**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Отчет о выполнении лабораторной работы №6**

**по дисциплине**

**«Технологии проектирования программного обеспечения»**

Направление подготовки 01.03.02 Прикладная математика и информатика

Направленность Информационные технологии

Работу выполнил студент группы 4ИТ/2 И.А. Молчанов

(подпись)

Работу проверил доц. каф. ИТ, к.т.н., доц. А.Н. Полетайкин

(подпись)

Краснодар

2023

### ВВЕДЕНИЕ

### Тема: Проектирование базы данных программной системы.

### Цель: изучение программных средств для разработки моделей информационной базы ПС, проработка методов нормализации отношений в БД, приобретение навыков применения CASE-средства ERwin для моделирования базы данных ПС.

### Задание:

1. На основе модели классов UML, разработанной при выполнении лабораторной работы №4.1, произвести идентификацию сущностей информационной базы ПС и связей между ними:

* изучить массив нормативно-справочной (условно-постоянной) информации, определить состав соответствующих справочников.
* изучить массив входной (текущей) информации, структурировать его по ключевым сущностям с указанием всех атрибутов.

Как правило, документ с обычным "бумажными" таблицами разбивается по принципу: одна "бумажная" таблица – одна сущность. Соответственно, каждому входному документу, выделенному при выполнении лабораторной работы №1, поставить в соответствие одну сущность или (в результате нормализации) структуру связанных сущностей. Количество сущностей в модели зависит от предметной области.

1. При помощи CASE-средства ERWin разработать ER-диаграмму логической модели данных.
2. Выполнить сравнительный анализ полученной логической модели с моделью классов, разработанной при выполнении лабораторной работы №4.1. При необходимости скорректировать диаграмму классов.
3. Провести нормализацию сущностей логической модели данных и разработать ER-диаграмму физической модели данных. Имена, атрибуты и назначение сущностей физической модели данных привести в формате табл. 8. Сущности разделить на оперативные и справочные. При разработке модели определить сущности, их первичные и *внешние* ключи и атрибуты, а также связи между сущностями. Цель физического моделирования – это таблицы в нормальных формах высшего, минимум, третьего (НФ3) порядка.
4. Средствами ERWin на основе физической модели данных выполнить генерацию SQL-кода для создания реляционной базы данных ПС.
5. В соответствии с требованиями технического задания, разработанного при выполнении лабораторной работы №3, провести обоснованный выбор СУБД.
6. В выбранной СУБД развернуть БД, доработать её структуру с учетом возможной нормализации отношений, а также доработать структуру таблиц с учетом ограничений на значения полей. Сформировать ER-диаграмму БД.
7. Выполнить описание таблиц БД в формате табл. 9. Краткое описание таблиц свести в табл. 10. Описание связей в БД и условия целостности данных привести в виде табл. 11.

Таблица 9. Структура таблицы \_\_\_\_\_\_\_\_\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Размер, байт | Условие на значение\* | Значение по умолчанию\* | Примечание\* |

Таблица 10. Список разработанных таблиц

|  |  |  |
| --- | --- | --- |
| №  п/п | Имя таблицы | Описание |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Родительская таблица | | Дочерняя таблица | | Тип связи |
| Название | Атрибут PK | Название | Атрибут FK |

