ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»  
(РУТ (МИИТ))

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

КУРСОВОЙ ПРОЕКТ

По дисциплине «Информационное обеспечение систем управления»

Выполнил: ст. гр. ТУУ – 411

Сесягина А. А.

Проверил: к.т.н., доц.

Васильева М.А.

Москва 2021

**Оглавление**

[Цель курсового проекта 3](#_Toc90982611)

[Задания на курсовой проект 3](#_Toc90982612)

[Инфологическое проектирование БД 3](#_Toc90982613)

[Введение 4](#_Toc90982614)

[Основная часть 7](#_Toc90982615)

[Логическое проектирование реляционной БД 7](#_Toc90982616)

[Составление реляционных отношений 8](#_Toc90982617)

[Нормализация полученных отношений 10](#_Toc90982618)

[Физическое проектирование БД 10](#_Toc90982619)

[Разработка необходимых запросов 15](#_Toc90982620)

[Вывод 18](#_Toc90982621)

[Список литературы 18](#_Toc90982622)

## 

# **Цель курсового проекта**

Целью курсового проекта является изучение методов и закрепление знаний в проектировании реляционных баз данных (РБД) с примерами и заданиями для исполнения в системе управления базами данных (СУБД) Microsoft SQL Server.я DELPHI и системе управления базами данных Paradox.

# Задания на курсовой проект

В данном курсовом проекте ставится задача разработать РБД в СУБД Microsoft SQL Server по заданной теме (формализующую заданную предметную область ПрО). Проектирование РБД проводится с помощью метода «Сущность-связь». Проверка построенной модели РБД осуществляется с помощью метода нормализации отношений. Пояснительная записка должна содержать пункты по проектированию РБД и пункты по разработке РБД в СУБД Microsoft SQL Server: разработка скриптов на создание и заполнение РБД, разработка необходимых функций, процедур, триггеров и представлений (views).

Заданная тема, формализующая заданную предметную область – «Доставка еды из ресторана».

# Инфологическое проектирование БД

Описание предметной области. БД создается для информационного обслуживания сотрудников службы доставки еды. Сотрудники видят информацию о составе заказа и его доставке.

Готовые запросы:

1. Показать содержимое выбранного заказа.
2. Показать, какие пиццы заказали клиенты.
3. Показать, сколько времени доставляли каждый заказ.
4. Показать, кому доставил заказы выбранный курьер.
5. Показать стоимость заказа выбранного клиента.
6. Показать, кто успел сделать заказ до полудня (открытие в 8:00).

# **Введение**

Реляционная модель данных основывается на математических принципах, которые вытекают из теории множеств и математической логики. Эти принципы впервые были применены в области моделирования данных в конце 1960-х годов доктором Е.Ф. Коддом, а впервые опубликованы – в 1970 году. Тогда же и появились первые прототипы реляционных систем управления базами данных (СУБД).

Кодд предложил применение реляционной алгебры к системе управления реляционной базой данных (СУРБД), для разбиения данных в связанные наборы. Он организовал свою СУБД вокруг теории, основанной на наборах данных.

Самая распространенная трактовка реляционной модели данных принадлежит известному последователю идей Кодда Кристоферу Дейту. Согласно трактовке Дейта, реляционная модель состоит из трех частей, описывающих разные аспекты реляционного подхода: структурной части, манипуляционной части и целостной части.

В структурной части модели фиксируется, что единственной родовой структурой данных, используемой в РБД, является нормализованное n-арное отношение (кортеж). Определяются понятия доменов, атрибутов, кортежей, заголовка, тела и переменной отношения.

В манипуляционной части модели определяются два фундаментальных механизма манипулирования РБД – реляционная алгебра и реляционное исчисление. Первый механизм базируется на классической теории множеств, а второй – на классическом логическом аппарате исчисления предикатов первого порядка. Основной функцией манипуляционной части реляционной модели является обеспечение меры реляционности любого конкретного языка РБД: язык называется реляционным, если он обладает не меньшей выразительностью и мощностью, чем реляционная алгебра или реляционное исчисление.

Наконец, в целостной части реляционной модели данных фиксируются два базовых требования целостности, которые должны поддерживаться в любой реляционной СУБД. Первое требование называется требованием целостности сущности (entity integrity). Объекту или сущности реального мира в РБД соответствуют кортеже отношений. Конкретно требование состоит в том, что любой кортеж любого значения-отношения любой переменной отношения должен быть отличим от любого другого кортежа этого значения отношения по составным значениям заранее определенного множества атрибутов переменной отношения, т.е. ***любая переменная отношения должна обладать первичным ключом***. Это требование автоматически удовлетворяется, если в системе не нарушаются базовые свойства отношений.

