|  |
| --- |
| **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ** |
| **ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ** |
| **УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ** |
| **«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»** |
|  |
| Институт информационных технологий и управления в технических системах |
| (полное название института) |
|  |
| кафедра «Информационные системы» |
| (полное название кафедры) |

**ОТЧЁТ**

по лабораторной работе №1

на тему«изучение основ языка манипулирования данными sql на базе сервера Firebird»

по дисциплине **«**Теория баз данных»

Вариант 16

Выполнил

студент ИИТУТС

группы ИС/б-18-2-о

Радыгина Екатерина

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | | | | | | | |
|  | (должность, учёная степень преподавателя) | | | | | | | | |
|  |  | | | | | | | | |
|  | (ФИО преподавателя) | | | | | | | | |
|  | « |  | » |  |  |  | 20 | 20 | г. |
|  |  | | | | | | | | |
|  | (оценка) | | | | | | | | |
|  |  | | | | | | | | |
|  |  | | | | | | | | |

Севастополь 2020

* 1. **Цель работы**

Рассмотреть правила организации работы в среде многопользовательской реляционной БД Firebird, а также основные операторы языка описания данных и языка манипулирования данными.

* 1. **Вариант задания**

Вариант 16

На рисунке 1.1 изображена структура системы, которая содержит информацию о грузовых перевозках, осуществляемых различными фирмами.

Категория водительских прав – A, B, C.

Характер груза – твердый, жидкий и т.д.

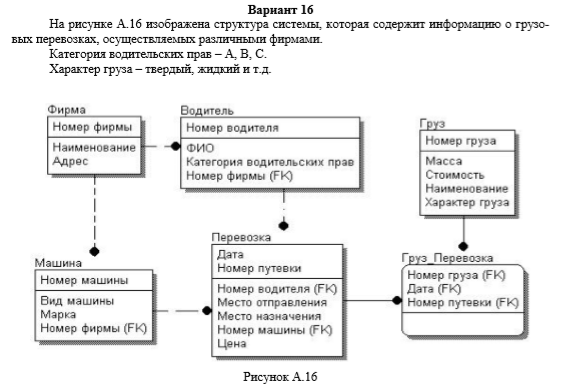


Рисунок 1.1 – Вариант задания

* 1. **Ход работы**
     1. Создадим базу данных (рисунок 1.2) и попробуем соединиться с ней (рисунок 1.3).

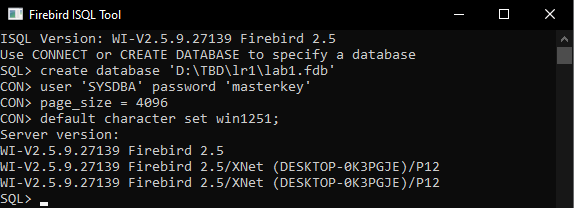


Рисунок 1.2 – Создание базы данных

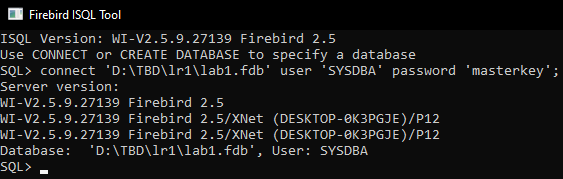


Рисунок 1.3 – Подключение к базе данных

* + 1. Создадим таблицы в соответствии с вариантом (рисунок 1.4).

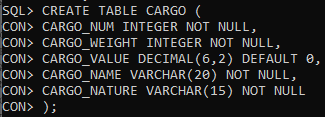
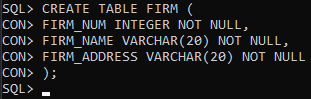


Рисунок 1.4 – Создание таблиц

* + 1. Заполним таблицу FIRM с помощью команды INSERT и выведем ее через команду SELECT (рисунок 1.5).

