

### Задание

1. Спроектировать базу данных согласно заданному набору атрибутов и сущностей
2. Выполнить основные виды операций в БД, таких как простой запрос, вложенный запрос, реализация процедур и функций, вывод информации при помощи конструкции view, создание триггеров
3. На основании проделанной работы необходимо создать отчет, содержащий в себе, все перечисленное выше

### Ход работы

#### Концептуальное проектирование

##### Определение типов сущности:

Сущности:	Описание сущности:
Склад	Содержит сведения о количестве продуктов на складе
Поставки	Содержит сведения об поставке
Поставщик	Содержит информацию о сотруднике
Продукты	Содержит информацию о продукте
Касса	Содержит информацию о количестве денег
Блюдо	Содержит информацию о названии блюд
Рецепт	Содержит информацию о составе блюда
Состав поставки	Содержит информацию о продуктах поставки
Регистрация	Содержит информацию о зарегистрированных номерах
Меню	Содержит информацию о блюдах и ценах
Трапеза	Содержит информацию о трапезе посетителя
Вид блюда	Содержит информацию о виде блюда
Столовая	Содержит информацию о посетителях в столовой
Посетитель	Содержит информацию о посетителе

Списание	Содержит информацию о количестве списанных продуктов
Путевки	Содержит информацию о путевках
Выбор блюда	Содержит информацию о выборе блюда

Определение типов связи:

Тип сущности 1:	Кратность:	Тип связи:	Кратность:	Тип сущности 2:
склад	1..1	имеет	1..*	продукт
поставки	1..*	осуществляет	1..1	поставщик
состав поставки	1..*	имеет	1..1	продукт
состав поставки	1..*	входит	1..1	поставки
меню	1..1	имеет	1..*	блюдо
рецепт	1..*	имеет	1..1	продукт
рецепт	1..*	имеет	1..1	блюдо
регистрация	1..1	делается	1..1	посетитель
трапеза	1..*	входит	1..1	выбор блюда
трапеза	1..*	осуществляется	1..1	посетитель
списание	1..1	списывается	1..1	продукт
выбор блюда	1..1	входит	1..*	блюдо
столовая	1..1	посещает	1..1	посетитель
путевки	1..1	выдается	1..1	посетитель

Определение атрибутов и связывание их с типами сущностей и связями.

Выбор блюда:

Наименование:	Описание:	Вид:	Тип:	Домен:	Ключи:	Обязательно к заполнению:
produkt	Идентификатор продукта	Составной однозначный	varchar(50)	Кириллица	PK, FK	+
trapeza	Идентификатор трапезы	Составной однозначный	int	Целые числа >0	PK, FK	+

Склад:

Наименование:	Описание:	Вид:	Тип:	Домен:	Ключи:	Обязательно к заполнению:
produkt	Идентификатор продукта	Составной однозначный	varchar(50)	Кириллица	PK, FK	+
ves	Количество продукта	Составной однозначный	int	Целые числа >0		+

Поставщики:

Наименование:	Описание:	Вид:	Тип:	Домен:	Ключи:	Обязательно к заполнению:
---------------	-----------	------	------	--------	--------	---------------------------

postavschik	Поставщик	Составной однозначный	VARCHAR(50)	Кириллица	PK	+
telephone	Номер телефона	Простой однозначный	varchar(11)	Целые числа >0		+

#### Продукты:

Наименование:	Описание:	Вид:	Тип:	Домен:	Ключи:	Обязательно к заполнению:
produkt	Продукт	Составной однозначный	VARCHAR(50)	Кириллица	PK	+
kilokalorii	Килокалории	Простой однозначный	decimal(6, 2)	Числа >0		+

#### Поставки:

Наименование:	Описание:	Вид:	Тип:	Домен:	Ключи:	Обязательно к заполнению:
id	Идентификатор поставки	Простой однозначный	INT	Целые числа > 0	PK	+
postavschik	Поставщик	Составной однозначный	varchar(50)	Кириллица	FK	+

data_postavki	Дата поставки	Составной однозначный	date	Дата		+
total_price	Общая цена	Простой однозначный	Decimal(6,2)	Числа >0		+

