Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»

Факультет среднего профессионального образования

**ОТЧЕТ**

**о Лабораторной работе № 1**

по теме: СОЗДАНИЕ, МОДИФИКАЦИЯ, УДАЛЕНИЕ И ЗАПОЛНЕНИЕ ТАБЛИЦ БАЗЫ ДАННЫХ

Специальность:

09.02.07 Информационные системы и программирование

УП.11 Разработка, администрирование и защита баз данных

|  |  |
| --- | --- |
| Проверил:  Ефимова Т.Н. \_\_\_\_\_\_\_\_\_\_  Дата: «\_\_» \_\_\_\_\_\_\_ 2019г.  Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Выполнил:  студент группы Y2336  \_\_\_\_\_\_ Заерко М.В. |

Санкт-Петербург 2019

1. Цель работы

Формирование умений и навыков для работы в среде СУБД MySQL. Получит практические навыки по созданию, модификации, удаления и заполнению таблиц в командной строке и с помощью файла, осуществлять правильный ввод данных в таблицы

1. Практическое задание

Практическое задание состоит из следующих пунктов:

1. Построить модель данных по заданной предметной области с использованием UML диаграмм (диаграмму классов). Создать базу данных и таблицы согласно индивидуальному заданию.
2. Заполнить таблицы базы данных рабочими данными. Содержание данных в таблицах должно отражать функционирование процессов в заданной предметной области.
3. Просмотреть структуру созданных таблиц.
4. Определить объем занимаемой памяти
5. Таблицы должны быть типа InnoDB и иметь внешний ключ с изменением или удалением ссылочной целостности. Проверить работу ключей
6. Выполнение лабораторной работы
7. Индивидуальное задание

**Вариант** 8. БД Аэропорт Описание предметной области: Необходимо обеспечить продажу билетов на нужный рейс, при отсутствии билетов (необходимого количества билетов) предложить билет на ближайший рейс. БД должна содержать следующий минимальный набор сведений:

• Бортовой номер самолета.

• Тип самолета.

• Количество мест.

• Грузоподъемность.

• Скорость.

• Дата выпуска.

• Налѐт в часах.

• Дата последнего ремонта.

• Назначение самолета.

• Расход топлива.

• Код экипажа.

• Паспортные данные членов экипажа.

• Номер рейса,

• Дата вылета,

• Время вылета.

• Аэропорт вылета,

• Аэропорт назначения.

• Расстояние.

• ФИО пассажира.

• Паспортные данные.

• Номер места.

• Тип места.

• Цена билета. 24

1. Создайте таблицы, используя необходимые средства поддержки целостности данных для реализации следующих требований:

a) Самолеты, выпущенные ранее 10 лет назад должны проходить профилактический ремонт ежегодно.

b) Билеты продаются не более чем за 30 дней до вылета и не менее чем за 1 час до вылета.

2. Создать запросы:

• Определить расчетное время полета по всем маршрутам.

• Определить расход топлива по всем маршрутам.

• Вывести данные экипажа, совершившего максимальное количество полетов за прошедшую неделю.

• Вывести данные о том, сколько свободных мест оставалось в самолетах, совершавших полет по одному из рейсов за вчерашний день.

• Рассчитать убытки компании за счет непроданных билетов за вчерашний день.

• Вывести список самолетов, которые не ремонтировались в течение более чем 3 лет.

• Определить, каким количеством самолетов каждого типа владеет компания.

• Определить, какой тип самолетов чаще всего летал в заданный аэропорт назначения

• Вывести список самолетов “возраст” которых превышает средний “возраст” самолетов этого типа,

• Определить тип самолетов, летающих во все аэропорты назначения

3. Создать представление для пассажиров авиакомпании.

4. Создать хранимые процедуры: Для поиска и продажи билетов на нужный рейс.

5. Создать необходимые триггеры.

1. Диаграмма классов

Диаграмма классов представлена на рисунке 1.