Таблица 11. Связи между таблицами БД

**Индивидуальная тема:** программное средство автоматизации продаж в интернет-магазине солнечных панелей «sunTechnic».

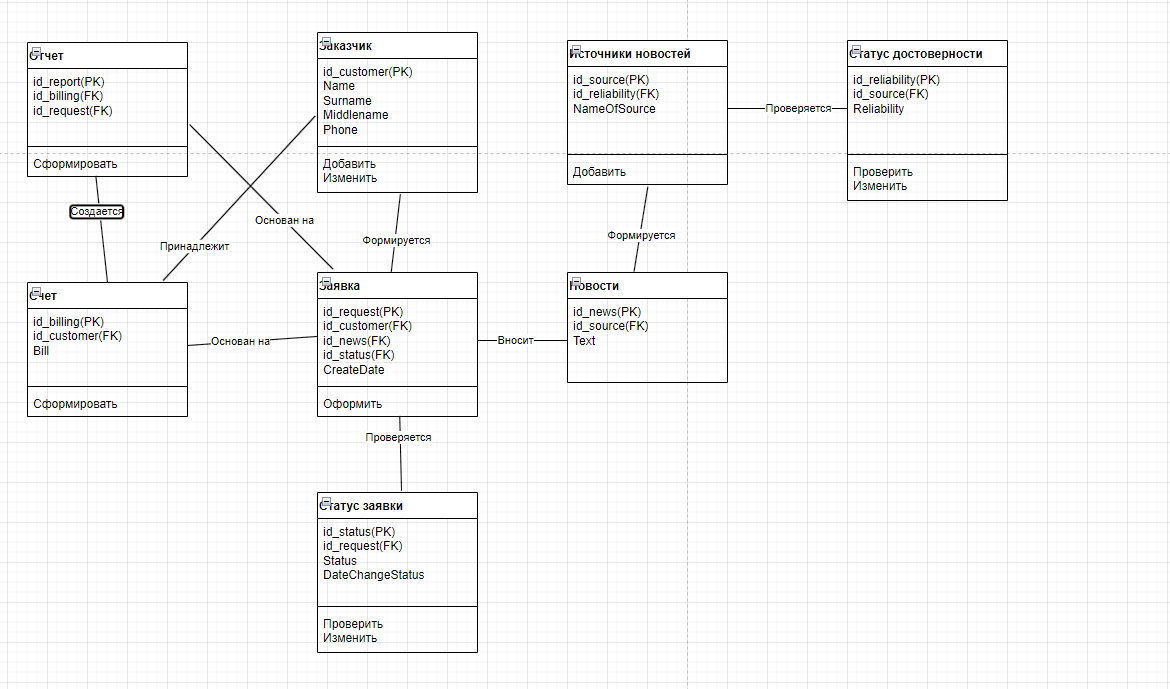
* + - 1. **Идентификация сущности**



Рисунок 1 – Формирование таблиц из сущностей

|  |  |
| --- | --- |
| Сущность | Название таблиц |
| Заказчик | Информация о заказчике (customer) |
| Заявка | Данные заявки (request) |
| Статус заявки | Информация о статусе заявки (request status) |
| Счет | Счет (bill) |
| Отчет | Информация об отчете (report) |
| Статус достоверности | Достоверность источника (reliability) |
| Источники новостей | Информация об источнике новостей (source) |
| Новости | Новостные сообщения (news) |

