**ЛАБОРАТОРНАЯ РАБОТА №1**

**«Язык SQL. Манипулирование базой данных. Запросы на основе нескольких таблиц»**

**1.1 Цель работы**

Научиться анализировать предметную область с целью создания схемы БД, учитывая ссылочную целостность набора. Изучить способы получения информации из нескольких таблиц, способы выполнения и принцип действия рекурсивных запросов и научится использовать вложенные подзапросы.

**1.2 Индивидуальный вариант**

Вариант 15: База данных растений и их ухода:

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);

**1.3 Ход выполнения работы**

1.3.1 В начале выполнения работы было проанализировано описание БД по варианту. База данных ведет учет по растениям, для этого требуются следующие данные:

1. Название растения идентифицирует растение.

2. Тип растений определяет разделение их на группы в соответствии с их общими и уникальными характеристиками.

3. Семейства растений определяет их разделение в зависимости от строения цветка, дает понимание об уходе за ними.

3. Стоимость растений фиксирует цену для его продажи.

4. Место происхождения дает определённую информацию, необходимую для ухода за растением.

5. Особенности растения и их значения так же необходимы для построения схемы ухода и отслеживания его нормального состояния.

6. Дата операции позволяет зафиксировать выполнение необходимых действий.

7. Название операции и комментарий раскрывает смысл проведенной операции.

База данных состоит из 5 сущностей, их свойства уже были выделены в описании варианта. Далее были выделены и классифицированы существующие связи:

1. Связь Типы\_растений – Растения

Связь типа 1:М, так как много растений могут быть одного и того же вида, связь является не идентифицирующей, так как сущности независимы между собой. Отношение Растения связано с отношением Типы\_растений внешним ключом (id\_типа).

2. Связь Семейства – Растения

Связь типа 1:М, так как много растений могут быть одного и того же семейства, связь является не идентифицирующей, так как сущности независимы между собой. Отношение Растения связано с отношением Семейства внешним ключом (id\_семейства).

3. Связь Растения – Уход

Связь типа 1:М, так как у растения может быть много операций, связанных с уходом, связь является идентифицирующей, так как сущность Уход зависит от сущности Растения. Отношение Растения связано с отношением Уход внешним ключом (id\_растения), который входит в состав первичного ключа сущности Уход.

4. Связь Растения – Характеристики

Связь типа 1:М, так как у растения может быть много пунктов характеристики, связь является идентифицирующей, так как сущность Характеристики зависит от сущности Растения. Отношение Растения связано с отношением Характеристики внешним ключом (id\_растения), который входит в состав первичного ключа сущности Характеристики.

Далее были выделены необходимые ограничения целостности. Их первичных ключей были выделены следующие атрибуты: id\_растения в таблице Растения, id\_типа в таблице Типы\_растений, id\_семейства в таблице Семейства, id\_характеристики в таблице Характеристики, id\_ухода в таблице Уход.

Ключи, которые было решено сделать внешними: id\_типа в таблице Растения ссылается на id\_типа в таблице Типы\_растений, id\_семейства в таблице Растения ссылается на id\_семейства в таблице Семейства, id\_растения в таблице Характеристики ссылается на id\_растения в таблице Растения и входит в состав составного ключа сущности Характеристики, id\_растения в таблице Уход ссылается на id\_растения в таблице Растения и входит в состав составного ключа сущности Уход.

Ограничения уникальности атрибута было наложено на название в таблице Типы\_растений и название в связи Семейства, также для атрибута стоимость была сделана проверка на ввод числа больше 0.

На основе анализа описания БД была построена схема в нотации IDEF1X, что показано на рисунке 1.1.

