

Министерство образования Республики Беларусь
Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ
Факультет компьютерных систем и сетей
Кафедра электронных вычислительных машин
Дисциплина: Базы данных

Тема «Столовая Лидо»
Лабораторная работа №2
Создание реляционной схемы данных

Студент:
Преподаватель:

М.С. Патюпин
Д.В. Куприянова

МИНСК 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 СОЗДАНИЕ РЕАЛИЦИОННОЙ СХЕМЫ БАЗЫ ДАННЫХ	4
1.1 Преобразование ER–объекта в реляционный объект	4
1.2 Отображение связи «один-ко-многим»	6
1.3 Отображение связи «многие-ко-многим»	8
1.4 Реляционная схема базы данных.....	9
2 СОЗДАНИЕ РЕАЛЯЦИОННОЙ МОДЕЛИ В pgAdmin	10
2.1 Создание базы данных в pgAdmin	10
2.2 Создание реляционной модели средствами pgAdmin.....	13
ЗАКЛЮЧЕНИЕ	17
ПРИЛОЖЕНИЕ А	18

ВВЕДЕНИЕ

Перевод ER-модели в реляционную модель выполняется с помощью алгоритма, состоящего из шести шагов:

Шаг 1. Каждый объект на ER-диаграмме превращается в реляционное отношение (далее для краткости – таблицу), имя объекта становится именем таблицы (следует указать понятное имя).

Шаг 2. Каждый атрибут объекта становится столбцом таблицы с тем же именем (также следует указать понятное имя) и требуемым типом данных.

Шаг 3. Уникальные (ключевые) атрибуты объекта превращаются в первичный ключ таблицы (при наличии нескольких возможных уникальных идентификаторов, выбирается наиболее подходящий для использования; если таковых атрибутов нет или они плохо подходят для долговременного использования в БД, то желательно создать суррогатный ключ). Каждая таблица в БД должна иметь первичный ключ!

Шаг 4. Связи «один-ко-многим» (в том числе и связи «один-к-одному») становятся ссылками в уже существующих таблицах, при этом внешний ключ добавляется в виде столбца (столбцов) в таблицу, соответствующую объекту со стороны «многие» связи. Внешние ключи должны ссылаться только на первичные ключи целевых таблиц!

Шаг 5. Связи «многие-ко-многим» реализуются каждая через отдельную промежуточную таблицу:

- эта промежуточная таблица обязательно будет содержать столбцы внешних ключей, ссылающиеся на соответствующие объекты связи;

- первичный ключ промежуточной таблицы для исключения дубликатов должен быть составным и включать в себя все внешние ключи на объекты, участвующие в связи.

Шаг 6. Если связь имеет дополнительные атрибуты, то, как и в случае атрибутов объектов, они становятся столбцом соответствующей таблицы:

- для связей «один-ко-многим» (встречаются на практике редко) – в таблице со стороны «многие» (там, где расположен внешний ключ);

- для связей «многие-ко-многим» – в промежуточной таблице (при этом атрибуты, расширяющие комбинацию в связи (например, «дата»), также должны войти в состав составного первичного ключа промежуточной таблицы).

1 СОЗДАНИЕ РЕАЛИЦИОННОЙ СХЕМЫ БАЗЫ ДАННЫХ

На рисунке 1.1 предоставлена ER–диаграмма из лабораторной работы №1.

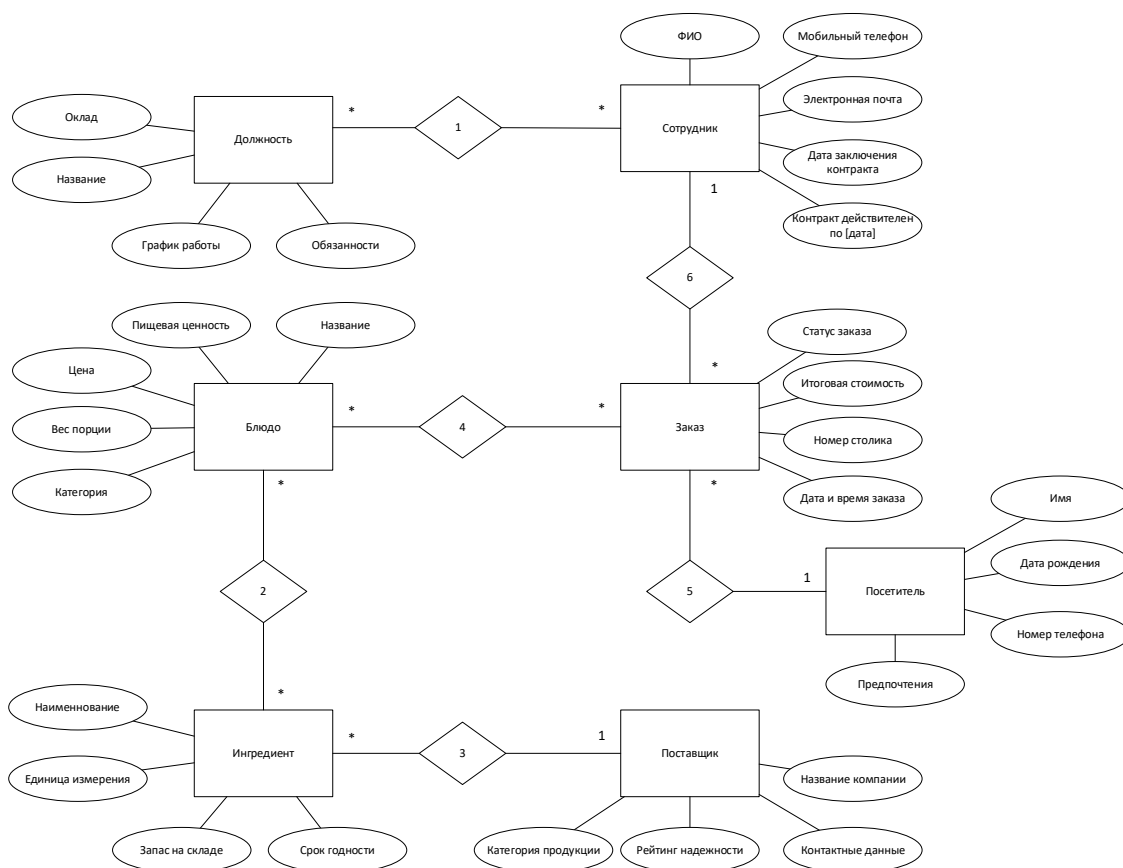


Рисунок 1.1 – ER–диаграмма

1.1 Преобразование ER–объекта в реляционный объект

Выполнение 1–3 шагов алгоритма предоставлено на рисунках 1.2 – 1.8, где слева изображен ER–объект, справа реляционный объект.

