8006663535 | adolk@mail.com | 2000-01-10 |
| 5 | Rozhkova Vladlena Artemovna    | 8005553535 | NULL          | 1976-07-19 |
| 6 | Mamontova Vesta Frolovna       | NULL       | NULL          | NULL     |
| 8 | Kudryavtsev Azariy Serapionovich | NULL       | NULL          | NULL     |
+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

Рисунок 20 – Удаление строки из таблицы

Практика 3

Задания:

1. Научится формировать модель БД с помощью инструментов СУБД (MySQL Workbench, dbForge Studio, PostgreSQL – по выбору студента) по своей теме.
2. Научится осуществлять перенос своей БД на другой сервер.
3. Изучить команды модификации данных (DML).
4. Осуществить выборку данных по своей теме с помощью различных операторов.
5. Изучить и применить к своей БД хранимые процедуры, функции и триггеры.

Создание модели произведено в программе MySQL Workbench – визуальном редакторе таблиц, генераторе кода и консоли исполнения запросов. Модель данных приведена на рисунке 13. Генерация кода для резервного копирования базы данных представлена на рисунках 14–17.

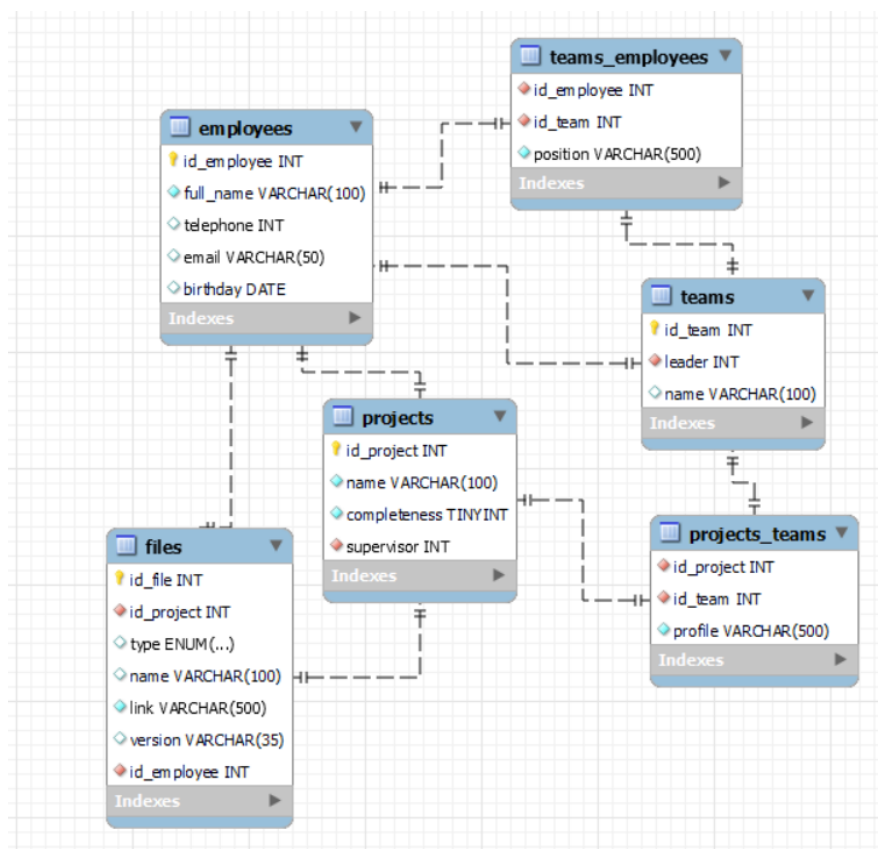


Рисунок 13 - Модель данных в WorkBench

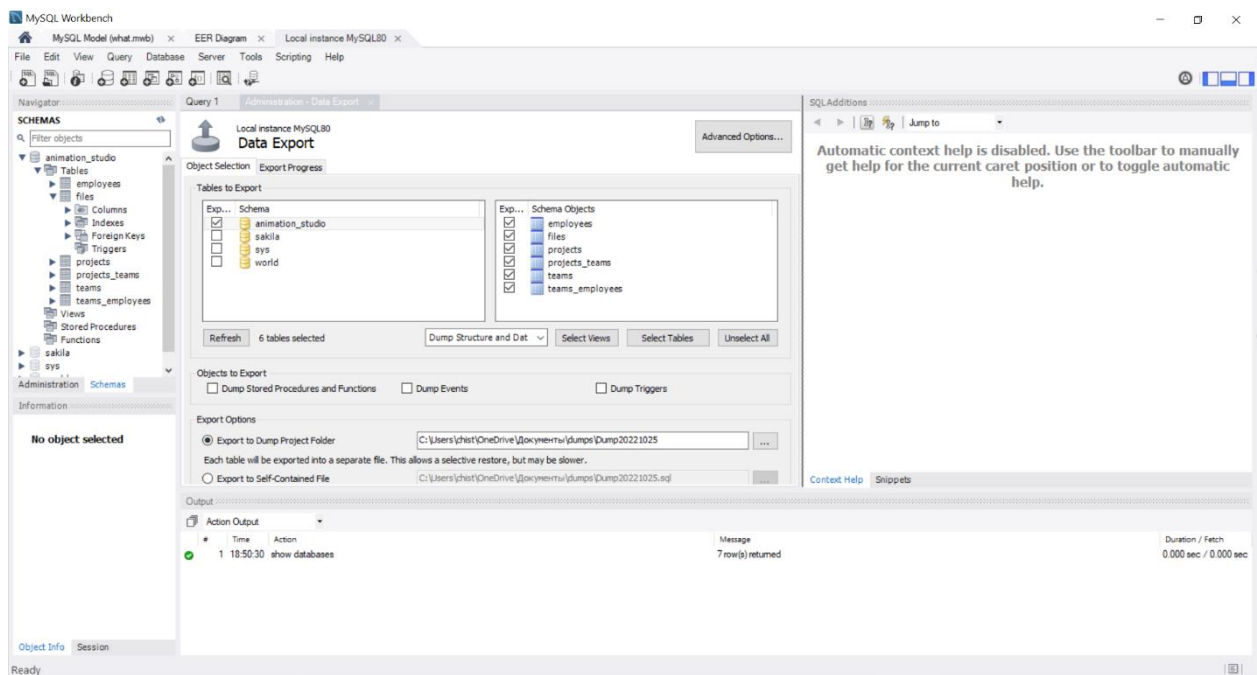


Рисунок 14 - Экспорт базы данных

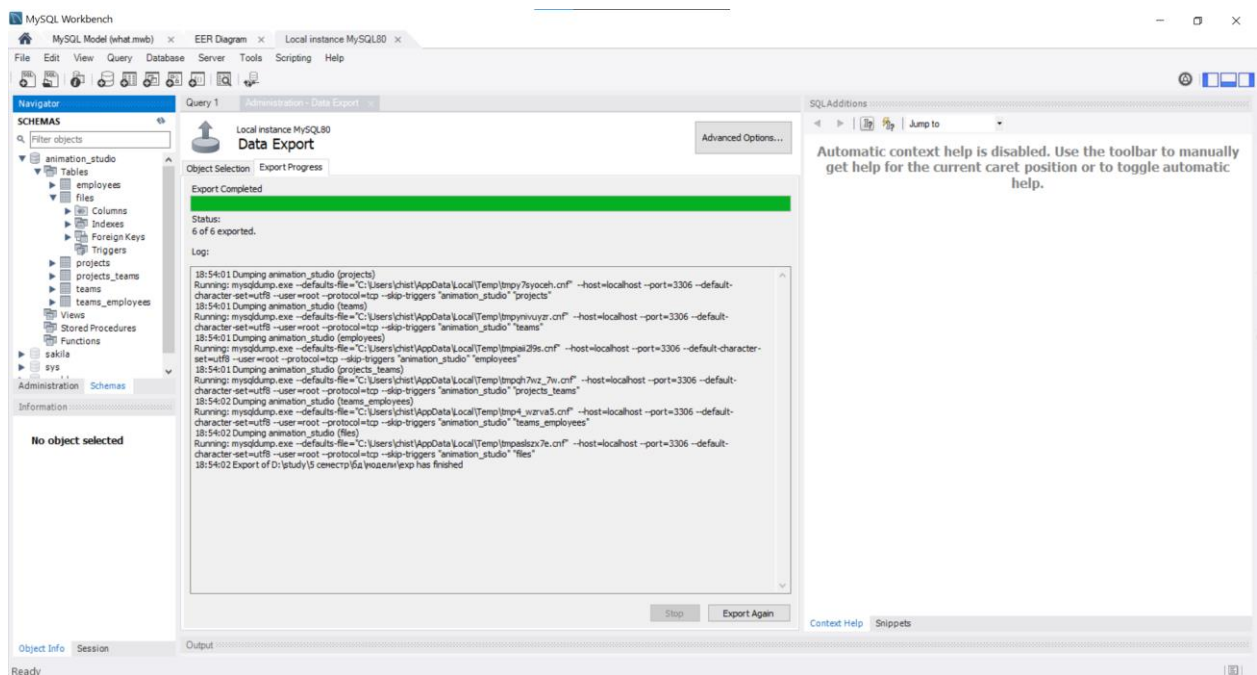


Рисунок 15 - Экспорт базы данных

Имя	Дата изменения	Тип	Размер
animation_studio_employees.sql	25.10.2022 18:54	Файл "SQL"	3 КБ
animation_studio_files.sql	25.10.2022 18:54	Файл "SQL"	3 КБ
animation_studio_projects.sql	25.10.2022 18:54	Файл "SQL"	3 КБ
animation_studio_projects_teams.sql	25.10.2022 18:54	Файл "SQL"	3 КБ
animation_studio_teams.sql	25.10.2022 18:54	Файл "SQL"	3 КБ
animation_studio_teams_employees.sql	25.10.2022 18:54	Файл "SQL"	3 КБ

