Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

Рязанский государственный радиотехнический университет имени В.Ф. Уткина

Кафедра ЭВМ

К защите

Руководитель работы:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата, подпись

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОЙ РАБОТЕ**

по дисциплине

**«Базы данных»**

Тема:

«Разработка базы данных для предметной области: «Тестирование»»

|  |  |  |
| --- | --- | --- |
| Выполнил студент группы 1413 |  |  |
| Веркин С. А. |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  дата сдачи на проверку, подпись |
|  |  |  |
| Руководитель работы |  |  |
| Асс. |  |  |
| Баранова С. Н. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  оценка | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  дата защиты, подпись |
|  |  |  |

СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 2](#_Toc122639548)

[ВВЕДЕНИЕ 3](#_Toc122639549)

[1. Семантическое описание предметной области 4](#_Toc122639550)

[2. Разработка базы данных 6](#_Toc122639551)

[2.1 Разработка логической модели данных 6](#_Toc122639552)

[2.1.1 Описание основных сущностей ПО. 6](#_Toc122639553)

[2.1.2 Построение логической модели 8](#_Toc122639554)

[2.2 Определение типов хранимых данных 10](#_Toc122639555)

[2.3 Определение типов хранимых данных 11](#_Toc122639556)

[2.4 Построение схемы БД 13](#_Toc122639557)

[2.5 Разработка сценария для создания БД и основных объектов структуры БД 14](#_Toc122639558)

[3. Разработка основных операций реляционной алгебры 18](#_Toc122639559)

[4. Разработка запросов к БД для реализации информационных потребностей пользователя 21](#_Toc122639560)

[ЗАКЛЮЧЕНИЕ 41](#_Toc122639561)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 41](#_Toc122639562)

ВВЕДЕНИЕ

На момент декабря 2022 года SQL занимает 8 место в рейтинге самых используемых языков программирования. Именно поэтому создание базы данных на этом языке будет полезным и актуальным. Язык SQL прост в освоении, но для создания полноценной базы данных необходимо изучить и использовать значительное число возможностей этого языка.

Целью курсовой работы является изучение и применение методов проектирования баз данных в Microsoft SQL Server, результатом курсовой работы является база данных на заданную тему.

**ВАРИАНТ 5.**

База данных «Тестирование». База данных содержит информацию о тестировании сотрудников организации: В базе требуется хранить данные:

1. О сотрудниках. Базовая информация по сотруднику + список пройденных тестов
2. О профессии. Информация по профессии, которой обладает сотрудник.
3. О тестах. Содержит тематику теста, создателя, время создания. Связан с профессией. Одна профессия может включать несколько тестов.
4. О вопросах. Содержит описание вопроса, тип вопроса (с выбором, ответ в свободной форме и тд), балл за верный ответ, время создания, создателя.

# Семантическое описание предметной области

Предметной областью базы данных текущего проекта является тестирование сотрудников известной фирмы.

Тестированием называется испытание качества, особенностей или функций или иных характеристик какого-либо субъекта. Тест — основной элемент предметной области — единица тестирования, направленная на проверку способностей сотрудника к участию в трудовой деятельности и ценность для фирмы. Виды тестов представлены следующим множеством значений: {Тест на наркотики, Тест на алкоголь, Тест на социальные навыки, Тест на физическую подготовку, Тест на технические навыки, Тест общего характера}. Все тесты содержат по несколько вопросов, за исключением тестов на наркотики и теста на алкоголь — данные тесты содержат единственный вопрос по поводу предоставления документа о медицинском освидетельствовании начальнику фирмы.

Базовая информация по сотруднику — его фамилия, имя, отчество (эти данные нужны для почтительного обращения к сотруднику). Возрастные, физические, этнические и иные вторичные характеристики компании не важны — все люди одинаковы, пока хорошо справляются с работой и верны делу фирмы.

Каждый сотрудник должен сдавать определённое количество тестов, число тестов определяет профессия сотрудника. Представители технических профессий должны пройти основной комплекс тестов, в то время как основные сотрудники сферы логистики — водители и пилоты сверхтяжёлых транспортных самолётов — обязаны пройти тест на наркотики и алкоголь.

# Разработка базы данных

# Разработка логической модели данных

# Описание основных сущностей ПО.

На основании проведённого анализа предметной области «Тестирование» можно выделить следующие сущности:

Сотрудники, Профессия, Тесты, Вопросы.

Представим список сущностей предметной области (таблица 1).

Таблица 1 – Список сущностей

|  |  |  |
| --- | --- | --- |
| № | Название | Назначение |
| 1 | Сотрудники | Перечень сотрудников (за исключением начальства), трудящихся на благо известной фирмы |
| 2 | Профессия | Описание должностей и специальностей сотрудников (за исключением начальства) известной фирмы |
| 3 | Тесты | Список тестов, необходимых к сдаче сотрудниками фирмы |
| 4 | Вопросы | Полный список вопросов из всех тестов |
| 5 | Прохождения тестов | Список пройденных сотрудниками тестов |

Сущность Сотрудники не имеет какого-либо поля, с помощью которого можно однозначно идентифицировать его, потому что их ФИО могут повторяться. Поэтому требуется ввести суррогатный атрибут ID, который и будет являться первичным ключом для данной сущности. Такой атрибут будет генерироваться СУБД автоматически с помощью свойства IDENTITY с начальным значением 1 и шагом 1.

Таблица 2 – Список атрибутов таблицы «Сотрудники»

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
|  | ФИО |  |
| ВК (внешний ключ) | ID Должности | Внешний ключ, с помощью которого можно определить, на какой должности находится сотрудник. |
|  | Служебный телефон сотрудника |  |
| ПК (первичный ключ) | ID Сотрудника | Ключевое поле. Представляет собой первичный ключ. Это уникальное значение, соответствующее каждому арендатору. Значения автоматически генерируются СУБД при вставке новой записи в таблицу. |

Таблица 3 – Список атрибутов таблицы «Профессия»

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
|  | Профессия |  |
|  | Заработная плата |  |
| ПК (первичный ключ) | ID Должности | Ключевое поле. Представляет собой первичный ключ. |

Таблица 4 – Список атрибутов таблицы «Тесты»

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
|  | Тематика теста |  |
|  | Время создания |  |
| Создатель |
| ПК (первичный ключ) + ВК (внешний ключ) | Номер теста | Ключевое поле. Представляет собой первичный ключ и внешний ключ, отвечающий за связь с таблицей Вопросы |

Таблица 5 – Список атрибутов таблицы «Вопросы»

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
|  | Описание вопроса |  |
|  | Тип вопроса |  |
|  | Балл за верный ответ |  |
|  | Время создания |  |
| Создатель |
| ПК (первичный ключ) | Номер вопроса | Ключевое поле. Представляет собой первичный ключ. |
| ВК (внешний ключ) | Номер теста | Составной внешний ключ, связующий с таблицей «тесты» |

Таблица 6 – Список атрибутов таблицы «Прохождения тестов»

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ВК (внешний ключ) | ID Сотрудника | Ключ. связывающий сотрудника с прохождением теста |
|  | Время прохождения теста |  |
| ВК (внешний ключ) | Тематика теста | Ключ, связывающий прохождение теста с тестом |
| ПК (первичный ключ) | Номер прохождения теста | Ключевое поле. Представляет собой первичный ключ. |

# 

# Построение логической модели

На основании семантического описания предметной области и списка атрибутов из таблиц 2-6 опишем классы объектов (сущностей) и их свойства, расставим существующие связи (таблица 7) между ними и приведём обоснование типов связей (1:1, 1:М и т. д.). Для визуализации составим схему этих связей (рисунок 1)

Таблица 7 – Список связей

|  |  |  |  |
| --- | --- | --- | --- |
| № | Сущности, участвующие в связи | Тип связи | Обоснование |
| 1 | Профессия — Сотрудники | 1: N | Одна и та же профессия может быть у нескольких сотрудников, но у сотрудника может быть только одна профессия. |
| 2 | Сотрудник – прохождения тестов | 1:N | Один и тот же сотрудник может иметь несколько разных результатов прохождения тестов, но один результат прохождения теста не может быть у нескольких сотрудников |
| 3 | Тесты – вопросы | 1:N | Один и тот же вопрос может относиться только к одному тесту, но к одному тесту могут относиться несколько разных вопросов |
| 4 | Тесты – прохождения тестов | 1:N | Один и тот же тест может быть пройден несколько раз, но одно прохождение не может относиться к разным тестам |

# 

Рисунок 1 – Логическая модель предметной области "Тестирование"

# Определение типов хранимых данных

Для каждой таблицы из таблиц 2-6 приведём список всех атрибутов с указанием типа данных. При выборе типов данных следует учитывать, что далее для реализации будет использоваться СУБД Microsoft SQL Server. Результаты приведены в таблицах 8–12.

