МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»

(САМАРСКИЙ УНИВЕРСИТЕТ)

Институт информатики и кибернетики

Кафедра информационных систем и технологий

Отчёт по лабораторной работе №2

по курсу «Базы данных»

Вариант 47

Выполнил:  
Алеев И.И.

(6304-090301D)

Проверила:  
Чигарина Е.И.

Самара 2023

Логическая схема:

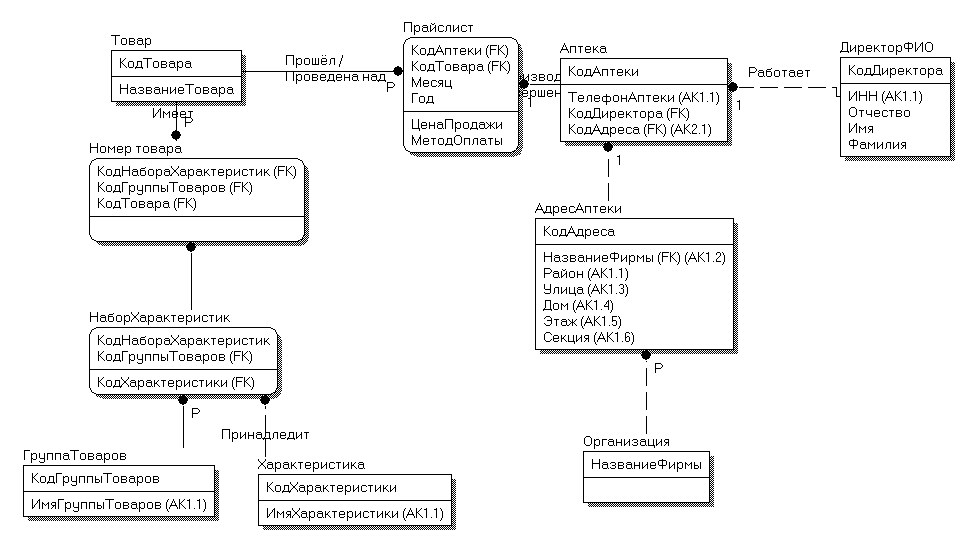


Рисунок 1 – Логическая схема базы данных

1. Используя язык Transact SQL и логическую модель базы данных в соответствии с индивидуальным заданием предыдущей лабораторной работы создать схему взаимосвязанных отношений.

CREATE TABLE Характеристика(

КодХарактеристики int primary key,

НазваниеХарактеристики varchar(20) unique

);

CREATE TABLE ГруппаТоваров(

КодГруппыТоваров int primary key,

ИмяГруппыТоваров varchar(18) not null

);

CREATE TABLE Товар(

КодТовара int primary key,

НазваниеТовара varchar(20) not null

);

CREATE TABLE Организация(

НазваниеФирмы varchar(18) primary key

);

CREATE TABLE ДиректорФИО(

КодДиректора int primary key,

ИНН char(12) not null,

Фамилия varchar(18) not null,

Имя varchar(18) not null,

Отчество varchar(18) null

);

CREATE TABLE НаборХарактеристик(

КодНабораХарактеристик int not null,

КодГруппыТоваров int

references ГруппаТоваров(КодГруппыТоваров),

primary key (

КодНабораХарактеристик,

КодГруппыТоваров

),

КодХарактеристики int

references Характеристика(КодХарактеристики)

);

CREATE TABLE АдресАптеки (

КодАдреса int primary key,

Район varchar(18) not null,

Улица varchar(18) not null,

Дом varchar(5) not null,

Этаж int null,

Секция varchar(5) null,

unique(

Район,

Улица,

Дом,

Этаж,

Секция

),

НазваниеФирмы varchar(18)

references Организация(НазваниеФирмы)

);

CREATE TABLE Номер\_товара(

КодНабораХарактеристик int not null,

КодГруппыТоваров int not null,

КодТовара int

references Товар(КодТовара),

foreign key (

КодНабораХарактеристик,

КодГруппыТоваров

)

references НаборХарактеристик(

КодНабораХарактеристик,

КодГруппыТоваров

),

primary key (

КодНабораХарактеристик,

КодГруппыТоваров,

КодТовара

)

);

CREATE TABLE Аптека(

КодАптеки int primary key,

ТелефонАптеки varchar(12),

КодДиректора int

references ДиректорФИО(КодДиректора) unique,

КодАдреса int

references АдресАптеки(КодАдреса) unique

);

CREATE TABLE Прайслист(

КодАптеки int

references Аптека(КодАптеки),

КодТовара int

references Товар(КодТовара),

Месяц int not null,

Год int not null,

ЦенаПродажи money not null,

МетодОплаты varchar(20) not null

primary key (

КодАптеки,

КодТовара,

Месяц,

Год

)

);

1. Построить диаграмму данных. Проверить соответствие физической схемы логической модели базы данных.

