**Государственное бюджетное профессиональное образовательное учреждение**

**Астраханской области «Астраханский колледж вычислительной техники»**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **К защите допустить:** | | | | | | | |
| **Старший методист** | | | | | | | |
|  | | | | | | | |
|  |  | | | **Л.О. Земцова** | | | |
| **«** |  | **»** |  | | **20** |  | **г.** |

**курсовая работа**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| по МДК | 02.02 | Технология разработки | | | | | | |
|  | | | | и защиты баз данных | | | | |
|  | | 6 семестр | | | | | | |
| Тема курсовой работы | | Разработка БД для автоматизации | | | | | | |
|  | | работы с заказами рекламного агенства | | | | | | |
|  | |  | | | | | | |
| Аквт.09.02.07.Кр00.06ПЗ | | | | | | | | |
| Листов: | | | 20 | |  | | | |
|  | |  | | | | | | |
|  | |  | | | | | | |
| Разработчик | |  | | | | Студент гр. | | ПБ – 32 |
|  | |  | | | | К.С. Буцков | | |
|  | |  | | | |  |  | |
| Руководитель | |  | | | | Преподаватель | | |
|  | |  | | | | И.А. Храмцовский | | |

**2023**

|  |
| --- |
| УТВЕРЖДЕНО |
| Цикловой комиссией |
| специальности 09.02.07 |
| ПЦК \_\_\_\_\_\_\_\_\_ Ю.С. Андрианова |
| «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г. |
| Специальность | *09.02.07 Информационные системы и программирование* | | |
| МДК 04.01 | *Технология разработки и защиты баз данных* | | |
| Группа | *ПБ– 32* | | |
| **ЗАДАНИЕ** | | | |
| **на курсовое проектирование** | | | |
| ***Буцков Кирилл Сергеевич*** | | | |
| (фамилия, имя, отчество студента) | | | |
| Тема курсового проекта | | ***Разработка БД для автоматизации*** | |
| ***работы с заказами рекламного агенства*** | | | |

Содержание пояснительной записки курсового проекта

Содержание

Введение

1. Проектирование базы данных
2. Анализ и описание предметной области информационной системы
3. Проектирование концептуальной модели
4. Построение логической модели, схемы базы данных
5. Выбор СУБД
6. Реализация базы данных
7. Физическое проектирование
8. Написание исходного кода БД
9. Написание запросов, функций и хранимых процедур

Заключение

Список использованных источников

Приложения

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Дата выдачи задания | « |  | » |  | |  | 20 |  | г. |
| Руководитель |  | | | | И.А. Храмцовский | | | | |
| Студент |  | | | | К.С. Буцков | | | | |

**Аннотация**

Данная курсовая работа посвящена созданию базы данных для автоматизации работы с заказами для рекламного агентства. Основной задачей проекта является проектирование функциональной базы данных, которая позволит хранить информацию о клиентах, сотрудниках, заказах, услугах и типах рекламы. Для решения поставленных задач была выбрана СУБД PostgreSQL.

В работе будут рассмотрены основные этапы проектирования баз данных, такие как создание схемы базы данных, определение атрибутов, создание таблиц, создание запросов для работы с данными

**Содержание**

[Введение 5](#_Toc135224632)

[1 Проектирование базы данных 6](#_Toc135224633)

[1.1 Анализ и описание предметной области информационной системы 6](#_Toc135224634)

[1.2 Проектирование концептуальной модели 7](#_Toc135224635)

[1.3 Построение логической модели, схемы базы данных 9](#_Toc135224636)

[1.4 Выбор СУБД 10](#_Toc135224637)

[2 Реализация базы данных 11](#_Toc135224638)

[2.1 Физическое проектирование 11](#_Toc135224639)

[2.2 Написание исходного кода БД 13](#_Toc135224640)

[2.3 Написание запросов, функций и хранимых процедур 15](#_Toc135224641)

[Заключение 19](#_Toc135224642)

[Список использованных источников 20](#_Toc135224643)

[Приложение 21](#_Toc135224644)

**Введение**

Как и многие другие организации, рекламные агентства, занимающиеся предоставлением рекламных услуг, также могут воспользоваться преимуществами автоматизации.

