**1 ЛАБОРАТОРНАЯ РАБОТА №1**

**«Исследование командного интерпретатора ОС семейства UNIX»**

**1.1 Цель работы**

Ознакомится с описанием организации работы реляционных БД. Изучить способы создания баз данных. Изучить запросы ISQL.

**1.2 Вариант задания – 5**

Требуется ознакомиться с описанием организации работы реляционных БД.

Создать тестовую БД.

− создать тестовую таблицу,

− занести в таблицу пять кортежей,

− просмотреть содержимое таблицы.

Ознакомиться с описанием базовых операторов для работы с таблицами.

Создать таблицу. Особое внимание надо уделить описанию первичного ключа, значений по умолчанию, описателям NOT NULL и конструкции CHECK.

Занести в таблицу образцы данных оператором INSERT INTO. Необходимо занести не менее 10 строк. Внимание! После того, как в таблицу занесены образцы данных, менять структуру таблицы можно только оператором ALTER TABLE.

Создать запрос, выводящий все строки таблицы.

Создать запрос, задающий порядок столбцов, отличный от исходного.

Продемонстрировать действие модификатора DISTINCT.

Ограничить вывод запроса, используя WHERE с простым условием.

Ограничить вывод запроса, используя WHERE и составное условие.

Продемонстрировать действие специальных функций IN, BETWEEN, LIKE, и IS NULL в условии.

Продемонстрировать работу специальных функций с условием NOT

**1.3 Ход работы**

1.3.1 Был скачан Firebird classic и запущена утилита Firebird ISQL Tool. С помощью запущенной утилиты была создана тестовая база данных (Рисунок 1.1).

Создание базы данных находящейся по “указанному пути”

Имя пользователя: “SYSDBA” пароль “masterkey”

Размер страницы = 4096;

Набор символов по умолчанию win1251;

