МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное   
учреждение высшего образования

**"Южно-Уральский государственный университет**

**(национальный исследовательский университет)"**

**Высшая школа электроники и компьютерных наук**

**Кафедра системного программирования**

**СВОДНЫЙ ОТЧЕТ**

о выполнении практического задания

по дисциплине

«Хранилища данных»

|  |  |
| --- | --- |
|  | Выполнил:   студент группы КЭ-303  Старостенок Д.В.  Проверил:  ст. преподаватель кафедры СП  Петрова Л.Н. |

Челябинск-2022

ОГЛАВЛЕНИЕ

[1. РАЗРАБОТКА МОДЕЛИ ПРЕДМЕТНОЙ ОБЛАСТИ 3](#_Toc114435818)

[2. РАЗРАБОТКА СХЕМЫ БАЗЫ ДАННЫХ И ОГРАНИЧЕНИЙ ЦЕЛОСТНОСТИ 3](#_Toc114435819)

[3. ПРЕДСТАВЛЕНИЯ 3](#_Toc114435820)

[4. ЗАПРОСЫ 3](#_Toc114435821)

[5. ОПРЕДЕЛЕНИЕ ОГРАНИЧЕНИЙ БЕЗОПАСНОСТИ ДАННЫХ. РАЗРАБОТКА ТРИГГЕРОВ 3](#_Toc114435822)

# 1. Разработка модели предметной области

Разработана формальная модель предметной области для небольшой строительной фирмы, которая занимается ремонтом помещений, в виде ER диаграммы по нотации Чена (Рис. 1), созданная на сайте draw.io.

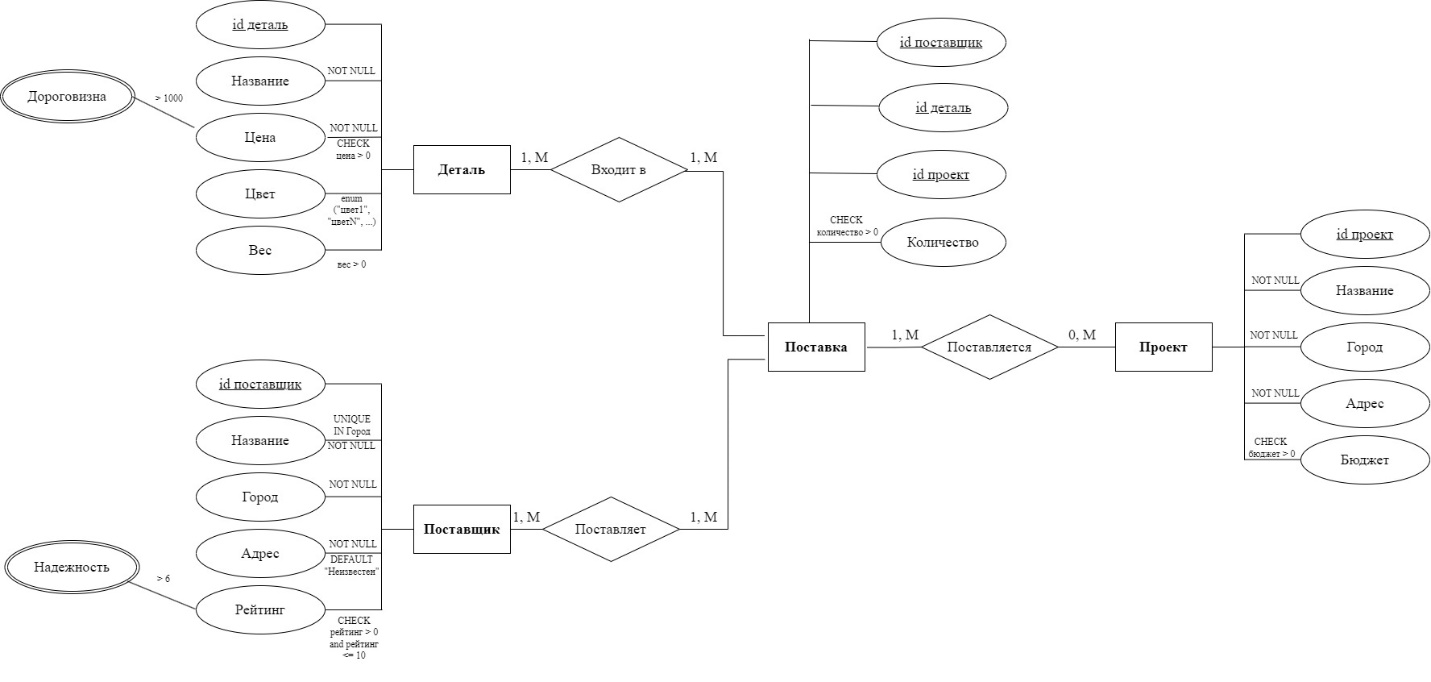


Рис. 1 – ER диаграмма

# 2. Разработка схемы базы данных и ограничений целостности

Были разработаны следующие запросы, для создания реляционной базы данных:

Запрос для создания собственного типа данных набора цветов:

CREATE TYPE COLORS\_CUSTOM

AS ENUM ('белый', 'черный', 'красный', 'синий', 'серый', 'зеленый', 'желтый', 'оранжевый');

Создание таблицы “Детали”:

CREATE TABLE details(

id\_detail INT PRIMARY KEY NOT NULL,

detail\_name VARCHAR(120) NOT NULL,

detail\_price INT NOT NULL CHECK (detail\_price > 0),

detail\_color colors\_custom NOT NULL,

detail\_weight INT NOT NULL CHECK (detail\_weight > 0))

Создание таблицы “Города”:

CREATE TABLE cities(

id\_city INT PRIMARY KEY NOT NULL,

city\_name VARCHAR(120) NOT NULL)

Создание таблицы “Поставщик”:

CREATE TABLE providers(

id\_provider INT PRIMARY KEY NOT NULL,

provider\_name VARCHAR(120) NOT NULL,

id\_city INT NOT NULL,

provider\_address VARCHAR(120) DEFAULT 'Неизвестен' NOT NULL,

provider\_rating INT NOT NULL CHECK (provider\_rating > 0 AND provider\_rating <= 10),

FOREIGN KEY (id\_city) REFERENCES cities(id\_city) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT Same\_provider\_name\_in\_city

UNIQUE (provider\_name, id\_city))

Создание таблицы “Проект”:

CREATE TABLE projects(

id\_project INT PRIMARY KEY NOT NULL,

project\_name VARCHAR(120) NOT NULL,

id\_city INT NOT NULL,

address VARCHAR(120),

budget INT NOT NULL CHECK(budget>0),

FOREIGN KEY (id\_city) REFERENCES cities(id\_city) ON DELETE CASCADE ON UPDATE CASCADE)

Создание таблицы “Поставка”:

CREATE TABLE supplies(

id\_provider INT NOT NULL,

id\_detail INT NOT NULL,

id\_project INT NOT NULL,

supplies\_quantity INT NOT NULL CHECK(supplies\_quantity>0),

PRIMARY KEY(id\_provider, id\_detail, id\_project),

FOREIGN KEY (id\_provider) REFERENCES providers(id\_provider) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (id\_detail) REFERENCES details(id\_detail) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (id\_project) REFERENCES projects(id\_project) ON DELETE CASCADE ON UPDATE CASCADE)

# 3. Представления

Задание 4. Разработка тестовой базы данных.

Разработайте тестовую базу данных таким образом, чтобы в каждой таблице содержалось не менее 20 записей с помощью команды INSERT и выгрузкой из csv файлов.

Вставка данных в таблицу “Города” с помощью команды INSERT:

INSERT INTO cities(city\_name)

VALUES ('Челябинск');

Вставка данных в таблицу “Города” с помощью команды COPY:

COPY cities(city\_name)

FROM 'C:\Users\Dmitriy\Desktop\PostgreQuery\csv files\cities.csv'

DELIMITER ','

CSV HEADER encoding 'UTF8';

Вставка данных в таблицу “Детали” с помощью команды INSERT:

INSERT INTO details(

detail\_name, detail\_price, detail\_color, detail\_weight)

VALUES ('Шуруп', 16, 'черный', 3);

Вставка данных в таблицу “Детали” с помощью команды COPY:

COPY details(detail\_name, detail\_price, detail\_color, detail\_weight)

FROM 'C:\Users\Dmitriy\Desktop\PostgreQuery\csv files\details.csv'

DELIMITER ';'

CSV HEADER encoding 'UTF8';

Вставка данных в таблицу “Проекты” с помощью команды INSERT:

INSERT INTO projects(

project\_name, id\_city, address, budget)

VALUES ('Стена', 3, 'Иванова 22', 12000);

Вставка данных в таблицу “Проекты” с помощью команды COPY:

COPY projects(project\_name, id\_city, address, budget)

FROM 'C:\Users\Dmitriy\Desktop\PostgreQuery\csv files\projects.csv'

DELIMITER ';'

CSV HEADER encoding 'UTF8';

Вставка данных в таблицу “Поставщики” с помощью команды INSERT:

INSERT INTO providers(

provider\_name, id\_city, provider\_address, provider\_rating)

VALUES ('Ремонт в дом', 3, 'Ленина 25', 7);

Вставка данных в таблицу “Поставщики” с помощью команды COPY:

COPY providers(provider\_name, id\_city, provider\_address, provider\_rating)

