Министерство науки и высшего образования РФ

Севастопольский государственный университет

Кафедра информационных систем

ОТЧЕТ

по лабораторной работе №1

на тему «Манипулирование базой данных. Реляционная алгебра и SQL»

по дисциплине «Управление данными»

Выполнил:

Студент группы ИС/б 17-2-о

Черняев Н.Г.

Проверил:

Абрамович А.Ю.

г. Севастополь 2019

Цель работы

Изучить основы реляционной алгебры как базового средства манипулирования. Научиться представлять запросы как на реляционной алгебре, так и на SQL.

Ход работы

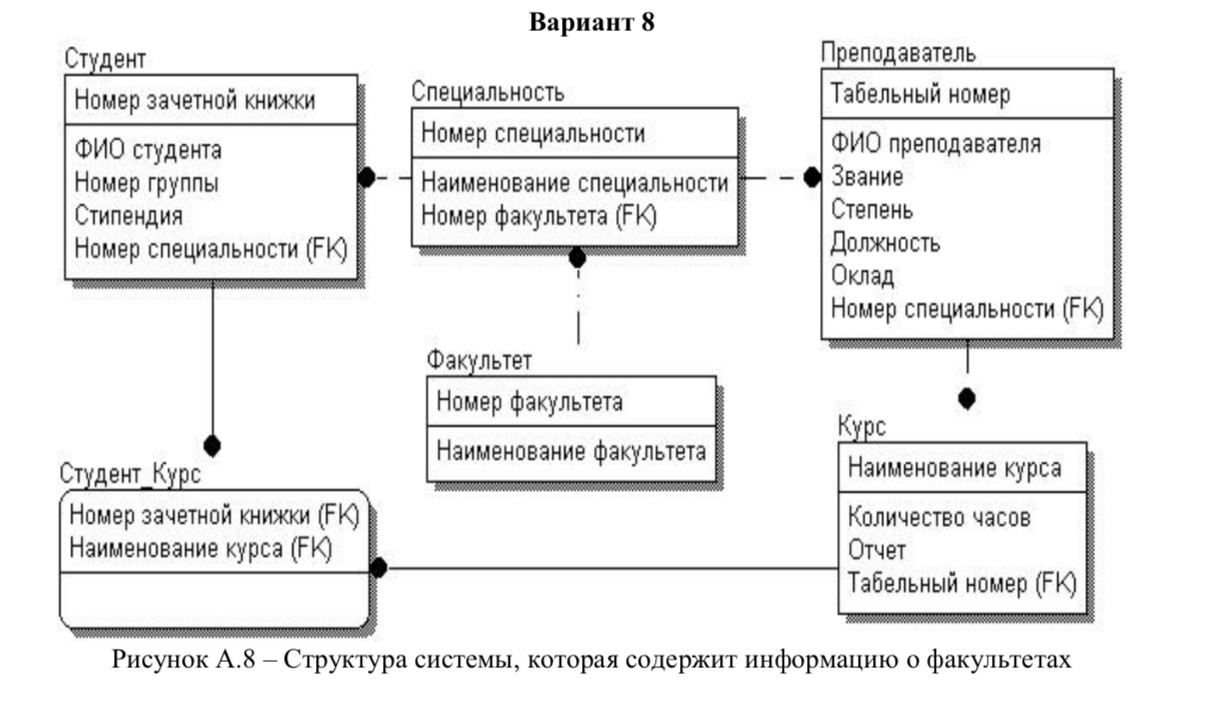


Рисунок 1 - вариант задания

SQL запросы создания таблиц

CREATE SCHEMA IF NOT EXISTS `LR-1` DEFAULT CHARACTER SET utf8 ;

USE `LR-1` ;

CREATE TABLE IF NOT EXISTS `LR-1`.`faculty` (

`id` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(45) NOT NULL,

PRIMARY KEY (`id`))

ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `LR-1`.`specialty` (

`noSpecialty` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(45) NULL,

`noFaculty` INT NULL,

PRIMARY KEY (`noSpecialty`),

INDEX `noFacul\_idx` (`noFaculty` ASC),

CONSTRAINT `noFacul`

FOREIGN KEY (`noFaculty`)

REFERENCES `LR-1`.`faculty` (`id`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `LR-1`.`student` (

`id` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(45) NULL,

`groupNo` INT NULL,

`grant` TINYINT NULL,

`noSpecialty` INT NULL,

PRIMARY KEY (`id`),

INDEX `noSpec\_idx` (`noSpecialty` ASC),

CONSTRAINT `noSpec`

FOREIGN KEY (`noSpecialty`)

REFERENCES `LR-1`.`specialty` (`noSpecialty`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `LR-1`.`educator` (

