

ИТОГОВЫЙ ОТЧЕТ

Крючков Василий

4 курс, 7 группа

Предметная область: компания и сотрудники

Описание предметной области:

В компании есть несколько отделов. Каждый отдел имеет уникальную идентификацию, название, местонахождение офиса и конкретного сотрудника, который управляет отделом. Отдел контролирует ряд проектов, каждый из которых имеет уникальное имя, уникальный номер и бюджет. У каждого сотрудника есть имя, идентификационный номер, адрес, зарплата и дата рождения. Сотрудник назначен в один отдел, но может участвовать в нескольких проектах. необходимо вести учёт дату начала работы сотрудника в каждом проекте, непосредственного руководителя каждого сотрудника. Необходимо хранить информацию об иждивенцах каждого сотрудника. У каждого иждивенца есть имя, дата рождения и отношения с работником.

В соответствии с предметной областью система строится с учётом следующих особенностей:

- Каждый отдел имеет конкретного сотрудника, который управляет отделом.
- Отдел контролирует ряд проектов.
- Сотрудник назначен в один отдел, но может участвовать в нескольких проектах.
- Необходимо хранить информацию об иждивенцах каждого сотрудника.

Выделим базовые сущности этой предметной области:

- Отделы атрибуты: уникальная идентификация, название, местонахождение офиса, руководитель отдела.
- Проекты атрибуты: уникальное имя, уникальный номер, бюджет.
- Сотрудники атрибуты: имя, идентификационный номер, адрес, зарплата, дата рождения; необходимо вести учёт дату начала работы сотрудника в каждом проекте, непосредственного руководителя каждого сотрудника.
- Иждивенцы атрибуты: попечитель, имя, дата рождения, отношение с работником.

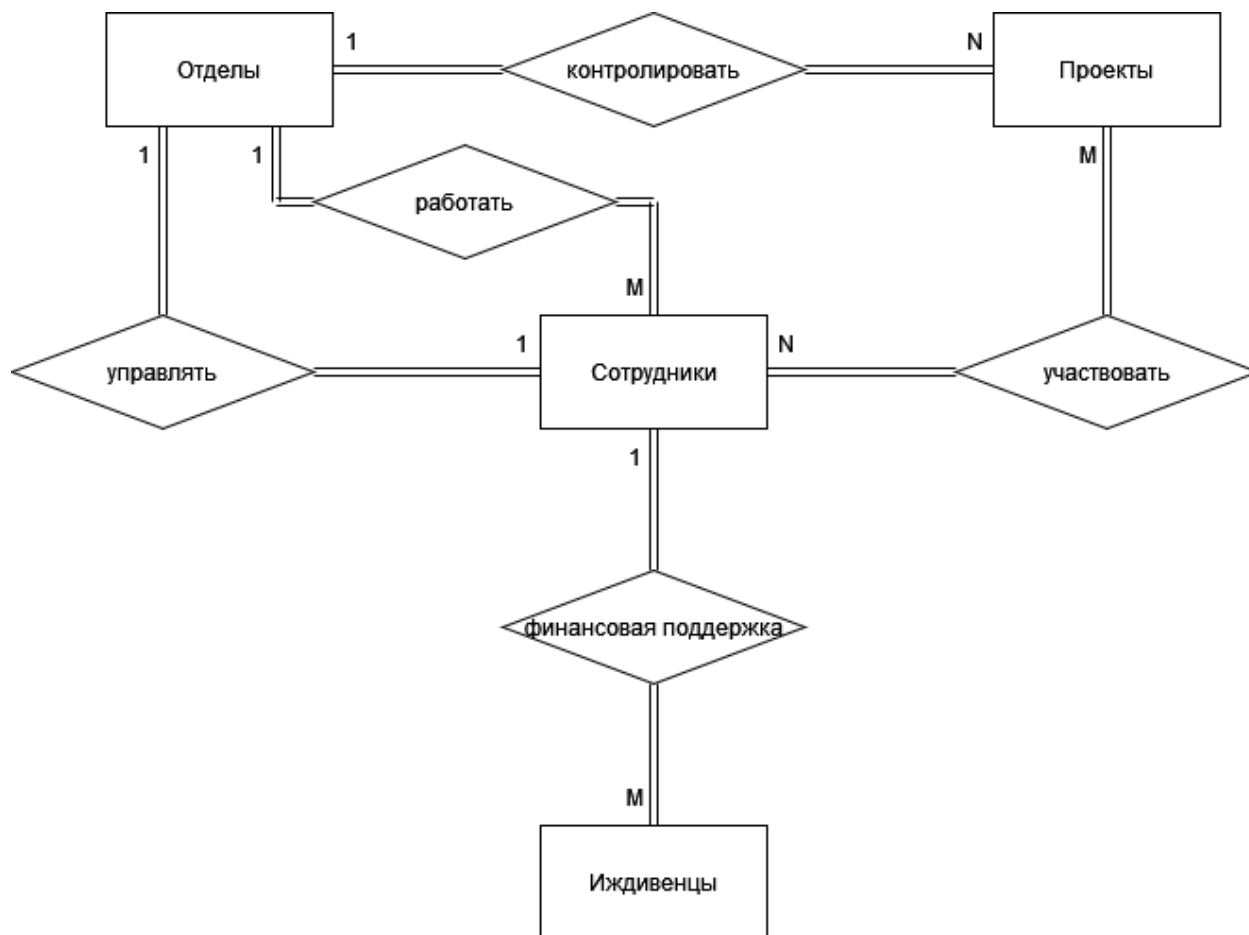


Рис. 1.1 – ER-диаграмма компании и сотрудников

Преобразование ER–диаграммы в схему базы данных

База данных создаётся на основании схемы базы данных. Для преобразования ER–диаграммы в схему БД приведём уточнённую ER–диаграмму, содержащая атрибуты сущностей.

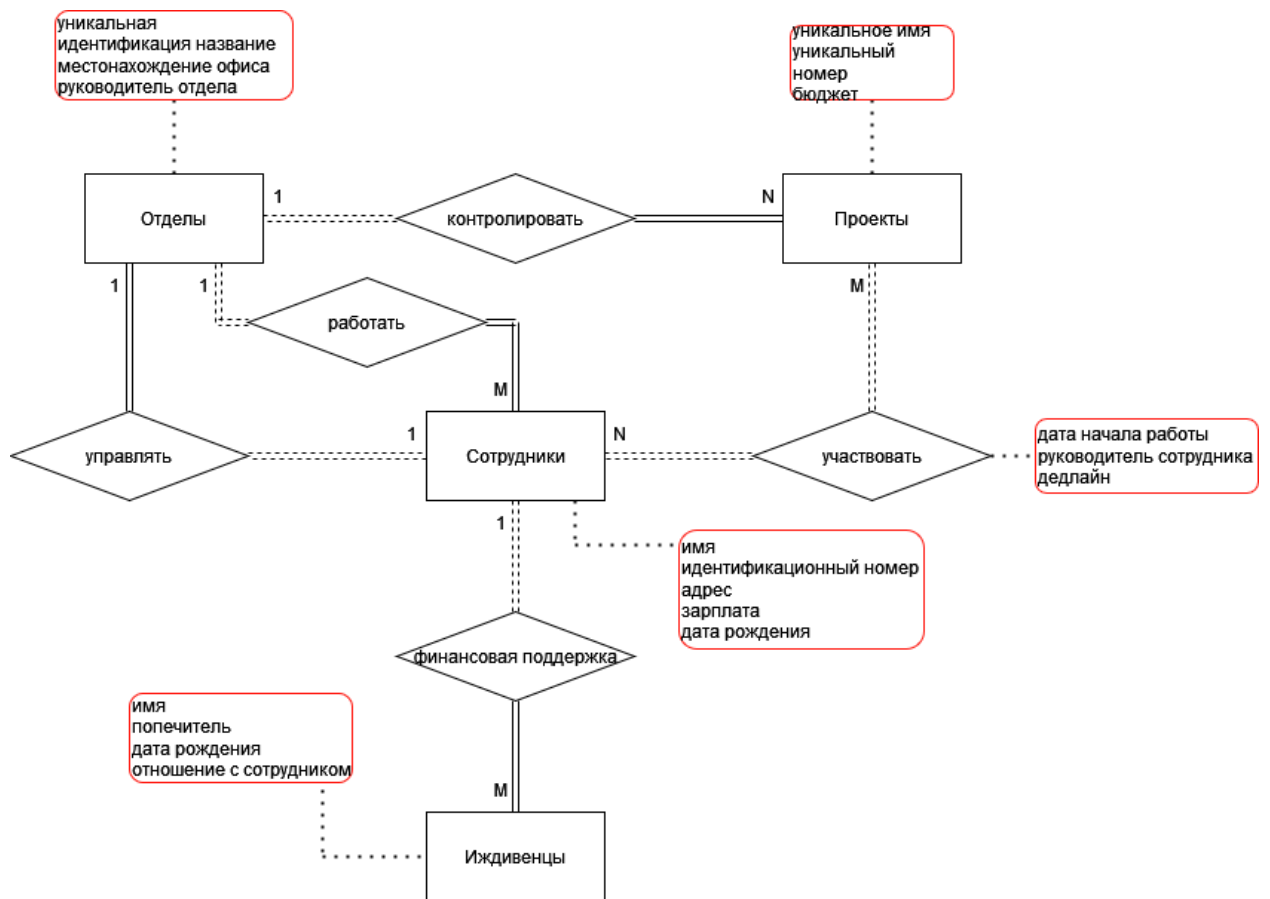
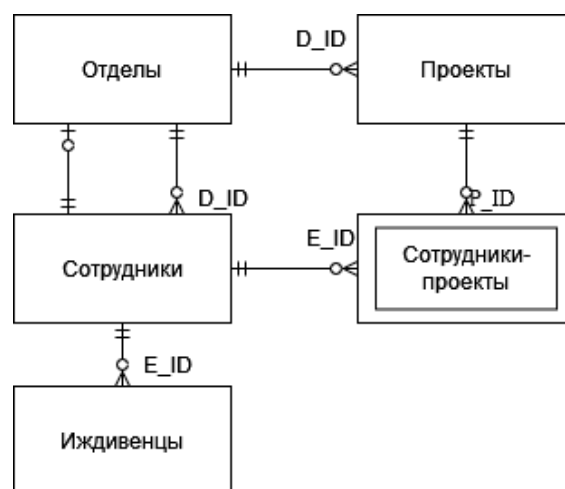


Рис. 1.2 – Уточненная ER-диаграмма компании и сотрудников

Логическое проектирование реляционной БД

Преобразование ER-диаграммы в схему БД выполняется путем сопоставления каждой сущности и каждой связи, имеющей атрибуты, отношения (таблицы БД).



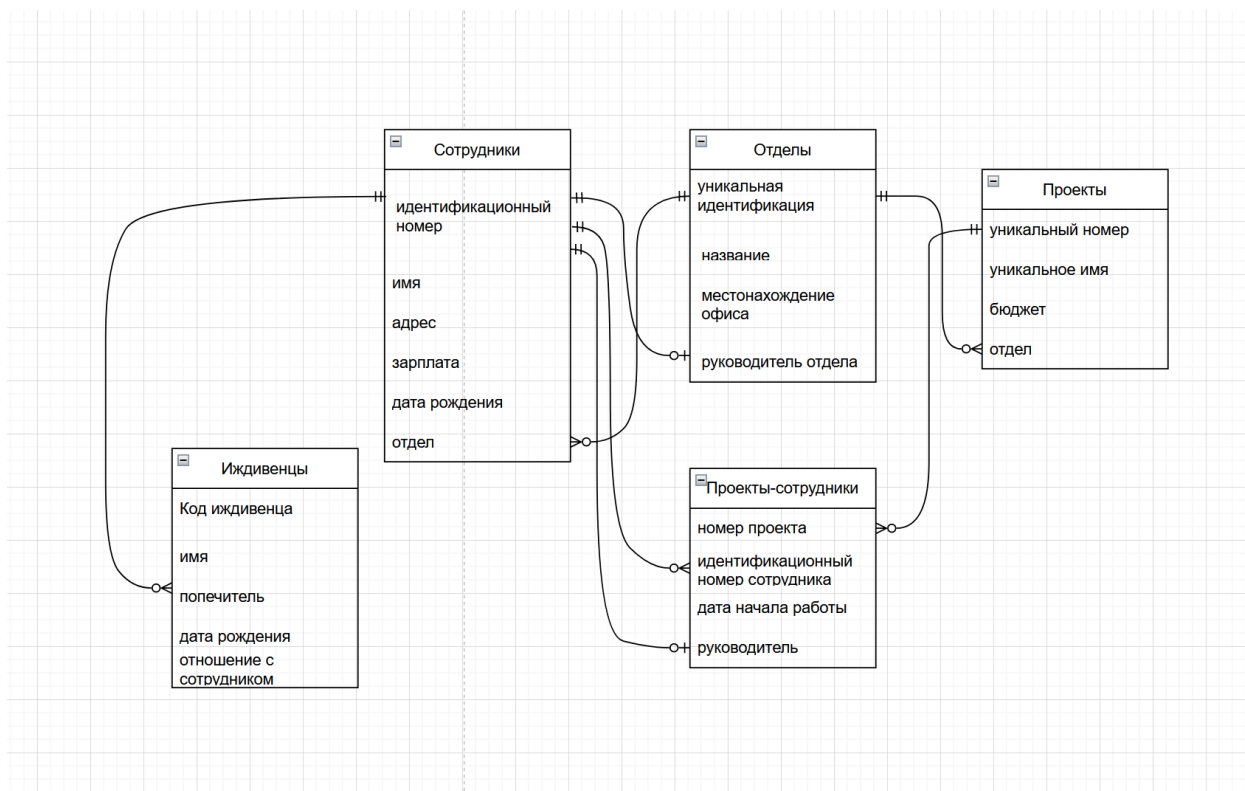


Рис. 1.3 – Схемы реляционной БД компании и сотрудников

Составление реляционных отношений

Таблица 1.1 – Схема отношения СОТРУДНИКИ (Employees)