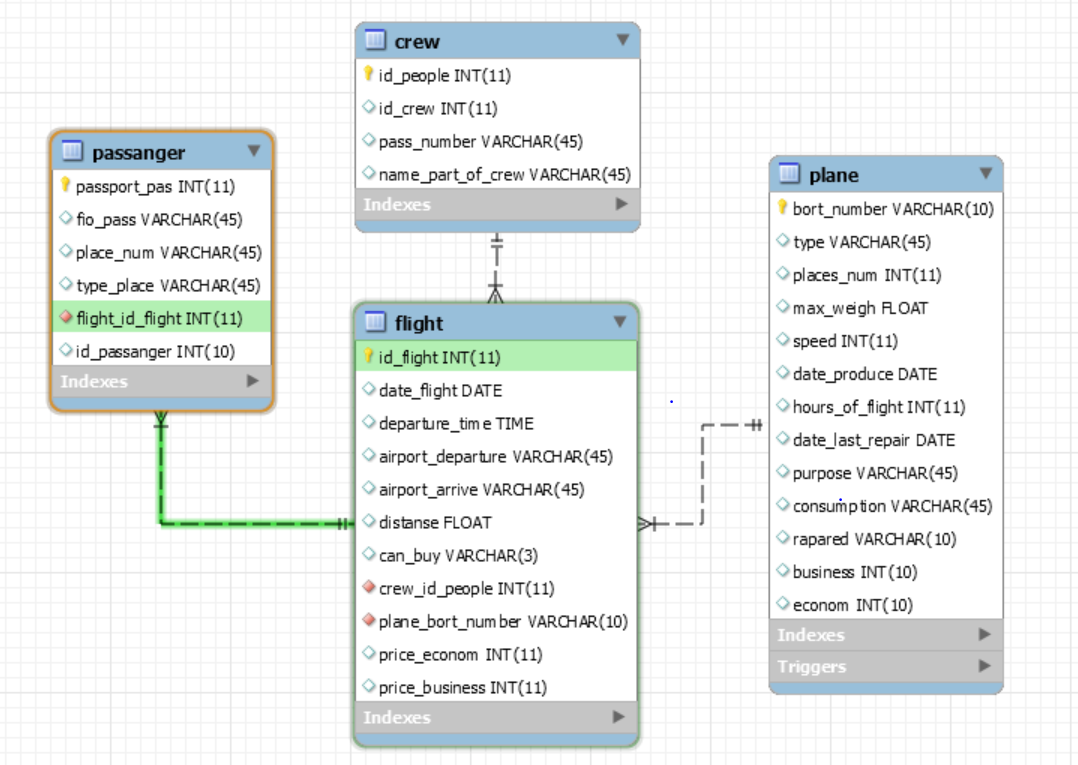


Рисунок 1 – UML диаграмма к базе данных

1. Описание структуры таблиц базы данных, с указанием имени, семантики и типа каждого поля

В таблице 1 представлены данные о полях таблиц базы данных.

Таблица 1 – Данные о полях таблицы «экипаж»

|  |  |  |
| --- | --- | --- |
| crew |  |  |
| id\_people | int(11) | Код человека |
| id\_crew | int(11) | Код экипажа |
| pass\_number | varchar(45) | Номер паспорта |
| name\_part\_of\_crew | varchar(45) | Имя члена экипажа |

Таблица 2 – Данные о полях таблицы «самолет»

|  |  |  |
| --- | --- | --- |
| plane |  |  |
| bort\_number | varchar(10) | Бортовой номер |
| type | varchar(45) | Тип |
| places\_num | int(11) | Общее количество мест |
| max\_weight | float | Максимальный вес |
| speed | int(11) | Скорость |
| date\_produce | date | Дата производства |
| hours\_of\_flight | int(11) | Часы налета |
| date\_last\_repair | date | Дата последнего ремонта |
| purpose | varchar(45) | Назначение |
| consumption | varchar(45) | Расход топлива |
| repared | varchar(10) | Был ли проведен ремонт |
| business | int(11) | Количество мест бизнес-класса |
| econom | int(11) | Количество мест эконом-класса |