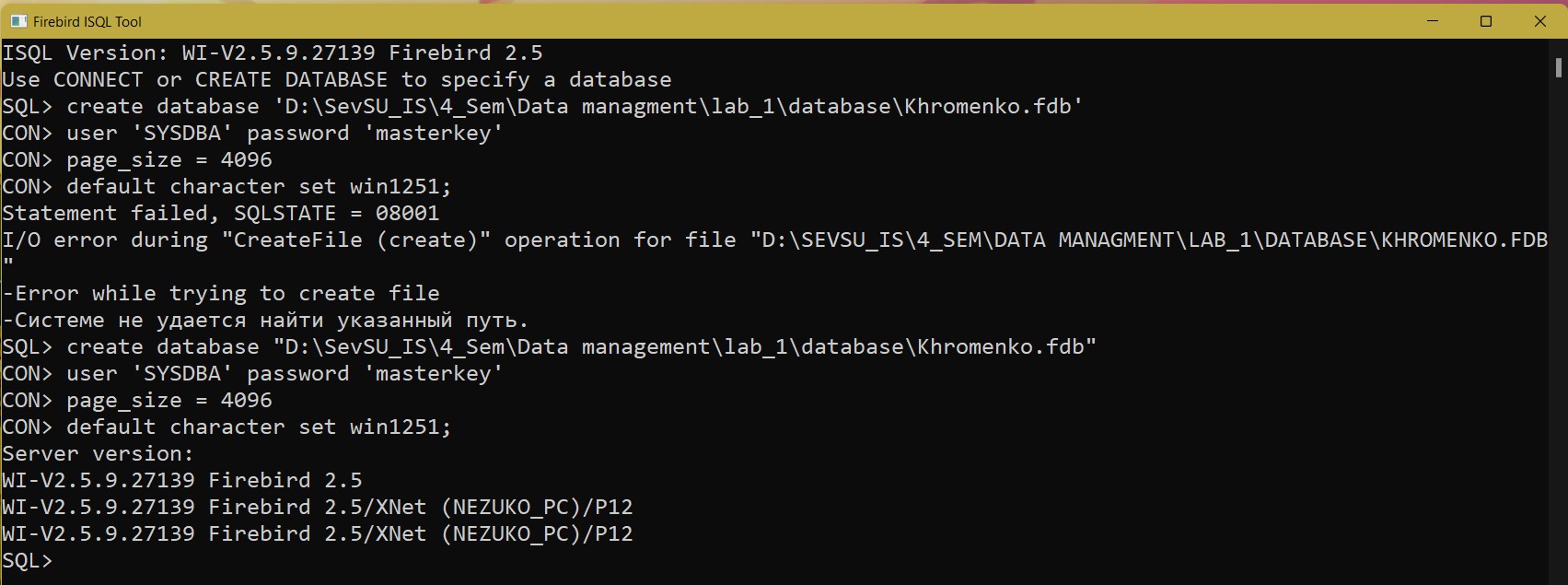


Рисунок 1.1 – Создание тестовой базы данных

Затем была создана тестовая таблица, названную “Friends” с помощью оператора “CREATE TABLE”. Процедуру создания таблицы можно лицезреть на рисунке 1.2.

Создать таблицу Друзья (

Атрибут: Номер\_друга целого типа не NULL,

Атрибут: Имя\_друга строкового типа (10 символьного) не NULL,

Атрибут: Фамилия\_друга строкового типа (20 символов) не NULL,

Атрибут: Возраст целого типа (от 17 до 21),

Атрибут: Зарплата целого типа по умолчанию 0);

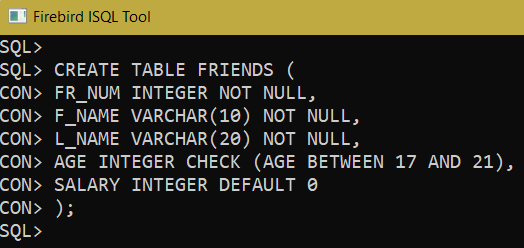


Рисунок 1.2 – Создание тестовой таблицы

Далее с помощью оператора “INSERT” таблица была заполнена тестовыми кортежами (Рисунок 1.3).

Вставить в таблицу Друзья значения (2 в атрибут номер друга, Алексей в атрибут имя друга, Филозоп в атрибут фамилия друга, 19 в атрибут возраст, 3100 в атрибут зарплата);

Вставить в таблицу друзья (список атрибутов) значения (значения атрибутов);

…

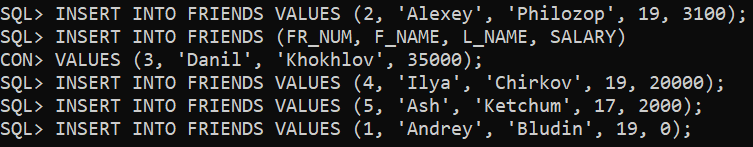


Рисунок 1.3 ­– Заполнение тестовой таблицы пятью кортежами

Затем с помощью оператора “SELECT” содержимое таблицы было выведено на экран (Рисунок 1.4).

Вывести все кортежи из таблицы друзья;

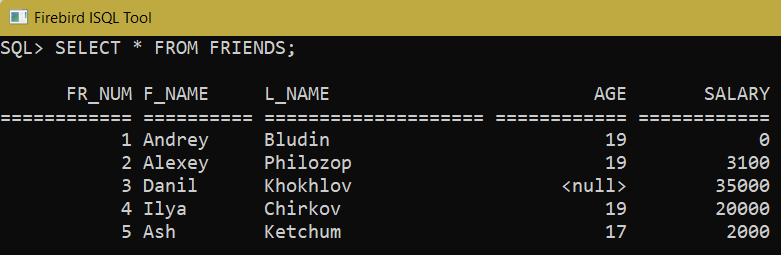


Рисунок 1.4 – Вывод содержимого таблицы на экран

Были созданы две таблицы согласно варианту с использованием описателей “NOT NULL” и конструкции “CHECK”. Создание таблица изображено на рисунке 1.5.