Таблица 1 – Соответствие название таблиц и сущностей

1. **ER-диаграмма логической модели данных**

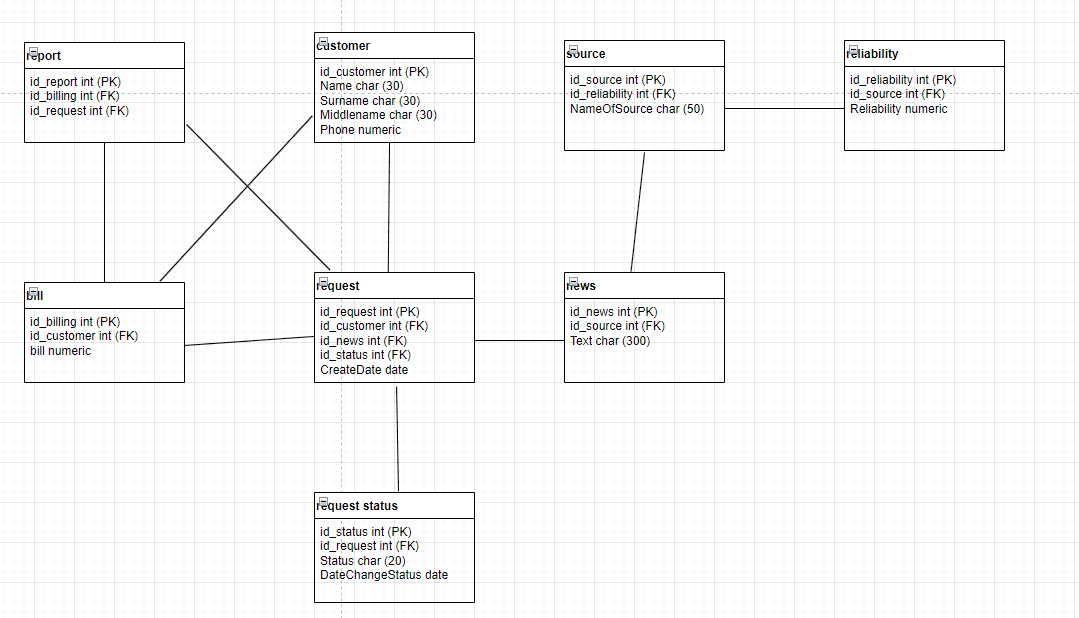


Рисунок 2 – Логическая модель данных

1. **Сравнительный анализ**

Сравнивая диаграмму классов с полученной логической схемой базы данных, можно отметить, что они идентичны. Сравнительный анализ с диаграммой классов показал на верную проектировку будущей БД. Также была проведена нормализация данных.

1. **ER-диаграмма БД**

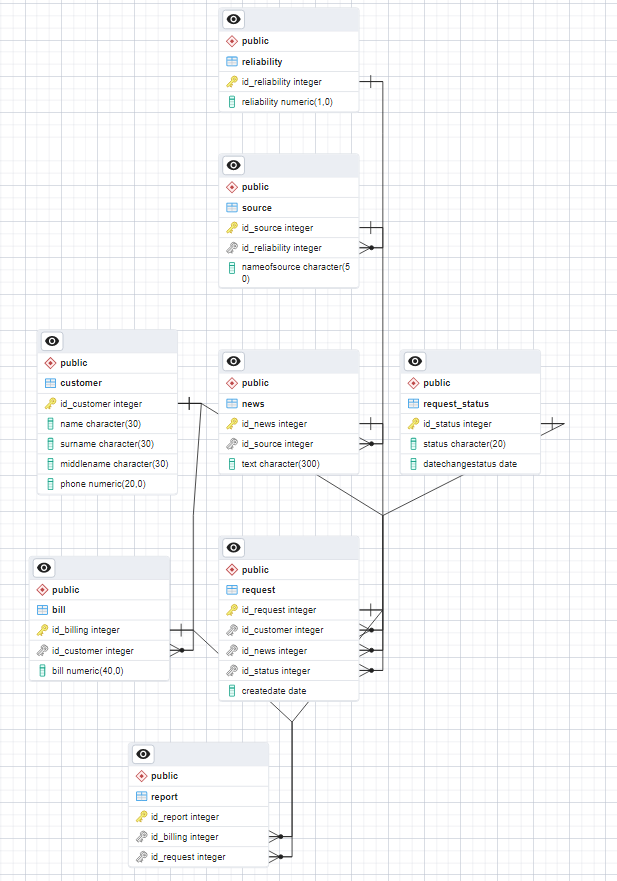


Рисунок 3 – ER-диаграмма

|  |  |  |
| --- | --- | --- |
| Сущность | Атрибуты | Описание |
| Справочные | | |
| 1 Заказчик (customer) | Идентификатор заказчика, имя, фамилия, отчество, номер телефона | Информация о заказчиках и соответствие им своего id |
| 2 Достоверность (reliability) | Идентификатор достоверности, статус достоверности | Информация о достоверности источника |
| 3 Статус заявки (request\_status) | Идентификатор статуса, статус заявки | Информация о статусе заявки |
| Оперативные | | |
| 4 Источники (source) | Идентификатор источника, идентификатор достоверности, название источника | Описывает источник новостных сообщений |
| 5. Новости (news) | Идентификатор новостей, идентификатор источника, текст новостей | Информация о новостях |
| 6 Заявка (request) | Идентификатор заявки, идентификатор новостей, идентификатор заказчика, идентификатор статуса, дата создания заявки | Общая информация о заявке |
| 7 Счет (bill) | Идентификатор счета, идентификатор заявки, номер счета | Информация о счете |
| 8 Отчет (report) | Идентификатор отчета, идентификатор счета, идентификатор заявки | Информация об отчете |

Таблица 3 – сущности физической модели

1. **SQL-код**

CREATE TABLE customer

(

id\_customer serial NOT NULL,

name char(30) NOT NULL,

surname char(30) NOT NULL,

middlename char(30) NOT NULL,

phone numeric(20) NOT NULL,

PRIMARY KEY (id\_customer)