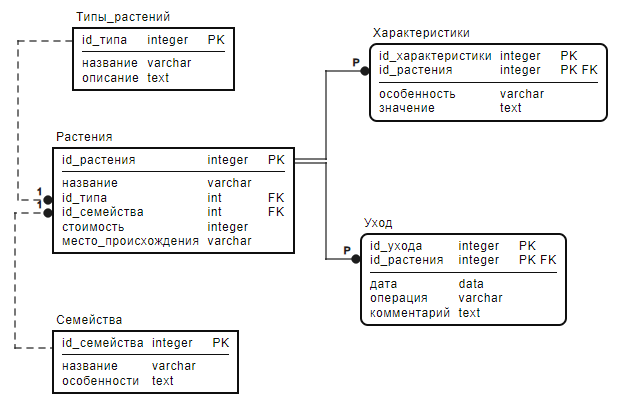


Рисунок 1.1 – Схема базы данных растений и их ухода

1.3.2 Далее с использованием PostgreSQL были созданы все таблицы, с необходимыми ограничениями целостности. Запрос для создания таблицы Типы\_растений отражен в листинге 1.1, где атрибут id\_типа является первичным ключом. Листинг 1.2 показывает создание ограничения уникальности для атрибута название таблицы Типы\_растений.

Листинг 1.1 – Запрос для создания таблицы Типы\_растений

create table Типы\_растений (

id\_типа serial primary key,

название varchar,

описание text

);

Листинг 1.2 –Создание ограничения таблицы Типы\_растений

alter table Типы\_растений add constraint типы\_уникальное\_название unique (название)

Запрос на создание таблицы Семейства отображен в листинге 1.3, где атрибут id\_семейства является первичным ключом, а листинг 1.4 содержит запрос на создание ограничения уникальности атрибута название таблицы Семейства.

Листинг 1.3 – Запрос для создания таблицы Семейства

create table Семейства (

id\_семейства serial primary key,

название varchar,

особенности text,

);

Листинг 1.4 – Создание ограничения таблицы Семейства

alter table Семейства add constraint типы\_уникальное\_название unique (название)

Создание таблицы Растения показано в листинге 1.5, содержащем запрос. Таблица содержит первичный ключ id\_растения и атрибуты id\_типа, id\_семейства с ограничением NOT NULL , которые являются внешними ключами, также атрибут стоимость содержит ограничение на ввод положительного числа больше 0 в конструкции CHECK.

Листинг 1.5 – Запрос для создания таблицы Растения

create table Растения (

id\_растения serial primary key,

id\_типа int not null,

id\_семейства int not null,

название varchar,

стоимость int check (стоимость > 0),

место\_происхождения varchar,

constraint fk\_растения\_типы foreign key (id\_типа) references Типы\_растений(id\_типа) on delete no action on update cascade,

constraint fk\_растения\_семейства foreign key (id\_семейства) references Семейства(id\_семейства) on delete no action on update cascade

);

Листинг 1.6 содержит запрос для создания таблицы Характеристики. Первичный ключ является составным из атрибутов id\_характеристики, и внешнего ключа id\_растения.

Листинг 1.6 – Запрос для создания таблицы Характеристики

create table Характеристики(

id\_характеристики serial not null,

id\_растения int not null,

особенность varchar,

значение text,

primary key (id\_характеристики, id\_растения),

constraint fk\_характ\_растения foreign key (id\_растения) references Растения(id\_растения) on delete cascade on update cascade

);

Запрос для создания таблицы Уход отражен в листинге 1.7. Первичный ключ является составным из атрибутов id\_ухода, и внешнего ключа id\_растения.

Листинг 1.7 – Запрос для создания таблицы Уход

create table Уход (

id\_ухода serial not null,

id\_растения int not null,

дата date,

операция varchar,

комментарий text,

primary key (id\_ухода, id\_растения),

constraint fk\_растения\_уход foreign key (id\_растения) references Растения(id\_растения) on delete cascade on update cascade

);

1.3.3 Созданные ранее таблицы были заполнены данными. Листинг 1.8 содержит запрос на заполнение данными таблицы Семейства 15 записями. На рисунке 1.2 показана итоговая заполненная таблица.