Физическая схема:

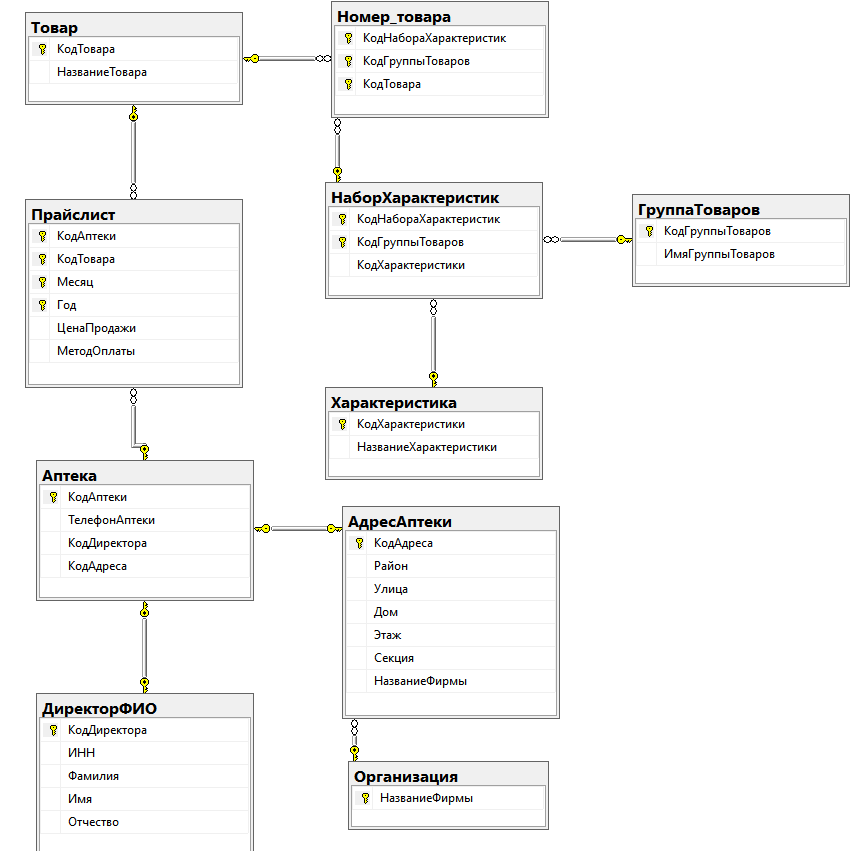


Рисунок 2 – Диаграмма базы данных

1. Заполнить записями каждое отношение, используя команду добавления записей.

insert into Товар(КодТовара,НазваниеТовара)

values

(11111,'Фенибут'),

(11112,'Феназипам'),

(11113,'Ношпа'),

(11114,'Анальгин'),

(11115,'Кетарол'),

(11116,'Кеторолак'),

(11117,'Спасзумугель'),

(11118,'Некст'),

(11119,'Йод');

insert into ГруппаТоваров(КодГруппыТоваров,ИмяГруппыТоваров)

values

(22221,'Спазмальгетики'),

(22222,'От давления'),

(22223,'Профилактика'),

(22224,'Обезболивающие');

insert into Характеристика(КодХарактеристики,НазваниеХарактеристики)

values

(33331,'Дозировка'),

(33332,'Формат упаковки'),

(33333,'Количество'),

(33334,'Рецептурное'),

(33335,'Показания');

insert into Организация(НазваниеФирмы)

values

('Аптека-Вита'),

('Аптека+'),

('Вита-Экспресс');

insert into ДиректорФИО(КодДиректора,ИНН,Фамилия,Имя,Отчество)

