

Практические задания

// Создаем БД

```
CREATE DATABASE `students`  
  CHARACTER SET 'utf8'  
  COLLATE 'utf8_general_ci';
```

// Студент

```
CREATE TABLE `student` (  
  `id` int(11) unsigned NOT NULL,  
  `first_name` varchar(20) NOT NULL,  
  `last_name` varchar(20) NOT NULL,  
  `birth_date` date NOT NULL,  
  `sex` bit(1) NOT NULL,  
  `hostel_live` bit(1) DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `id` (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

// Предмет

```
CREATE TABLE `training_course` (  
  `id` INTEGER(11) UNSIGNED NOT NULL AUTO_INCREMENT UNIQUE,  
  `name` VARCHAR(50) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB;
```

// Преподаватель

```
CREATE TABLE `teacher` (  
  `id` INTEGER(11) UNSIGNED NOT NULL AUTO_INCREMENT UNIQUE,  
  `first_name` VARCHAR(20) NOT NULL,  
  `last_name` VARCHAR(20) NOT NULL UNIQUE,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB;
```

// Исправляем ошибки в таблице student

```
ALTER TABLE `students`.`student` CHANGE COLUMN `id` `id` INT(11) UNSIGNED NOT NULL AUTO_INCREMENT ;  
ALTER TABLE `students`.`training_course` ADD COLUMN `teacher_id` INT(11) UNSIGNED NOT NULL AFTER `name` ,  
ADD CONSTRAINT `teacher_fk`  
FOREIGN KEY (`teacher_id` )  
REFERENCES `students`.`teacher` (`id` )  
ON DELETE CASCADE  
ON UPDATE RESTRICT  
, ADD INDEX `teacher_fk_idx` (`teacher_id` ASC) ;
```

Практические задания. Структура БД

```
CREATE TABLE `students`.`exam` (  
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT ,  
  `date` TIMESTAMP NOT NULL ,  
  PRIMARY KEY (`id`) ,  
  UNIQUE INDEX `id_UNIQUE` (`id` ASC) );
```

```
ALTER TABLE `students`.`exam` ADD COLUMN `teacher_id` INT(11) UNSIGNED NOT NULL,  
  ADD CONSTRAINT `exam_teacher_fk`  
  FOREIGN KEY (`teacher_id` )  
  REFERENCES `students`.`teacher` (`id` )  
  ON DELETE RESTRICT  
  ON UPDATE RESTRICT  
, ADD INDEX `exam_teacher_fk_idx` (`teacher_id` ASC) ;
```

```
ALTER TABLE `students`.`exam` ADD COLUMN `training_course_id` INT(11) UNSIGNED NOT NULL,  
  ADD CONSTRAINT `exam_training_course_fk`  
  FOREIGN KEY (`training_course_id` )  
  REFERENCES `students`.`training_course` (`id` )  
  ON DELETE RESTRICT  
  ON UPDATE RESTRICT  
, ADD INDEX `exam_training_course_fk_idx` (`training_course_id` ASC) ;
```