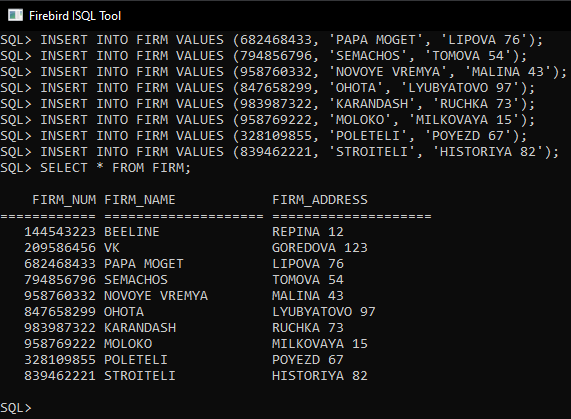


Рисунок 1.5 – Заполнение таблицы FIRM

Аналогично для таблицы CARGO (рисунок 1.6).

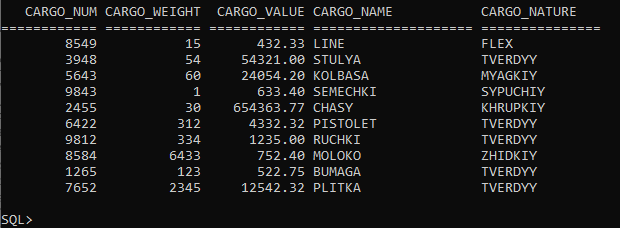


Рисунок 1.6 – Таблица CARGO

1.3.4 Создадим запрос, меняющий порядок столбцов в таблице и выведем ее (рисунок 1.7).

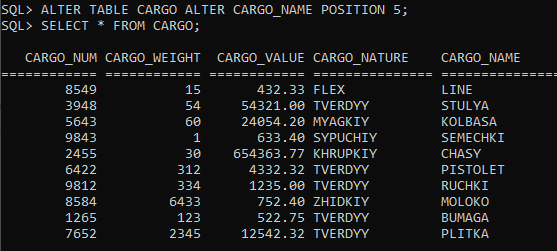


Рисунок 1.7 – Замена CARGO\_NATURE на CARGO\_NAME

1.3.5 Выведем с помощью команды DISTINCT количество типов груза в столбце CARGO\_NATURE (рисунок 1.8).

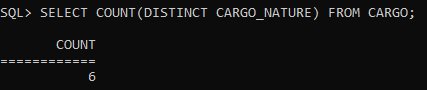


Рисунок 1.8 – Количество типов груза

1.3.6 Ограничим вывод запроса, используя WHERE с простым условием (рисунок 1.9) и составным (рисунок 1.10).

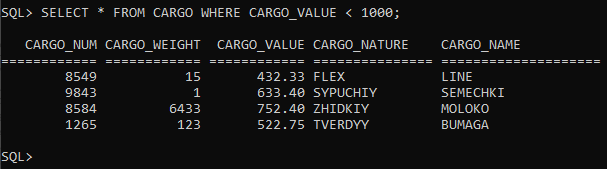


Рисунок 1.9 – Простое условие

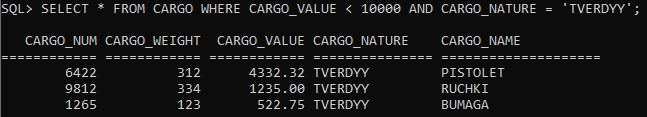


Рисунок 1.10 – Составное условие

1.3.7 С помощью специальной функций IN в условии выведем только твердый и гибкий тип груза (рисунок 1.11).

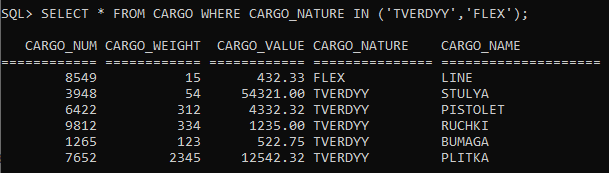


Рисунок 1.11 – Использование специальной функции IN

С помощью специальной функций BETWEEN в условии выведем только груз весом от 100 до 400 (рисунок 1.12).

