Практика по базам данных

2020

ОТЧЕТ

Платонова Екатерина

243 группа

Предметная область: «Отдел Кадров»

Реализация: MS SQL Server

Содержание

[ОПИСАНИЕ СИСТЕМЫ](file:///C:\Users\natalya\Desktop\Отчет%2002_Аэропорт.docx#_heading=h.30j0zll) **2**

[Требования](file:///C:\Users\natalya\Desktop\Отчет%2002_Аэропорт.docx#_heading=h.1fob9te) 2

[Модель данных](file:///C:\Users\natalya\Desktop\Отчет%2002_Аэропорт.docx#_heading=h.3znysh7) 2

[Функциональность](file:///C:\Users\natalya\Desktop\Отчет%2002_Аэропорт.docx#_heading=h.2et92p0) 2

[Серверная часть](file:///C:\Users\natalya\Desktop\Отчет%2002_Аэропорт.docx#_heading=h.tyjcwt) 3

[Клиентская часть](file:///C:\Users\natalya\Desktop\Отчет%2002_Аэропорт.docx#_heading=h.rqt7dafcw4n3) 3

[СКРИПТЫ](file:///C:\Users\natalya\Desktop\Отчет%2002_Аэропорт.docx#_heading=h.1t3h5sf) **4**

[Серверная часть](file:///C:\Users\natalya\Desktop\Отчет%2002_Аэропорт.docx#_heading=h.4d34og8) 5

[Хранимые процедуры и функции](file:///C:\Users\natalya\Desktop\Отчет%2002_Аэропорт.docx#_heading=h.2s8eyo1) 5

[Триггеры](file:///C:\Users\natalya\Desktop\Отчет%2002_Аэропорт.docx#_heading=h.17dp8vu) 6

[Представления](file:///C:\Users\natalya\Desktop\Отчет%2002_Аэропорт.docx#_heading=h.ues0p47kbu35) 6

[Клиентская часть](file:///C:\Users\natalya\Desktop\Отчет%2002_Аэропорт.docx#_heading=h.26in1rg) 8

[Создание и заполнение базы данных](file:///C:\Users\natalya\Desktop\Отчет%2002_Аэропорт.docx#_heading=h.lnxbz9) 9