FROM 'C:\Users\Dmitriy\Desktop\PostgreQuery\csv files\providers.csv'

DELIMITER ';'

CSV HEADER encoding 'UTF8';

Вставка данных в таблицу “Поставки” с помощью команды INSERT:

INSERT INTO public.supplies(

id\_provider, id\_detail, id\_project, supplies\_quantity)

VALUES (1, 1, 1, 22);

Вставка данных в таблицу “Поставки” с помощью команды COPY:

COPY supplies(id\_provider, id\_detail, id\_project, supplies\_quantity)

FROM 'C:\Users\Dmitriy\Desktop\PostgreQuery\csv files\supplies.csv'

DELIMITER ';'

CSV HEADER encoding 'UTF8';

Задание 5. Разработка представлений с помощью VIEW для ролей “Экономист” и “Директор”.

Для роли Экономиста.

1. Экономист имеет право осуществлять выборку, добавление, обновление и удаление данных о поставщиках и поставках, а также выборку данных о деталях и проектах.

Выборка данных о поставщиках:

SELECT id\_provider, provider\_name, cities.city\_name, provider\_address, provider\_rating

FROM providers

INNER JOIN cities ON cities.id\_city = providers.id\_city

Добавление данных о поставщиках:

INSERT INTO providers(

id\_provider, provider\_name, id\_city, provider\_address, provider\_rating)

VALUES (?, ?, ?, ?, ?);

Обновление данных о поставщиках:

UPDATE public.providers

SET id\_provider=?, provider\_name=?, id\_city=?, provider\_address=?, provider\_rating=?

WHERE <condition>;

Удаление данных о поставщиках:

DELETE FROM public.providers

WHERE <condition>;

Выборка данных о поставках:

SELECT providers.provider\_name, details.detail\_name, projects.project\_name, supplies.supplies\_quantity

FROM supplies

INNER JOIN providers ON providers.id\_provider = supplies.id\_provider

INNER JOIN details ON details.id\_detail = supplies.id\_detail

INNER JOIN projects ON projects.id\_project = supplies.id\_project

Добавление данных о поставках:

INSERT INTO public.supplies(

id\_provider, id\_detail, id\_project, supplies\_quantity)

VALUES (?, ?, ?, ?);

Обновление данных о поставках:

UPDATE public.supplies

SET id\_provider=?, id\_detail=?, id\_project=?, supplies\_quantity=?

WHERE <condition>;

Удаление данных о поставках:

DELETE FROM public.supplies

WHERE <condition>;

Выборка данных о деталях:

SELECT \* FROM details

Выборка данных о проектах:

SELECT id\_project, project\_name, cities.city\_name, address, budget

FROM projects

INNER JOIN cities ON cities.id\_city = projects.id\_city

1. При выборке данных Поставщик для Экономиста должен быть представлен следующими атрибутами: название, город, адрес, рейтинг, признак «ненадежный» или «надежный» (символьная строка). Данные должны быть отсортированы следующим образом: по городу, названию, убыванию рейтинга.

CREATE VIEW data\_from\_provider\_economist AS

SELECT

provider\_name AS Название,

cities.city\_name AS Город,

provider\_address AS Адрес,

provider\_rating AS Рейтинг,

CASE

WHEN provider\_rating>=6

THEN 'Надежный'

WHEN provider\_rating<6

THEN 'Ненадежный'

END Надежность

FROM providers

INNER JOIN cities ON cities.id\_city = providers.id\_city

ORDER BY cities.city\_name ASC, providers.provider\_name ASC, providers.provider\_rating DESC

1. Деталь для Экономиста должна быть представлена следующими атрибутами: название, цена в тыс.руб., цвет, вес в килограммах, признак «дешевая» или «дорогая» (символьная строка). Данные должны быть отсортированы следующим образом: по убыванию цены, названию, цвету, возрастанию веса.

CREATE VIEW data\_from\_details\_economist AS

SELECT

detail\_name AS Название,

detail\_price AS Цена,

detail\_color AS Цвет,

detail\_weight/1000::float4 AS "Вес, кг",

CASE

WHEN detail\_price>=1000

THEN 'Дорогая'

WHEN detail\_price<1000

THEN 'Дешевая'

END Дороговизна

FROM details

ORDER BY detail\_price DESC, detail\_name ASC, detail\_color ASC, detail\_weight ASC

1. При выборке данных Проект для Экономиста должен быть представлен следующими атрибутами: название, город, адрес, бюджет. Данные должны быть отсортированы следующим образом: по городу, названию, убыванию бюджета.