Состав поставки:

Наименование:	Описание:	Вид:	Тип:	Домен:	Ключи:	Обязательно к заполнению:
produkt	Продукт	Составной однозначный	varchar(20)	Кириллица	FK	+
kolichество	Количество	Простой однозначный	int	Целые числа > 0		+
cena	Цена	Простой однозначный	Decimal(6,2)	Числа > 0		+
id_postavki	Идентификатор поставки	Простой однозначный	int	Целые числа > 0	FK	+

Блюда:

Наименование:	Описание:	Вид:	Тип:	Домен:	Ключи:	Обязательно к заполнению:
bludo	Блюдо	Простой однозначный	varchar(50)	Кириллица	PK	+

Vid_bluda	Вид блюда	Простой однозначный	varchar(50)	Кириллица	FK	+
Osnovnoe	Основное меню	Простой однозначный	bool	'1', '0'		+

#### Касса:

Наименование:	Описание:	Вид:	Тип:	Домен:	Ключи:	Обязательно к заполнению:
total	Количество денег	Простой однозначный	numeric(8,2)	Числа > 0		+

#### Меню:

Наименование:	Описание:	Вид:	Тип:	Домен:	Ключи:	Обязательно к заполнению:
bludo	Блюдо	Составной однозначный	varchar(50)	Кириллица	PK, FK	+
vid_bluda	Вид блюда	Составной однозначный	varchar(10)	Кириллица		+
price	Цена	Простой однозначный	numeric(6,2)	Числа > 0		+

### Рецепт:

Наименование:	Описание:	Вид:	Тип:	Домен:	Ключи:	Обязательно к заполнению:
bludo	Блюдо	Составной однозначный	varchar(50)	Кириллица	PK, FK	+
produkt	Продукт	Составной однозначный	varchar(50)	Кириллица	PK, FK	+
weight	Вес	Простой однозначный	int	Целые числа > 0		+

### Регистрация:

Наименование:	Описание:	Вид:	Тип:	Домен:	Ключи:	Обязательно к заполнению:
nomer	Номер отеля	Простой однозначный	int	Целые числа > 0	PK	+
visitor	Посетитель	Простой однозначный	int	Целые числа > 0	FK	+

### Трапеза:

Наименование:	Описание:	Вид:	Тип:	Домен:	Ключи:	Обязательно к заполнению:
id	Идентификатор трапезы	Простой однозначный	INT	Целые числа > 0	PK	+

man_id	Посетитель	Простой однозначный	int	Целые числа > 0	FK	+
eat_day	Дата трапезы	Составной однозначный	date	Дата		+
price	Стоимость	Простой однозначный	numeric (8,2)	Числа > 0		+
bludo	Блюдо	Составной однозначный	varchar(50)	Кириллица	FK	+

Вид блюда:

Наименование:	Описание	Вид	Тип	Домен:	Ключи	Обязательно к заполнению
meal	Вид блюда	Простой однозначный	varchar(10)	Кириллица	PK, FK	+

Посетитель:

Наименование	Описание	Вид	Тип	Домен	Ключи	Обязательно к заполнению
id	Идентификатор посетителя	Простой однозначный	INT	Целые числа > 0	PK	+
name_	Имя	Простой однозначный	varchar(50)	Кириллица		+



surname	Фамилия	Простой однознач ный	varchar( 50)	Кирилли ца		+
---------	---------	----------------------------	-----------------	---------------	--	---

Списание:

Наименова ние:	Описани е:	Вид:	Тип:	Домен:	Ключ чи:	Обязател ьно к заполнени ю:
product	Продукт	Простой однознач ный	varchar( 50)	Кирилл ица	FK	+
kolichestvo	Количес тво	Простой однознач ный	int	Целые числа > 0		+

Столовая:

Наименован ие:	Описани е:	Вид:	Тип :	Домен :	Ключ и:	Обязатель но к заполнению :
visitor	Посетите ль	Простой однозначн ый	int	Целые числа > 0	FK	+
mesto	Место	Простой однозначн ый	int	Целые числа > 0		+

Путевки:

Наименова ние:	Описание:	Вид:	Тип :	Доме н:	Ключ и:	Обязател ьно к заполнени ю:
id_visitor	Идентифика тор посетителя	Простой однозначн ый	int	Целы е числа > 0	FK	+

id	Идентифика тор путевки	Простой однозначн ый	int	Целы е числа > 0	РК	+
price	Цена	Простой однозначн ый	int	Целы е числа > 0		+

## ER-диаграмма

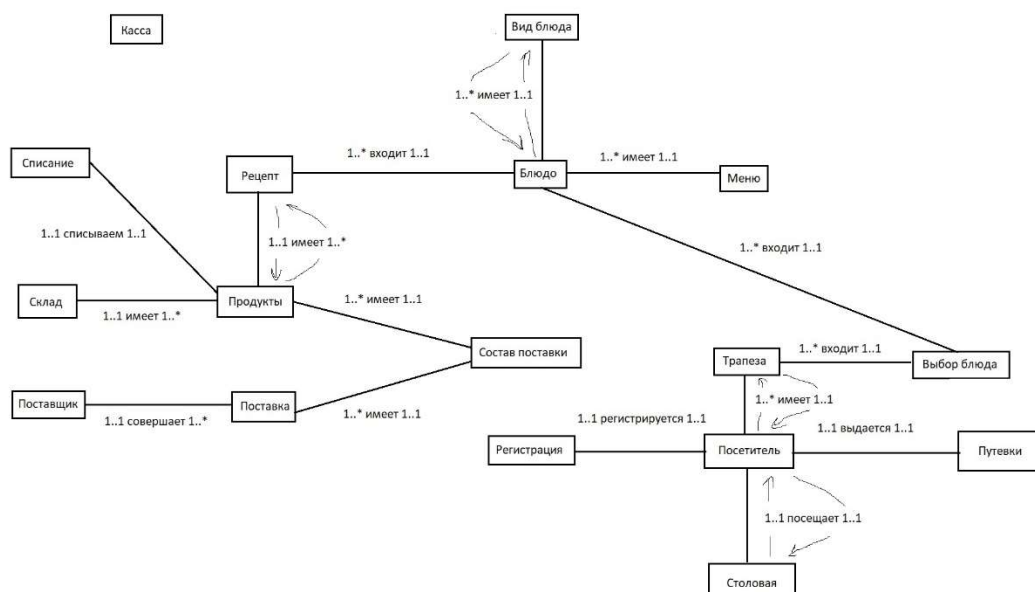


Рисунок 1 – ER-диаграмма

### Логическое проектирование

В ходе проверки не были найдены связи М:М.

Рекурсивных связей обнаружено не было.

Сложных связей и связей с атрибутами обнаружено не было.

Многозначных атрибутов обнаружено не было.

### Определение набора отношений

#### Sostav\_postavki

№	Идентификатор поля	Тип данных	Ограничения целостности	Ограничения	Ключи
1	produkt	Varchar(50)	Not null	Кириллица	FK
2	kolichество	integer	Not null	Целые числа >0	
3	cena	integer	Not null	Целые числа >0	
4	id_postavki	integer	Not null	Целые числа >0	FK

#### bluda

№	Идентификатор поля	Тип данных	Ограничения целостности	Ограничения	Ключи
1	bludo	Varchar(50)	Not null	Кириллица	PK
2	Vid_bluda	Varchar(50)	Not null	Кириллица	FK
3	Osnovnoe	bool	Not null	'1', '0'	

kassa

№	Идентификатор поля	Тип данных	Ограничения целостности	Ограничения	Ключи
1	Total	numeric(8, 2)	Not null	Числа >0	

menu

№	Идентификатор поля	Тип данных	Ограничения целостности	Ограничения	Ключи
1	bludo	Varchar(50)	Not null	Кириллица	PK, FK
2	vid_bluda	Varchar(10)	Not null	Кириллица	
3	price	numeric(6, 2)	Not null	Числа >0	

postavki

№	Идентификатор поля	Тип данных	Ограничения целостности	Ограничения	Ключи
1	id	integer	Not null	Целые числа >0	PK
2	postavschik	Varchar(50)	Not null	Кириллица	FK
3	total_price	numeric(6, 2)	Not null	Числа >0	
4	data_postavki	data	Not null	Дата	