);

CREATE TABLE bill

(

id\_billing serial NOT NULL,

id\_customer serial NOT NULL,

bill numeric(40) NOT NULL,

PRIMARY KEY (id\_billing),

FOREIGN KEY (id\_customer) REFERENCES customer (id\_customer)

);

CREATE TABLE reliability

(

id\_reliability serial NOT NULL,

reliability numeric(1) NOT NULL,

PRIMARY KEY (id\_reliability)

);

CREATE TABLE source

(

id\_source serial NOT NULL,

id\_reliability serial NOT NULL,

nameofsource char(50) NOT NULL,

PRIMARY KEY (id\_source),

FOREIGN KEY (id\_reliability) REFERENCES reliability (id\_reliability)

);

CREATE TABLE news

(

id\_news serial NOT NULL,

id\_source serial NOT NULL,

text char(300) NOT NULL,

PRIMARY KEY (id\_news),

FOREIGN KEY (id\_source) REFERENCES source (id\_source)

);

CREATE TABLE request\_status

(

id\_status serial NOT NULL,

status char(20) NOT NULL,

datechangestatus date NOT NULL,

PRIMARY KEY (id\_status)

);

CREATE TABLE request

(

id\_request serial NOT NULL,

id\_customer serial NOT NULL,

id\_news serial NOT NULL,

id\_status serial NOT NULL,

createdate date NOT NULL,

PRIMARY KEY (id\_request),

FOREIGN KEY (id\_customer) REFERENCES customer (id\_customer),

FOREIGN KEY (id\_news) REFERENCES news (id\_news),

FOREIGN KEY (id\_status) REFERENCES request\_status (id\_status)

);

CREATE TABLE report

(

id\_report serial NOT NULL,

id\_billing serial NOT NULL,

id\_request serial NOT NULL,

PRIMARY KEY (id\_report),

FOREIGN KEY (id\_billing) REFERENCES bill (id\_billing),

FOREIGN KEY (id\_request) REFERENCES request (id\_request)

);

