МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГАОУ ВО «СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт информационных технологий

ОТЧЕТ

о выполнении практической работы № 1  
««Язык SQL. Манипулирование базой данных. Реляционная алгебра и язык SQL»

по дисциплине   
«Управление данными»

Выполнил:

ст.гр.ИС/б-23-1-о

Смирнов Б.В.

Проверила:

Ст. пр. Сикорская Я.Е.

Севастополь, 2025

**Цель работы:**

Научиться анализировать предметную область с целью создания схемы БД, учитывая ссылочную целостность набора. Изучить основы реляционной алгебры, как базового средства манипулирования. Выработать у обучающихся практические навыки по работе с реляционными базами данных и представлению запросов как на языке реляционной алгебре, так и SQL.

**Ход работы:**

Вариант - 15: База данных растений и их уходе

Растения (id\_растения, название, тип, семейство, стоимость, место\_происхождения)

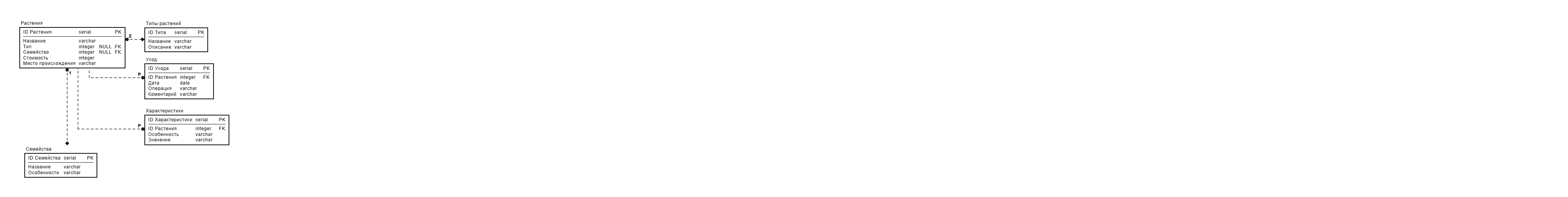
Типы\_растений (id\_типа, название, описание)

Семейства (id\_семейства, название, особенности)

Характеристики (id\_характеристики, id\_растения, особенность, значение)

Уход (id\_ухода, id\_растения, дата, операция, комментарий)

Для создания БД в СУБД была создана ее схема в нотации IDEFIX (Рисунок 1).

  
Рисунок 1 – Схема базы данных в нотации IDEFIX

Запрос 1:

Деление и соединение (и проекция для уменьшения изображения) (Рисунок 2)

Описание:

SQL:

WITH required\_ops(op) AS (

VALUES

('Watering'),

('Fertilizing'),

('Pruning')

)

SELECT

p.plantName,

t.TplantName,

f.familyName

FROM Plants p

JOIN Tplant t

ON p.plantType = t.TplantID

JOIN Families f

ON p.plantFamily = f.familyID

WHERE NOT EXISTS (

SELECT 1

FROM required\_ops ro

WHERE NOT EXISTS (

SELECT 1

FROM CareTakes c

WHERE c.plantID = p.plantID

AND c.operation = ro.op

)

);

РА:

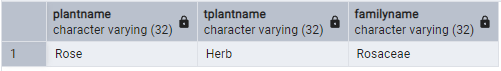


Рисунок 2 – Запрос с делением и соединением

Запрос 2:

Проекция и деление (Рисунок 3)

Описание:

SQL:

WITH required\_ops(op) AS (

VALUES

('Watering'),

('Fertilizing'),

('Pruning')

)

SELECT

p.plantName,

f.familyName

FROM Plants AS p

JOIN Families AS f

ON p.plantFamily = f.familyID

WHERE NOT EXISTS (

SELECT 1

FROM required\_ops AS ro

WHERE NOT EXISTS (

SELECT 1

FROM CareTakes AS c

WHERE c.plantID = p.plantID

AND c.operation = ro.op

)

);

РА:



Рисунок 3 – Запрос с проекцией и делением

Запрос 3:

Операции вычитания и дизъюнкции

Описание:

SQL:

SELECT \*

FROM Tplant AS Tp

WHERE NOT EXISTS(

SELECT 1

FROM plants

WHERE Tp.TplantID = plantType

)

UNION

SELECT \*

FROM Tplant

WHERE TplantID = 3

РА:

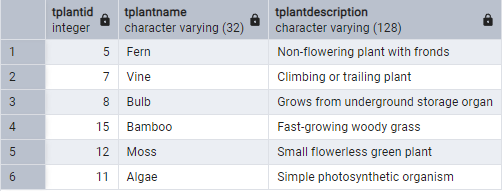


Рисунок 4 – Запрос с дизъюнкцией и вычитанием

Запрос 4:

Использование операции проекции, объединения и конъюнкции

Описание:

SQL:

SELECT DISTINCT plants.plantID

FROM plants

JOIN tplant ON tplant.tplantID = plants.plantType

INTERSECT

SELECT DISTINCT plantID

FROM stats

WHERE plantID IN(1, 3, 5, 6)

РА:

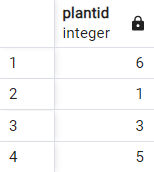


Рисунок 5 – Запрос с дизъюнкцией и вычитанием

Запрос 5:

Использование селекции с соединением в одном запросе

Описание:

SQL:

SELECT plants.plantID, plants.plantName, Tplant.TplantName,

Tplant.TplantDescription

FROM plants

JOIN Tplant ON plants.plantType = Tplant.TplantID

WHERE Tplant.TplantName = 'Tree'

РА:

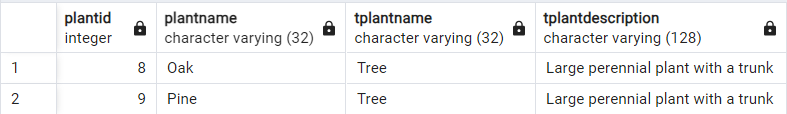


Рисунок 6 – Запрос с селекцией и соединением

Запрос 6:

Простые вычисления

Описание:

SQL:

SELECT p1.plantcost - p2.plantcost AS easy\_equation

FROM plants AS p1, plants AS p2

WHERE p1.plantID = 1 AND p2.plantID = 3

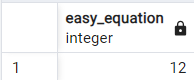


Рисунок 7 – Отношение, полученное запросом с простыми вычислениями

Запрос 7:

Использование простого вычисления как параметр агрегатной функции

Описание:

SQL:

SELECT AVG(p1.plantcost - p2.plantcost) AS easy\_equation\_AVG

FROM plants AS p1, plants AS p2

WHERE p1.plantID = 1 AND p2.plantID = 3

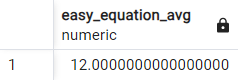


Рисунок 8 – Отношение, полученное запросом с использованием простого вычисления как параметр агрегатной функции

Запрос 8:

Использование группировки GROUP BY

Описание:

SQL:

SELECT AVG(plantcost), plantType

FROM plants

GROUP BY plantType

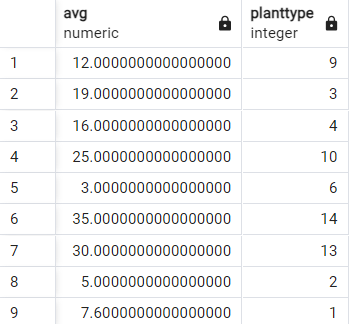


Рисунок 9 – Отношение, полученное запросом с предложением Group By

Запрос 9:

Использование предложения HAVING

Описание:

SQL:

SELECT AVG(plantcost), plantType

FROM plants

GROUP BY plantType

HAVING AVG(plantcost) > 25

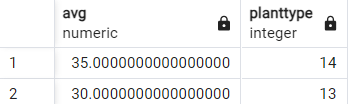


Рисунок 10 – Отношение, полученное запросом с предложением Having

Запрос 10:

Проверка ограничения целостности (при удалении типа растения замена на NULL значение в основной таблице)

