Министерство образования Республики Беларусь  
Учреждение образования  
«Брестский государственный технический университет»  
Кафедра ИИТ

Лабораторная работа №3  
По дисциплине: «Проектирование баз знаний»

Выполнил:  
студент 3 курса  
группы ИИ-24  
Крейдич А.А.

Проверил:  
Козик И.Д.

Брест 2025

Цель работы:

1. Изучить материал, приведенный в “К лабораторной работе 3.doc”.

2. На основании логической модели (в соответствии с вариантом Лабораторной работы №1) создать и проверить на работоспособность запросы и фильтры для каждой из таблиц БД:

2.1. Реализовать запросы разных видов для каждой таблицы (команды DML SQL и DDL SQL): на создание таблицы, добавление/удаление/изменение информации, выборку с условием/параметром и т.д. (не менее 7 на каждую таблицу).

2.2. Реализовать фильтры для каждой таблицы по различным полям (не менее 7 на каждую таблицу).

**Вариант 8**

Ход работы:

Table 1 (Товары):

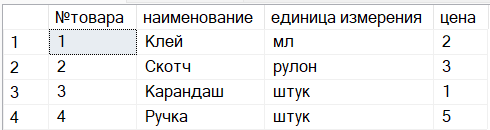


Table 2 (Чека):

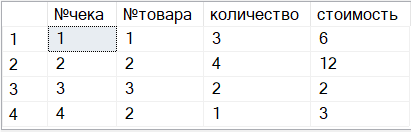


Table 3 (Продавец):



**Задание 1:**

**Таблица Товары:**

-- 1. Создание таблицы "Товары"

CREATE TABLE Товары (

"№товара" INTEGER PRIMARY KEY,

наименование VARCHAR(255),

"единица измерения" VARCHAR(255),

цена INTEGER

);

-- 2. Добавление данных в таблицу "Товары"

INSERT INTO Товары ("№товара", наименование, "единица измерения", цена)

VALUES

(1, 'Клей', 'мл', 2),

(2, 'Скотч', 'рулон', 3),

(3, 'Карандаш', 'штук', 1),

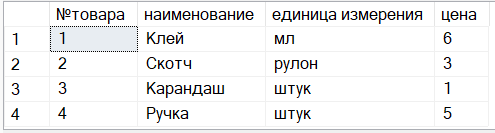
(4, 'Ручка', 'штук', 5);

-- 3. Изменение данных в таблице "Товары" (обновление цены товара)

UPDATE Товары

SET цена = 6

WHERE наименование = 'Клей';



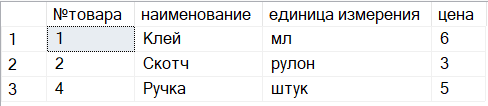
-- 4. Удаление записи из таблицы "Товары"

DELETE FROM Товары

WHERE "№товара" = 3;

-- 5. Выбор всех данных из таблицы "Товары"

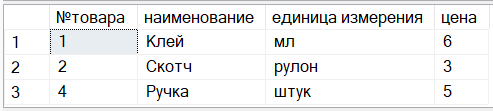
SELECT \* FROM Товары;



-- 6. Выбор товаров с ценой больше 2

SELECT \* FROM Товары

WHERE цена > 2;



-- 7. Подсчет количества товаров

SELECT COUNT(\*) AS Количество\_товаров FROM Товары;



**Таблица Чек:**

-- 1. Создание таблицы "Чек"

CREATE TABLE Чек (

"№чека" INTEGER PRIMARY KEY,

"№товара" INTEGER,

количество INTEGER,

стоимость INTEGER

);

-- 2. Добавление данных в таблицу "Чек"

INSERT INTO Чек ("№чека", "№товара", количество, стоимость)