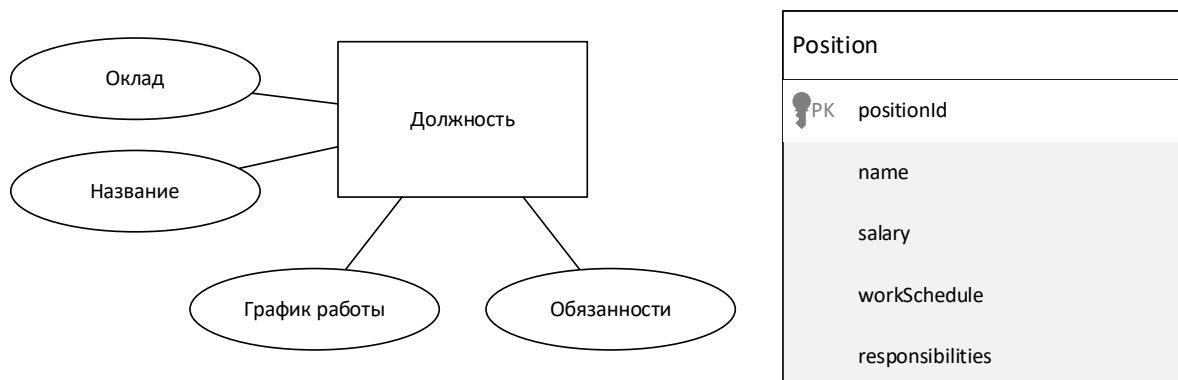
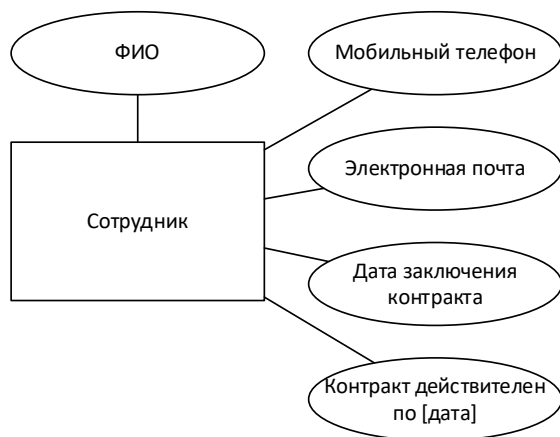
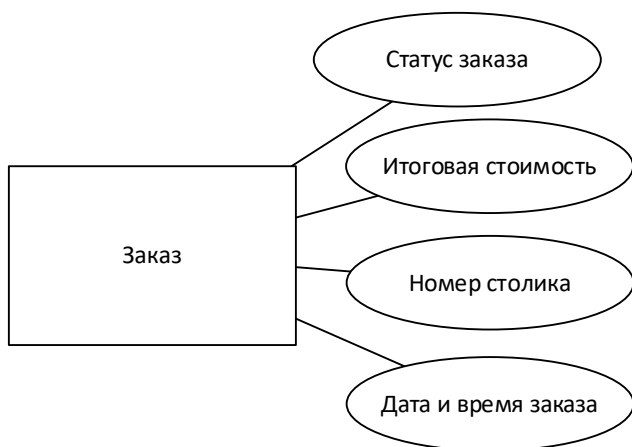


Рисунок 1.2 – ER–объект, реляционный объект «Должность»



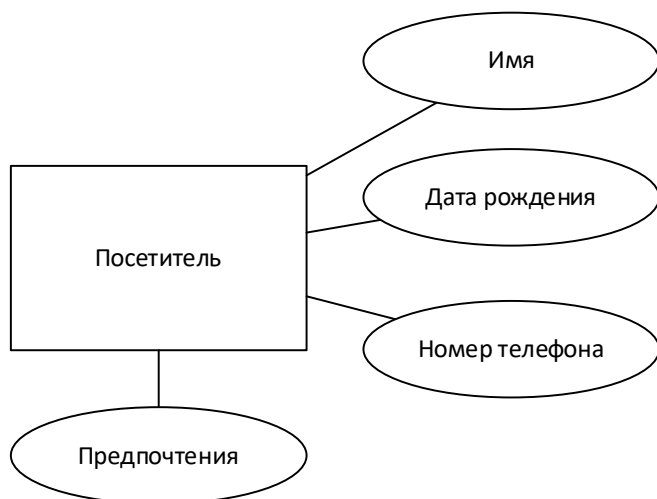
Employee	
PK	employeeId
fullName	
phoneNumber	
hireDate	
contractEndDate	

Рисунок 1.3 – ER–объект, реляционный объект «Сотрудник»



Order	
PK	orderId
status	
totalCost	
tableNumber	
orderDateTime	

Рисунок 1.4 – ER–объект, реляционный объект «Заказ»



Visitor	
PK	visitorId
name	
birthDate	
phoneNumber	
preferences	

Рисунок 1.5 – ER–объект, реляционный объект «Посетитель»

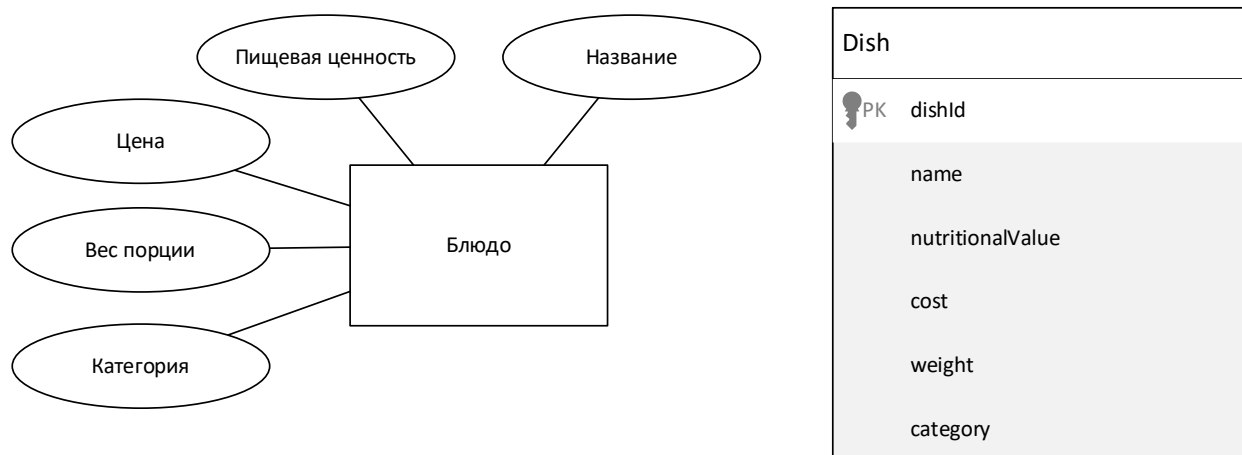


Рисунок 1.6 – ER–объект, реляционный объект «Блюдо»

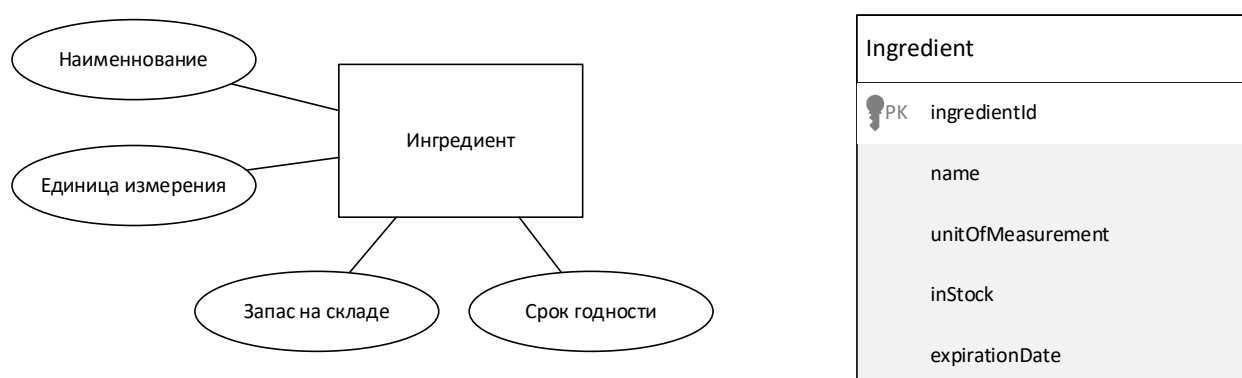


Рисунок 1.7 – ER–объект, реляционный объект «Ингредиент»



Рисунок 1.8 – ER–объект, реляционный объект «Поставщик»

1.2 Отображение связи «один-ко-многим»

Выполнение 4 шага алгоритма предоставлено на рисунках 1.9 – 1.11. Соотношение реляционных объектов к ER–объектам приведены на рисунках 1.2–1.8.