# ОПИСАНИЕ СИСТЕМЫ

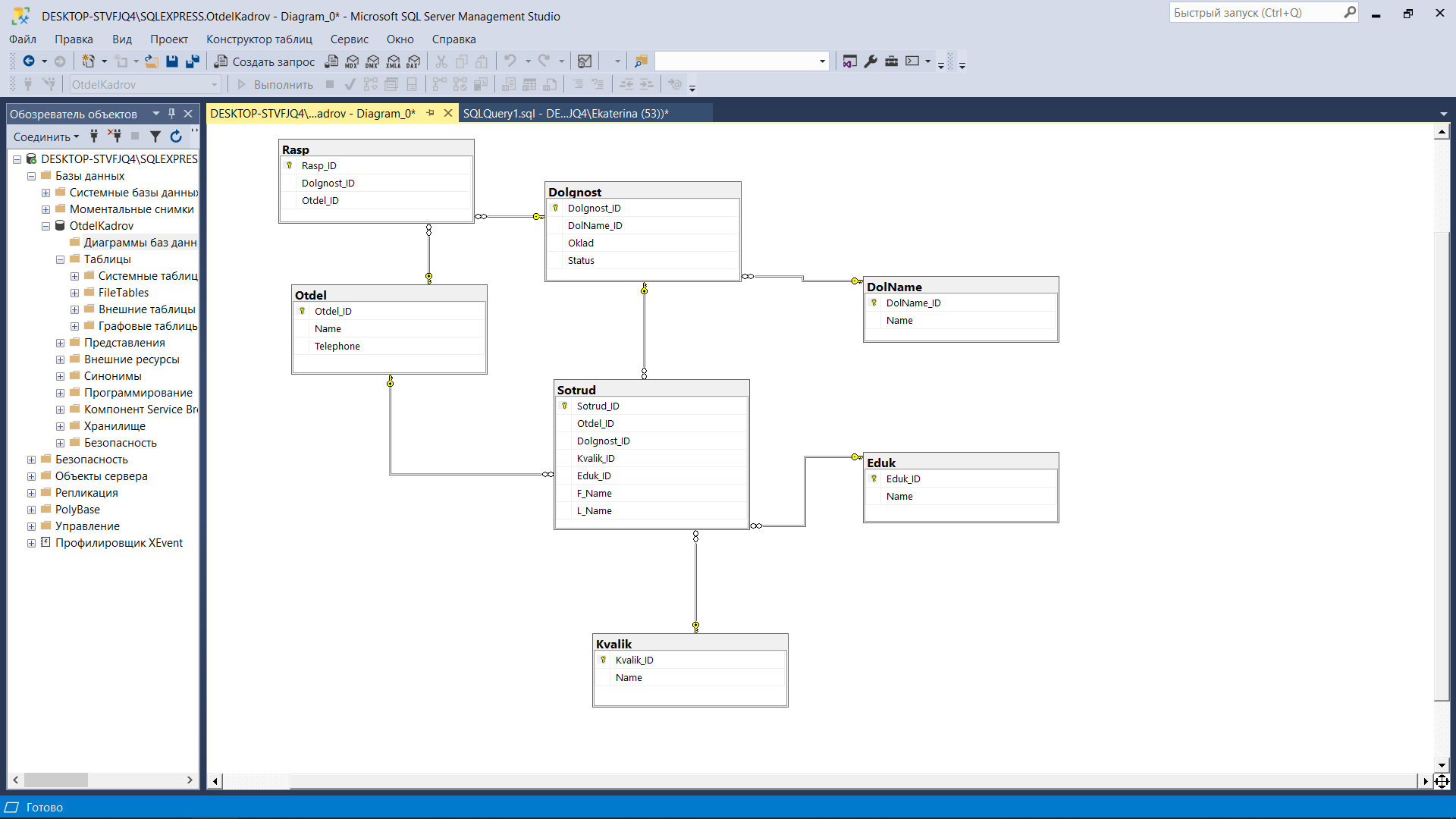
## Требования

Система предназначена для быстрого доступа к информации в формуляре, хранящем информацию о каждом отделе, сотруднике, в этом отделе, и штатном расписании по каждому отделу.

Каждая позиция штатного расписания занята, на текущий момент, только одним сотрудником (либо пустовать). Так же, учитывается, что сотрудник отдела может иметь не только основную должность, но и другие – по совместительству.

По запросу выдается штатное расписание по каждому отделу с указанием вакантных должностей, краткие списки сотрудников по подразделениям, подробные списки сотрудников с указанием данных об основном образовании и занимаемых на текущих момент должностей.

**Модель данных**



## Функциональность

### Серверная часть

|  |  |  |
| --- | --- | --- |
| **Хранимые процедуры/функции** | **Реалзация** | **Комментарии** |
| Добавление должности | AddNewDolgnost() | (название, отдел, оклад, …) |
| Добавление отдела | AddNewOtdel() | (название, номер, …) |
| Статус должности | UpdateStatus () | (номер должности)  Изменение статуса «свободна» |
| Кол-во сотрудников в конкретном отделе | GetCountSotrudOtdel () | (отдел), функция |
| Добавление сотрудника |  | (фамилия, имя, должность…) |

|  |  |  |
| --- | --- | --- |
| **Триггеры** | **Реализация** | **Комментарии** |
| Запрет на удаление физ. лица | Tr\_Del\_Sotrud |  |
| Запрет на изменение Status в таблице Dolgnost | Tr\_Edit\_Status | Т.к. для изменения есть соответствующая функция |
| Формирование номера при добавлении струдника |  | С использованием секвенции |
| Контроль за кол-вом мест на должность |  | Не должно быть 0 мест на основную должность и 0 мест на второстепенную одновременно |
| Запрет на добавление должности сразу с Status=’Занята’ | Tr\_Insert\_Status | Когда добавляется должность она еще свободна |

|  |  |  |
| --- | --- | --- |
| **Представления** | **Реализация** | **Комментарии** |
| Свободные должности | V\_Status1 | (отдел,название должности, номер должности, …) |
| Занятные должности | V\_Spisok | (отдел,название должности, номер должности, …) |
| Отделы | V\_Otdel | (Название, направление деятельности) |
| Сотрудники |  | (имя,фамилия,должность) |