Рисунок 16 - Результат экспорта

```

animation_studio_employees.sql – Блокнот
Файл  Правка  Формат  Вид  Справка
/*!40101 SET @@SQL_MODE=@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @@OLD_SQL_NOTES=@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `employees`
--

DROP TABLE IF EXISTS `employees`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `employees` (
  `id_employee` int NOT NULL AUTO_INCREMENT,
  `full_name` varchar(100) NOT NULL,
  `telephone` varchar(20) DEFAULT NULL,
  `email` varchar(50) DEFAULT NULL,
  `birthday` date DEFAULT NULL,
  PRIMARY KEY (`id_employee`),
  CONSTRAINT `employees_chk_1` CHECK ((`email` like _cp866'%@%.%'))
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `employees`
--

LOCK TABLES `employees` WRITE;
/*!40000 ALTER TABLE `employees` DISABLE KEYS */;
INSERT INTO `employees` VALUES (1,'Bespalov Mechaslav Protasyevich','8005553535','sad@mail.com',NULL),(2,'Isakov
Martyn Yaroslavovich','8006663535','sadar@mail.com','1986-10-01'),(3,'Schukin Emelvan
Стр 1, столб 1    100%  Windows (CRLF)  UTF-8

```

Рисунок 17 - Результат экспорта

Написание и выполнение запросов команд модификации данных (DML) было выполнено в практике 2 (добавление данных в таблицы и изменение данных в таблицах). Примеры запросов выборки данных (DQL) будут рассмотрены далее в процедурах и функциях.

Созданные триггеры, хранимые процедуры и функции показаны на рисунках 18–19. Их листинги представлены ниже (листинг 1–3).

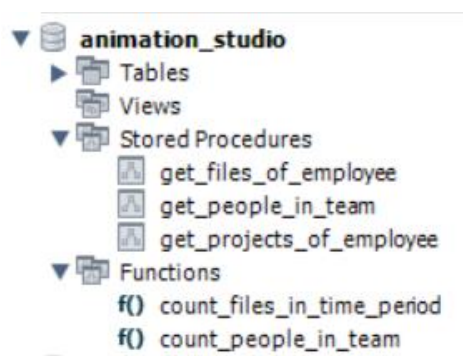


Рисунок 18 – Созданные хранимые процедуры и функции

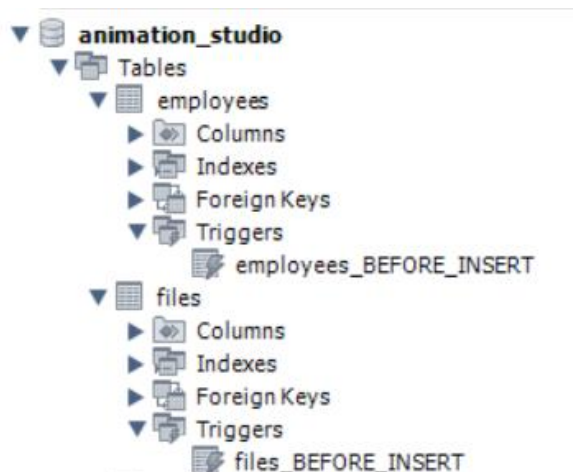


Рисунок 19 – Созданные триггеры

Функция count_files_in_time_period принимает два аргумента (две даты) и считает количество файлов, загруженных в заданный промежуток времени.

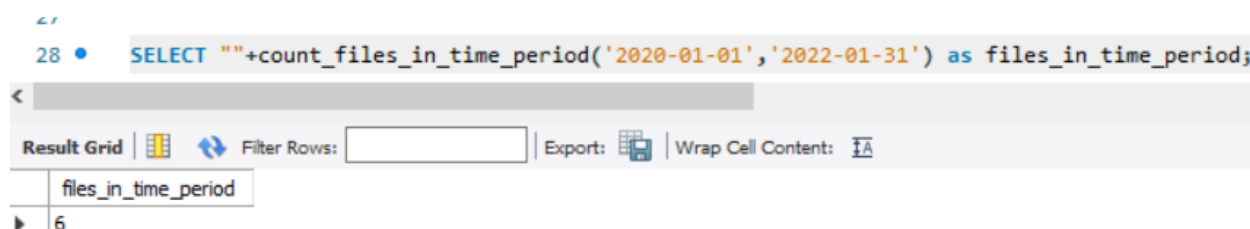


Рисунок 20 – Выполнение функции count_files_in_time_period

Функция count_people_in_team принимает id команды и считает количество человек в этой команде.