```
CREATE TABLE `students`.`exam_result` (  
  `id` INT(11) UNSIGNED NOT NULL AUTO_INCREMENT ,  
  `teacher_id` INT(11) UNSIGNED NOT NULL ,  
  `student_id` INT(11) UNSIGNED NOT NULL ,  
  `result` TINYINT NOT NULL ,  
  `note` VARCHAR(50) NULL ,  
  PRIMARY KEY (`id`) ,  
  INDEX `exam_result_teacher_fk_idx` (`teacher_id` ASC) ,  
  INDEX `exam_result_student_fk_idx` (`student_id` ASC) ,  
  CONSTRAINT `exam_result_teacher_fk`  
  FOREIGN KEY (`teacher_id` )  
  REFERENCES `students`.`teacher` (`id` )  
  ON DELETE RESTRICT  
  ON UPDATE RESTRICT,  
  CONSTRAINT `exam_result_student_fk`  
  FOREIGN KEY (`student_id` )  
  REFERENCES `students`.`student` (`id` )  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION);
```

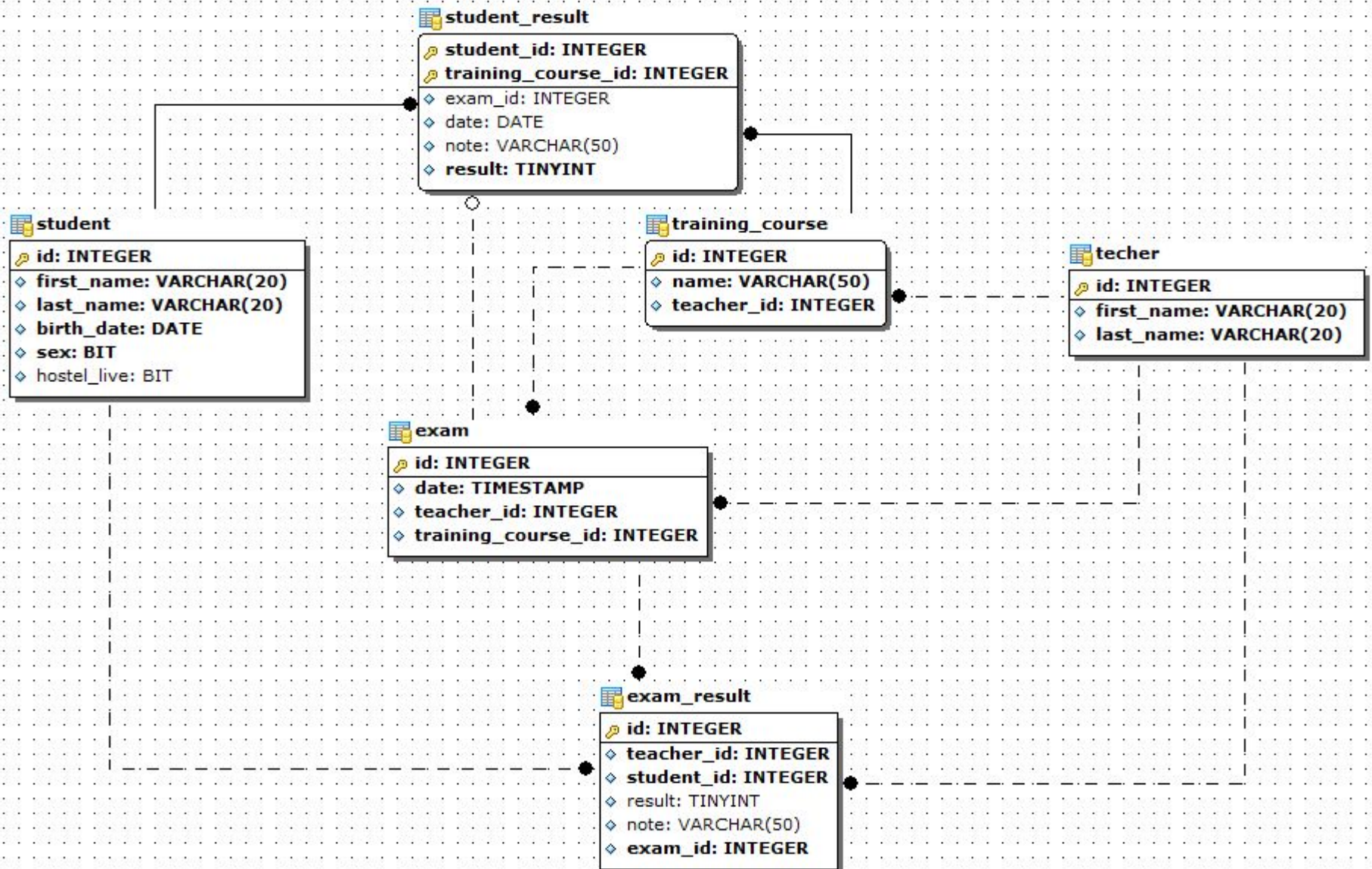
Практические задания. Структура БД

```
CREATE TABLE `students`.`student_result` (  
  `student_id` INT(11) UNSIGNED NOT NULL ,  
  `training_course_id` INT(11) UNSIGNED NOT NULL ,  
  `exam_id` INT(11) UNSIGNED ,  
  `result` TINYINT NOT NULL ,  
  `date` DATE NOT NULL ,  
  `note` VARCHAR(50) NULL ,  
  UNIQUE INDEX `student_result_unique` (`student_id` ASC) ,  
  PRIMARY KEY (`student_id`, `training_course_id`) ,  
  INDEX `student_result__idx` (`training_course_id` ASC) ,  
  INDEX `student_result_student_id_idx` (`student_id` ASC) ,  
  CONSTRAINT `student_result_training_course`  
    FOREIGN KEY (`training_course_id` )  
      REFERENCES `students`.`training_course` (`id` )  
    ON DELETE RESTRICT  
    ON UPDATE RESTRICT,  
  CONSTRAINT `student_result_student_id`  
    FOREIGN KEY (`student_id` )  
      REFERENCES `students`.`student` (`id` )  
    ON DELETE RESTRICT  
    ON UPDATE RESTRICT);
```

```
ALTER TABLE `students`.`exam_result` ADD COLUMN `exam_id` INT(11) UNSIGNED NOT NULL AFTER `note` ,  
  ADD CONSTRAINT `exam_result_exam_fk`  
    FOREIGN KEY (`exam_id` )  
      REFERENCES `students`.`exam` (`id` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION  
, ADD INDEX `exam_result_exam_fk_idx` (`exam_id` ASC);
```

```
ALTER TABLE `students`.`student_result`  
  ADD CONSTRAINT `student_result_exam_fk`  
    FOREIGN KEY (`exam_id` )  
      REFERENCES `students`.`exam` (`id` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION  
, ADD INDEX `student_result_exam_fk_idx` (`exam_id` ASC);
```

Практические задания. Структура БД



Практические задания.

1. Выбрать имена и фамилии студентов, успешно сдавших экзамен, упорядоченных по результату экзамена (отличники первые в результате)
2. Посчитать количество студентов, успешно сдавших экзамен на 4 и 5
3. Посчитать количество студентов, сдавших экзамен “автоматом” (нет записи в таблице exam_result но есть положительный результат в таблице student_result)
4. Посчитать средний балл студентов по предмету с наименованием “Системы управления базами данных”
5. Выбрать имена и фамилии студентов, не сдававших экзамен по предмету “Теория графов” (2 вида запроса)
6. Выбрать идентификатор преподавателей, читающих лекции по больше чем по 2 предметам
7. Выбрать идентификатор и фамилии студентов, пересдававших хотя бы 1 предмет
8. Вывести имена и фамилии 5 студентов с максимальными оценками
9. Вывести фамилию преподавателя, у которого наилучшие результаты по его предметам
10. Вывести успеваемость студентов по годам по предмету “Математическая статистика”