### 

### **Клиентская часть**

|  |  |  |  |
| --- | --- | --- | --- |
| **Экранные формы основные** | **Дополнительные** | **Реализация** | **Что можно использовать из**  **серверной части** |
| Должность |  | 1. Данные об отделе , включая количество сотрудников 2. Список должностей в отделах со статусом (занята должность или нет) |  |
|  | Добавить должность |  | AddNewDolgnost() |
|  | Изменить данные |  | UpdateStatus () |
|  | Фильтры | 1. Средний оклад   Среди занятых должностей | V\_Status |
| Сотрудники |  | 1. Должности, с количеством работающих на них сотрудников 2. Списки сотрудников с указанием данных об основном образовании и квалификации 3. Данные о сотрудниках с их окладом |  |
|  | Добавление |  | Add\_New\_Sotrud() |
|  | Удаление |  | Tr\_Del\_Sotrud |
|  | Фильтр | 1. Сортировка по имени 2. Сотрудники, имеющие зарплату 35000 |  |
| Отделы |  | 1. Отделы, у которых есть свободные должности | V\_Otdel |
|  | Добавление отдела |  | AddNewOtdel() |
|  | Кол-во должностей |  | GetCountDolgnost() |
| Служебные запросы |  |  |  |
|  |  | 1. Список всех сотрудников с указанием их должности 2. Список всех должностей с указанием сотрудников на этих должностях (Если должность пустует – NULL) 3. Штатное расписание отделов 4. Общий список всех Сотрудников и Должностей |  |

## Серверная часть

### Хранимые процедуры и функции

-------------------------------------------------------------------------------------------------------------------------------------------------------

--Добавление нового отдела

-------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE PROCEDURE AddNewOtdel  
@Otdel\_ID AS INTEGER,  
@Name AS VARCHAR(30),

[@Telephone](https://vk.com/telephone) AS VARCHAR(20),  
@Dolgnost\_ID AS INTEGER  
AS  
BEGIN  
DECLARE @New\_Otdel\_ID INTEGER;  
SELECT @New\_Otdel\_ID = MAX(Otdel\_ID) + 1 FROM Otdel;  
DECLARE @New\_Rasp\_ID INTEGER;  
SELECT @New\_Rasp\_ID = MAX(Rasp\_ID) + 1 FROM Rasp;  
INSERT INTO Otdel(Otdel\_ID, Name, Telephone)  
VALUES (@New\_Otdel\_ID, @Name, [@Telephone](https://vk.com/telephone))  
INSERT INTO Rasp (Rasp\_ID, Dolgnost\_ID, Otdel\_ID)  
VALUES (@New\_Rasp\_ID, @Dolgnost\_ID, @New\_Otdel\_ID);  
SELECT\* FROM Otdel  
WHERE Otdel\_ID = @New\_Otdel\_ID  
SELECT\* FROM Rasp  
WHERE Rasp\_ID=@New\_Rasp\_ID  
END

--Вызов  
--EXECUTE AddNewOtdel @Otdel\_ID=5, @Dolgnost\_ID=5, @Name='Пятый' , @Telephone='89114789021'  
--Удаление  
--DROP PROCEDURE AddNewOtdel;

-----------------------------------------------------------------------------------------------------------------------------------------------

-- Статус должности(перевод в состояние «занята»

------------------------------------------------------------------------------------------------------------------------------------------------

CREATE PROCEDURE UpdateStatus  
@Status AS INTEGER,  
@Dolnost\_ID AS INTEGER  
AS  
BEGIN  
DECLARE @New\_Dolgnost\_ID INTEGER;  
IF @Status=0  
UPDATE Dolgnost SET Status=1 WHERE Dolgnost\_ID=@Dolnost\_ID  
END;  
  
--Вызов  
--EXECUTE UpdateStatus @Status=1, @Dolnost\_ID=5;  
--Удаление  
--DROP PROCEDURE UpdateStatus

-----------------------------------------------------------------------------------------------------------------------------------------------

-- Количество сотрудников в конкретном отделе

CREATE FUNCTION GetCountSotrud (@Otdel\_ID INTEGER)

RETURNS INTEGER

BEGIN

DECLARE @Result INTEGER;