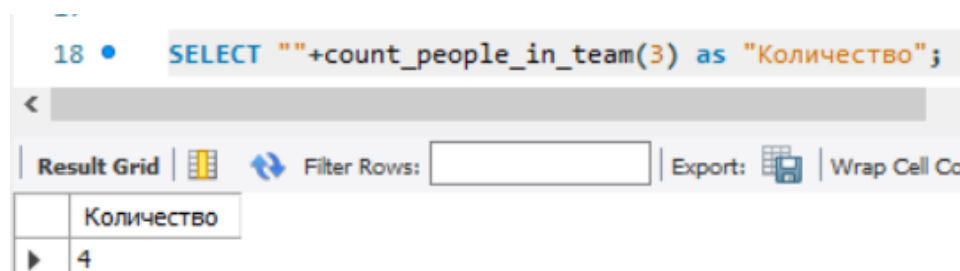


Рисунок 21 – Выполнение функции count_people_in_team

Листинг 1 – Код функций

```
CREATE DEFINER=`root`@`localhost` FUNCTION `count_files_in_time_period`(date_from DATE,
date_to DATE) RETURNS int
DETERMINISTIC
BEGIN
DECLARE amount INT;
select COUNT(*)
INTO amount
from files
```



```

where date > date_from
and date < date_to;
RETURN amount;
END

CREATE DEFINER=`root`@`localhost` FUNCTION `count_people_in_team`(id_team_in INT)
RETURNS int
DETERMINISTIC
BEGIN DECLARE amount INT;
select COUNT(*)
INTO amount
from employees
inner join teams_employees on employees.id_employee = teams_employees.id_employee
inner join teams on teams.id_team = teams_employees.id_team
where teams.id_team = id_team_in;
RETURN amount;
END

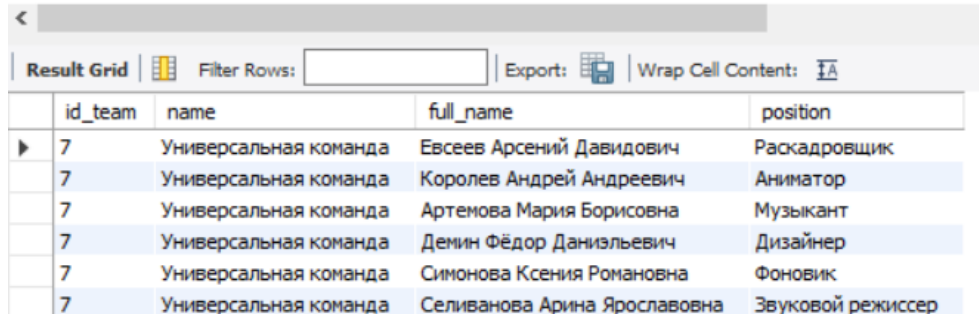
```

Процедура `get_people_in_team` принимает `id` команды и выводит ФИО и должность работников в этой команде.

```

18 • call get_people_in_team(7);
19

```



id_team	name	full_name	position
7	Универсальная команда	Евсеев Арсений Давидович	Раскадровщик
7	Универсальная команда	Королев Андрей Андреевич	Аниматор
7	Универсальная команда	Артемова Мария Борисовна	Музыкант
7	Универсальная команда	Демин Фёдор Даниэльевич	Дизайнер
7	Универсальная команда	Симонова Ксения Романовна	Фофик
7	Универсальная команда	Селиванова Арина Ярославовна	Звуковой режиссер

Рисунок 22 – Выполнение процедуры `get_people_in_team`

Процедура `get_projects_of_employee` принимает `id` работника и выводит названия проектов, в которых участвует данный работник.

20 • `call get_projects_of_employee(9);`

Result Grid | Filter Rows: | Export:

	id_project	name
▶	1	Совиный дом
	2	С приветом по планетам
	4	Самурай Джек

Рисунок 23 – Выполнение процедуры `get_projects_of_employee`

Процедура `get_files_of_employee` принимает `id` работника и выводит названия файлов, созданных этим работником.

24 • `call get_files_of_employee(1);`

Result Grid | Filter Rows: | Exp

	id_file	name
▶	5	Сцена 5, фон

Рисунок 24 – Выполнение процедуры `get_files_of_employee`

Листинг 2 – Код хранимых процедур

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_files_of_employee`(in id_empl INT)
BEGIN
    select id_file, name from files where id_employee = id_empl;
END

CREATE DEFINER=`root`@`localhost` PROCEDURE `get_people_in_team`(in id_team_in INT)
BEGIN
    select teams.id_team, teams.name, employees.full_name, teams_employees.position
    from employees
        inner join teams_employees on employees.id_employee = teams_employees.id_employee
        inner join teams on teams.id_team = teams_employees.id_team
    where teams.id_team = id_team_in;
END

CREATE DEFINER=`root`@`localhost` PROCEDURE `get_projects_of_employee`(in empl_id INT)
BEGIN
    select projects.id_project, projects.name
    from employees
        inner join teams_employees on employees.id_employee = teams_employees.id_employee
        inner join teams on teams.id_team = teams_employees.id_team
        inner join projects_teams on projects_teams.id_team = teams.id_team
```

```

inner join projects on projects.id_project = projects_teams.id_project
where employees.id_employee = empl_id
or teams.leader = empl_id
group by projects.id_project;
END

```

Триггер при вставке (before insert) данных в таблицу files присваивает полю дата текущую дату, если не введена другая.

