Министерство образования и науки РФ

Государственное общеобразовательное учреждение высшего профессионального образования

Владимирский Государственный Университет

имени Александра Григорьевича и Николая Григорьевича Столетовых

Кафедра информационных систем и информационного менеджмента

Лабораторная работа № 1

**Знакомство с MySQL и выполнение элементарных операций при работе с СУБД.**

Выполнил:

ст. гр. ИСТ-109

Ларин А.А.

Проверила:

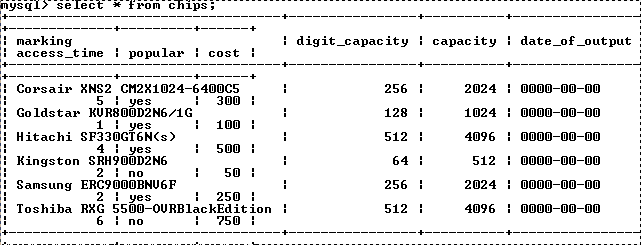
Проскурина Г.В.

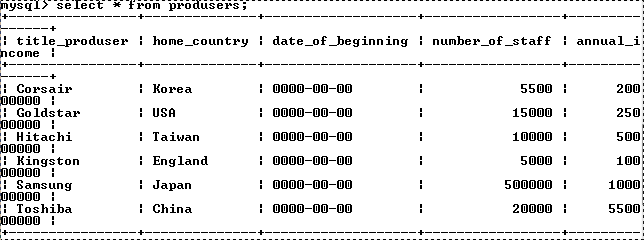
Владимир 2011

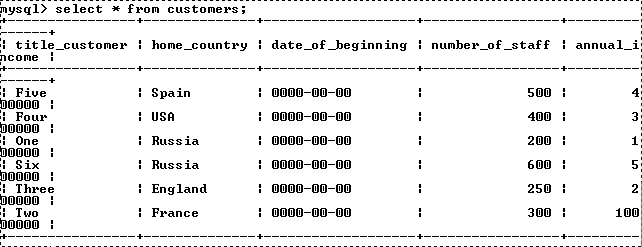
**Цель работы**: Изучить технические возможности СУБД MySQL при работе с простейшими базами данных и средств формирования SQL запросов к СУБД.

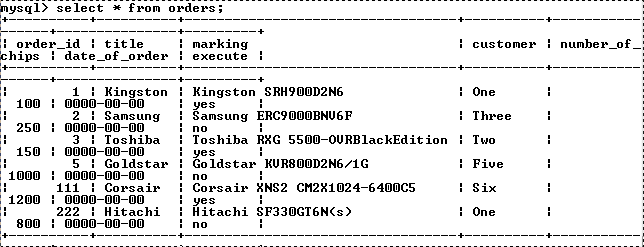
**Индивидуальное задание**: ИС «Микросхемы памяти».

**Созданные таблицы:**

****

****

****

****

**Описание связей:**

|  |
| --- |
| Производители |
| название фирмы  страна  дата основания  количество персонала  годовой доход |

|  |
| --- |
| Микросхемы |
| маркировка  разрядность  емкость  дата начала выпуска  время доступа  популярность  стоимость |

PK

PK

|  |
| --- |
| Заказы  FK |
| Id\_заказа  название производителя  маркировка микросхем  FK  название заказчика  количество микросхем  FK  дата заказа  выполнен ли заказ  PK |

|  |
| --- |
| Заказчики |
| Название заказчика  Страна  Количество персонала  Годовой доход |

**SQL-скрипты:**

create database chips;

use chips;

create table chips(marking varchar(40) not null primary key, digit\_capacity int, capacity int, date\_of\_output date, access\_time int, popular varchar(10), cost int);

create table produsers(title\_produser varchar(40) not null primary key, home\_country varchar(20), date\_of\_beginning date, number\_of\_staff int, annual\_income int);

create table customers(title\_customer varchar(40) not null primary key, home\_country varchar(20), date\_of\_beginning date, number\_of\_staff int, annual\_income int);

create table orders(order\_id int not null primary key, title varchar(40), marking varchar(40), customer varchar(40), number\_of\_chips int,date\_of\_order date, execute varchar(10), foreign key (title) references produsers(title\_produser), foreign key (marking) references chips(marking), foreign key (customer) references customers(title\_customer));