postavschiki

№	Идентификатор поля	Тип данных	Ограничения целостности	Ограничения	Ключи
1	postavschik	Varchar(50)	Not null	Кириллица	РК
2	telephone	varying(11)	Not null	Целые числа >0	

produkts

№	Идентификатор поля	Тип данных	Ограничения целостности	Ограничения	Ключи
1	produkt	Varchar(50)	Not null	Кириллица	РК
2	kilokalorii	numeric(6, 2)	Not null	Числа >0	

Vibor\_bluda

№	Идентификатор поля	Тип данных	Ограничения целостности	Ограничения	Ключи
1	produkt	Varchar(50)	Not null	Кириллица	РК,FK
2	trapeza	int	Not null	Числа >0	РК,FK

## receipt

№	Идентификатор поля	Тип данных	Ограничения целостности	Ограничения	Ключи
1	bludo	Varchar(50)	Not null	Кириллица	PK, FK
2	produkt	Varchar(50)	Not null	Кириллица	PK, FK
3	weight	integer	Not null	Целые числа >0	

## registration

№	Идентификатор поля	Тип данных	Ограничения целостности	Ограничения	Ключи
1	nomer	integer	Not null	Целые числа >0	PK
2	visitor	integer	Not null	Целые числа >0	FK

## sklad

№	Идентификатор поля	Тип данных	Ограничения целостности	Ограничения	Ключи
1	produkt	Varchar(50)	Not null	Кириллица	PK, FK
2	ves	integer	Not null	Целые числа >0	

trapeza

№	Идентификатор поля	Тип данных	Ограничения целостности	Ограничения	Ключи
1	id	integer	Not null	Целые числа >0	PK
2	man_id	integer	Not null	Кириллица	FK
3	eat_day	date	Not null	Числа >0	
4	price	numeric(8, 2)	Not null	Дата	
5	bludo	Varchar(50)	Not null	Кириллица	FK

vid\_bluda

№	Идентификатор поля	Тип данных	Ограничения целостности	Ограничения	Ключи
1	meal	Varchar(10)	Not null	Кириллица	PK, FK

visitors

№	Идентификатор поля	Тип данных	Ограничения целостности	Ограничения	Ключи
1	id	integer	Not null	Целые числа >0	PK
2	name_	Varchar(50)	Not null	Кириллица	
3	surname	Varchar(50)	Not null	Кириллица	



### Stolovaya

№	Идентификатор поля	Тип данных	Ограничения целостности	Ограничения	Ключи
1	visitor	int	Not null	Целые числа > 0	FK
2	mesto	int	Not null	Целые числа > 0	

### Spisat

№	Идентификатор поля	Тип данных	Ограничения целостности	Ограничения	Ключи
1	product	varchar(50)	Not null	Кириллица	FK
2	kolichество	int	Not null	Целые числа > 0	

### Putevki

№	Идентификатор поля	Тип данных	Ограничения целостности	Ограничения	Ключи
1	id_visitor	int	Not null	Целые числа > 0	FK
2	id	int	Not null	Целые числа > 0	PK
3	price	int	Not null	Целые числа > 0	

## Запросы

1) Доля веса продукта в блюде

```
SELECT
    r.bludo,
    r.produkt,
    (
        r.weight * 100.0 / (
            SELECT
                SUM(weight)
            FROM Receipt
            WHERE
                bludo = r.bludo
        )
    ) AS weight_percent
FROM Receipt AS r;
```

	<b>bludo</b> [PK] character varying (50) 	<b>produkt</b> [PK] character varying (20) 	<b>weight_percent</b> numeric 
1	Борщ	Картофель	57.1428571428571429
2	Борщ	Морковь	28.5714285714285714
3	Борщ	Лук	14.2857142857142857
4	Стейк	Говядина	100.0000000000000000
5	Салат Греческий	Огурец	28.5714285714285714
6	Салат Греческий	Помидор	35.7142857142857143