Создать таблицу Фирмы (Атрибуты: Номер компании целого типа не NULL, Имя компании строкового типа (20 символов) не NULL, Адрес компании строкового типа(30 символов), контактное лицо строкового типа (15 символов) не NULL, номер счета в банке большого целого типа не NULL);

Создать таблицу типы помещений (Атрибуты: Номер помещения (между 14 и 2222), Тип помещения строкового типа (20 символов) не NULL);

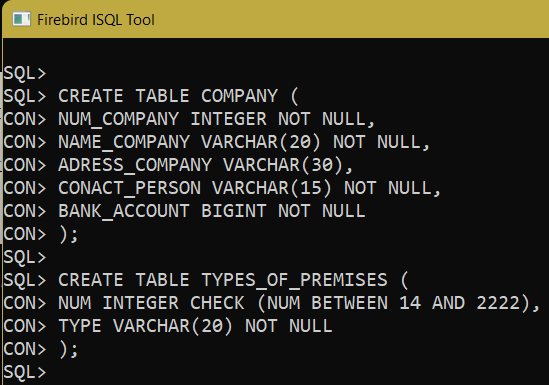


Рисунок 1.5 – Создание таблиц согласно варианту

Затем в каждую из таблиц было добавлен по 7 различных кортежей. Часть создания кортежей изображена на рисунке 1.6.

Добавить запись в таблицу Фирмы со значениями (3689, …);

… Добавить запись в таблицу Типы\_помещений со значениями (14, “Склад”); …

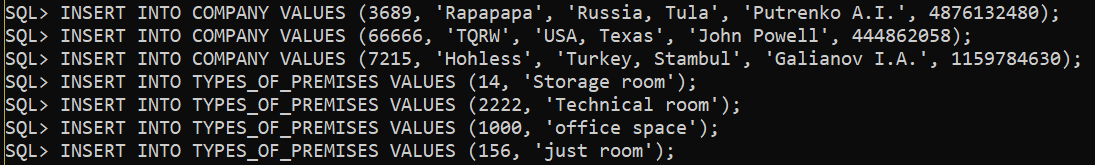


Рисунок 1.6 – Заполнение таблиц

Далее на экран была выведена информация, содержащаяся в таблицах с помощью конструкции “SELECT \* FROM” (Рисунок 1.7).

Вывести все кортежи из таблицы фирмы;

Вывести все кортежи из таблицы типы\_помещений;