`tabNo` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(45) NULL,

`title` VARCHAR(45) NULL,

`degree` VARCHAR(45) NULL,

`position` VARCHAR(45) NULL,

`salary` INT NULL,

`noSpecialty` INT NULL,

PRIMARY KEY (`tabNo`),

INDEX `noSpec\_idx` (`noSpecialty` ASC),

CONSTRAINT `noSpeci`

FOREIGN KEY (`noSpecialty`)

REFERENCES `LR-1`.`specialty` (`noSpecialty`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `LR-1`.`stage` (

`id` INT NOT NULL AUTO\_INCREMENT,

`hours` INT NULL,

`report` VARCHAR(45) NULL,

`tabNo` INT NULL,

PRIMARY KEY (`id`),

INDEX `tabNo\_idx` (`tabNo` ASC),

CONSTRAINT `tabNo`

FOREIGN KEY (`tabNo`)

REFERENCES `LR-1`.`educator` (`tabNo`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `LR-1`.`student\_stage` (

`studentId` INT NOT NULL AUTO\_INCREMENT,

`stageId` INT NOT NULL,

PRIMARY KEY (`studentId`, `stageId`),

INDEX `stage\_idx` (`stageId` ASC),

CONSTRAINT `student`

FOREIGN KEY (`studentId`)

REFERENCES `LR-1`.`student` (`id`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `stage`

FOREIGN KEY (`stageId`)

REFERENCES `LR-1`.`stage` (`id`)

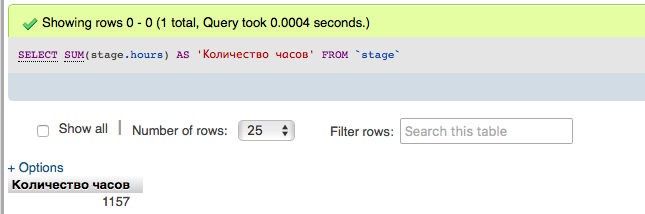
ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

**Продемонстрировать выполнение простых вычислений в запросе**

Посчитать общее количество часов по всем курсам:

[SELECT](http://localhost:8888/phpMyAdmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) [SUM](http://localhost:8888/phpMyAdmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_sum)(stage.hours) AS 'Количество часов' FROM `stage`

Использовать простое вычисление как параметр агрегатной функции

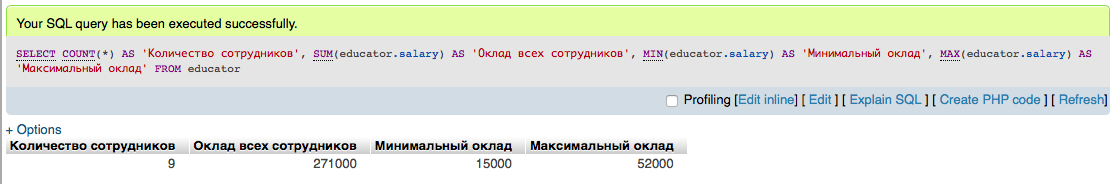
Вывод информации об окладе преподавателей

SELECT COUNT(\*) AS 'Количество сотрудников',

SUM(educator.salary) AS 'Оклад всех сотрудников',

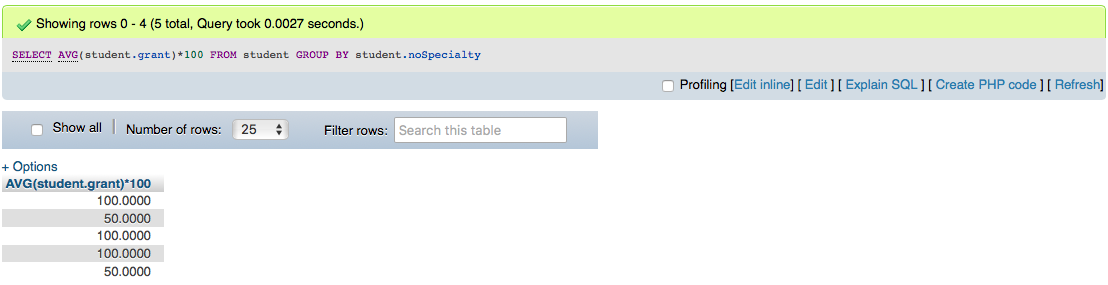
MIN(educator.salary) AS 'Минимальный оклад',

MAX(educator.salary) AS 'Максимальный оклад'

FROM educator

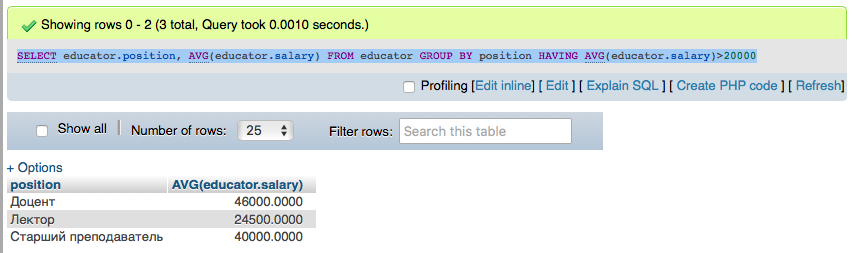
Продемонстрировать работу предложения GROUP BY