Модель данных, или концептуальное описание ПрО – начальный уровень проектирования баз данных.

С точки зрения теории РБД, основные принципы реляционной модели на концептуальном уровне можно сформулировать следующим образом:

* все данные представляются в виде структуры, определенной в виде строк и столбцов и называемой отношением;
* все значения являются скалярами. Это означает, что для любой строки и столбца любого отношения существует одно и только одно значение;
* все операции выполняются над целым отношением, и результатом их выполнения также является целое отношение. Этот принцип называется замыканием.

Формулируя принципы реляционной модели, доктор Кодд выбрал термин «отношение» (relation), т.к. он считал, что этот термин однозначен (в то время как, например, термин «таблица» имеет множество различных видов – таблица в тексте, электронная таблица и т.д.).

Каждая строка, содержащая данные, называется ***кортежем***, каждый столбец отношения называется ***атрибутом*** (на уровне практической работы с современными РБД используются термины «запись» и «поле»).

Элементами описания реляционной модели данных на концептуальном уровне являются сущности, атрибуты, домены и связи.

***Сущность*** – некоторый обособленный объект или событие, информацию о котором необходимо сохранять в БД; сущность имеет определенный набор свойств – атрибутов. Сущности могут быть как физические, так и абстрактные. Для сущностей различают ее тип и экземпляр. ***Тип*** характеризуется именем и списком свойств, а ***экземпляр*** – конкретными значениями свойств.

***Домен*** – это набор всех допустимых значений, которые может содержать атрибут. Понятие «домен» часто путают с понятием «тип данных». Необходимо различать эти два понятия. Тип данных – это физическая концепция, а домен – логическая. Например, «целое число» - это тип данных, а «возраст» - это домен.

Существует несколько типов связей между двумя сущностями: это связи «один – к – одному», «один – ко – многим» и «многие – ко – многим».

Каждая связь в реляционной модели характеризуется именем, обязательностью, типов и степенью. Различают *факультативные* и *обязательные* связи. Если сущность одного типа оказывается по необходимости связанной с сущностью другого типа, то между этими типами объектов существует *обязательная* связь. Иначе связь является *факультативной*.

Степень связи определяется количеством сущностей, которые охвачены данной связью.

*Диаграмма «сущности-связи»* (Entity-Relationship diagrams, или ER-diagram) служит для описания схемы базы на концептуальном уровне проектирования. Метод был предложен в 1976 г. Питером Пин Шань Ченом (Peter Pin Shan Chen).

# **Основная часть**

*Выделим базовые сущности ПрО:*

Person – полное имя (FullName), номер телефона (PhoneNumber);

Role – название роли (RoleName);

PersonRole – человек (PersonID), роль (RoleID);

Product – название в меню (MenuName), цена (Price);

Order – покупатель (CustomerID), курьер (CourierID), адрес доставки (Address), время получения заказа(GetTime), время доставки заказа(DeliveryTime);

OrderProduct – заказ(OrderID), блюдо (ProductID).

# Логическое проектирование реляционной БД



**Рисунок 1** – ER диаграмма будущей БД

# Составление реляционных отношений

Таблица 1 – Схема отношения люди (Person)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | PersonID | int | первичный ключ |
| Полное имя | FullName | varchar  (255) | обязательное поле |
| Номер телефона | PhoneNumber | varchar  (255) | обязательное поле |

Таблица 2 – Схема отношения роли (Role)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | RoleID | int | первичный ключ |
| Название роли | RoleName | varchar  (255) | обязательное поле |

Таблица 3 – Схема отношения люди – роли (PersonRole)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | PersonRoleID | int | первичный ключ |
| Человек | PersonID | int | внешний ключ (Person), обязательное поле |
| Роль | RoleID | int | внешний ключ (Role), обязательное поле |

Таблица 4 – Схема отношения блюда (Product)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | ProductID | int | первичный ключ |
| Название в меню | MenuName | varchar  (255) | обязательное поле |
| Цена | Price | float | обязательное поле |

Таблица 5 – Схема отношения заказы (Order)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | OrderID | int | первичный ключ |
| Покупатель | CustomerID | int | внешний ключ (Person), обязательное поле |
| Курьер | CourierID | int | внешний ключ (Person), обязательное поле |
| Адрес доставки | Address | varchar (255) | обязательное поле |
| Время получения заказа | GetTime | time | обязательное поле |
| Время доставки заказа | DeliveryTime | time | обязательное поле |

