Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский государственный технический университет имени Н.Э. Баумана

(национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕ	ЕТ Информ	иатика и системы упра	вления	
КАФЕДРА	Системы обр	Системы обработки информации и управления		
Отчет по лабораторной работе № 4				
«Нечеткие запросы в PostgreSQL»				
	по дисциплине «Оперативный анализ данных»			
Студент _	<u>ИУ5-51Б</u> (Группа)	(Подпись, дата)	<u>Е.И. Бирюкова</u> (И.О.Фамилия)	
Преподаватель			К.Ю. Маслеников	

Москва

(Подпись, дата)

(И.О.Фамилия)

Цель работы

Целью работы является:

- изучение подходов к реализации нечетких запросов к реляционным базам данных средствами СУБД PostgreSQL
- получение навыков адаптации реляционных баз данных к реализации нечетких запросов

Полученный вариант задания – Вариант 3

Вы работаете в ломбарде. Вашей задачей является отслеживание финансовой стороны его работы. Деятельность компании организована следующим образом: к вам обращаются различные лица с целью получения денежных средств под залог определенных товаров. У каждого из приходящих к вам клиентов вы запрашиваете фамилию, имя, отчество и другие паспортные данные. После оценивания стоимости принесенного в качестве залога товара вы определяете сумму, которую готовы выдать на руки клиенту, а также свои комиссионные. Кроме того, определяете срок возврата денег. Если клиент согласен, то ваши договоренности фиксируются в виде документа, деньги выдаются клиенту, а товар остается у вас. В случае если в указанный срок не происходит возврата денег, товар переходит в вашу собственность.

Ход лабораторной работы

- 1. Разработать физическую схему базы данных согласно варианту задания
- Client (Клиент) Хранит информацию о клиентах, которые обращаются в ломбард.
 - о ID (SERIAL, PRIMARY KEY) уникальный идентификатор клиента.
 - о LastName (VARCHAR) фамилия клиента.
 - о FirstName (VARCHAR) имя клиента.
 - о MiddleName (VARCHAR) отчество клиента.

- о PassportData (VARCHAR) паспортные данные клиента.
- Pledge (Залог) Хранит информацию о заложенных товарах.
 - о ID (SERIAL, PRIMARY KEY) уникальный идентификатор залога.
 - ClientID (INTEGER, FOREIGN KEY REFERENCES Client(ID)) идентификатор клиента, которому принадлежит залог.
 - о Item (VARCHAR) название заложенного товара.
 - о ItemValue (NUMERIC) оценочная стоимость товара.
 - о IssueAmount (NUMERIC) сумма, выданная клиенту.
 - o Commission (NUMERIC) комиссия ломбарда.
 - о ReturnDate (DATE) дата возврата товара.
 - Status (VARCHAR) статус залога (например, 'Выдан', 'Возвращен', 'Просрочен').
- History (История) Хранит информацию о всех операциях, связанных с залогами.
 - о ID (SERIAL, PRIMARY KEY) уникальный идентификатор операции.
 - PledgeID (INTEGER, FOREIGN KEY REFERENCES Pledge(ID)) идентификатор залога, к которому относится операция.
 - OperationDate (DATE) дата проведения операции.
 - о OperationType (VARCHAR) тип операции (например, 'Выдача', 'Возврат', 'Просрочка').
 - о Amount (NUMERIC) сумма, участвующая в операции.

```
Query Query History
1 -- Создание таблицы Client
3 CREATE TABLE Client (
       ID SERIAL PRIMARY KEY.
       LastName VARCHAR(255),
       FirstName VARCHAR(255)
       MiddleName VARCHAR(255)
       PassportData VARCHAR(255)
10
11
12 -- Создание таблицы Pledge
14 CREATE TABLE Pledge
       ID SERIAL PRIMARY KEY.
       ClientID INTEGER REFERENCES Client(ID),
       Item VARCHAR(255),
18
       ItemValue NUMERIC(10, 2),
       IssueAmount NUMERIC(10, 2),
19
       Commission NUMERIC(10, 2),
       ReturnDate DATE,
21
22
       Status VARCHAR(20)
23 );
26 -- Создание таблицы History
27
28 CREATE TABLE History
      ID SERIAL PRIMARY KEY,
       PledgeID INTEGER REFERENCES Pledge(ID),
31
       OperationDate DATE,
       OperationType VARCHAR(20),
       Amount NUMERIC(10, 2)
```

