**Министерство образования и науки Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**“ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**УНИВЕРСИТЕТ ИТМО”**

**ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**ОТЧЕТ**

**по Лабораторной работе № 5**

**«Построение запросов»**

Специальность 09.02.07 «Информационные системы и программирование»

Дисциплина «Основы проектирования баз данных»

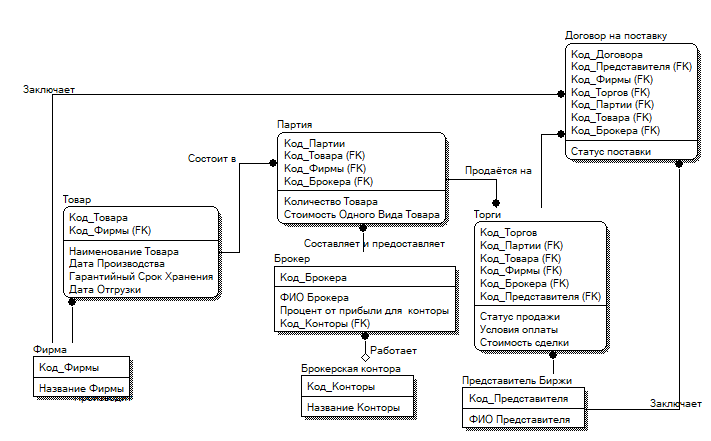
|  |  |
| --- | --- |
| Преподаватель:  Говоров А.И.\_\_\_\_\_\_\_\_\_\_  «\_\_» \_\_\_\_\_\_\_ 2020 г.  Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Выполнил:  студент группы Y2337  Примаченко А.А. |

Санкт-Петербург

2019/2020

Цель работы: овладеть практическими навыками создания и написания запросов к базам данных PostgreSQL 11, заполненных рабочими данными.

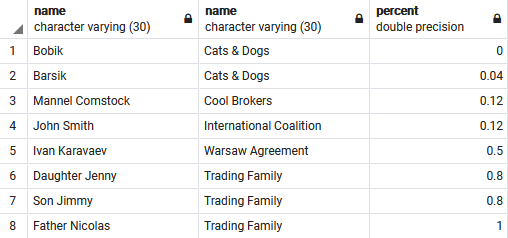
Индивидуальное задание: реализовать не менее 10 SQL-запросов на PostgreSql, следуя индивидуальному заданию.

Название создаваемой БД – Birzha. Её схема представлена на рисунке 1.

*Рисунок 1 – Схема инфологической модели данных БД, выполненная в среде CA ERwin Data Modeler.*

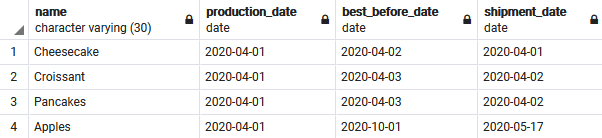
Ниже по тексту представлены запросы и соответствующие им рисунки 2-11, содержащие результат выполнения самого запроса.

Запрос 1. Вывести имена брокеров, их процент и бюро, в которых они состоят. Отсортировать результат по проценту в порядке возрастания.

select "Broker".name, "BrokersBureu".name, "Broker".percent from public."Broker", public."BrokersBureu" where "Broker".bureu\_id = "BrokersBureu".id order by "Broker".percent asc

*Рисунок 2 – Результат выполнения запроса 1*

Запрос 2. Вывести названия всех продуктов, доставленных до истечения срока годности.

select "Product".name, "Product".production\_date, "Product".best\_before\_date, "Product".shipment\_date from public."Product" where "Product".shipment\_date is not null and "Product".shipment\_date < "Product".best\_before\_date

*Рисунок 3 – Результат выполнения запроса 2*

Запрос 3. Вывести все просроченные товары, которые не были доставлены, вместе с тем временем, что они пролежали на складе на момент 17 мая 2020 года.

select "Product".name, "Product".production\_date, "Product".best\_before\_date, age("Product".best\_before\_date) from public."Product" where "Product".best\_before\_date is not null and "Product".shipment\_date is null

*Рисунок 4 – Результат выполнения запроса 3*

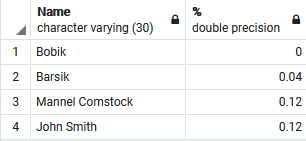
*Рисунок 4 – Результат выполнения запроса 3*

Запрос 4. Вывести имена всех представителей биржи в верхнем регистре.