Описание:

SQL:

DELETE FROM Tplant WHERE TplantID = 1;

SELECT \* FROM plants

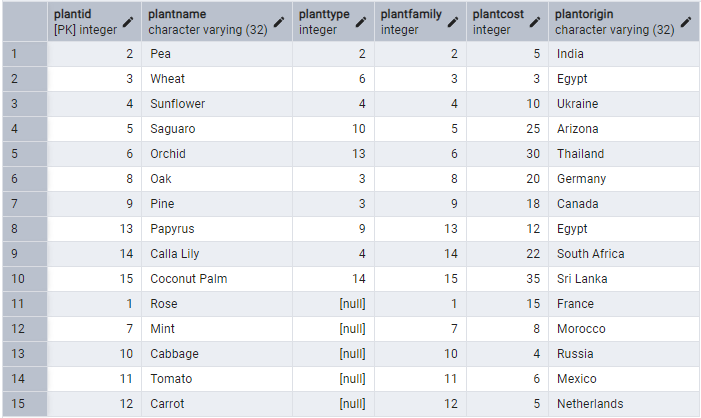


Рисунок 11 – отношение, полученное запросом после замены типа на NULL ввиду удаления данных из другой таблицы

Запрос 11:

Проверка на каскадирование удаления

Описание:

SQL:

DELETE FROM Families WHERE FamilyID = 1;

SELECT \* FROM plants

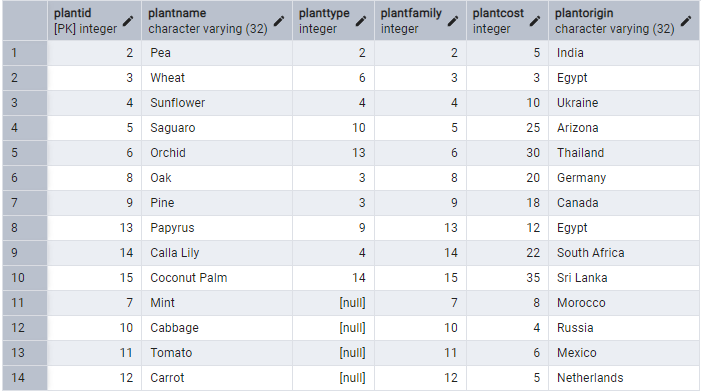


Рисунок 12 – Отношение, полученное после каскадного удаления

Запрос 12:

Проверка на каскадирование изменений

Описание:

SQL:

UPDATE Tplant SET TplantID = 100 WHERE TplantID = 3;

SELECT \* FROM plants

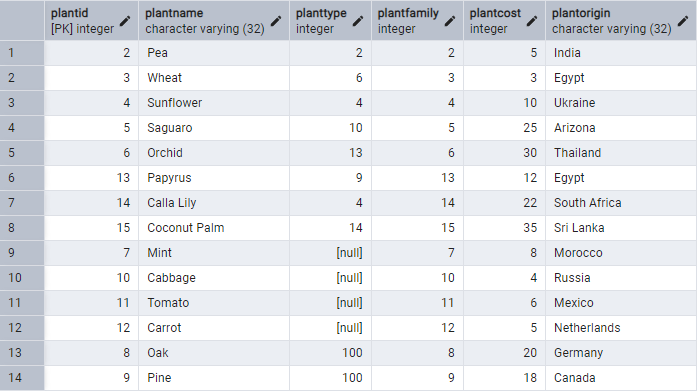


Рисунок 13 – Отношение, приведённое каскадированием изменённых данных в другой таблице

Вывод:

В ходе выполнения лабораторной работы были изучены операции реляционной алгебры и способы их интерпретации на языке SQL, используя СУБД PostgreSQL были рассмотрены такие операции как: деление, вычитание, соединение, проекция, объединение, селекция, дизъюнкция и конъюнкция, а также их комбинации. Для данной работы была разработана база данных, а также создана ее схема в нотации IDEFIX, для проверки всех операций в базу были занесены тестовые данные, напоминающие реальные данные.