Таблица 6 – Схема отношения продукты в заказах (OrderProduct)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | OrderProductID | int | первичный ключ |
| Заказ | OrderID | int | внешний ключ (Order), обязательное поле |
| Блюдо | ProductID | int | внешний ключ (Product), обязательное поле |

# Нормализация полученных отношений

**1НФ.** Для приведения таблиц к 1НФ требуется составить таблицы (один атрибут – один столбец) и разбить сложные атрибуты на простые, а многозначные атрибуты вынести в отдельные отношения.

Все отношения находятся в 1 НФ.

**2НФ.** Все отношения находятся во 2НФ, так как для каждого из них введен уникальный ключ.

# Физическое проектирование БД

*Разработка скриптов на создание базы данных и таблиц:*

1. **InitDB** – создаёт БД FoodDelivery:

CREATE DATABASE FoodDelivery;

1. **InitTables** – создаёт таблицы:

USE FoodDelivery;

*Создание таблицы Person:*

CREATE TABLE Person

(

PersonID INT PRIMARY KEY IDENTITY(1,1),

FullName VARCHAR(255) NOT NULL,

PhoneNumber VARCHAR(255) NOT NULL

);

*Создание таблицы Role:*

CREATE TABLE [Role]

(

RoleID INT PRIMARY KEY IDENTITY(1,1),

RoleName VARCHAR(255) NOT NULL

);

*Создание таблицы PersonRole:*

CREATE TABLE PersonRole

(

PersonRoleID INT PRIMARY KEY IDENTITY(1,1),

PersonID INT NOT NULL,

FOREIGN KEY (PersonID) REFERENCES Person (PersonID),

RoleID INT NOT NULL,

FOREIGN KEY (RoleID) REFERENCES Role (RoleID)

);

*Создание таблицы Product:*

CREATE TABLE Product

(

ProductID INT PRIMARY KEY IDENTITY(1,1),

MenuName VARCHAR(255) NOT NULL,

Price FLOAT NOT NULL

);

*Создание таблицы Order:*

CREATE TABLE [Order]

(

OrderID INT PRIMARY KEY IDENTITY(1,1),

CustomerID INT NOT NULL,

FOREIGN KEY (CustomerID) REFERENCES Person (PersonID),

CourierID INT NOT NULL,

FOREIGN KEY (CourierID) REFERENCES Person (PersonID),

[Address] VARCHAR(255) NOT NULL,

GetTime TIME NOT NULL,

DeliveryTime TIME NOT NULL

);

*Создание таблицыOrderProduct:*

CREATE TABLE OrderProduct

(

OrderProductID INT PRIMARY KEY IDENTITY(1,1),

OrderID INT NOT NULL,

FOREIGN KEY (OrderID) REFERENCES [Order] (OrderID),

ProductID INT NOT NULL,

FOREIGN KEY (ProductID) REFERENCES Product (ProductID)

);

1. **InsertValues** – заполняет таблицы значениями:

--PERSON

INSERT INTO Person (FullName, PhoneNumber) VALUES

('Губанова Дарья Дмитриевна', '89201473971'),

('Киреева Анна Марковна', '89104445501'),

('Пантелеева Валерия Егоровна', '89626991262'),

('Сальникова Кира Вадимовна', '89261703708'),

('Крылов Олег Степанович', '89201356864 '),

('Новиков Дмитрий Михайлович', '89852133194'),

('Акимов Марк Робертович', '89164796307'),

('Громов Иван Георгиевич', '89265656522'),

('Дмитриев Павел Владимирович', '89627181595'),

('Ильин Александр Артёмович', '89201630213'),

('Фадеев Максим Алексеевич', '89855885559'),

('Федосеев Артём Семёнович', '89162218129');

--ROLE

INSERT INTO [Role] (RoleName) VALUES

('Покупатель'),

('Курьер');

--Покупатель (1, 2, 3 , 4 , 5, 7, 8, 10, 11)

--Курьер (6, 9, 12)

--PERSONROLE

INSERT INTO PersonRole (PersonID, RoleID) VALUES

(1, 1),

(2, 1),

(3, 1),

(4, 1),

(5, 1),

(6, 2),

(7, 1),

(8, 1),

(9, 2),

(10, 1),

(11, 1),

(12, 2);