Таблица 8 – Список атрибутов таблицы «Сотрудники»

|  |  |
| --- | --- |
| Название | Тип данных |
| ФИО | Varchar(40) |
| ID Должности | int |
| Служебный телефон сотрудника | Varchar(15) |
| ID Сотрудника | int, нач. значение = 1, приращение = 1 |

Таблица 9 – Список атрибутов таблицы «Профессия»

|  |  |
| --- | --- |
| Название | Тип данных |
| Профессия | Varchar(25) |
| Заработная плата | money |
| ID Должности | int |

Таблица 10 – Список атрибутов таблицы «Тесты»

|  |  |
| --- | --- |
| Название | Тип данных |
| Тематика теста | Varchar(50) |
| Создатель | Varchar(40) |
| Номер теста | int |
| Время создания | date |

Таблица 11 – Список атрибутов таблицы «Вопросы»

|  |  |
| --- | --- |
| Название | Тип данных |
| Описание вопроса | Varchar(100) |
| Тип вопроса | Varchar(30) |
| Балл за верный ответ | int |
| Время создания | date |
| Создатель | Varchar(40) |
| Номер вопроса | int |
| Номер теста | int |

Таблица 12 – Список атрибутов таблицы «Прохождения тестов»

|  |  |
| --- | --- |
| Название | Тип данных |
| ID Сотрудника | int |
| Время прохождения теста | Time(7) |
| Тематика теста | Varchar(50) |
| Номер прохождения теста | int |

# Определение типов хранимых данных

Для каждой таблицы необходимо выделить ограничения, накладываемые на значения атрибутов. Это могут быть ограничения на значения, значения по умолчанию, запрет/разрешение использования NULL-значений, поддержка уникальности значений и др. Результаты представлены в таблицах 13-17.

Таблица 13 – Ограничения таблицы «Сотрудники»

|  |  |
| --- | --- |
| Название | Ограничение |
| ФИО | Должно содержать только буквы, не может быть NULL |
| ID Должности | Целое, не может принимать NULL-значения. |
| Служебный телефон сотрудника | +N (NNN) NNN-NN-NN, где N – цифра от 0 до 9. |
| ID Сотрудника | Первичный ключ. Это уникальное значение, не может принимать NULL-значения. Значения автоматически генерируются СУБД, начиная от 1 с шагом 1. |

Таблица 14 – Ограничения таблицы «Профессия»

|  |  |
| --- | --- |
| Название | Ограничение |
| Профессия | Должно содержать только буквы, не может быть NULL |
| Заработная плата | Значение по умолчанию – 90 000 рублей. |
| ID Должности | Первичный ключ. Это уникальное значение, не может принимать NULL-значения. Значения автоматически генерируются СУБД, начиная от 1 с шагом 1. |

Таблица 15 – Ограничения таблицы «Тесты»

|  |  |
| --- | --- |
| Название | Ограничение |
| Тематика теста | Внешний ключ. Допустимые значения: Тест на наркотики, Тест на алкоголь, Тест на социальные навыки, Тест на физическую подготовку, Тест на технические навыки, Тест общего характера.  Не может принимать NULL-значения. Значение по умолчанию – Тест общего характера. |
| Время создания | ДД-ММ-ГГГГ, не может принимать NULL-значения. |
| Создатель | Состоит из букв, по умолчанию – Альберт Эйнштейн, не может принимать NULL-значения. |
| Номер теста | Первичный ключ. Это уникальное значение, не может принимать NULL-значения. Значения автоматически генерируются СУБД, начиная от 1 с шагом 1. |

Таблица 16 – Ограничения таблицы «Прохождения тестов»

|  |  |
| --- | --- |
| Название | Ограничение |
| ID Сотрудника | Внешний ключ |
| Время прохождения теста | Time(7) |
| Тематика теста | Внешний ключ |
| Номер прохождения теста | Первичный ключ. Это уникальное значение, не может принимать NULL-значения. Значения автоматически генерируются СУБД, начиная от 1 с шагом 1. |

Таблица 17 – Ограничения таблицы «Вопросы»

|  |  |
| --- | --- |
| Название | Ограничение |
| Описание вопроса | Varchar(100) |
| Тип вопроса | С выбором, ответ в свободной форме, несколько вариантов. По умолчанию – ответ в свободной форме |
| Балл за верный ответ | Int, больше 0 |
| Время создания | ДД-ММ-ГГ |
| Создатель | По умолчанию – Альберт Эйнштейн |
| Номер вопроса | Первичный ключ. Это уникальное значение, не может принимать NULL-значения. Значения автоматически генерируются СУБД, начиная от 1 с шагом 1. |
| Номер теста | Внешний ключ |

# 2.4 Построение схемы БД

Схема БД представляет собой набор заголовков таблиц. То есть на схеме БД указаны имена таблиц (сущностей), их атрибуты и связи между таблицами.

Для уточнения схемы БД для каждой сущности (таблицы) указывают наименование атрибута, идентификатор, тип, длина, формат, ограничения. На рисунке 2 представлена схема базы данных «Тестирование».

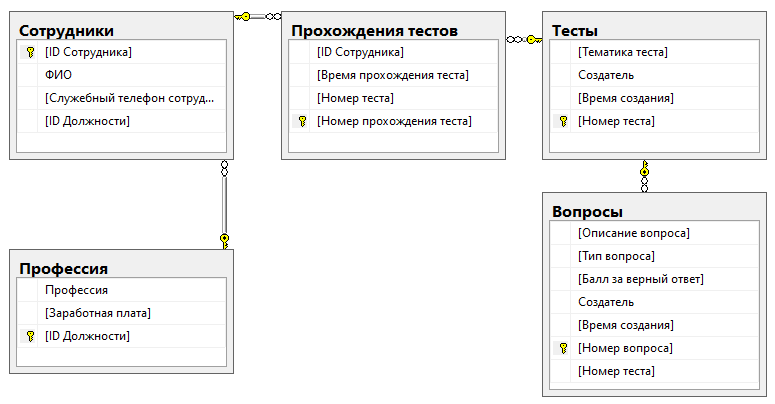


Рисунок – схема базы данных "Тестирование"

# 2.5 Разработка сценария для создания БД и основных объектов структуры БД

При создании таблиц будем использовать следующие способы задания ограничений:

1. При описании оператора CREATE TABLE будем задавать следующие ограничения:

–ограничения NOT NULL как ограничение на столбец;

–ограничения DEFAULT как ограничение на столбец;

–неповторяющиеся ограничения CHECK и UNIQUE, то есть, если конкретное ограничение будет использоваться только один раз в БД.

Такие ограничения будем задавать как ограничения на столбец или на таблицу (в зависимости от количества столбцов, входящих в ограничение).

При описании оператора ALTER TABLE будем задавать следующие ограничения:

– повторяющиеся ограничения CHECK и UNIQUE, то есть, если конкретное ограничение будет использоваться несколько раз в БД;

– первичные ключи (простые и составные);

– внешние ключи.

Всем ограничениям, описанным в операторе ALTER TABLE, будем задавать имена.