Листинг 1.8 – Запрос на заполнение данными таблицы Семейства

INSERT INTO Семейства (название, особенности) VALUES

('Астровые', 'Одно из крупнейших семейств цветковых растений, включающее ромашки, подсолнечники и астры'),

('Орхидные', 'Экзотические растения с уникальными цветами, такие как орхидеи'),

('Бобовые', 'Семейство, включающее горох, фасоль и люцерну'),

('Крестоцветные', 'Семейство, включающее капусту, брокколи и горчицу'),

('Паслёновые', 'Семейство, включающее картофель, томаты и баклажаны'),

('Розоцветные', 'Семейство, включающее розы, яблони и малины'),

('Зонтичные', 'Семейство, включающее морковь, сельдерей и укроп'),

('Лилейные', 'Семейство, включающее лилии, тюльпаны и чеснок'),

('Мятликовые', 'Семейство, включающее пшеницу, кукурузу и бамбук'),

('Гречишные', 'Семейство, включающее гречиху и щавель'),

('Мальвовые', 'Семейство, включающее хлопок, гибискус и мальву'),

('Кипарисовые', 'Семейство, включающее кипарисы, можжевельники и туи'),

('Берёзовые', 'Семейство, включающее берёзы, ольхи и лещины'),

('Сосновые', 'Семейство, включающее сосны, ели и кедры'),

('Лавровые', 'Семейство, включающее лавр, авокадо и корицу');

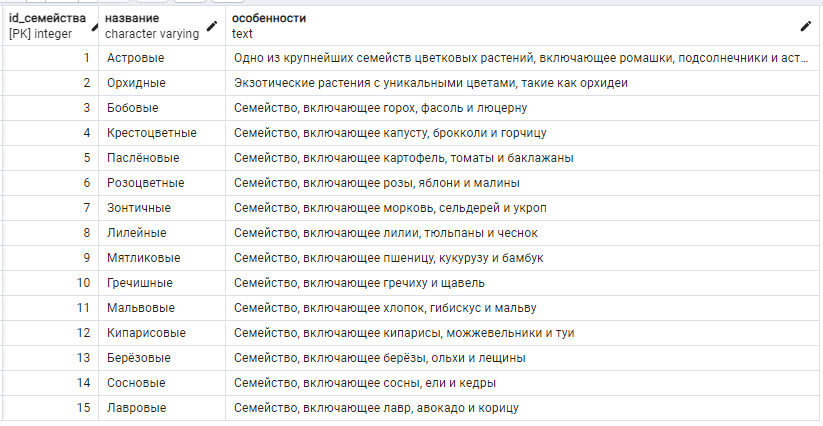


Рисунок 1.2 – Таблица Семейства

Запрос на заполнение данными таблицы 15 записями Типы\_растений показан в листинге 1.9. Рисунок 1.3 показывает заполненную таблицу Типы\_растений

Листинг 1.9 – Запрос на заполнение данными таблицы Типы\_растений

INSERT INTO Типы\_растений (название, описание) VALUES

('Мхи', 'Небольшие, нецветковые растения, часто встречающиеся в влажных местах.'),

('Плауны', 'Древние сосудистые растения с мелкими листьями, напоминающими иголки.'),

('Хвощи', 'Растения с полыми, сегментированными стеблями, часто растущие в болотистых местах.'),

('Папоротники', 'Растения с крупными, перистыми листьями, размножающиеся спорами.'),

('Голосеменные', 'Растения, производящие семена, не заключенные в плоды, такие как сосны и ели.'),

('Однодольные', 'Цветковые растения с одним семядолем, включающие лилии, орхидеи и злаки.'),

('Двудольные', 'Цветковые растения с двумя семядолями, включающие розы, дубы и подсолнечники.'),

('Зелёные водоросли', 'Водные растения, обладающие хлорофиллом, часто встречающиеся в пресной воде.'),

('Бурые водоросли', 'Морские водоросли, включающие крупные виды, такие как ламинария.'),

('Красные водоросли', 'Морские водоросли, известные своими красными пигментами и использованием в пищевой промышленности.'),

('Цианобактерии', 'Фотосинтезирующие бактерии, часто называемые сине-зелеными водорослями.'),

('Печёночники', 'Небольшие, нецветковые растения, часто встречающиеся на влажных почвах и камнях.'),

('Саговники', 'Древние, медленно растущие растения с крупными, перистыми листьями.'),

('Гинкговые', 'Единственный современный представитель — гинкго двулопастный, известный своими веерообразными листьями.'),