insert into chips(marking, digit\_capacity, capacity, date\_of\_output, access\_time, popular, cost) values ('Goldstar KVR800D2N6/1G',128, 1024, 08/02/2008, 1, 'yes', 100);

insert into chips(marking, digit\_capacity, capacity, date\_of\_output, access\_time, popular, cost) values ('Kingston SRH900D2N6',64, 512, 01/01/2001, 2, 'no', 50);

insert into chips(marking, digit\_capacity, capacity, date\_of\_output, access\_time, popular, cost) values ('Samsung ERC9000BNV6F',256, 2024, 30/01/2010, 2, 'yes', 250);

insert into chips(marking, digit\_capacity, capacity, date\_of\_output, access\_time, popular, cost) values ('Hitachi SF330GT6N(s)',512, 4096, 25/10/2011, 4, 'yes', 500);

insert into chips(marking, digit\_capacity, capacity, date\_of\_output, access\_time, popular, cost) values ('Corsair XNS2 CM2X1024-6400C5',256, 2024, 05/11/2009, 5, 'yes', 300);

insert into chips(marking, digit\_capacity, capacity, date\_of\_output, access\_time, popular, cost) values ('Toshiba RXG 5500-OVRBlackEdition',512, 4096, 10/03/2011, 6, 'no', 750);

insert into produsers(title\_produser, home\_country, date\_of\_beginning, number\_of\_staff, annual\_income) values ('Kingston', 'England', 09/03/1990, 5000, 10000000);

insert into produsers(title\_produser, home\_country, date\_of\_beginning, number\_of\_staff, annual\_income) values ('Samsung', 'Japan', 09/02/1946, 500000, 100000000);

insert into produsers(title\_produser, home\_country, date\_of\_beginning, number\_of\_staff, annual\_income) values ('Corsair', 'Korea', 01/11/1998, 5500, 20000000);

insert into produsers(title\_produser, home\_country, date\_of\_beginning, number\_of\_staff, annual\_income) values ('Hitachi', 'Taiwan', 23/12/1988, 10000, 50000000);

insert into produsers(title\_produser, home\_country, date\_of\_beginning, number\_of\_staff, annual\_income) values ('Goldstar', 'USA', 03/02/1995, 15000, 25000000);

insert into produsers(title\_produser, home\_country, date\_of\_beginning, number\_of\_staff, annual\_income) values ('Toshiba', 'China', 04/08/1973, 20000, 550000000);

insert into customers(title\_customer, home\_country, date\_of\_beginning, number\_of\_staff, annual\_income) values ('One', 'Russia', 22/11/2011, 200, 100000);

insert into customers(title\_customer, home\_country, date\_of\_beginning, number\_of\_staff, annual\_income) values ('Two', 'France', 21/10/2011, 300, 10000000);

insert into customers(title\_customer, home\_country, date\_of\_beginning, number\_of\_staff, annual\_income) values ('Three', 'England', 11/11/2011, 250, 200000);

insert into customers(title\_customer, home\_country, date\_of\_beginning, number\_of\_staff, annual\_income) values ('Four', 'USA', 1/11/2011, 400, 300000);

insert into customers(title\_customer, home\_country, date\_of\_beginning, number\_of\_staff, annual\_income) values ('Five', 'Spain', 2/11/2011, 500, 400000);

insert into customers(title\_customer, home\_country, date\_of\_beginning, number\_of\_staff, annual\_income) values ('Six', 'Russia', 23/11/2011, 600, 500000);

insert into orders(order\_id, title, marking, customer, number\_of\_chips, date\_of\_order, execute) values (001,'Kingston','Kingston SRH900D2N6', 'One', 100, 25/10/2010, 'yes');

insert into orders(order\_id, title, marking, customer,number\_of\_chips, date\_of\_order, execute) values (002, 'Samsung','Samsung ERC9000BNV6F', 'Three', 250, 05/11/2011, 'no');