```

34 • INSERT INTO files (id_project, type, name, link, version, id_employee) values
35   (4, 'audio', 'Вступительная песня', 'https://tlum.ru/news/19-multifilmov-pro-kotov-i-kosek/', 'v3', 28);
36 • select * from files;

```

id_file	id_project	type	name	link	version	date	id_employee
14	4	audio	Песня концовки	https://bigpicture.ru/30-sposobov-ispolzovaniy...	v1	2020-09-01	28
15	4	audio	Вступительная песня	https://tlum.ru/news/19-multifilmov-pro-kotov-i...	v2	2022-10-29	28
16	4	picture	Сцена 5, фон	https://klike.net/1899-koty-krasivye-kartinki-50...	v1	2022-10-29	23
17	4	audio	Вступительная песня	https://tlum.ru/news/19-multifilmov-pro-kotov-i...	v3	2022-10-29	28
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 25 – Вывод данных после выполнение запроса и триггера

Триггер при вставке (before insert) данных в таблицу employees меняет поле full_name так, чтобы каждое слово начиналось с заглавной буквы.

```

40 • INSERT INTO employees (full_name) VALUES ('коЗлова Алиса ильинична');
41 • select * from employees;

```

id_employee	full_name	telephone	email	birthday
27	Селиванова Арина Ярославовна	NULL	NULL	NULL
28	Уткин Арсений Арсентьевич	NULL	NULL	NULL
29	Гордеева Софья Максимовна	NULL	NULL	NULL
30	Козлова Алиса Ильинична	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL

Рисунок 26 – Вывод данных после выполнение запроса и триггера

Листинг 3 – Код триггеров

```

CREATE DEFINER=`root`@`localhost` TRIGGER `employees_BEFORE_INSERT` BEFORE INSERT
ON `employees` FOR EACH ROW BEGIN
  DECLARE len INT;
  DECLARE i INT;
  DECLARE charnum INT;
  declare SortedName varchar(100);
  declare input varchar(100);

```

```

SET input = LOWER(new.full_name);
SET len  = LENGTH(input);
SET i = 1;
set charnum = 1;
set SortedName = "";

WHILE (i <= len) DO
    if charnum = 1 then
        set SortedName = concat(SortedName,upper(mid(input,i,1)));
        set charnum = charnum + 1;
    else
        if mid(input,i,1) = ' ' then
            set SortedName = concat(SortedName,' ');
            set charnum = 1;
        else
            set SortedName = concat(SortedName,mid(input,i,1));
            set charnum = charnum + 1;
        end if;
    end if;
    SET i = i + 1;
END WHILE;
SET new.full_name=SortedName;
END