Содержание поля	Имя поля	Тип, длина	Примечания
идентификационный номер	E_ID	N(4)	первичный ключ
имя	E_NAME	C(50)	обязательное поле
адрес	E_ADRESS	C(50)	
зарплата	E_SALARY	N(8,2)	обязательное поле
дата рождения	E_BORN	D	обязательное поле
отдел	D_ID	N(4)	внешний ключ (к Departments)

Таблица 1.2 – Схема отношения ОТДЕЛЫ (Departments)

Содержание поля	Имя поля	Тип, длина	Примечания
уникальная идентификация	D_ID	N(4)	первичный ключ
название	D_NAME	C(50)	обязательное поле
местонахождение офиса	D_ADRESS	C(50)	обязательное поле
руководитель отдела	D_HEAD	N(4)	внешний ключ (к Employees)

Таблица 1.3 – Схема отношения ПРОЕКТЫ (Projects)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
уникальный номер		N(4)	первичный ключ
уникальное имя	P_NAME	C(50)	обязательное уникальное поле
бюджет	P_BUDGET	N(12,2)	обязательное поле
отдел	D_ID	N(4)	внешний ключ (к Departments)

Таблица 1.4 – Схема отношения ПРОЕКТЫ-СОТРУДНИКИ (Participation)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
номер проекта	P_ID	N(4)	внешний ключ (к Projects)
идентификационный номер сотрудника	E_ID	N(4)	внешний ключ (к Employees)
дата начала работы	E_START	D	обязательное поле
руководитель	E_HEAD	N(4)	внешний ключ (к Employees)

Таблица 1.5 – Схема отношения ИЖДИВЕНЦЫ (Dependents)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Код иждивенца	DE_ID	N(4)	суррогатный первичный ключ
имя	DE_NAME	C(50)	внешний ключ (к Projects)
попечитель	E_ID	N(4)	внешний ключ (к Employees)
дата рождения	DE_BORN	C(50)	обязательное поле
отношение с сотрудником	DE_STATUS	C(50)	обязательное поле

Лабораторная работа №3

Задание:

Создать на языке T-SQL БД своей выбранной предметной области (по индивидуальному заданию).

Создать и описать на языке T-SQL наборы данных (таблицы) спроектированной вами реляционной БД предметной области, отношения, которых должны быть у вас согласно Лаб. Раб. 2. описаны в 3НФ.

Создать диаграмму (диаграммы) БД в SSMS, созданных вами
вышетаблиц данных на языке T-SQL.

Заполнить БД, наборы данных, данные ввести используя инструкцию
INSERT [INTO] имя_таблицы [(список_столбцов)] VALUES (значение1,
значение2, ... значениеN).

Сделать резервную копию созданной вами БД и прикрепить к отчету.

Создание таблиц:

```
1  -- Проверим наличие таблицы Participations и выполняем действия, если она существует
2  IF OBJECT_ID('dbo.Participations', 'U') IS NOT NULL
3  BEGIN
4      -- Удаляем внешние ключи, если они существуют
5      ALTER TABLE Participations DROP CONSTRAINT IF EXISTS FK_Projects
6      ALTER TABLE Participations DROP CONSTRAINT IF EXISTS FK_Employees_Participation
7      ALTER TABLE Participations DROP CONSTRAINT IF EXISTS FK_Employees_Head
8      -- Удаляем таблицу Participations
9      DROP TABLE Participations
10 END;
11
12 -- Проверим наличие таблицы Dependents и выполняем действия, если она существует
13 IF OBJECT_ID('dbo.Dependents', 'U') IS NOT NULL
14 BEGIN
15     -- Удаляем внешний ключ, если он существует
16     ALTER TABLE Dependents DROP CONSTRAINT IF EXISTS FK_Employees
17     -- Удаляем таблицу Dependents
18     DROP TABLE Dependents
19 END;
20
21 -- Проверим наличие таблицы Projects и выполняем действия, если она существует
22 IF OBJECT_ID('dbo.Projects', 'U') IS NOT NULL
23 BEGIN
24     -- Удаляем внешний ключ, если он существует
25     ALTER TABLE Projects DROP CONSTRAINT IF EXISTS FK_Departments_Projects
26     -- Удаляем таблицу Projects
27     DROP TABLE Projects
28 END;
29
30 -- Проверим наличие таблицы Departments и удаляем внешний ключ, если он существует
31 IF OBJECT_ID('dbo.Departments', 'U') IS NOT NULL
32     ALTER TABLE Departments DROP CONSTRAINT IF EXISTS FK_Departments_Head;
33
34 -- Проверим наличие таблицы Employees и удаляем внешний ключ, если он существует
35 IF OBJECT_ID('dbo.Employees', 'U') IS NOT NULL
36     ALTER TABLE Employees DROP CONSTRAINT IF EXISTS FK_Departments;
37
38 -- Удаляем таблицу Departments, если она существует
39 IF OBJECT_ID('dbo.Departments', 'U') IS NOT NULL
40     DROP TABLE Departments;
41
42 -- Удаляем таблицу Employees, если она существует
43 IF OBJECT_ID('dbo.Employees', 'U') IS NOT NULL
44     DROP TABLE Employees;
45
46 -- Создаем таблицу Departments с указанными столбцами
47 CREATE TABLE Departments (
48     D_ID INT PRIMARY KEY,
49     D_NAME NVARCHAR(50) NOT NULL,
50     D_ADDRESS NVARCHAR(50) NOT NULL,
51     D_HEAD INT NULL
52 );
```

```

54 -- Создаем таблицу Employees с указанными столбцами и внешним ключом, связывающим с таблицей Departments
55 CREATE TABLE Employees (
56     E_ID INT PRIMARY KEY,
57     E_NAME NVARCHAR(50) NOT NULL,
58     E_ADDRESS NVARCHAR(50),
59     E_SALARY DECIMAL(8, 2) NOT NULL CHECK (E_SALARY >= 0),
60     E_BORN DATE NOT NULL,
61     D_ID INT NOT NULL,
62     CONSTRAINT FK_Departments FOREIGN KEY (D_ID) REFERENCES Departments(D_ID) ON DELETE CASCADE
63 );
64
65 -- Добавляем внешний ключ в таблицу Departments, связывающий D_HEAD с E_ID в таблице Employees
66 ALTER TABLE Departments
67 ADD CONSTRAINT FK_Employees FOREIGN KEY (D_HEAD) REFERENCES Employees(E_ID) ON DELETE NO ACTION;
68
69 -- Создаем таблицу Projects с указанными столбцами и внешним ключом, связывающим с таблицей Departments
70 CREATE TABLE Projects (
71     P_ID INT PRIMARY KEY,
72     P_NAME NVARCHAR(50) UNIQUE NOT NULL,
73     P_BUDGET DECIMAL(12, 2) NOT NULL CHECK (P_BUDGET >= 0),
74     D_ID INT NOT NULL,
75     CONSTRAINT FK_Departments_Projects FOREIGN KEY (D_ID) REFERENCES Departments(D_ID) ON DELETE CASCADE
76 );
77
78 -- Создаем таблицу Dependents с указанными столбцами и внешним ключом, связывающим E_ID с Employees
79 CREATE TABLE Dependents (
80     DE_ID INT IDENTITY(1, 1) PRIMARY KEY,
81     DE_NAME NVARCHAR(50),
82     E_ID INT NOT NULL,
83     DE_BORN DATE NOT NULL,
84     DE_STATUS NVARCHAR(50) NOT NULL,
85     CONSTRAINT FK_Employees_Dependents FOREIGN KEY (E_ID) REFERENCES Employees(E_ID) ON DELETE CASCADE
86 );
87
88 -- Создаем таблицу Participations с указанными столбцами и внешними ключами, связывающими P_ID, E_ID и E_HEAD с другими таблицами
89 CREATE TABLE Participations (
90     P_ID INT NOT NULL,
91     E_ID INT NOT NULL,
92     E_START DATE NOT NULL,
93     E_HEAD INT NOT NULL,
94     CONSTRAINT PK_Participations PRIMARY KEY (P_ID, E_ID),
95     CONSTRAINT FK_Participations_Projects FOREIGN KEY (P_ID) REFERENCES Projects(P_ID) ON DELETE CASCADE,
96     CONSTRAINT FK_Participations_Employees FOREIGN KEY (E_ID) REFERENCES Employees(E_ID) ON DELETE NO ACTION,
97     CONSTRAINT FK_Participations_Head FOREIGN KEY (E_HEAD) REFERENCES Employees(E_ID) ON DELETE NO ACTION
98 );