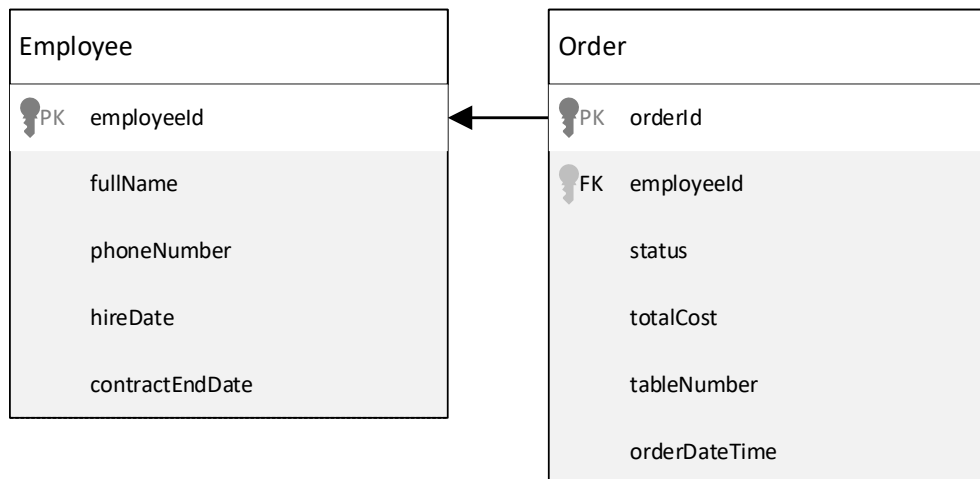


Рисунок 1.9 – Отображение связи «Сотрудник – Заказ»

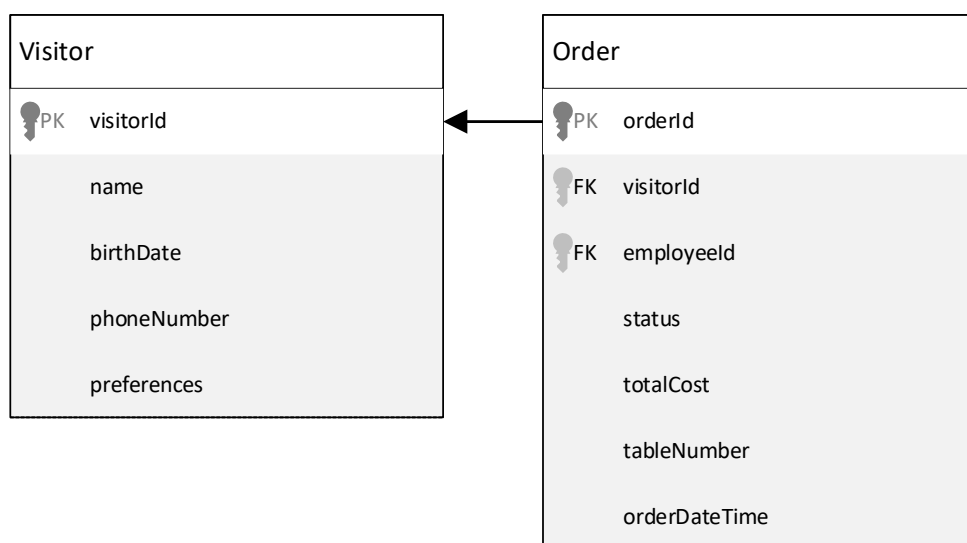


Рисунок 1.10 – Отображение связи «Посетитель – Заказ»

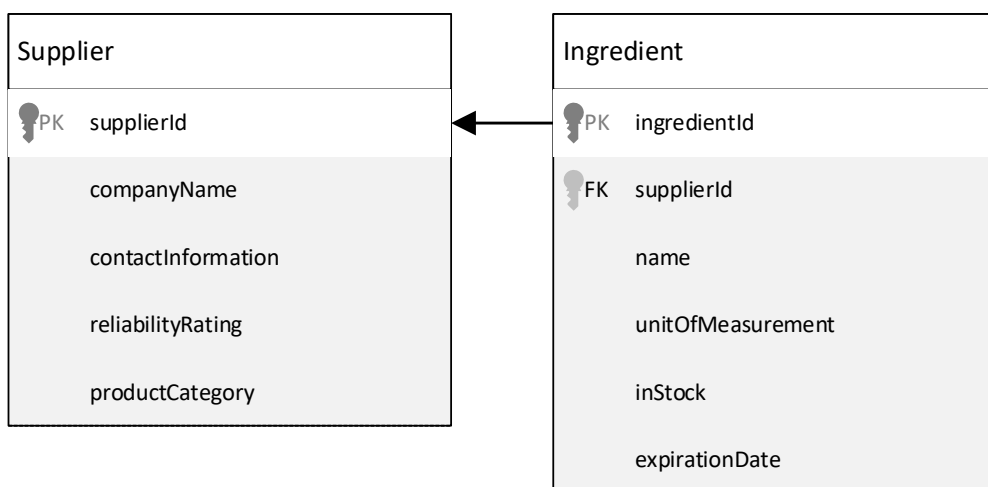


Рисунок 1.11 – Отображение связи «Поставщик – Ингредиент»

1.3 Отображение связи «многие-ко-многим»

Выполнение 5 шага алгоритма предоставлено на рисунках 1.12 – 1.13, где отображены связи «многие-ко-многим» через отдельную таблицу. Соотношение реляционных объектов к ER-объектам приведены на рисунках 1.2–1.8.