Таблица 3 – Данные о полях таблицы «полет»

|  |  |  |
| --- | --- | --- |
| flight |  |  |
| id\_flight | int(11) | Код полета |
| date\_flight | date | Дата полета |
| departure\_time | time | Время отправления |
| airport\_departure | varchar(45) | Аэропорт отправления |
| airport\_arrive | varchar(45) | Аэропорт прибытия |
| distanse | float | Дистанция |
| can\_buy | varchar(3) | Можно ли забронировать билет |
| price\_econom | int(11) | Цена билета эконом-класса |
| price\_business | int(11) | Цена билета бизнес-класса |
| crew\_id\_people | int(11) | Номер экипажа |
| plane\_bort\_number | varchar(10) | Бортовой номер самолета |

Таблица 4 – Данные о полях таблицы «пассажир»

|  |  |  |
| --- | --- | --- |
| passanger |  |  |
| id\_passanger | int(11) | Код пассажира |
| passport\_pass | int(11) | Номер паспорта пассажира |
| fio\_pass | varchar(45) | ФИО пассажира |
| place\_num | int(11) | Количество мест |
| type\_place | varchar(45) | Тип места |
| flight\_id\_flight | int(11) | Код полета |

1. Скрипты создания таблиц

На листинге 1 представлен скрипт создания таблиц к базе данных.

Листинг 1 – Скрипт создания таблиц к базе данных

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema mydb

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema mydb

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `itog` DEFAULT CHARACTER SET utf8 ;

USE `itog` ;

-- -----------------------------------------------------

-- Table `mydb`.`crew`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `itog`.`crew` (

`id\_people` INT(11) NOT NULL,

`id\_crew` INT(11) NULL,

`pass\_number` VARCHAR(45) NULL DEFAULT NULL,

`name\_part\_of\_crew` VARCHAR(45) NULL DEFAULT NULL,

PRIMARY KEY (`id\_people`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `mydb`.`plane`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `itog`.`plane` (

`bort\_number` VARCHAR(10) NOT NULL,

`type` VARCHAR(45) NULL DEFAULT NULL,

`places\_num` INT(11) NULL DEFAULT NULL,

`max\_weigh` FLOAT NULL DEFAULT NULL,

`speed` INT(11) NULL DEFAULT NULL,

`date\_produce` DATE NULL DEFAULT NULL,

`hours\_of\_flight` INT(11) NULL DEFAULT NULL,

`date\_last\_repair` DATE NULL DEFAULT NULL,

`purpose` VARCHAR(45) NULL DEFAULT NULL,

`consumption` VARCHAR(45) NULL DEFAULT NULL,

`rapared` VARCHAR(10) NULL DEFAULT NULL,

PRIMARY KEY (`bort\_number`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `mydb`.`flight`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `itog`.`flight` (

`id\_flight` INT(11) NOT NULL,

`date\_flight` DATE NULL DEFAULT NULL,

`departure\_time` TIME NULL DEFAULT NULL,

`airport\_departure` VARCHAR(45) NULL DEFAULT NULL,

`airport\_arrive` VARCHAR(45) NULL DEFAULT NULL,

`distanse` FLOAT NULL DEFAULT NULL,

`can\_buy` VARCHAR(3) NULL DEFAULT NULL,

`crew\_id\_people` INT(11) NOT NULL,

`plane\_bort\_number` VARCHAR(10) NOT NULL,

PRIMARY KEY (`id\_flight`),

CONSTRAINT `fk\_flight\_plane1`

FOREIGN KEY (`plane\_bort\_number`)

REFERENCES `itog`.`plane` (`bort\_number`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_flight\_crew1`

FOREIGN KEY (`crew\_id\_people`)

REFERENCES `itog`.`crew` (`id\_people`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `itog`.`passanger`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `itog`.`passanger` (