```

Заполнение таблиц:

```

1 use Ucheb_7_Kryuchkov;
2
3 ALTER TABLE Employees DROP CONSTRAINT IF EXISTS FK_Departments;
4
5 INSERT Employees (E_ID, E_NAME, E_ADDRESS, E_SALARY, E_BORN, D_ID)
6 VALUES (1, 'John Doe', '123 Main St', 50000.00, '1990-05-15', 1),
7         (2, 'Jane Smith', '456 Elm St', 60000.00, '1988-02-28', 4),
8         (3, 'Alice Johnson', '789 Oak St', 55000.00, '1992-11-10', 8),
9         (4, 'Bob Brown', '101 Pine St', 48000.00, '1995-07-20', 7),
10        (5, 'Eva White', '202 Maple St', 52000.00, '1985-04-03', 10),
11        (6, 'Michael Davis', '303 Cedar St', 65000.00, '1993-09-18', 3),
12        (7, 'Olivia Lee', '404 Birch St', 70000.00, '1987-12-07', 5),
13        (8, 'Daniel Wilson', '505 Spruce St', 75000.00, '1984-08-25', 6),
14        (9, 'Sophia Martinez', '606 Ash St', 48000.00, '1991-03-14', 9),
15        (10, 'Liam Thomas', '707 Fir St', 55000.00, '1994-06-29', 1),
16        (11, 'Sarah Johnson', '808 Pine St', 62000.00, '1993-08-12', 2),
17        (12, 'David Lee', '909 Oak St', 58000.00, '1994-02-19', 3),
18        (13, 'Megan Brown', '101 Elm St', 56000.00, '1991-06-25', 4),
19        (14, 'Kevin Miller', '202 Cedar St', 59000.00, '1989-11-07', 5),
20        (15, 'Amanda Wilson', '303 Spruce St', 64000.00, '1990-09-15', 6),
21        (16, 'Christopher Hall', '404 Birch St', 67000.00, '1987-04-28', 7),
22        (17, 'Emma Martinez', '505 Maple St', 60000.00, '1992-12-04', 8),
23        (18, 'Ryan Taylor', '606 Ash St', 57000.00, '1988-07-21', 9),
24        (19, 'Grace Robinson', '707 Fir St', 61000.00, '1995-03-09', 10),
25        (20, 'Brandon Young', '808 Pine St', 63000.00, '1986-10-13', 2);
26
27
28 INSERT Departments (D_ID, D_NAME, D_ADDRESS, D_HEAD)
29 VALUES (1, 'Sales', '123 Sales St', 1),
30         (2, 'Marketing', '456 Marketing St', NULL),
31         (3, 'IT', '789 IT St', 3),
32         (4, 'Finance', '101 Finance St', 4),
33         (5, 'HR', '202 HR St', 5),
34         (6, 'Operations', '303 Operations St', 6),
35         (7, 'Research', '404 Research St', 7),
36         (8, 'Customer Service', '505 Customer Service St', 8),
37         (9, 'Product Development', '606 Product Development St', NULL),
38         (10, 'Quality Assurance', '707 QA St', 10);
39
40 ALTER TABLE Employees
41 ADD CONSTRAINT FK_Departments FOREIGN KEY (D_ID) REFERENCES Departments(D_ID) ON DELETE CASCADE;
42

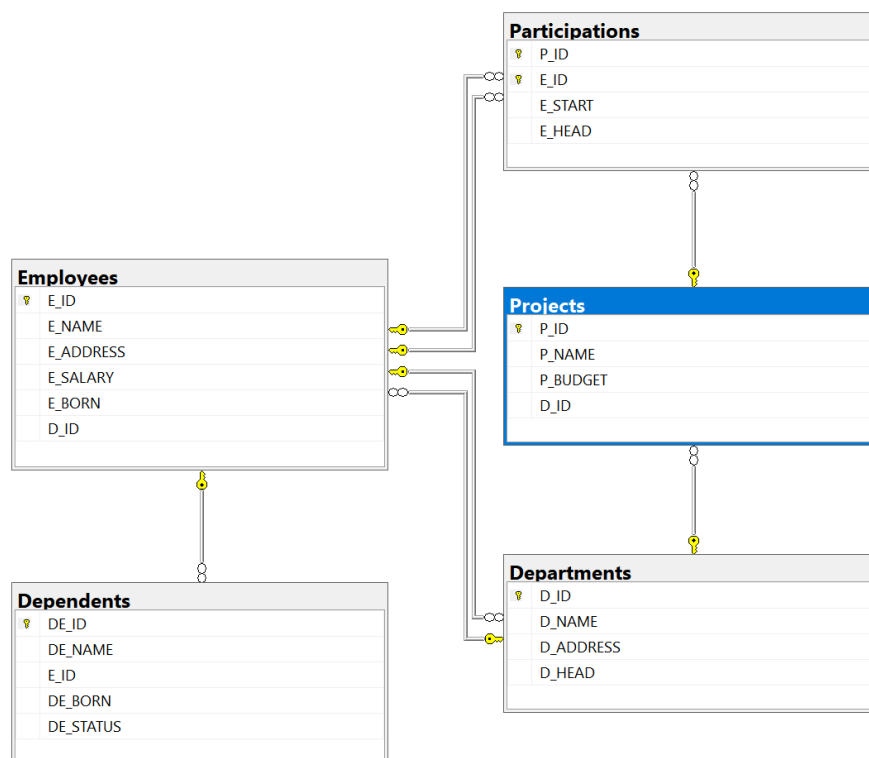
```

```

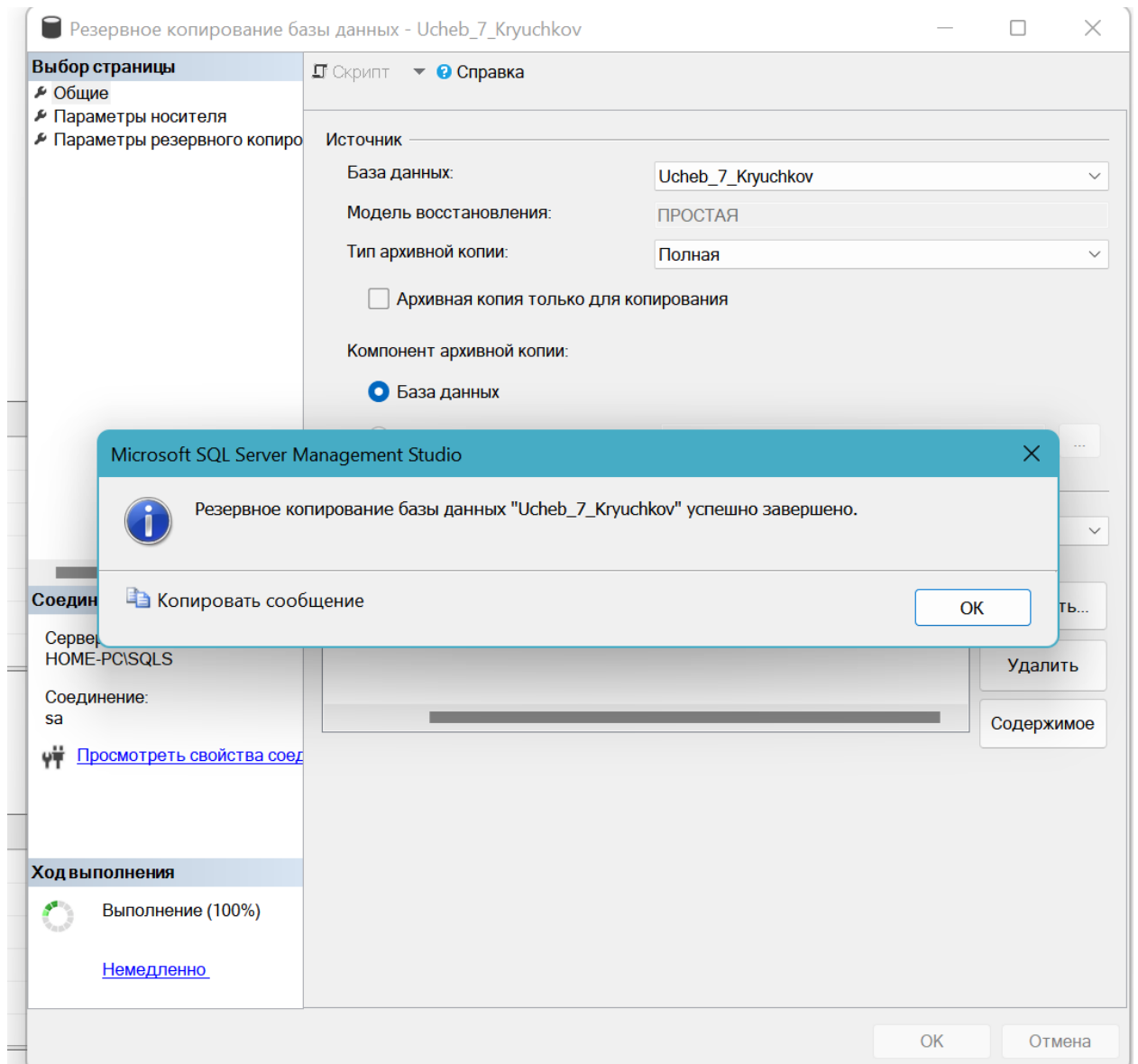
43
44 INSERT Projects (P_ID, P_NAME, P_BUDGET, D_ID)
45 VALUES (1, 'Project A', 10000.00, 1),
46          (2, 'Project B', 80000.00, 2),
47          (3, 'Project C', 120000.00, 3),
48          (4, 'Project D', 90000.00, 4),
49          (5, 'Project E', 95000.00, 5),
50          (6, 'Project F', 110000.00, 6),
51          (7, 'Project G', 85000.00, 7),
52          (8, 'Project H', 130000.00, 8),
53          (9, 'Project I', 70000.00, 9),
54          (10, 'Project J', 75000.00, 10);
55
56 INSERT Participations (P_ID, E_ID, E_START, E_HEAD)
57 VALUES (1, 1, '2023-01-15', 2),
58          (1, 3, '2023-01-20', 2),
59          (2, 2, '2023-02-01', 4),
60          (2, 4, '2023-02-05', 4),
61          (3, 5, '2023-03-10', 6),
62          (3, 6, '2023-03-12', 6),
63          (4, 7, '2023-04-03', 8),
64          (4, 8, '2023-04-05', 8),
65          (5, 9, '2023-05-20', 10),
66          (5, 10, '2023-05-25', 10);
67
68 INSERT Dependents (DE_NAME, E_ID, DE_BORN, DE_STATUS)
69 VALUES ('Child 1', 1, '2000-03-05', 'Son'),
70          ('Child 2', 3, '2002-07-12', 'Daughter'),
71          ('Spouse 1', 2, '1975-11-18', 'Spouse'),
72          ('Child 3', 7, '2005-01-30', 'Son'),
73          ('Child 4', 9, '2008-09-14', 'Daughter'),
74          ('Spouse 2', 6, '1980-06-22', 'Spouse'),
75          ('Child 5', 5, '1999-04-08', 'Son'),
76          ('Child 6', 4, '2003-12-19', 'Daughter'),
77          ('Child 7', 10, '2010-02-28', 'Son'),
78          ('Spouse 3', 8, '1978-09-07', 'Spouse');

```

Диаграмма построенной БД:



Резервное копирование:



Частичный вывод:

```
1 SELECT TOP (1000) [P_ID]
2      , [E_ID]
3      , [E_START]
4      , [E_HEAD]
5 FROM [Ucheb 7_Kryuchkov].[dbo].[Participations]
6
```

75 %

Результаты Сообщения

	P_ID	E_ID	E_START	E_HEAD
1	1	1	2023-01-15	2
2	1	3	2023-01-20	2
3	2	2	2023-02-01	4
4	2	4	2023-02-05	4
5	3	5	2023-03-10	6
6	3	6	2023-03-12	6
7	4	7	2023-04-03	8
8	4	8	2023-04-05	8
9	5	9	2023-05-20	10
10	5	10	2023-05-25	10

```
SQLQuery10.sql - H:\...Kryuchkov (sa (7/6))
SQLQuery9.sql - H:\...Kryuchkov (sa (7/4))

1 SELECT TOP (1000) [D_ID]
2     , [D_NAME]
3     , [D_ADDRESS]
4     , [D_HEAD]
5 FROM [Ucheb_7_Kryuchkov].[dbo].[Departments]
6
```

5 %

Результаты Сообщения

	D_ID	D_NAME	D_ADDRESS	D_HEAD
1	1	Sales	123 Sales St	1
2	2	Marketing	456 Marketing St	NULL
3	3	IT	789 IT St	3
4	4	Finance	101 Finance St	4
5	5	HR	202 HR St	5
6	6	Operations	303 Operations St	6
7	7	Research	404 Research St	7
8	8	Customer Service	505 Customer Service St	8
9	9	Product Development	606 Product Development St	NULL
10	10	Quality Assurance	707 QA St	10

```

1 SELECT TOP (1000) [E_ID]
2     , [E_NAME]
3     , [E_ADDRESS]
4     , [E_SALARY]
5     , [E_BORN]
6     , [D_ID]
7 FROM [Ucheb_7_Kryuchkov].[dbo].[Employees]
8

```

6

Результаты

Сообщения

E_ID	E_NAME	E_ADDRESS	E_SALARY	E_BORN	D_ID
1	John Doe	123 Main St	50000.00	1990-05-15	1
2	Jane Smith	456 Elm St	60000.00	1988-02-28	4
3	Alice Johnson	789 Oak St	55000.00	1992-11-10	8
4	Bob Brown	101 Pine St	48000.00	1995-07-20	7
5	Eva White	202 Maple St	52000.00	1985-04-03	10
6	Michael Davis	303 Cedar St	65000.00	1993-09-18	3
7	Olivia Lee	404 Birch St	70000.00	1987-12-07	5
8	Daniel Wilson	505 Spruce St	75000.00	1984-08-25	6
9	Sophia Martinez	606 Ash St	48000.00	1991-03-14	9
10	Liam Thomas	707 Fir St	55000.00	1994-06-29	1
11	Sarah Johnson	808 Pine St	62000.00	1993-08-12	2
12	David Lee	909 Oak St	58000.00	1994-02-19	3
13	Megan Brown	101 Elm St	56000.00	1991-06-25	4
14	Kevin Miller	202 Cedar St	59000.00	1989-11-07	5
15	Amanda Wilson	303 Spruce St	64000.00	1990-09-15	6
16	Christopher Hall	404 Birch St	67000.00	1987-04-28	7
17	Emma Martinez	505 Maple St	60000.00	1992-12-04	8
18	Ryan Taylor	606 Ash St	57000.00	1988-07-21	9
19	Grace Robinson	707 Fir St	61000.00	1995-03-09	10
20	Brandon Young	808 Pine St	63000.00	1986-10-13	2

С помощью языка T-SQL выполнить следующие запросы

- вывести все данные из таблиц

```

1  -- Вывести все данные из таблицы Departments
2  SELECT * FROM Departments;
3
4  -- Вывести все данные из таблицы Employees
5  SELECT * FROM Employees;
6
7  -- Вывести все данные из таблицы Projects
8  SELECT * FROM Projects;
9
0  -- Вывести все данные из таблицы Participations
1  SELECT * FROM Participations;
2
3  -- Вывести все данные из таблицы Dependents
4  SELECT * FROM Dependents;

```

Результаты Сообщения

	D_ID	D_NAME	D_ADDRESS	D_HEAD
1	1	Sales	123 Sales St	1
2	2	Marketing	456 Marketing St	NULL
3	3	IT	789 IT St	3
4	4	Finance	101 Finance St	4
5	5	HR	202 HR St	5
6	6	Operations	303 Operations St	6
7	7	Research	404 Research St	7
8	8	Customer Service	505 Customer Service St	8
9	9	Product Development	606 Product Development St	NULL
10	10	Quality Assurance	707 QA St	10

Результаты Сообщения

	E_ID	E_NAME	E_ADDRESS	E_SALARY	E_BORN	D_ID
1	1	John Doe	123 Main St	50000.00	1990-05-15	1
2	2	Jane Smith	456 Elm St	60000.00	1988-02-28	4
3	3	Alice Johnson	789 Oak St	55000.00	1992-11-10	8
4	4	Bob Brown	101 Pine St	48000.00	1995-07-20	7
5	5	Eva White	202 Maple St	52000.00	1985-04-03	10
6	6	Michael Davis	303 Cedar St	65000.00	1993-09-18	3
7	7	Olivia Lee	404 Birch St	70000.00	1987-12-07	5
8	8	Daniel Wilson	505 Spruce St	75000.00	1984-08-25	6
9	9	Sophia Martinez	606 Ash St	48000.00	1991-03-14	9
10	10	Liam Thomas	707 Fir St	55000.00	1994-06-29	1
11	11	Sarah Johnson	808 Pine St	62000.00	1993-08-12	2
12	12	David Lee	909 Oak St	58000.00	1994-02-19	3
13	13	Megan Brown	101 Elm St	56000.00	1991-06-25	4
14	14	Kevin Miller	202 Cedar St	59000.00	1989-11-07	5
15	15	Amanda Wilson	303 Spruce St	64000.00	1990-09-15	6
16	16	Christopher Hall	404 Birch St	67000.00	1987-04-28	7
17	17	Emma Martinez	505 Maple St	60000.00	1992-12-04	8
18	18	Ryan Taylor	606 Ash St	57000.00	1988-07-21	9
19	19	Grace Robinson	707 Fir St	61000.00	1995-03-09	10
20	20	Brandon Young	808 Pine St	63000.00	1986-10-13	2

Результаты Сообщения				
	P_ID	P_NAME	P_BUDGET	D_ID
1	1	Project A	100000.00	1
2	2	Project B	80000.00	2
3	3	Project C	120000.00	3
4	4	Project D	90000.00	4
5	5	Project E	95000.00	5
6	6	Project F	110000.00	6
7	7	Project G	85000.00	7
8	8	Project H	130000.00	8
9	9	Project I	70000.00	9
10	10	Project J	75000.00	10

Результаты Сообщения				
	P_ID	E_ID	E_START	E_HEAD
1	1	1	2023-01-15	1
2	1	3	2023-01-20	1
3	2	2	2023-02-01	NULL
4	2	4	2023-02-05	NULL
5	3	5	2023-03-10	3
6	3	6	2023-03-12	3
7	4	7	2023-04-03	4
8	4	8	2023-04-05	4
9	5	9	2023-05-20	5
10	5	10	2023-05-25	5

Результаты Сообщения					
	DE_ID	DE_NAME	E_ID	DE_BORN	DE_STATUS
1	1	Child 1	1	2000-03-05	Son
2	2	Child 2	3	2002-07-12	Daughter
3	3	Spouse 1	2	1975-11-18	Spouse
4	4	Child 3	7	2005-01-30	Son
5	5	Child 4	9	2008-09-14	Daughter
6	6	Spouse 2	6	1980-06-22	Spouse
7	7	Child 5	5	1999-04-08	Son
8	8	Child 6	4	2003-12-19	Daughter
9	9	Child 7	10	2010-02-28	Son
10	10	Spouse 3	8	1978-09-07	Spouse