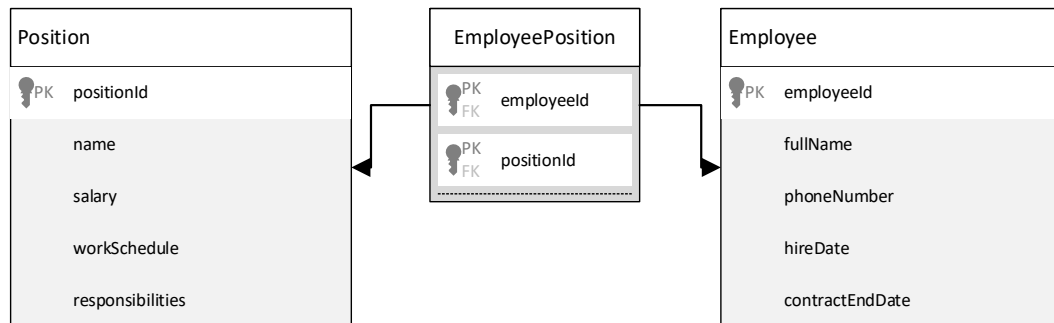


Рисунок 1.12 – Отображение связи «Должность – Сотрудник»

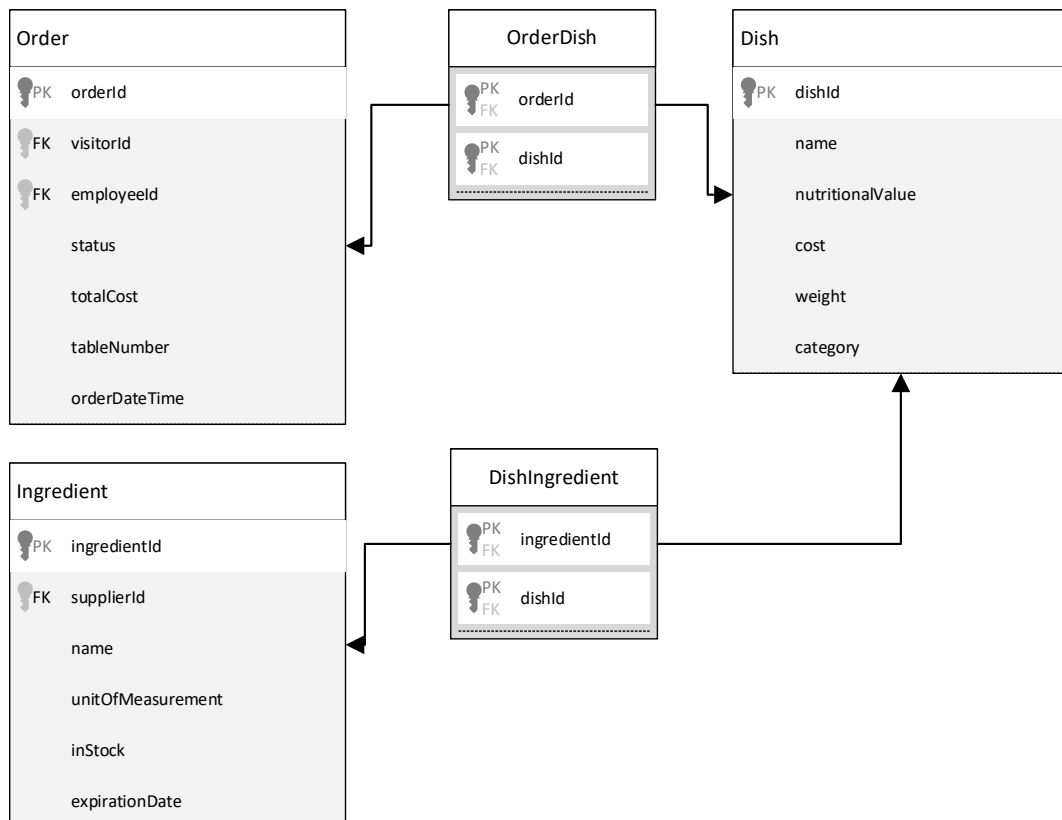


Рисунок 1.13 – Отображение связей «Заказ – Блюдо»,
«Ингредиент – Блюдо»

Шаг 6 алгоритма пропускается в связи с отсутствием дополнительных атрибутов у связей.

1.4 Реляционная схема базы данных

Реляционная схема базы данных «Столовая Лидо» приведена на рисунке 1.14. Соотношение реляционных объектов к ER–объектам приведены на рисунках 1.2–1.8.

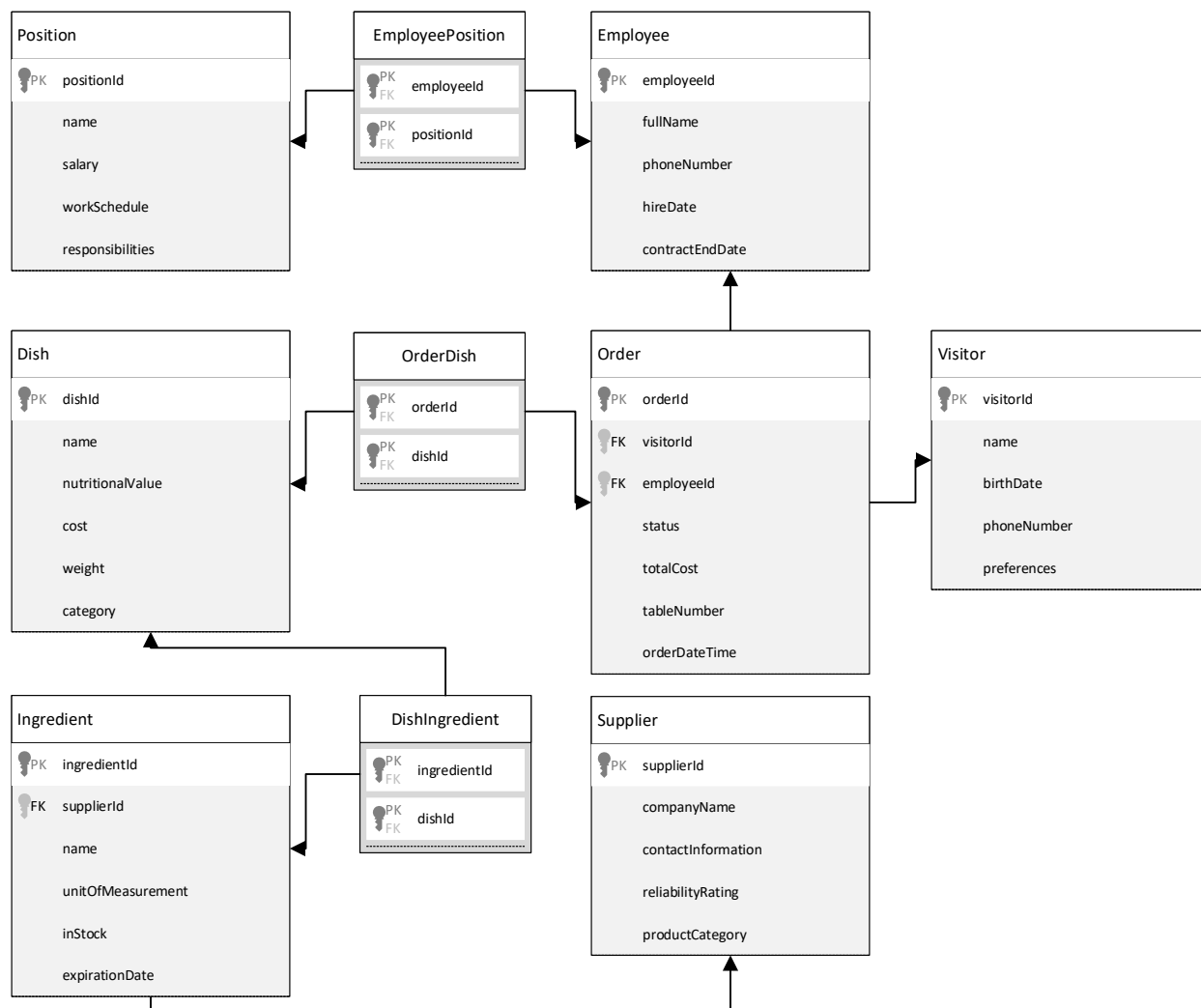


Рисунок 1.14 – Реляционная схема базы данных

2 СОЗДАНИЕ РЕАЛЯЦИОННОЙ МОДЕЛИ В pgAdmin

2.1 Создание базы данных в pgAdmin

Для создания базы данных postgresql с помощью клиента pgAdmin необходимо:

- 1 Запустить клиент pgAdmin, рисунок 2.1.



