**Контрольная работа по курсу МДиСУБД**

**Елисеева Данилы Ивановича 4го курса 6 группы  
8 Вариант**

1. Создать базу данных «Учет успеваемости». Схема БД и состав полей таблиц

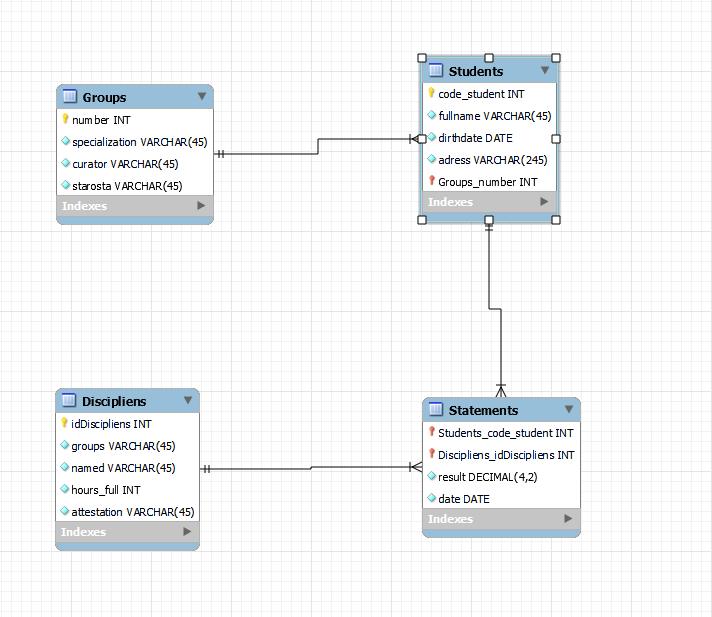


Схема базы данных

-- 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 students\_statement

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

DROP SCHEMA IF EXISTS `students\_statement` ;

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

-- Schema students\_statement

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

CREATE SCHEMA IF NOT EXISTS `students\_statement` DEFAULT CHARACTER SET utf8 ;

USE `students\_statement` ;

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

-- Table `students\_statement`.`Groups`

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

CREATE TABLE IF NOT EXISTS `students\_statement`.`Groups` (

`number` INT NOT NULL,

`specialization` VARCHAR(45) NOT NULL,

`curator` VARCHAR(45) NOT NULL,

`starosta` VARCHAR(45) NOT NULL,

PRIMARY KEY (`number`))

ENGINE = InnoDB;

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

-- Table `students\_statement`.`Students`

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

CREATE TABLE IF NOT EXISTS `students\_statement`.`Students` (

`code\_student` INT NOT NULL,

`fullname` VARCHAR(45) NOT NULL,

`dirthdate` DATE NOT NULL,

`adress` VARCHAR(245) NOT NULL,

`Groups\_number` INT NOT NULL,

PRIMARY KEY (`code\_student`, `Groups\_number`),

INDEX `fk\_Students\_Groups\_idx` (`Groups\_number` ASC) VISIBLE,

CONSTRAINT `fk\_Students\_Groups`

FOREIGN KEY (`Groups\_number`)

REFERENCES `students\_statement`.`Groups` (`number`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

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

-- Table `students\_statement`.`Discipliens`

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

CREATE TABLE IF NOT EXISTS `students\_statement`.`Discipliens` (

`idDiscipliens` INT NOT NULL,

`groups` VARCHAR(45) NOT NULL,

`named` VARCHAR(45) NOT NULL,

`hours\_full` INT NOT NULL,

`attestation` VARCHAR(45) NOT NULL,

PRIMARY KEY (`idDiscipliens`))

ENGINE = InnoDB;

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

-- Table `students\_statement`.`Statements`

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

CREATE TABLE IF NOT EXISTS `students\_statement`.`Statements` (

`Students\_code\_student` INT NOT NULL,

`Discipliens\_idDiscipliens` INT NOT NULL,

`result` DECIMAL(4,2) NOT NULL,

`date` DATE NOT NULL,

PRIMARY KEY (`Students\_code\_student`, `Discipliens\_idDiscipliens`),

INDEX `fk\_Statements\_Students1\_idx` (`Students\_code\_student` ASC) VISIBLE,

CONSTRAINT `fk\_Statements\_Discipliens1`

FOREIGN KEY (`Discipliens\_idDiscipliens`)

REFERENCES `students\_statement`.`Discipliens` (`idDiscipliens`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `fk\_Statements\_Students1`