Заполнение таблиц данными

```
37 -- Вставка данных в таблицу Client
38 INSERT INTO Client (LastName, FirstName, MiddleName, PassportData) VALUES
   ('Иванов', 'Иван', 'Иванович', '1234 567890'),
40 ('Петров', 'Петр', 'Петрович', '9876 543210'),
41 ('Сидоров', 'Сидор', 'Сидорович', '1011 121314'),
42 ('Кузнецов', 'Кузьма', 'Кузьмич', '2222 333344'),
43 ('Смирнов', 'Сергей', 'Сергеевич', '5555 666677'),
44 ('Васильев', 'Василий', 'Васильевич', '8888 999900'),
45 ('Соколов', 'Софрон', 'Софронович', '1111 222233'),
46 ('Михайлов', 'Михаил', 'Михайлович', '4444 555566'),
47 ('Федоров', 'Федор', 'Федорович', '7777 888899'),
48 ('Попов', 'Павел', 'Павлович', '0000 111122');
49
50 -- Вставка данных в таблицу Pledge
51 INSERT INTO Pledge (ClientID, Item, ItemValue, IssueAmount, Commission, ReturnDate, Status) VALUES
52 (1, 'Золотые часы', 5000, 4000, 500, '2023-12-31', 'Выдан'),
53 (2, 'Ноутбук', 15000, 12000, 1000, '2024-01-15', 'Выдан'),
54 (3, 'Телевизор', 8000, 6000, 400, '2023-11-20', 'Возвращен'),
<sup>55</sup> (1, 'Кольцо с бриллиантом', 12000, 9000, 800, '2024-02-10', 'Выдан'),
<sup>56</sup> (2, 'Смартфон', 3000, 2500, 200, '2023-12-20', 'Выдан'),
57 (3, 'Фотоаппарат', 7000, 5000, 300, '2023-11-25', 'Возвращен'),
<sup>58</sup> (1, 'Серебряная цепочка', 2000, 1500, 100, '2024-01-05', 'Выдан'),
<sup>59</sup> (2, 'Планшет', 10000, 8000, 500, '2024-02-01', 'Выдан'),
60 (3, 'Видеокамера', 12000, 9000, 600, '2023-12-10', 'Выдан'),
61 (1, 'Золотая подвеска', 4000, 3000, 200, '2024-03-15', 'Выдан');
63 -- Вставка данных в таблицу History
64 INSERT INTO History (PledgeID, OperationDate, OperationType, Amount) VALUES
<sup>65</sup> (1, '2023-11-10', 'Выдача', 4000),
66 (2, '2023-12-01', 'Выдача', 12000),
67 /2 12022 11 1ET IDURANT COMA
Data Output Messages Notifications
TNSFRT 0 10
```

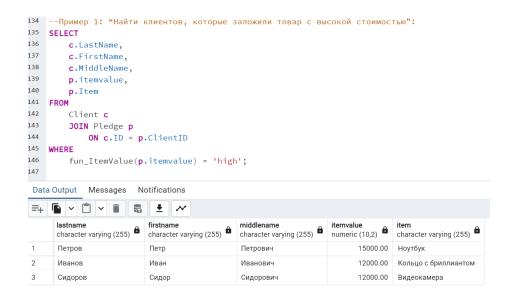
2. Предложить для физической схемы базы данных лингвистические переменные (не менее трех).

- Стоимость товара (ItemValue):
 - о Диапазон: 0 20000
 - о Лингвистические значения:
 - Низкая (Low): 0 4000
 - Средняя (Medium): 4000 10000
 - Высокая (High): 10000 20000
- Сумма выдачи (IssueAmount):
 - о Диапазон: 0 15000
 - о Лингвистические значения:
 - Малая (Small): 0 2500
 - Средняя (Medium): 2500 7500
 - Большая (Large): 7500 15000
- Срок возврата (ReturnDate):
 - Диапазон: 0 365 (в днях)
 - о Лингвистические значения:
 - Короткий (Short): 0 30 (дней)
 - Средний (Medium): 30 90 (дней)
 - Длинный (Long): 90 365 (дней)
- 3. Разработать SQL-реализации функций принадлежности и лингвистических переменных.