Создание структуры БД

USE MASTER

GO

CREATE DATABASE Тестирование

ON

(NAME='Course\_Data',

FILENAME='C:\Program Files\Microsoft SQL Server\MSSQL12.SQLEXPRESS\MSSQL\DATA\Course\_Data.mdf',

SIZE = 5,

MAXSIZE = 10,

FILEGROWTH = 1)

LOG ON

(NAME='Course\_Log',

FILENAME='C:\Program Files\Microsoft SQL Server\MSSQL12.SQLEXPRESS\MSSQL\DATA\Course\_Log.ldf',

SIZE = 2,

MAXSIZE = 20,

FILEGROWTH = 2)

GO

USE Тестирование

CREATE TABLE Профессия

(Профессия nvarchar(50) NOT NULL CHECK (Профессия LIKE '[а-яА-ЯA-Za-z1-9]%'),

[Заработная плата] money NOT NULL check ([Заработная плата]> 0) default 90000,

[ID Должности] int IDENTITY (1,1) PRIMARY KEY NOT NULL,)

GO

CREATE TABLE Сотрудники

([ID Сотрудника] int IDENTITY (1,1) PRIMARY KEY NOT NULL,

ФИО varchar(60) NOT NULL CHECK ([ФИО] LIKE '[а-яА-ЯA-Za-z]%'),

[Служебный телефон сотрудника] char(50) CHECK([Служебный телефон сотрудника] LIKE '+[1-9] ([0-9][0-9][0-9]) [0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9]'),

[ID Должности] int NOT NULL,)

GO

ALTER TABLE Сотрудники

ADD CONSTRAINT FK\_Сотрудники\_ID\_Должности foreign key ([ID Должности]) references Профессия([ID Должности])

GO

CREATE TABLE Тесты

([Тематика теста] varchar(50) NOT NULL CHECK ([Тематика теста] IN ('Тест на наркотики', 'Тест на алкоголь', 'Тест на социальные навыки', 'Тест на физическую подготовку', 'Тест на технические навыки', 'Тест общего характера')) default 'Тест общего характера',

[Создатель] varchar(40) NOT NULL CHECK ([Создатель] LIKE '[а-яА-ЯA-Za-z1-9]%') default 'Альберт Эйнштейн',

[Время создания] date NOT NULL,

[Номер теста] int IDENTITY (1,1) PRIMARY KEY NOT NULL,)

GO

CREATE TABLE Вопросы

([Описание вопроса] varchar(100) NOT NULL CHECK ([Описание вопроса] LIKE '[а-яА-ЯA-Za-z1-9]%'),

[Тип вопроса] varchar(30) NOT NULL CHECK ([Тип вопроса] IN ('С выбором', 'Ответ в свободной форме', 'Несколько вариантов')) default 'Ответ в свободной форме',

[Балл за верный ответ] int NOT NULL check ([Балл за верный ответ] > 0),

[Создатель] varchar(40) NOT NULL CHECK ([Создатель] LIKE '[а-яА-ЯA-Za-z1-9]%') default 'Альберт Эйнштейн',

[Время создания] date NOT NULL,

[Номер вопроса] int IDENTITY (1,1) PRIMARY KEY NOT NULL,

[Номер теста] int NOT NULL,)

GO

ALTER TABLE Вопросы

ADD CONSTRAINT FK\_Вопросы\_Номер\_теста foreign key ([Номер теста]) references Тесты([Номер теста])

GO

CREATE TABLE [Прохождения тестов]

([ID Сотрудника] int NOT NULL,

[Время прохождения теста] Time(2) CHECK ([Время прохождения теста]>'00:00:05'),

[Номер теста] int NOT NULL,

[Номер прохождения теста] int IDENTITY (1,1) NOT NULL,)

GO

ALTER TABLE [Прохождения тестов]

ADD CONSTRAINT PK\_Прохождения PRIMARY KEY ([Номер прохождения теста])

GO

ALTER TABLE [Прохождения тестов]

ADD CONSTRAINT FK\_Прохождения\_ID\_Сотрудника foreign key ([ID Сотрудника]) references Сотрудники([ID Сотрудника])

GO

ALTER TABLE [Прохождения тестов]

ADD CONSTRAINT FK\_Прохождения\_Номер\_теста foreign key ([Номер теста]) references Тесты([Номер теста])

GO

INSERT INTO Профессия VALUES

('Механик-водитель', 120000),

('Специалист по кадрам', 120000),

('Специалист по оборудованию', 250000),

('Пилот транспортного самолёта', 600000),

('Раздатчик чая', 120000),

('Продавец товаров', 120000),

('Оператор механизированного сборщика', 120000),

('Руководитель отдела', 200000),

('Строитель', 120000),

('Инженер', 250000)

GO

INSERT INTO Сотрудники VALUES

('Курков Антон Егорович', '+7 (910) 001-00-01', 1),

('Хмелько Николай Николаевич', '+7 (910) 002-00-02', 1),

('Куркова Анастасия Егоровна', '+7 (910) 003-00-03', 2),

('Редькин Игорь Богданович', '+7 (910) 004-00-04', 3),

('Алмасов Ясир Мухаммедович', '+7 (910) 005-00-05', 3),

('Касумова Айдан Олеговна', '+7 (910) 006-00-06', 4),

('Балин Сергей Эдуардович', '+7 (910) 007-00-07', 1),

('Толкуев Денис Исаевич', '+7 (910) 008-00-08', 1),

('Шишкин Виталий Григорьевич', '+7 (910) 009-00-09', 4),

('Редькин Игорь Анатольевич', '+7 (910) 010-00-10', 2)

GO

INSERT INTO Тесты VALUES

('Тест на наркотики', 'Доктор Стоун', '2000-01-20'),

('Тест на алкоголь', 'Доктор Стоун', '2000-02-20'),

('Тест на социальные навыки', 'Дейл Карнеги', '2000-01-20'),

('Тест на физическую подготовку', 'Юрий Гагарин', '2000-01-20'),

('Тест на технические навыки', 'Леонардо да Винчи', '2000-01-20'),

('Тест общего характера', default, '2000-01-20'),

('Тест на социальные навыки', 'Денис Денисов', '2001-01-20'),

('Тест на физическую подготовку', 'Иван Иванов', '2001-01-20'),

('Тест на технические навыки', 'Пётр Петров', '2001-01-20'),

('Тест общего характера', 'Мухаммед Боксёров', '2001-01-20')

GO

INSERT INTO Вопросы VALUES

('Было ли пройдено медобследование?', 'С выбором', 1, 'Доктор Стоун', '2000-01-20', 1),

('Было ли пройдено медобследование?', 'С выбором', 1, 'Доктор Стоун', '2000-02-20', 2),

('Каково число клиентов, с которыми Вам приходилось иметь дело?', default, 1, 'Дейл Карнеги', '2000-01-20', 3),

('Как часто Вы делаете зарядку?', 'С выбором', 1, 'Юрий Гагарин', '2000-01-20', 4),

('Как Вы будете реагировать на ненормальное поведение клиента?', default, 1, 'Дейл Карнеги', '2000-01-20', 3),

('Как Вы будете решать проблему наладки оборудования?', 'С выбором', 1, default, '2000-01-20', 5),

('Как Вы будете решать проблему ремонта вашего транспортного средства?', default, 1, 'Леонардо да Винчи', '2000-01-20', 5),

('Что Вы можете рассказать о себе?', default, 1, default, '2000-01-20', 6),

('Каково число клиентов, с которыми Вам приходилось иметь дело?', default, 1, 'Дейл Карнеги', '2000-01-20', 3),