values

(44441,'7742332352','Прохорович','Роман','Дмитриевич'),

(44442,'2123013902','Андреева','Виктория','Владимировна'),

(44443,'21323032423','Медведева','Лея','Ивановна'),

(44444,'2423897392','Борисова','Глеб','Львович'),

(44445,'2243013243','Игнатов','Милана','Артёмович'),

(44446,'53323032423','Гуленко','Милана','Михайловна');

insert into АдресАптеки(КодАдреса,НазваниеФирмы, Район, Улица, Дом, Этаж, Секция)

values

(55551,'Аптека-Вита', 'Владимирский ' ,'Трактовая', '23', '1', 'а4'),

(55552,'Аптека-Вита', 'Владимирский ' ,'Комсомольская', '234', '-1', '55'),

(55553,'Аптека-Вита', 'Тольяттинский', 'Вокзальная', '42', '2', '0'),

(55554,'Аптека+' , 'Тольяттинский', 'Московская', '112', '1', ''),

(55555,'Вита-Экспресс', 'Можайский', 'Весенняя', '86', '1', '11'),

(55556,'Вита-Экспресс', 'Можайский', 'Юбилейная', '95а', '1', '');

insert into НаборХарактеристик(КодНабораХарактеристик,КодГруппыТоваров,КодХарактеристики)

values

(666601,22221,33331),

(666602,22221,33332),

(666603,22221,33333),

(666604,22221,33334),

(666605,22221,33335),

(666611,22222,33331),

(666612,22222,33332),

(666613,22222,33333),

(666614,22222,33334),

(666615,22222,33335),

(666621,22224,33331),

(666622,22224,33332),

(666623,22224,33333),

(666624,22224,33334),

(666625,22224,33335),

(666631,22223,33331),

(666632,22223,33332),

(666633,22223,33333),

(666634,22223,33334),

(666635,22223,33335);

insert into Аптека(КодАптеки, ТелефонАптеки, КодДиректора, КодАдреса)

values

(77771,'88005553535',44441,55551),

(77772,'88005553214',44442,55552),

(77773,'88005214212',44443,55553),

(77774,'88002029035',44444,55554),

(77775,'88971278920',44445,55555),

(77776,'88923902220',44446,55556);

insert into Прайслист(КодАптеки,КодТовара,Месяц,Год,ЦенаПродажи,МетодОплаты)

values

(77771,11111,1, 2022, 2000, 'Наличные'),

(77771,11112,2, 2020, 1299, 'Расчёт'),

(77771,11113,3, 2021, 1332, 'Наличные'),

(77772,11114,1, 2019, 100, 'Бартер'),

(77772,11115,5, 2018, 200, 'СБП'),

(77773,11116,4, 2020, 20, 'Наличные'),

(77774,11117,8, 2022, 1082, 'Бесконтактная'),

(77775,11111,6, 2022, 1232, 'Наличные'),

(77776,11112,11,2017, 32, 'Наличные');

insert into Номер\_товара(КодНабораХарактеристик,КодГруппыТоваров,КодТовара)

values

(666601,22221,11111),

(666602,22221,11112),

(666603,22221,11113),

(666604,22221,11114),

(666605,22221,11115),

(666611,22222,11111),

(666612,22222,11112),

(666613,22222,11116),

(666614,22222,11116),

(666615,22222,11117),

(666621,22224,11111),

(666622,22224,11112),

(666623,22224,11117),

(666624,22224,11113),

(666625,22224,11114),

(666631,22223,11111),

(666632,22223,11112),

(666633,22223,11114),

(666634,22223,11117),

(666635,22223,11115);

1. Выполнить операции модификации записей, используя подзапрос. Выполнить изменение одновременно нескольких полей.

* Директор аптеки изменил номер телефона отделения:

update Аптека set ТелефонАптеки='88462074078' where КодДиректора in

(

select top(1) КодДиректора from ДиректорФИО where Фамилия = 'Прохорович'

);

* Директор изменил своё имя и фамилию

update ДиректорФИО set Фамилия = 'Куценко', Имя = 'Гоша' where ИНН = '7742332352';

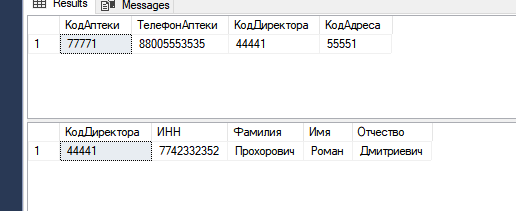


Рисунок 3 – До изменения

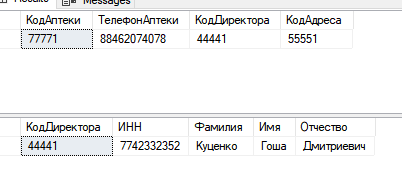


Рисунок 4 – После изменения

1. Удаление

* Выполнить удаление записей, удовлетворяющих условию.