SELECT @Result=COUNT(\*) FROM Otdel, Sotrud

WHERE Otdel.Otdel\_ID=@Otdel\_ID

AND Sotrud.Otdel\_ID=@Otdel\_ID

RETURN @Result;

END;

--Вызов

--SELECT dbo.GetCountSotrud (1);

--Удаление

--DROP FUNCTION GetCountSotrud;

### **Триггеры**

-----------------------------------------------------------------------------------------------------------------------------------------------

-- Запрет на удаление сотрудника

----------------------------------------------------------------------------------------------------------------------------------------------

CREATE TRIGGER TR\_Del\_Sotrud ON Sotrud FOR DELETE

AS

ROLLBACK;

-- Вызов

--DELETE FROM Sotrud;

-- Удаление

--DROP TRIGGER TR\_Del\_Sotrud ;

-----------------------------------------------------------------------------------------------------------------------------------------------

-- Запрет на удаление отдела

-----------------------------------------------------------------------------------------------------------------------------------------------

CREATE TRIGGER TR\_Del\_Otdel ON Otdel FOR DELETE

AS

ROLLBACK;

-- Вызов

--DELETE FROM Otdel;

-- Удаление

--DROP TRIGGER TR\_Del\_Otdel ;

-----------------------------------------------------------------------------------------------------------------------------------------------

--Запрет на добавление должности с Status=1

-----------------------------------------------------------------------------------------------------------------------------------------------

CREATE TRIGGER Tr\_Insert\_Status ON Dolgnost

FOR INSERT

AS

IF (SELECT Status FROM inserted)=1 ROLLBACK;

--Вызов

--INSERT INTO Dolgnost(Dolgnost\_ID, DolName\_ID, Oklad, Status) VALUES (16, 1, 23020, 1);

--Удаление

--DROP TRIGGER Tr\_Insert\_Status;

### 

### **Представления**

----------------------------------------------------------------------------------------------------------------------------------------------

-- Cписок должностей и сотрудников

------------------------------------------------------------------------------------------------------------------------------------------------

CREATE VIEW V\_Spisok (F\_Name, L\_Name, DolName\_ID)

AS

SELECT Sotrud.F\_Name, Sotrud.L\_Name, Dolgnost.DolName\_ID

FROM Dolgnost, Sotrud

--удаление

--DROP VIEW V\_Spisok;

-----------------------------------------------------------------------------------------------------------------------------------------------

-- Свободные должности

-----------------------------------------------------------------------------------------------------------------------------------------------

CREATE VIEW V\_Status1 (F\_Name, L\_Name, Oklad)

AS

(SELECT Sotrud.F\_Name, Sotrud.L\_Name, Dolgnost.Oklad

FROM Dolgnost, Sotrud

WHERE Status =0);

--удаление

--DROP VIEW V\_Status1;

-----------------------------------------------------------------------------------------------------------------------------------------------

-- Отделы

-----------------------------------------------------------------------------------------------------------------------------------------------

CREATE VIEW V\_Otdel (Name,Telephone)

AS

(SELECT Name, Telephone FROM Otdel);

--удаление

--DROP VIEW V\_Otdel;

**Клиентская часть**

(запросы для экранных форм и отчетов)

1. Данные об отделе, включая количество сотрудников

SELECT [Otdel, Name](https://vk.com/away.php?to=http%3A%2F%2FOtdel.Name&cc_key=" \t "_blank), Otdel.Telephone, cnt  
FROM Otdel JOIN( SELECT Otdel.Otdel\_ID, COUNT(\*) AS cnt  
FROM Otdel JOIN Sotrud on  
Sotrud.Otdel\_ID=Sotrud.Otdel\_ID  
GROUP BY Otdel.Otdel\_ID)SSS  
ON Otdel.Otdel\_ID=SSS.Otdel\_ID

1. Список должностей в отделах со статусом (занята должность или нет)

SELECT AA.Name, BB.Name FROM

(SELECT Rasp\_id, Name FROM Otdel FULL JOIN Rasp ON Otdel.Otdel\_ID = Rasp.Otdel\_ID) AS AA