CREATE DEFINER=`root`@`localhost` TRIGGER `files_BEFORE_INSERT` BEFORE INSERT ON
`files` FOR EACH ROW BEGIN
    if new.date is null then
        SET new.date=current_date();
    end if;
END

```

Создание приложения.

Задания:

1. Требования к функционалу:

- регистрация и авторизация пользователя в системе;
- добавление, изменение, удаление, обновление информации;
- поиск данных по 2-м и более параметрам;
- просмотр информации по запросу;
- включение в программный код триггеров, процедур и функций.

2. Требования к интерфейсу:

Интерфейс системы должен поддерживать русский язык.

Интерфейс системы должен быть спроектирован с учетом ролевой модели и уровней доступа пользователей.

Интерфейс системы должен обеспечивать наглядное, интуитивно понятное представление структуры размещенной информации, быстрый и логичный переход к соответствующим разделам.

Навигационные элементы интерфейса должны обеспечивать понимание пользователем их смысла и обеспечивать навигацию по всем доступным пользователю разделам и отображать соответствующую информацию.

Интерфейс системы должен позволять решать задачи пользователя наиболее быстрым и удобным из возможных способов.

Приложение разработано на python с помощью Flask, SQLAlchemy, pymysql, jinja2 и bootstrap.

Скриншоты приложения представлены на рисунках 27–45.

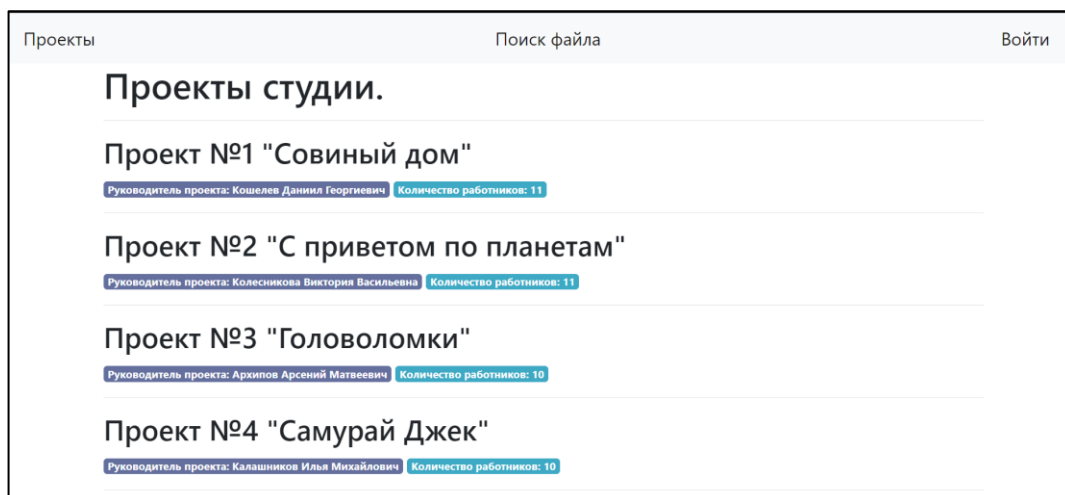


Рисунок 27 – Главная страница приложения

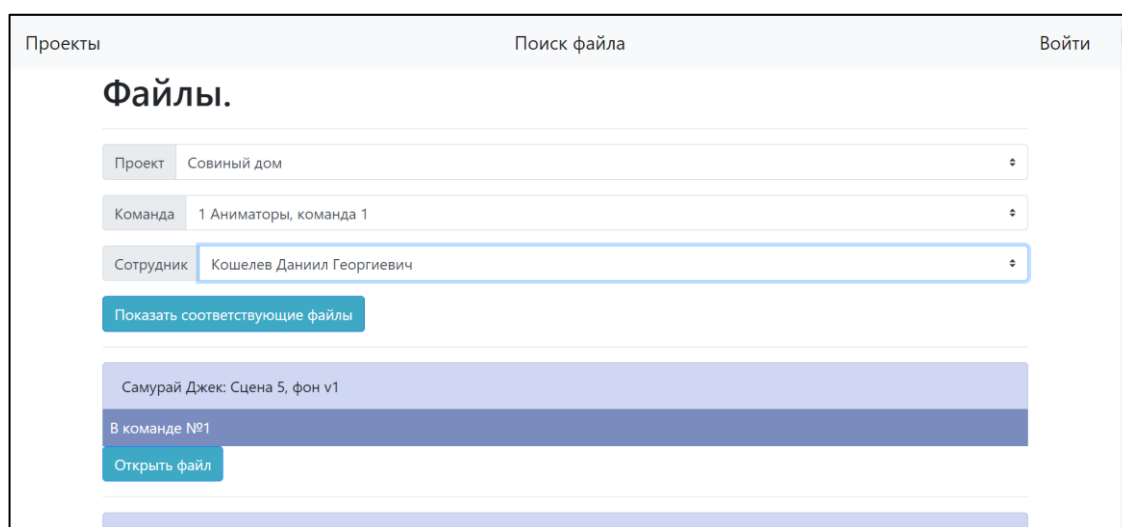


Рисунок 28 – Фильтрация всех файлов по 3-м полям: названию проекта, команде и сотруднику



Рисунок 29 – Найденный файл после нажатия кнопки «Показать соответствующие файлы»

Регистрация

Логин (ФИО)

Антонова Кристина Давидовна

Пароль

...

Повторите пароль

...|

Зарегистрироваться

[Войти](#)

Рисунок 30 – Страница регистрации

На странице регистрации предусмотрены ситуации, если пользователь не заполнит поля или одно из полей (рисунок 32) или если введет несовпадающие пароли (рисунок 33).

Регистрация

- Заполните все поля

Логин (ФИО)

Иванов Иван Иванович

Рисунок 31 – Поля не заполнены

Регистрация

- Пароли не совпадают

Рисунок 32 – Пароли не совпадают

Войти

Логин (ФИО)

Иванов Иван Иванович

Пароль

123

Войти

Нет аккаунта? [Зарегистрироваться](#)

Рисунок 33 – Страница входа

На странице входа предусмотрены ситуации, если пользователь не заполнит поля (рисунок 34) или пользователь введет неверные логин или пароль (рисунок 35).

Войти

- Введите логин и пароль

Логин (ФИО)

Рисунок 34 – Пустые поля

Войти

- Введите корректные логин и пароль

Рисунок 35 – Неверный логин или пароль

Страница пользователя.

Ваши команды:

Создать новую команду

№5 - Дизайнеры

Ваши проекты и файлы внутри них:

Создать новый файл

Головоломки: Песня концовки v1

В команде №5

Открыть файл

Все ваши файлы:

Головоломки: Песня концовки v1

Открыть файл

Самурай Джек: Вступительная песня v1

Открыть файл

Рисунок 36 – Главная страница пользователя (здесь отображены команды сотрудника и созданные им файлы)

Проекты

Поиск файла

Моя страница

Выйти

Создать новую команду

Создатель команды: 123

Название команды

Команда №5

Сотрудник