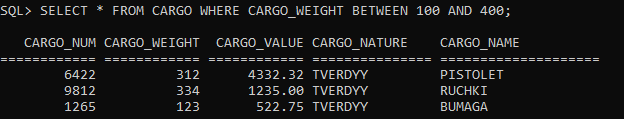


Рисунок 1.12 – Использование специальной функции BETWEEN

С помощью специальной функций LIKE в условии выведем только груз, наименование которого начинается с буквы Р (рисунок 1.13).

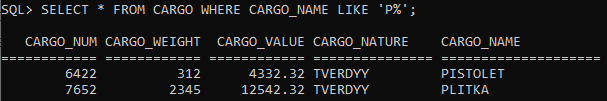


Рисунок 1.13 – Использование специальной функции LIKE

С помощью специальной функций IS NULL в условии выведем только груз, наименование которого равно NULL. Здесь не будет никакого вывода, потому что мы не имеем никаких значений NULL (рисунок 1.14).



Рисунок 1.14 – Использование специальной функции IS NULL

1.3.8 Продемонстрируем работу специальных функций с условием NOT (рисунки 1.15-1.18).

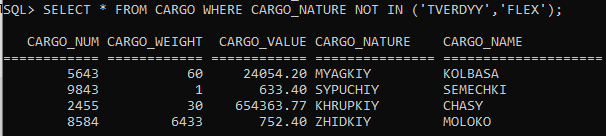


Рисунок 1.15 – Использование специальной функции IN с NOT

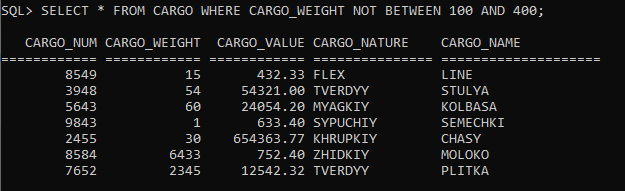


Рисунок 1.16 – Использование специальной функции BETWEEN с NOT

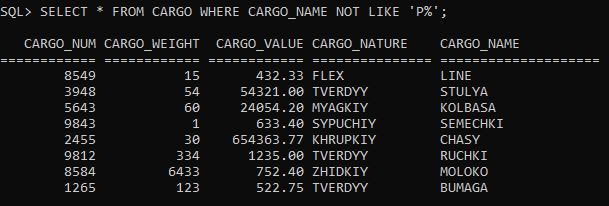


Рисунок 1.17 – Использование специальной функции LIKE с NOT

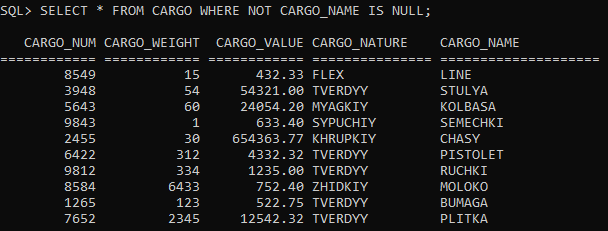


Рисунок 1.18 – Использование специальной функции IS NULL с NOT

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Объясните отличие архитектуры superserver и classic?

Основное различие между ними состоит в том, что Classic создает параллельный процесс для каждого присоединяемого пользователя, а SuperServer состоит из одного процесса, который обрабатывает запросы клиентов в разных нитях (threads) этого же процесса. Архитектура Classic считается более надежной, а SuperServer более производительной.

1. Какая информация указывается при создании базы данных?

Необходимо указать следующую информацию:

− имя файла (File Name)

– имя физического файла базы данных, а также указать путь сохранения файла на диске;

− имя пользователя (User Name)

– SYSDBA (SYStem DataBase Administrator);

− пароль (Password) – masterkey;

− размер страницы базы данных задается равным 4096 (одно из допустимых значений);

− Обязательно надо установить набор символов по умолчанию (Default character set) в UTF8 или WIN1251.

