Министерство образования и науки Российской Федерации

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,

МЕХАНИКИ И ОПТИКИ»

(УНИВЕРСИТЕТ ИТМО)

Факультет среднего профессионального образования

ОТЧЁТ

О ПРАКТИЧЕСКОЙ РАБОТЕ № 5

по теме: Анализ данных. Создание таблиц бд PostgreSQL. Заполнение таблиц рабочими данными. Создание запросов

по дисциплине: Основы проектирования баз данных

Специальность 09.02.07 Информационные системы и программирование

Проверил:

\_\_\_\_\_\_\_\_ Говоров А.И.

Дата: «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2020г.

Оценка \_\_\_\_\_\_\_\_\_\_\_

Выполнил:

студент группы Y2336

\_\_\_\_\_\_\_\_Морус Е.Л.

Санкт-Петербург 2020

постАНОВКА ЗАДАЧИ

Цель лабораторной работы №5: овладеть практическими навыками создания таблиц базы данных PostgreSQL 12, заполнения их рабочими данными, резервного копирования и восстановления баз данных.

задание

Необходимо реализовать некоторое количество SQL-запросов. Примерный набор требуемых запросов:

* выбор значений, заданных атрибутов из более, чем двух таблиц, с сортировкой – от 1 балла;
* использование условий WHERE, состоящих из более, чем одного условия – от 1 балла;
* использование функций для работы с датами – от 2 баллов;
* использование строковых функций – от 3 баллов;
* запрос с использованием подзапросов – от 2 баллов (многострочный подзапрос - от 3 баллов);
* вычисление групповой (агрегатной) функции – от 1 балла (с несколькими таблицами – от 3 баллов);
* вычисление групповой (агрегатной) функции с условием HAVING – от 2 баллов;
* использование предикатов EXISTS, ALL, SOME и ANY - от 4 баллов;
* использование запросов с операциями реляционной алгебры (объединение, пересечение и т.д.) - от 3 баллов;
* использование объединений запросов (inner join и т.д.) - от 3 баллов. Запросов должно быть не менее 10 (30 баллов на 5, 25 баллов на 4, 20 баллов на 3)(5, 4, 3 - это оценки, а не суммы баллов в журнале).

Выполнение

Схема физической модели бд, спроектированной с помощью CA Erwin Data Moduler представлена на рисунке 1.