Рисунок 1 – Доля продукта в блюде по весу

2) Определить первых 3 поставщиков, осуществивших поставки на наибольшую сумму.

```
SELECT postavki.postavschik, SUM(total_price) AS total_value
FROM sostav_postavki join postavki on postavki.id = id_postavki
GROUP BY postavki.postavschik
ORDER BY total_value DESC
LIMIT 3;
```

	<b>postavschik</b> character varying (50)	<b>total_value</b> numeric
1	АО "МегаМаркет"	5000.00
2	ООО "Рога и Копыта"	4500.00
3	ИП Иванов Иван Иванович	2000.00

Рисунок 2 – Поставщики с наибольшей суммой поставок

### 3) Ранг веса продукта

```
SELECT produkt, weight,
       DENSE_RANK() OVER (PARTITION BY bludo ORDER
BY weight ASC) AS weight_rank
FROM receipt;
```

	<b>produkt</b> character varying (20)	<b>weight</b> integer	<b>weight_rank</b> bigint
1	Лук красный	15	1
2	Маслины	15	1
3	Брынза	20	2
4	Огурец	40	3
5	Помидор	50	4

Рисунок 3 – Относительный ранг веса продукта в блюде

### 4) Скользящее значение

```
WITH ProductPrices AS (
  SELECT
    p.produkt,
    data_postavki,
    total_price / p.kolichestvo AS price
  FROM sostav_postavki AS p join postavki on postavki.id =
id_postavki
), Avgc AS (
  SELECT
```

```

    produkt,
    data_postavki,
    price,
    AVG(price) OVER (PARTITION BY produkt ORDER BY
data_postavki) AS avg_c
FROM ProductPrices
)
SELECT
    r.produkt,
    r.price,
    r.data_postavki,
    r.avg_c
FROM Avgc AS r

```





	produkt character varying (20) 	price numeric 	data_postavki date 	avg_c numeric 
1	Картофель	50.0000000000000000	2024-01-11	50.0000000000000000
2	Молоко	15.0000000000000000	2024-01-11	17.5000000000000000
3	Молоко	20.0000000000000000	2024-01-11	17.5000000000000000

Рисунок 4 – Реализация скользящего значения цены

## Функции

### 1. Калории

```

CREATE FUNCTION calculate_calories4(name varchar(50))
RETURNS INT as '
    DECLARE total_calories int;
BEGIN
    SELECT SUM(weight * kilokalorii) INTO total_calories
    FROM Receipt
    JOIN Produkts
        ON Receipt.produkt = Produkts.produkt
    where name = receipt.bludo
    GROUP BY Bludo;

```

```

RETURN total_calories / 100;
END;
' language plpgsql;

```



	<b>bludo</b> [PK] character varying (50) 	<b>calculate_calories4</b>  integer
1	Борщ	104
2	Салат Цезарь	[null]
3	Салат Греческий	92

Рисунок 5 – Расчет калорий блюда

## 2. Стоимость блюд

```

select count_cost_bluda('Молочный суп')
DECLARE resulting dec(6,2);
BEGIN
SELECT
SUM(c.weight * (SELECT SUM(cena) FROM Sostav_postavki
WHERE produkt = c.produkt) / (SELECT
SUM(kolichestvo) FROM Sostav_postavki WHERE produkt =
c.produkt) ) into resulting
FROM
receipt as c

JOIN
bluda as d ON c.bludo = d.bludo
where
d.bludo = current_bludo;
return resulting;
end;

```


	count_cost_bluda 
1	126.00

Рисунок 6 – Расчет стоимости блюда

3. Количество порций, которые можно приготовить

`select bludo, calculateportionsfordish3(bludo) from bluda`

`DECLARE available_portions INT;`

`DECLARE min_portions INT;`

`BEGIN`

`-- Получаем минимальное количество порций, которое можно  
приготовить из всех составляющих продуктов`

`SELECT MIN(FLOOR(Sklad.ves / Receipt.weight))`

`INTO min_portions`

`FROM Sklad`

`JOIN Receipt ON Sklad.produkt = Receipt.produkt`

`WHERE Receipt.bludo = name;`

`-- Получаем доступное количество порций, основываясь на  
минимальном количестве порций`