1. Базовое понятие многопользовательских систем «транзакция»?

Транзакция - это неделимая, с точки зрения воздействия на СУБД, последовательность операций манипулирования данными. Для пользователя транзакция выполняется по принципу "все или ничего", т.е. либо транзакция выполняется целиком и переводит базу данных из одного целостного состояния в другое целостное состояние, либо, если по каким-либо причинам, одно из действий транзакции невыполнимо, или произошло какое-либо нарушение работы системы, база данных возвращается в исходное состояние, которое было до начала транзакции (происходит откат транзакции).

В многопользовательских системах транзакции служат для обеспечения изолированной работы отдельных пользователей - пользователям, одновременно работающим с одной базой данных, кажется, что они работают как бы в однопользовательской системе и не мешают друг другу.

1. Модели данных. Реляционная модель данных?

Ядром любой БД является модель данных. Модель данных – это совокупность структуры данных и операций их обработки.

Бывают: иерархическая, сетевая, реляционная.

Реляционная модель данных - это способ рассмотрения данных, то есть предписание для способа представления данных (посредством таблиц) и для способа работы с таким представлением (посредством операторов). Она связана с тремя аспектами данных: структурой (объекты), целостностью и обработкой данных (операторы).

1. СУБД – назначение?

Основная задача СУБД - предоставить пользователю базы данных возможность работать с ней, не вникая в детали на уровне аппаратного обеспечения (пользователь более отстранен от этих деталей, чем прикладной программист, использующий среду программирования). Иными словами, СУБД позволяет конечному пользователю рассматривать базу данных как объект более высокого уровня по сравнению с аппаратным обеспечением, а также предоставляет в его распоряжение набор операций, выражаемых в терминах языка высокого уровня (например, набор операций, которые можно выполнять с помощью языка SQL)

1. Основные понятие реляционных БД: отношение, атрибут, кортеж, домен, схема отношения?

Отношение - это множество кортежей, соответствующих одной схеме отношения.

Атрибут (элемент данных) - наименьшая единица структуры данных. Обычно каждому элементу при описании базы данных присваивается уникальное имя. По этому имени к нему обращаются при обработке. Элемент данных также часто называют полем

Кортеж- это множество пар {имя атрибута - значение}, которое содержит одно вхождение каждого имени атрибута, принадлежащего схеме отношения. Кортеж - это набор именованных значений заданного типа.

Домен - тип данных, то есть множество допустимых значений.

Схема отношения - это именованное множество пар {имя атрибута, имя домена (или типа, если понятие домена не поддерживается)}.

1. Правила задания таблиц?
2. имя таблицы указывается после ключевого слова CREATE TABLE (если имя состоит из нескольких слов, то его следует заключить в одинарные кавычки);
3. далее в круглых скобках следуют имена столбцов таблицы (полей), после которых указывается [тип данных](https://labs-org.ru/mysql-1/#__mySQL), которому будет принадлежать поле;
4. не обязательно: затем указывается может ли поле содержать пустые значения (NULL — может быть пустым или NOT NULL — обязательно для заполнения);
5. поля отделяются запятыми.
6. Ключевой атрибут, первичный ключ?

Ключи или ключевой атрибут — атрибут (столбец) или набор атрибутов, который однозначно идентифицирует сущность/объект/таблицу в базе данных.

Первичный ключ — ключ, который используется для идентификации объекта.

1. Требования к вводу данных оператором INSERT INTO?

Инструкция INSERT INTO и INSERT используются для вставки новых строк в таблицу. Существует два способа использования инструкций:

Только значения: Первый метод предусматривает указание только значений данных, которые нужно вставить без имен столбцов.

INSERT INTO имя\_таблицы VALUES (значение1, значение2, значение3,...);

Имена столбцов и значения: При втором методе указываются имена столбцов и значения строк для вставки:

INSERT INTO имя\_таблицы (столбец1, столбец2, столбец3,..) VALUES (значение1, значение2, значение3,...);