('Хвойные', 'Древесные растения с игольчатыми листьями, включающие сосны, ели и кедры.');

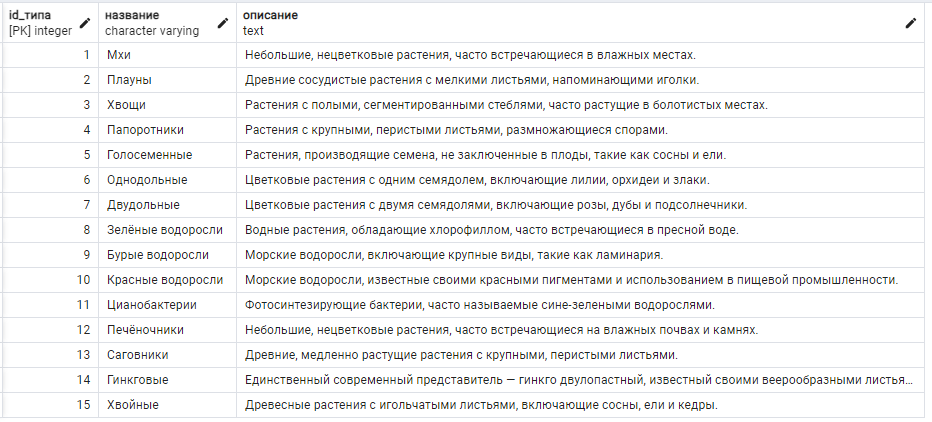


Рисунок 1.3 – Таблица Типы\_растений

Листинг 1.10 содержит запрос на запись 15 строк в таблицу Растения. На рисунке 1.4 видно заполненную запросом таблицу.

Листинг 1.10 – Запрос на заполнение данными таблицы Растения

INSERT INTO Растения (id\_типа, id\_семейства, название, стоимость, место\_происхождения) VALUES

(7, 1, 'Ромашка', 100, 'Россия'),

(6, 2, 'Орхидея', 500, 'Таиланд'),

(7, 3, 'Горох', 50, 'Китай'),

(7, 4, 'Капуста', 30, 'Германия'),

(6, 5, 'Томат', 70, 'Италия'),

(12, 9, 'Маршанция', 200, 'Франция'),

(7, 7, 'Морковь', 40, 'США'),

(6, 8, 'Лилия', 150, 'Нидерланды'),

(6, 9, 'Пшеница', 20, 'Канада'),

(7, 10, 'Гречиха', 60, 'Россия'),

(7, 11, 'Хлопчатник', 80, 'Индия'),

(15, 12, 'Кипарис', 300, 'Греция'),

(6, 13, 'Берёза', 90, 'Россия'),

(15, 14, 'Сосна', 120, 'Швеция'),

(7, 15, 'Лавр', 110, 'Испания');



Рисунок 1.4 – Таблица Растения

Внесение 45 записей в таблицу Характеристики показано в листинге 1.11. Рисунок 1.5 демонстрирует заполненную запросом таблицу.

Листинг 1.11 – Запрос на заполнение данными таблицы Характеристики

INSERT INTO характеристики (id\_растения, особенность, значение) VALUES

(1, 'Цвет', 'Белый'),

(1, 'Аромат', 'Сладкий'),

(1, 'Высота', '50 см'),

(2, 'Цвет', 'Фиолетовый'),

(2, 'Аромат', 'Легкий'),

(2, 'Высота', '60 см'),

(3, 'Цвет', 'Зеленый'),

(3, 'Плоды', 'Стручки'),

(3, 'Высота', '30 см'),

(4, 'Цвет', 'Зеленый'),

(4, 'Плоды', 'Кочан'),

(4, 'Высота', '40 см'),

(5, 'Цвет', 'Красный'),

(5, 'Плоды', 'Ягоды'),

(5, 'Высота', '70 см'),

(6, 'Цвет', 'Зеленый'),

(6, 'Листья', 'Крупные'),

(6, 'Высота', '20 см'),

(7, 'Цвет', 'Оранжевый'),

(7, 'Плоды', 'Корнеплод'),

(7, 'Высота', '30 см'),

(8, 'Цвет', 'Белый'),

(8, 'Аромат', 'Сильный'),

(8, 'Высота', '50 см'),

(9, 'Цвет', 'Зеленый'),

(9, 'Плоды', 'Зерна'),

(9, 'Высота', '100 см'),

(10, 'Цвет', 'Зеленый'),

(10, 'Плоды', 'Зерна'),

(10, 'Высота', '80 см'),

(11, 'Цвет', 'Белый'),

(11, 'Плоды', 'Коробочки');