- вывести определенные данные, отсортированные в алфавитном порядке по определенному полю

```

16 --Вывести данные из таблицы Employees, отсортированные по полю E_NAME в алфавитном порядке:
17 SELECT * FROM Employees ORDER BY E_NAME;

```

121 %

Результаты Сообщения

	E_ID	E_NAME	E_ADDRESS	E_SALARY	E_BORN	D_ID
1	3	Alice Johnson	789 Oak St	55000.00	1992-11-10	8
2	15	Amanda Wilson	303 Spruce St	64000.00	1990-09-15	6
3	4	Bob Brown	101 Pine St	48000.00	1995-07-20	7
4	20	Brandon Young	808 Pine St	63000.00	1986-10-13	2
5	16	Christopher Hall	404 Birch St	67000.00	1987-04-28	7
6	8	Daniel Wilson	505 Spruce St	75000.00	1984-08-25	6
7	12	David Lee	909 Oak St	58000.00	1994-02-19	3
8	17	Emma Martinez	505 Maple St	60000.00	1992-12-04	8
9	5	Eva White	202 Maple St	52000.00	1985-04-03	10
10	19	Grace Robinson	707 Fir St	61000.00	1995-03-09	10
11	2	Jane Smith	456 Elm St	60000.00	1988-02-28	4
12	1	John Doe	123 Main St	50000.00	1990-05-15	1
13	14	Kevin Miller	202 Cedar St	59000.00	1989-11-07	5
14	10	Liam Thomas	707 Fir St	55000.00	1994-06-29	1
15	13	Megan Brown	101 Elm St	56000.00	1991-06-25	4
16	6	Michael Davis	303 Cedar St	65000.00	1993-09-18	3
17	7	Olivia Lee	404 Birch St	70000.00	1987-12-07	5
18	18	Ryan Taylor	606 Ash St	57000.00	1988-07-21	9
19	11	Sarah Johnson	808 Pine St	62000.00	1993-08-12	2
20	9	Sophia Martinez	606 Ash St	48000.00	1991-03-14	9

- вывести определенные данные, отсортированные в обратном алфавитном порядке по двум полям

```

19 --Вывести данные из таблицы Participations, отсортированные в обратном алфавитном порядке:
20 SELECT * FROM Dependents ORDER BY DE_STATUS DESC, DE_NAME DESC;
21

```

110 %

Результаты Сообщения

	DE_ID	DE_NAME	E_ID	DE_BORN	DE_STATUS
1	10	Spouse 3	8	1978-09-07	Spouse
2	6	Spouse 2	6	1980-06-22	Spouse
3	3	Spouse 1	2	1975-11-18	Spouse
4	9	Child 7	10	2010-02-28	Son
5	7	Child 5	5	1999-04-08	Son
6	4	Child 3	7	2005-01-30	Son
7	1	Child 1	1	2000-03-05	Son
8	8	Child 6	4	2003-12-19	Daughter
9	5	Child 4	9	2008-09-14	Daughter
10	2	Child 2	3	2002-07-12	Daughter

- создать вычисляемое поле, и вывести данные, убрав дубликаты

```

22 --Создать вычисляемое поле на основе столбца и убрать дубликаты:
23 SELECT DISTINCT DE_STATUS FROM Dependents;

```

110 %

Результаты Сообщения

	DE_STATUS
1	Daughter
2	Son
3	Spouse

- вывести 30% строк

```

25  --Вывести 30% строк из таблицы Employees:
26  SELECT TOP 30 PERCENT * FROM Employees;

```

110 %

Результаты Сообщения

	E_ID	E_NAME	E_ADDRESS	E_SALARY	E_BORN	D_ID
1	1	John Doe	123 Main St	50000.00	1990-05-15	1
2	2	Jane Smith	456 Elm St	60000.00	1988-02-28	4
3	3	Alice Johnson	789 Oak St	55000.00	1992-11-10	8
4	4	Bob Brown	101 Pine St	48000.00	1995-07-20	7
5	5	Eva White	202 Maple St	52000.00	1985-04-03	10
6	6	Michael Davis	303 Cedar St	65000.00	1993-09-18	3

- вывести первые 5 строк

```

29  --Вывести первые 5 строк из таблицы Employees:
30  SELECT TOP 5 * FROM Employees;

```

110 %

Результаты Сообщения

	E_ID	E_NAME	E_ADDRESS	E_SALARY	E_BORN	D_ID
1	1	John Doe	123 Main St	50000.00	1990-05-15	1
2	2	Jane Smith	456 Elm St	60000.00	1988-02-28	4
3	3	Alice Johnson	789 Oak St	55000.00	1992-11-10	8
4	4	Bob Brown	101 Pine St	48000.00	1995-07-20	7
5	5	Eva White	202 Maple St	52000.00	1985-04-03	10

- вывести строки начиная с 4 строки по 7 строку, отсортированные в обратном порядке

```

32  --Вывести строки с 4 по 7, отсортированные в обратном порядке:
33  SELECT * FROM Employees
34  ORDER BY E_NAME DESC
35  OFFSET 3 ROWS
36  FETCH NEXT 4 ROWS ONLY;

```

110 %

Результаты Сообщения

	E_ID	E_NAME	E_ADDRESS	E_SALARY	E_BORN	D_ID
1	7	Olivia Lee	404 Birch St	70000.00	1987-12-07	5
2	6	Michael Davis	303 Cedar St	65000.00	1993-09-18	3
3	13	Megan Brown	101 Elm St	56000.00	1991-06-25	4
4	10	Liam Thomas	707 Fir St	55000.00	1994-06-29	1

- отсортировать данные, используя в качестве критерия сортировки сложно выражение на основе столбцов


```

38 | --отсортировать данные, используя в качестве критерия сортировки сложно выражение на основе столбцов
39 | SELECT *
40 | FROM Employees
41 | ORDER BY E_SALARY * DATEDIFF(YEAR, E_BORN, GETDATE()) DESC;
42 | --E_SALARY * DATEDIFF(YEAR, E_BORN, GETDATE()) - это сложное выражение,
43 | --которое умножает значение столбца E_SALARY на разницу в годах между датой в столбце E_BORN и текущей датой (GETDATE()).

```

	E_ID	E_NAME	E_ADDRESS	E_SALARY	E_BORN	D_ID
1	8	Daniel Wilson	505 Spruce St	75000.00	1984-08-25	6
2	7	Olivia Lee	404 Birch St	70000.00	1987-12-07	5
3	16	Christopher Hall	404 Birch St	67000.00	1987-04-28	7
4	20	Brandon Young	808 Pine St	63000.00	1986-10-13	2
5	15	Amanda Wilson	303 Spruce St	64000.00	1990-09-15	6
6	2	Jane Smith	456 Elm St	60000.00	1988-02-28	4
7	14	Kevin Miller	202 Cedar St	59000.00	1989-11-07	5
8	18	Ryan Taylor	606 Ash St	57000.00	1988-07-21	9
9	5	Eva White	202 Maple St	52000.00	1985-04-03	10
10	6	Michael Davis	303 Cedar St	65000.00	1993-09-18	3
11	17	Emma Martinez	505 Maple St	60000.00	1992-12-04	8
12	11	Sarah Johnson	808 Pine St	62000.00	1993-08-12	2
13	13	Megan Brown	101 Elm St	56000.00	1991-06-25	4
14	19	Grace Robinson	707 Fir St	61000.00	1995-03-09	10
15	3	Alice Johnson	789 Oak St	55000.00	1992-11-10	8
16	12	David Lee	909 Oak St	58000.00	1994-02-19	3
17	1	John Doe	123 Main St	50000.00	1990-05-15	1
18	10	Liam Thomas	707 Fir St	55000.00	1994-06-29	1
19	9	Sophia Martinez	606 Ash St	48000.00	1991-03-14	9
20	4	Bob Brown	101 Pine St	48000.00	1995-07-20	7

- запрос на выборку с добавлением SELECT INTO

```

45 | --Запрос на выборку с добавлением SELECT INTO (создать новую таблицу на основе выборки из Participations):
46 | SELECT * INTO NewTable FROM Dependents
47 | WHERE DE_STATUS = 'Son' OR DE_STATUS = 'Daughter';
48 | SELECT * FROM NewTable;

```

	DE_ID	DE_NAME	E_ID	DE_BORN	DE_STATUS
1	1	Child 1	1	2000-03-05	Son
2	2	Child 2	3	2002-07-12	Daughter
3	4	Child 3	7	2005-01-30	Son
4	5	Child 4	9	2008-09-14	Daughter
5	7	Child 5	5	1999-04-08	Son
6	8	Child 6	4	2003-12-19	Daughter
7	9	Child 7	10	2010-02-28	Son

С помощью языка T-SQL написать и выполнить 7 запросов на фильтрацию данных с использованием в запросах операторов сравнения, логических операторов:

```

50 | --Выборка сотрудников, у которых зарплата больше 60000 и которые работают в отделе с ID 1:
51 | SELECT *
52 | FROM Employees
53 | WHERE E_SALARY > 60000 AND D_ID = 2;

```

	E_ID	E_NAME	E_ADDRESS	E_SALARY	E_BORN	D_ID
1	11	Sarah Johnson	808 Pine St	62000.00	1993-08-12	2
2	20	Brandon Young	808 Pine St	63000.00	1986-10-13	2

```

56 | --Выборка сотрудников, которые работают в отделе с названием 'IT' или 'Sales':
57 | SELECT *
58 | FROM Employees
59 | WHERE D_ID IN (SELECT D_ID FROM Departments WHERE D_NAME IN ('IT', 'Sales'));

```

	E_ID	E_NAME	E_ADDRESS	E_SALARY	E_BORN	D_ID
1	1	John Doe	123 Main St	50000.00	1990-05-15	1
2	6	Michael Davis	303 Cedar St	65000.00	1993-09-18	3
3	10	Liam Thomas	707 Fir St	55000.00	1994-06-29	1
4	12	David Lee	909 Oak St	58000.00	1994-02-19	3

```

62 SELECT *
63 FROM Employees
64 WHERE E_SALARY > 60000 AND D_ID = (SELECT D_ID FROM Departments WHERE D_NAME = 'IT');
65
66

```

110 %

Результаты Сообщения

	E_ID	E_NAME	E_ADDRESS	E_SALARY	E_BORN	D_ID
1	6	Michael Davis	303 Cedar St	65000.00	1993-09-18	3

```

67 --Выборка проектов, у которых бюджет больше 100000 и на которые сотрудники зарегистрированы после 2019 года
68 SELECT *
69 FROM Projects
70 WHERE P_BUDGET > 100000
71 AND P_ID IN (SELECT P_ID FROM Participations WHERE YEAR(E_START) > 2019);
72
73

```

110 %

Результаты Сообщения

	P_ID	P_NAME	P_BUDGET	D_ID
1	3	Project C	120000.00	3

```

74 --Выборка проектов, у которых бюджет больше 50000 и на которые зарегистрированы сотрудники из отдела 'HR' или 'IT' или 'Marketing':
75 SELECT *
76 FROM Projects
77 WHERE P_BUDGET > 50000
78 AND P_ID IN (SELECT P_ID FROM Participations WHERE E_ID IN
79             (SELECT E_ID FROM Employees WHERE D_ID IN
80              (SELECT D_ID FROM Departments WHERE D_NAME IN ('HR', 'IT', 'Marketing'))));
81

```

100 %

Результаты Сообщения

	P_ID	P_NAME	P_BUDGET	D_ID
1	3	Project C	120000.00	3
2	4	Project D	90000.00	4

```

83 --Выборка сотрудников, у которых зарплата больше средней по всем отделам:
84 SELECT *
85 FROM Employees
86 WHERE E_SALARY > (SELECT AVG(E_SALARY) FROM Employees);

```

100 %

Результаты Сообщения

	E_ID	E_NAME	E_ADDRESS	E_SALARY	E_BORN	D_ID
1	2	Jane Smith	456 Elm St	60000.00	1988-02-28	4
2	6	Michael Davis	303 Cedar St	65000.00	1993-09-18	3
3	7	Olivia Lee	404 Birch St	70000.00	1987-12-07	5
4	8	Daniel Wilson	505 Spruce St	75000.00	1984-08-25	6
5	11	Sarah Johnson	808 Pine St	62000.00	1993-08-12	2
6	15	Amanda Wilson	303 Spruce St	64000.00	1990-09-15	6
7	16	Christopher Hall	404 Birch St	67000.00	1987-04-28	7
8	17	Emma Martinez	505 Maple St	60000.00	1992-12-04	8
9	19	Grace Robinson	707 Fir St	61000.00	1995-03-09	10
10	20	Brandon Young	808 Pine St	63000.00	1986-10-13	2

```

89 SELECT *
90 FROM Employees
91 WHERE LEFT(E_NAME, 1) = 'A' AND YEAR(E_BORN) > 1990;

```

100 %

Результаты Сообщения

	E_ID	E_NAME	E_ADDRESS	E_SALARY	E_BORN	D_ID
1	3	Alice Johnson	789 Oak St	55000.00	1992-11-10	8

Переименуйте одну из таблиц вашей БД

```
93  
94 --Переименование таблицы:  
95 EXEC sp_rename 'NewTable', 'Childs';
```

100 %

Сообщения

Внимание! Изменение любой части имени объекта может разрушить скрипты и хранимые процедуры.

