МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«Севастопольский государственный университет»

Институт информационных технологий и управления в технических системах

кафедра Информационные системы

09.03.02 Информационные системы и технологии (уровень бакалавриата)

Лабораторная работа №1

по дисциплине: «Управление данными»

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

Вариант – 2

Выполнил

студент 3 курса группы ИС/б-33-о

Генералов Николай Николаевич

Отметка о зачете\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

(дата)

Проверил

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(должность) (подпись) (фамилия, инициалы)

г. Севастополь

2018 г.

# ЦЕЛЬ РАБОТЫ

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

# ПОСТАНОВКА ЗАДАЧИ



Рисунок 1 – Схема реализуемой базы данных

* Проанализировать схему БД рисунок 1, выделить и классифицировать все существующие связи, определить необходимые ограничения целостности.
* Создать базу данных и все ее таблицы.
* Установить связи между таблицами.
* Занести в таблицу образцы данных оператором INSERT INTO. Необходимо занести не менее 10 строк.
* Проверить работу ограничений целостности (каскадирование удаления, модификации и др.)
* Продемонстрировать выполнение простых вычислений в запросе.
* Использовать простое вычисление как параметр агрегатной функции.
* Продемонстрировать работу предложения GROUP BY.
* Продемонстрировать работу предложения HAVING.
* Ознакомиться с операциями реляционной алгебры.
* Применить к БД операции селекции и соединения в одном запросе.
* Создать запрос, использующий операции проекции и деления (в одном запросе).
* Создать запрос, использующий операции проекции, объединения и конъюнкции (в одном запросе).
* Создать запрос, использующий операции соединения и деления (в одном).
* Создать запрос, использующий операции вычитания и дизъюнкции (в одном запросе).

# ХОД РАБОТЫ

1. Выполним запросы для создания базы данных, таблиц и реализации связей между таблицами:

create database ud\_lab\_db2;

create table ud\_lab\_db2.teacher(

id int unsigned auto\_increment,

fio varchar(20) not null,

PRIMARY KEY(id)

);

create table ud\_lab\_db2.discipline\_course(

course\_num int unsigned,

teacher\_num int unsigned,

course\_name varchar(30) not null,

course\_description varchar(200),

pre\_course\_num int unsigned,

PRIMARY KEY(course\_num),

constraint `teacher\_num\_constraint` FOREIGN KEY (teacher\_num) references teacher(id)

on delete set null on update cascade,

constraint `pre\_course\_constraint` FOREIGN KEY(pre\_course\_num) references discipline\_course(course\_num)

on delete set null on update cascade

);

create table ud\_lab\_db2.lesson(

date\_of\_lesson date not null,

class\_room\_num int(3) unsigned not null,

course\_num int unsigned not null,

lesson\_type varchar(7),

PRIMARY KEY(date\_of\_lesson, class\_room\_num),

constraint `course\_num\_constraint` FOREIGN KEY (course\_num) references discipline\_course(course\_num)

on delete cascade on update cascade

);

create table ud\_lab\_db2.student(

id int unsigned not null auto\_increment,

fio varchar(20) not null,

PRIMARY KEY(id)

);

create table ud\_lab\_db2.stud\_course(

student\_id int unsigned not null,

course\_num int unsigned not null,

PRIMARY KEY (student\_id, course\_num),

constraint `s\_id\_constraint` foreign key (student\_id) references student(id) on delete cascade on update cascade,

constraint `couse\_num\_constraint` foreign key (course\_num) references discipline\_course(course\_num) on delete cascade on update cascade

);

Подтверждение выполнения запросов изображено на рисунке 2.

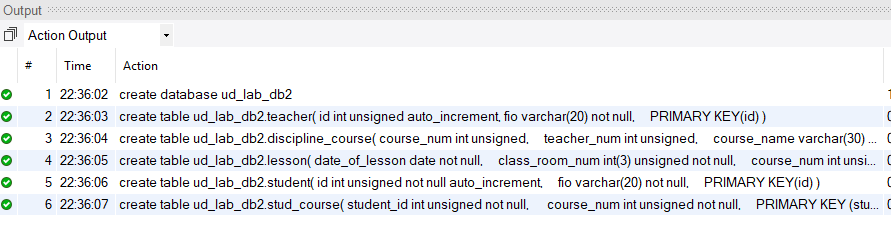


Рисунок 2 – Результат выполнения запросов на создание базы данных, таблиц и связей между таблицами.