Рисунок 2.1 – Запуск клиента pgAdmin

- 2 Создать новую базу данных, Object Explorer -> Servers -> PostgreSQL 16 -> Create -> Database, рисунок 2.2.

- 3 Задать имя новой базе данных -> «Lido», рисунок 2.3.

- 4 Открыть окно исполнения команд, Object Explorer -> Servers -> PostgreSQL 16 -> Lido -> Query Tool, рисунок 2.4.

- 5 В появившейся вкладке, написать исходный код программы на языке программирования SQL для создания таблиц базы данных соответствующей схеме приведенной на рисунке 1.14. Выполнить нажатием клавиши F5. Рисунок 2.5. Исходный код программы приведен в приложении А.

- 6 Ожидаем сообщения о создании новых таблиц, рисунок 2.6.

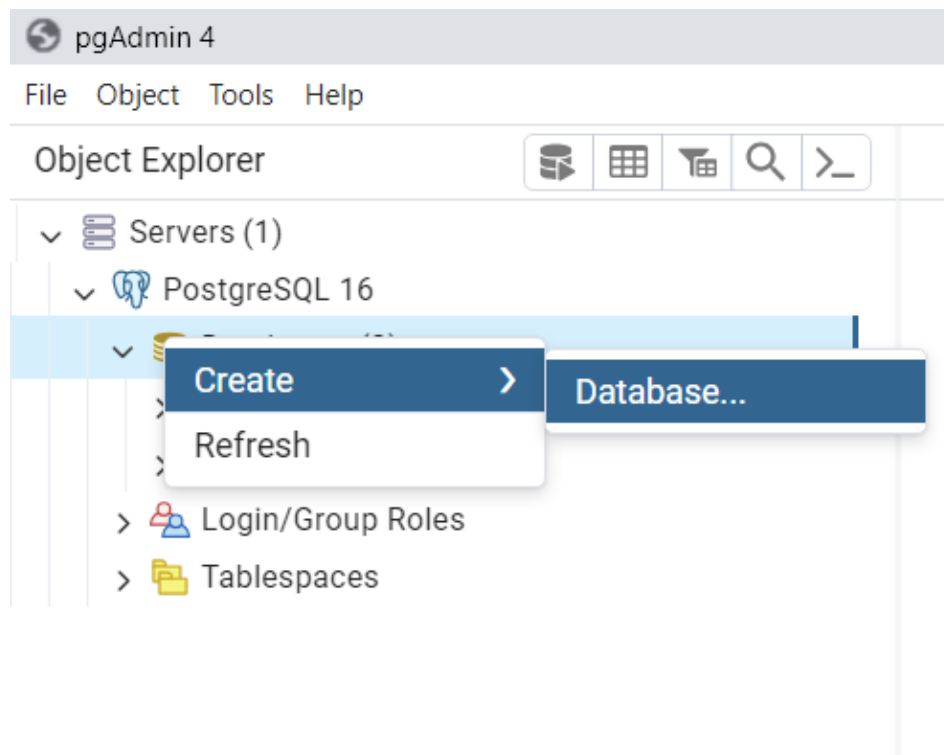


