Автономная некоммерческая организация

профессионального образования

Пермский гуманитарно-технологический колледж

Кафедра математических и

естественно-научных дисциплин

**КУРСОВАЯ РАБОТА**

по дисциплине

«Технология разработки и защиты БД»

на тему:

«Разработка базы данных продуктового магазина»

Выполнил студент:

Иванов Иван Иванович

Группа: ПКС-9-4

Курс: 4

Руководитель:

Ст. преподаватель

Могильникова Наталья Сергеевна

Пермь, 2023

**СОДЕРЖАНИЕ**

**ВВЕДЕНИЕ**

За последние 30 лет в области теории систем баз данных была проведен ряд исключительно продуктивных исследований. Полученные результаты вполне можно считать наиболее важным достижением информатики за этот период. Базы данных стали основой информационных систем и в корне изменили методы работы многих организаций. В частности, в последние годы развитие технологии баз данных привело к созданию весьма мощных и удобных в эксплуатации систем. Благодаря этому системы баз данных стали доступными широкому кругу пользователей. Но, к сожалению, кажущаяся простота таких систем способствовала тому, что пользователи стали самостоятельно создавать базы данных и приложения, не имея достаточных знаний о методах проектирования эффективно работающих систем, что часто приводило к непроизводительным затратам ресурсов и некачественным результатам. Вызванная этим неудовлетворенность пользователей стала причиной возникновения известного "кризиса программного обеспечения", или так называемой "депрессии программного обеспечения", последствия которой не устранены и поныне.

Реляционная база данных – это совокупность отношений, содержащих всю информацию, которая должна храниться в БД. Пользователи воспринимают такую базу данных как совокупность таблиц, связанных между собой.

Основными задачами системы SQL Server являются: организация одновременного доступа к данным большого количества пользователей; манипуляция информацией, хранимой в БД и др.

SQL Server поддерживает реляционную модель данных и выполняет функции создания объектов БД (таблиц, индексов, представлений и т.д.), осуществляет проверку целостности БД и отвечает за безопасность данных в системе.

Доступ пользователя к данным обычно осуществляется с компьютера рабочей станции. При этом создаются соответствующие приложения (например, в средах Visual Basic, Delphi и др.), которые позволяют выполнять операции над данными.

Задачи, связанные с администрированием БД системы SQL Server, обращение к системе удобно выполнять непосредственно с компьютера-сервера. В SQL Server для манипулирования данными используется язык Transact SQL, переработанной компанией Microsoft версией языка SQL.

Система SQL Server сохраняет создаваемые объекты в соответствующих файлах на дисках компьютера сервера. При этом для БД создаются специальные таблицы, в которых хранится информация о различных элементах базы данных: индексах, таблицах, пользователях и т.д.

Файлы БД сохраняются с расширением MDF, а системные файлы с расширением LDF.

Основные операции, связанные с управлением работой SQL сервера, осуществляются с помощью ряда утилит, входящих в состав системы: SQL Server Books Online – представляет пользователю справочную поддержку; SQL Server Query Analyzer – предоставляет пользователю возможность выполнения операторов Transact SQL в БД SQL Server; ISQL – предоставляет возможность выполнять команды Transact SQL в режиме командной строки; SQL Server Service Manager – предоставляет возможность запуска, остановки и временной приостановки работы SQL Server. Запуск данной утилиты возможен только на компьютере сервере; SQL Server Enterprise Manager – позволяет выполнять все основные операции администрирования SQL Server. Данная утилита позволяет осуществлять запуск всех утилит и приложений, входящих в состав SQL Server. Данная утилита может запускаться и на компьютере пользователя и на сервере.

1. **Инфологическая модель базы данных**

Цель инфологического моделирования – обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных. Поэтому инфологическую модель данных пытаются строить по аналогии с естественным языком. Основными конструктивными элементами инфологических моделей являются сущности, связи между ними и их свойства (атрибуты).

