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

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

КУРСОВАЯ РАБОТА

По дисциплине «Информационное обеспечение систем управления»

Выполнил: ст. гр. ТУУ – 411

Иконников А.С.

Проверил: к.т.н., доц.

Васильева М.А.

Москва 2021

Оглавление

[Цель курсового проекта 2](#_Toc90984462)

[Задания на курсовой проект 2](#_Toc90984463)

[Инфологическое проектирование БД 2](#_Toc90984464)

[Логическое проектирование реляционной БД 4](#_Toc90984465)

[Составление реляционных отношений 5](#_Toc90984466)

[Нормализация полученных отношений 10](#_Toc90984467)

[Физическое проектирование БД 10](#_Toc90984468)

[Разработка необходимых запросов 14](#_Toc90984469)

[Вывод 18](#_Toc90984470)

## **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.

## 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).

Пояснительная записка оформляется согласно ГОСТ 7.32-2017 «ОТЧЕТ О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ» [1].

## Инфологическое проектирование БД

Описание предметной области. База данных создаётся для информационного обслуживания работников отдела кадров. В фирме работают сотрудники на разных должностях и разных ставках. Сотрудники могут иметь детей, быть пенсионерами или иметь инвалидность. Сотрудникам предоставляет очередной отпуск. Некоторые сотрудники могут быть в отпуске по уходу за ребенком.

Готовые запросы:

1. Показывать список сотрудников фирмы по ФИО, по должности, ставке, имеющих детей, пенсионеров, находящихся в очередном отпуске, находящихся в отпуске по уходу за ребенком.
2. Выдавать информацию о должностях сотрудников.
3. Выдавать информацию о предыдущих местах работы сотрудников.
4. Находить сотрудников по ФИО.

Выделим базовые сущности этой ПрО:

Работник – Имя, Отчество, Фамилия, день рождения, домашний адресс, телефон, пенсионер, рабочий договор, подразделение, пол.

Прошлая работа – Название, позиция, дата начал работы, дата конца работы,

работник.

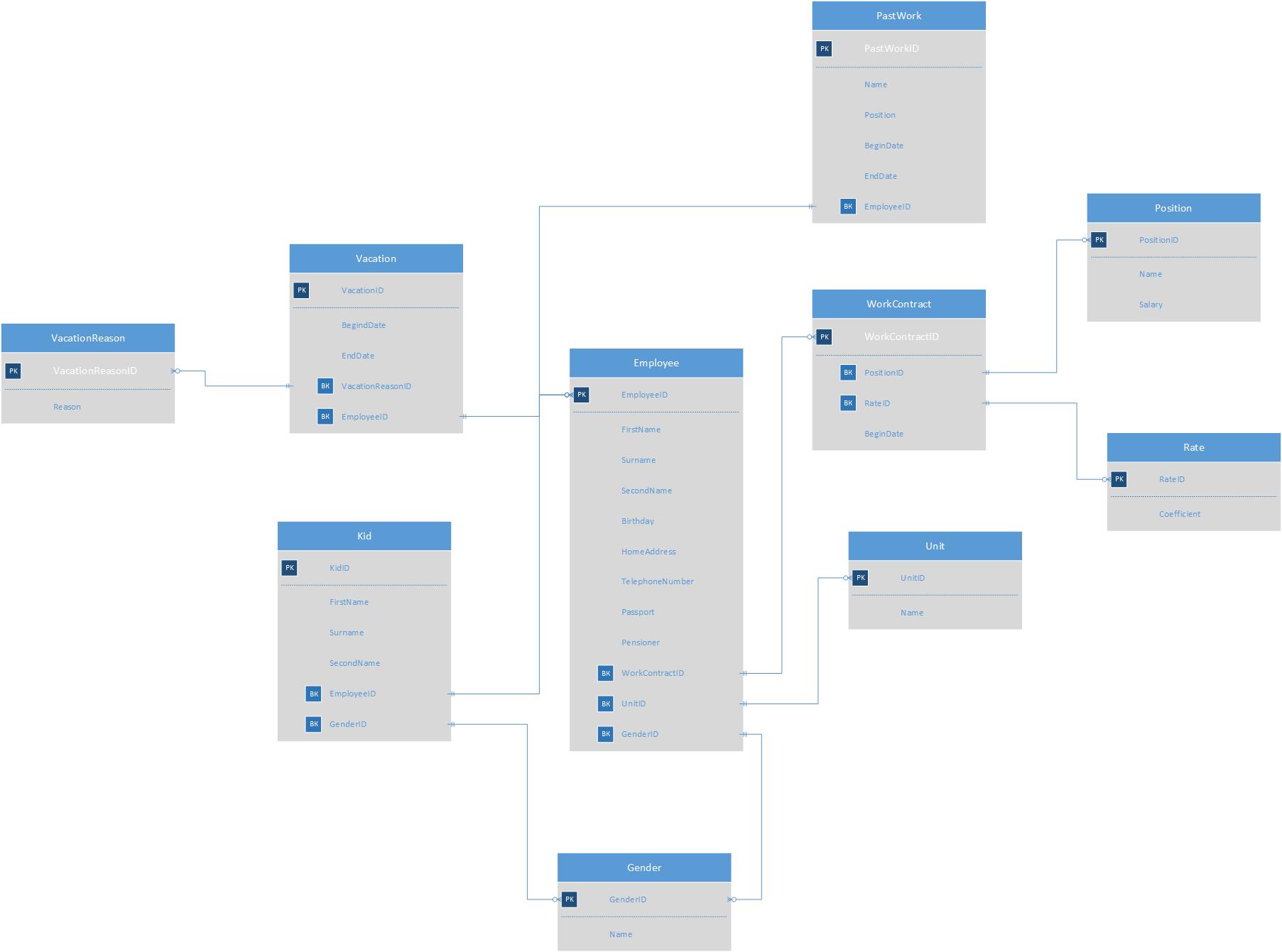
Рабочий договор – Позиция, ставка, дата начала работы.