('В чём смысл жизни?', default, 1, 'Философ', '2000-01-20', 7),

('Зачем нужно работать?', default, 1, 'Философ', '2000-01-20', 7),

('Почему вы хотите работать?', default, 1, 'Философ', '2000-01-20', 7),

('Хотите работать?', default, 1, 'Философ', '2000-01-20', 8),

('У вас есть свободное время?', default, 1, 'Философ', '2000-01-20', 8),

('У вас есть семья?', default, 1, 'Философ', '2000-01-20', 8),

('У вас есть мотивация?', default, 1, 'Философ', '2000-01-20', 9),

('Никогда не хотели отдохнуть?', default, 1, 'Философ', '2000-01-20', 9),

('Что у вас на уме?', default, 1, 'Философ', '2000-01-20', 9),

('Размышляли о бытие?', default, 1, 'Философ', '2000-01-20', 10),

('Какой сегодня день?', default, 1, 'Философ', '2000-01-20', 10)

GO

INSERT INTO [Прохождения тестов] VALUES

(1, '00:01:00', 6),

(2, '00:01:00', 6),

(3, '00:01:00', 6),

(4, '00:01:00', 6),

(5, '00:01:00', 6),

(6, '00:01:00', 6),

(7, '00:01:00', 6),

(8, '00:01:00', 6),

(9, '00:01:00', 6),

(1, '00:01:00', 1),

(2, '00:02:00', 1),

(7, '00:02:00', 1),

(8, '00:03:00', 1),

(1, '00:03:00', 2),

(2, '00:00:50', 2),

(7, '00:03:00', 2),

(8, '00:03:00', 2),

(3, '00:03:00', 3),

(7, '00:03:00', 3),

(1, '00:03:00', 3),

(1, '00:03:00', 4),

(3, '00:03:00', 4),

(6, '00:03:00', 4),

(5, '00:03:00', 4),

(4, '00:03:00', 4),

(2, '00:03:00', 4),

(4, '00:03:00', 5),

(5, '00:03:00', 5),

(6, '00:03:00', 5)

# Разработка основных операций реляционной алгебры

Для созданной базы данных необходимо создать и подробно описать не менее 15 запросов с использованием операций реляционной алгебры.

1. Вывести все тесты на наркотики

R=Тесты WHERE Тематика теста = ‘Тест на наркотики’

2. Вывести прохождения тестов без ID сотрудников

R=(Прохождения тестов) [Время прохождения теста], [Номер теста], [Номер прохождения теста]

3. Вывести всех сотрудников с их данными о профессиях

R=Сотрудники JOIN Профессия

4. Вывести ФИО как личные данные

R=Сотрудники RENAME ФИО AS Личные данные

5. Вывести ФИО всех сотрудников, не прошедших тест на наркотики

R1=Тесты WHERE Тематика теста = ‘Тест на наркотики’

R2=R1 JOIN Прохождения тестов

R3=Сотрудники LEFT JOIN R2

R4=R3 WHERE Номер теста IS NULL

R5=(R4) [ФИО]

6. Вывести данные о 1 и 3 сотрудниках

R1=Сотрудники WHERE ID Сотрудника = 1

R2=Сотрудники WHERE ID Сотрудника = 3

R3=R2 UNION R1

7. Вывести данные о всех профессиях, кроме второй

R1=Профессия WHERE ID Должности = 2

R2=Профессия EXCEPT R1

8. Вывести ФИО сотрудников, проходивших и общие тесты, и тесты на алкоголь

R1=Сотрудники JOIN Прохождения

R2=R1 JOIN Тесты

R3=(R2 WHERE Тематика теста = ‘Тест на алкоголь’) [ФИО]

R4=(R2 WHERE Тематика теста = ‘Тест общего характера’) [ФИО]

R5=R3 INTERSECT R4

9. Вывести ФИО сотрудников, проходивших каждый из тестов под номерами 7, 8 и 12

R1=Тесты WHERE Номер теста= 7

R2=Тесты WHERE Номер теста= 8

R3=Тесты WHERE Номер теста= 12

R4=R1 UNION R2

R5=(R4 UNION R3)[Номер теста]

R6=Сотрудники JOIN Прохождения тестов

R7=R6 JOIN Тесты

R8=(R7 DIVIDE BY R5) [ФИО]

10. Вывести все вопросы от создателя тестов Альберта Эйнштейна

R1=Тесты WHERE Создатель = ‘Альберт Эйнштейн’

R2=(R1)[Номер теста]

R3=Вопросы JOIN R2

11. Вывести служебные телефоны сотрудников с профессией Механик-водитель

R1=Профессия WHERE Профессия = ‘Механик-водитель’

R2=Сотрудники JOIN R1

R3=(R2)[Служебный телефон сотрудника]

12. Вывести ФИО сотрудников и номера вопросов, пройденные ими.

R1=Сотрудники JOIN Прохождения тестов

R2=R1 JOIN Тесты

R3=R2 JOIN Вопросы

R4=(R3)[ФИО][Номер вопроса]

13. Вывести ФИО сотрудников, зарплата которых больше 100 тысяч

R1=Профессия WHERE Заработная плата>100000

R2=R1 JOIN Сотрудники

R3=(R2)[ФИО]

14. Вывести номера прохождений тестов для Специалистов по кадрам

R1=Профессия WHERE Профессия = ‘Специалист по кадрам’

R2=R1 JOIN Сотрудники

R3=R2 JOIN Прохождения тестов

R4=(R3)[Номер прохождения теста]

15. Вывести всех сотрудников, кроме Курковой Анастасии Егоровны

R1=Сотрудники WHERE ФИО = ‘Куркова Анастасия Егоровна’

R2=Сотрудники EXCEPT R1

# Разработка запросов к БД для реализации информационных потребностей пользователя

1. Запросы с выбором строк по условию:

- запрос с использованием операторов сравнения;

* 1. Вывести все профессии с зарплатой более 150000 (Результат на рисунке 3)

SELECT \*

FROM Профессия

WHERE [Заработная плата]>150000

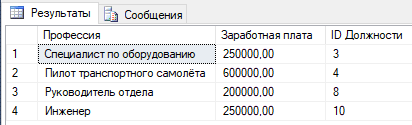


Рисунок – Запрос 1.1

- запрос с использованием логических операторов AND, OR и NOT;

* 1. Вывести все профессии, где зарплата больше 100 тысяч и меньше 400 тысяч или больше 500 тысяч и меньше 700 тысяч, но не равна 120 тысячам (Результат на рисунке 4)

SELECT \*

FROM Профессия

WHERE (([Заработная плата]>100000 AND [Заработная плата]<400000) OR ([Заработная плата]>500000 AND [Заработная плата]<700000)) AND NOT([Заработная плата]=120000)

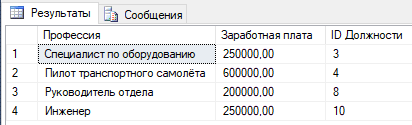


Рисунок – Запрос 1.2

- запрос на использование комбинации логических операторов;

* 1. Вывести всех сотрудников с ID Должности больше 1 но меньше 4 или с ID Сотрудника больше 1 но меньше 4 (Результат на рисунке 5)

SELECT \*

FROM Сотрудники

WHERE ([ID Должности]>1 AND [ID Должности]<4) OR ([ID Сотрудника]>1 AND [ID Сотрудника]<4)

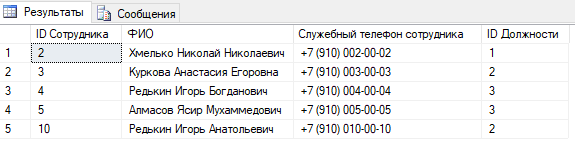


Рисунок – Запрос 1.3

- запрос на использование выражений над столбцами;

* 1. Вывести все данные о профессиях, но со столбцом «Премия» со значениями в 50% от зарплаты (Результат на рисунке 6)

SELECT \*, [Заработная плата]\*0.5 AS Премия

FROM Профессия

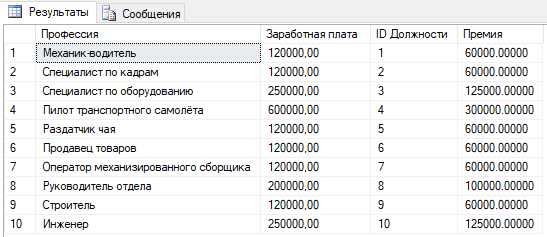


Рисунок – Запрос 1.4

- запрос с проверкой на принадлежность множеству;