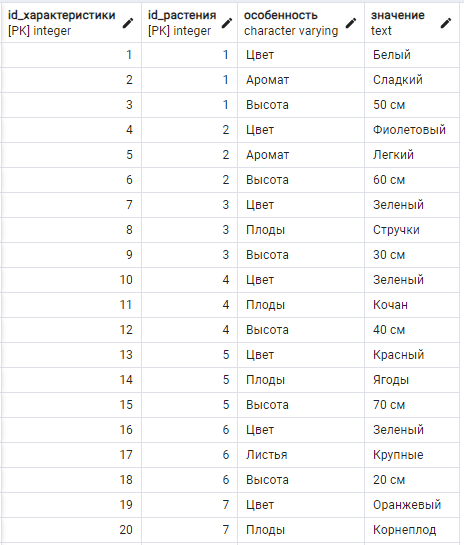


Рисунок 1.5 – Таблица Характеристики

Далее было внесено 30 записей в таблицу Уход, запрос показан в листинге 1.12. На рисунке 1.6 показан результат внесенных данных в виде заполненной таблицы.

Листинг 1.12 – Запрос на заполнение данными таблицы Уход

INSERT INTO уход (id\_растения, дата, операция, комментарий) VALUES

(1, '2024-01-10', 'Полив', 'Полив теплой водой'),

(1, '2024-01-20', 'Подкормка', 'Удобрение органическое'),

(2, '2024-02-05', 'Обрезка', 'Удаление сухих листьев'),

(2, '2024-02-15', 'Полив', 'Полив дождевой водой'),

(3, '2024-03-01', 'Подкормка', 'Минеральное удобрение'),

(3, '2024-03-10', 'Полив', 'Полив теплой водой'),

(4, '2024-03-20', 'Обрезка', 'Формирование кроны'),

(4, '2024-03-30', 'Полив', 'Полив дождевой водой'),

(5, '2024-04-05', 'Подкормка', 'Органическое удобрение'),

(5, '2024-04-15', 'Полив', 'Полив теплой водой'),

(6, '2024-04-25', 'Обрезка', 'Удаление сухих листьев'),

(6, '2024-05-05', 'Полив', 'Полив дождевой водой'),

(7, '2024-05-15', 'Подкормка', 'Минеральное удобрение'),

(7, '2024-05-25', 'Полив', 'Полив теплой водой'),

(8, '2024-06-05', 'Обрезка', 'Формирование кроны'),

(8, '2024-06-15', 'Полив', 'Полив дождевой водой'),

(9, '2024-06-25', 'Подкормка', 'Органическое удобрение'),

(9, '2024-07-05', 'Полив', 'Полив теплой водой'),

(10, '2024-07-15', 'Обрезка', 'Удаление сухих листьев'),

(10, '2024-07-25', 'Полив', 'Полив дождевой водой'),

(11, '2024-08-05', 'Подкормка', 'Минеральное удобрение'),

(11, '2024-08-15', 'Полив', 'Полив теплой водой'),

(12, '2024-08-25', 'Обрезка', 'Формирование кроны'),

(12, '2024-09-05', 'Полив', 'Полив дождевой водой'),

(13, '2024-09-15', 'Подкормка', 'Органическое удобрение'),

(13, '2024-09-25', 'Полив', 'Полив теплой водой'),

(14, '2024-10-05', 'Обрезка', 'Удаление сухих листьев'),

(14, '2024-10-15', 'Полив', 'Полив дождевой водой'),

(15, '2024-10-25', 'Подкормка', 'Минеральное удобрение');