Удалить старые товары из прайслиста

delete from Прайслист where Год < 2020;

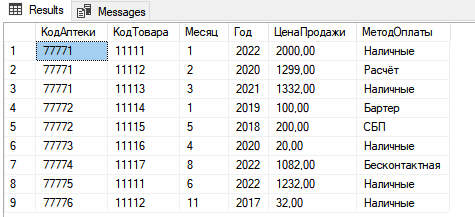


Рисунок 5 – До удаления

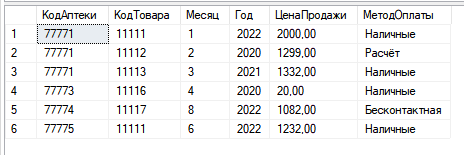


Рисунок 6 – После удаления

* Выполнить удаление указанного количества записей.

Удалить 3 первые позиции в прайслисте

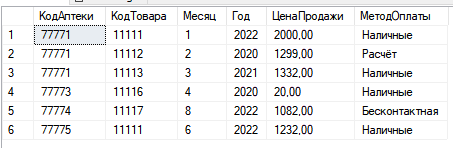


Рисунок 7 – До удаления

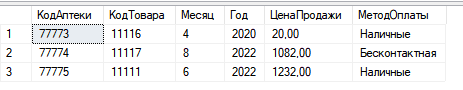


Рисунок 8 – После удаления

6. Создать следующие запросы:

1. Создать и реализовать запросы, используя операторы IN, LIKE, BETWEEN, AND, OR. Выполнить переименование заголовков столбцов в результате выборки.

IN OR BETWEEN

* Показать позиции в прайслисте с ценой в диапазоне между 500 и 1000 или с наличной оплатой

select Месяц as 'Месяц поступления', Год as 'Год поступления', ЦенаПродажи as 'Цена (Р)', МетодОплаты as 'Способ оплаты' from Прайслист where МетодОплаты in ('Наличные') or ЦенаПродажи between 500 and 10000 ;

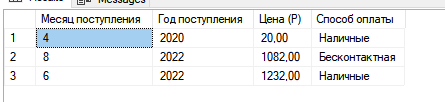


Рисунок 9 – Результат SELECT

LIKE AND

* Показать адреса аптек, имена которых схожи с «аптека вита» в тольяттинском районе

select НазваниеФирмы as 'Название', Район,Улица, Дом, Этаж, Секция from АдресАптеки where НазваниеФирмы like 'Аптека\_Вита' and Район = 'Тольяттинский';

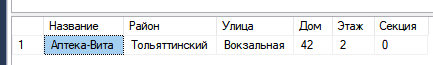


Рисунок 10 – Результат SELECT

1. Создать и реализовать запрос, реализующий сортировку по нескольким полям с различным видом сортировки.

* Вывести сначала новые позиции в прайслисте

select \* from Прайслист order by Год desc, Месяц desc;

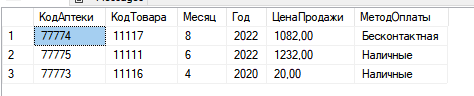


Рисунок 11 – Результат SELECT

* Вывести сначала дешёвые позиции в прайслисте

select \* from Прайслист order by ЦенаПродажи asc;

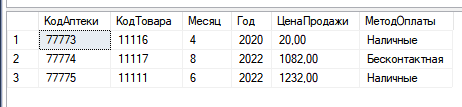


Рисунок 12 – Результат SELECT

1. Создать и реализовать запрос, используя группировку с функцией агрегирования. Создать и реализовать запрос, используя одновременно с группировкой ограничение HAVING.

* Сгруппировать позиции в прайслисте по методам оплаты для суммы дороже 300 р

select count(\*) as 'Количество', sum(ЦенаПродажи) as 'Сумма',МетодОплаты from Прайслист group by МетодОплаты having sum(ЦенаПродажи) > 300;

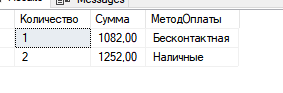


Рисунок 13 – Результат SELECT

1. Создать и реализовать запросы, используя все виды соединения таблиц.

