Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Институт информационных технологий и управления в технических системах

Кафедра ИС

ОТЧЕТ

по лабораторной работе №1

ИЗУЧЕНИЕ ОСНОВ ЯЗЫКА МАНИПУЛИРОВАНИЯ ДАННЫМИ SQL НА БАЗЕ СЕРВЕРА FIREBIRD

Выполнил:

ст. гр. ИС/б-21-о

Куркчи А. Э.

Проверил:

Тимофеева Т. А.

Севастополь

2016

1. ЦЕЛЬ РАБОТЫ

Изучить основы организации сервера Firebird; научится устанавливать соединение с сервером. Научиться создать базу данных и производить элементарные действия над ней. Изучить формы оператора CREATE TABLE и простейшие формы оператора SELECT.

2. ВАРИАНТ ЗАДАНИЯ

Вариант № 7

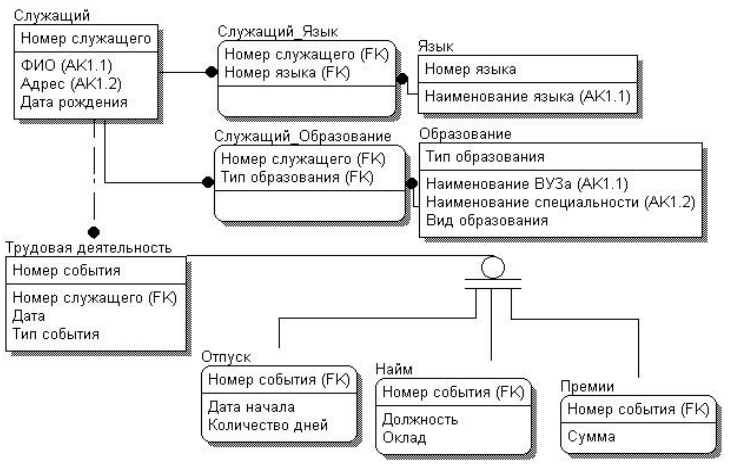


Рисунок 1 — Вариант задания

3. ХОД РАБОТЫ

Ниже представлен набор команд на языке SQL, реализующий создание БД и заполнение её тестовыми данными.

CREATE DATABASE 'D:\university\semestr.04\DM\Lab.01\db.fdb'

USER 'SYSDBA' PASSWORD 'masterkey'

page\_size = 4096

DEFAULT CHARACTER SET win1251;

CONNECT 'D:\university\semestr.04\DM\Lab.01\db.fdb' USER 'SYSDBA' PASSWORD 'masterkey';

CREATE TABLE employee (

id INTEGER NOT NULL,

name VARCHAR(32) NOT NULL,

address VARCHAR(32),

b\_date DATE

);

INSERT INTO employee VALUES (1, 'Andrew', 'Sevastopol', '12.09.1978');

INSERT INTO employee VALUES (2, 'John', 'Yalta', NULL);

INSERT INTO employee VALUES (3, 'Bob', 'Kerch', NULL);

INSERT INTO employee VALUES (4, 'Mary', 'Sevastopol', '22.09.1996');

INSERT INTO employee VALUES (5, 'Jerry', 'Simferopol', '09.05.1945');

INSERT INTO employee VALUES (6, 'Arif', 'Simferopol', '06.10.1997');

INSERT INTO employee VALUES (7, 'Jane', 'Alushta', NULL);

INSERT INTO employee VALUES (8, 'Kate', NULL, NULL);

INSERT INTO employee VALUES (9, 'Iris', NULL, NULL);

INSERT INTO employee VALUES (10, 'Vladimir', NULL, NULL);

SELECT \* FROM employee;

SELECT \* FROM employee ORDER BY b\_date;

SELECT DISTINCT address FROM employee;

SELECT \* FROM employee WHERE address = 'Simferopol';

SELECT \* FROM employee WHERE address = 'Simferopol' OR address = 'Yalta';

SELECT \* FROM employee WHERE id IN (2, 4, 6, 8, 10);

SELECT \* FROM employee WHERE b\_date BETWEEN '01.01.1990' AND '01.01.2000';

SELECT \* FROM employee WHERE address LIKE ‘S%’;

SELECT \* FROM employee WHERE b\_date IS NULL’;

4. ПРИМЕРЫ ЗАПРОСОВ

4.1 Отображение всех полей таблицы.

При помощи команды SELECT \* FROM employee; на экране отобразятся все поля таблицы employee (рисунок 2).

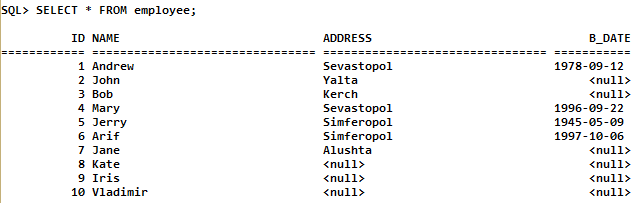


Рисунок 2 — Отображение всех столбцов таблицы

4.2 Сортировка таблицы по полю b\_date:

Запрос: SELECT \* FROM employee ORDER BY b\_date;

Как видно на рисунке 3 таблица теперь упорядочена по возрастанию даты рождения.