RIGHT JOIN

(SELECT Rasp\_ID, Name FROM Rasp, Dolgnost, DolName

WHERE Dolgnost.DolName\_ID = DolName.DolName\_ID AND Rasp.Dolgnost\_ID = Dolgnost.Dolgnost\_ID)

AS BB ON AA.Rasp\_ID = BB.Rasp\_ID

ORDER BY 1

1. Средний оклад по занимаемым должностям

SELECT AVG(Oklad) FROM Dolgnost WHERE Status=1

1. Должности, с количеством работающих на них сотрудников

SELECT Dolgnost\_ID , COUNT(\*) AS quantity FROM Sotrud

GROUP BY Dolgnost\_ID ORDER BY Dolgnost\_ID

1. Списки сотрудников с указанием данных об основном образовании и квалификации

SELECT L\_Name, F\_Name, Eduk.Eduk\_ID, Kvalik.Kvalik\_ID FROM Sotrud, Eduk, Kvalik

WHERE Eduk.Eduk\_ID=Sotrud.Eduk\_ID AND Kvalik.Kvalik\_ID =Sotrud.Kvalik\_ID

1. Данные о сотрудниках с их окладом

SELECT Sotrud\_ID, L\_Name, F\_Name, Oklad

FROM Sotrud LEFT JOIN (SELECT Dolgnost\_ID, Oklad FROM Dolgnost) AS TTT

ON Sotrud.Dolgnost\_ID = TTT.Dolgnost\_ID

1. Сортировка по имени

SELECT Sotrud.F\_Name, Sotrud.L\_Name FROM Sotrud

ORDER BY F\_Name, L\_Name

1. Сотрудники, имеющие зарплату 35000

SELECT L\_Name, F\_Name, Oklad

FROM Sotrud, Dolgnost

GROUP BY L\_Name, F\_Name, Oklad

HAVING Oklad=35000

1. Отделы, у которых есть свободные должности

SELECT Dolgnost\_ID, Otdel\_ID,Status FROM Otdel, Dolgnost

WHERE Status=0

1. Список всех сотрудников с указанием их должности

SELECT Sotrud.L\_Name, Sotrud.F\_Name, Dolgnost.Dolgnost\_ID FROM Sotrud, Dolgnost

WHERE Sotrud.Sotrud\_ID=Dolgnost.Dolgnost\_ID

1. Список всех должностей с указанием сотрудников на этих должностях (Если должность пустует – NULL)

SELECT F\_Name, L\_Name, Name FROM

(SELECT Dolgnost.Dolgnost\_ID, F\_Name, L\_Name FROM Sotrud FULL JOIN Dolgnost

ON Sotrud.Dolgnost\_ID = Dolgnost.Dolgnost\_ID) AS AA

RIGHT JOIN

(SELECT Dolgnost.Dolgnost\_ID, Name FROM Dolgnost, DolName

WHERE Dolgnost.DolName\_ID = DolName.DolName\_ID)

AS BB ON AA.Dolgnost\_ID = BB.Dolgnost\_ID

ORDER BY 1

1. Штатное расписание отделов

SELECT Otdel.Otdel\_ID, Dolgnost.Dolgnost\_ID

FROM Rasp LEFT JOIN Otdel ON Rasp.Otdel\_ID=Otdel.Otdel\_ID

RIGHT JOIN Dolgnost ON Rasp.Dolgnost\_ID = Dolgnost.Dolgnost\_ID

ORDER BY 1

1. Общий список всех Сотрудников и Должностей

SELECT L\_Name, F\_Name FROM Sotrud

UNIUN

SELECT Dolgnost\_ID FROM Dolgnost

**Создание и заполнение базы**

------------------------------------------------------------------------------------------------------------------------

-- Создание таблиц и PK

------------------------------------------------------------------------------------------------------------------------

CREATE TABLE Otdel(

Otdel\_ID INTEGER NOT NULL,

Name VARCHAR(30) NOT NULL,

Telephone VARCHAR(20) NOT NULL,

CONSTRAINT Otdel\_PK PRIMARY KEY (Otdel\_ID)

)

;

CREATE TABLE Sotrud(

Sotrud\_ID INTEGER NOT NULL,

Otdel\_ID INTEGER NOT NULL,

Dolgnost\_ID INTEGER NOT NULL,

Kvalik\_ID INTEGER NOT NULL,

Eduk\_ID INTEGER NOT NULL,

F\_Name VARCHAR(30) NOT NULL,

L\_Name VARCHAR(30) NOT NULL,

CONSTRAINT Sotrud\_PK PRIMARY KEY (Sotrud\_ID)