Позиция – Название, полная ставка.

Ставка – Коэффициент ставки.

.

## Логическое проектирование реляционной БД



**Рисунок 1** – ER диаграмма будущей БД

## Составление реляционных отношений

Таблица 1 – Схема отношения работник (Employee)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | EmployeeID | Целое число | первичный ключ |
| Имя | FirstName | Строка (255) | обязательное поле |
| Отчество | SurName | Строка (255) | обязательное поле |
| Фамилия | SecondName | Строка (255) | обязательное поле |
| День рождения | Birthday | Дата | Обязательное поле |
| Домашний адрес | HomeAddress | Строка (255) | обязательное поле |
| Телефонный номер | TelephoneNumber | Строка (255) | обязательное поле |
| Пенсионер | Pensioner | Бит | обязательное поле |
| Рабочий договор | WorkContractID | Целое число | внешний ключ (WorkContraract), обязательное поле |
| Подразделение | UnitID | Целое число | внешний ключ (Unit), обязательное поле |
| Пол | GenderID | Целое число | внешний ключ (Gender), обязательное поле |

Таблица 2 – Схема отношения позиция (Position)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | PositionID | Целое число | первичный ключ |
| Название позиции | Name | Строка (255) | обязательное поле |
| Полная ставка | Salary | Целое число | Обязательное поле |

Таблица 3 – Схема отношения ставка (Rate)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | RateID | Целое число | первичный ключ |
| Коэффициент | Coefficient | Число с плавающей точкой | обязательное поле |

Таблица 4 – Схема отношения отдел (Unit)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | UnitID | Целое число | первичный ключ |
| Название отдела | Name | Строка (255) | обязательное поле |

Таблица 5 – Схема отношения пол (Gender)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | GenderID | Целое число | первичный ключ |
| Название альбома | Name | Строка (255) | обязательное поле |

Таблица 6 – Схема отношения рабочий договор (Composer)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | WorkContractID | Целое число | первичный ключ |
| Позиция | PositionID | Целое число | внешний ключ (Position), обязательное поле |
| Ставка | RateID | Целое число | внешний ключ (Rate), обязательное поле |
| Дата начала договора | BeginDate | Дата | обязательное поле |
| Дата конца договора | EndDate | Дата |  |

Таблица 7 – Схема прошлая работа (PastWork)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | PastWorkID | Целое число | первичный ключ |
| Название | Name | Строка (255) | обязательное поле |
| Позиция | Position | Строка (255) | обязательное поле |
| Дата начала работы | BeginDate | Дата | обязательное поле |
| Дата конца работы | EndDate | Дата | обязательное поле |
| Работник | EmployeeID | Целое число | внешний ключ (Employee), обязательное поле |