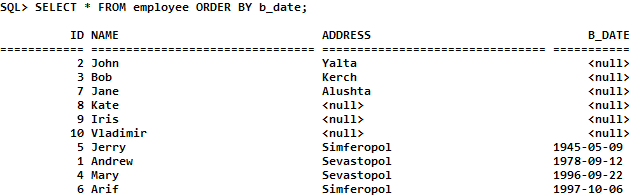


Рисунок 3 — Упорядочивание таблицы по дате рождения

4.3 Отображение уникальных значений поля address:

С помощью команды SELECT DISTINCT address FROM employee; на экран можно вывести уникальные значения столбца адреса (рисунок 4).

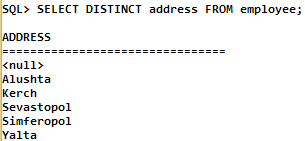


Рисунок 4 — Отображение уникальных значений столбца

4.4 Отображение данных о сотрудниках, проживающих в Симферополе.

Запрос: SELECT \* FROM employee WHERE address = 'Simferopol';

Результат запроса приведен на рисунке 5.

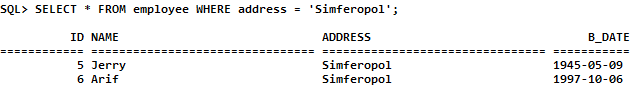


Рисунок 5 — Использование простого условия поиска

4.5 Отображение списка всех сотрудников, проживающих в Симферополе или Ялте.

Запрос: SELECT \* FROM employee WHERE address = 'Simferopol' OR address = 'Yalta';

Результат запроса приведен на рисунке 6.

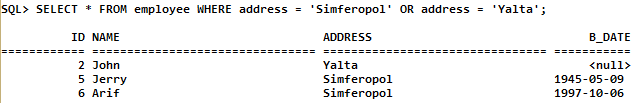


Рисунок 6 — Использование составного условия поиска

4.6 Отображение списка всех сотрудников, номер которых равен 2, 4, 6, 8 или 10.

Запрос: SELECT \* FROM employee WHERE id IN (2, 4, 6, 8, 10);

Результат запроса приведен на рисунке 7.

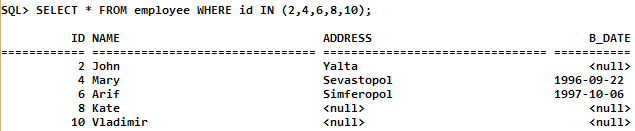


Рисунок 7 — Использование условия вхождения в множество

4.7 Отображение cписка сотрудников, день рождения которых лежит в диапазоне между 01.01.1990 и 01.01.2000.

Запрос: SELECT \* FROM employee WHERE b\_date BETWEEN '01.01.1990' AND '01.01.2000';

Результат представлен на рисунке 8.

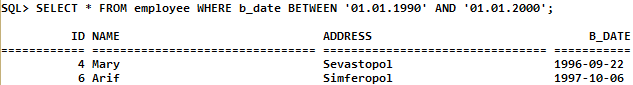


Рисунок 8 — Отображения столбца по диапазону значений

4.8 Отображение сотрудников, адрес которых начинается на «S».

Запрос: SELECT \* FROM employee WHERE address LIKE ‘S%’;

Результат запроса представлен на рисунке 9.

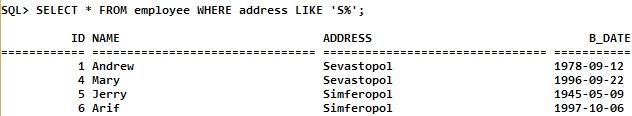


Рисунок 9 — Отображения строк, содержащих в адресе подстроку

4.9 Отображение сотрудников, адрес которых не начинаются на «S»

Запрос: SELECT \* FROM employee WHERE address NOT LIKE ‘S%’;

Результат выполнения запроса представлен на рисунке 10.

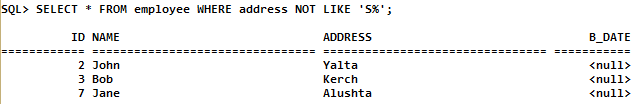


Рисунок 10 — Отображения строк исключая адреса с входящей подстрокой

4.10 Отображение сотрудников, для которых не указана дата рождения

Запрос: SELECT \* FROM employee WHERE b\_date IS NULL;

Результат выполнения запроса представлен на рисунке 11.

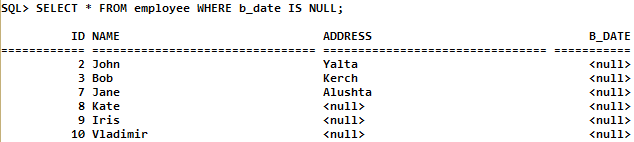


Рисунок 11 — Отображение сотрудников, для которых не указана дата рождения

Вывод

В ходе данной лабораторной работы были изучены основы организации и настройки сервера Firebird, получены навыки работы с реляционными базами данных, заполнение таблиц путем использования запросов SQL. Результатом выполнения лабораторной работы стало создание таблицы при помощи запроса CREATE TABLE, а также её заполнение данными и вывод при помощи операторов INSERT и SELECT. Были изучены и продемонстрированы на примерах простые и составные формы оператора SELECT.