Данная курсовая работа посвящена созданию базы данных для автоматизации работы с заказами для рекламного агентства. Она позволит сократить время и усилия сотрудников, занятых рутинными задачами, такими как учет заказов, расчет стоимости рекламных кампаний и отслеживание статуса заказов.

Целью данной курсовой работы является создание функциональной базы данных для автоматизации работы с заказами для рекламного агентства, что позволит сократить временные и финансовые затраты, повысить эффективность работы и улучшить качество обслуживания клиентов.

# **Проектирование базы данных**

## **Анализ и описание предметной области информационной системы**

Предметной областью данного проекта является автоматизация работы с заказами для рекламного агентства. Рекламное агентство является компанией, которая занимается созданием, производством и продвижением рекламных материалов для своих клиентов. Заказы, которые должна обрабатывать база данных, могут быть различными: от создания рекламных баннеров до размещения рекламы в социальных сетях.

В данной предметной области важно учитывать несколько факторов. Во-первых, необходимо учитывать, что заказы могут быть очень разными и иметь различные параметры, например, тип рекламы, стоимость, дата заказа и т.д. Во-вторых, необходимо учитывать, что заказчиком может являться компания и в базе данных должна быть возможность хранения информации о её представителе.

Для успешной автоматизации работы с заказами необходимо создание базы данных, которая позволит хранить и обрабатывать информацию обо всех заказах и клиентах. База данных должна быть спроектирована таким образом, чтобы обеспечить быстрый доступ к информации и возможность изменения данных в соответствии с потребностями компании.

## **Проектирование концептуальной модели**

По описанным выше анализу и описанию предметной области разрабатываемой информационной системы представляется возможным спроектировать концептуальную модель разрабатываемой системы, состоящую из следующих элементов:

**Сущность «Employee» (Сотрудник):**

Свойства:

* id (Уникальный идентификатор сотрудника)
* login (Логин сотрудника в системе)
* name (Имя сотрудника)
* phone (Номер телефона сотрудника)

**Сущность «Client» (Клиент)**

Свойства:

* id (Уникальный идентификатор клиента)
* organization\_name (Название организации)
* surname\_representative (Фамилия представителя)
* name\_representative (Имя представителя)
* patronymic\_representative (Отчество представителя)
* phone (Номер телефона представителя)
* email (Эл. почта организации)
* description (Описание клиента)

**Сущность «Ordering» (Заказ)**

Свойства:

* id (Уникальный идентификатор заказа)
* id\_client (Уникальный идентификатор клиента, совершающего заказ)
* id\_emloyee (Уникальный идентификатор сотрудника, оформляющего заказ)
* order\_date (Дата заказа)
* id\_advertising\_type (Уникальный идентификатор типа рекламы в данном заказе)
* cost (Стоимость заказа)

**Cущность «OrderDetails» (Детали заказа)**

Свойства:

* id (Уникальный идентификатор деталей для заказа)
* id\_order (Уникальный идентификатор заказа, для которого представлены данные детали)
* id\_service (Уникальный идентификатор услуги, дополняющей заказ)
* status (Статус выполнения заказа)

**Сущность «Advertising\_type» (Тип рекламы) – справочная таблица**

Свойства:

* id (Уникальный идентификатор типа рекламы)
* name (Название типа рекламы)
* min\_price (Минимальная цена за соответствующий тип рекламы)

**Сущность «Service» (Услуга) - справочная таблица**

Свойства:

* id (Уникальный идентификатор услуги)
* name (Название услуги)

## **Построение логической модели, схемы базы данных**

Построенная по концептуальной модели схема базы данных («ER-диаграмма») имеет следующий вид:

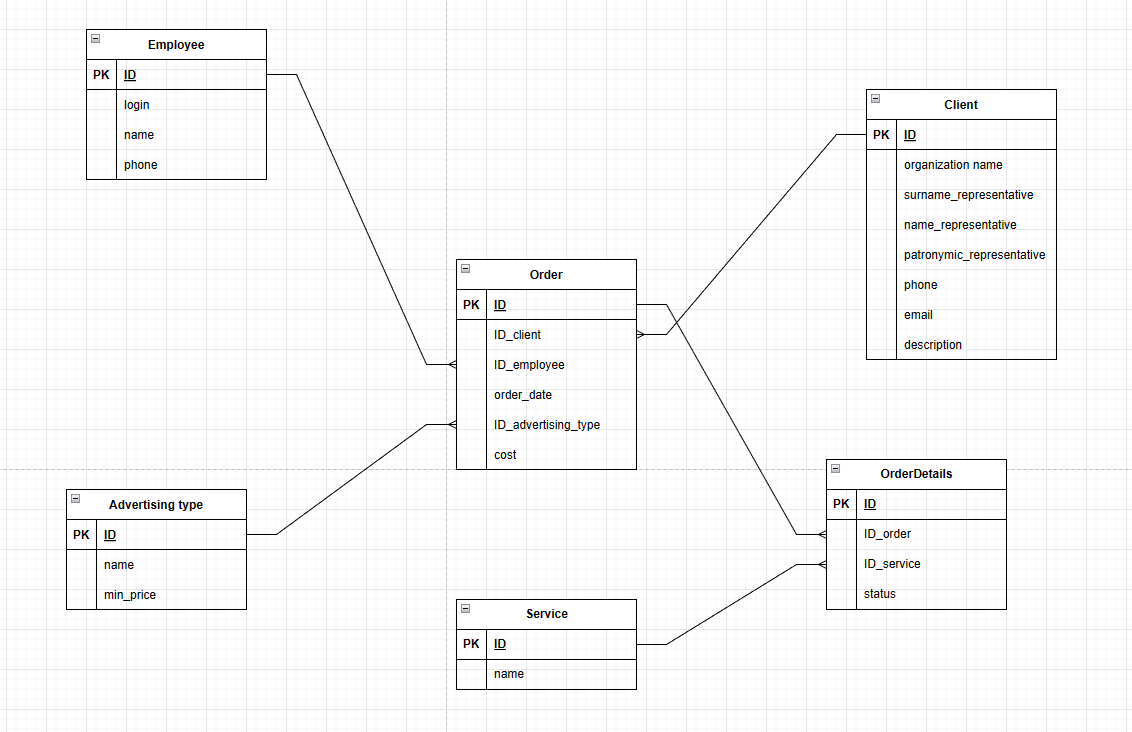


Рисунок 1 - ER-диаграмма

## **Выбор СУБД**

PostgreSQL — это открытая реляционная СУБД, которая имеет ряд преимуществ перед другими СУБД, такими как MySQL или SQL Server:

1. Масштабируемость: PostgreSQL позволяет масштабировать базу данных на тысячи и даже миллионы строк, обеспечивая при этом высокую производительность.
2. Поддержка различных типов данных: PostgreSQL имеет широкий спектр типов данных, включая диапазоны, массивы, геоданные и JSON-объекты, что позволяет хранить и обрабатывать данные различных типов.
3. Открытый исходный код: PostgreSQL имеет открытый исходный код, что означает, что он бесплатно доступен для загрузки и использования.
4. Высокая надежность: PostgreSQL обеспечивает высокую надежность и защиту данных, включая возможность резервного копирования и восстановления, репликацию и транзакции.
5. Многопоточность: PostgreSQL поддерживает множество процессов и потоков одновременно, что повышает производительность при обработке запросов.
6. Поддержка расширений: PostgreSQL предоставляет возможность создавать свои собственные функции, типы данных и операторы, что позволяет расширять функциональность базы данных.

В целом, PostgreSQL является надежной и мощной СУБД для многих задач, включая те, которые требуют высокой производительности и гибкости. Он также имеет большое сообщество пользователей и разработчиков, которые активно вносят свой вклад в его развитие и поддержку.

# **Реализация базы данных**

## **Физическое проектирование**

Согласно логической модели, физическое проектирование выглядит так:

**Сущность «Employee» (Сотрудник):**

Свойства:

* id: SERIAL PRIMARY KEY
* login: VARCHAR (50) NOT NULL
* name: VARCHAR (100) NOT NULL
* phone: VARCHAR (20) NOT NULL

**Сущность «Client» (Клиент)**

Свойства:

* id: SERIAL PRIMARY KEY,
* organization\_name: VARCHAR(100) NOT NULL,
* surname\_representative: VARCHAR(50) NOT NULL,
* name\_representative: VARCHAR(50) NOT NULL,
* patronymic\_representative: VARCHAR(50),
* phone: VARCHAR(20) NOT NULL,
* email: VARCHAR(100) NOT NULL,
* description: VARCHAR(500)

**Сущность «Ordering» (Заказ)**

Свойства:

* id: SERIAL PRIMARY KEY
* id\_client: INT NOT NULL FOREIGN KEY client (id)
* id\_emloyee: INT NOT NULL FOREIGN KEY emloyee (id)
* order\_date: DATE NOT NULL
* id\_advertising\_type: INT NOT NULL FOREIGN KEY advertising\_type (id)
* cost: MONEY NOT NULL

**Cущность «OrderDetails» (Детали заказа)**

Свойства:

* id: SERIAL PRIMARY KEY
* id\_order: INT NOT NULL FOREIGN KEY order (id)
* id\_service: INT NOT NULL FOREIGN KEY service (id)
* status: BOOLEAN NOT NULL

**Сущность «Advertising\_type» (Тип рекламы) – справочная таблица**

Свойства:

* id: SERIAL PRIMARY KEY
* name: VARCHAR (100) NOT NULL
* min\_price: MONEY NOT NULL

**Сущность «Service» (Услуга) - справочная таблица**

Свойства:

* id: SERIAL PRIMARY KEY
* name: VARCHAR (100) NOT NULL

## **Написание исходного кода БД**

Исходный код создания описанной БД:

CREATE DATABASE Advertising\_agency;

CREATE TABLE Employee (

ID SERIAL PRIMARY KEY,

login VARCHAR(50) NOT NULL,

name VARCHAR(100) NOT NULL,

phone VARCHAR(20) NOT NULL

);

CREATE TABLE Client (

ID SERIAL PRIMARY KEY,

organization\_name VARCHAR(100) NOT NULL,

surname\_representative VARCHAR(50) NOT NULL,

name\_representative VARCHAR(50) NOT NULL,

patronymic\_representative VARCHAR(50),

phone VARCHAR(20) NOT NULL,

email VARCHAR(100) NOT NULL,

description VARCHAR(500)

);

CREATE TABLE Advertising\_type (

ID SERIAL PRIMARY KEY,

name VARCHAR(100) NOT NULL,

min\_price MONEY NOT NULL

);

CREATE TABLE Service (

ID SERIAL PRIMARY KEY,

name VARCHAR(100) NOT NULL

);

CREATE TABLE OrderDetails (

ID SERIAL PRIMARY KEY,

ID\_order INT NOT NULL,

ID\_service INT NOT NULL,

status BOOLEAN NOT NULL

);

CREATE TABLE Ordering (

ID SERIAL PRIMARY KEY,

ID\_Client INT NOT NULL,

ID\_Employee INT NOT NULL,

order\_date DATE NOT NULL,

ID\_Advertising\_type INT NOT NULL,

cost MONEY NOT NULL

);

ALTER TABLE ordering ADD CONSTRAINT fk\_ordering\_employee

FOREIGN KEY (ID\_employee)

REFERENCES employee(ID);

ALTER TABLE ordering ADD CONSTRAINT fk\_ordering\_advertising\_type

FOREIGN KEY (ID\_advertising\_type)

REFERENCES Advertising\_type(ID);

ALTER TABLE OrderDetails ADD CONSTRAINT fk\_orderdetails\_ordering

FOREIGN KEY (ID\_order)

REFERENCES Ordering(ID);

ALTER TABLE OrderDetails ADD CONSTRAINT fk\_orderdetails\_service

FOREIGN KEY (ID\_service)

REFERENCES Service(ID);

ALTER TABLE ordering ADD CONSTRAINT fk\_ordering\_client

FOREIGN KEY (ID\_client)

REFERENCES CLient(ID);

## **Написание запросов, функций и хранимых процедур**

**Вывести список всех заказов, которые были сделаны в определенный период времени (например, за последний месяц)**

SELECT \* FROM "Order" WHERE order\_date BETWEEN '2023-05-17' AND '2023-06-17';

**Вывести список всех клиентов, которые сделали заказы на определенную услугу.**

SELECT "Client".\*, "OrderDetails".\* FROM "Client"