)

;

CREATE TABLE Rasp(

Rasp\_ID INTEGER NOT NULL,

Dolgnost\_ID INTEGER NOT NULL,

Otdel\_ID INTEGER NOT NULL,

CONSTRAINT Rasp\_PK PRIMARY KEY (Rasp\_ID)

)

;

CREATE TABLE Dolgnost(

Dolgnost\_ID INTEGER NOT NULL,

DolName\_ID INTEGER NOT NULL,

Oklad INTEGER NOT NULL,

Status INTEGER NOT NULL,

CONSTRAINT Dolgnost\_PK PRIMARY KEY (Dolgnost\_ID)

)

;

CREATE TABLE Eduk(

Eduk\_ID INTEGER NOT NULL,

Name VARCHAR(30) NOT NULL,

CONSTRAINT Eduk\_PK PRIMARY KEY (Eduk\_ID)

)

;

CREATE TABLE Kvalik(

Kvalik\_ID INTEGER NOT NULL,

Name VARCHAR(30) NOT NULL,

CONSTRAINT Kvalik\_PK PRIMARY KEY ( Kvalik\_ID)

)

;

CREATE TABLE DolName(

DolName\_ID INTEGER NOT NULL,

Name VARCHAR(30) NOT NULL,

CONSTRAINT DolName\_PK PRIMARY KEY ( DolName\_ID)

)

;

-------------------------------------------------------------—---------------------------------------------------------------------

-- Создание FK

-------------------------------------------------------------—----------------------------------------------------------------------

ALTER TABLE Sotrud ADD CONSTRAINT FK\_Sotrud\_Otdel

FOREIGN KEY (Otdel\_ID)

REFERENCES Otdel(Otdel\_ID)

;

ALTER TABLE Sotrud ADD CONSTRAINT FK\_Sotrud\_Dolgnost

FOREIGN KEY (Dolgnost\_ID)

REFERENCES Dolgnost(Dolgnost\_ID)

;

ALTER TABLE Sotrud ADD CONSTRAINT FK\_Sotrud\_Kvalik

FOREIGN KEY (Kvalik\_ID)

REFERENCES Kvalik(Kvalik\_ID)

;

ALTER TABLE Sotrud ADD CONSTRAINT FK\_Sotrud\_Eduk

FOREIGN KEY (Eduk\_ID)

REFERENCES Eduk(Eduk\_ID)

;

ALTER TABLE Dolgnost ADD CONSTRAINT FK\_Dolgnost\_DolName

FOREIGN KEY (DolName\_ID)

REFERENCES DolName(DolName\_ID)

;

ALTER TABLE Rasp ADD CONSTRAINT FK\_Rasp\_Dolgnost

FOREIGN KEY (Dolgnost\_ID)

REFERENCES Dolgnost(Dolgnost\_ID)

;

ALTER TABLE Rasp ADD CONSTRAINT FK\_Rasp\_Otdel

FOREIGN KEY (Otdel\_ID)

REFERENCES Otdel(Otdel\_ID)

;

-------------------------------------------------------------—-------------------------

-- Индесы

-------------------------------------------------------------—-------------------------

CREATE index idx\_Dolgnost\_Status on Dolgnost(Status);

CREATE index idx\_Otdel\_Telephone on Otdel(Telephone);

CREATE index idx\_Sotrud\_Name on Sotrud (F\_Name, L\_Name);

-------------------------------------------------------------—--------------------------

--Заполнение таблиц тестовыми данными

-------------------------------------------------------------—--------------------------

INSERT INTO DolName(DolName\_ID, Name) VALUES (1, 'Директор' );

INSERT INTO DolName(DolName\_ID, Name) VALUES (2, 'Программист' );

INSERT INTO DolName(DolName\_ID, Name) VALUES (3, 'Секретарь' );

INSERT INTO DolName(DolName\_ID, Name) VALUES (4, 'Корректор' );

INSERT INTO DolName(DolName\_ID, Name) VALUES (5, 'Бухгалтер' );