Время выполнения: 2023-10-20T01:46:57.6447112+03:00

Добавьте по две записи в каждую из таблиц вашей БД

```
98 -- Добавление записей в таблицу Departments  
99 INSERT INTO Departments (D_ID, D_NAME, D_ADDRESS, D_HEAD)  
100 VALUES  
101 (11, 'Legal', '808 Legal St', 13),  
102 (12, 'Purchasing', '909 Purchasing St', 15);  
103  
104  
105 -- Добавление записей в таблицу Employees  
106 INSERT INTO Employees (E_ID, E_NAME, E_ADDRESS, E_SALARY, E_BORN, D_ID)  
107 VALUES  
108 (21, 'Olav Olsen', '111 Bjørk St', 65000.00, '1993-08-12', 1),  
109 (22, 'Ingrid Pedersen', '222 Gran St', 70000.00, '1994-02-19', 2);  
110  
111  
112 -- Добавление записей в таблицу Projects  
113 INSERT INTO Projects (P_ID, P_NAME, P_BUDGET, D_ID)  
114 VALUES  
115 (11, 'Project K', 100000.00, 3),  
116 (12, 'Project L', 80000.00, 11);  
117  
118  
119 -- Добавление записей в таблицу Participation  
120 INSERT INTO Participations (P_ID, E_ID, E_START, E_HEAD)  
121 VALUES  
122 (8, 15, '2023-01-01', 8),  
123 (12, 21, '2023-02-01', 13);  
124  
125  
126 -- Добавление записей в таблицу Dependents  
127 INSERT INTO Dependents (DE_NAME, E_ID, DE_BORN, DE_STATUS)  
128 VALUES  
129 ('Spouse', 14, '2000-01-01', 'Spouse'),  
130 ('Child 8', 20, '2001-02-01', 'Son');
```

100 %

Сообщения

(затронута строк: 2)

(затронута строк: 2)

(затронута строк: 2)

(затронута строк: 2)

(затронута строк: 2)

Измените одну из таблиц вашей БД:

- добавьте по смыслу один столбец
- добавьте ограничение с именем на новый введенный вами столбец

```
133 -- Добавление столбца Deadline
134 ALTER TABLE Participations
135 ADD E_DEADLINE DATE;
136
137 -- Добавление ограничения на столбец Deadline
138 ALTER TABLE Participations
139 ADD CONSTRAINT CHK_Decline CHECK (E_DEADLINE >= GETDATE());
```

100 %

Сообщения

Выполнение команд успешно завершено.

Время выполнения: 2023-10-20T02:03:39.2452554+03:00

Напишите 5 запросов, используя агрегатные функции (SUM, MAX, MIN, AVG, COUNT)

```
141 --Вычислить общее количество отделов:
142 SELECT COUNT(*) AS TotalDepartments
143 FROM Departments;
```

100 %

Результаты Сообщения

	TotalDepartments
1	12

```
145 --Найти максимальный и минимальный возраст зависимых лиц:
146 SELECT MIN(YEAR(GETDATE()) - YEAR(DE_BORN)) AS MinAge, MAX(YEAR(GETDATE()) - YEAR(DE_BORN)) AS MaxAge
147 FROM Dependents;
```

100 %

Результаты Сообщения

	MinAge	MaxAge
1	13	48

```
149 --Найти суммарный бюджет всех проектов:
150 SELECT SUM(P_BUDGET) AS TotalBudget
151 FROM Projects;
```

100 %

Результаты Сообщения

	TotalBudget
1	1135000.00

```

153 --Вычислить средний возраст всех сотрудников:
154 SELECT AVG(YEAR(GETDATE()) - YEAR(E_BORN)) AS AverageAge
155 FROM Employees;

```

100 %

Результаты Сообщения

	AverageAge
1	32

```

157 --Поиск максимальной зарплаты среди всех сотрудников:
158 SELECT MAX(E_SALARY) AS MaxSalary
159 FROM Employees;

```

100 %

Результаты Сообщения

	MaxSalary
1	75000.00

Написать 2 запроса на группировку данных, используя оператор GROUP BY

```

161 --Группировка сотрудников по отделам и подсчет количества сотрудников в каждом отделе:
162 SELECT D_ID, COUNT(*) AS EmployeesCount
163 FROM Employees
164 GROUP BY D_ID;

```

100 %

Результаты Сообщения

	D_ID	EmployeesCount
1	1	3
2	2	3
3	3	2
4	4	2
5	5	2
6	6	2
7	7	2
8	8	2
9	9	2
10	10	2

```

173 --Группировка проектов по отделам и вычисление среднего бюджета проектов в каждом отделе:
174 SELECT D_ID, AVG(P_BUDGET) AS AverageBudget
175 FROM Projects
176 GROUP BY D_ID;

```

100 %

Результаты Сообщения

	D_ID	AverageBudget
1	1	100000.000000
2	2	80000.000000
3	3	110000.000000
4	4	90000.000000
5	5	95000.000000
6	6	110000.000000
7	7	85000.000000
8	8	130000.000000
9	9	70000.000000
10	10	75000.000000
11	11	80000.000000

Написать 2 запроса на фильтрацию групп, используя HAVING.

```

179  --Фильтрация отделов по средней зарплате сотрудников:
180  SELECT D_ID, D_NAME
181  FROM Departments
182  WHERE D_ID IN (
183      SELECT D_ID
184      FROM Employees
185      GROUP BY D_ID
186      HAVING AVG(E_SALARY) > 60000

```

100 %

Результаты Сообщения

	D_ID	D_NAME
1	2	Marketing
2	3	IT
3	5	HR
4	6	Operations

```

189  --Фильтрация отделов по среднему бюджету проектов:
190  SELECT D_ID, D_NAME
191  FROM Departments
192  WHERE D_ID IN (
193      SELECT D_ID
194      FROM Projects
195      GROUP BY D_ID
196      HAVING AVG(P_BUDGET) > 90000
197  );

```

100 %

Результаты Сообщения

	D_ID	D_NAME
1	1	Sales
2	3	IT
3	5	HR
4	6	Operations
5	8	Customer Service

Написать запросы на расширения SQL Server для группировки (ROLLUP, CUBE, GROUPING SETS, OVER)

```

199 --Вычислить среднюю зарплату для каждой уникальной комбинации значений D_ID и E_ID в отделах с помощью ROLLUP.
200 SELECT
201     IIF (GROUPING(D_ID) = 1 , 'Среднее по всем отделам', CAST(D_ID AS NVARCHAR(255))) as D_ID,
202     IIF (GROUPING(E_ID) = 1 , 'Среднее по отделу', CAST(E_ID AS NVARCHAR(255))) as E_ID,
203     CAST(AVG(E_SALARY) AS DECIMAL(10,2)) AS AVGSalary
204 FROM Employees
205 GROUP BY ROLLUP (D_ID, E_ID);

```

100 %

Результаты Сообщения

	D_ID	E_ID	AVGSalary
1	1	1	50000.00
2	1	10	55000.00
3	1	21	65000.00
4	1	Среднее по отделу	56666.67
5	2	11	62000.00
6	2	20	63000.00
7	2	22	70000.00
8	2	Среднее по отделу	65000.00
9	3	6	65000.00
10	3	12	58000.00
11	3	Среднее по отделу	61500.00
12	4	2	60000.00
13	4	13	56000.00
14	4	Среднее по отделу	58000.00
15	5	7	70000.00
16	5	14	59000.00
17	5	Среднее по отделу	64500.00
18	6	8	75000.00
19	6	15	64000.00
20	6	Среднее по отделу	69500.00
21	7	4	48000.00
22	7	16	67000.00
23	7	Среднее по отделу	57500.00
24	8	3	55000.00
25	8	17	60000.00
26	8	Среднее по отделу	57500.00
27	9	9	48000.00
28	9	18	57000.00
29	9	Среднее по отделу	52500.00
30	10	5	52000.00
31	10	19	61000.00
32	10	Среднее по отделу	56500.00
33	Среднее по всем отделам	Среднее по отделу	60000.00

```

207 --Вычислить среднюю зарплату для каждой комбинации значений D_ID и E_ID в отделах с помощью CUBE.
208 SELECT
209     IF (GROUPING(D_ID) = 1, 'Среднее по E_ID', CAST(D_ID AS NVARCHAR(255))) as D_ID,
210     IF (GROUPING(E_ID) = 1, 'Среднее по отделу', CAST(E_ID AS NVARCHAR(255))) as E_ID,
211     CAST(AVG(E_SALARY) AS DECIMAL(10,2)) AS AVGSalary
212 FROM Employees

```

100 %

Результаты Сообщения

D_ID	E_ID	AVGSalary
15	6	75000.00
16	Среднее по E_ID	75000.00
17	9	48000.00
18	Среднее по E_ID	48000.00
19	1	55000.00
20	Среднее по E_ID	55000.00
21	2	62000.00
22	Среднее по E_ID	62000.00
23	3	58000.00
24	Среднее по E_ID	58000.00
25	4	56000.00
26	Среднее по E_ID	56000.00
27	5	59000.00
28	Среднее по E_ID	59000.00
29	6	64000.00
30	Среднее по E_ID	64000.00
31	7	67000.00
32	Среднее по E_ID	67000.00
33	8	60000.00
34	Среднее по E_ID	60000.00
35	9	57000.00
36	Среднее по E_ID	57000.00
37	10	61000.00
38	Среднее по E_ID	61000.00
39	2	63000.00
40	Среднее по E_ID	63000.00
41	1	65000.00
42	Среднее по E_ID	65000.00
43	2	70000.00
44	Среднее по E_ID	70000.00
45	Среднее по E_ID	Среднее по отделу
46	1	Среднее по отделу
47	2	Среднее по отделу
48	3	Среднее по отделу
49	4	Среднее по отделу
50	5	Среднее по отделу
51	6	Среднее по отделу
52	7	Среднее по отделу
53	8	Среднее по отделу
54	9	Среднее по отделу
55	10	Среднее по отделу

100 %

Результаты Сообщения

100 %

Результаты Сообщения

100 % ← → ⌂ Результаты ⌂ Сообщения

98 %										
<div> <div> <div></div> <div>Результаты</div> </div> <div> <div></div> <div>Сообщения</div> </div> </div>										
	E_ID	E_NAME	E_ADDRESS	E_SALARY	E_BORN	D_ID	D_ID	D_NAME	D_ADDRESS	D_HEAD
1	1	John Doe	123 Main St	50000.00	1990-05-15	1	1	Sales	123 Sales St	1
2	2	Jane Smith	456 Elm St	60000.00	1988-02-28	4	4	Finance	101 Finance St	4
3	3	Alice Johnson	789 Oak St	55000.00	1992-11-10	8	8	Customer Service	505 Customer Service St	8
4	4	Bob Brown	101 Pine St	48000.00	1995-07-20	7	7	Research	404 Research St	7
5	5	Eva White	202 Maple St	52000.00	1985-04-03	10	10	Quality Assurance	707 QA St	10
6	6	Michael Davis	303 Cedar St	65000.00	1993-09-18	3	3	IT	789 IT St	3
7	7	Olivia Lee	404 Birch St	70000.00	1987-12-07	5	5	HR	202 HR St	5
8	8	Daniel Wilson	505 Spruce St	75000.00	1984-08-25	6	6	Operations	303 Operations St	6
9	9	Sophia Martinez	606 Ash St	48000.00	1991-03-14	9	9	Product Development	606 Product Development St	NULL
10	10	Liam Thomas	707 Fir St	55000.00	1994-06-29	1	1	Sales	123 Sales St	1
11	11	Sarah Johnson	808 Pine St	62000.00	1993-08-12	2	2	Marketing	456 Marketing St	NULL
12	12	David Lee	909 Oak St	58000.00	1994-02-19	3	3	IT	789 IT St	3
13	13	Megan Brown	101 Elm St	56000.00	1991-06-25	4	4	Finance	101 Finance St	4
14	14	Kevin Miller	202 Cedar St	59000.00	1989-11-07	5	5	HR	202 HR St	5
15	15	Amanda Wilson	303 Spruce St	64000.00	1990-09-15	6	6	Operations	303 Operations St	6
16	16	Christopher Hall	404 Birch St	67000.00	1987-04-28	7	7	Research	404 Research St	7
17	17	Emma Martinez	505 Maple St	60000.00	1992-12-04	8	8	Customer Service	505 Customer Service St	8
18	18	Ryan Taylor	606 Ash St	57000.00	1988-07-21	9	9	Product Development	606 Product Development St	NULL
19	19	Grace Robinson	707 Fir St	61000.00	1995-03-09	10	10	Quality Assurance	707 QA St	10
20	20	Brandon Young	808 Pine St	65000.00	1986-10-13	2	2	Marketing	456 Marketing St	NULL
21	21	Olav Olsen	111 Bjørk St	63000.00	1993-08-12	1	1	Sales	123 Sales St	1
22	22	Ingrid Pedersen	222 Gran St	70000.00	1994-02-19	2	2	Marketing	456 Marketing St	NULL

```
5 |  
6 | --Внешнее левое соединение таблиц Departments и Projects  
7 | SELECT *  
8 | FROM Departments  
9 | LEFT JOIN Projects ON Departments.D_ID = Projects.D_ID;  
  
98 %  
Результаты Сообщения  


|    | D_ID | D_NAME              | D_ADDRESS                  | D_HEAD | P_ID | P_NAME    | P_BUDGET  | D_ID |
|----|------|---------------------|----------------------------|--------|------|-----------|-----------|------|
| 1  | 1    | Sales               | 123 Sales St               | 1      | 1    | Project A | 100000.00 | 1    |
| 2  | 2    | Marketing           | 456 Marketing St           | NULL   | 2    | Project B | 80000.00  | 2    |
| 3  | 3    | IT                  | 789 IT St                  | 3      | 3    | Project C | 120000.00 | 3    |
| 4  | 3    | IT                  | 789 IT St                  | 3      | 11   | Project K | 100000.00 | 3    |
| 5  | 4    | Finance             | 101 Finance St             | 4      | 4    | Project D | 90000.00  | 4    |
| 6  | 5    | HR                  | 202 HR St                  | 5      | 5    | Project E | 95000.00  | 5    |
| 7  | 6    | Operations          | 303 Operations St          | 6      | 6    | Project F | 110000.00 | 6    |
| 8  | 7    | Research            | 404 Research St            | 7      | 7    | Project G | 85000.00  | 7    |
| 9  | 8    | Customer Service    | 505 Customer Service St    | 8      | 8    | Project H | 130000.00 | 8    |
| 10 | 9    | Product Development | 606 Product Development St | NULL   | 9    | Project I | 70000.00  | 9    |
| 11 | 10   | Quality Assurance   | 707 QA St                  | 10     | 10   | Project J | 75000.00  | 10   |
| 12 | 11   | Legal               | 808 Legal St               | 13     | 12   | Project L | 80000.00  | 11   |
| 13 | 12   | Purchasing          | 909 Purchasing St          | 15     | NULL | NULL      | NULL      | NULL |