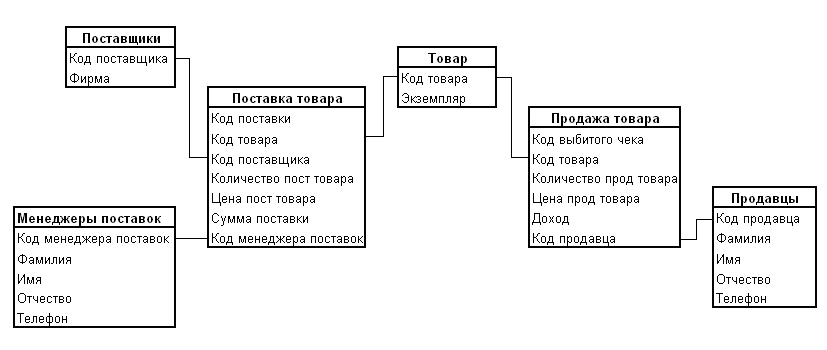
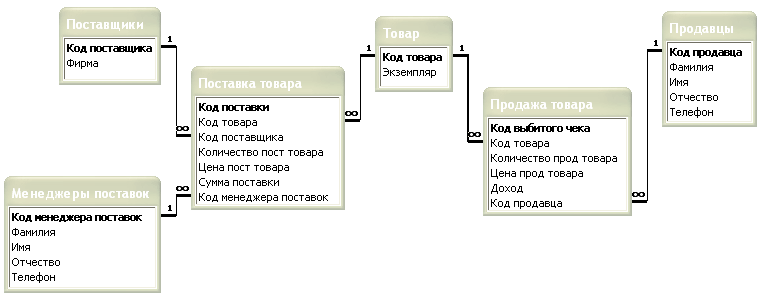


Рисунок 1 - Инфологическая модель

1. **Логическая модель базы данных**

Шесть сущностей находятся в третьей нормальной форме, т. е. все атрибуты сущностей содержат атомарные значения (значения в домене не являются ни списками, ни множествами простых или сложных значений); каждый неключевой атрибут полностью зависит от первичного ключа; никакой неключевой атрибут не зависит от другого неключевого атрибута.

Рисунок 2 - Логическая модель

1. **Физическая модель базы данных**

Создадим базу данных «Магазин продуктов»:

create database Magazin\_Productov

Затем создадим таблицы:

«Товар»

create table Tovar (codtovara int not null primary key, ekzemplar char(30) not null)

Таблица №1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибуты | Тип | Длина | Ключ | Описание |
| codtovara | int | 4 | primary | Код товара |
| ekzemplar | char | 30 |  | Наименование товара |

«Поставщики»

create table Postavsiki (codpostavsika int not null primary key, firma char(20) not null)

Таблица №2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибуты | Тип | Длина | Ключ | Описание |
| codpostavsika | int | 4 | primary | Код поставщика |
| firma | char | 20 |  | Наименование фирмы |

«Менеджеры поставок»

create table Manageri\_postavok (cod\_meng\_postav int not null primary key, familia char(20) not null, imya char(10) not null, otchestvo char(20) not null, telefon int not null)

Таблица №3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибуты | Тип | Длина | Ключ | Описание |
| cod\_meng\_postav | int | 4 | primary | Код менеджера поставок |
| familia | char | 20 |  | Фамилия менеджера поставок |
| imya | char | 10 |  | Имя менеджера поставок |
| otchestvo | char | 20 |  | Отчество менеджера поставок |
| telefon | int | 4 |  | Номер телефона менеджера поставок |

«Поставка товара»

create table Postavka\_tovara (codpostavki int not null primary key, codtovara int not null references Tovar, codpostavsika int not null references Postavsiki, colich\_post\_tov int not null, cena\_post\_tov money not null, symmapostavki money not null, cod\_meng\_postav int not null references Manageri\_postavok)