Таблица 8 – Схема отношения ребёнок (Kid)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | KidID | Целое число | первичный ключ |
| Имя | FirstName | Строка (255) | обязательное поле |
| Отчество | SurName | Строка (255) | обязательное поле |
| Фамилия | SecondName | Строка (255) | обязательное поле |
| Работник | EmployeeID | Целое число | внешний ключ (Employee), обязательное поле |
| Пол | GenderID | Целое число | внешний ключ (Gender), обязательное поле |

Таблица 9 – Схема причина отпуска (VacationReason)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | VacationReasonID | Целое число | первичный ключ |
| Причина | Reason | Строка (255) | обязательное поле |

Таблица 10 – Схема причина отпуска (VacationReason)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | VacationID | Целое число | первичный ключ |
| Дата начала отпуска | BeginDate | Дата | обязательное поле |
| Дата конца отпуска | EndDate | Дата | обязательное поле |
| Работник | EmployeeID | Целое число | внешний ключ (Employee), обязательное поле |
| Причина отпуска | VacationReasonID | Целое число | внешний ключ (VacationReason), 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:

InitDB – создаёт БД HumanResourcesDepartment

CREATE DATABASE HumanResourcesDepartment;

InitTable – создаёт таблицы.

USE HumanResourcesDepartment;

CREATE TABLE Position (

PositionID INT PRIMARY KEY IDENTITY(1,1),

[Name] VARCHAR(255) NOT NULL,

Salary INT NOT NULL,

);

CREATE TABLE Rate (

RateID INT PRIMARY KEY IDENTITY(1,1),

[Coefficient] FLOAT NOT NULL,

);

CREATE TABLE Unit (

UnitID INT PRIMARY KEY IDENTITY(1,1),

[Name] VARCHAR(255) NOT NULL,

);

CREATE TABLE Gender (

GenderID INT PRIMARY KEY IDENTITY(1,1),

[Name] VARCHAR(255) NOT NULL,

);

CREATE TABLE WorkContract (

WorkContractID INT PRIMARY KEY IDENTITY(1,1),

PositionID INT NOT NULL,

FOREIGN KEY (PositionID) REFERENCES Position (PositionID),

RateID INT NOT NULL,

FOREIGN KEY (RateID) REFERENCES Rate (RateID),

BeginDate DATE NOT NULL,

);

CREATE TABLE Employee (

EmployeeID INT PRIMARY KEY IDENTITY(1,1),

FirstName VARCHAR(255) NOT NULL,

SurName VARCHAR(255) NOT NULL,

SecondName VARCHAR(255) NOT NULL,

Birthday DATE NOT NULL,

HomeAddress VARCHAR(255) NOT NULL,

TelephoneNumber VARCHAR(255) NOT NULL,

Passport VARCHAR(255) NOT NULL,

Pensioner BIT NOT NULL,

WorkContractID INT NOT NULL,

FOREIGN KEY (WorkContractID) REFERENCES WorkContract (WorkContractID),

UnitID INT NOT NULL,

FOREIGN KEY (UnitID) REFERENCES Unit (UnitID),

GenderID INT NOT NULL,

FOREIGN KEY (GenderID) REFERENCES Gender (GenderID),

);

CREATE TABLE PastWork (

PastWorkID INT PRIMARY KEY IDENTITY(1,1),

[Name] VARCHAR(255) NOT NULL,

Position VARCHAR(255) NOT NULL,

BeginDate DATE NOT NULL,

EndDate DATE NOT NULL,

EmployeeID INT NOT NULL,

FOREIGN KEY (EmployeeID) REFERENCES Employee (EmployeeID),

);

CREATE TABLE Kid (

KidID INT PRIMARY KEY IDENTITY(1,1),

FirstName VARCHAR(255) NOT NULL,

SurName VARCHAR(255) NOT NULL,

SecondName VARCHAR(255) NOT NULL,

EmployeeID INT NOT NULL,

FOREIGN KEY (EmployeeID) REFERENCES Employee (EmployeeID),

GenderID INT NOT NULL,

FOREIGN KEY (GenderID) REFERENCES Gender (GenderID),

);

CREATE TABLE VacationReason (

VacationReasonID INT PRIMARY KEY IDENTITY(1,1),

Reason VARCHAR(255) NOT NULL,

);

CREATE TABLE Vacation (

VacationID INT PRIMARY KEY IDENTITY(1,1),

BeginDate DATE NOT NULL,

EndDate DATE NOT NULL,

EmployeeID INT NOT NULL,

FOREIGN KEY (EmployeeID) REFERENCES Employee (EmployeeID),

VacationReasonID INT NOT NULL,

FOREIGN KEY (VacationReasonID) REFERENCES VacationReason (VacationReasonID),

);