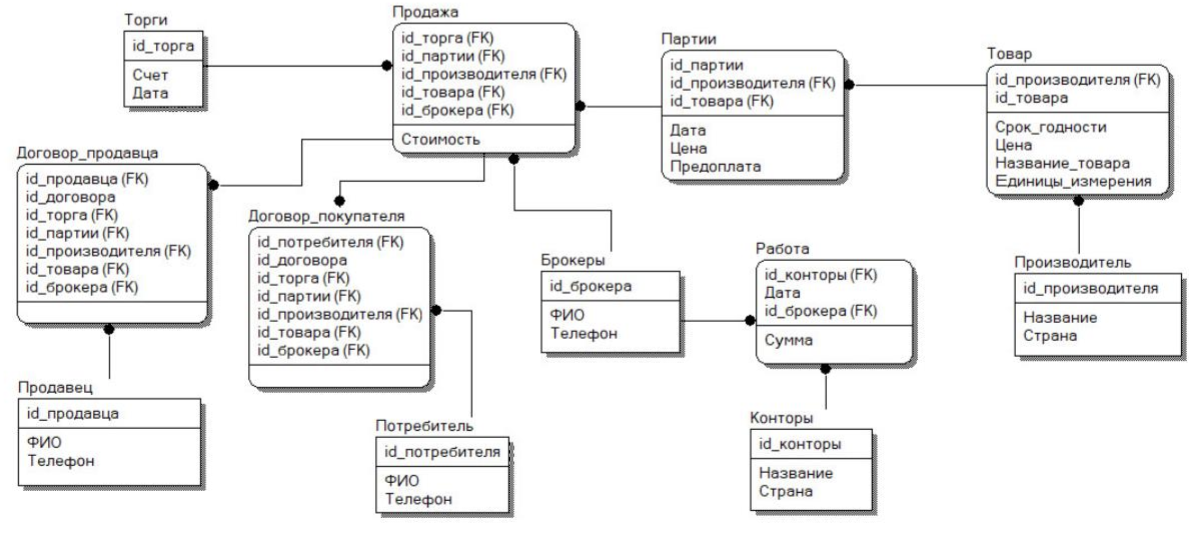


Рисунок 1 Схема информационной модели БД

Запросы

1. Найти зарплату всех брокеров заданных фирм:

SELECT "Offices"."Name", "Amount"

FROM "Work"

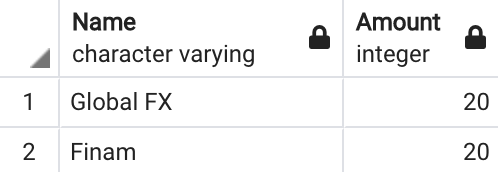
inner join "Brokers"

on "Brokers"."ID\_Offices" = "Work"."ID\_Office"

inner join "Offices"

on "Offices"."ID\_Office" = "Work"."ID\_Office"

where "Offices"."Name" IN ('Finam', 'Global FX');



2) Найти все факты выставления на продажу товара с просроченной годностью (номер партии, код товара, наименование товара, данные о брокере):

SELECT "Goods\_Name", "Name", "Phone\_Number", "Expiration\_Date", "Date\_Bidding"

FROM "Brokers"

inner join "Consignments"

on "Consignments"."ID\_Broker" = "Brokers"."ID\_Broker"

inner join "Sale"

on "Sale"."ID\_Consignment" = "Consignments"."ID\_Consignment"

inner join "Bidding"

on "Bidding"."ID\_Bidding" = "Sale"."ID\_Bidding"

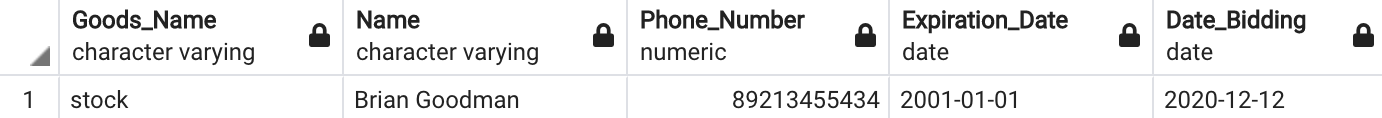
inner join "Goods\_In\_Consignments"

on "Consignments"."ID\_Consignment" = "Goods\_In\_Consignments"."ID\_Consignment"

inner join "Goods"

on "Goods\_In\_Consignments"."ID\_Good" = "Goods"."ID\_Good"

WHERE "Expiration\_Date" < "Date\_Bidding";



3) Для каждой фирмы посчитать сколько они заработали за все время:

SELECT "Offices"."Name", (

select sum("Consignments"."Sell\_Price" \* "Amount\_Goods")

from "Consignments"

inner join "Goods\_In\_Consignments"

on "Consignments"."ID\_Consignment" = "Goods\_In\_Consignments"."ID\_Consignment"

inner join "Brokers"

on "Consignments"."ID\_Broker" = "Brokers"."ID\_Broker"

where "Brokers"."ID\_Offices" = "Offices"."ID\_Office"

) as "Total revenue"

FROM "Offices";



4) Найти товары, которые никогда не выставляли на продажу брокеры заданной фирмы:

select distinct "Goods\_Name"

from "Goods"

where "Goods\_Name" not in (

select "Goods\_Name"

from "Goods"

inner join "Goods\_In\_Consignments"

on "Goods"."ID\_Good" = "Goods\_In\_Consignments"."ID\_Good"

inner join "Consignments"

on "Goods\_In\_Consignments"."ID\_Consignment" = "Consignments"."ID\_Consignment"