Рисунок 2.2 – Создание новой БД

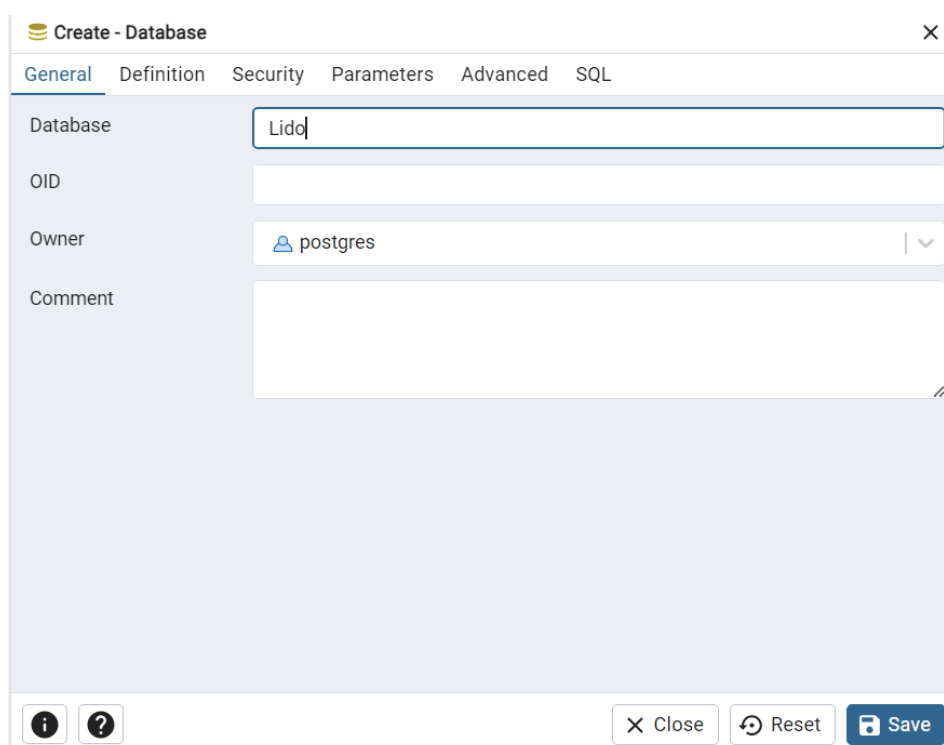


Рисунок 2.3 – Задание имени БД

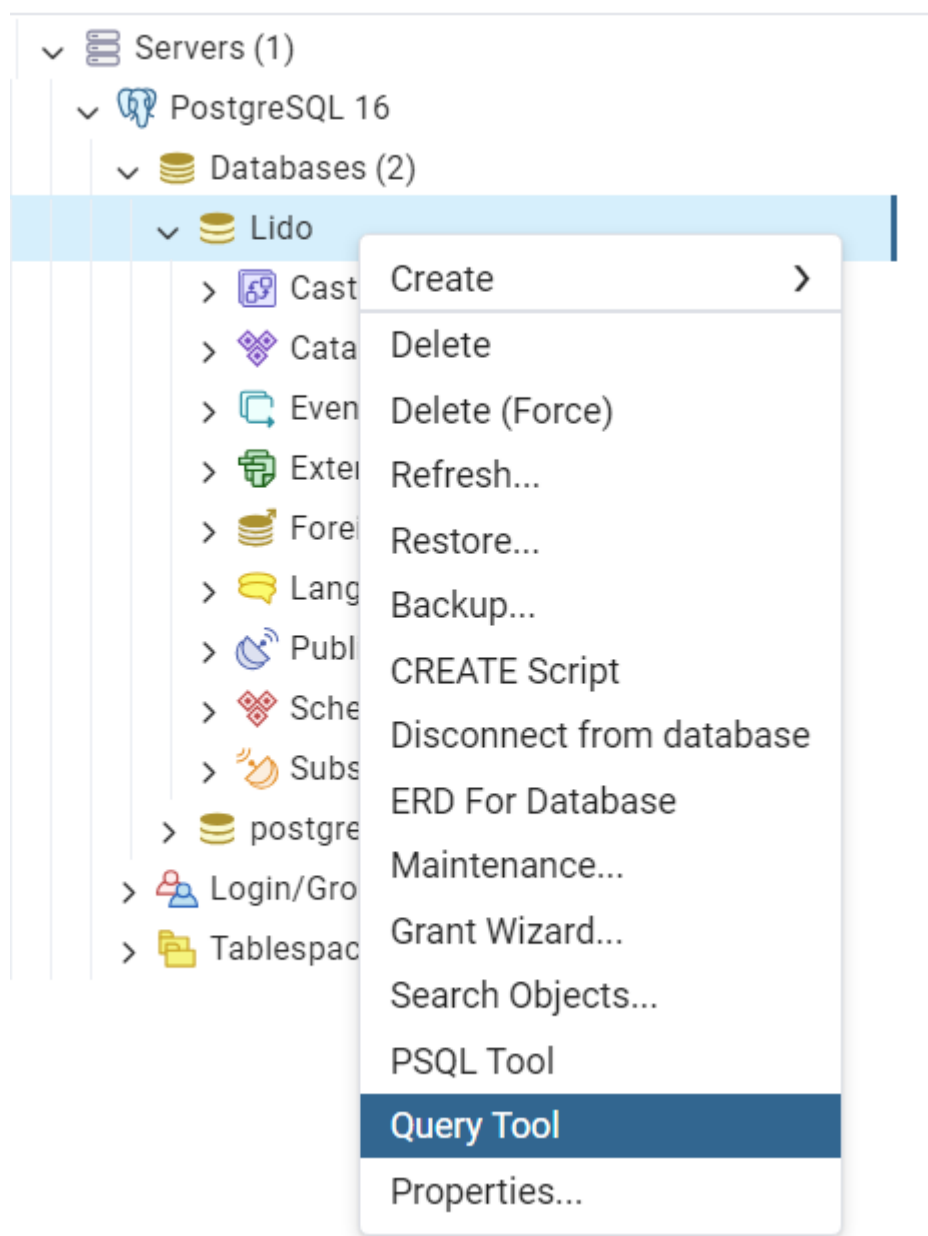


Рисунок 2.4 – Открытие Query Tool

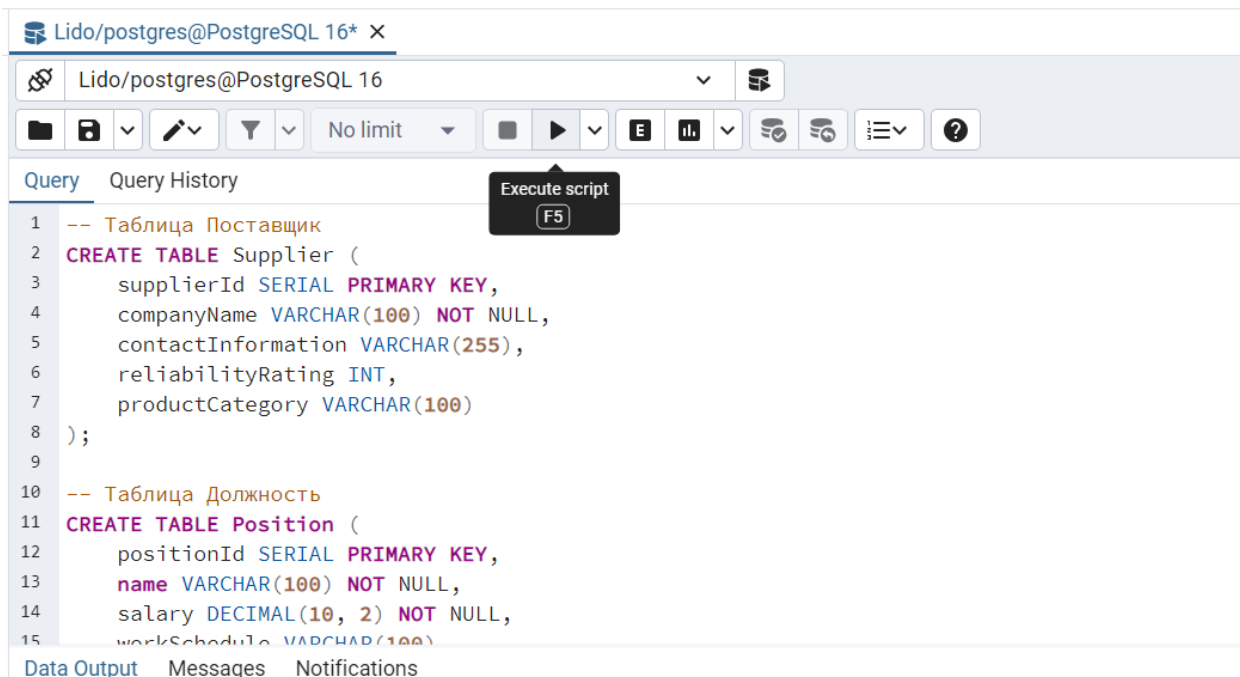


Рисунок 2.5 – Окно написания и для выполнения исходного кода программы

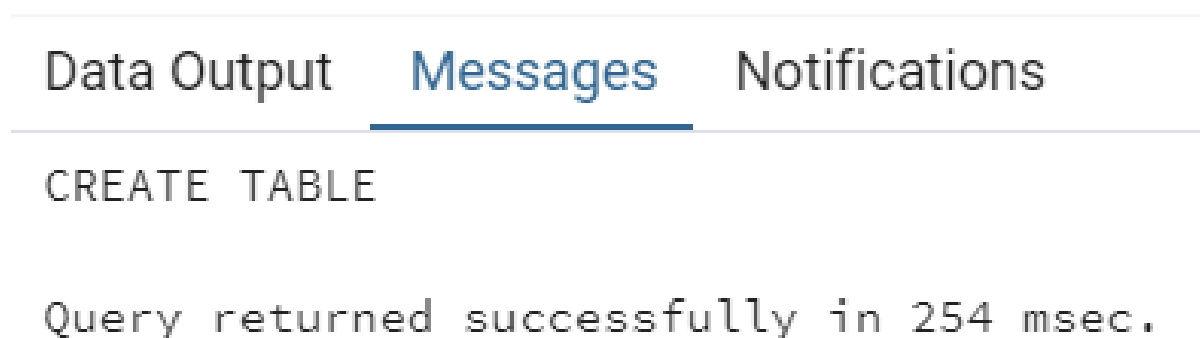


Рисунок 2.6 – Сообщение об успешном создании таблиц

2.2 Создание реляционной модели средствами pgAdmin

Для автоматического создания реляционной модели использовалось свободное программное обеспечение – десктоп клиент БД pgAdmin.

Для создания модели необходимо выполнить Object Explorer → Servers → PostgreSQL 16 → Lido → ERD For Database. – рисунок 2.7.