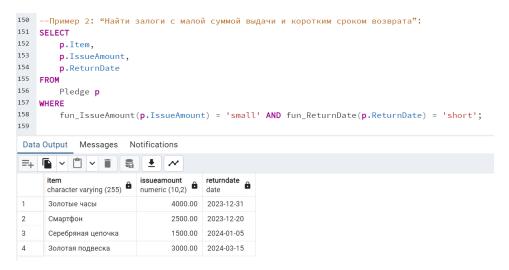
```
85 -- Функция принадлежности для "Стоимость товара (ItemValue)"
87 CREATE FUNCTION fun_ItemValue(cur_cost numeric)
88 RETURNS VARCHAR AS $$
89▼ BEGIN
90₩
     IF cur cost < 4000 THEN
91
          RETURN 'low';
     ELSIF cur_cost BETWEEN 4000 AND 10000 THEN
92
93
        RETURN 'medium';
94
95
          RETURN 'high';
     END IF;
96
97 END;
98 $$ LANGUAGE plpgsql;
99
100
101 -- Функция принадлежности для "Сумма выдачи (IssueAmount)"
102 --/*
103 CREATE OR REPLACE FUNCTION fun_IssueAmount(cur_amount numeric)
104 RETURNS VARCHAR AS $$
105▼ BEGIN
106▼ IF cur_amount < 2500 THEN
107
          RETURN 'small';
108
     ELSIF cur_amount BETWEEN 2500 AND 7500 THEN
109
       RETURN 'medium';
110
111
        RETURN 'large';
112
     END IF;
113 END;
114 $$ LANGUAGE plpgsql;
   117 -- Функция принадлежности для "Срок возврата (ReturnDate)"
   118 --/*
   119 CREATE FUNCTION fun_ReturnDate(cur_date date)
   120 RETURNS VARCHAR AS $$
   121▼ BEGIN
   122▼ IF cur_date - CURRENT_DATE <= 30 THEN
   123
              RETURN 'short';
   124
          ELSIF cur_date - CURRENT_DATE BETWEEN 31 AND 90 THEN
   125
              RETURN 'medium';
   126
          ELSE
   127
               RETURN 'long';
   128
          END IF;
   129 END;
   130 $$ LANGUAGE plpgsql;
   131 --*/
```

4. На основе SQL реализации лингвистических переменных разработать нечеткие запросы к базе данных (не менее пяти).

Пример 1: "Найти клиентов, которые заложили товар с высокой стоимостью"

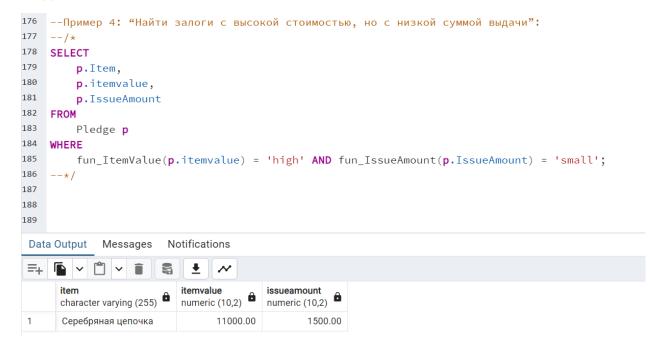


Пример 2: "Найти залоги с малой суммой выдачи и коротким сроком возврата"

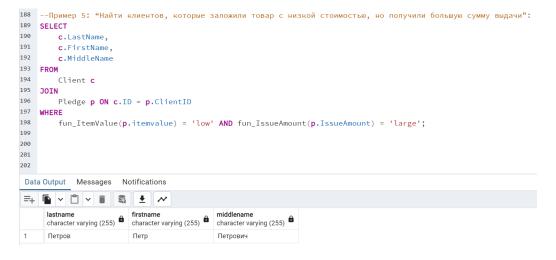


Пример 3: "Найти клиентов, которые заложили товар с высокой стоимостью и получили большую сумму выдачи"

Пример 4: "Найти залоги с высокой стоимостью, но с низкой суммой выдачи"