Таблица №4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибуты | Тип | Длина | Ключ | Описание |
| codpostavki | int | 4 | primary | Код поставки |
| codtovara | int | 4 |  | Код товара |
| codpostavsika | int | 4 |  | Код поставщика |
| colich\_post\_tov | int | 4 |  | Количество поставляемого товара |
| cena\_post\_tov | money | 8 |  | Цена поставляемого товара |
| symmapostavki | money | 8 |  | Сумма поставки |
| cod\_meng\_postav | int | 4 |  | Код менеджера поставки |

«Продавцы»

create table Prodavci (codprodavca int not null primary key, familia char(20) not null, imya char(10) not null, otchestvo char(20) not null, telefon int not null)

Таблица №6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибуты | Тип | Длина | Ключ | Описание |
| codprodavca | int | 4 | primary | Код продавца |
| familia | char | 20 |  | Фамилия продавца |
| imya | char | 10 |  | Имя продавца |
| otchestvo | char | 20 |  | Отчество продавца |
| telefon | int | 4 |  | Номер телефона продавца |

«Продажа товара»

create table Prodaza\_tovara (cod\_vibitogo\_cheka int not null primary key, codtovara int not null references Tovar, colich\_prod\_tov int not null, cena\_prod\_tov money not null, dohod money not null, codprodavca int not null references Prodavci)

Таблица №7

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибуты | Тип | Длина | Ключ | Описание |
| cod\_vibitogo\_cheka | int | 4 | primary | Код выбитого чека |
| codtovara | int | 4 |  | Код товара |
| CodFirmaClient | int | 4 |  | Код фирмы-клиента |
| colich\_prod\_tov | int | 4 |  | Количество проданного товара |
| cena\_prod\_tov | money | 8 |  | Цена проданного товара |
| dohod | money | 8 |  | Доход |
| codprodavca | int | 4 |  | Код продавца |

Заполним таблицы следующими SQL запросами:

Таблица «Товар»

insert into Tovar values (1, 'Сыр копченый')

insert into Tovar values (2, 'Молоко')

insert into Tovar values (3, 'Творог')

insert into Tovar values (4, 'Сметана')

insert into Tovar values (5, 'Йогурт')

insert into Tovar values (6, 'Сыр плавленый')

insert into Tovar values (7, 'Кефир')

insert into Tovar values (8, 'Ряженка')

insert into Tovar values (9, 'Сыр российский')

insert into Tovar values (10, 'Снежок')

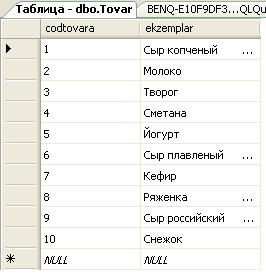


Рисунок 3 - Таблица «Товар»

«Поставщики»

insert into Postavsiki values (11, 'ОАО "Молоко"')

insert into Postavsiki values (12, 'ОАО "МолПрод"')

insert into Postavsiki values (13, 'ОАО "Снежинка"')

insert into Postavsiki values (14, 'ОАО "Меркурий+"')

insert into Postavsiki values (15, 'ОАО "Молочный дом"')

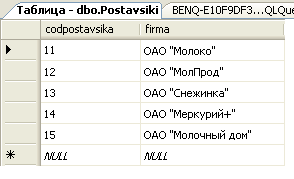


Рисунок 4 - Таблица «Поставщики»

«Менеджеры поставок»

insert into Manageri\_postavok values (21, 'Банин', 'Александр', 'Абрамович', 22431)

insert into Manageri\_postavok values (22, 'Сидорова', 'Раиса', 'Николаевна', 22543)

insert into Manageri\_postavok values (23, 'Викенов', 'Дмитрий', 'Александрович', 22564)

insert into Manageri\_postavok values (24, 'Воробьев', 'Николай', 'Иванович', 22468)

insert into Manageri\_postavok values (25, 'Николаева', 'Юлия', 'Михайловна', 22541)