SELECT upper("ExchangesAgent".name) as "Name" FROM public."ExchangesAgent"

*Рисунок 5 – Результат выполнения запроса 4*

Запрос 5. Вывести брокеров чей процент, что они отдают бюро, ниже среднего.

SELECT "Broker".name as "Name", "Broker".percent as "%" FROM public."Broker" WHERE "Broker".percent < (SELECT AVG("Broker".percent) FROM public."Broker" );

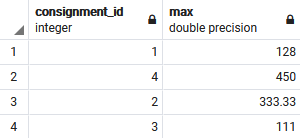
*Рисунок 6 – Результат выполнения запроса 5*

Запрос 6. Вывести наименование продукта, который стоит больше всего единиц за штуку, среди всех сделок. Также вывести эту цену.

SELECT "Product".name as "Name", (SELECT max("Product-Consignment".price\_for\_unit) FROM public."Product-Consignment") as "Max Price" FROM public."Product-Consignment", public."Product" WHERE "Product-Consignment".price\_for\_unit = (SELECT max("Product-Consignment".price\_for\_unit) FROM public."Product-Consignment") and "Product-Consignment".product\_id = "Product".id

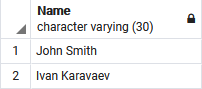
*Рисунок 7 – Результат выполнения запроса 6*

Запрос 7. Отобразить ID сделки и стоимость самого дорогого товара оттуда, но выводить нужно только те позиции, где самый дорогой товар стоит меньше 500.

SELECT "Product-Consignment".consignment\_id, max("Product-Consignment".price\_for\_unit) FROM public."Product-Consignment" GROUP BY "Product-Consignment".consignment\_id HAVING max("Product-Consignment".price\_for\_unit) < 500

*Рисунок 8 – Результат выполнения запроса 7*

Запрос 8. Вывести Представителей Биржи, у которых полностью совпадает имя с каким-либо из Брокеров, и, соответственно, наоборот.

SELECT "ExchangesAgent".name as "Name" FROM public."ExchangesAgent" WHERE "ExchangesAgent".name = ANY(SELECT "Broker".name FROM public."Broker")

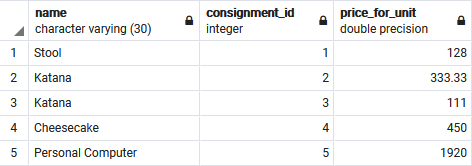
*Рисунок 9 – Результат выполнения запроса 8*

Запрос 9. Вывести всех людей занесённых в базу данных «Биржа». Представленные в базе люди – Представители Биржы и Брокеры. Стоит заметить, что в предыдущем запросе было найдено, что есть люди являющиеся и брокерами, и представителями биржи.

SELECT "ExchangesAgent".name as "Name" FROM public."ExchangesAgent" WHERE "ExchangesAgent".name not in (SELECT "Broker".name FROM public."Broker") UNION SELECT "Broker".name as "Name" FROM public."Broker"

*Рисунок 10 – Результат выполнения запроса 9*

Запрос 10. Вывести наименование самого дорогого товара из каждой сделки. Также вывести ID сделки.

SELECT "Product".name, "Product-Consignment".consignment\_id, "Product-Consignment".price\_for\_unit FROM public."Product-Consignment" LEFT JOIN "Product" ON "Product-Consignment".product\_id = "Product".id WHERE "Product-Consignment".price\_for\_unit in (SELECT MAX("Product-Consignment".price\_for\_unit) FROM public."Product-Consignment" GROUP BY "Product-Consignment".consignment\_id)

*Рисунок 11 – Результат выполнения запроса 10*

Вывод: в ходе работы было составлено десять запросов, в следствии чего были получены практические навыки составления и реализации запросов к базам данных PostrgeSQL 11.