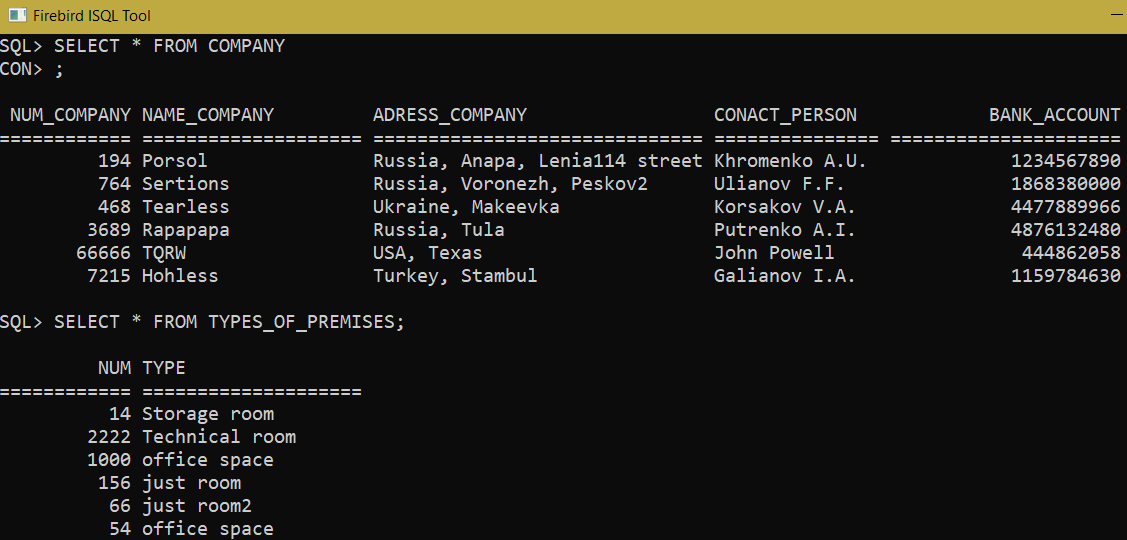


Рисунок 1.7 – Вывод на экран информации содержащейся в таблицах

Был создан запрос, задающий порядок столбцов, отличный от исходного (Рисунок 1.8).

Вывести все кортежи с атрибутами контактное лицо, имя компании, банковский счет из таблицы Фирмы с сортировкой по банковским счетам;

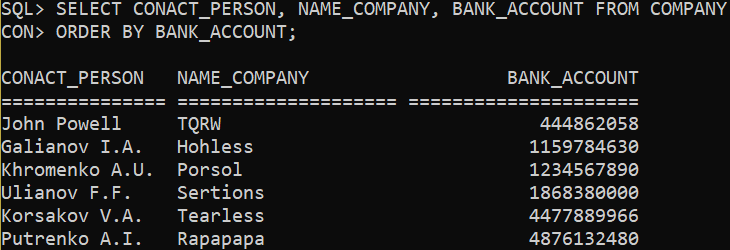


Рисунок 1.8 – Отображение информации в таблице с иным порядком столбцов

Затем для демонстрации действия модификатора “DISTINCT”, были созданы несколько одинаковых записей во второй таблице. Действие модификатора DISTINCT можно видеть на рисунке 1.9.

Добавление трех одинаковых кортежей в таблицу Типов помещений со значениями (69, ‘просто помещение 3’);

Вывести на экран без повторений все кортежи из таблицы Типы помещений;

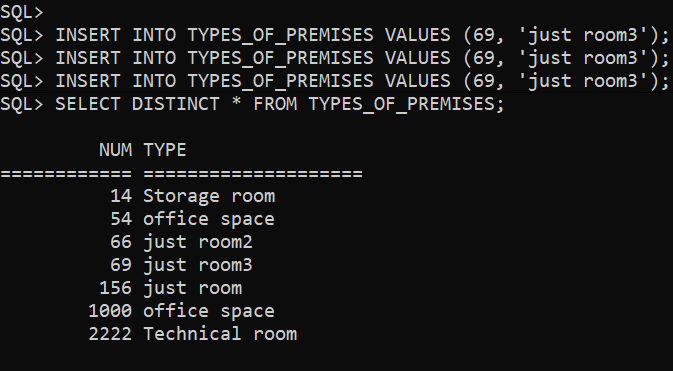


Рисунок 1.9 – Демонстрация действия модификатора DISTINCT

Далее для демонстрации конструкции, использующей WHERE, был проделан запрос, изображенный на рисунке 1.10.

Вывести на экран все кортежи из таблицы друзья где возраст равен 19 и ЗП больше 3000;

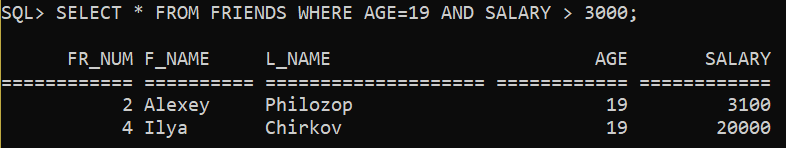


Рисунок 1.10 – Запрос SELECT с использованием WHERE

**Выводы**

При выполнении данной лабораторной работы были получены навыки работы с ISQL. Были изучены способы создания баз данных, запросы ISQL. Полученные навыки помогут в будущем эффективнее создавать базы данных.