`passport\_pas` INT(11) NOT NULL,

`fio\_pass` VARCHAR(45) NULL DEFAULT NULL,

`place\_num` VARCHAR(45) NULL DEFAULT NULL,

`type\_place` VARCHAR(45) NULL DEFAULT NULL,

`price\_ticket` int(10) NULL DEFAULT NULL,

`flight\_id\_flight` INT(11) NOT NULL,

PRIMARY KEY (`passport\_pas`),

CONSTRAINT `fk\_passanger\_flight`

FOREIGN KEY (`flight\_id\_flight`)

REFERENCES `itog`.`flight` (`id\_flight`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

alter table itog.plane add column business int(10);

alter table itog.plane add column econom int(10);

alter table itog.flight add column price\_econom int;

alter table itog.flight add column price\_business int;

alter table itog.passanger drop column ticket\_price;

alter table itog.passanger add column id\_passanger INT (10);

1. Команды заполнения таблицы

На листинге 2 представлены команды заполнения таблиц баз данных.

Листинг 2 – Команды заполнения таблиц

INSERT INTO `itog`.`crew` (`id\_people`, `id\_crew`, `pass\_number`, `name\_part\_of\_crew`) VALUES ('1', '1', '0000000001', 'Иванов Иван Иванович');

INSERT INTO `itog`.`crew` (`id\_people`, `id\_crew`, `pass\_number`, `name\_part\_of\_crew`) VALUES ('2', '1', '0000000002', 'Иванов Иван Анатольевич');

INSERT INTO `itog`.`crew` (`id\_people`, `id\_crew`, `pass\_number`, `name\_part\_of\_crew`) VALUES ('3', '1', '0000000003', 'Иванов Иван Владиславович');

INSERT INTO `itog`.`crew` (`id\_people`, `id\_crew`, `pass\_number`, `name\_part\_of\_crew`) VALUES ('4', '1', '0000000004', 'Иванов Иван Владимирович');

INSERT INTO `itog`.`crew` (`id\_people`, `id\_crew`, `pass\_number`, `name\_part\_of\_crew`) VALUES ('5', '2', '0000000005', 'Иванов Владимир Иванович');

INSERT INTO `itog`.`crew` (`id\_people`, `id\_crew`, `pass\_number`, `name\_part\_of\_crew`) VALUES ('6', '2', '0000000006', 'Иванов Владимир Анатольевич');

INSERT INTO `itog`.`crew` (`id\_people`, `id\_crew`, `pass\_number`, `name\_part\_of\_crew`) VALUES ('7', '2', '0000000007', 'Иванов Владимир Владиславович');

INSERT INTO `itog`.`crew` (`id\_people`, `id\_crew`, `pass\_number`, `name\_part\_of\_crew`) VALUES ('8', '2', '0000000008', 'Иванов Владимир Владимирович');

INSERT INTO `itog`.`crew` (`id\_people`, `id\_crew`, `pass\_number`, `name\_part\_of\_crew`) VALUES ('9', '3', '0000000009', 'Солнцев Иван Иванович');

INSERT INTO `itog`.`crew` (`id\_people`, `id\_crew`, `pass\_number`, `name\_part\_of\_crew`) VALUES ('10', '3', '0000000010', 'Солнцев Иван Анатольевич');

INSERT INTO `itog`.`crew` (`id\_people`, `id\_crew`, `pass\_number`, `name\_part\_of\_crew`) VALUES ('11', '3', '0000000011', 'Солнцев Иван Владиславович');

INSERT INTO `itog`.`crew` (`id\_people`, `id\_crew`, `pass\_number`, `name\_part\_of\_crew`) VALUES ('12', '3', '0000000012', 'Солнцев Иван Владимирович');

INSERT INTO `itog`.`plane` (`bort\_number`, `type`, `places\_num`, `max\_weigh`, `speed`, `date\_produce`, `hours\_of\_flight`, `date\_last\_repair`, `purpose`, `consumption`) VALUES ('M-ANSL', 'Airbus', '10', '45000', '917', '2008-09-21', '11000', '2015-09-19', 'Far away trips', '2000');