JOIN "Order" ON "Client".id = "Order".id\_client

JOIN "OrderDetails" ON "Order".id = "OrderDetails".id\_order

WHERE "OrderDetails".id\_service = 1;

**Вывести список всех заказов, которые были сделаны определенным сотрудником**

SELECT \* FROM "Order" WHERE id\_employee = 1;

**Вывести список всех заказов, которые были сделаны на определенный тип рекламы**

SELECT \* FROM "Order" WHERE id\_advertising\_type = 2;

**Вывести список всех клиентов, которые не сделали ни одного заказа**

SELECT \* FROM "Client" WHERE id NOT IN (SELECT DISTINCT id\_client FROM "Order");

**Вывести список всех заказов, которые были сделаны на услуги, стоимость которых выше средней стоимости всех услуг.**

SELECT "Order".\* FROM "Order"

JOIN "OrderDetails" ON "Order".id = "OrderDetails".id\_order

JOIN "Service" ON "OrderDetails".id\_service = "Service".id

WHERE "Service".cost > (SELECT AVG(cost) FROM "Service");

**Вывести список всех клиентов, которые сделали заказы на услуги, стоимость которых выше средней стоимости всех услуг.**

SELECT DISTINCT "Client".\* FROM "Client"

JOIN "Order" ON "Client".id = "Order".id\_client

JOIN "OrderDetails" ON "Order".id = "OrderDetails".id\_order

JOIN "Service" ON "OrderDetails".id\_service = "Service".id

WHERE "Service".cost > (SELECT AVG(cost) FROM "Service");

**Вывести список всех заказов, которые были сделаны на услуги, название которых начинается на определенную букву.**

SELECT "Order".\* FROM "Order"

JOIN "OrderDetails" ON "Order".id = "OrderDetails".id\_order

JOIN "Service" ON "OrderDetails".id\_service = "Service".id

WHERE "Service".name LIKE 'A%';

**Вывести список всех клиентов, которые сделали заказы на услуги, название которых начинается на определенную букву.**

SELECT DISTINCT "Client".\* FROM "Client"

JOIN "Order" ON "Client".id = "Order".id\_client

JOIN "OrderDetails" ON "Order".id = "OrderDetails".id\_order

JOIN "Service" ON "OrderDetails".id\_service = "Service".id

WHERE "Service".name LIKE 'A%';

**Вывести список всех заказов, которые были сделаны на услуги, стоимость которых выше средней стоимости услуг определенного типа.**

SELECT "Order".\* FROM "Order"

JOIN "OrderDetails" ON "Order".id = "OrderDetails".id\_order

JOIN "Service" ON "OrderDetails".id\_service = "Service".id

WHERE "Service".id = 1 AND "Service".cost > (SELECT AVG(cost) FROM "Service" WHERE id = 1);

**Заключение**

В заключении данной курсовой работы можно отметить, что автоматизация работы с заказами для рекламного агентства является важным и актуальным вопросом в современном мире. Создание базы данных, которая позволяет хранить и обрабатывать информацию обо всех заказах и клиентах, может значительно ускорить и упростить процесс работы рекламного агентства.

В данной курсовой работе была предложена концептуальная модель базы данных, которая учитывает все особенности предметной области и обеспечивает быстрый доступ к информации. Также были предложены SQL-скрипты для создания таблиц и связей между ними, что позволит создать работающую базу данных.

В процессе выполнения курсовой работы были рассмотрены основные принципы проектирования баз данных, а также основные концепции SQL. Работа над проектом помогла углубить знания в области баз данных и приобрести практические навыки работы с SQL.

Таким образом, выполнение данной курсовой работы позволило получить ценный опыт в проектировании, создании и использовании баз данных, что может быть полезным в будущей профессиональной деятельности в области информационных технологий.

**Список использованных источников**

[PostgreSQL | Введение (metanit.com)](https://metanit.com/sql/postgresql/1.1.php)

**Приложение**

База данных (БД) — это организованное в единую структуру хранилище данных, которое обеспечивает удобный, быстрый и надёжный доступ к хранимой информации.

Реляционная база данных – база данных представленная в виде таблиц

Система Управления Базами Данных (СУБД) – программный комплекс по управлению хранением, созданием, удалением, изменением и т.д. баз данных