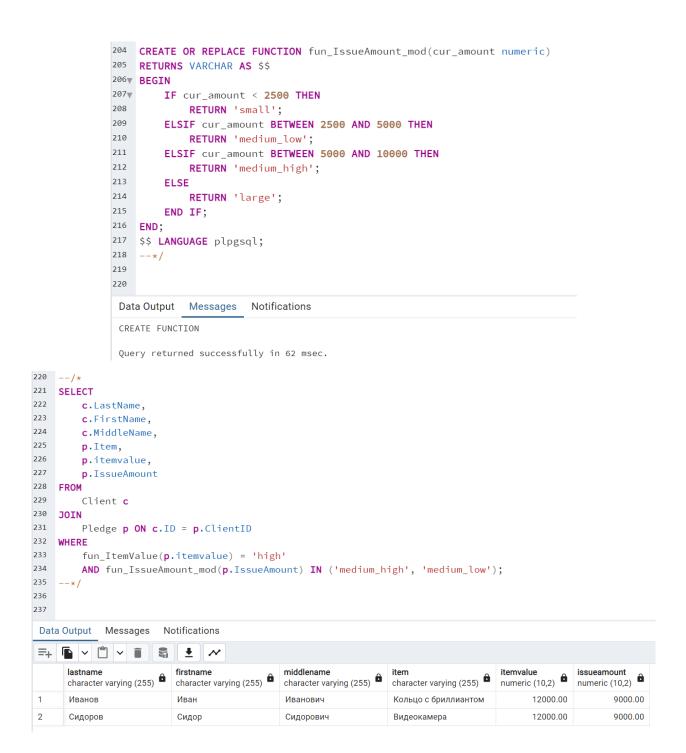


Пример 5: "Найти клиентов, которые заложили товар с низкой стоимостью, но получили большую сумму выдачи"



5. На основе SQL реализации модификации лингвистических переменных разработать нечеткие запросы к базе данных

Найти клиентов, которые заложили товар с высокой стоимостью (ItemValue), но получили сумму выдачи (IssueAmount), которая является "не очень большой" и "не очень маленькой", то есть не больше 10000, но и не меньше 2500.



6. Выбрать параметр для индекса соответствия (CI), разработать нечеткие запросы к базе данных (не менее пяти) с учетом индекса соответствия. В качестве параметра CI можно выбрать IssueAmount.

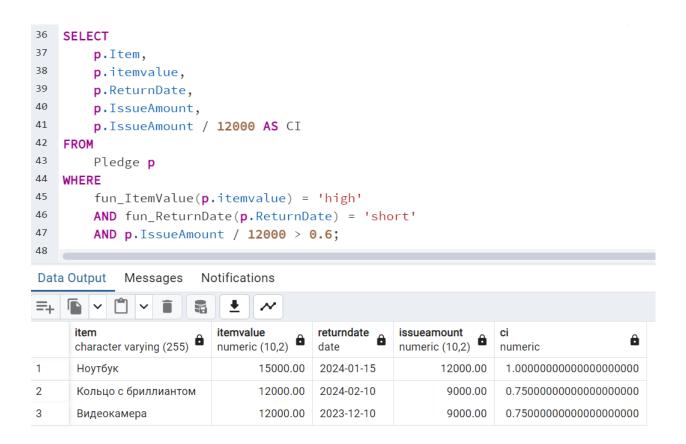
Пример 1: "Найти клиентов, у которых уровень соответствия высокой стоимости товара и большой суммы выдачи выше 0.8"



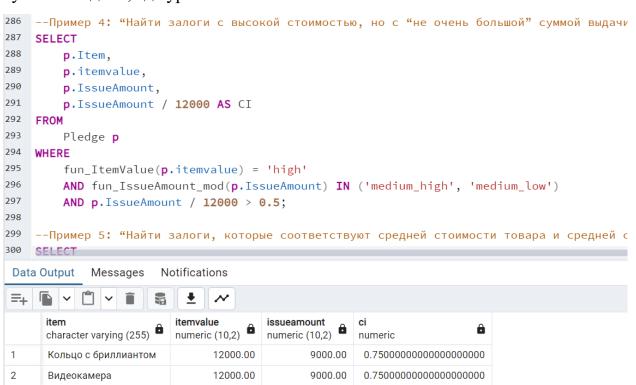
Пример 2: "Найти клиентов, у которых уровень соответствия низкой стоимости товара и маленькой суммы выдачи выше 0.7"



Пример 3: "Найти залоги с высокой стоимостью и коротким сроком возврата, где уровень соответствия выше 0.6"



Пример 4: "Найти залоги с высокой стоимостью, но с "не очень большой" суммой выдачи, где уровень соответствия выше 0.5"



Пример 5: "Найти залоги, которые соответствуют средней стоимости товара и средней сумме выдачи, где уровень соответствия выше 0.4"

