**Министерство транспорта Российской Федерации**

**Федеральное государственное автономное образовательное  
учреждение высшего образования  
«РУТ (МИИТ)»**

**Институт транспортной техники и систем управления**

**Кафедра «Управление и защита информации»**

КУРСОВОЙ ПРОЕКТ

по дисциплине

**«Основы построения защищенных баз данных»**

**на тему  
«База данных компания по услугам связи»**

Выполнил: ст. гр. ТКИ-441  
Пирогова К.д.  
Проверил: Васильева М.А.

Москва 2023

Оглавление

[ЦЕЛЬ КУРСОВОГО ПРОЕКТА 5](#_Toc136855496)

[ЗАДАНИЕ НА КУРСОВОВОЙ ПРОЕКТ 5](#_Toc136855497)

[ПРОЕКТИРОВАНИЕ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ 6](#_Toc136855498)

[1 Инфологическое проектирование 6](#_Toc136855499)

[1.1 Анализ предметной области 6](#_Toc136855500)

[1.2 Анализ информационных задач и круга пользователей системы 7](#_Toc136855501)

[2 Логическое проектирование реляционной БД 8](#_Toc136855502)

[2.1 Составление реляционных отношений 8](#_Toc136855503)

[3 Физическое проектирование БД 11](#_Toc136855504)

[3.1 Разработка скриптов на создание базы данных и таблиц 11](#_Toc136855505)

[3.2 Разработка скриптов на добавление данных в таблицы 13](#_Toc136855506)

[3.3 Разработка необходимых запросов, представлений (view), процедур, функций и триггеров 16](#_Toc136855507)

[Список литературы 19](#_Toc136855508)

# ЦЕЛЬ КУРСОВОГО ПРОЕКТА

Целью курсового проекта является изучение методов и закрепление знаний в проектировании реляционных баз данных (РБД) в системе управления базами данных (СУБД) Postgres Pro.

# ЗАДАНИЕ НА КУРСОВОВОЙ ПРОЕКТ

В данном курсовом проекте ставится задача разработать РБД в СУБД Postgres Pro по заданной теме «Компания по услугам связи». Проектирование РБД проводится с помощью метода «Сущность-связь». Проверка построенной модели РБД осуществляется с помощью метода нормализации отношений.

Описание предметной области.

БД создается для компании по услугам связи.

Ниже приведено описание основных сущностей предметной области:

1. Таблица "Contracts", которая содержит информацию о клиентах, работниках заключивших контракт, дату заключения и тариф, включая их id.
2. Таблица "Customers" содержит информацию о клиентах, включая их id, ФИО, паспортные данные, номер телефона и id контракта к ним относящегося.
3. Таблица "Employees" содержит информацию об о работниках, включая их id, ФИО, паспортные данные, номер телефона и id должности к ним относящейся.
4. Таблица "Posts" содержит информацию о должностях, их id, название.
5. Таблица "ServiceNames" содержит информацию о дополнительных услугах, включая их id, название, цену.
6. Таблица "Tariffs" содержит информацию о тарифах, включая их id, название, цену и id дополнительной услуги, входящей в стоимость тарифа.

# ПРОЕКТИРОВАНИЕ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ

## Инфологическое проектирование

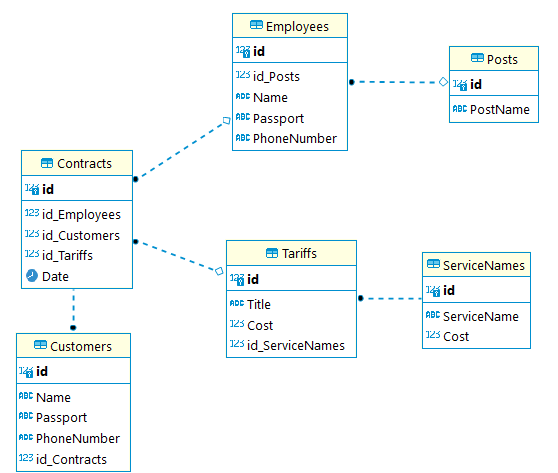
### Анализ предметной области

База данных создаётся для компании по услугам связи.

Выделим базовые сущности этой предметной области:

* Contracts – заключенные контракты с клиентами.
* Customers – клиенты.
* Employees – работники.
* Posts - должности.
* ServiceNames – дополнительные услуги.
* Tariffs – тарифы.