Правое внешнее соединение

select \* from Прайслист

right join Товар ON Прайслист.КодТовара = Товар.КодТовара;

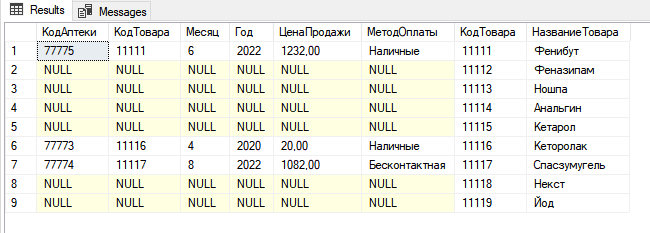


Рисунок 14 – Результат SELECT

Левое внешнее соединение

select \* from Прайслист

left join Товар ON Прайслист.КодТовара = Товар.КодТовара;

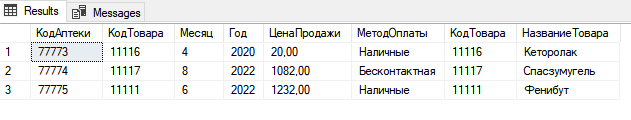


Рисунок 15 – Результат SELECT

Внутреннее соединение

select \* from Прайслист

inner join Товар on Прайслист.КодТовара = Товар.КодТовара;

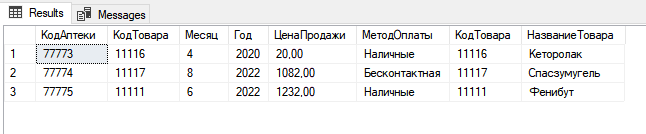


Рисунок 16 – Результат SELECT

Полное соединение

select \* from Прайслист

full outer join Товар ON Прайслист.КодТовара = Товар.КодТовара;



Рисунок 17 – Результат SELECT

1. Создать и реализовать запросы, используя оператор CASE (два варианта синтаксиса).

* Не принимается бесконтактная оплата в отделениях

select МетодОплаты = case МетодОплаты

when 'Бесконтактная'

THEN 'Оплата не приминается'

else МетодОплаты

END

, НазваниеТовара

from Прайслист

left join Товар ON Прайслист.КодТовара = Товар.КодТовара;

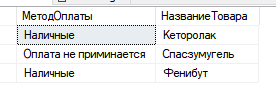


Рисунок 18 – Результат SELECT

* Дешёвые товары меняют цену

select НазваниеТовара, ЦенаПродажи = case

when ЦенаПродажи < 300

THEN 'Уточните в отделениии'

ELSE CAST(ЦенаПродажи AS CHAR(8))

END

from Прайслист

left join Товар ON Прайслист.КодТовара = Товар.КодТовара;

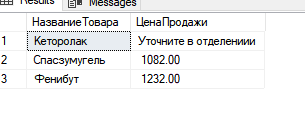


Рисунок 19 – Результат SELECT

1. Создать и реализовать запросы, используя подзапросы в списке выводимых значений, в разделе FROM, в разделе WHERE.

В списке выводимых значений:

* Вывести номера телефонов аптек и директоров

select ТелефонАптеки,(

select Фамилия from ДиректорФИО where Аптека.КодДиректора = ДиректорФИО.КодДиректора

) as 'Директор' from Аптека;

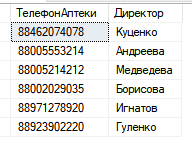


Рисунок 20 – Результат SELECT

В разделе FROM:

* Вывести телефон апетк, где у директора фамилия заканчивается на «ко»

select ТелефонАптеки from Аптека where Аптека.КодДиректора in (

select КодДиректора from ДиректорФИО where Фамилия like '%ко'

)

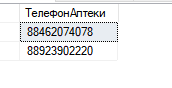


Рисунок 21 – Результат SELECT

В разделе Where:

* Вывести телефоны аптек в тольяттинском районе

select Район, ТелефонАптеки from

(

select Район,ТелефонАптеки from Аптека

left join АдресАптеки on Аптека.КодАдреса = АдресАптеки.КодАдреса

) subq where Район like 'тольятти%'

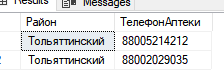


Рисунок 22 – Результат SELECT