* 1. Вывести всю информацию о тестах, созданных Доктором Стоуном, Альбертом Эйнштейном или Юрием Гагариным (Результат на рисунке 7)

SELECT \*

FROM Тесты

WHERE Создатель IN ('Юрий Гагарин', 'Альберт Эйнштейн', 'Доктор Стоун')

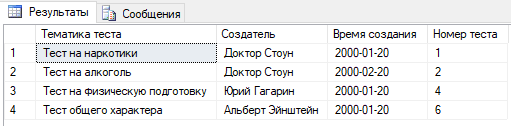


Рисунок – Запрос 1.5

- запрос с проверкой на принадлежность диапазону значений;

* 1. Вывести все вопросы с номером вопроса в диапазоне от 2 до 7 (Результат на рисунке 8)

SELECT \*

FROM Вопросы

WHERE [Номер вопроса] BETWEEN 2 AND 7

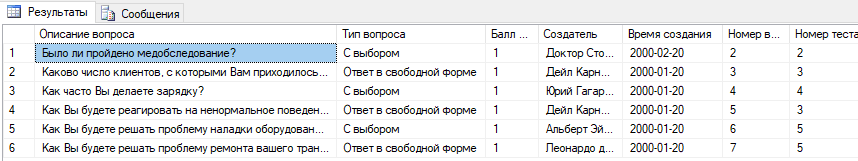


Рисунок – запрос 1.6

- запрос с проверкой на соответствие шаблону;

* 1. Вывести ФИО всех сотрудников с фамилией на К (Результат на рисунке 9)

SELECT ФИО

FROM Сотрудники

WHERE ФИО LIKE 'К%'

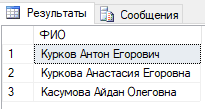


Рисунок – запрос 1.7

- запрос с проверкой на неопределенное значение.

* 1. Вывести сотрудников с номером телефона (Результат на рисунке 10)

SELECT \*

FROM Сотрудники

WHERE [Служебный телефон сотрудника] IS NOT NULL

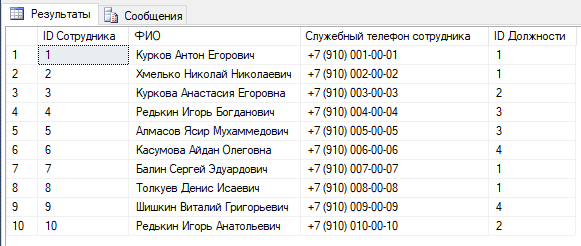


Рисунок – запрос 1.8

1. Запросы на теоретико-множественные реляционные операции:

- запрос на объединение таблиц;

* 1. Вывести ФИО сотрудников с фамилиями, начинающимися на К или на Х (Результат на рисунке 11)

SELECT ФИО

FROM Сотрудники

WHERE ФИО LIKE 'К%'

UNION

SELECT ФИО

FROM Сотрудники

WHERE ФИО LIKE 'Х%'

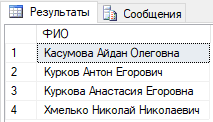


Рисунок – запрос 2.1

- запрос на пересечение таблиц;

2.2 Вывести ФИО механиков-водителей, прошедших шестой тест (Результат на рисунке 12)

SELECT Сотрудники.ФИО

FROM (Сотрудники JOIN [Прохождения тестов] ON Сотрудники.[ID Сотрудника]=[Прохождения тестов].[ID Сотрудника]) JOIN Профессия ON Сотрудники.[ID Должности]=Профессия.[ID Должности]

WHERE Профессия='Механик-водитель'

INTERSECT

SELECT Сотрудники.ФИО

FROM (Сотрудники JOIN [Прохождения тестов] ON Сотрудники.[ID Сотрудника]=[Прохождения тестов].[ID Сотрудника]) JOIN Профессия ON Сотрудники.[ID Должности]=Профессия.[ID Должности]

WHERE [Номер теста]=6

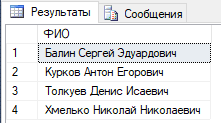


Рисунок – запрос 2.2

- запрос на вычитание таблиц;

2.3 Вывести ФИО пилотов тяжёлых самолётов, не прошедших пятый тест (Результат на рисунке 13)

SELECT Сотрудники.ФИО

FROM (Сотрудники JOIN [Прохождения тестов] ON Сотрудники.[ID Сотрудника]=[Прохождения тестов].[ID Сотрудника]) JOIN Профессия ON Сотрудники.[ID Должности]=Профессия.[ID Должности]

WHERE Профессия='Пилот транспортного самолёта'

EXCEPT

SELECT Сотрудники.ФИО

FROM (Сотрудники JOIN [Прохождения тестов] ON Сотрудники.[ID Сотрудника]=[Прохождения тестов].[ID Сотрудника]) JOIN Профессия ON Сотрудники.[ID Должности]=Профессия.[ID Должности]

WHERE [Номер теста]=5

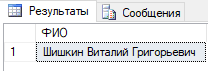


Рисунок – запрос 2.3

- запрос с использованием декартового произведения таблиц;

2.4 Вывести номера тестов, в которых нет вопросов от Альберта Эйнштейна (Результат на рисунке 14)

SELECT Тесты.[Номер теста]

FROM Тесты, Вопросы

EXCEPT

SELECT Тесты.[Номер теста]

FROM Тесты JOIN Вопросы ON Тесты.[Номер теста]=Вопросы.[Номер теста]

WHERE Вопросы.Создатель='Альберт Эйнштейн'

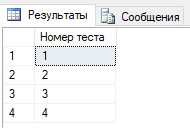


Рисунок – запрос 2.4

1. Многотабличные запросы:

- запрос с использованием соединения двух таблиц по равенству и условием отбора;

3.1 Вывести ФИО специалистов по кадрам, у которых есть служебный телефона (Результат на рисунке 15)

SELECT ФИО, Профессия

FROM Профессия, Сотрудники

WHERE Профессия.[ID Должности]=Сотрудники.[ID Должности] AND [Служебный телефон сотрудника] IS NOT NULL AND Профессия = 'Специалист по кадрам'

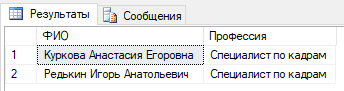


Рисунок – запрос 3.1

- запрос с использованием соединения трех таблиц по равенству и условием отбора;

3.2 Вывести профессии сотрудников, проходивших третий тест (Результат на рисунке 16)

SELECT DISTINCT Профессия

FROM Профессия, Сотрудники, [Прохождения тестов]

WHERE Профессия.[ID Должности]=Сотрудники.[ID Должности] AND Сотрудники.[ID Сотрудника]=[Прохождения тестов].[ID Сотрудника] AND [Номер теста]=3

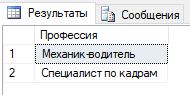


Рисунок – запрос 3.2

- запрос с использованием соединения по четырем таблицам;

3.3 Вывести ФИО сотрудников и номера тестов, пройденных специалистами по кадрам (Результат на рисунке 17)

SELECT DISTINCT ФИО, Тесты.[Номер теста]

FROM Профессия, Сотрудники, [Прохождения тестов], Тесты

WHERE Профессия.[ID Должности]=Сотрудники.[ID Должности] AND Сотрудники.[ID Сотрудника]=[Прохождения тестов].[ID Сотрудника] AND [Прохождения тестов].[Номер теста]=Тесты.[Номер теста] AND Профессия='Специалист по кадрам'

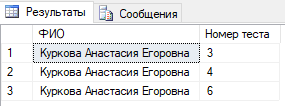


Рисунок – запрос 3.3

- запрос с использованием левого внешнего соединения;

3.4 Вывести ФИО сотрудников, зарабатывающих более 150 тысяч рублей (Результат на рисунке 18)