ER-диаграмма БД, приведена на (Рисунке 1).



*Рисунок 1 - ER-диаграмма*

### Анализ информационных задач и круга пользователей системы

Система создаётся для обслуживания следующих групп пользователей:

* Администратор
* Директор
* Клиенты

Определим границы информационной поддержки пользователей:

**1) Функциональные возможности:**

* ведение БД (запись, чтение, модификация, удаление в архив);
* обеспечение логической непротиворечивости БД;
* обеспечение защиты данных от несанкционированного или случайного доступа (определение прав доступа);
* реализация наиболее часто встречающихся запросов в готовом виде.

**2) Готовые запросы:**

* выбрать названия из таблицы "ServiceNames";
* показывать список клиентов в соответствии с их тарифами;
* выбрать тарифы, цена которых меньше заданной;
* выбрать контракты, которые были заключены в заданный промежуток времени.

## Логическое проектирование реляционной БД

### Составление реляционных отношений

Каждое реляционное отношение соответствует одной сущности (объекту ПрО), и в него вносятся все атрибуты сущности. Для каждого отношения необходимо определить первичный ключ и внешние ключи (если они есть). В том случае, если базовое отношение не имеет потенциальных ключей, вводится *суррогатный первичный ключ*, который не несёт смысловой нагрузки и служит только для идентификации записей.

Предметная область включает в себя несколько базовых сущностей: жанры, стили, альбомы, песни(треки) и исполнители.

1. Отношение один-ко-многим между Employees и Contracts: один работник может заключить много контрактов, но каждый контракт связан только с одним работником. Это отношение устанавливается через внешний ключ id\_Employees в таблице Contracts, который ссылается на первичный ключ id в таблице Employees.
2. Отношение один-ко-многим между Posts и Employees: один работник может иметь несколько должностей, но каждая должность связана только с одним работником. Это отношение устанавливается через внешний ключ id\_Posts в таблице Employees, который ссылается на первичный ключ id в таблице Posts.

Таким образом, были определены связи между сущностями, которые позволят хранить информацию и связывать данные между собой.

**Таблица 1 – Схема отношения** **Должности** (Posts)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | id | Целое число | Первичный ключ |
| Название | PostName | Текст | Обязательное поле |

Таблица 2 – Схема отношения Работники (Employees)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | id | Целое число | Первичный ключ |
| Имя | Name | Текст | Обязательное поле |
| Должность | id\_Posts | Целое число | Обязательное поле |
| Паспортные данные | Passport | Текст | Обязательное поле |
| Номер телефона | PhoneNumber | Текст | Обязательное поле |

**Таблица 3 – Схема отношения** **Контракты** (Contracts)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | id | Целое число | Первичный ключ |
| Работник | id\_Employees | Целое число | Обязательное поле |
| Клиент | id\_Customers | Целое число | Обязательное поле |
| Тариф | id\_Tariffs | Целое число | Обязательное поле |
| Дата | Date | Дата | Обязательное поле |

**Таблица 4 – Схема отношения** **Тарифов** (Tariffs)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | id | Целое число | Первичный ключ |
| Название | Title | Текст | Обязательное поле |
| Цена | Cost | Деньги | Обязательное поле |
| Доп. услуга | id\_ServiceNames | Целое число | Обязательное поле |

Таблица 5 – Схема отношения Доп. услуг (ServiceNames)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | id | Целое число | Первичный ключ |
| Название | ServiceName | Текст | Обязательное поле |
| Цена | Cost | Деньги | Обязательное поле |

Таблица 6 – Схема отношения Клиентов (Customers)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | id | Целое число | Первичный ключ |
| Имя | Name | Текст | Обязательное поле |
| Контракт | id\_Contracts | Целое число | Обязательное поле |
| Паспортные данные | Passport | Текст | Обязательное поле |
| Номер телефона | PhoneNumber | Текст | Обязательное поле |

## Физическое проектирование БД

### Разработка скриптов на создание базы данных и таблиц

БД «Компания по услугам связи» проектируется в среде Postgres Pro

*Создание базы данных «*Компания по услугам связи*»:*

*Создание таблицы Контрактов:*

CREATE TABLE Contracts

(

ID INTEGER

, ID\_Employees INTEGER

, ID\_Customer INTEGER

, ID\_Tariff INTEGER

, "Date" DATE

);