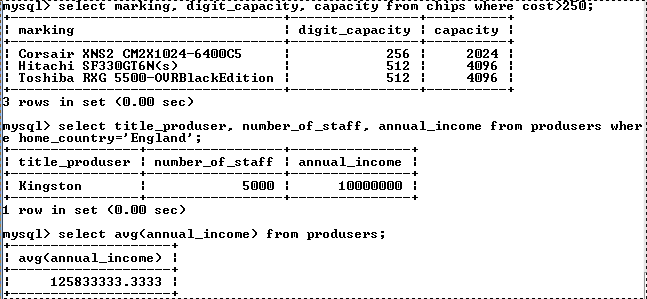
insert into orders(order\_id, title, marking, customer,number\_of\_chips, date\_of\_order, execute) values (003, 'Toshiba','Toshiba RXG 5500-OVRBlackEdition', 'Two', 150, 10/01/2011, 'yes');

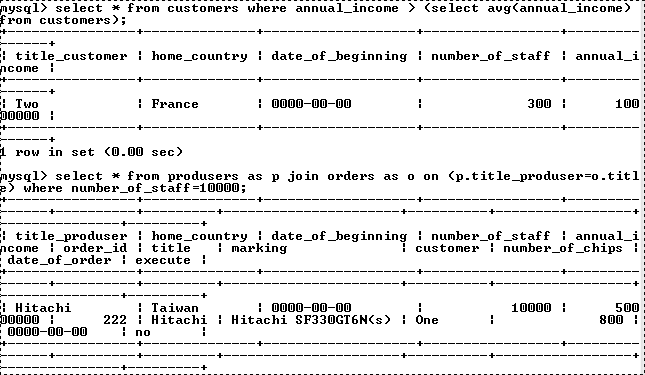
insert into orders(order\_id, title, marking, customer, number\_of\_chips, date\_of\_order, execute) values (005, 'Goldstar','Goldstar KVR800D2N6/1G', 'Five', 1000, 18/10/2011, 'no');

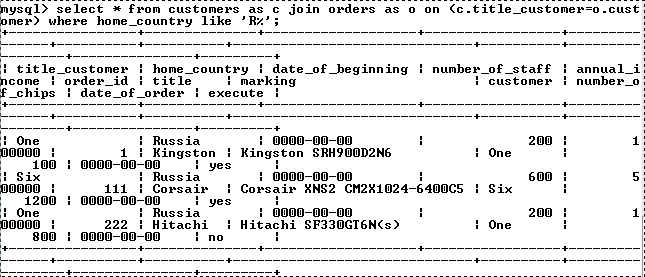
insert into orders(order\_id, title, marking, customer, number\_of\_chips, date\_of\_order, execute) values (111, 'Corsair','Corsair XNS2 CM2X1024-6400C5', 'Six', 1200, 10/10/2010, 'yes');

insert into orders(order\_id, title, marking, customer, number\_of\_chips, date\_of\_order, execute) values (222, 'Hitachi','Hitachi SF330GT6N(s)', 'One', 800, 26/10/2011, 'no');

**Запросы:**

****

****

****

Вывод: в ходе работы изучены технические возможности СУБД MySQL при работе с простейшими базами данных и средств формирования SQL запросов к СУБД. Реализованы некоторые операции реляционной алгебры.

**Ответы на контрольные вопросы:**

1. Реляционная алгебра — формальная система манипулирования отношениями в реляционной модели данных. Существует в двух несколько различающихся вариантах:

алгебра Кодда (Э. Кодд, 1970)

алгебра A (К. Дейт, Х. Дарвен)

Наряду с реляционным исчислением является способом получения результирующего отношения в реляционной модели данных.

1. **Теоретико-множественные операторы**

**Объединение**

Отношение с тем же заголовком, что и у совместимых по типу отношений A и B, и телом, состоящим из кортежей, принадлежащих или A, или B, или обоим отношениям.

Синтаксис:

A UNION B

**Пересечение**

Отношение с тем же заголовком, что и у отношений A и B, и телом, состоящим из кортежей, принадлежащих одновременно обоим отношениям A и B.

Синтаксис:

A INTERSECT B

**Вычитание**

Отношение с тем же заголовком, что и у совместимых по типу отношений A и B, и телом, состоящим из кортежей, принадлежащих отношению A и не принадлежащих отношению B.

Синтаксис:

A MINUS BСпециальные реляционные операторы

1. Выборка (ограничение)

Отношение с тем же заголовком, что и у отношения A, и телом, состоящим из кортежей, значения атрибутов которых при подстановке в условие c дают значение ИСТИНА. c представляет собой логическое выражение, в которое могут входить атрибуты отношения A и/или скалярные выражения.