1. Выполним запросы на заполнение таблиц данными с учетом ссылочной целостности:

insert into student (fio) values

("Иванов Иван Иванович"),

("Петров Петр Петрович"),

("Елена Петровна Хренова"),

("Альбиносов Петр Львович"),

("Кузнецов Иван Владимирович"),

("Белый Сергей Вадимович"),

("Созонов Фёдор Даниилович"),

("Селезнёв Константин Игоревич"),

("Вероломный Кирилл Викторович"),

("Авдеев Михаил Львович");

insert into teacher (fio) values

("Овечкин Степан Григорьевич"),

("Уникальный Андрей Константинович"),

("Нехороший Кирилл Викторович"),

("Закрученный Эльдар Сидович"),

("Бурый Владимир Владимирович"),

("Мурзов Биткоин Алексеевич"),

("Пальчик Людмила Даниловна"),

("Картавая Анастасия Валерьевна"),

("Крылатый Серафим Михайлович"),

("Богатый Рогат Муратович");

insert into discipline\_course values

(111011,1, "Высшая математика","ВМ",null),

(111012, 2, "Информатика", "ИС",null),

(111017,2, "Алгоритмизация и программирование", "АИП", 111012),

(111021, 8, "Русский язык", "РЯиСФ", null),

(111027, 5, "Численные методы", "ЧМ", 111017),

(111037,3, "Теория принятия решений", "ТПР", 111027),

(111044, 9, "Параллельные вычисления", "ПВ", 111017),

(111032, null, "Методы исследования операций", "МИО",111027),

(111035,1, "Экономическая теория","ВМ",null);

insert into lesson values

('2018-09-01', 512, 111011, "лекция"),

('2018-09-01', 507, 111012,"л.р."),

('2018-09-01', 506, 111017,"п.з."),

('2018-09-02', 512, 111035, null),

('2018-09-02', 501, 111011, "л.р."),

('2018-09-02', 419, 111027,"л.р."),

('2018-09-02', 510, 111044,"п.з."),

('2018-09-03', 512, 111035, "п.з"),

('2018-09-03', 419, 111011, "п.з"),

('2018-09-03', 101, 111012, "п.з"),

('2018-09-03', 207, 111017, "п.з"),

('2018-09-03', 202, 111021, "п.з"),

('2018-09-03', 509, 111027, "п.з"),

('2018-09-03', 402, 111032, "п.з"),

('2018-09-03', 501, 111037, "п.з"),

('2018-09-03', 502, 111044, "п.з");

insert into stud\_course values

(1, 111011),

(2,111011),

(3, 111011),

(4,111011),

(1, 111012),

(2,111012),

(3, 111012),

(4,111012),

(5,111021);

Подтверждения выполнения запросов и содержимое таблиц базы данных, изображены на рисунках 3 – 8.

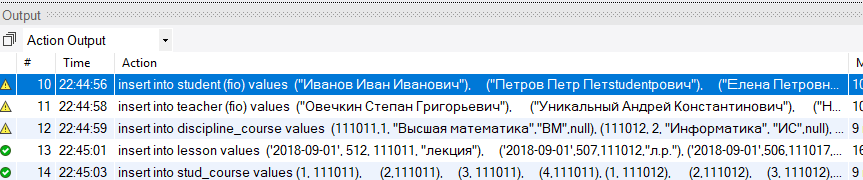


Рисунок 3 – Результат выполнения запросов на добавление данных в таблицы базы данных