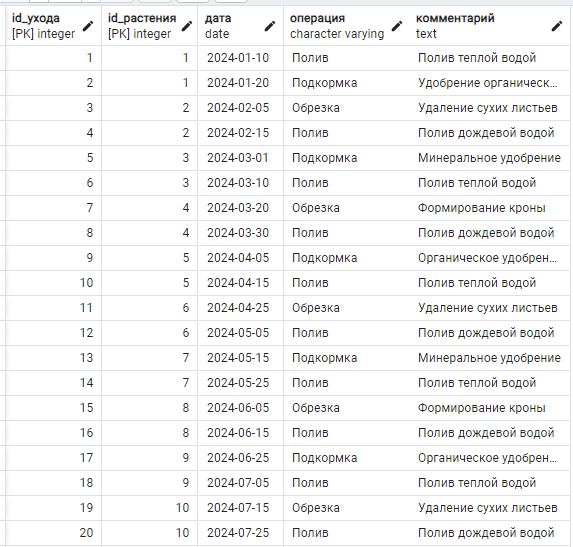


Рисунок 1.6 – Таблица Уход

1.3.4 Была выполнена проверка ограничений целостности, добавленных при создании таблиц. При попытке добавить в таблицу Типы\_растений строку, содержащей уже существующее значение атрибута название, появляется ошибка, связанная с ограничением UNIQUE (рисунок 1.7).

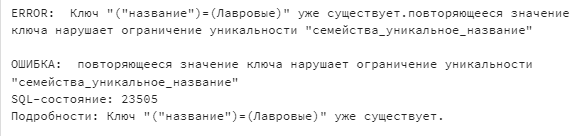


Рисунок 1.7 – Ошибка при проверке ограничения UNIQUE таблицы Типы\_растений

При попытке добавить в таблицу Семейства строку с существующим названием, тоже появляется идентичная ошибка из-за ограничения UNIQUE, что показано на рисунке 1.8.

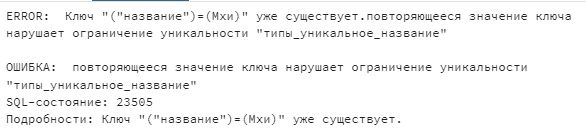


Рисунок 1.8 – Ошибка при проверке ограничения UNIQUE таблицы Семейства

Далее было проверена работа ограничения CHECK атрибута стоимость таблицы Растения. На рисунке 1.9 показано, что при стоимости равной 0, внести строку не удастся из-за наложенного ограничения.

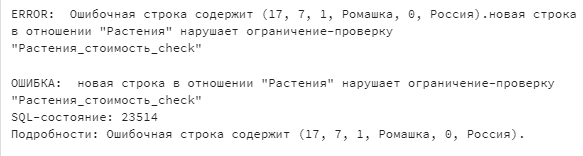


Рисунок 1.9 – Ошибка при проверке ограничения CHECK таблицы Семейства

При попытке удалить запись из таблицы Семейства, внешний ключ которой используется в таблице Растения, появляется ошибка из-за поставленного ограничения ON DELETE NO ACTION, что показано на рисунке 1.10.

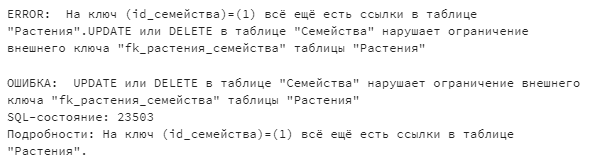


Рисунок 1.10 – Ошибка при проверке удаления записи таблицы Семейства

При модификации атрибута id\_растения таблицы Растения происходят соответствующие модификации в таблице Характеристики, связанные с ограничением ON UPDATE CASCADE. Строки таблицы Характеристики до изменения id\_растения показаны на рисунке 1.11, а после изменения – на рисунке 1.12. При удалении строки из таблицы Растения удалятся и соответствующие ей записи из таблиц Уход и Характеристики из-за ограничения ON DELETE CASCADE.