INSERT INTO `itog`.`plane` (`bort\_number`, `type`, `places\_num`, `max\_weigh`, `speed`, `date\_produce`, `hours\_of\_flight`, `date\_last\_repair`, `purpose`, `consumption`) VALUES ('HB-JEG', 'Airbus', '15', '50000', '850', '2016-09-25', '4000', '2016-09-19', 'Short trips', '1500');

UPDATE `itog`.`plane` SET `business` = '5', `econom` = '10' WHERE (`bort\_number` = 'HB-JEG');

UPDATE `itog`.`plane` SET `business` = '3', `econom` = '7' WHERE (`bort\_number` = 'M-ANSL');

INSERT INTO `itog`.`flight` (`id\_flight`, `date\_flight`, `departure\_time`, `airport\_departure`, `airport\_arrive`, `distanse`, `plane\_bort\_number`, `can\_buy`, `crew\_id\_people`) VALUES ('1', '2019-05-29', '15:30:00', 'Pulkovo', 'Sheremetevo', '800', 'HB-JEG', 'yes', '1');

INSERT INTO `itog`.`flight` (`id\_flight`, `date\_flight`, `departure\_time`, `airport\_departure`, `airport\_arrive`, `distanse`, `plane\_bort\_number`, `can\_buy`, `crew\_id\_people`) VALUES ('2', '2019-05-29', '16:00:00', 'Pulkovo', 'Monako', '2000', 'HB-JEG', 'yes', '1');

INSERT INTO `itog`.`flight` (`id\_flight`, `date\_flight`, `departure\_time`, `airport\_departure`, `airport\_arrive`, `distanse`, `plane\_bort\_number`, `can\_buy`, `crew\_id\_people`) VALUES ('3', '2019-05-29', '17:00:00', 'Sheremetevo', 'Pulkovo', '800', 'HB-JEG', 'yes', '2');

INSERT INTO `itog`.`flight` (`id\_flight`, `date\_flight`, `departure\_time`, `airport\_departure`, `airport\_arrive`, `distanse`, `plane\_bort\_number`, `can\_buy`, `crew\_id\_people`) VALUES ('4', '2019-07-30', '18:30:00', 'Pulkovo', 'Volgograd', '1700', 'M-ANSL', 'yes', '3');

INSERT INTO `itog`.`flight` (`id\_flight`, `date\_flight`, `departure\_time`, `airport\_departure`, `airport\_arrive`, `distanse`, `plane\_bort\_number`, `can\_buy`, `crew\_id\_people`) VALUES ('5', '2019-06-05', '19:30:00', 'Domodedovo', 'Pulkovo', '800', 'HB-JEG', 'yes', '3');

INSERT INTO `itog`.`flight` (`id\_flight`, `date\_flight`, `departure\_time`, `airport\_departure`, `airport\_arrive`, `distanse`, `can\_buy`, `crew\_id\_people`, `plane\_bort\_number`, `price\_econom`, `price\_business`) VALUES ('6', '2019-05-30', '19:00:00', 'Pulkovo', 'Domodedovo', '800', 'no', '3', 'HB-JEG', '10000', '30000');

INSERT INTO `itog`.`flight` (`id\_flight`, `date\_flight`, `departure\_time`, `airport\_departure`, `airport\_arrive`, `distanse`, `can\_buy`, `crew\_id\_people`, `plane\_bort\_number`, `price\_econom`, `price\_business`) VALUES ('7', '2019-05-30', '19:00:00', 'Pulkovo', 'Domodedovo', '800', 'no', '2', 'M-ANSL', '10000', '30000');

UPDATE `itog`.`flight` SET `price\_econom` = '10000', `price\_business` = '30000' WHERE (`id\_flight` = '1');

UPDATE `itog`.`flight` SET `price\_econom` = '40000', `price\_business` = '60000' WHERE (`id\_flight` = '2');

UPDATE `itog`.`flight` SET `price\_econom` = '10000', `price\_business` = '30000' WHERE (`id\_flight` = '3');