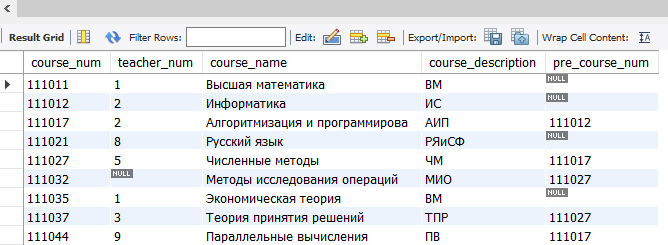


Рисунок 4 – Содержимое таблицы discipline\_course

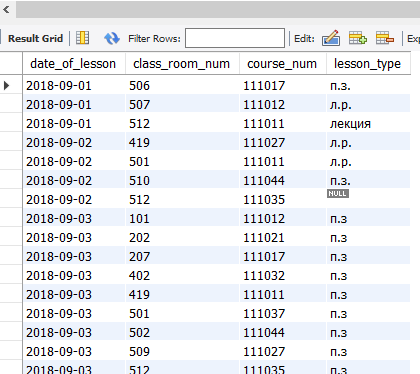


Рисунок 5 – Содержимое таблицы lesson

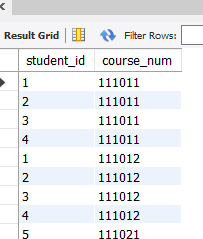


Рисунок 6 – Содержимое таблицы stud\_course

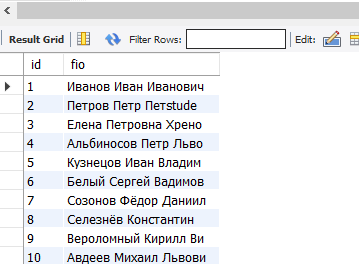


Рисунок 7 – Содержимое таблицы student

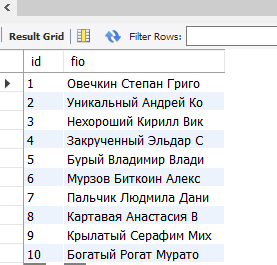


Рисунок 8 – Содержимое таблицы teacher

1. Выполним запросы для проверки ссылочной целостности при удалении и изменении данных таблиц:

DELETE FROM student where id=1; - Запрос на удаление студента имеющего идентификационный номер равный 1.

При выполнении данного запроса, удаляются соответствующие данные из таблицы stud\_course:

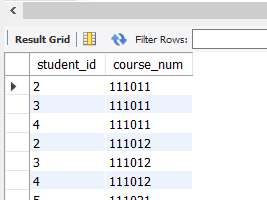


Рисунок 9 – Каскадирование данных в таблице stud\_course, при удалении данных из таблицы student

UPDATE student SET id = 15 WHERE id = 2; - запрос на изменение идентификационного номера студента имеющего идентификационный номер равный 1 имеющего идентификационный номер равный 2.

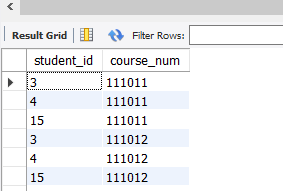


Рисунок 10 – Каскадирование данных в таблице stud\_course, при изменении данных в таблице student

Аналогичным образом проверим целостность остальных таблиц:

DELETE FROM discipline\_course WHERE course\_num = 111035; - запрос на удаление дисциплины с кодом 111035.

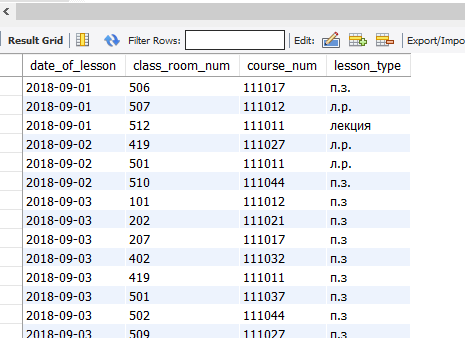


Рисунок 11 – Каскадирование таблицы lesson, при удалении данных из таблицы discipline\_course

1. Запрос с использованием простого вычисления:

SELECT \* FROM student WHERE (id > 5);

1. Запрос с использованием простого вычисления, как параметр агрегатной функции

SELECT min(class\_room\_num) AS `class\_room\_num` FROM lesson;

1. Запрос с использованием GROUP BY:

SELECT fio, count(course\_num) FROM discipline\_course, teacher WHERE id = teacher\_num GROUP BY fio;

1. Запрос с использованием HAVING:

SELECT fio, count(course\_num) FROM discipline\_course, teacher WHERE id = teacher\_num GROUP BY fio HAVING count(course\_num) >= 2;

1. Запрос с использованием селекции и объединения:

Определить номера дисциплин, по которым студентам выставлены оценки или которые запланированы по расписанию, но отличные от дисциплины с кодом 111011.