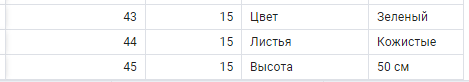


Рисунок 1.11 – Строки таблицы Характеристики до изменения

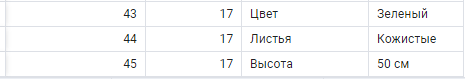


Рисунок 1.12 – Строки таблицы Характеристики после изменения

1.3.5 Далее представлены запросы на выборку из базы данных. Запрос, содержащий простые вычисления: необходимо вывести список всех растений и их цену с учетом налога 20%. Текст запроса показан в листинге 1.13, результат выполнения на рисунке 1.13.

Листинг 1.13 ­– Запрос, содержащий простые вычисления

select id\_растения, название, стоимость, стоимость\*1.2 as стоимость\_с\_налогом from Растения;

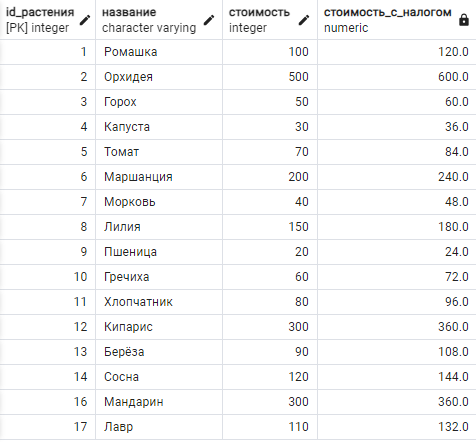


Рисунок 1.13 – Результат вывода запроса с простыми вычислениями

Запрос, содержащий простые вычисления как параметр агрегатной функции: вычислить среднюю стоимость типов растений с учетом налога. Текст запроса показан в листинге 1.14, результат выполнения на рисунке 1.14.

Листинг 1.14 ­– Запрос, содержащий простые вычисления в агрегатной функции

select id\_типа, avg(стоимость \* 1.2) as средняя\_стоимость\_с\_налогом from Растения group by id\_типа;

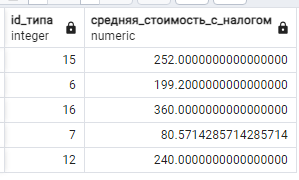


Рисунок 1.14 – Результат вывода запроса с простыми вычислениями в агрегатной функции

Запрос, содержащий предложение GROUP BY: узнать количество операций по уходу за каждым растением. Текст запроса показан в листинге 1.15, результат выполнения на рисунке 1.15.

Листинг 1.15 ­– Запрос, содержащий предложение GROUP BY

select id\_растения, count(\*) as количество\_операций from Уход group by id\_растения;

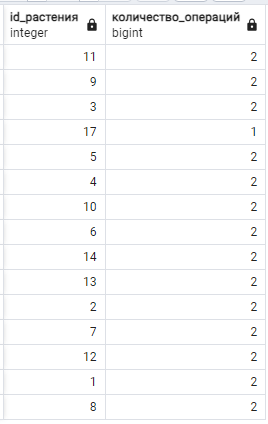


Рисунок 1.15 – Результат вывода запроса с предложением GROUP BY

Запрос, содержащий предложение HAVING: вывести id растений с сортировкой по возрастанию, имеющих более 2 характеристик. Текст запроса показан в листинге 1.16, результат выполнения на рисунке 1.16.

Листинг 1.16 ­– Запрос, содержащий предложение HAVING

select id\_растения, count(\*) as количествр\_характеристик from Характеристики group by id\_растения having count(\*)>2 order by id\_растения asc;

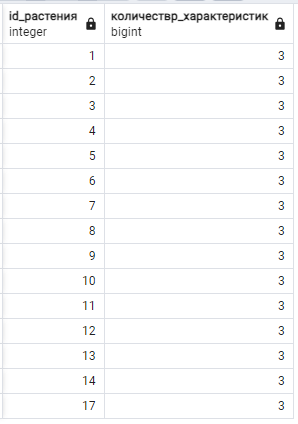


Рисунок 1.16 – Результат вывода запроса с предложением HAVING

Запрос, соединяющий две таблицы с помощью JOIN: вывести информацию об операциях по уходу за орхидеей. Текст запроса показан в листинге 1.17, результат выполнения на рисунке 1.17.