FOREIGN KEY (`Students\_code\_student`)

REFERENCES `students\_statement`.`Students` (`code\_student`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

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

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

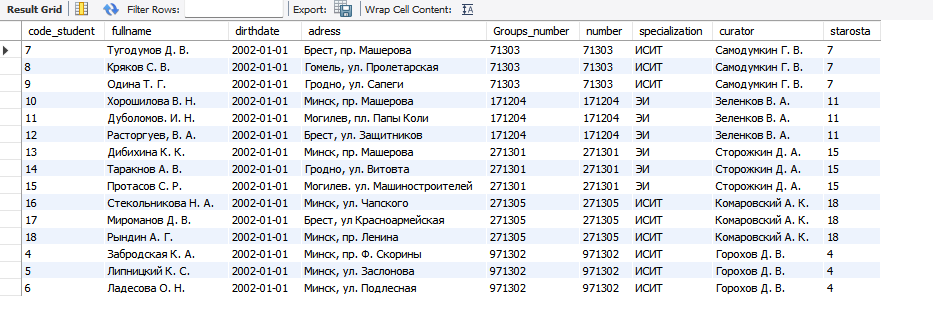
SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

1. Создать запросы на выборку для решения следующих задач  
   1. Получить сведения о студентах групп специализации ИСИТ и ЭИ

select \* from students p

join `groups` g on p.Groups\_number = g.number

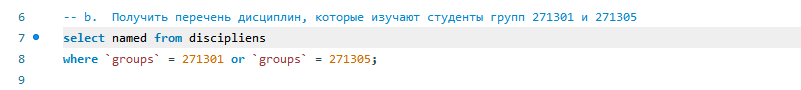
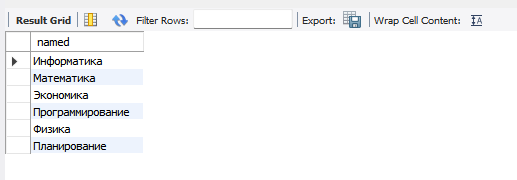
where specialization = "ИСИТ" or specialization = "ЭИ";



* 1. Получить перечень дисциплин, которые изучают студенты групп 271301 и 271305

select named from discipliens

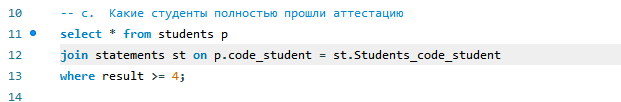
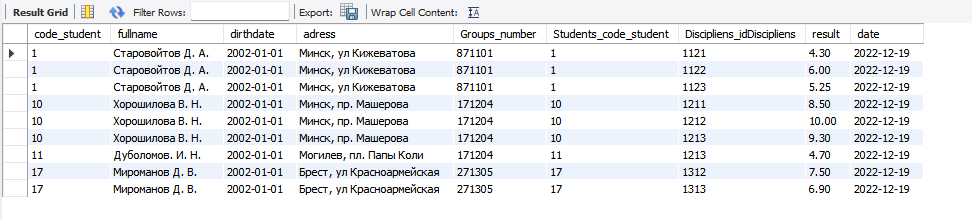
where `groups` = 271301 or `groups` = 271305;

* 1. Какие студенты полностью прошли аттестацию

select \* from students p

join statements st on p.code\_student = st.Students\_code\_student

where result >= 4;  
  
 

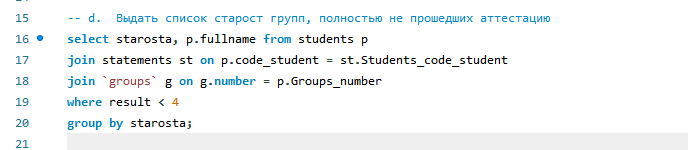
* 1. Выдать список старост групп, полностью не прошедших аттестацию

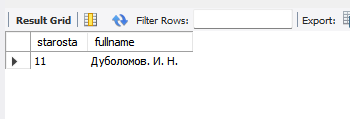
select starosta, p.fullname from students p

join statements st on p.code\_student = st.Students\_code\_student

join `groups` g on g.number = p.Groups\_number

where result < 4

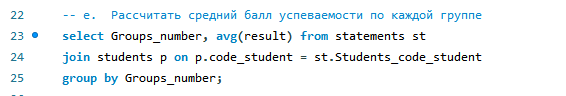
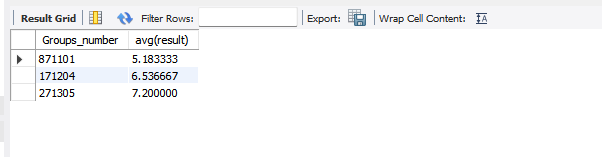
group by starosta;  
  