Кошелев Даниил Георгиевич

Добавить поле

Сотрудник

Калашников Илья Михайлович

✕

Сотрудник

Самсонов Матвей Тимофеевич

✕

Создать команду

Рисунок 37 – Страница создания новой команды

Проекты

Поиск файла

Моя страница

Выйти

Страница пользователя.

Ваши команды:

Создать новую команду

№11 - Команда для теста

№12 - Лучшая команда

№14 - Команда для грустных

№15 - Команда №5

Ваши проекты и файлы внутри них:

Создать новый файл

Все ваши файлы:

Рисунок 38 – Команда №5 создана и отображается в командах сотрудника

Проекты

Поиск файла

Моя страница

Выйти

Команда "Команда №5"

Название команды

Команда №5

Сотрудник

Кошелев Даниил Георгиевич

↕

Сотрудник

Калашников Илья Михайлович

↕

Сотрудник

Самсонов Матвей Тимофеевич

↕

Сотрудник

123

↕

Рисунок 39 – Просмотр состава команды

Проекты	Поиск файла	Моя страница	Выйти
---------	-------------	--------------	-------

Создание файла.

*Название.

Файл №5

Тип файла

picture

Проект

С приветом по планетам

Версия.

вер1

Дата (можно не заполнять).

дд.мм.гггг

Выберите файл

cat.jpg

Создать файл

Рисунок 40 – Страница создания нового файла

Если поле «дата» не заполнена при создании, в это поле будет записано значение текущей на момент создания файла даты (с помощью триггера, созданного в 3 практике).

Проекты	Поиск файла	Моя страница	Выйти
---------	-------------	--------------	-------

Файл №34 - Файл №5.

Файл №5

Тип файла

picture

Проект

С приветом по планетам

вер1

12.11.2022

Заменить файл.

Выберите файл

Файл не выбран

Открыть файл

Обновить файл

Удалить файл

Рисунок 41 – Страница просмотра созданного файла

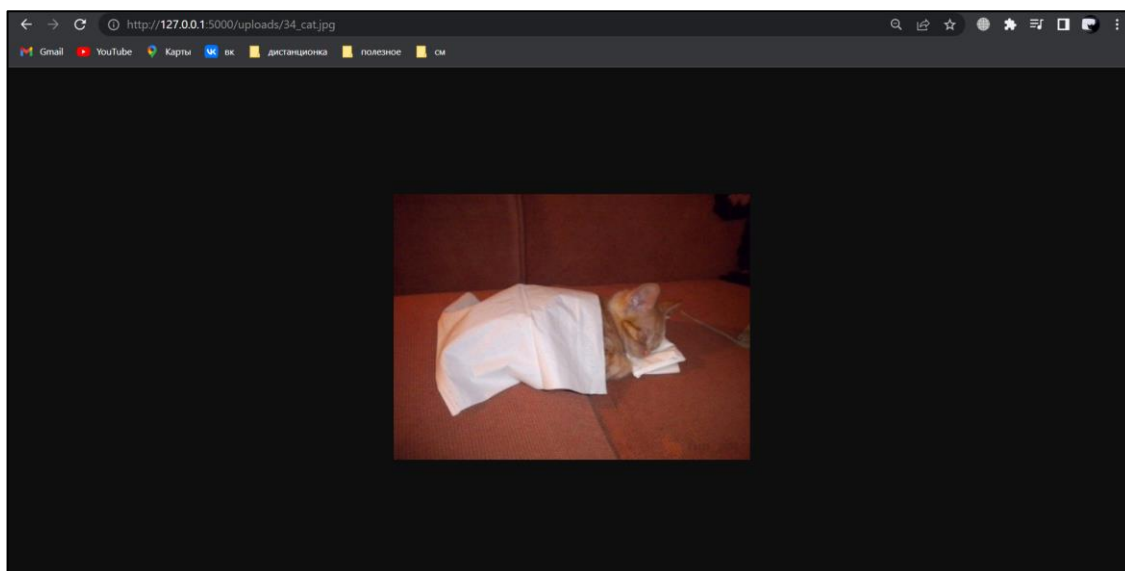


Рисунок 42 – Страница просмотра прикрепленного файла (при нажатии кнопки «Открыть файл»)

Проекты	Поиск файла	Моя страница	Выйти
Файл №34 - Файл №5.			
Файл №5			
Тип файла: picture			
Проект: С приветом по планетам			
вер2			
12.11.2022			
Заменить файл.			
Выберите файл: Файл не выбран			
Открыть файл Обновить файл Удалить файл			

Рисунок 43 – Страница просмотра и редактирования файла

Все ваши файлы:	
С приветом по планетам: Рассылка1 v1	Открыть файл
С приветом по планетам: Файл №5 вер2	Открыть файл

Рисунок 44 – Файл был обновлен и отображается на странице сотрудника

Все ваши файлы:	
С приветом по планетам: Рассылка1 v1	Открыть файл

Рисунок 45 – Файл также можно удалить при нажатии на кнопку «Удалить файл» (рисунок 43)

Файлы сохраняются в локальную директорию внутри проекта, уникальность названий обеспечивается с помощью добавления id файла в начало названия (рисунок 46).

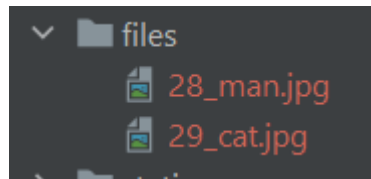


Рисунок 46 – Файлы внутри проекта

Открытие файла в браузере осуществляется с помощью функции `send_from_directory` из `flask` (листинг 4). Также при создании файла происходит проверка на допустимость его расширения (в целях безопасности), допустимы на данный момент расширения: `png`, `jpg`, `txt`, `jpeg`, `mp4`. Ниже представлен листинг (листинг 4) кода функций создания (`create_file`) и просмотра (а также редактирования и удаления) файла (`show_file`).

Листинг 4 – Код функций создания и просмотра файла