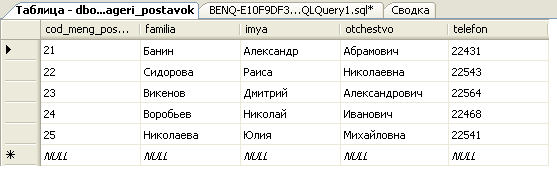


Рисунок 5 - Таблица «Менеджеры поставок»

«Поставка товара»

insert into Postavka\_tovara values (31, 1, 15, 210, 190, 0, 22)

insert into Postavka\_tovara values (32, 2, 12, 280, 25, 0, 25)

insert into Postavka\_tovara values (33, 3, 14, 190, 80, 0, 23)

insert into Postavka\_tovara values (34, 4, 13, 200, 40, 0, 21)

insert into Postavka\_tovara values (35, 5, 11, 270, 30, 0, 21)

insert into Postavka\_tovara values (36, 6, 13, 220, 15, 0, 23)

insert into Postavka\_tovara values (37, 7, 11, 170, 17, 0, 24)

insert into Postavka\_tovara values (38, 8, 14, 200, 16, 0, 25)

insert into Postavka\_tovara values (39, 9, 12, 160, 165, 0, 24)

insert into Postavka\_tovara values (40, 10, 13, 240, 20, 0, 22)

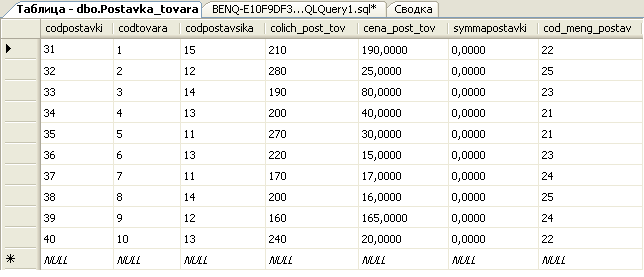


Рисунок 6 - Таблица «Поставка товара»

«Продавцы»

insert into Prodavci values (41, 'Краснов', 'Алексей', 'Петрович', 22235)

insert into Prodavci values (42, 'Вдовин', 'Андрей', 'Константинович', 22371)

insert into Prodavci values (43, 'Михайлова', 'Ольга', 'Ивановна', 22364)

insert into Prodavci values (44, 'Демидова', 'Анна', 'Викторовна', 22216)

insert into Prodavci values (45, 'Семенов', 'Игорь', 'Алексеевич', 22353)

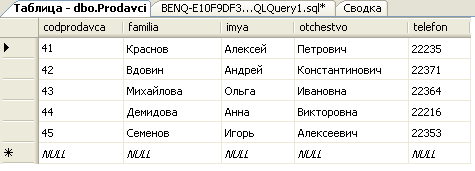


Рисунок 7 - Таблица «Продавцы»

«Продажа товара»

insert into Prodaza\_tovara values (51, 1, 177, 210, 0, 44)

insert into Prodaza\_tovara values (52, 2, 265, 30, 0, 45)

insert into Prodaza\_tovara values (53, 3, 146, 100, 0, 43)

insert into Prodaza\_tovara values (54, 4, 180, 60, 0, 45)

insert into Prodaza\_tovara values (55, 5, 119, 40, 0, 41)

insert into Prodaza\_tovara values (56, 6, 110, 25, 0, 43)

insert into Prodaza\_tovara values (57, 7, 170, 30, 0, 42)

insert into Prodaza\_tovara values (58, 8, 190, 27, 0, 43)

insert into Prodaza\_tovara values (59, 9, 150, 180, 0, 44)

insert into Prodaza\_tovara values (60, 10, 120, 30, 0, 41)

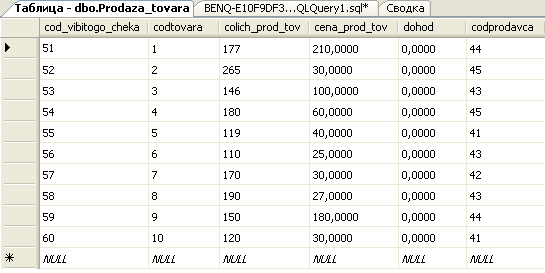


Рисунок 8 - Таблица «Продажа товара»