```

внешнее право;

```

12  --Внешнее правое соединение таблиц Employees и Departments
13  SELECT *
14  FROM Employees
15  RIGHT JOIN Departments ON Employees.D_ID = Departments.D_ID;

```

98 %

Результаты Сообщения

	E_ID	E_NAME	E_ADDRESS	E_SALARY	E_BORN	D_ID	D_ID	D_NAME	D_ADDRESS	D_HEAD
1	1	John Doe	123 Main St	50000.00	1990-05-15	1	1	Sales	123 Sales St	1
2	10	Liam Thomas	707 Fir St	55000.00	1994-06-29	1	1	Sales	123 Sales St	1
3	21	Olav Olsen	111 Bjork St	65000.00	1993-08-12	1	1	Sales	123 Sales St	1
4	11	Sarah Johnson	808 Pine St	62000.00	1993-08-12	2	2	Marketing	456 Marketing St	NULL
5	20	Brandon Young	808 Pine St	63000.00	1986-10-13	2	2	Marketing	456 Marketing St	NULL
6	22	Ingrid Pedersen	222 Gran St	70000.00	1994-02-19	2	2	Marketing	456 Marketing St	NULL
7	6	Michael Davis	303 Cedar St	65000.00	1993-09-18	3	3	IT	789 IT St	3
8	12	David Lee	909 Oak St	58000.00	1994-02-19	3	3	IT	789 IT St	3
9	2	Jane Smith	456 Elm St	60000.00	1988-02-28	4	4	Finance	101 Finance St	4
10	13	Megan Brown	101 Elm St	56000.00	1991-06-25	4	4	Finance	101 Finance St	4
11	7	Olivia Lee	404 Birch St	70000.00	1987-12-07	5	5	HR	202 HR St	5
12	14	Kevin Miller	202 Cedar St	59000.00	1989-11-07	5	5	HR	202 HR St	5
13	8	Daniel Wilson	505 Spruce St	75000.00	1984-08-25	6	6	Operations	303 Operations St	6
14	15	Amanda Wilson	303 Spruce St	64000.00	1990-09-15	6	6	Operations	303 Operations St	6
15	4	Bob Brown	101 Pine St	48000.00	1995-07-20	7	7	Research	404 Research St	7
16	16	Christopher Hall	404 Birch St	67000.00	1987-04-28	7	7	Research	404 Research St	7
17	3	Alice Johnson	789 Oak St	55000.00	1992-11-10	8	8	Customer Service	505 Customer Service St	8
18	17	Emma Martinez	505 Maple St	60000.00	1992-12-04	8	8	Customer Service	505 Customer Service St	8
19	9	Sophia Martinez	606 Ash St	48000.00	1991-03-14	9	9	Product Development	606 Product Development St	NULL
20	18	Ryan Taylor	606 Ash St	57000.00	1988-07-21	9	9	Product Development	606 Product Development St	NULL
21	5	Eva White	202 Maple St	52000.00	1985-04-03	10	10	Quality Assurance	707 QA St	10
22	19	Grace Robinson	707 Fir St	61000.00	1995-03-09	10	10	Quality Assurance	707 QA St	10
23	NULL	NULL	NULL	NULL	NULL	NULL	11	Legal	808 Legal St	13
24	NULL	NULL	NULL	NULL	NULL	NULL	12	Purchasing	909 Purchasing St	15

ПОЛНОЕ ВНЕШНЕЕ СОЕДИНЕНИЕ;

```

12  --Полное внешнее соединение таблиц Employees и Dependents
13  SELECT *
14  FROM Employees
15  FULL JOIN Dependents ON Employees.E_ID = Dependents.E_ID;

```

98 %

Результаты Сообщения

	E_ID	E_NAME	E_ADDRESS	E_SALARY	E_BORN	D_ID	DE_ID	DE_NAME	E_ID	DE_BORN	DE_STATUS
1	1	John Doe	123 Main St	50000.00	1990-05-15	1	1	Child 1	1	2000-03-05	Son
2	2	Jane Smith	456 Elm St	60000.00	1988-02-28	4	3	Spouse 1	2	1975-11-18	Spouse
3	3	Alice Johnson	789 Oak St	55000.00	1992-11-10	8	2	Child 2	3	2002-07-12	Daughter
4	4	Bob Brown	101 Pine St	48000.00	1995-07-20	7	8	Child 6	4	2003-12-19	Daughter
5	5	Eva White	202 Maple St	52000.00	1985-04-03	10	7	Child 5	5	1999-04-08	Son
6	6	Michael Davis	303 Cedar St	65000.00	1993-09-18	3	6	Spouse 2	6	1980-06-22	Spouse
7	7	Olivia Lee	404 Birch St	70000.00	1987-12-07	5	4	Child 3	7	2005-01-30	Son
8	8	Daniel Wilson	505 Spruce St	75000.00	1984-08-25	6	10	Spouse 3	8	1978-09-07	Spouse
9	9	Sophia Martinez	606 Ash St	48000.00	1991-03-14	9	5	Child 4	9	2008-09-14	Daughter
10	10	Liam Thomas	707 Fir St	55000.00	1994-06-29	1	9	Child 7	10	2010-02-28	Son
11	11	Sarah Johnson	808 Pine St	62000.00	1993-08-12	2	NULL	NULL	NULL	NULL	NULL
12	12	David Lee	909 Oak St	58000.00	1994-02-19	3	NULL	NULL	NULL	NULL	NULL
13	13	Megan Brown	101 Elm St	56000.00	1991-06-25	4	NULL	NULL	NULL	NULL	NULL
14	14	Kevin Miller	202 Cedar St	59000.00	1989-11-07	5	15	Spouse	14	2000-01-01	Spouse
15	15	Amanda Wilson	303 Spruce St	64000.00	1990-09-15	6	NULL	NULL	NULL	NULL	NULL
16	16	Christopher Hall	404 Birch St	67000.00	1987-04-28	7	NULL	NULL	NULL	NULL	NULL
17	17	Emma Martinez	505 Maple St	60000.00	1992-12-04	8	NULL	NULL	NULL	NULL	NULL
18	18	Ryan Taylor	606 Ash St	57000.00	1988-07-21	9	NULL	NULL	NULL	NULL	NULL
19	19	Grace Robinson	707 Fir St	61000.00	1995-03-09	10	NULL	NULL	NULL	NULL	NULL
20	20	Brandon Young	808 Pine St	63000.00	1986-10-13	2	16	Child 8	20	2001-02-01	Son
21	21	Olav Olsen	111 Bjork St	65000.00	1993-08-12	1	NULL	NULL	NULL	NULL	NULL
22	22	Ingrid Pedersen	222 Gran St	70000.00	1994-02-19	2	NULL	NULL	NULL	NULL	NULL

перекрестное соединение.

```

17 --Перекрестное соединение таблиц Projects и Participations
18 SELECT *
19 FROM Projects
20 CROSS JOIN Participations;

```

98 %

Результаты Сообщения

	P_ID	P_NAME	P_BUDGET	D_ID	P_ID	E_ID	E_START	E_HEAD	E_DEADLINE
10	10	Project J	75000.00	10	1	1	2023-01-15	1	NULL
11	11	Project K	100000.00	3	1	1	2023-01-15	1	NULL
12	12	Project L	80000.00	11	1	1	2023-01-15	1	NULL
13	1	Project A	100000.00	1	1	3	2023-01-20	1	NULL
14	2	Project B	80000.00	2	1	3	2023-01-20	1	NULL
15	3	Project C	120000.00	3	1	3	2023-01-20	1	NULL
16	4	Project D	90000.00	4	1	3	2023-01-20	1	NULL
17	5	Project E	95000.00	5	1	3	2023-01-20	1	NULL
18	6	Project F	110000.00	6	1	3	2023-01-20	1	NULL
19	7	Project G	85000.00	7	1	3	2023-01-20	1	NULL
20	8	Project H	130000.00	8	1	3	2023-01-20	1	NULL
21	9	Project I	70000.00	9	1	3	2023-01-20	1	NULL
22	10	Project J	75000.00	10	1	3	2023-01-20	1	NULL
23	11	Project K	100000.00	3	1	3	2023-01-20	1	NULL
24	12	Project L	80000.00	11	1	3	2023-01-20	1	NULL
25	1	Project A	100000.00	1	2	2	2023-02-01	NULL	NULL
26	2	Project B	80000.00	2	2	2	2023-02-01	NULL	NULL
27	3	Project C	120000.00	3	2	2	2023-02-01	NULL	NULL
28	4	Project D	90000.00	4	2	2	2023-02-01	NULL	NULL
29	5	Project E	95000.00	5	2	2	2023-02-01	NULL	NULL
30	6	Project F	110000.00	6	2	2	2023-02-01	NULL	NULL
31	7	Project G	85000.00	7	2	2	2023-02-01	NULL	NULL
32	8	Project H	130000.00	8	2	2	2023-02-01	NULL	NULL
33	9	Project I	70000.00	9	2	2	2023-02-01	NULL	NULL
34	10	Project J	75000.00	10	2	2	2023-02-01	NULL	NULL
35	11	Project K	100000.00	3	2	2	2023-02-01	NULL	NULL
36	12	Project L	80000.00	11	2	2	2023-02-01	NULL	NULL
37	1	Project A	100000.00	1	2	4	2023-02-05	NULL	NULL
38	2	Project B	80000.00	2	2	4	2023-02-05	NULL	NULL
39	3	Project C	120000.00	3	2	4	2023-02-05	NULL	NULL
40	4	Project D	90000.00	4	2	4	2023-02-05	NULL	NULL
41	5	Project E	95000.00	5	2	4	2023-02-05	NULL	NULL
42	6	Project F	110000.00	6	2	4	2023-02-05	NULL	NULL
43	7	Project G	85000.00	7	2	4	2023-02-05	NULL	NULL
44	8	Project H	130000.00	8	2	4	2023-02-05	NULL	NULL
45	9	Project I	70000.00	9	2	4	2023-02-05	NULL	NULL
46	10	Project J	75000.00	10	2	4	2023-02-05	NULL	NULL
47	11	Project K	100000.00	3	2	4	2023-02-05	NULL	NULL
48	12	Project L	80000.00	11	2	4	2023-02-05	NULL	NULL
49	1	Project A	100000.00	1	3	5	2023-03-10	3	NULL
50	2	Project B	80000.00	2	3	5	2023-03-10	3	NULL
51	3	Project C	120000.00	3	3	5	2023-03-10	3	NULL
52	4	Project D	90000.00	4	3	5	2023-03-10	3	NULL
53	5	Project E	95000.00	5	3	5	2023-03-10	3	NULL

14. Написать по два запроса:

на пересечение;

```

22  --Запрос на пересечение
23  SELECT *
24  FROM Employees
25  WHERE E_SALARY > 50000
26  INTERSECT
27  SELECT *
28  FROM Employees
29  WHERE E_NAME LIKE 'B%';
30
31  --Запрос на разность
32  SELECT *

```

98 %

Результаты Сообщения

	E_ID	E_NAME	E_ADDRESS	E_SALARY	E_BORN	D_ID
1	20	Brandon Young	808 Pine St	63000.00	1986-10-13	2

на разность;

```

31  --Запрос на разность
32  SELECT *
33  FROM Employees
34  EXCEPT
35  SELECT *
36  FROM Employees
37  WHERE E_NAME LIKE 'A%';
38

```

98 %

Результаты Сообщения

	E_ID	E_NAME	E_ADDRESS	E_SALARY	E_BORN	D_ID
1	1	John Doe	123 Main St	50000.00	1990-05-15	1
2	2	Jane Smith	456 Elm St	60000.00	1988-02-28	4
3	4	Bob Brown	101 Pine St	48000.00	1995-07-20	7
4	5	Eva White	202 Maple St	52000.00	1985-04-03	10
5	6	Michael Davis	303 Cedar St	65000.00	1993-09-18	3
6	7	Olivia Lee	404 Birch St	70000.00	1987-12-07	5
7	8	Daniel Wilson	505 Spruce St	75000.00	1984-08-25	6
8	9	Sophia Martinez	606 Ash St	48000.00	1991-03-14	9
9	10	Liam Thomas	707 Fir St	55000.00	1994-06-29	1
10	11	Sarah Johnson	808 Pine St	62000.00	1993-08-12	2
11	12	David Lee	909 Oak St	58000.00	1994-02-19	3
12	13	Megan Brown	101 Elm St	56000.00	1991-06-25	4
13	14	Kevin Miller	202 Cedar St	59000.00	1989-11-07	5
14	16	Christopher Hall	404 Birch St	67000.00	1987-04-28	7
15	17	Emma Martinez	505 Maple St	60000.00	1992-12-04	8
16	18	Ryan Taylor	606 Ash St	57000.00	1988-07-21	9
17	19	Grace Robinson	707 Fir St	61000.00	1995-03-09	10
18	20	Brandon Young	808 Pine St	63000.00	1986-10-13	2
19	21	Olav Olsen	111 Bjork St	65000.00	1993-08-12	1
20	22	Ingrid Pedersen	222 Gran St	70000.00	1994-02-19	2