SELECT ФИО

FROM Профессия LEFT JOIN Сотрудники ON Профессия.[ID Должности]=Сотрудники.[ID Должности]

WHERE ФИО IS NOT NULL AND [Заработная плата]>150000

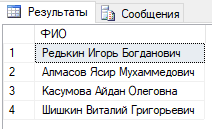


Рисунок – запрос 3.4

- запрос на использование правого внешнего соединения;

3.5 Вывести тематики тестов, автор вопросов которых – Юрий Гагарин (Результат на рисунке 19)

SELECT [Тематика теста]

FROM Тесты RIGHT JOIN Вопросы ON Тесты.[Номер теста]=Вопросы.[Номер теста]

WHERE Вопросы.Создатель='Юрий Гагарин' AND Тесты.[Тематика теста] IS NOT NULL

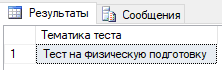


Рисунок – запрос 3.5

- запрос с использованием функции COUNT;

3.6 Вывести число людей на каждой профессии (Результат на рисунке 20)

SELECT Профессия, COUNT(Профессия.Профессия) AS [Число людей]

FROM Профессия JOIN Сотрудники ON Профессия.[ID Должности]=Сотрудники.[ID Должности]

GROUP BY Профессия.Профессия

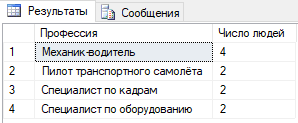


Рисунок – запрос 3.6

- запрос с использованием функции SUM;

3.7 Вывести суммарную заработную плату по профессиям (Результат на рисунке 21)

SELECT Профессия, SUM(Профессия.[Заработная плата]) AS [Суммарная заработная плата]

FROM Профессия JOIN Сотрудники ON Профессия.[ID Должности]=Сотрудники.[ID Должности]

GROUP BY Профессия.Профессия

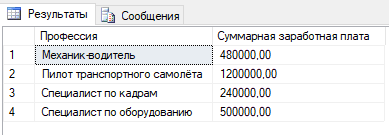


Рисунок – запрос 3.7

- 3 запроса с использованием временных функций;

3.8.1 Вывести число дней, прошедшее с момента создания тестов (Результат на рисунке 22)

SELECT datediff(DAY, Тесты.[Время создания],convert(date,getdate())) AS [Прошедшее время]

FROM Тесты

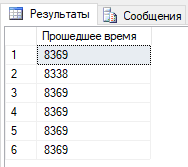


Рисунок – запрос 3.8.1

3.8.2 Вывести время, которое на 20 лет больше времени создания вопросов (Результат на рисунке 23)

SELECT dateadd(YEAR, 20,Вопросы.[Время создания]) AS [Новое время]

FROM Вопросы

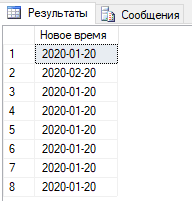


Рисунок – запрос 3.8.2

3.8.3 Вывести количество дней с момента создания самого раннего теста до момента создания самого позднего (Результат на рисунке 24)

SELECT datediff(DAY, MIN(Тесты.[Время создания]), MAX(Тесты.[Время создания])) AS [Максимальная разница]

FROM Тесты

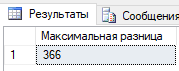


Рисунок – запрос 3.8.3

- запрос с использованием группировки по одному столбцу;

3.9 Вывести число вопросов в каждом тесте (Результат на рисунке 25)

SELECT Тесты.[Номер теста], COUNT(Вопросы.[Номер теста]) AS [Число вопросов]

FROM Тесты JOIN Вопросы ON Тесты.[Номер теста]=Вопросы.[Номер теста]

GROUP BY Тесты.[Номер теста]

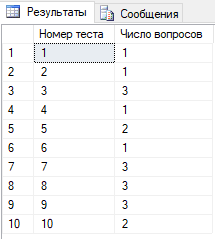


Рисунок – запрос 3.9

- запрос на использование группировки по нескольким столбцам;

3.10 Вывести профессии в алфавитном порядке, ФИО сотрудника и число отвеченных вопросов (Результат на рисунке 26)

SELECT Профессия, ФИО, COUNT(Вопросы.[Номер теста]) AS [Число вопросов]

FROM Тесты JOIN Вопросы ON Тесты.[Номер теста]=Вопросы.[Номер теста] JOIN [Прохождения тестов] ON Тесты.[Номер теста]=[Прохождения тестов].[Номер теста] JOIN Сотрудники ON Сотрудники.[ID Сотрудника]=[Прохождения тестов].[ID Сотрудника] JOIN Профессия ON Сотрудники.[ID Должности]=Профессия.[ID Должности]

GROUP BY ФИО, Профессия

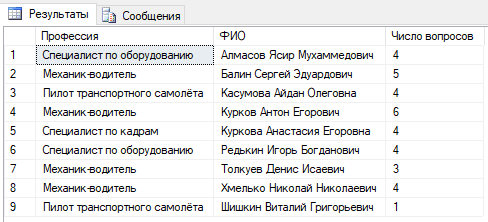


Рисунок – запрос 3.10

- запрос с использованием условия отбора групп HAVING;

3.11 Вывести ФИО сотрудников, время прохождения теста у которых отличается от любого из других прохождений (Результат на рисунке 27)

SELECT MIN(ФИО) AS ФИО

FROM Сотрудники JOIN [Прохождения тестов] ON Сотрудники.[ID Сотрудника]=[Прохождения тестов].[ID Сотрудника]

GROUP BY [Время прохождения теста]

HAVING COUNT([Время прохождения теста])=1

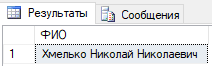


Рисунок – запрос 3.11

- запрос с использованием сортировки по столбцу;

3.12 Вывести все данные о вопросах с создателями в алфавитном порядке (Результат на рисунке 28)

SELECT \*

FROM Вопросы

ORDER BY Вопросы.Создатель

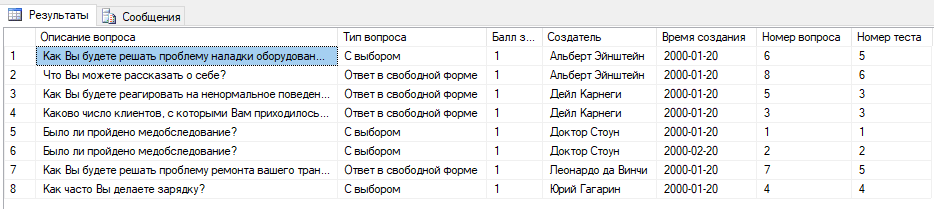


Рисунок – запрос 3.12

1. Запросы на модификацию таблиц:

- запрос на добавление новых данных в таблицу;

* 1. Добавить профессию штатного медика с зарплатой 200 000 (Результат на рисунке 29)

INSERT INTO Профессия VALUES

('Штатный медик',200000)

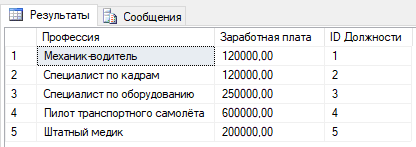


Рисунок – Таблица «Профессия» после выполнения запроса 4.1

- запрос на добавление новых данных по результатам выполненного запроса;

4.2 Добавить четвёртый вопрос в первый тест (Результат на рисунке 30)

INSERT INTO Вопросы

SELECT [Описание вопроса], [Тип вопроса], [Балл за верный ответ], Создатель, [Время создания], 1 AS [Номер теста]

FROM Вопросы

WHERE [Номер вопроса]=4

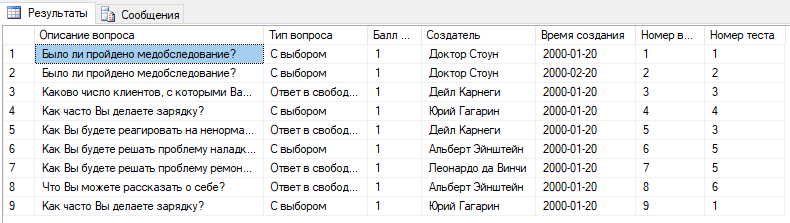


Рисунок – Таблица «Вопросы» после выполнения запроса 4.2

- запрос на обновление существующих данных в таблице ;