**Здесь добавить EER-диаграмму**

1. **Создание запросов, представлений, процедур и триггеров**

Создадим представления:

«Поставка»

create view Postavka as select Postavka\_tovara.codpostavki, Tovar.ekzemplar, Postavsiki.firma,Postavka\_tovara.colich\_post\_tov, Postavka\_tovara.cena\_post\_tov, Postavka\_tovara.symmapostavki, Manageri\_postavok.familia, Manageri\_postavok.imya from Postavka\_tovara inner join Postavsiki on Postavka\_tovara.codpostavsika = Postavsiki.codpostavsika inner join Tovar on Postavka\_tovara.codtovara = Tovar.codtovara inner join Manageri\_postavok on Manageri\_postavok.cod\_meng\_postav = Postavka\_tovara.cod\_meng\_postav

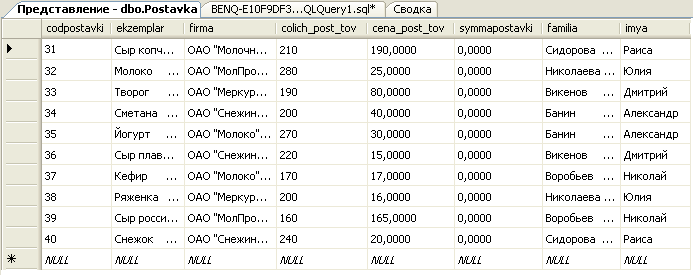


Рисунок 9 - Представление «Поставка»

«Продажа»

create view Prodaza as select Prodaza\_tovara.cod\_vibitogo\_cheka, Tovar.ekzemplar, Prodaza\_tovara.colich\_prod\_tov, Prodavci.familia, Prodavci.imya, Prodavci.telefon from Prodaza\_tovara inner join Tovar on Prodaza\_tovara.codtovara = Tovar.codtovara inner join Prodavci on Prodaza\_tovara.codprodavca = Prodavci.codprodavca

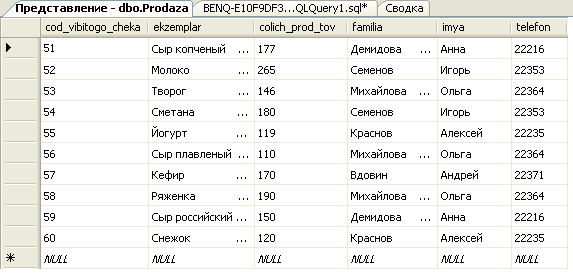


Рисунок 10 - Представление «Продажа»

Создадим несколько запросов:

1. Товар, выбранный по коду в диапазоне от 4 до 8

select \* from Tovar where codtovara between 4 and 8

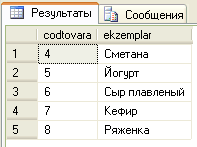


Рисунок 14 – Запрос, представляющий информацию о товарах

2. Товар, выбранный по коду из указанного множества

select \* from Tovar where codtovara in (2, 5, 6, 11, 12, 4)

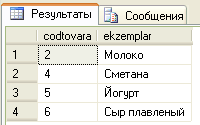


Рисунок 15 - Запрос 2

3. Выбор записей из таблицы, являющейся соединением двух таблиц

select Tovar.ekzemplar, Prodaza\_tovara.colich\_prod\_tov, Prodaza\_tovara.cena\_prod\_tov, Prodaza\_tovara.Dohod from Tovar, Prodaza\_tovara where Tovar.codtovara = Prodaza\_tovara.codtovara