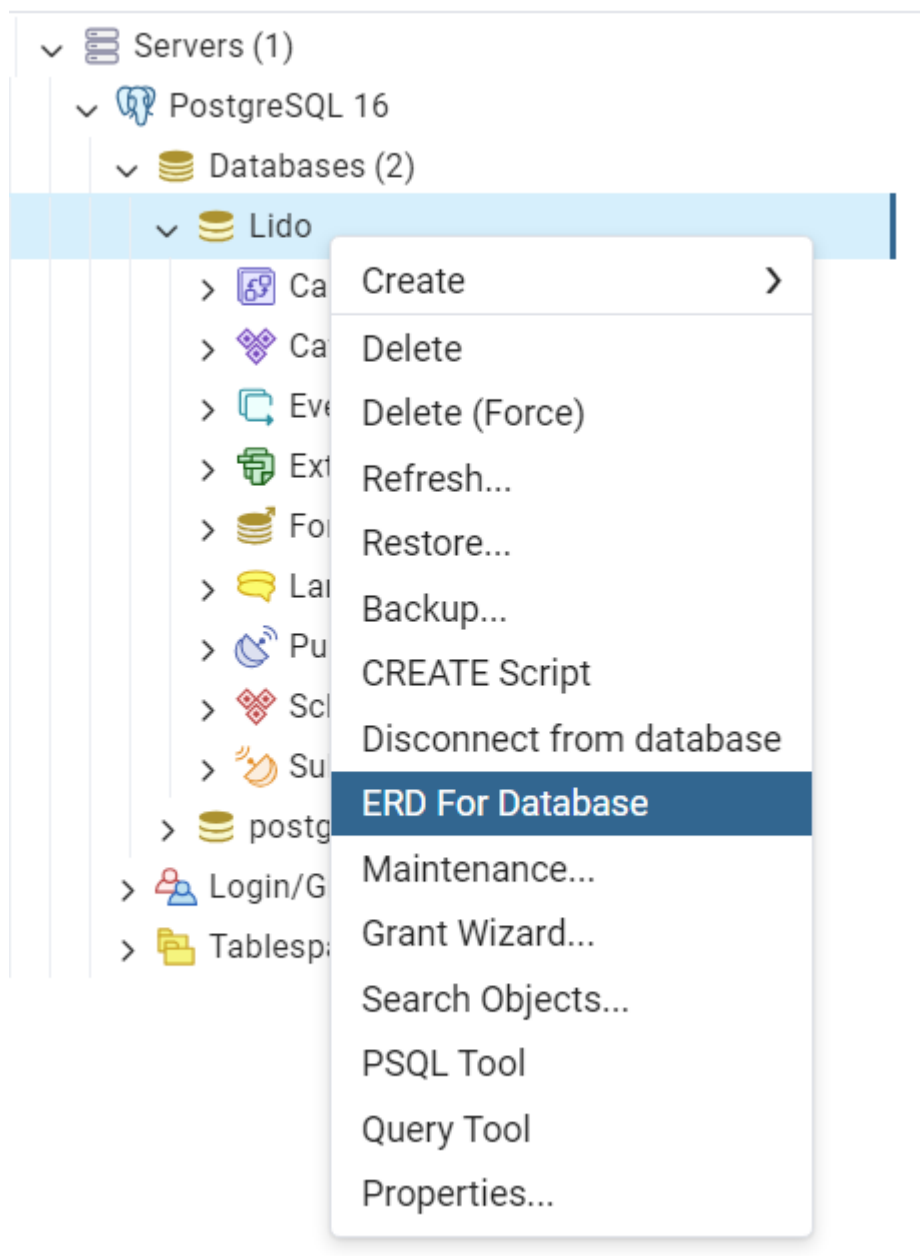


Рисунок 2.7 – Открытие реляционной модели

В открытой модели – рисунок 2.8, есть возможность выбрать предпочтительную нотацию – рисунок 2.9, сохранить как изображение – рисунок 2.10 и перемещать объекты по рабочей поверхности, при этом не обрывая связи между ними. Для наилучшего представления объекты были перемещены – рисунок 2.11.