`SELECT FLOOR(Sklad.ves / Receipt.weight)`

`INTO available_portions`

`FROM Sklad`

`JOIN Receipt ON Sklad.produkt = Receipt.produkt`

`WHERE Receipt.bludo = name`

`LIMIT 1;`

`RETURN available_portions;`

`END;`



	<b>bludo</b> [PK] character varying (50) 	<b>calculateportionsfordish3</b>  integer
1	Борщ	10
2	Салат Цезарь	[null]
3	Салат Греческий	180

Рисунок 7 – Расчет количества порций

#### 4. Регистрация гостей

--добавить гостя

```
declare free_mesto int;
```

```
Begin
```

```
select nomer into free_mesto from registration
```

```
where visitor is null limit 1;
```

```
update registration set visitor = name where nomer = free_mesto;
```

```
return true;
```

```
End;
```

--убрать гостя

```
begin
```

```
update registration
```

```
set visitor = null where visitor = name;
```

```
return true;
```

```
end;
```

#### 5. Списание со склада

```
DECLARE
```

```
product_record RECORD;
```

```
stock_quantity INTEGER;
```

```
BEGIN
```

```
-- Итерируем по записям в таблице "выбор"
```

```
FOR product_record IN SELECT * FROM spisat LOOP
```

```
-- Проверяем наличие достаточного количества
```

товара на складе

```
SELECT ves INTO stock_quantity
FROM sklad
WHERE produkt = product_record.produkt;
IF stock_quantity < product_record.kolichestvo THEN
    -- Недостаточно товара на складе, выводим ошибку
    RAISE EXCEPTION 'Недостаточно товара на
складе для продукта: %', product_record.produkt;
END IF;
-- Обновляем количество товара на складе
UPDATE sklad
SET ves = ves - product_record.kolichestvo
WHERE produkt = product_record.produkt;
END LOOP;
-- Очищаем таблицу "выбор"
DELETE FROM spisat;
END;
```

	produkt [PK] character varying (20)	ves integer
1	Помидор	6000
2	Брынза	2000
3	Маслины	3500

Рисунок 8 – Количество продуктов на складе

	produkt [PK] character varying (20)	kolichestvo integer
1	Помидор	7000

Рисунок 9 – Количество списываемых продуктов

---

ERROR: Недостаточно товара на складе для продукта: Помидор

Рисунок 10 – Функция списания товаров со склада



## Триггеры

### 1. Формирование меню

```
CREATE OR REPLACE FUNCTION
calculate_price_trigger() RETURNS TRIGGER AS $$
BEGIN
NEW.price := count_cost_bluda(NEW.bludo) * 5;
RETURN NEW; END; $$ LANGUAGE plpgsql;

CREATE TRIGGER calculate_price_trigger BEFORE
INSERT OR UPDATE ON menu
FOR EACH ROW EXECUTE PROCEDURE
calculate_price_trigger();
```

### 2. Деньги в кассе

```
CREATE OR REPLACE FUNCTION check_kassa_balance()
RETURNS TRIGGER AS $$
DECLARE
    available_balance NUMERIC;
BEGIN
    -- Проверяем баланс в кассе
    SELECT total INTO available_balance FROM kassa;
    -- Если баланс меньше стоимости поставки, вызываем
ошибку
    IF available_balance < NEW.total_price THEN
        RAISE EXCEPTION 'Недостаточно средств в кассе
для оплаты поставки.';
    END IF;
    -- Если все хорошо, обновляем баланс кассы
    UPDATE kassa SET total = total - NEW.total_price;
    -- Возвращаем "продолжить"
```

```

        RETURN NEW;
    END;
    $$ LANGUAGE plpgsql;
    -- Создаем триггер, который будет срабатывать перед
    вставкой новой поставки
    CREATE TRIGGER check_kassa_balance_trigger
    BEFORE INSERT ON postavki
    FOR EACH ROW
    EXECUTE PROCEDURE check_kassa_balance();