Вывести для каждой специальности процент студентов, получающих стипендию

[SELECT](http://localhost:8888/phpMyAdmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) [AVG](http://localhost:8888/phpMyAdmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_avg)(student.grant)\*100 FROM student GROUP BY student.noSpecialty

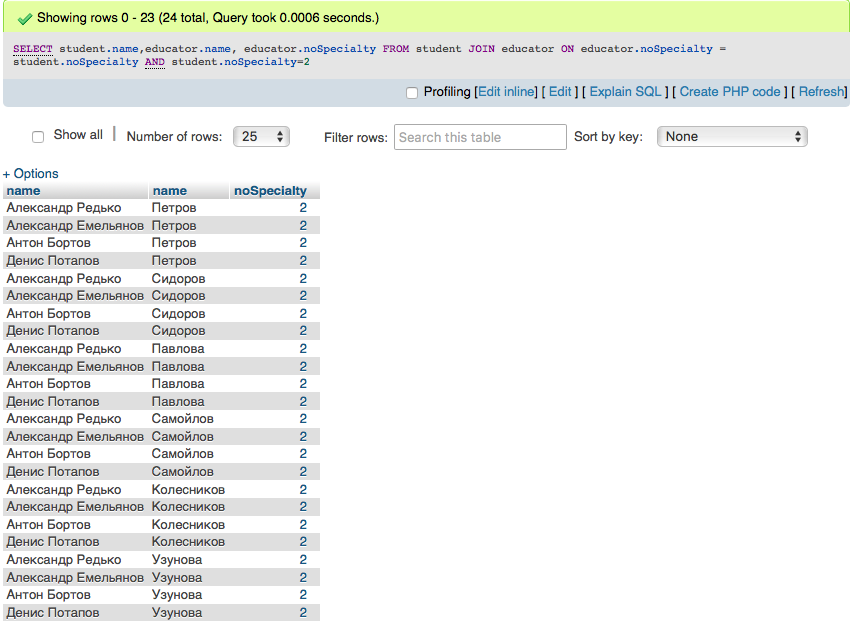
Продемонстрировать работу предложения HAVING

Вывод должностей, и среднего оклада для каждой, учитывая лишь оклад выше 20000.

[SELECT](http://localhost:8888/phpMyAdmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) educator.position, [AVG](http://localhost:8888/phpMyAdmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_avg)(educator.salary) FROM educator GROUP BY position HAVING [AVG](http://localhost:8888/phpMyAdmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_avg)(educator.salary)>20000

Применить к БД операции селекции и соединения в одном запросе

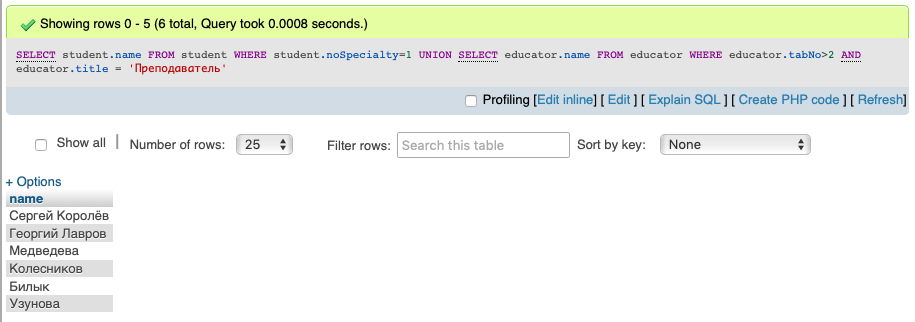
Вывод всех людей, причастных к специальности №2.

[SELECT](http://localhost:8888/phpMyAdmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) student.name, educator.name, educator.noSpecialty FROM student JOIN educator ON educator.noSpecialty = student.noSpecialty [AND](http://localhost:8888/phpMyAdmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and) student.noSpecialty=2

Создать запрос, использующий операции проекции, объединения и конъюнкции

Выбор имен студентов, занимающихся на специальности №1 и имен преподавателей, с табельным номером больше 2 и должностью «преподователь».

[SELECT](http://localhost:8888/phpMyAdmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) student.name FROM student WHERE student.noSpecialty=1 UNION [SELECT](http://localhost:8888/phpMyAdmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html)

educator.name FROM educator WHERE educator.tabNo>2 [AND](http://localhost:8888/phpMyAdmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and) educator.title = 'Преподаватель';

Вывод

В ходе выполнения лабораторной работы были изучены основы реляционной алгебры как базового средства манипулирования, а также получены навыки составления запросов как на реляционной алгебре, так и на SQL.