*Создание таблицы Клиентов:*

CREATE TABLE Customers

(

ID INTEGER

, ID\_Contracts INTEGER

, "Name" TEXT

, "Passport" TEXT

, "PhoneNumber" TEXT

);

*Создание таблицы Работников:*

CREATE TABLE Employees

(

ID INTEGER

, ID\_Post INTEGER

, "Name" TEXT

, "Passport" TEXT

, "PhoneNumber" TEXT

);

*Создание таблицы Должностей:*

CREATE TABLE Posts

(

ID\_Post INTEGER

, "PostName" text

);

*Создание таблицы Услуг:*

CREATE TABLE ServiceNames

(

ID INTEGER

, "ServiceName" TEXT

, "Cost" MONEY

);

*Создание таблицы Тарифов:*

CREATE TABLE Tariffs

(

ID INTEGER

, ID\_Employees INTEGER

, "Title" TEXT

, "Cost" MONEY

, "ServiceName" TEXT

, "Date" DATE

);

### Разработка скриптов на добавление данных в таблицы

*Заполнение таблицы* *контрактов:*

INSERT INTO "Contracts" ("id", "id\_Employees" , "id\_Customers" , "id\_Tariffs" , "Date") VALUES

(1, 2, 1, 2, '2020-04-12')

, (2, 3, 4, 4, '2021-10-03')

, (3, 2, 5, 1, '2019-07-14')

, (4, 4, 2, 2, '2022-11-04')

, (5, 2, 3, 3, '2022-01-09')

, (6, 5, 6, 1, '2023-02-02');

*Заполнение таблицы* *клиентов:*

INSERT INTO "Customers" ("id", "id\_Contracts" , "Name", "Passport", "PhoneNumber") VALUES

(1, 1, 'Александрова Виктория Тимуровна', '29.11.1989/4608783405', '89675498632')

, (2, 2, 'Алексеев Александр Ярославович', '17.05.1999/4515089605', '89996914115')

, (3, 3, 'Соловьев Глеб Никитич', '13.09.1997/4819171234', '89768906755')

, (4, 4, 'Шубина Милена Данииловна', '24.05.1979/4518136389', '89324520815')

, (5, 5, 'Щербаков Пётр Лукич', '9.01.1975/4785614714', '89435617273')

, (6, 6, 'Мартынов Валерий Георгиевич', '11.09.2000/4562517111', '893111222335');

*Заполнение таблицы работников:*

INSERT INTO "Employees" ("id", "id\_Posts" , "Name", "Passport", "PhoneNumber") VALUES

(1, 1, 'Иванов Сергей Викторович', '27.12.1988/4608082405', '89675417632')

, (2, 2, 'Петров Дмитрий Владимирович', '13.04.1979/4515084505', '89996994615')

, (3, 1, 'Сергеева Анастасия Дмитриевна', '7.07.1999/4819251234', '89768906782')

, (4, 3, 'Полежаева Дарья Андреевна', '18.05.1998/4518916389', '89324510815')

, (5, 4, 'Самсонов Игорь Николаевич', '2.01.1989/4785612214', '89435617273')

, (6, 5, 'Сергеев Анатолий Александрович', '10.09.2002/4563017111', '89311122215');

*Заполнение таблицы* *должностей:*

INSERT INTO "Posts" ("id", "PostName") VALUES

(1, 'Продавец-консультант')

, (2, 'Директор')

, (3, 'Менеджер по продажам')

, (4, 'Специалист по продажам')

, (5, 'Кассир');

*Заполнение таблицы* *доп. услуг:*

INSERT INTO "ServiceNames" ("id", "ServiceName", "Cost") VALUES

(1, 'Безлимитный интернет', 2)

, (2, 'Обещанный платеж', 5)

, (3, 'Красивый номер', 800)

, (4, 'Защита от спам звонков', 3)

, (5, 'Бесплатные смс', 1)

, (6, 'Роуминг', 450)

, (7, 'Дополнительный пакет', 160);

*Заполнение таблицы* *тарифов:*

INSERT INTO "Tariffs" ("id", "Title" , "Cost", "id\_ServiceNames") VALUES

(1, 'Тариф+', 990, 1)