--PRODUCT

INSERT INTO Product (MenuName, Price) VALUES

('Пицца Пепперони', 649),--1

('Пицца Маргарита', 549),--2

('Пицца Ветчина и грибы', 549),--3

('Пицца Гавайская', 649),--4

('Салат Цезарь', 199),--5

('Паста Карбонара', 269),--6

('Паста Цыпленок и грибы', 269),--7

('Чизкейк Нью-Йорк', 129),--8

('Вишневый пирог', 169),--9

('Шоколадный молочный коктель', 169),--10

('Ванильный молочный коктель', 159),--11

('Клубничный молочный коктель', 169),--12

('Coca-Cola', 99),--13

('Sprite', 99),--14

('BonAqua негазированная', 69);--15

--ORDER

INSERT INTO [Order] (CustomerID, CourierID, [Address], GetTime, DeliveryTime) VALUES

(1, 6, 'ул. Бориса Галушкина, д. 18, кв. 34', '08:43:54', '09:23:32'),

(2, 9, 'ул. Сущевская, д. 29, кв. 73', '09:01:15', '09:56:43'),

(3, 6, 'ул. Космонавтов, д. 14, кв. 62', '10:35:24', '11:16:46'),

(4, 9, 'ул. Палиха, д. 9, кв. 46', '10:53:23', '11:28:41'),

(5, 12, 'ул. Верхняя Красносельская, д. 11а, кв. 110', '11:14:53', '11:53:36'),

(7, 6, 'ул. Ярославская, д. 10 к1, кв. 12', '12:21:42', '13:02:45'),

(8, 9, 'ул. Тихвинская, д. 4, кв. 191', '12:46:31', '13:30:54'),

(10, 12, 'ул. Верхняя Красносельская, д. 10, кв. 55', '14:56:43', '15:40:32'),

(11, 12, 'ул. Лобачика, д. 11', '16:08:03', '16:56:42');

--ORDERPRODUCT

INSERT INTO OrderProduct (OrderID, ProductID) VALUES

(1, 1),

(1, 4),

(1, 11),

(1, 11),

(2, 2),

(2, 5),

(2, 14),

(3, 6),

(3, 7),

(3, 8),

(3, 13),

(3, 15),

(4, 1),

(4, 3),

(4, 6),

(4, 9),

(5, 3),

(5, 10),

(6, 6),

(6, 9),

(6, 12),

(7, 1),

(7, 2),

(8, 7),

(8, 14),

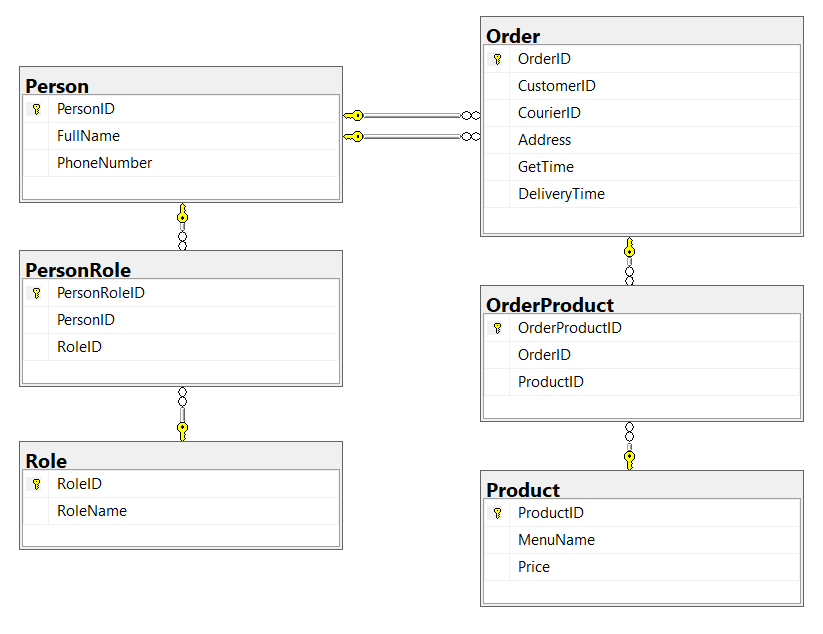
(9, 3),

(9, 4),

(9, 10),

(9, 13);

На рисунке 2 приведена диаграмма базы данных:



**Рисунок 2** – ER диаграмма сгенерированная MS SQL Server Management Studio 2018

# Разработка необходимых запросов

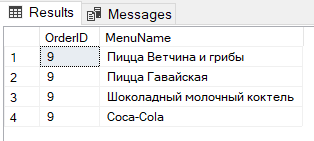
1. Показать содержимое выбранного заказа.