* 1. Рассчитать средний балл успеваемости по каждой группе

select Groups\_number, avg(result) from statements st

join students p on p.code\_student = st.Students\_code\_student

group by Groups\_number;  
 

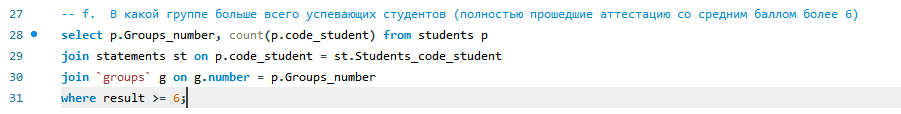
* 1. В какой группе больше всего успевающих студентов (полностью прошедшие аттестацию со средним баллом более 6)

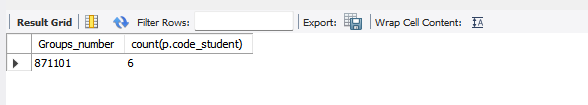
select p.Groups\_number, count(p.code\_student) from students p

join statements st on p.code\_student = st.Students\_code\_student

join `groups` g on g.number = p.Groups\_number

where result >= 6;





* 1. \* Номер группы состоит из 6 цифр: 1 – год поступления, 2 – номер факультета, 3 и 4 – номер специальности, 5 и 6 – порядковый номер группы на потоке. Обозначения специальностей: 11 – «Экономика», 12 – «Маркетинг», 13 – «Информационные системы».

select st.fullname, d.named as предметы, g.specialization as специальность, GROUP\_CONCAT(result SEPARATOR ',') as оценки,

round(avg(result), 2) as средний\_балл from students st

join discipliens d

on d.`groups` = st.Groups\_number

join statements s

on s.Students\_code\_student = st.code\_student

join `groups` g

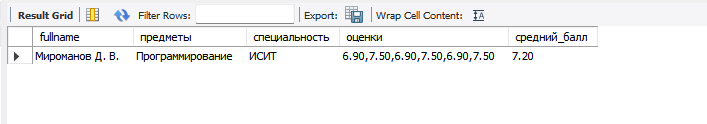
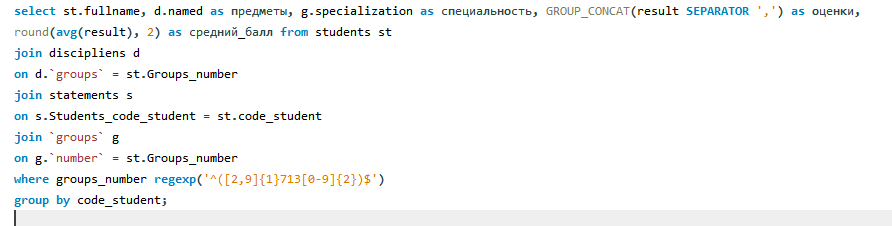
on g.`number` = st.Groups\_number

where groups\_number regexp('^([2,9]{1}713[0-9]{2})$')

group by code\_student;

select \* from students

where groups\_number regexp('^([2,9]{1}713[0-9]{2})$');



**Задача**. Составить рейтинг студентов специальности «Информационные системы» 1 и 3 курса. В рейтинговой таблице вывести фамилию студента, предметы, по которым он аттестован, оценки и средний балл успеваемости.

Создать триггер для предотвращения удаления сведений о студентах, не завершивших сессию. (сессия продолжается с 5 по 20 января и с 5 по 20 июня)

delimiter //

CREATE TRIGGER delete\_students BEFORE DELETE ON students FOR EACH ROW

begin

declare id\_stud varchar(45);

(select code\_student from student

where code\_student in (select distinct stud from statements st1

left join student st on st.code\_student = st1.Students\_code\_student

where 4 > any

(select Students\_code\_student, d.named, d.attestation, result from statements st2

join discipliens d on d.idDiscipliens = statements.Discipliens\_idDiscipliens

where ((MONTH(st2.date) = 12 and DAYOFMONTH(st2.date) between 5 and 20)

or (MONTH(st2.date) = 6 and DAYOFMONTH(st2.date) between 5 and 20))))) into id\_stud;

end;

//