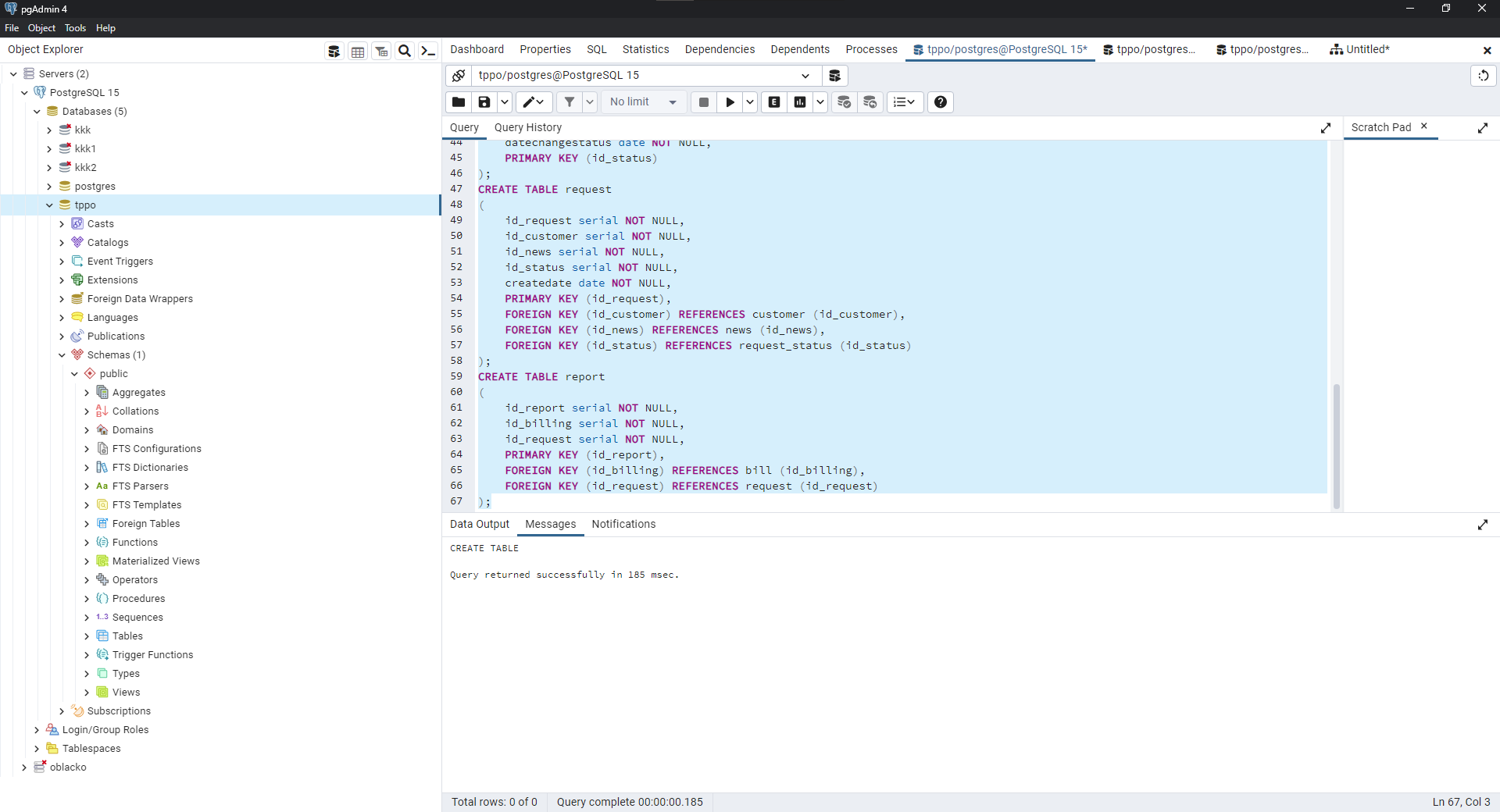


Рисунок 4 – Реализация Базы данных

1. **Выбор СУБД**

В качестве используемой для данной системы СУБД был выбран PostgreSQL — это популярная свободная объектно-реляционная система управления базами данных. Среди преимуществ: поддержка БД неограниченного размера, мощные и надёжные механизмы транзакций и репликации, легкая расширяемость, широкая функциональность.

Также стоит подчеркнуть, что это enterprise решение, которое используется во многих крупных проектах. Поэтому не возникнет проблем с поддержкой и помощью от сообщества. В том числе это хороший опыт для работы с ней в будущем.

1. **Связи между БД**

|  |  |  |
| --- | --- | --- |
| № п/п | Имя таблицы | Описание |
| 1 | Заказчик (customer) | Таблица с данными о заказчике. |
| 2 | Статус достоверности (reliability) | Таблица с данными о достоверности источника. |
| 3 | Источники новостей (source) | Таблица с данными об источниках новостей. |
| 4 | Новости (news) | Таблица с данными о новостях |
| 5 | Заявка (request) | Таблица с заявками |
| 6 | Статус заявки (request\_status) | Таблица с данными о статусе заявки |
| 7 | Счет (bill) | Таблица с информацией о счете |
| 8 | Отчет (report) | Таблица с информацией об отчете |

Таблица 4 Список разработанных таблиц

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Родительская таблица | | Дочерняя таблица | | Тип связи |
| Название | Атрибут | Название | Атрибут |
| reliability | id\_reliability | source | id\_reliability | 1:N |
| source | id\_sorce | news | id\_sorce | 1:N |
| news | id\_news | request | id\_news | 1:N |
| request\_status | id\_status | request | id\_status | 1:N |
| customer | id\_customer | request | id\_customer | 1:N |
| customer | id\_customer | bill | id\_customer | 1:N |
| bill | id\_billing | report | id\_billing | 1:N |
| requests | id\_request | report | id\_request | 1:N |

Таблица 5 - Связи между таблицами БД

**Вывод :**

В ходе проделанной работы был получен опыт по:

– идентификации сущностей информационной базы ПС и связей между ними;

– разработке ER-диаграммы логической модели данных;

– сравнительному анализу полученной логической модели с моделью классов, разработанной раннее;

– нормализации сущностей логической модели данных и разработке ER-диаграммы физической модели данных;

– генерации SQL-кода для создания реляционной базы данных ПС;

– описанию таблиц БД.