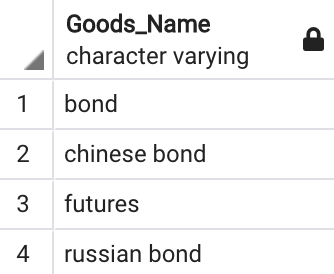
inner join "Brokers"

on "Consignments"."ID\_Broker" = "Brokers"."ID\_Broker"

inner join "Offices"

on "Offices"."ID\_Office" = "Brokers"."ID\_Offices"

where "Offices"."Name" = 'Global FX')



5) Вывести для каждого продукта вывести его количество во всех партиях:

SELECT "Goods\_Name", SUM("Goods\_In\_Consignments"."Amount\_Goods")

AS "Amount\_Goods"

FROM "Goods\_In\_Consignments"

left join "Goods"

on "Goods"."ID\_Good" = "Goods\_In\_Consignments"."ID\_Good"

left join "Consignments"

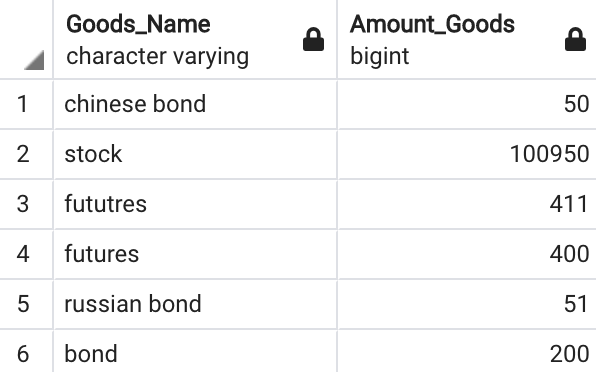
on "Goods\_In\_Consignments"."ID\_Consignment" = "Consignments"."ID\_Consignment"

left join "Sale"

on "Consignments"."ID\_Consignment" = "Sale"."ID\_Consignment"

left join "Bidding" on "Sale"."ID\_Bidding" = "Bidding"."ID\_Bidding"

AND "Date\_Bidding" < '2000-10-05' GROUP BY "Goods\_Name";



6) Для заданной страны показать все партии, в которых есть товары этих стран:

SELECT "Consignments"."ID\_Consignment", "Consignments"."Sell\_Price"

FROM public."Consignments"

where "ID\_Consignment" = any(

select "Goods\_In\_Consignments"."ID\_Consignment"

from "Firm"

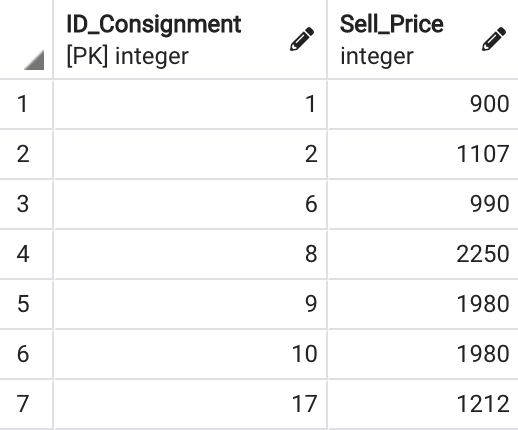
inner join "Goods"

on (public."Firm"."Firm\_Number" = "Goods"."Firm\_Number")

inner join "Goods\_In\_Consignments"

on (public."Goods"."ID\_Good" = "Goods\_In\_Consignments"."ID\_Good")

where "Country" = 'USA' );



7) Вывести информацию об аукционе и информацию о партиях, которые были проданы на этом аукционе:

SELECT "Bidding"."ID\_Bidding", "Date\_Bidding", "Manager\_Response", "Consignments"."Sell\_Price"

FROM public."Bidding"