CREATE VIEW data\_from\_projects\_economist AS

SELECT

project\_name AS Название,

cities.city\_name AS Город,

address AS Адрес,

budget AS Бюджет

FROM projects

INNER JOIN cities ON cities.id\_city = projects.id\_project

ORDER BY cities.city\_name ASC, projects.project\_name ASC, projects.budget DESC

1. При выборке данных Поставка для Экономиста должна быть представлена следующими атрибутами: название детали, город детали, цвет детали, признак детали «дешевая» или «дорогая» (символьная строка), название поставщика, город поставщика, признак поставщика «ненадежный» или «надежный» (символьная строка), количество деталей, вес поставки в килограммах, стоимость поставки в тыс.руб. Данные должны быть отсортированы следующим образом: названию детали, названию поставщика, по убыванию стоимости, по убыванию веса.

CREATE VIEW data\_from\_supplies\_economist AS

SELECT

details.detail\_name AS "Название детали",

details.detail\_color AS "Цвет детали",

CASE

WHEN details.detail\_price>=1000

THEN 'Дорогая'

WHEN details.detail\_price<1000

THEN 'Дешевая'

END Дороговизна\_детали,

providers.provider\_name AS "Название поставщика",

cities.city\_name AS "Город поставщика",

CASE

WHEN providers.provider\_rating>=6

THEN 'Надежный'

WHEN providers.provider\_rating<6

THEN 'Ненадежный'

END Надежность\_провайдера,

supplies.supplies\_quantity AS "Количество деталей",

(details.detail\_weight \* supplies.supplies\_quantity)/1000::float4 AS "Вес поставки, кг",

(details.detail\_price \* supplies.supplies\_quantity) AS "Стоимость поставки"

FROM supplies

INNER JOIN providers ON providers.id\_provider = supplies.id\_provider

INNER JOIN details ON details.id\_detail = supplies.id\_detail

INNER JOIN projects ON projects.id\_project = supplies.id\_project

INNER JOIN cities ON cities.id\_city = providers.id\_city

ORDER BY details.detail\_name,

providers.provider\_name,

(details.detail\_price \* supplies.supplies\_quantity) DESC,

details.detail\_weight DESC

Для роли Директора

1. Директор имеет право осуществлять выборку, добавление, обновление и удаление данных о поставщиках, деталях и проектах, а также выборку и удаление данных о поставках.

Выборка данных о поставщиках:

SELECT id\_provider, provider\_name, cities.city\_name, provider\_address, provider\_rating

FROM providers

INNER JOIN cities ON cities.id\_city = providers.id\_city

Добавление данных о поставщиках:

INSERT INTO providers(

id\_provider, provider\_name, id\_city, provider\_address, provider\_rating)

VALUES (?, ?, ?, ?, ?);

Обновление данных о поставщиках:

UPDATE public.providers

SET id\_provider=?, provider\_name=?, id\_city=?, provider\_address=?, provider\_rating=?

WHERE <condition>;

Удаление данных о поставщиках:

DELETE FROM public.providers

WHERE <condition>;

Выборка данных о деталях:

SELECT \* FROM details

Добавление данных о деталях:

INSERT INTO public.details(

id\_detail, detail\_name, detail\_price, detail\_color, detail\_weight)

VALUES (?, ?, ?, ?, ?);

Обновление данных о деталях:

UPDATE public.details

SET id\_detail=?, detail\_name=?, detail\_price=?, detail\_color=?, detail\_weight=?

WHERE <condition>;

Удаление данных о деталях:

DELETE FROM public.details

WHERE <condition>;

Выборка данных о проектах:

Добавление данных о проектах:

Обновление данных о проектах:

Удаление данных о проектах:

1. При выборке данных Поставщик для Директора должен быть представлен следующими атрибутами: название, город, признак «ненадежный» или «надежный» (символьная строка). Данные должны быть отсортированы следующим образом: по признаку, городу, названию.
2. При выборке данных Деталь для Директора должна быть представлена следующими атрибутами: название, цена в тыс.руб., цвет, признак «дешевая» или «дорогая» (символьная строка). Данные должны быть отсортированы следующим образом: по убыванию цены, названию.
3. При выборке данных Проект для Директора должен быть представлен следующими атрибутами: название, город, бюджет. Данные должны быть отсортированы следующим образом: по убыванию бюджета, городу, названию.
4. При выборке данных Поставка для Директора должна быть представлена следующими атрибутами: название детали, город детали, цвет детали, признак детали «дешевая» или «дорогая» (символьная строка), название поставщика, город поставщика, признак поставщика «ненадежный» или «надежный» (символьная строка), количество деталей, вес поставки в килограммах, стоимость поставки в тыс.руб. Данные должны быть отсортированы следующим образом: по признаку детали, городу детали, признаку поставщика, городу поставщика, названию детали, названию поставщика, по убыванию стоимости, по убыванию веса.

# 4. Запросы

# 5. Определение ограничений безопасности данных. Разработка триггеров