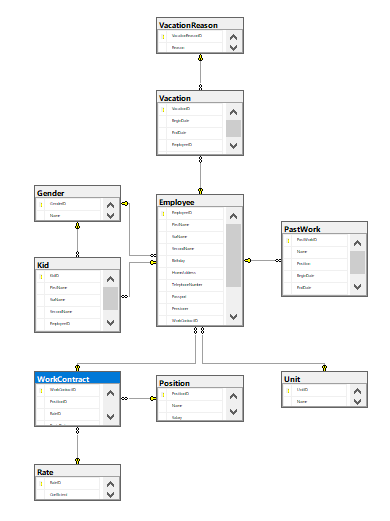


Рисунок 2 – ER диаграмма сгенерированная MS Management Studio

Разработка скриптов на добавление данных в таблицы

-- Insert Positions

INSERT INTO Position VALUES ('DevOps', '150000');

INSERT INTO Position VALUES ('Backend Developer', '200000');

INSERT INTO Position VALUES ('Frontend Developer', '180000');

-- Insert Rates

INSERT INTO Rate VALUES (0.25);

INSERT INTO Rate VALUES (0.5);

INSERT INTO Rate VALUES (0.75);

INSERT INTO Rate VALUES (1);

-- Insert WorkContracts

INSERT INTO WorkContract VALUES (1, 1, '1999-04-12');

INSERT INTO WorkContract VALUES (2, 2, '2002-08-15');

INSERT INTO WorkContract VALUES (3, 3, '2002-05-29');

-- Insert Genders

INSERT INTO Gender VALUES ('Male');

INSERT INTO Gender VALUES ('Female');

-- Insert Units

INSERT INTO Unit VALUES ('BigOS');

INSERT INTO Unit VALUES ('LittleOS');

-- Insert Employees

INSERT INTO Employee VALUES ('Andrey', 'Ivanovich', 'Prutko',

'1970-03-03', 'Moskva, Pushika 43', '88187714',

'1712 252037', 0 , 1, 1, 1);

INSERT INTO Employee VALUES ('Evgeniy', 'Petrovich', 'Tushin',

'1970-03-23', 'Moskva, Pushika 47', '88423423423',

'1712 212332', 1 , 2, 1, 1);

INSERT INTO Employee VALUES ('Valeriya', 'Mihaylovna', 'Horosheva',

'1970-03-23', 'Moskva, Kosmonavtov 47', '88424243242',

'4332 252347', 0 , 3, 2, 2);

-- Insert Kids

INSERT INTO Kid VALUES ('Sergey', 'Andreevich', 'Prutko',1,1);

INSERT INTO Kid VALUES ('Anya', 'Andreevna', 'Prutko',1,2);

-- Insert Past Works

INSERT INTO PastWork VALUES ('Yandex', 'SysAdmin', '1990-03-23','1999-03-02', 1);

INSERT INTO PastWork VALUES ('MailRu', 'SysAdmin', '1999-03-03','1999-03-27', 1);

-- Insert Vacation Reasons

INSERT INTO VacationReason VALUES ('Usual');

INSERT INTO VacationReason VALUES ('Baby Care');

-- Insert Vacations

INSERT INTO Vacation VALUES ('2002-03-19', '2005-03-19', 1, 2);

INSERT INTO Vacation VALUES ('2003-09-01', '2003-09-28', 2, 1);

## Разработка необходимых запросов

Показать список сотрудников фирмы по ФИО

--1. Показывать список сотрудников фирмы по ФИО, по должности, ставке, имеющих детей,   
--пенсионеров, находящихся в очередном отпуске, находящихся в отпуске по  
--уходу за ребенком.  
SELECT \* FROM Employee

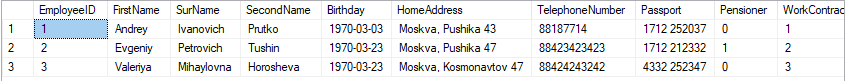


Рисунок 3 ‑ Результат выполнения запроса

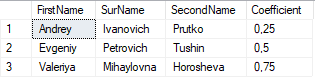
Показать список сотрудников фирмы по должности.