на объединение таблиц.

```

39  --Запрос на объединение таблиц
40  SELECT *
41  FROM Employees
42  WHERE E_SALARY > 60000
43  UNION
44  SELECT *
45  FROM Employees
46  WHERE YEAR(E_BORN) > 1990;
47

```

98 %

Результаты Сообщения

	E_ID	E_NAME	E_ADDRESS	E_SALARY	E_BORN	D_ID
1	3	Alice Johnson	789 Oak St	55000.00	1992-11-10	8
2	4	Bob Brown	101 Pine St	48000.00	1995-07-20	7
3	6	Michael Davis	303 Cedar St	65000.00	1993-09-18	3
4	7	Olivia Lee	404 Birch St	70000.00	1987-12-07	5
5	8	Daniel Wilson	505 Spruce St	75000.00	1984-08-25	6
6	9	Sophia Martinez	606 Ash St	48000.00	1991-03-14	9
7	10	Liam Thomas	707 Fir St	55000.00	1994-06-29	1
8	11	Sarah Johnson	808 Pine St	62000.00	1993-08-12	2
9	12	David Lee	909 Oak St	58000.00	1994-02-19	3
10	13	Megan Brown	101 Elm St	56000.00	1991-06-25	4
11	15	Amanda Wilson	303 Spruce St	64000.00	1990-09-15	6
12	16	Christopher Hall	404 Birch St	67000.00	1987-04-28	7
13	17	Emma Martinez	505 Maple St	60000.00	1992-12-04	8
14	19	Grace Robinson	707 Fir St	61000.00	1995-03-09	10
15	20	Brandon Young	808 Pine St	63000.00	1986-10-13	2
16	21	Olav Olsen	111 Bjork St	65000.00	1993-08-12	1
17	22	Ingrid Pedersen	222 Gran St	70000.00	1994-02-19	2

15. Написать 4 запроса с использованием подзапросов, используя операторы сравнения, операторы IN, ANY|SOME и ALL, предикат EXISTS

```

48  --Выбрать сотрудников, зарплата которых больше средней зарплаты
49  SELECT *
50  FROM Employees
51  WHERE E_SALARY > (SELECT AVG(E_SALARY) FROM Employees);
52

```

98 %

Результаты Сообщения

	E_ID	E_NAME	E_ADDRESS	E_SALARY	E_BORN	D_ID
1	6	Michael Davis	303 Cedar St	65000.00	1993-09-18	3
2	7	Olivia Lee	404 Birch St	70000.00	1987-12-07	5
3	8	Daniel Wilson	505 Spruce St	75000.00	1984-08-25	6
4	11	Sarah Johnson	808 Pine St	62000.00	1993-08-12	2
5	15	Amanda Wilson	303 Spruce St	64000.00	1990-09-15	6
6	16	Christopher Hall	404 Birch St	67000.00	1987-04-28	7
7	19	Grace Robinson	707 Fir St	61000.00	1995-03-09	10
8	20	Brandon Young	808 Pine St	63000.00	1986-10-13	2
9	21	Olav Olsen	111 Bjork St	65000.00	1993-08-12	1
10	22	Ingrid Pedersen	222 Gran St	70000.00	1994-02-19	2


```

53 | --Выбрать сотрудников, которые являются руководителями отделов
54 | SELECT *
55 | FROM Employees
56 | WHERE E_ID IN (SELECT D_HEAD FROM Departments);

```

98 %

Результаты Сообщения

	E_ID	E_NAME	E_ADDRESS	E_SALARY	E_BORN	D_ID
1	1	John Doe	123 Main St	50000.00	1990-05-15	1
2	3	Alice Johnson	789 Oak St	55000.00	1992-11-10	8
3	4	Bob Brown	101 Pine St	48000.00	1995-07-20	7
4	5	Eva White	202 Maple St	52000.00	1985-04-03	10
5	6	Michael Davis	303 Cedar St	65000.00	1993-09-18	3
6	7	Olivia Lee	404 Birch St	70000.00	1987-12-07	5
7	8	Daniel Wilson	505 Spruce St	75000.00	1984-08-25	6
8	10	Liam Thomas	707 Fir St	55000.00	1994-06-29	1
9	13	Megan Brown	101 Elm St	56000.00	1991-06-25	4
10	15	Amanda Wilson	303 Spruce St	64000.00	1990-09-15	6

```

58 |
59 | -- Выбрать сотрудников, участвующих в проектах
60 | SELECT *
61 | FROM Employees
62 | WHERE EXISTS (SELECT 1 FROM Participations WHERE E_ID = Employees.E_ID);
63 |
64 |

```

98 %

Результаты Сообщения

	E_ID	E_NAME	E_ADDRESS	E_SALARY	E_BORN	D_ID
1	1	John Doe	123 Main St	50000.00	1990-05-15	1
2	2	Jane Smith	456 Elm St	60000.00	1988-02-28	4
3	3	Alice Johnson	789 Oak St	55000.00	1992-11-10	8
4	4	Bob Brown	101 Pine St	48000.00	1995-07-20	7
5	5	Eva White	202 Maple St	52000.00	1985-04-03	10
6	6	Michael Davis	303 Cedar St	65000.00	1993-09-18	3
7	7	Olivia Lee	404 Birch St	70000.00	1987-12-07	5
8	8	Daniel Wilson	505 Spruce St	75000.00	1984-08-25	6
9	9	Sophia Martinez	606 Ash St	48000.00	1991-03-14	9
10	10	Liam Thomas	707 Fir St	55000.00	1994-06-29	1
11	15	Amanda Wilson	303 Spruce St	64000.00	1990-09-15	6
12	21	Olav Olsen	111 Bjork St	65000.00	1993-08-12	1

```

66 | --Запрос на выбор сотрудников без иждивенцев, у которых зарплата больше любой зарплаты работника, имеющим иждивенцев.
67 | SELECT E_ID, E_NAME
68 | FROM Employees
69 | WHERE E_ID NOT IN (SELECT E_ID FROM Dependents) AND E_SALARY > ANY (SELECT E_SALARY FROM Employees
70 | WHERE E_ID IN (SELECT E_ID FROM Dependents));

```

98 %

Результаты Сообщения

	E_ID	E_NAME
1	11	Sarah Johnson
2	12	David Lee
3	13	Megan Brown
4	15	Amanda Wilson
5	16	Christopher Hall
6	17	Emma Martinez
7	18	Ryan Taylor
8	19	Grace Robinson
9	21	Olav Olsen
10	22	Ingrid Pedersen

Дополнение

Задание 1:

Написать 4 запроса по строковым функциям


```
1  --Выборка имен сотрудников в верхнем регистре
2  SELECT E_NAME, UPPER(E_NAME) AS UPPERCASE_NAME
3  FROM Employees;
4
5  --Конкатенация названия отделения и адреса в таблице Departments
6  SELECT D_NAME + ', ' + D_ADDRESS AS NAME_ADDRESS
7  FROM Departments;
8
9  --Использование функции RIGHT для сокращения названия проектов
10 SELECT P_NAME, RIGHT(P_NAME, 1) AS SHORT_NAME
11 FROM Projects;
12
13 --Использование функции REPLACE для замены статуса для иждивенцев
14 SELECT DE_STATUS, REPLACE(DE_STATUS, 'Son', 'Child') AS UPDATED_Son
15 FROM Dependents;
16
```

98 %

Результаты Сообщения

	E_NAME	UPPERCASE_NAME
1	John Doe	JOHN DOE
2	Jane Smith	JANE SMITH
3	Alice Johnson	ALICE JOHNSON
4	Bob Brown	BOB BROWN
5	Eva White	EVA WHITE
6	Michael Davis	MICHAEL DAVIS
7	Olivia Lee	OLIVIA LEE
8	Daniel Wilson	DANIEL WILSON
9	Sophia Martinez	SOPHIA MARTINEZ
10	Liam Thomas	LIAM THOMAS
11	Sarah Johnson	SARAH JOHNSON
12	David Lee	DAVID LEE
13	Megan Brown	MEGAN BROWN
14	Kevin Miller	KEVIN MILLER
15	Amanda Wilson	AMANDA WILSON
16	Christopher Hall	CHRISTOPHER HALL
17	Emma Martinez	EMMA MARTINEZ
18	Ryan Taylor	RYAN TAYLOR
19	Grace Robinson	GRACE ROBINSON

```

4
5 --Конкатенация названия отделения и адреса в таблице Departments
6 SELECT D_NAME + ', ' + D_ADDRESS AS NAME_ADDRESS
7 FROM Departments;
8
9 --Использование функции RIGHT для сокращения названия проектов
10 SELECT P_NAME, RIGHT(P_NAME, 1) AS SHORT_NAME
11 FROM Projects;
12
13 --Использование функции REPLACE для замены статуса для иждивенцев
14 SELECT DE_STATUS, REPLACE(DE_STATUS, 'Son', 'Child') AS UPDATED_Son
15 FROM Dependents;
16

```

98 %

Результаты Сообщения

	NAME_ADDRESS
1	Sales, 123 Sales St
2	Marketing, 456 Marketing St
3	IT, 789 IT St
4	Finance, 101 Finance St
5	HR, 202 HR St
6	Operations, 303 Operations St
7	Research, 404 Research St
8	Customer Service, 505 Customer Service St
9	Product Development, 606 Product Development St
10	Quality Assurance, 707 QA St
11	Legal, 808 Legal St
12	Purchasing, 909 Purchasing St

```

8
9 --Использование функции RIGHT для сокращения названия проектов
10 SELECT P_NAME, RIGHT(P_NAME, 1) AS SHORT_NAME
11 FROM Projects;
12
13 --Использование функции REPLACE для замены статуса для иждивенцев
14 SELECT DE_STATUS, REPLACE(DE_STATUS, 'Son', 'Child') AS UPDATED_Son
15 FROM Dependents;
16

```

98 %

Результаты Сообщения

	P_NAME	SHORT_NAME
1	Project A	A
2	Project B	B
3	Project C	C
4	Project D	D
5	Project E	E
6	Project F	F
7	Project G	G
8	Project H	H
9	Project I	I
10	Project J	J
11	Project K	K
12	Project L	L

```

11 FROM Projects;
12
13 --Использование функции REPLACE для замены статуса для иждивенцев
14 SELECT DE_STATUS, REPLACE(DE_STATUS, 'Son', 'Child') AS UPDATED_Son
15 FROM Dependents;
16

```

98 %

Результаты Сообщения

	DE_STATUS	UPDATED_Son
1	Son	Child
2	Daughter	Daughter
3	Spouse	Spouse
4	Son	Child
5	Daughter	Daughter
6	Spouse	Spouse
7	Son	Child
8	Daughter	Daughter
9	Son	Child
10	Spouse	Spouse
11	Spouse	Spouse
12	Son	Child

Задание 2:

Написать 4 запроса по числовым функциям

```

16
17 --Использование функции FLOOR для округления вверх до целого числа среднего бюджета по отделам
18 SELECT D_ID, FLOOR(AVG(P_BUDGET)) AS ROUNDED_AVGSAL
19 FROM Projects
20 GROUP BY D_ID;
21

```

98 %

Результаты Сообщения

	D_ID	ROUNDED_AVGSAL
1	1	100000
2	2	80000
3	3	110000
4	4	90000
5	5	95000
6	6	110000
7	7	85000
8	8	130000
9	9	70000
10	10	75000
11	11	80000

```

21
22 --Использование функции ROUND для средней зарплаты до двух знаков
23 SELECT D_ID, ROUND(AVG(E_SALARY),2) AS ROUNDED_AVGSAL
24 FROM Employees
25 GROUP BY D_ID;
26
27 --Использование функции SQRT для расчета корня из бюджета проекта

```

98 %

Результаты Сообщения

	D_ID	ROUNDED_AVGSAL
1	1	56666.670000
2	2	65000.000000
3	3	61500.000000
4	4	58000.000000
5	5	64500.000000
6	6	69500.000000
7	7	57500.000000
8	8	57500.000000
9	9	52500.000000
10	10	56500.000000

```

26
27 --Использование функции CEILING для кругления вниз до целого числа средней зарплаты сотрудников
28 SELECT D_ID, CEILING(AVG(E_SALARY)) AS ROUNDED_AVGSAL
29 FROM Employees
30 GROUP BY D_ID;

```

98 %

Результаты Сообщения

	D_ID	ROUNDED_AVGSAL
1	1	56667
2	2	65000
3	3	61500
4	4	58000
5	5	64500
6	6	69500
7	7	57500
8	8	57500
9	9	52500
10	10	56500

Задание 3:

Создайте 5 представлений по своей БД

```

31
32 --Представление всех участников проектов и их руководителей
33 go
34 CREATE VIEW View_ParticipationAndHeads
35 AS
36 SELECT Pr.P_NAME, E1.E_NAME AS Participant, E2.E_NAME AS Head
37 FROM Participations P
38 JOIN Employees E1 ON P.E_ID = E1.E_ID
39 JOIN Employees E2 ON P.E_HEAD = E2.E_ID
40 JOIN Projects Pr ON P.P_ID = Pr.P_ID;
41 go
42 SELECT * FROM View_ParticipationAndHeads

```

```

44 --Представление проектов с бюджетом выше среднего
45 go
46 CREATE VIEW View_ProjectsAvgBudget
47 AS
48 SELECT *
49 FROM Projects
50 WHERE P_BUDGET > (SELECT AVG(P_BUDGET) FROM Projects);
51 go
52 SELECT * FROM View_ProjectsAvgBudget;