VALUES

(1, 1, 3, 6),

(2, 2, 4, 12),

(3, 3, 2, 2),

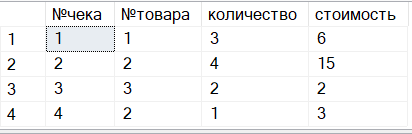
(4, 2, 1, 3);

-- 3. Изменение данных в таблице "Чек" (обновление стоимости)

UPDATE Чек

SET стоимость = 15

WHERE "№чека" = 2;



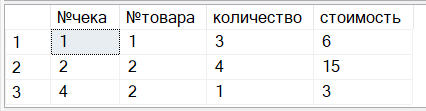
-- 4. Удаление записи из таблицы "Чек"

DELETE FROM Чек

WHERE "№чека" = 3;

-- 5. Выбор всех данных из таблицы "Чек"

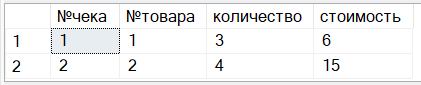
SELECT \* FROM Чек;



-- 6. Выбор чеков с количеством больше 2

SELECT \* FROM Чек

WHERE количество > 2;



-- 7. Подсчет общей стоимости всех товаров

SELECT SUM(стоимость) AS Общая\_стоимость FROM Чек;



**Таблица Продавец:**

-- 1. Создание таблицы "Продавец"

CREATE TABLE Продавец (

"№чека" INTEGER PRIMARY KEY,

"дата покупки" DATE,

"№кассы" INTEGER,

продавец VARCHAR(255)

);

-- 2. Добавление данных в таблицу "Продавец"

INSERT INTO Продавец ("№чека", "дата покупки", "№кассы", продавец)

VALUES

(1, '2023-03-15', 1, 'Иванов Иван'),

(2, '2023-07-20', 2, 'Петров Петр'),

(3, '2023-11-10', 3, 'Сидоров Сидор'),

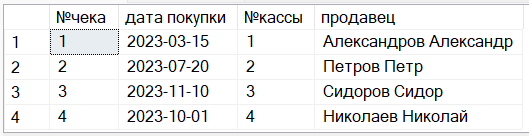
(4, '2023-10-01', 4, 'Николаев Николай');

-- 3. Изменение данных в таблице "Продавец" (обновление продавца)

UPDATE Продавец

SET продавец = 'Александров Александр'

WHERE "№чека" = 1;



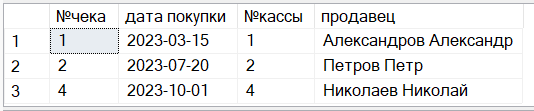
-- 4. Удаление записи из таблицы "Продавец"

DELETE FROM Продавец

WHERE "№чека" = 3;

-- 5. Выбор всех данных из таблицы "Продавец"

SELECT \* FROM Продавец;



-- 6. Выбор продавцов по номеру кассы

SELECT \* FROM Продавец

WHERE "№кассы" = 1;



-- 7. Подсчет количества продаж

SELECT COUNT(\*) AS Количество\_продаж FROM Продавец;



**Задание 2:**

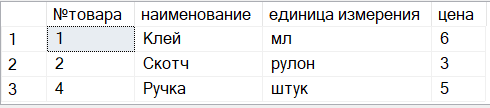
**Таблица Товары:**

-- Фильтры для таблицы "Товары":

-- 1. Фильтр по цене

SELECT \* FROM Товары

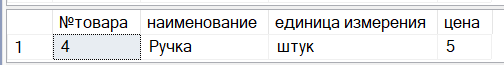
WHERE цена > 2;



-- 2. Фильтр по единице измерения

SELECT \* FROM Товары

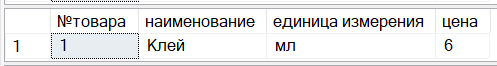
WHERE "единица измерения" = 'штук';



-- 3. Фильтр по наименованию (частичное совпадение)

SELECT \* FROM Товары

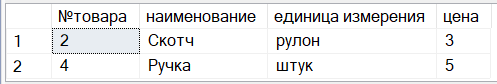
WHERE наименование LIKE '%Клей%';