UPDATE `itog`.`flight` SET `price\_econom` = '30000', `price\_business` = '50000' WHERE (`id\_flight` = '4');

UPDATE `itog`.`flight` SET `price\_econom` = '10000', `price\_business` = '30000' WHERE (`id\_flight` = '5');

INSERT INTO `itog`.`passanger` (`passport\_pas`, `fio\_pass`, `place\_num`, `type\_place`, `price\_ticket`, `flight\_id\_flight`) VALUES ('1000000000', 'Муратов Мурат Муратович', '1', 'Business', '30000', '2');

INSERT INTO `itog`.`passanger` (`passport\_pas`, `fio\_pass`, `place\_num`, `type\_place`, `price\_ticket`, `flight\_id\_flight`) VALUES ('1000000001', 'Муратов Мурат Владимирович', '2', 'Econom', '20000', '2');

INSERT INTO `itog`.`passanger` (`passport\_pas`, `fio\_pass`, `place\_num`, `type\_place`, `price\_ticket`, `flight\_id\_flight`) VALUES ('1000000002', 'Муратов Мурат Владиславович', '3', 'Business', '30000', '2');

INSERT INTO `itog`.`passanger` (`passport\_pas`, `fio\_pass`, `place\_num`, `type\_place`, `price\_ticket`, `flight\_id\_flight`) VALUES ('1000000003', 'Муратов Мурат Анатольевич', '4', 'Econom', '20000', '2');

INSERT INTO `itog`.`passanger` (`passport\_pas`, `fio\_pass`, `place\_num`, `type\_place`, `price\_ticket`, `flight\_id\_flight`) VALUES ('1000000004', 'Муратов Аскар Муратович', '5', 'Business', '30000', '2');

INSERT INTO `itog`.`passanger` (`passport\_pas`, `fio\_pass`, `place\_num`, `type\_place`, `price\_ticket`, `flight\_id\_flight`) VALUES ('1000000005', 'Муратов Аскар Владимирович', '1', 'Econom', '10000', '3');

INSERT INTO `itog`.`passanger` (`passport\_pas`, `fio\_pass`, `place\_num`, `type\_place`, `price\_ticket`, `flight\_id\_flight`) VALUES ('1000000006', 'Муратов Аскар Владиславович', '2', 'Business', '20000', '3');

INSERT INTO `itog`.`passanger` (`passport\_pas`, `fio\_pass`, `place\_num`, `type\_place`, `price\_ticket`, `flight\_id\_flight`) VALUES ('1000000007', 'Муратов Аскар Анатольевич', '3', 'Econom', '10000', '3');

INSERT INTO `itog`.`passanger` (`passport\_pas`, `fio\_pass`, `place\_num`, `type\_place`, `price\_ticket`, `flight\_id\_flight`) VALUES ('1000000008', 'Муратов Лерой Муратович', '4', 'Business', '20000', '3');

INSERT INTO `itog`.`passanger` (`passport\_pas`, `fio\_pass`, `place\_num`, `type\_place`, `price\_ticket`, `flight\_id\_flight`) VALUES ('1000000009', 'Муратов Лерой Владимирович', '5', 'Econom', '10000', '3');

INSERT INTO `itog`.`passanger` (`passport\_pas`, `fio\_pass`, `place\_num`, `type\_place`, `price\_ticket`, `flight\_id\_flight`) VALUES ('1000000010', 'Муратов Лерой Владиславович', '6', 'Business', '20000', '3');

UPDATE `itog`.`passanger` SET `id\_passanger` = '1' WHERE (`passport\_pas` = '1000000000');

UPDATE `itog`.`passanger` SET `id\_passanger` = '2' WHERE (`passport\_pas` = '1000000001');

UPDATE `itog`.`passanger` SET `id\_passanger` = '3' WHERE (`passport\_pas` = '1000000002');

UPDATE `itog`.`passanger` SET `id\_passanger` = '4' WHERE (`passport\_pas` = '1000000003');