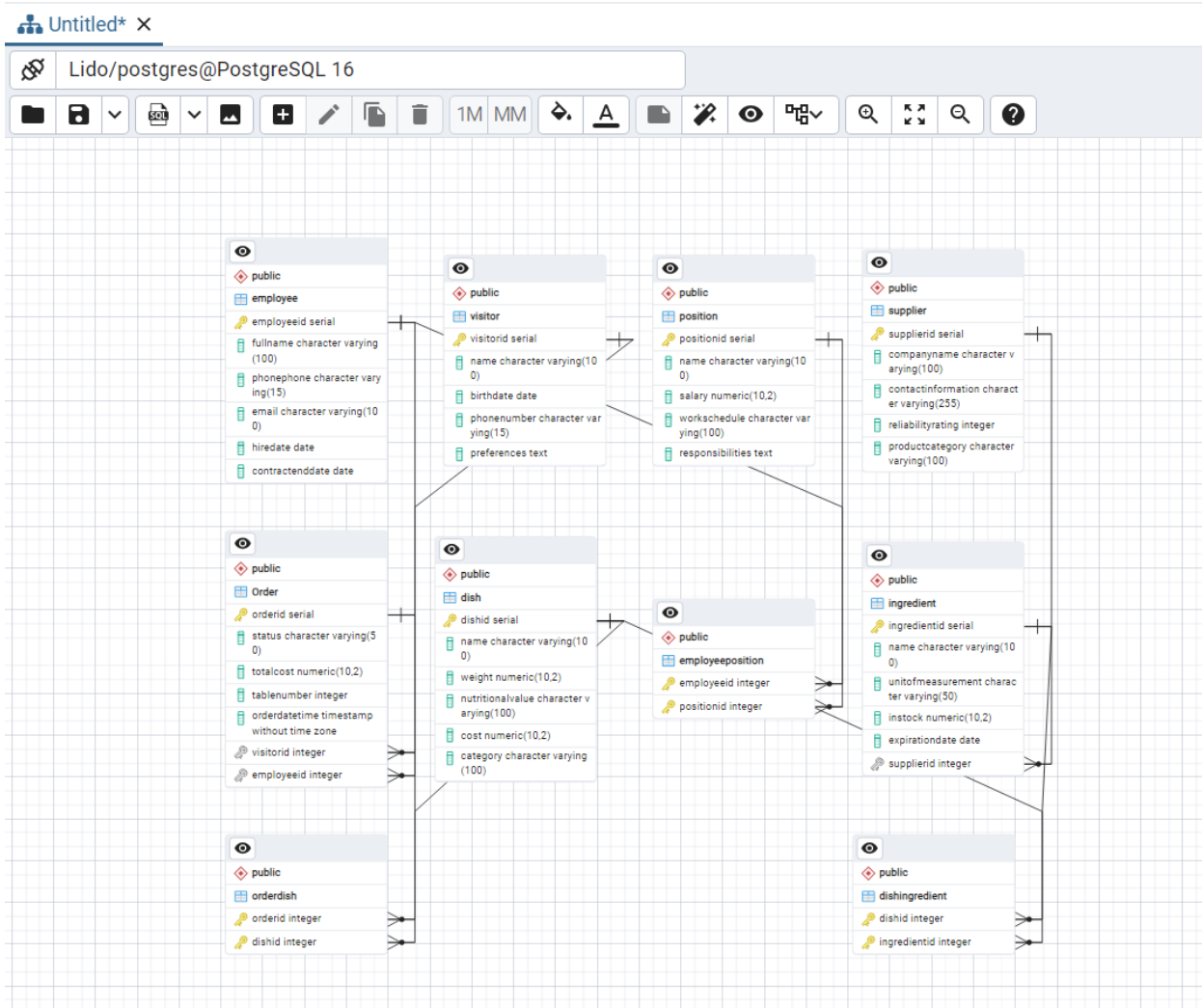


Рисунок 2.8 – Модель БД

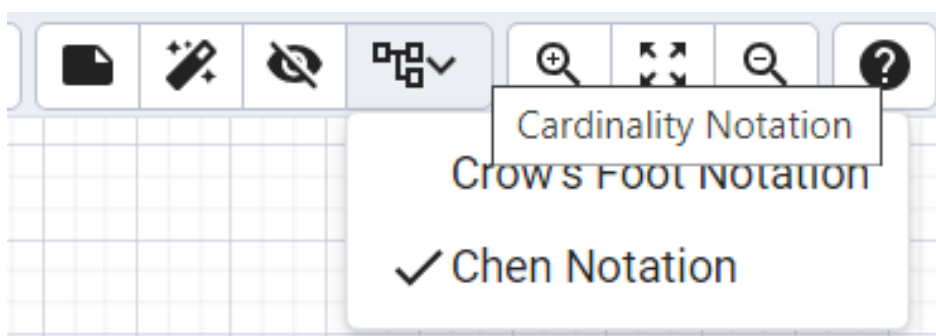


Рисунок 2.9 –Выбор нотации

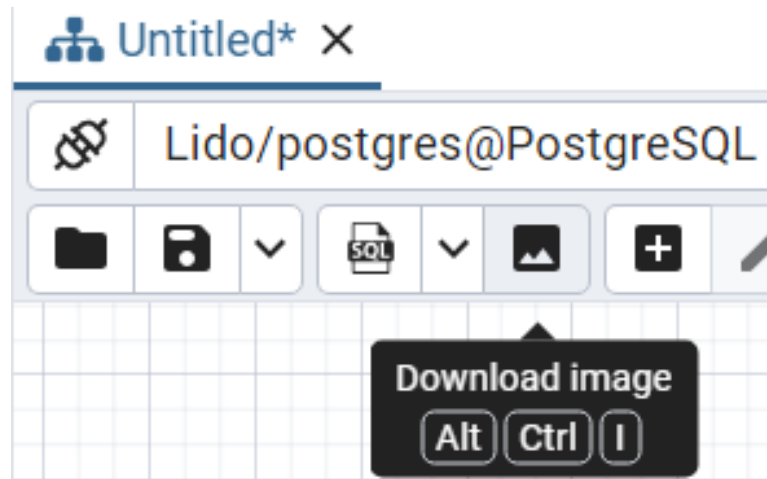


Рисунок 2.10 – Сохранить рабочее поле как изображение



Рисунок 2.11 – Реляционная модель БД

ЗАКЛЮЧЕНИЕ

В результате работы была преобразована ER-диаграмма в реляционную схему двумя способами:

- с помощью алгоритма, приведенного в разделе «Введение»;
- с помощью встроенных инструментов в программном обеспечении pgAdmin.

Для этого были определены и созданы таблицы в postgresql.

ПРИЛОЖЕНИЕ А
(обязательное)
Исходный код программы

```
00 -- Таблица Поставщик
01 CREATE TABLE Supplier (
02     supplierId SERIAL PRIMARY KEY,
03     companyName VARCHAR(100) NOT NULL,
04     contactInformation VARCHAR(255),
05     reliabilityRating INT,
06     productCategory VARCHAR(100)
07 );
08 -- Таблица Должность
09 CREATE TABLE Position (
10     positionId SERIAL PRIMARY KEY,
11     name VARCHAR(100) NOT NULL,
12     salary DECIMAL(10, 2) NOT NULL,
13     workSchedule VARCHAR(100),
14     responsibilities TEXT
15 );
16 -- Таблица Сотрудник
17 CREATE TABLE Employee (
18     employeeId SERIAL PRIMARY KEY,
19     fullName VARCHAR(100) NOT NULL,
20     phonePhone VARCHAR(15),
21     email VARCHAR(100),
22     hireDate DATE,
23     contractEndDate DATE
24 );
25 -- Таблица Блюдо
26 CREATE TABLE Dish (
27     dishId SERIAL PRIMARY KEY,
28     name VARCHAR(100) NOT NULL,
29     weight DECIMAL(10, 2) NOT NULL,
30     nutritionalValue VARCHAR(100),
31     cost NUMERIC(10, 2),
32     category VARCHAR(100)
33 );
34 -- Таблица Ингредиент
35 CREATE TABLE Ingredient (
36     ingredientId SERIAL PRIMARY KEY,
37     name VARCHAR(100) NOT NULL,
38     unitOfMeasurement VARCHAR(50),
39     inStock DECIMAL(10, 2) NOT NULL,
40     expirationDate DATE,
41     supplierId INT,
42     FOREIGN KEY (supplierId) REFERENCES Supplier(supplierId)
43 );
44 -- Таблица Посетитель
45 CREATE TABLE Visitor (
46     visitorId SERIAL PRIMARY KEY,
```

```

47     name VARCHAR(100) NOT NULL,
48     birthDate DATE,
49     phoneNumber VARCHAR(15),
50     preferences TEXT
51 );
52 -- Таблица Заказ
53 CREATE TABLE "Order" (
54     orderId SERIAL PRIMARY KEY,
55     status VARCHAR(50),
56     totalCost DECIMAL(10, 2) NOT NULL,
57     tableNumber INT,
58     orderDateTime TIMESTAMP NOT NULL,
59     visitorId INT,
60     employeeId INT,
61     FOREIGN KEY (visitorId) REFERENCES Visitor(visitorId),
62     FOREIGN KEY (employeeId) REFERENCES Employee(employeeId)
63 );
64 -- Таблица Должность-Сотрудник (связь многие-ко-многим)
65 CREATE TABLE EmployeePosition (
66     employeeId INT,
67     positionId INT,
68     PRIMARY KEY (employeeId, positionId),
69     FOREIGN KEY (employeeId) REFERENCES Employee(employeeId),
70     FOREIGN KEY (positionId) REFERENCES Position(positionId)
71 );
72 -- Таблица Блюдо-Ингредиент (связь многие-ко-многим)
73 CREATE TABLE DishIngredient (
74     dishId INT,
75     ingredientId INT,
76     PRIMARY KEY (dishId, ingredientId),
77     FOREIGN KEY (dishId) REFERENCES Dish(dishId),
78     FOREIGN KEY (ingredientId) REFERENCES
79     Ingredient(ingredientId));
80 -- Таблица Блюдо-Заказ (связь многие-ко-многим)
81 CREATE TABLE OrderDish (
82     orderId INT,
83     dishId INT,
84     PRIMARY KEY (orderId, dishId),
85     FOREIGN KEY (orderId) REFERENCES "Order"(orderId),
86     FOREIGN KEY (dishId) REFERENCES Dish(dishId)
87 );

```