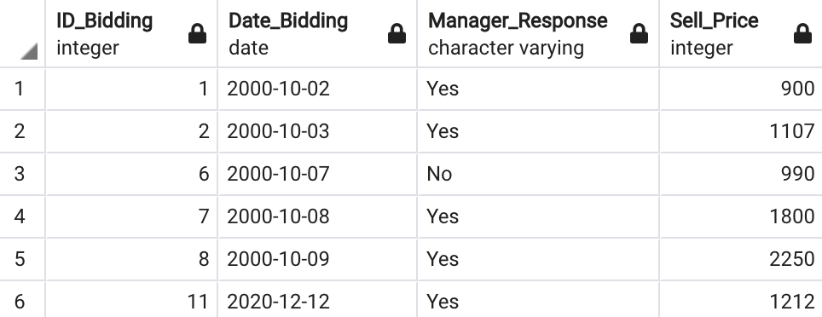
inner join "Sale"

on ("Bidding"."ID\_Bidding" = "Sale"."ID\_Bidding")

inner join "Consignments"

on ("Consignments"."ID\_Consignment" = "Sale"."ID\_Consignment")

where "Consignments"."Status" = 'Sold';



8) Брокеров, которые не продали ни одну партию:

SELECT "ID\_Broker", "Name", "Phone\_Number", "ID\_Offices"

FROM public."Brokers"

where (

select count(\*)

from "Consignments"

where "Brokers"."ID\_Broker" = "Consignments"."ID\_Broker"

) = 0;



9) Вывести партии, цена продажи которых меньше стоимость товаров в ней, и стоимость товаров:

SELECT "ID\_Consignment", "Start\_Date", "Prepayment", "Status", "ID\_Broker", "Sell\_Price", (

select sum("Price" \* "Amount\_Goods")

from "Goods"

inner join "Goods\_In\_Consignments"

on ("Goods"."ID\_Good" = "Goods\_In\_Consignments"."ID\_Good")

where "Consignments"."ID\_Consignment" = "Goods\_In\_Consignments"."ID\_Consignment"

) as "Must be price"

FROM public."Consignments"

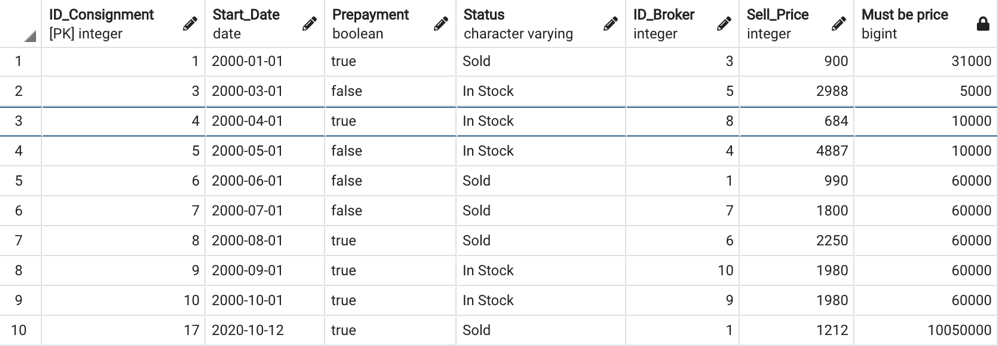
where "Sell\_Price" < (select sum("Price" \* "Amount\_Goods")

from "Goods"

inner join "Goods\_In\_Consignments" on ("Goods"."ID\_Good" = "Goods\_In\_Consignments"."ID\_Good")

where "Consignments"."ID\_Consignment" = "Goods\_In\_Consignments"."ID\_Consignment"

);



10) Вывести партии, цена продажи которых меньше стоимость товаров в ней, и стоимость товаров:

SELECT "ID\_Office", "Name", "Country", (

select count(\*)

from "Brokers"

inner join "Consignments" on ("Consignments"."ID\_Broker" = "Brokers"."ID\_Broker")

where "Brokers"."ID\_Offices" = "Offices"."ID\_Office"

) as "Goods sold"

FROM public."Offices";



Вывод

В ходе выполнения лабораторной работы были изучены SELECT-запросы и подзапросы.