-- 4. Фильтр с использованием оператора BETWEEN (для цены)

SELECT \* FROM Товары

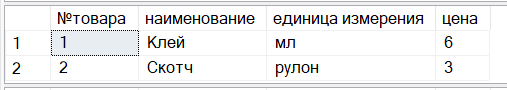
WHERE цена BETWEEN 1 AND 5;



-- 5. Фильтр с использованием оператора IN (для наименования)

SELECT \* FROM Товары

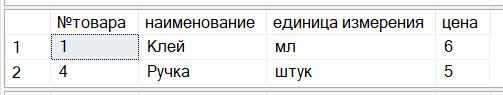
WHERE наименование IN ('Клей', 'Скотч');



-- 6. Фильтр с использованием оператора <> (для исключения значений)

SELECT \* FROM Товары

WHERE цена <> 3;



-- 7. Фильтр с использованием логического оператора AND (для комбинации условий)

SELECT \* FROM Товары

WHERE "единица измерения" = 'штук' AND цена > 2;



**Таблица Чек:**

-- Фильтры для таблицы "Чек":

-- 1. Фильтр по количеству

SELECT \* FROM Чек

WHERE количество > 3;



-- 2. Фильтр по стоимости

SELECT \* FROM Чек

WHERE стоимость = 6;



-- 3. Фильтр по номеру товара

SELECT \* FROM Чек

WHERE "№товара" = 1;



-- 4. Фильтр с использованием оператора BETWEEN (для стоимости)

SELECT \* FROM Чек

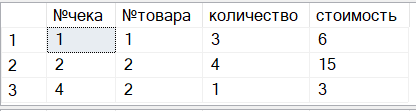
WHERE стоимость BETWEEN 5 AND 10;



-- 5. Фильтр с использованием оператора IN (для номера товара)

SELECT \* FROM Чек

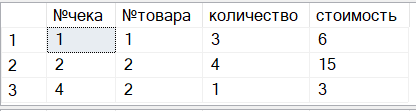
WHERE "№товара" IN (1, 2);



-- 6. Фильтр с использованием оператора <> (для исключения значений)

SELECT \* FROM Чек

WHERE количество <> 2;



-- 7. Фильтр с использованием логического оператора AND (для комбинации условий)

SELECT \* FROM Чек

WHERE количество > 1 AND стоимость > 5;



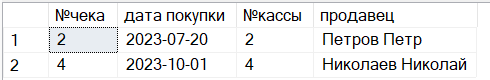
**Таблица Продавец:**

-- Фильтры для таблицы "Продавец":

-- 1. Фильтр по дате покупки

SELECT \* FROM Продавец

WHERE "дата покупки" > '2023-07-01';



-- 2. Фильтр по номеру кассы

SELECT \* FROM Продавец

WHERE "№кассы" = 1;



-- 3. Фильтр по имени продавца

SELECT \* FROM Продавец

WHERE продавец = 'Иванов Иван';



-- 4. Фильтр с использованием оператора BETWEEN (для даты)

SELECT \* FROM Продавец

WHERE "дата покупки" BETWEEN '2023-01-01' AND '2023-06-30';



-- 5. Фильтр с использованием оператора IN (для продавца)

SELECT \* FROM Продавец

WHERE продавец IN ('Иванов Иван', 'Петров Петр');



-- 6. Фильтр с использованием оператора <> (для исключения значений)

SELECT \* FROM Продавец

WHERE продавец <> 'Сидоров Сидор';

-- 7.

Фильтр с использованием логического оператора AND (для комбинации условий)

SELECT \* FROM Продавец

WHERE "№кассы" = 2 AND продавец = 'Петров Петр';



**Вывод:** в ходе данной лабораторной работы мной были изучены способы реализации запросов и фильтров для баз данных в SQLite.