INSERT INTO DolName(DolName\_ID, Name) VALUES (6, 'Заместитель Директора' );

INSERT INTO DolName(DolName\_ID, Name) VALUES (7, 'Системный Администратор' );

INSERT INTO DolName(DolName\_ID, Name) VALUES (8, 'Редактор' );

INSERT INTO Kvalik(Kvalik\_ID, Name) VALUES (0, 'Нет квалификации');

INSERT INTO Kvalik(Kvalik\_ID, Name) VALUES (1, 'Бакалавр');

INSERT INTO Kvalik(Kvalik\_ID, Name) VALUES (2, 'Магистр');

INSERT INTO Kvalik(Kvalik\_ID, Name) VALUES (3, 'Доктор');

INSERT INTO Kvalik(Kvalik\_ID, Name) VALUES (4, 'Кандидант');

INSERT INTO Eduk(Eduk\_ID, Name) VALUES (1, 'Высшее');

INSERT INTO Eduk(Eduk\_ID, Name) VALUES (2, 'Среднее');

INSERT INTO Eduk(Eduk\_ID, Name) VALUES (3, 'Неоконченное высшее');

INSERT INTO Dolgnost(Dolgnost\_ID, DolName\_ID, Oklad, Status) VALUES (1, 1, 23000, 0);

INSERT INTO Dolgnost(Dolgnost\_ID, DolName\_ID, Oklad, Status) VALUES (2, 1, 40000, 0);

INSERT INTO Dolgnost(Dolgnost\_ID, DolName\_ID, Oklad, Status) VALUES (3, 2, 40000, 1);

INSERT INTO Dolgnost(Dolgnost\_ID, DolName\_ID, Oklad, Status) VALUES (4, 3, 60000, 1);

INSERT INTO Dolgnost(Dolgnost\_ID, DolName\_ID, Oklad, Status) VALUES (5, 4, 45000, 1);

INSERT INTO Dolgnost(Dolgnost\_ID, DolName\_ID, Oklad, Status) VALUES (6, 5, 15000, 0);

INSERT INTO Dolgnost(Dolgnost\_ID, DolName\_ID, Oklad, Status) VALUES (7, 3, 50000, 1);

INSERT INTO Dolgnost(Dolgnost\_ID, DolName\_ID, Oklad, Status) VALUES (8, 2, 78000, 0);

INSERT INTO Dolgnost(Dolgnost\_ID, DolName\_ID, Oklad, Status) VALUES (9, 4, 25000, 1);

INSERT INTO Dolgnost(Dolgnost\_ID, DolName\_ID, Oklad, Status) VALUES (10, 6, 70000, 0);

INSERT INTO Dolgnost(Dolgnost\_ID, DolName\_ID, Oklad, Status) VALUES (11, 1, 100000,1);

INSERT INTO Dolgnost(Dolgnost\_ID, DolName\_ID, Oklad, Status) VALUES (12, 2, 10000, 0);

INSERT INTO Dolgnost(Dolgnost\_ID, DolName\_ID, Oklad, Status) VALUES (13, 7, 60000, 1);

INSERT INTO Dolgnost(Dolgnost\_ID, DolName\_ID, Oklad, Status) VALUES (14, 8, 35000, 1);

INSERT INTO Otdel(Otdel\_ID, Name, Telephone) VALUES (1, 'Первый', '+79234567890');

INSERT INTO Otdel(Otdel\_ID, Name, Telephone) VALUES (2, 'Второй', '+7678901234');

INSERT INTO Otdel(Otdel\_ID, Name, Telephone) VALUES (3, 'Третий', '+7567540123');

INSERT INTO Otdel(Otdel\_ID, Name, Telephone) VALUES (4, 'Четвертый', '+7532890123');

INSERT INTO Sotrud(Sotrud\_ID, Otdel\_ID, Dolgnost\_ID, F\_Name,L\_Name, Kvalik\_ID, Eduk\_ID) VALUES (1, 1, 1, 'Иван',' Иванов',1, 1);

INSERT INTO Sotrud(Sotrud\_ID, Otdel\_ID, Dolgnost\_ID, F\_Name,L\_Name, Kvalik\_ID, Eduk\_ID) VALUES (2, 2, 2, 'Иван',' Федоров',0, 2);