```
def allowed_file(filename):
    return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS

def get_file_info(new_file, name, user_id, filename):
    new_file.name = name
    new_file.id_employee = user_id
    new_file.id_project = db.session.query(Projects). \
        filter(Projects.name == request.form.get('project')).one().id_project
    new_file.date = request.form.get('date')
    new_file.link = os.path.join(app.config['UPLOAD_FOLDER'], filename)
    new_file.version = request.form.get('version')
    new_file.type = request.form.get('type')
    if not new_file.date:
        new_file.date = None
    return new_file

@app.route('/create_file', methods=['GET', 'POST'])
@login_required
```

```

def create_file():
    user_id = get_current_id()
    user_logged = user_id is not None
    projects = db.session.query(Projects).all()
    files_amount = db.session.query(Files.id_file).order_by(Files.id_file.desc()).first().id_file
    if request.method == 'POST':
        if 'file' not in request.files:
            flash('Невозможно прочитать файл')
            return redirect(request.url)
        file = request.files['file']
        if file.filename == '':
            flash('Выберите файл для загрузки')
            return redirect(request.url)
        elif file and allowed_file(file.filename):
            filename = str(files_amount + 1) + "_" + secure_filename(file.filename)
            file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
            name = request.form.get('name')
            if not name:
                flash('Введите имя')
            else:
                new_file = get_file_info(Files(), name, user_id, filename)
                db.session.add(new_file)
                db.session.commit()
                return redirect("/user_page")
    return render_template('create_file.html', user_logged=user_logged, projects=projects)

@app.route('/uploads/<name>')
@login_required
def download_file(name):
    return send_from_directory(app.config["UPLOAD_FOLDER"], name)

@app.route('/file/<int:id_file>', methods=['GET', 'POST'])
@login_required
def show_file(id_file):
    projects = db.session.query(Projects).all()
    user_id = get_current_id()

```

```

user_logged = user_id is not None
file = db.session.query(Files).filter(Files.id_file == id_file).one()
if request.method == 'POST':
    button = request.form.get('button')
    if button == 'Удалить файл':
        db.session.delete(file)
        db.session.commit()
        if os.path.exists(file.link):
            os.remove(file.link)
        return redirect('/user_page')
    elif button == 'Обновить файл':
        if user_id != file.id_employee:
            flash('Вы не можете редактировать чужой файл')
        else:
            downloaded_file = request.files['file']
            if downloaded_file.filename == "":
                filename = file.link.split("\\")[2]
            elif allowed_file(downloaded_file.filename):
                filename = str(file.id_file) + "_" + secure_filename(downloaded_file.filename)
                downloaded_file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
                os.remove(file.link)
            name = request.form.get('name')
            if not name:
                flash('Введите имя')
            else:
                file.id_file = id_file
                file = get_file_info(file, name, user_id, filename)
                db.session.commit()

return render_template('file.html', user_logged=user_logged, file=file, projects=projects)

```

Таже приведём листинг функции отображения главной страницы приложения (листинг 5). В ней используется процедура `get_projects_and_supervisors`, возвращающая все проекты компании и руководителей этими проектами, и функция `count_people_in_project`, которая возвращает количество человек во всех командах, которые работают над данным проектом. Главная страница показана на рисунке 27.

Листинг 5 – Код функции отображения главной страницы приложения

```
@app.route('/')
def show_main_page():
    user_id = get_current_id()
    con = get_db_connection()
    with con.cursor() as cursor:
        cursor.execute("call get_projects_and_supervisors")
        projects = cursor.fetchall()
        for project in projects:
            cursor.execute(f"select count_people_in_project({project['id_project']}) as amount")
            amount_of_people = cursor.fetchone()
            project["amount"] = amount_of_people["amount"]
    con.close()
    user_logged = user_id is not None
    return render_template('projects.html', projects=projects, user_logged=user_logged)
```

Итоговый код доступен на гитхабе по следующей ссылке:
https://github.com/Innlock/animation_studio.

Вывод

В ходе выполнения данных практических работ были получены навыки по проектированию и созданию баз данных. В первой части работы была создана модель базы данных и сама база данных. Во второй – база была заполнена данными и были протестированы различные запросы. В третьей части работы были созданы и протестированы триггеры, процедуры и функции. Также были предприняты меры по обеспечению и поддержанию целостности базы данных.

В последней части было реализовано web-приложение с использованием Flask, SQLAlchemy, pymysql, jinja2 и bootstrap. Выполнение работ по дисциплине улучшило навыки разработки приложений с использованием баз данных, написания sql-запросов.