Листинг 1.17 ­– Запрос, соединяющий две таблицы с помощью JOIN

select название, дата, операция from Растения join Уход on Растения.id\_растения=Уход.id\_растения where название='Орхидея';

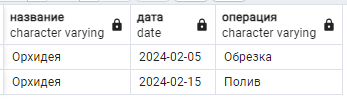


Рисунок 1.17 – Результат вывода запроса, соединяющего две таблицы с помощью JOIN

Запрос, соединяющий две таблицы: вывести информацию об особенностях растений из семейств лавровых и зонтичных. Текст запроса показан в листинге 1.18, результат выполнения на рисунке 1.18.

Листинг 1.18 ­– Запрос, соединяющий две таблицы

select Растения.название, особенности from Растения, Семейства where Растения.id\_семейства=Семейства.id\_семейства and Семейства.название in ('Лавровые', 'Зонтичные');

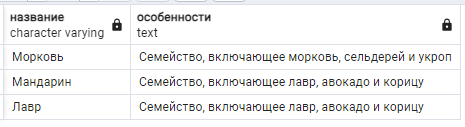


Рисунок 1.18 – Результат вывода запроса, соединяющего две таблицы

Запрос, соединяющий больше, чем две таблицы с помощью JOIN: вывести информацию о семействах и типах растений, которые заканчиваются на букву «А». Текст запроса показан в листинге 1.19, результат выполнения на рисунке 1.19.

Листинг 1.19 ­– Запрос, соединяющий больше, чем две таблицы с помощью JOIN

select Растения.название, Семейства.название, Типы\_растений.название from Растения

join Семейства on Растения.id\_семейства=Семейства.id\_семейства

join Типы\_растений on Растения.id\_типа=Типы\_растений.id\_типа

where Растения.название like '%а';

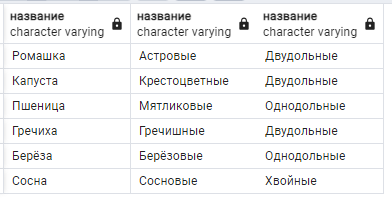


Рисунок 1.19 – Результат вывода запроса, соединяющего больше, чем две таблицы с помощью JOIN

Запрос, соединяющий больше, чем две таблицы: вывести информацию о типах растений, у которых цвет – одна из особенностей. Текст запроса показан в листинге 1.20, результат выполнения на рисунке 1.20.

Листинг 1.20 ­– Запрос, соединяющий больше, чем две таблицы

select Растения.название, Типы\_растений.название,Характеристики.значение

from Растения, Типы\_растений, Характеристики

where Растения.id\_типа=Типы\_растений.id\_типа and Растения.id\_растения=Характеристики.id\_растения

and Характеристики.особенность='Цвет';

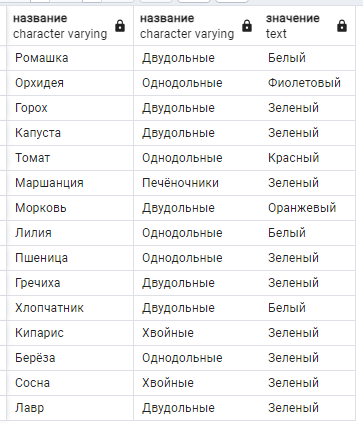


Рисунок 1.20 – Результат вывода запроса, соединяющего больше, чем две таблицы

**Выводы**

В ходе выполнения работы была проанализирована предметная область, связанная с классификацией растений. Были выделены и классифицированы все существующие связи, определены необходимые ограничения целостности. Также была построена схема базы данных в нотации IDEF1X. С помощью PostrgreSQL были написаны запросы на создание необходимых таблиц, создание определенных ранее ограничений. Таблицы были заполнены с использованием соотвествующего оператора. Далее была протестирована работа ограничений целостности. Были написаны запросы на выборку данных с использованием простых вычислений, агрегатных функций, предложений GROUP BY и HAVING. Также были рассмотрены способы соединения двух и более таблиц с использованием JOIN и без него. В конце выполнения лабораторной работы блы написан отчет.