```

### 3. Удаление старых поставок

```

    CREATE OR REPLACE FUNCTION delete_old_postavki()
    RETURNS TRIGGER AS $$
    BEGIN
        -- Удаляем поставки, дата которых старше текущего года
        DELETE FROM postavki WHERE data_postavki <
CURRENT_DATE - INTERVAL '1 year';
        RETURN NEW;
    END;
    $$ LANGUAGE plpgsql;

    CREATE TRIGGER delete_old_postavki_trigger
    AFTER INSERT OR UPDATE OR DELETE ON postavki
    FOR EACH ROW EXECUTE PROCEDURE
delete_old_postavki();

```

### 4. Оформление путевок

```

    CREATE OR REPLACE FUNCTION update_kassa_total()
    RETURNS TRIGGER AS $$
    BEGIN
        UPDATE kassa SET total = total + NEW.price;
    END;

```

```
RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER update_kassa_on_insert  
AFTER INSERT ON putevki  
FOR EACH ROW  
EXECUTE PROCEDURE update_kassa_total();
```

#### 5. Место в столовой

```
CREATE OR REPLACE FUNCTION update_stolovaya_trigger()  
RETURNS TRIGGER AS $$  
BEGIN  
UPDATE stolovaya  
SET visitor = NEW.visitor  
WHERE mesto = NEW.nomer;  
RETURN NEW;  
END; $$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER update_stolovaya_trigger BEFORE INSERT  
OR UPDATE ON registration  
FOR EACH ROW EXECUTE PROCEDURE  
update_stolovaya_trigger();
```

## ER-диаграмма

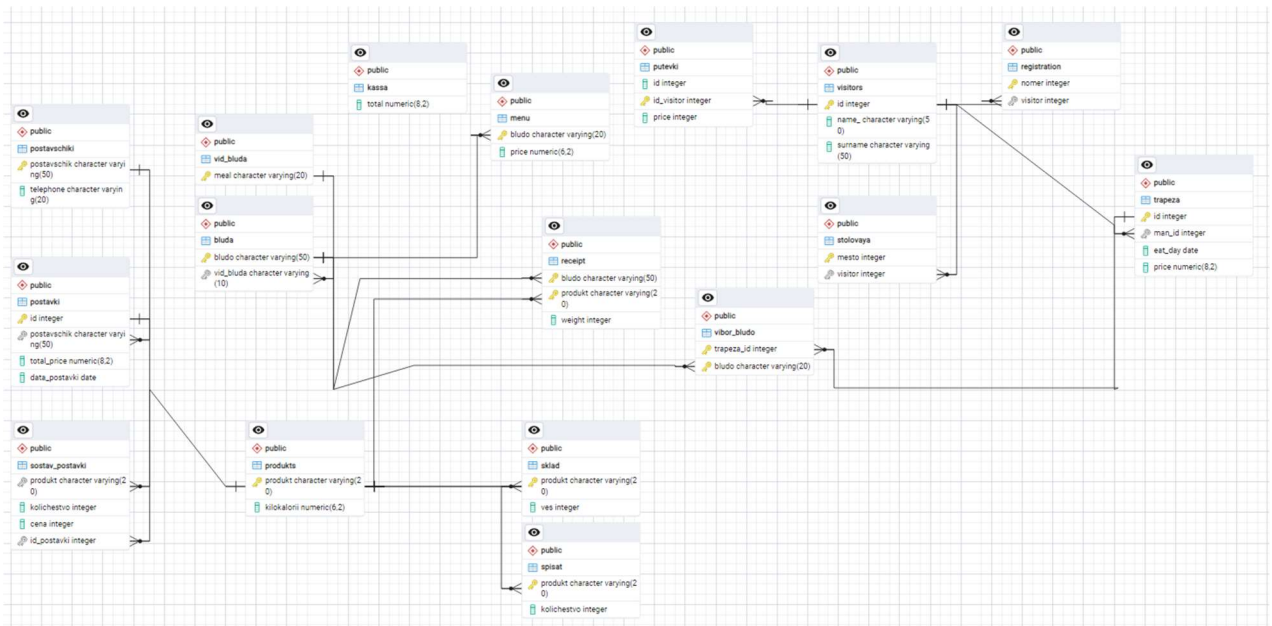


Рисунок 11 – ER-диаграмма