--Показать содержимое выбранного заказа.

SELECT O.OrderID, P.MenuName

FROM [Order] AS O JOIN OrderProduct ON O.OrderID = OrderProduct.OrderID JOIN Product AS P ON OrderProduct.ProductID = P.ProductID

WHERE O.OrderID = 9;



**Рисунок 3** – Результат выполнения запроса

1. Показать, какие пиццы заказали клиенты.

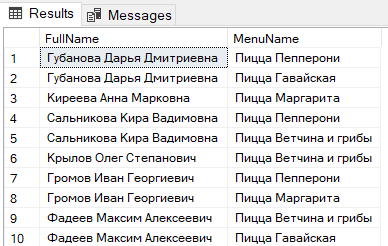
--Показать, какие пиццы заказали клиенты.

SELECT Per.FullName, Pr.MenuName

FROM Person AS Per JOIN [Order] ON Per.PersonID = [Order].CustomerID JOIN OrderProduct ON [Order].OrderID = OrderProduct.OrderID

JOIN Product AS Pr ON OrderProduct.ProductID = Pr.ProductID

WHERE Pr.MenuName LIKE 'Пицца%';



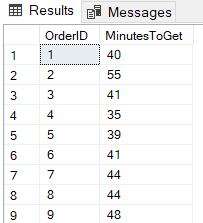
**Рисунок 4** – Результат выполнения запроса

1. Показать, сколько времени доставляли каждый заказ.

--Показать, сколько времени доставляли каждый заказ

SELECT OrderID, DATEDIFF(MINUTE, GetTime, DeliveryTime) AS [MinutesToGet]

FROM [Order];



**Рисунок 5** – Результат выполнения запроса

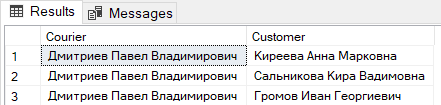
1. Показать, кому доставил заказы выбранный курьер.

--Показать, кому доставил заказы выбранный курьер.

SELECT P1.FullName AS Courier, P2.FullName AS Customer

FROM Person AS P1 JOIN [Order] AS O1 ON O1.CourierID = P1.PersonID JOIN Person AS P2 ON O1.CustomerID = P2.PersonID

WHERE P1.FullName LIKE 'Дмитриев Павел Владимирович';



**Рисунок 6** – Результат выполнения запроса

1. Показать стоимость заказа выбранного клиента.

--Показать стоимость заказа выбранного клиента.

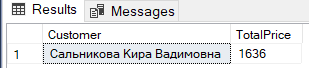
SELECT Per.FullName AS Customer, SUM(Pr.Price) AS TotalPrice

FROM Person AS Per JOIN [Order] AS O ON Per.PersonID = O.CustomerID JOIN OrderProduct AS OP ON O.OrderID = OP.OrderID

JOIN Product AS Pr ON OP.ProductID = Pr.ProductID

WHERE Per.FullName LIKE 'Сальникова Кира Вадимовна'

GROUP BY Per.FullName;



**Рисунок 7** – Результат выполнения запроса

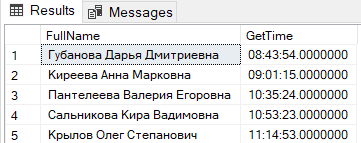
1. Показать, кто успел сделать заказ до полудня (открытие в 8:00).

--Показать, кто успел сделать заказ до полудня (открытие в 8:00).

SELECT Per.FullName, O.GetTime

FROM Person AS Per JOIN [Order] AS O ON Per.PersonID = O.CustomerID

WHERE O.GetTime LIKE '08:%' OR O.GetTime LIKE '09:%' OR O.GetTime LIKE '10:%' OR O.GetTime LIKE '11:%';



**Рисунок 8** – Результат выполнения запроса

# Вывод

Изучены методы и закреплены знания в проектировании реляционных баз данных (РБД) с примерами и заданиями для исполнения в системе управления базами данных (СУБД) Microsoft SQL Server для БД «Доставка еды из ресторана».

# Список литературы

1. Балакина Е.П., Васильева М.А., Филипченко К.М. Информационное обеспечение систем управления. Учебно-методическое пособие. Издание второе, исправленное и дополненное – М.: РУТ (МИИТ), 2020. – с.