, (2, 'ТарифСупер', 800, 2)

, (3, 'ТарифПремиум', 1000, 4)

, (4, 'ТарифЭкстра', 1500, 6);

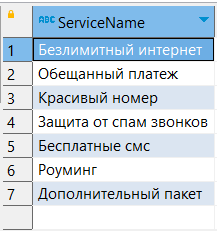
### Разработка необходимых запросов, представлений (view), процедур, функций и триггеров

* Выбрать названия из таблицы "ServiceNames":

SELECT

a."ServiceName"

FROM "ServiceNames" AS a;

**

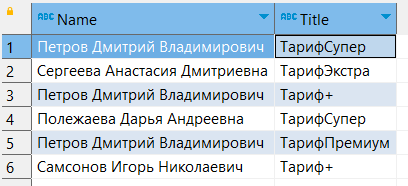
* Показывать список клиентов в соответствии с их тарифами:

SELECT a."Name", c."Title"

FROM "Employees" AS a

JOIN "Contracts" AS b ON a."id" = b."id\_Employees"

JOIN "Tariffs" AS c ON b."id\_Tariffs" = c."id"

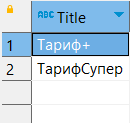
**

* Выбрать тарифы, цена которых меньше заданной:

SELECT a."Title"

FROM "Tariffs" AS a

where a."Cost" < '1000'

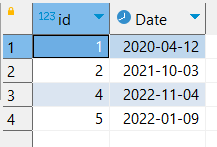
**

* Выбрать контракты, которые были заключены в заданный промежуток времени.

SELECT a."id", a."Date"

FROM "Contracts" AS a

WHERE a."Date" BETWEEN '2020-01-01' AND '2023-01-01'



* Пример триггера на DROP TABLE, на удаление таблицы Tariffs:

CREATE FUNCTION "delete\_tariffs\_relations"() RETURNS TRIGGER AS $$

BEGIN

DELETE FROM "Contracts" WHERE "id\_Tariffs" = OLD.id;

RETURN OLD;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER delete\_tariffs\_relations\_trigger

BEFORE DELETE ON "Tariffs"

FOR EACH ROW

EXECUTE FUNCTION "delete\_tariffs\_relations"();

* Пример процедуры добавляющей новый тариф:

CREATE PROCEDURE create\_tariff(IN p\_title VARCHAR(255), IN p\_cost MONEY, IN p\_id\_serviceNames INTEGER)

LANGUAGE plpgsql

AS $$

BEGIN

INSERT INTO "Tariffs" ("title", "Cost", "id\_ServiceNames") VALUES (p\_title, p\_cost, p\_id\_serviceNames);

END;

$$;

Вызов процедуры:

CALL create\_tariff ('НовыйТариф', '200',7 );

* Пример view, которая показывает название каждого тарифа вместе с названием соответствующей услуги:

CREATE VIEW Tariffs\_Services AS

SELECT

"Tariffs"."Title" AS "Tariff\_Title",

"ServiceNames"."ServiceName" AS "Service\_Name"

FROM

"ServiceNames"

JOIN

"Tariffs" ON "Tariffs"."id\_ServiceNames"= "ServiceNames".id;

Вызов процедуры: SELECT \* FROM Tariffs\_Services

# Список литературы

* + - 1. Васильева М.А., Хобта Д.О., Фильтрация набора данных. Рекомендации по выполнению работы и перечень типовых заданий: Учебно-методическое пособие. Издание второе, исправленное и дополненное–М.:РУТ(МИИТ). 2023.–105с.
      2. Васильева М.А., Меркулов Д.А. Группировка и обобщение данных. Рекомендации по выполнению работы и перечень типовых заданий. Учебно-методическое пособие. М.:РУТ(МИИТ), 2023. 46–с.
      3. Васильева М.А., Ракинцев Н.А. Соединение данных из множества таблиц. Рекомендации по выполнению работы и перечень типовых заданий. Учебно-методическое пособие. М.:РУТ(МИИТ), 2023. 63–с.
      4. Балакина Е.П., Васильева М.А., Филипченко К.М. Информационное обеспечение систем управления. Методические указания к курсовому проектированию. Учебно-методическое пособие. Издание второе, исправленное и дополненное, 2023.102–с.