Рисунок 16 - Запрос 3

4. Подсчет количества выбитых чеков

select count (\*) as 'Количество выбитых чеков' from Prodaza\_tovara

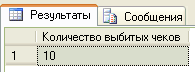


Рисунок 17 - Запрос 4

5. Минимальное количество поставленного товара

select min (colich\_post\_tov) as 'Минимальное количество поставленного товара' from Postavka\_tovara

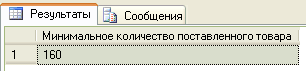


Рисунок 18 - Запрос 5

6. Наименования товара по алфавиту

select \* from Tovar order by ekzemplar ASC

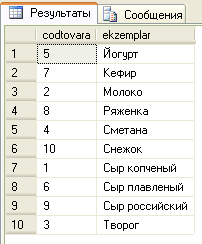


Рисунок 19 - Запрос 6

7. Количество проданного товара ниже среднего количества продаж

select \* from Prodaza where colich\_prod\_tov < (select avg (colich\_prod\_tov) from Prodaza)

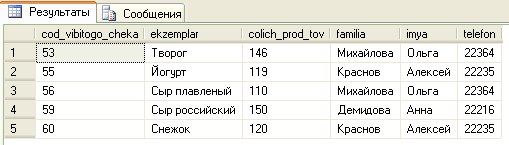


Рисунок 20 - Запрос 7

Создадим триггер, который удаляет запись в таблице «Поставка товара», соответствующую одиннадцатому коду товара.

create trigger Udalen on Tovar for delete as if @@rowcount = 1 begin declare @x int select @x = Tovar.codtovara from Tovar delete Postavka\_tovara.codtovara where Tovar.codtovara = Postavka\_tovara.codtovara if exists (select \* from Postavka\_tovara where codtovara = @x) delete from Postavka\_tovara where codtovara = 11 end

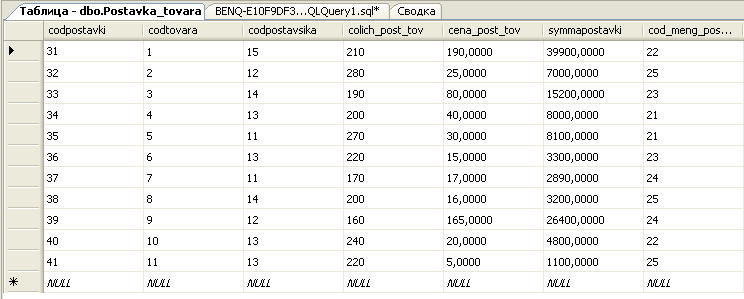


Рисунок 22 – Триггер 1

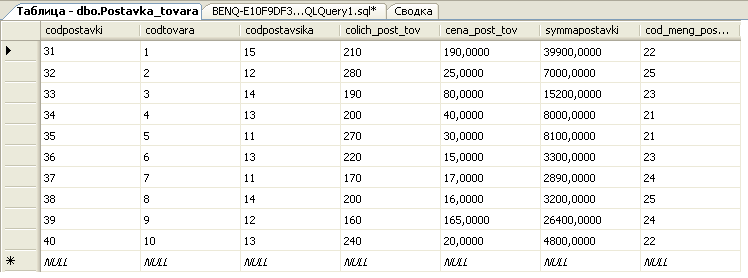


Рисунок 23 – Триггер 1 (продолжение)

Создадим триггер на событие обновления одной из таблиц:

Рисунок 24 – Триггер 2

Создадим триггер на событие удаления данных из таблицы:

Рисунок 25 – Триггер 3

Создадим процедуру с использованием цикла:

Рисунок 26 – Процедура 1

Создадим процедуру с использованием входного и выходного параметра:

Рисунок 27 – Процедура 2

**ЗАКЛЮЧЕНИЕ**

В данной курсовой работе с помощью СУБД mySQL было рассмотрено создание шести сущностей с определенными атрибутами, представлений, основанных на соединении столбцов нескольких таблиц; создание связей между таблицами, а также создание процедур и запросов.

**СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**