```

98 %

Результаты Сообщения

	P_ID	P_NAME	P_BUDGET	D_ID
1	1	Project A	100000.00	1
2	3	Project C	120000.00	3
3	5	Project E	95000.00	5
4	6	Project F	110000.00	6
5	8	Project H	130000.00	8
6	11	Project K	100000.00	3

```

54 --Представление проектов, которые работают в отделе с названием 'IT' или 'Sales':
55 go
56 CREATE VIEW View_EmployeesITSales
57 AS
58 SELECT *
59 FROM Employees
60 WHERE D_ID IN (SELECT D_ID FROM Departments WHERE D_NAME IN ('IT', 'Sales'));
61 go
62 SELECT * FROM View_EmployeesITSales;
63
64 --Представление, у которых зарплата больше 60000 и работают в отделе 'IT':

```

98 %

Результаты Сообщения

	E_ID	E_NAME	E_ADDRESS	E_SALARY	E_BORN	D_ID
1	1	John Doe	123 Main St	50000.00	1990-05-15	1
2	6	Michael Davis	303 Cedar St	65000.00	1993-09-18	3
3	10	Liam Thomas	707 Fir St	55000.00	1994-06-29	1
4	12	David Lee	909 Oak St	58000.00	1994-02-19	3
5	21	Olav Olsen	111 Bjork St	65000.00	1993-08-12	1

```

64 --Представление, у которых зарплата больше 60000 и работают в отделе 'IT':
65 go
66 CREATE VIEW View_EmployeesITFixSal
67 AS
68 SELECT *
69 FROM Employees
70 WHERE E_SALARY > 60000 AND D_ID = (SELECT D_ID FROM Departments WHERE D_NAME = 'IT');
71 go
72 SELECT * FROM View_EmployeesITFixSal;

```

98 %

Результаты Сообщения

	E_ID	E_NAME	E_ADDRESS	E_SALARY	E_BORN	D_ID
1	6	Michael Davis	303 Cedar St	65000.00	1993-09-18	3

```

74 --Представление проектов, у которых бюджет больше 100000 и на которые сотрудники зарегистрированы после 2019 года
75 go
76 CREATE VIEW View_ProjectsFixBUDData
77 AS
78 SELECT *
79 FROM Projects
80 WHERE P_BUDGET > 100000
81 AND P_ID IN (SELECT P_ID FROM Participations WHERE YEAR(E_START) > 2019);
82 go
83 SELECT * FROM View_ProjectsFixBUDData;

```

98 %

Результаты Сообщения

	P_ID	P_NAME	P_BUDGET	D_ID
1	3	Project C	120000.00	3
2	8	Project H	130000.00	8

Задание 4:

Покажите применение табличных переменных, временные локальных и глобальных таблиц, а также обобщенных табличных выражений.

```

85 --Использование локальной временной переменной:
86 SELECT D_NAME AS D_NAME, D_ADDRESS AS D_ADDRESS INTO #TempTable
87 FROM Departments;
88
89 -- Выполняем операции с временной таблицей
90 SELECT * FROM #TempTable;
91
92 -- Удаляем временную таблицу
93 DROP TABLE #TempTable;
94
95 --Использование табличной переменной:
96 Объявление табличной переменной

```

98 %

Результаты Сообщения

	D_NAME	D_ADDRESS
1	Sales	123 Sales St
2	Marketing	456 Marketing St
3	IT	789 IT St
4	Finance	101 Finance St
5	HR	202 HR St
6	Operations	303 Operations St
7	Research	404 Research St
8	Customer Service	505 Customer Service St
9	Product Development	606 Product Development St

```

94  --Использование табличной переменной:
95  -- Объявляем табличную переменную
96  DECLARE @TableVariable TABLE (
97      ID INT PRIMARY KEY,
98      Name NVARCHAR(50)
99  );
100
101  -- Вставляем данные в табличную переменную
102  INSERT INTO @TableVariable (ID, Name)
103  SELECT E_ID, E_NAME
104  FROM Employees;
105
106  SELECT * FROM @TableVariable;
107
108  --Использование временной глобальной таблицы

```

}%

Результаты Сообщения

	ID	Name
1	1	John Doe
2	2	Jane Smith
3	3	Alice Johnson
4	4	Bob Brown
5	5	Eva White
6	6	Michael Davis
7	7	Olivia Lee
8	8	Daniel Wilson
9	9	Sophia Martinez

```

107
108 --Использование временной глобальной таблицы:
109 -- Создаем временную глобальную таблицу
110 CREATE TABLE ##GlobalTempTable (
111     ID INT PRIMARY KEY,
112     Name NVARCHAR(50)
113 );
114
115 -- Вставляем данные во временную глобальную таблицу
116 INSERT INTO ##GlobalTempTable (ID, Name)
117 SELECT E_ID, E_NAME
118 FROM Employees;
119
120 -- Выполняем операции с временной глобальной таблицей
121 SELECT * FROM ##GlobalTempTable;
122
123 -- Удаляем временную глобальную таблицу
124 DROP TABLE ##GlobalTempTable;
125
126 --Использование обобщенного табличного выражения:

```

98 %

Результаты Сообщения

	ID	Name
1	1	John Doe
2	2	Jane Smith
3	3	Alice Johnson
4	4	Bob Brown
5	5	Eva White
6	6	Michael Davis
7	7	Olivia Lee
8	8	Daniel Wilson
9	9	Sophia Martinez

```

125
126 --Использование обобщенного табличного выражения:
127 WITH CTE AS (
128     SELECT E_ID, E_NAME, E_SALARY
129     FROM Employees
130     WHERE E_SALARY > 50000
131 )
132 SELECT *
133 FROM CTE;
134
135 --Увеличиваем процедуру без параметров

```

98 %

Результаты Сообщения

	E_ID	E_NAME	E_SALARY
1	2	Jane Smith	60000.00
2	3	Alice Johnson	55000.00
3	5	Eva White	52000.00
4	6	Michael Davis	65000.00
5	7	Olivia Lee	70000.00
6	8	Daniel Wilson	75000.00
7	10	Liam Thomas	55000.00
8	11	Sarah Johnson	62000.00
9	12	David Lee	58000.00

Задание 5:

Создать 2 хранимые процедуры, одна из которых будет иметь выходные параметры. Покажите применение ваши созданных хранимых процедур для вашей БД

```
85  --Хранимая процедура без параметров
86  go
87  CREATE PROCEDURE dbo.ProcEmp
88  AS
89  BEGIN
90      SELECT * FROM Employees;
91  END;
92  EXEC ProcEmp;
```

98 %

Результаты Сообщения

	E_ID	E_NAME	E_ADDRESS	E_SALARY	E_BORN	D_ID
1	1	John Doe	123 Main St	50000.00	1990-05-15	1
2	2	Jane Smith	456 Elm St	60000.00	1988-02-28	4
3	3	Alice Johnson	789 Oak St	55000.00	1992-11-10	8
4	4	Bob Brown	101 Pine St	48000.00	1995-07-20	7
5	5	Eva White	202 Maple St	52000.00	1985-04-03	10
6	6	Michael Davis	303 Cedar St	65000.00	1993-09-18	3
7	7	Olivia Lee	404 Birch St	70000.00	1987-12-07	5
8	8	Daniel Wilson	505 Spruce St	75000.00	1984-08-25	6
9	9	Sophia Martinez	606 Ash St	48000.00	1991-03-14	9
10	10	Liam Thomas	707 Fir St	55000.00	1994-06-29	1
11	11	Sarah Johnson	808 Pine St	62000.00	1993-08-12	2
12	12	David Lee	909 Oak St	58000.00	1994-02-19	3
13	13	Megan Brown	101 Elm St	56000.00	1991-06-25	4
14	14	Kevin Miller	202 Cedar St	59000.00	1989-11-07	5
15	15	Amanda Wilson	303 Spruce St	64000.00	1990-09-15	6
16	16	Christopher Hall	404 Birch St	67000.00	1987-04-28	7
17	17	Emma Martinez	505 Maple St	60000.00	1992-12-04	8
18	18	Ryan Taylor	606 Ash St	57000.00	1988-07-21	9
19	19	Grace Robinson	707 Fir St	61000.00	1995-03-09	10

```
144 | --Хранимая процедура с выходными параметрами
145 | go
146 | CREATE PROCEDURE dbo.ProcCOUNTEmp
147 |     @DepartmentID INT,
148 |     @EmployeeCount INT OUTPUT
149 | AS
150 | BEGIN
151 |     SELECT @EmployeeCount = COUNT(*)
152 |     FROM Employees
153 |     WHERE D_ID = @DepartmentID;
154 | END;
155 | DECLARE @Count INT;
156 | EXEC dbo.ProcCOUNTEmp
157 |     @DepartmentID = 1,
158 |     @EmployeeCount = @Count OUTPUT;
159 | PRINT 'Количество сотрудников в отделе: ' + CAST(@Count AS NVARCHAR(10));
```

98 %

Сообщения

Количество сотрудников в отделе: 3

Время выполнения: 2023-12-15T05:04:43.1174717+03:00

Задание 6:

Создать 2 определяемые пользователем функции, одна из которых скалярная функция, другая возвращая табличное значение. Покажите применение ваших созданных функции для вашей БД

```

160
161 --Скалярная функция для вычисления возраста сотрудника
162 go
163 CREATE FUNCTION dbo.CalculateAge
164 (
165     @BirthDate DATE
166 )
167 RETURNS INT
168 AS
169 BEGIN
170     DECLARE @Age INT;
171     SET @Age = DATEDIFF(YEAR, @BirthDate, GETDATE());
172     RETURN @Age;
173 END;
174 go
175 SELECT E_ID, E_NAME, dbo.CalculateAge(E_BORN) AS AGE
176 FROM Employees;

```

98 %

Результаты Сообщения

	E_ID	E_NAME	AGE
1	1	John Doe	33
2	2	Jane Smith	35
3	3	Alice Johnson	31
4	4	Bob Brown	28
5	5	Eva White	38
6	6	Michael Davis	30
7	7	Olivia Lee	36
8	8	Daniel Wilson	39
9	9	Sophia Martinez	32
10	10	Liam Thomas	29
11	11	Sarah Johnson	30
12	12	David Lee	29
13	13	Megan Brown	32
14	14	Kevin Miller	34
15	15	Amanda Wilson	33
16	16	Christopher Hall	36
17	17	Emma Martinez	31

178	--Функция, возвращающая список сотрудников в указанном отделе
179	go
180	CREATE FUNCTION dbo.GetEmployeesDepartment
181	(
182	@DepartmentID INT
183)
184	RETURNS TABLE
185	AS
186	RETURN
187	(
188	SELECT E_ID, E_NAME, E_SALARY, E_BORN
189	FROM Employees
190	WHERE D_ID = @DepartmentID
191);
192	go
193	SELECT *
194	FROM dbo.GetEmployeesDepartment(1);

98 %				
Результаты Сообщения				
	E_ID	E_NAME	E_SALARY	E_BORN
1	1	John Doe	50000.00	1990-05-15
2	10	Liam Thomas	55000.00	1994-06-29
3	21	Olav Olsen	65000.00	1993-08-12

Задание 7:

Создайте два триггера: триггер AFTER и триггер INSTEAD OF. Покажите применение ваших триггеров для вашей БД

196	--AFTER-триггер для обновления даты последнего изменения сотрудника
197	CREATE TABLE EmployeeLog (
198	LogID INT PRIMARY KEY IDENTITY(1,1),
199	EmployeeID INT,
200	UpdateDate DATETIME
201);
202	go
203	CREATE TRIGGER AfterEmployeeUpdate
204	ON Employees
205	AFTER UPDATE
206	AS
207	BEGIN
208	INSERT INTO EmployeeLog (EmployeeID, UpdateDate)
209	SELECT E_ID, GETDATE()
210	FROM INSERTED;
211	END;
212	go
213	UPDATE Employees
214	SET E_SALARY = E_SALARY + 1000
215	WHERE E_ID = 1;
216	SELECT * FROM EmployeeLog;
217	

98 %			
Результаты Сообщения			
	LogID	EmployeeID	UpdateDate
1	1	1	2023-12-15 05:16:40.680

```

218 --INSTEAD OF-триггер для проверки ограничения на заработную плату при вставке нового сотрудника
219 go
220 CREATE TRIGGER InsteadOfInsertEmployee
221 ON Employees
222 INSTEAD OF INSERT
223 AS
224 BEGIN
225     DECLARE @MaxSalary DECIMAL(8, 2) = 100000.00;
226
227     INSERT INTO Employees (E_ID, E_NAME, E_SALARY, E_BORN, D_ID)
228     SELECT E_ID, E_NAME, E_SALARY, E_BORN, D_ID
229     FROM INSERTED
230     WHERE E_SALARY <= @MaxSalary;
231 END;
232 go
233 -- Вставляем нового сотрудника с заработной платой в пределах ограничения
234 INSERT INTO Employees (E_ID, E_NAME, E_SALARY, E_BORN, D_ID)
235 VALUES (1002, 'Jane Doe', 95000.00, '1995-03-20', 2);
236
237 -- Вставляем нового сотрудника с заработной платой, превышающей ограничение
238 INSERT INTO Employees (E_ID, E_NAME, E_SALARY, E_BORN, D_ID)
239 VALUES (1003, 'Bob Smith', 105000.00, '1980-11-15', 1);
240 Select * FROM Employees;

```

98 %

Результаты Сообщения

	E_ID	E_NAME	E_ADDRESS	E_SALARY	E_BORN	D_ID
16	16	Christopher Hall	404 Birch St	67000.00	1987-04-28	7
17	17	Emma Martinez	505 Maple St	60000.00	1992-12-04	8
18	18	Ryan Taylor	606 Ash St	57000.00	1988-07-21	9
19	19	Grace Robinson	707 Fir St	61000.00	1995-03-09	10
20	20	Brandon Young	808 Pine St	63000.00	1986-10-13	2
21	21	Olav Olsen	111 Bjork St	65000.00	1993-08-12	1
22	22	Ingrid Pedersen	222 Gran St	70000.00	1994-02-19	2
23	1002	Jane Doe	NULL	95000.00	1995-03-20	2