INSERT INTO Sotrud(Sotrud\_ID, Otdel\_ID, Dolgnost\_ID, F\_Name,L\_Name, Kvalik\_ID, Eduk\_ID) VALUES (3, 3, 3, 'Дмитрий',' Иванов',3, 1);

INSERT INTO Sotrud(Sotrud\_ID, Otdel\_ID, Dolgnost\_ID, F\_Name,L\_Name, Kvalik\_ID, Eduk\_ID) VALUES (4, 4, 4, 'Александр',' Фомов',0, 2);

INSERT INTO Sotrud(Sotrud\_ID, Otdel\_ID, Dolgnost\_ID, F\_Name,L\_Name, Kvalik\_ID, Eduk\_ID) VALUES (5, 1, 5, 'Федор',' Булкин',2, 2);

INSERT INTO Sotrud(Sotrud\_ID, Otdel\_ID, Dolgnost\_ID, F\_Name,L\_Name, Kvalik\_ID, Eduk\_ID) VALUES (6, 2, 6, 'Владислав',' Гусь',2, 3);

INSERT INTO Sotrud(Sotrud\_ID, Otdel\_ID, Dolgnost\_ID, F\_Name,L\_Name, Kvalik\_ID, Eduk\_ID) VALUES (7, 3, 7, 'Евгений',' Дулкин',1 ,3);

INSERT INTO Sotrud(Sotrud\_ID, Otdel\_ID, Dolgnost\_ID, F\_Name,L\_Name, Kvalik\_ID, Eduk\_ID) VALUES (8, 4, 8, 'Анастасия',' Уткина',3, 3);

INSERT INTO Sotrud(Sotrud\_ID, Otdel\_ID, Dolgnost\_ID, F\_Name,L\_Name, Kvalik\_ID, Eduk\_ID) VALUES (9, 1, 9, 'Даниил',' Лебедев',2, 1);

INSERT INTO Rasp(Rasp\_ID, Dolgnost\_ID, Otdel\_ID) VALUES (1, 1, 1);

INSERT INTO Rasp(Rasp\_ID, Dolgnost\_ID, Otdel\_ID) VALUES (5, 2, 1);

INSERT INTO Rasp(Rasp\_ID, Dolgnost\_ID, Otdel\_ID) VALUES (6, 3, 1);

INSERT INTO Rasp(Rasp\_ID, Dolgnost\_ID, Otdel\_ID) VALUES (7, 4, 1);

INSERT INTO Rasp(Rasp\_ID, Dolgnost\_ID, Otdel\_ID) VALUES (2, 5, 2);

INSERT INTO Rasp(Rasp\_ID, Dolgnost\_ID, Otdel\_ID) VALUES (8, 6, 2);

INSERT INTO Rasp(Rasp\_ID, Dolgnost\_ID, Otdel\_ID) VALUES (9, 7, 2);

INSERT INTO Rasp(Rasp\_ID, Dolgnost\_ID, Otdel\_ID) VALUES (10, 8, 2);

INSERT INTO Rasp(Rasp\_ID, Dolgnost\_ID, Otdel\_ID) VALUES (3, 9, 3);

INSERT INTO Rasp(Rasp\_ID, Dolgnost\_ID, Otdel\_ID) VALUES (12, 10, 3);

INSERT INTO Rasp(Rasp\_ID, Dolgnost\_ID, Otdel\_ID) VALUES (13, 11, 3);

INSERT INTO Rasp(Rasp\_ID, Dolgnost\_ID, Otdel\_ID) VALUES (14, 12, 4);

INSERT INTO Rasp(Rasp\_ID, Dolgnost\_ID, Otdel\_ID) VALUES (4, 13, 4);

INSERT INTO Rasp(Rasp\_ID, Dolgnost\_ID, Otdel\_ID) VALUES (11, 14, 3);

-------------------------------------------------------------—------------------------------

-- Удаление таблиц

------------------------------------------------------------—--------------------------------

/\*

DROP TABLE Rasp;

DROP TABLE Sotrud;

DROP TABLE Otdel;

DROP TABLE Dolgnost;

DROP TABLE Eduk;

DROP TABLE Kvalik;

DROP TABLE DolName;

\*/