4.3 Всем сотрудникам с зарплатой меньше 200 000 поднять зарплату до 200 000 (Результат на рисунке 31)

UPDATE Профессия

SET [Заработная плата]=200000

WHERE [Заработная плата]<200000

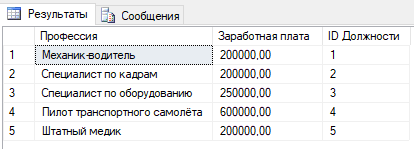


Рисунок – Таблица "Профессия" после выполнения запроса 4.3

- запрос на обновление существующих данных по результатам подзапроса во фразе WHERE;

4.4 Убрать служебный телефон сотрудника у всех, проходивших тесты с номерами меньше 3 (Результат на рисунке 32)

UPDATE Сотрудники

SET [Служебный телефон сотрудника]=NULL

WHERE [Служебный телефон сотрудника] IN (

SELECT [Служебный телефон сотрудника]

FROM Сотрудники JOIN [Прохождения тестов] ON Сотрудники.[ID Сотрудника]=[Прохождения тестов].[ID Сотрудника]

WHERE [Номер теста]<3

)

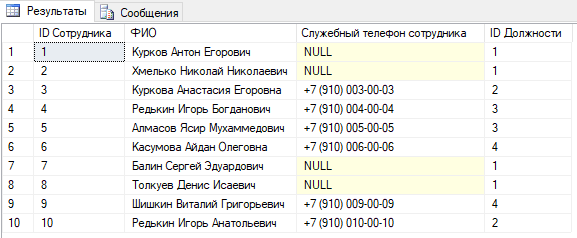


Рисунок – Таблица "Сотрудники" после выполнения запроса 4.4

- запрос на удаление существующих данных;

4.5 Удалить девятый вопрос (Результат на рисунке 33)

DELETE

FROM Вопросы

WHERE [Номер вопроса]=9

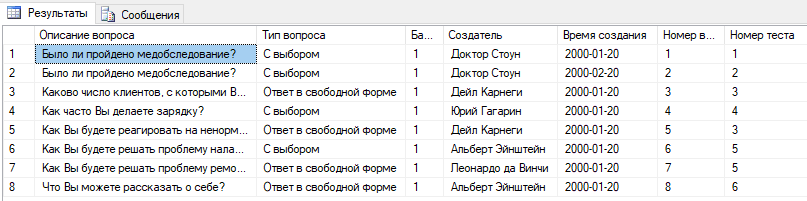


Рисунок – Таблица "Вопросы" после удаления запроса 4.5

- запрос на удаление существующих данных из одной таблицы на основе связанных с ней таблиц.

4.6 Удалить все прохождения тестов, завершённые не Редькиным (Результат на рисунке 34)

DELETE [Прохождения тестов]

FROM [Прохождения тестов] JOIN Сотрудники ON [Прохождения тестов].[ID Сотрудника]=Сотрудники.[ID Сотрудника]

WHERE ФИО NOT LIKE ('Редькин%')

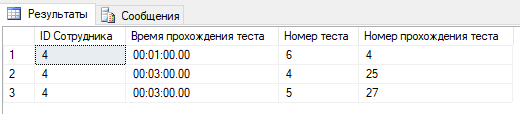


Рисунок – Таблица "Прохождения тестов" после выполнения запроса 4.6

- запрос на удаление существующих данных по результатам подзапроса во фразе WHERE;

4.7 Удалить все вопросы из тестов, где автором является Леонардо да Винчи (Результат на рисунке 35)

DELETE

FROM Вопросы

WHERE [Номер теста] IN (

SELECT Вопросы.[Номер теста]

FROM Вопросы JOIN Тесты ON Вопросы.[Номер теста]=Тесты.[Номер теста]

WHERE Тесты.Создатель='Леонардо да Винчи'

)

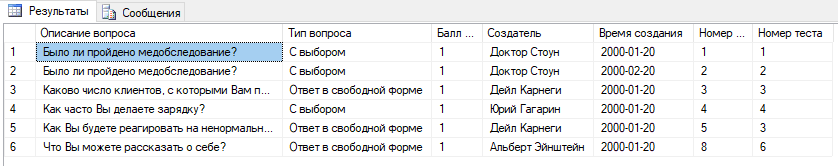


Рисунок – Таблица "Вопросы" после выполнения запроса 4.7

5. Многотабличные запросы с использованием подзапросов:

- 3 запроса с использованием операций сравнения;

5.1.1 Вывести все данные о сотрудниках, зарплата которых максимальна (Результат на рисунке 36)

SELECT Сотрудники.\*

FROM Сотрудники JOIN Профессия ON Сотрудники.[ID Должности]=Профессия.[ID Должности]

WHERE [Заработная плата]>=(

SELECT MAX([Заработная плата])

FROM Профессия)

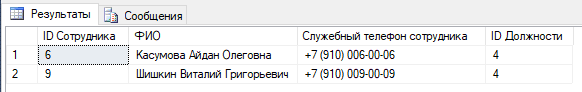


Рисунок –запрос 5.1.1

5.1.2 Вывести все данные о вопросах из тестов, тематика которых не находится среди тестов, номер которых находится в диапазоне от 3 до 6 (Результат на рисунке 37)

SELECT Вопросы.\*

FROM Вопросы JOIN Тесты ON Вопросы.[Номер теста]=Тесты.[Номер теста]

WHERE Тесты.[Тематика теста]<>ALL(

SELECT [Тематика теста]

FROM Тесты

WHERE [Номер теста] BETWEEN 3 AND 6

)

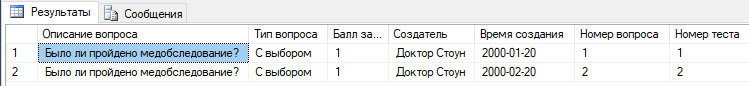


Рисунок – запрос 5.1.2

5.1.3 Вывести все данные о сотрудниках, которые проходили тест, номер которого равен номеру теста, содержащего вопрос «Как часто вы делаете зарядку?» (Результат на рисунке 38)

SELECT Сотрудники.\*

FROM Сотрудники JOIN [Прохождения тестов] ON Сотрудники.[ID Сотрудника]=[Прохождения тестов].[ID Сотрудника]

WHERE [Прохождения тестов].[Номер теста]=(

SELECT Тесты.[Номер теста]

FROM Вопросы JOIN Тесты ON Вопросы.[Номер теста]=Тесты.[Номер теста]

WHERE [Описание вопроса]='Как часто вы делаете зарядку?'

)

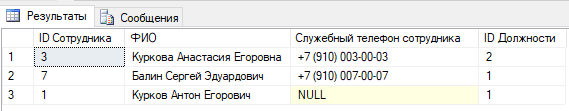


Рисунок – запрос 5.1.3

- запрос с использованием операции ANY;

5.2 Вывести сотрудников, которые прошли хотя бы один тест с вопросом от Юрия Гагарина (Результат на рисунке 39)

SELECT Сотрудники.\*

FROM Сотрудники JOIN [Прохождения тестов] ON Сотрудники.[ID Сотрудника]=[Прохождения тестов].[ID Сотрудника]

WHERE [Номер теста] = ANY(

SELECT [Номер теста]

FROM Вопросы

WHERE Создатель='Юрий Гагарин')

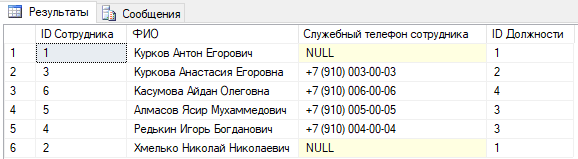


Рисунок – запрос 5.2

- запрос с использованием операции ALL;