UPDATE `itog`.`passanger` SET `id\_passanger` = '5' WHERE (`passport\_pas` = '1000000004');

UPDATE `itog`.`passanger` SET `id\_passanger` = '6' WHERE (`passport\_pas` = '1000000005');

UPDATE `itog`.`passanger` SET `id\_passanger` = '7' WHERE (`passport\_pas` = '1000000006');

UPDATE `itog`.`passanger` SET `id\_passanger` = '8' WHERE (`passport\_pas` = '1000000007');

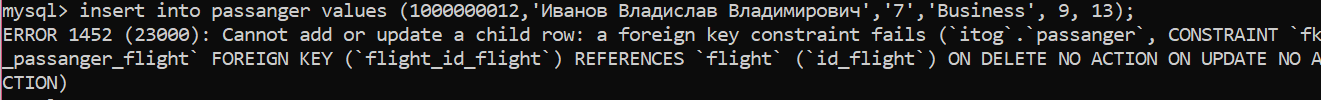
UPDATE `itog`.`passanger` SET `id\_passanger` = '9' WHERE (`passport\_pas` = '1000000008');

UPDATE `itog`.`passanger` SET `id\_passanger` = '10' WHERE (`passport\_pas` = '1000000009');

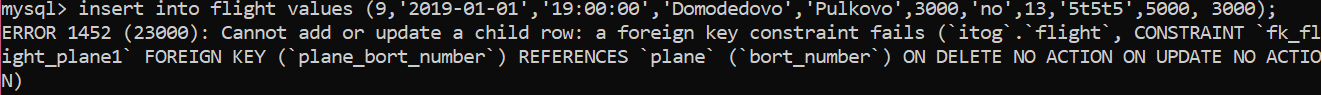
UPDATE `itog`.`passanger` SET `id\_passanger` = '11' WHERE (`passport\_pas` = '1000000010');

1. **Скриншоты проверки внешних ключей**

**Скриншоты проверки внешних ключей представлены на рисунках 1-.**



**Рисунок 1 – Скриншот проверки внешнего ключа таблицы «passanger»**



**Рисунок 2 – Скриншот проверки внешних ключей таблицы «crew»**

1. Определение объема занимаемой памяти

Объем занимаемой памяти отображен на рисунке 3.

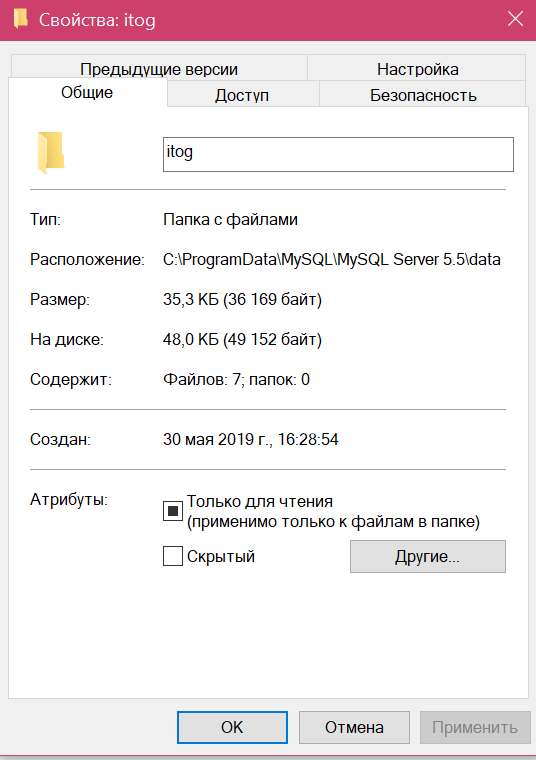


Рисунок 3 – Скриншот свойств папки с базой данных

1. Вывод

В процессе выполнения лабораторной работы были приобретены практические навыки создания базы данных, таблиц к ней. Изучены способы модификации данных, в том числе удаления, создания и редактирования записей.