Синтаксис:

A WHERE c

Проекция

Отношение с заголовком (X, Y, …, Z) и телом, содержащим множество кортежей вида (x, y, …, z), таких, для которых в отношении A найдутся кортежи со значением атрибута X равным x, значением атрибута Y равным y, …, значением атрибута Z равным z. При выполнении проекции выделяется «вертикальная» вырезка отношения-операнда с естественным уничтожением потенциально возникающих кортежей-дубликатов.

Синтаксис:

A[X, Y, …, Z]

или

PROJECT A {x, y, …, z}

Соединение

Операция соединения есть результат последовательного применения операций декартового произведения и выборки. Если в отношениях и имеются атрибуты с одинаковыми наименованиями, то перед выполнением соединения такие атрибуты необходимо переименовать.

Синтаксис:

(A TIMES B) WHERE c

Деление

Отношение с заголовком (X1, X2, …, Xn) и телом, содержащим множество кортежей (x1, x2, …, xn), таких, что для всех кортежей (y1, y2, …, ym) ∈ B в отношении A(X1, X2, …, Xn, Y1, Y2, …, Ym) найдется кортеж (x1, x2, …, xn, y1, y2, …, ym).

Синтаксис:

A DIVIDEBY B

**Data Definition Language (DDL)** – язык определения данных, позволяющий создавать, удалять и изменять объекты в базах данных; 4. **Синтаксис DDL команд**

Оператор создания таблиц БД позволяет создавать новые таблицы и указывать связи между ними. Синтаксис:

CREATE TABLE <Имя таблицы> {<Определение столбца> |

[<Ограничение таблицы>]},

где:

<Определение столбца>::=

<имя столбца>

{<имя домена>|<тип данных>}

[<ограничение столбца>]

<Ограничение столбца>::=

UNIQUE|

PRIMARY KEY|

FOREIGN KEY (<имя столбца>)|

REFERENCES <имя таблицы>[(<имя столбца>…)] NOT NULL.

Имена таблиц и определения столбцов должны быть уникальными.

Назначение каждого атрибута в <Ограничение столбца> приведено далее:

• UNIQUE – определяет уникальное (не повторяющееся значение атрибута);

• PRIMARY KEY – первичный ключ;

• FOREIGN KEY – внешний ключ;

• REFERENCES – определение ссылки на другую таблицу;

• NOT NULL – указывает, что значение атрибута не должно быть пустым;

Оператор изменения структуры таблицы позволяет изменить структуру таблицы, имена атрибутов и их типы. Синтаксис:

ALTER TABLE <имя таблицы>

{ADD <определение столбца>}|

ALTER <имя столбца>|

DROP <имя столбца>|

ADD <ограничение таблицы>.

Следует заметить, что если пытаться изменить имя или тип столбца, на который имеется ссылка из другой таблицы, то будет выдано сообщение об ошибке, и изменения нельзя будет произвести. Для этого сначала необходимо разорвать связь между таблицами.

Оператор создания БД позволяет создавать новые БД в СУБД для

последующего использования при хранении и доступе к данным. Синтаксис:

CREATE DATABASE '<имя файла БД>'

USER '<имя пользователя>'

PASSWORD '<пароль пользователя>',

где:

<имя файла БД> – имя физического файла, в котором будет храниться БД, расширение в имени указывается явно, как правило “gbd”. Создание базы данных происходит в два этапа. Сначала физически создается файл БД оператором CREATE DATABASE, затем выполняется регистрация БД в СУБД.

1. Язык SQL состоит из набора операторов, которые можно разделить на несколько категорий:

• Data Definition Language (DDL) – язык определения данных, позволяющий создавать, удалять и изменять объекты в базах данных;

• Data Manipulation Language (DML) – язык управления данными, позволяющий модифицировать, добавлять и удалять данные в имеющихся объектах базы данных;

• Data Control Language (DCL) – язык, используемый для управления пользовательскими привилегиями;

• Transaction Control Language (TCL) – язык для управления изменениями, сделанными группами операторов;

• Cursor Control Language (CCL) – язык для определения курсоров, подготовки операторов SQL к выполнению и некоторых других операций