--1. Показывать список сотрудников фирмы по ФИО, по должности, ставке, имеющих детей,   
--пенсионеров, находящихся в очередном отпуске, находящихся в отпуске по  
--уходу за ребенком.  
SELECT FirstName, SurName, SecondName, [Name] FROM Employee  
JOIN WorkContract ON Employee.WorkContractID = WorkContract.WorkContractID  
JOIN Position ON WorkContract.PositionID = Position.PositionID

  
Рисунок 4 ‑ Результат выполнения запроса

Показать список сотрудников фирмы по ставке.

--1. Показывать список сотрудников фирмы по ФИО, по должности, ставке, имеющих детей,   
--пенсионеров, находящихся в очередном отпуске, находящихся в отпуске по  
--уходу за ребенком.  
SELECT FirstName, SurName, SecondName, Coefficient FROM Employee  
JOIN WorkContract ON Employee.WorkContractID = WorkContract.WorkContractID  
JOIN Rate ON WorkContract.RateID = Rate.RateID

  
Рисунок 5 ‑ Результат выполнения запроса

Показать список сотрудников фирмы имеющих детей.

--1. Показывать список сотрудников фирмы по ФИО, по должности, ставке, имеющих детей,   
--пенсионеров, находящихся в очередном отпуске, находящихся в отпуске по  
--уходу за ребенком.  
SELECT DISTINCT Employee.FirstName, Employee.SurName, Employee.SecondName FROM Employee  
JOIN Kid ON Kid.EmployeeID = Employee.EmployeeID

  
Рисунок 6 ‑ Результат выполнения запроса

Показать список сотрудников фирмы пенсионеров.

--1. Показывать список сотрудников фирмы по ФИО, по должности, ставке, имеющих детей,   
--пенсионеров, находящихся в очередном отпуске, находящихся в отпуске по  
--уходу за ребенком.  
SELECT \* FROM Employee  
WHERE Pensioner = 1

  
Рисунок 7 ‑ Результат выполнения запроса

Показать список сотрудников фирмы находящихся в очередном отпуске.

--1. Показывать список сотрудников фирмы по ФИО, по должности, ставке, имеющих детей,   
--пенсионеров, находящихся в очередном отпуске, находящихся в отпуске по  
--уходу за ребенком.  
SELECT DISTINCT FirstName, SurName, SecondName FROM Employee  
JOIN Vacation ON Vacation.EmployeeID = Employee.EmployeeID  
WHERE Vacation.VacationID = (SELECT VacationReasonID FROM VacationReason WHERE Reason = 'Usual')

  
Рисунок 8 ‑ Результат выполнения запроса

Показать список сотрудников фирмы находящихся в отпуске по уходу за ребенком.

--1. Показывать список сотрудников фирмы по ФИО, по должности, ставке, имеющих детей,   
--пенсионеров, находящихся в очередном отпуске, находящихся в отпуске по  
--уходу за ребенком.  
SELECT DISTINCT FirstName, SurName, SecondName FROM Employee  
JOIN Vacation ON Vacation.EmployeeID = Employee.EmployeeID  
WHERE Vacation.VacationID = (SELECT VacationReasonID FROM VacationReason WHERE Reason = 'Baby care')

  
Рисунок 9 ‑ Результат выполнения запроса

Выдавать информацию о должностях сотрудников.

--2. Выдавать информацию о должностях сотрудников.  
SELECT FirstName, SurName, SecondName, [Name] FROM Employee  
JOIN WorkContract ON Employee.WorkContractID = WorkContract.WorkContractID  
JOIN Position ON WorkContract.PositionID = Position.PositionID

  
Рисунок 10 ‑ Результат выполнения запроса

Выдавать информацию о предыдущих местах работы сотрудников.

--3. Выдавать информацию о предыдущих местах работы сотрудников.  
SELECT FirstName, SurName, SecondName, [Name] FROM Employee  
JOIN PastWork ON Employee.EmployeeID = PastWork.EmployeeID

  
Рисунок 11 ‑ Результат выполнения запроса

Находить сотрудников по ФИО.

--4. Находить сотрудников по ФИО.  
SELECT \* FROM Employee  
WHERE FirstName = 'Andrey' AND SurName = 'Ivanovich' AND SecondName = 'Prutko'

  
Рисунок 12 ‑ Результат выполнения запроса

## Вывод

Изучил методы и закрепил знания в проектировании реляционных баз данных (РБД) с примерами и заданиями для исполнения в системе управления базами данных (СУБД) Microsoft SQL Server.