5.3 Вывести сотрудников, зарплата которых отличается от зарплаты любого из Курковых (Результат на рисунке 40)

SELECT Сотрудники.\*

FROM Сотрудники JOIN Профессия ON Сотрудники.[ID Должности]=Профессия.[ID Должности]

WHERE [Заработная плата] <> ALL

(SELECT [Заработная плата]

FROM Сотрудники JOIN Профессия ON Сотрудники.[ID Должности]=Профессия.[ID Должности]

WHERE ФИО LIKE ('Курков%'))

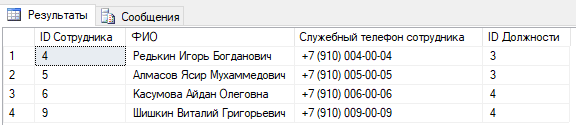


Рисунок – запрос 5.3

- запрос с использованием операции IN;

5.4 Вывести тесты, в которых есть вопросы того же типа, что и вопросы от Юрия Гагарина (Результат на рисунке 41)

SELECT Тесты.\*

FROM Тесты JOIN Вопросы ON Тесты.[Номер теста]=Вопросы.[Номер теста]

WHERE [Тип вопроса] IN

(SELECT [Тип вопроса]

FROM Вопросы

WHERE Вопросы.Создатель='Юрий Гагарин'

)

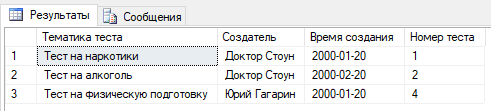


Рисунок – запрос 5.4

- запрос с использованием операции EXISTS;

5.5 Вывести сотрудников, не прошедших ни одного теста (Результат на рисунке 42)

SELECT Сотрудники.\*

FROM Сотрудники

WHERE NOT EXISTS

(SELECT \*

FROM [Прохождения тестов]

WHERE [Прохождения тестов].[ID Сотрудника]=Сотрудники.[ID Сотрудника]

)

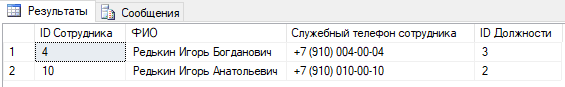


Рисунок – запрос 5.5

- запрос с использованием двух вложенных друг в друга подзапросов;

5.6 Вывести все данные о профессиях сотрудников, зарплата у которых отличается от зарплаты сотрудников, не сдавших ни одного теста (Результат на рисунке 43)

SELECT Профессия.\*

FROM Профессия

WHERE [Заработная плата]<>ALL(

SELECT [Заработная плата]

FROM Сотрудники JOIN Профессия ON Сотрудники.[ID Должности]=Профессия.[ID Должности]

WHERE NOT EXISTS

(SELECT \*

FROM [Прохождения тестов]

WHERE [Прохождения тестов].[ID Сотрудника]=Сотрудники.[ID Сотрудника]

)

)

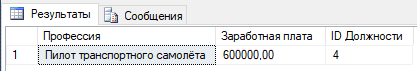


Рисунок – запрос 5.6

- запрос с использованием трех вложенных друг в друга подзапросов;

5.7 Вывести данные сотрудников, зарплата которых меньше любого из сотрудников, прошедших тест не медленнее, чем сотрудники, прошедшие любой из тестов быстрее пилотов транспортных самолётов, не учитывая результаты пятого теста. (Результат на рисунке 44)

SELECT Сотрудники.\*

FROM Сотрудники JOIN Профессия ON Сотрудники.[ID Должности]=Профессия.[ID Должности]

WHERE [Заработная плата]<=ALL(SELECT [Заработная плата]

FROM Профессия JOIN Сотрудники ON Профессия.[ID Должности]=Сотрудники.[ID Должности] JOIN [Прохождения тестов] ON Сотрудники.[ID Сотрудника]=[Прохождения тестов].[ID Сотрудника]

WHERE [Время прохождения теста]<ALL(SELECT MAX([Время прохождения теста])

FROM (SELECT Профессия.\*, [Прохождения тестов].\*

FROM Профессия JOIN Сотрудники ON Профессия.[ID Должности]=Сотрудники.[ID Должности] JOIN [Прохождения тестов] ON Сотрудники.[ID Сотрудника]=[Прохождения тестов].[ID Сотрудника]

WHERE Профессия.Профессия='Пилот транспортного самолёта'

) AS Таблица

WHERE Таблица.[Номер теста]<>5

) AND [Номер теста]<>5

)

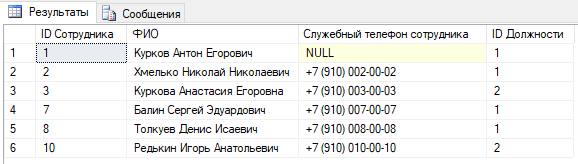


Рисунок – запрос 5.7

6. Запросы на изменение структуры существующей таблицы:

- запрос на добавление нового столбца к таблице;

6.1 Добавить столбец с набранными за тест баллами (Результат на рисунке 45)

ALTER TABLE [Прохождения тестов]

ADD [Набранные баллы] int



Рисунок – запрос 6.1

- запрос на добавление нового ограничения к таблице;

6.2 Добавить ограничение на ввод только неотрицательного числа баллов за тест (Результат на рисунке 46)

ALTER TABLE [Прохождения тестов]

ADD CONSTRAINT Только\_неотрицательные\_баллы check ([Набранные баллы]>=0)



Рисунок – запрос 6.2

- запрос на удаление нового столбца к таблице;

6.3 Удалить столбец с набранными за тест баллами (Результат на рисунке 47)

ALTER TABLE [Прохождения тестов]

DROP COLUMN [Набранные баллы]



Рисунок – запрос 6.3

- запрос на удаление нового ограничения к таблице;

6.4 Удалить ограничение на ввод только неотрицательного числа баллов за тест (Результат на рисунке 48)

ALTER TABLE [Прохождения тестов]

DROP CONSTRAINT Только\_неотрицательные\_баллы



Рисунок – запрос 6.4

- запрос на изменение типа данных в таблице;

6.5 Изменить тип набранных баллов на float (Результат на рисунке 49)

ALTER TABLE [Прохождения тестов]

ALTER COLUMN [Набранные баллы] float



Рисунок – запрос 6.5

- запрос на добавление нового первичного ключа к таблице с сохранением целостности таблицы

6.6 Сделать номер прохождения первичным ключом (Результат на рисунке 50)

ALTER TABLE [Прохождения тестов]

ADD [Набранные баллы] int NOT NULL DEFAULT 0

ALTER TABLE [Прохождения тестов]

DROP CONSTRAINT PK\_Прохождения

ALTER TABLE [Прохождения тестов]

ADD CONSTRAINT UQ\_ID\_Время\_Номер UNIQUE ([ID Сотрудника], [Время прохождения теста], [Номер теста])

ALTER TABLE [Прохождения тестов]

ADD CONSTRAINT PK\_Прохождения PRIMARY KEY ([Номер прохождения теста])



Рисунок – запрос 6.6

ЗАКЛЮЧЕНИЕ

Результатом курсовой работы стала база данных для фирмы, которую можно дополнить, усовершенствовать или адаптировать для различных коммерческих и некоммерческих предприятий.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Базы данных: методические указания к курсовой работе/ Рязан. гос. радиотехн. ун-т; сост.: Н.Н. Гринченко, Н.И. Хизриева, С.Н. Баранова. – Рязань, 2020. – 23 с.
2. Документация по Microsoft SQL: сайт. – URL: <https://learn.microsoft.com/ru-ru/sql/?view=sql-server-ver16> (дата обращения: 22.12.2022). – Текст: электронный.
3. Электронный курс "Базы данных. Язык SQL": сайт. – URL: https://cdo.rsreu.ru (дата обращения: 22.12.2022). – Текст: электронный.
4. Электронный курс "Программирование в СУБД Microsoft SQL Server": сайт. – URL: https://cdo.rsreu.ru (дата обращения: 22.12.2022). – Текст: электронный.