R1 =πcourse\_num(stud\_course);

R2 = πcourse\_num (Ϭcourse\_num != 111011(lesson));

R = R1 U R2;

SELECT course\_num from stud\_course UNION SELECT course\_num from lesson WHERE course\_num != 111011;

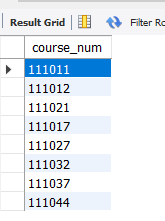


Рисунок 12 – Результат выполнения запроса п. 3.8

1. Запрос с использованием проекции и деления:

Определить даты когда проводятся занятия по всевозможным дисциплинам.

R1 = πcourse\_num (discipline\_course);

R2 = lesson;

R = πdate\_of\_lesson(R2 / R1);

SELECT distinct date\_of\_lesson FROM

lesson A WHERE NOT EXISTS

(SELECT course\_num

FROM discipline\_course

WHERE NOT EXISTS

(SELECT course\_num

FROM lesson B

WHERE B.course\_num = discipline\_course.course\_num AND

A.date\_of\_lesson = B.date\_of\_lesson));

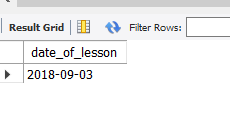


Рисунок 13 – Результат выполнения запроса п.3.9

1. Запрос с использованием операций проекции, объединения и конъюнкции

Определить коды дисциплин которые, либо имею номер дисциплины – 111035 и определен ведущий преподаватель, либо добавлены в расписание.

R1 = πcourse\_num­(Ϭcourse\_num=111035 ∩ teacher\_num != NULL(discipline\_course));

R2 = πcourse\_num­(lesson);

R = R1 U R2;

SELECT course\_num FROM discipline\_course WHERE course\_num = 111035 AND teacher\_num != NULL UNION SELECT course\_num FROM lesson;

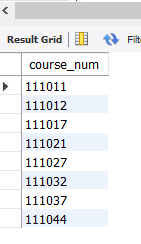


Рисунок 14 – Результат выполнения запроса п.3.10

1. Запрос с использованием операций соединения и деления.

Определить даты, когда проводятся всевозможные предметы и их названия.

R1 = πcourse\_num (discipline\_course);

R2 = (lesson >< discipline course)course\_num;

R = πdate\_of\_lesson,course\_name(R2 / R1);

SELECT date\_of\_lesson, course\_name FROM

lesson A JOIN discipline\_course C ON A.course\_num = C.course\_num WHERE NOT EXISTS

(SELECT course\_num

FROM discipline\_course

WHERE NOT EXISTS

(SELECT course\_num

FROM lesson B

WHERE B.course\_num = discipline\_course.course\_num AND

A.date\_of\_lesson = B.date\_of\_lesson)) ;

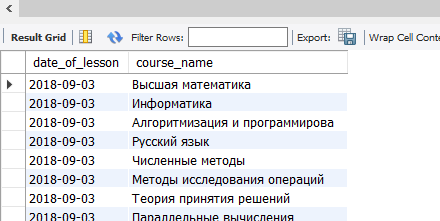


Рисунок 15 – Результат выполнения запроса п.3.11

1. Запрос с использованием операций вычитания и дизъюнкции.

Вывести номера предметов, проводимых 01.09.2018 или которые ожидаются по расписанию, но по которым не выставлены оценки.

R1 = πcourse\_num(lesson U Ϭdate\_of\_lesson = 2018-09-01(lesson));

R2 = πcourse\_num(stud\_course);

R = R1 – R2;

SELECT DISTINCT course\_num FROM lesson T1 WHERE NOT EXISTS(SELECT course\_num FROM stud\_course T2 WHERE T1.course\_num = T2.course\_num) OR date\_of\_lesson = '2018-09-01';

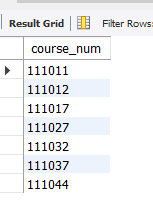


Рисунок 16 – Результат выполнения запроса п.3.12

1. Запрос не реализующийся в реляционной алгебре.

Вывести все данные о всех предметах.

SELECT \* FROM discipline\_course.

ВЫВОД

При выполнении лабораторной работы были изучены, а так же закреплены на практике основы реляционной алгебры. Так же были закреплены практические навыки построения запросов для создания базы данных, создания таблиц, создания связей между таблицами, и добавления данных в таблицы. Для этого была реализована база данных соответствующая схеме на рисунке 1.

По результатам выполнения запросов, можно сделать вывод, что база работоспособна и операции реляционной алгебры выполнены верно, а значит поставленная цель достигнута.