|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА − Российский технологический университет»**  **РТУ МИРЭА** |

**Институт информационных технологий (ИИТ)**

**Кафедра цифровой трансформации (ЦТ)**

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 1**

по дисциплине «Разработка баз данных»

|  |  |  |  |
| --- | --- | --- | --- |
| Студент группы ИНБО-01-17 | *ИНБО-12-23. Албахтин И.В.* | (подпись) | |
| Ассистент | *Брайловский А.В.* | (подпись) | |
|  |  | |  | |

Москва 2025 г.

ПРАКТИЧЕСКАЯ РАБОТА №1. ОСНОВЫ DDL И ЗАПРОСЫ НА ВЫБОРКУ ДАННЫХ В POSTGRES PRO

**Цель**:

Формирование и закрепление у студентов фундаментальных навыков работы с реляционными базами данных на примере СУБД Postgres Pro. По завершении работы студент должен уметь:

• Сформировать практический навык определения структуры базы данных с использованием языка определения данных DDL (Data Definition Language).

• Научиться применять ограничения целостности (PRIMARY KEY, FOREIGN KEY, CHECK, UNIQUE, NOT NULL) для реализации бизнесправил и обеспечения консистентности (согласованности) данных, основываясь на теоретических положениях реляционной модели.

• Освоить составление SQL-запросов на выборку данных с использованием расширенного синтаксиса инструкции SELECT, включая выражения в списке выборки, псевдонимы и фильтрацию дубликатов с помощью DISTINCT.

• Развить умение применять разнообразные условия фильтрации записей в предложении WHERE, охватывая логические операции, проверку принадлежности диапазону и множеству, сравнение с шаблоном и корректную проверку на NULL.

• Получить базовые навыки агрегации данных с использованием GROUP BY и агрегатных функций, а также научиться корректно фильтровать агрегированные результаты с помощью предложения HAVING.

**Постановка задачи**:

1. На основе логической модели данных, созданной в прошлом семестре, письменно описать не менее 5 различных бизнес-правил и не менее 3 ограничений целостности для таблиц. Выбор бизнес-правил и ограничений целостности производится на усмотрение студента. Результаты представить в виде таблицы.

2. С использованием DDL-оператора CREATE TABLE создать все необходимые таблицы (согласно созданной в прошлом семестре логической модели данных) в СУБД Postgres Pro, корректно реализовав все описанные ограничения целостности.

3. Заполнить созданные таблицы согласованными тестовыми данными (не менее 5-7 записей на таблицу, где это применимо) с помощью оператора INSERT INTO.

4. Составить и выполнить не менее 6 SQL-запросов к таблицам, иллюстрирующих использование различных операторов SELECT и WHERE, согласно перечню, указанному в задании (см. Ход выполнения работы). В составленных запросах должны быть использованы все приведённые примеры.

5. Составить и выполнить по два SQL-запроса к таблицам для демонстрации работы предложений ORDER BY, GROUP BY и HAVING.

6. Каждый SQL-запрос сопроводить комментарием, объясняющим его назначение и логику работы.

**ВЫПОЛНЕНИЕ ПРАКТИЧЕСКОЙ РАБОТЫ**

Таблица 1. Пример описания ограничений для таблицы maintenance (ТО автомобиля)

|  |  |  |  |
| --- | --- | --- | --- |
| **Название**  **столбца** | **Тип данных** | **Ограничение** | **Обоснование (Бизнес-правило)** |
| maintenance\_id | SERIAL | PRIMARY KEY | Уникальный идентификатор записи о техническом обслуживании, генерируется автоматически |
| car\_id | INTEGER | FOREGEN KEY | Ссылка на таблицу car. Каждое обслуживание связано с конкретным автомобилем |
| worker\_id | INTEGER | FOREIGN KEY | Ссылка на таблицу worker. Обслуживание выполняется конкретным сотрудником |
| part\_id | INTEGER | FOREIGN KEY | Ссылка на таблицу part. Указывается деталь, которая была заменена/обслужена |
| start\_date | DATE | NOT NULL | Дата начала обслуживания обязательна для фиксации хронологии работ |
| |  | | --- | | end\_date |  |  | | --- | |  | | DATE | |  | | --- | |  |   CHECK (end\_date IS NULL OR end\_date >= start\_date) | Дата окончания должна быть больше или равна дате начала. Может быть NULL, если работа не завершена |
| status | VARCHAR(20) | DEFAULT 'planned', CHECK (status IN ('planned','in progress','completed','waiting')) | Статус указывает состояние обслуживания. По умолчанию «planned»; может быть только одно из допустимых значений |

1. **Создание структуры данных**

На этом этапе вы преобразуете спроектированные на Шаге 1 правила в SQL-код. Для создания таблиц используется оператор CREATE TABLE.

**Создание таблицы** maintenance.

**Определяются столбцы**: maintenance\_id - первичный ключ (SERIAL), car\_id(INTEGER) – внешний ключ на таблицу car,

worker\_id(INTEGER) – внешний ключ на таблицу worker, part\_id(INTEGER) – внешний ключ на таблицу part, start\_date(DATA) – дата начала обслуживания автомобиля, end\_date(DATA) – дата окончания обслуживания автомобиля.

В рамках практической работы была построена БД в соответствии с поставленным вариантом.

**set** **search**.**path** **to** albakhtin\_iv;

**CREATE** **TABLE** work\_type (

work\_type\_id **integer** **NOT** **NULL** **PRIMARY** **KEY**,

name **varchar**(500),

description **text**

);

**CREATE** **TABLE** car (

car\_id **integer** **NOT** **NULL** **PRIMARY** **KEY**,

client\_id **integer**,

brand **varchar**(30),

model **varchar**(30),

year **integer**,

license\_plate **varchar**(17)

);

**CREATE** **TABLE** supplier (

supplier\_id **integer** **NOT** **NULL** **PRIMARY** **KEY**,

name **varchar**(300),

phone **varchar**(15)

);

**CREATE** **TABLE** warranty (

warranty\_id serial **NOT** **NULL** **PRIMARY** **KEY**,

maintenance\_id **integer**,

expiry\_date **date**

);

**CREATE** **TABLE** warehouse (

warehouse\_id **integer** **NOT** **NULL** **PRIMARY** **KEY**,

address **varchar**(100)

);

**CREATE** **TABLE** part\_warehouse (

part\_id **integer** **NOT** **NULL** **PRIMARY** **KEY**,

warehouse\_id **integer**,

quantity **integer**

);

**CREATE** **TABLE** maintenance (

maintenance\_id serial **NOT** **NULL** **PRIMARY** **KEY**,

car\_id **integer**,

worker\_id **integer**,

part\_id **integer**,

start\_date **date**,

end\_date **date**,

status **varchar**(20)

);

**CREATE** **TABLE** review (

review\_id serial **NOT** **NULL** **PRIMARY** **KEY**,

client\_id **integer**,

text **varchar**(1000),

rating **integer**,

date **date**

);

**CREATE** **TABLE** client (

client\_id **integer** **NOT** **NULL** **PRIMARY** **KEY**,

**name** **varchar**(15),

surname **varchar**(15),

phone **varchar**(15),

email **varchar**(254)

);

**CREATE** **TABLE** invoice (

invoice\_id serial **NOT** **NULL** **PRIMARY** **KEY**,

maintenance\_id **integer**,

total\_amount **numeric**,

payment\_status **varchar**(20)

);

**CREATE** **TABLE** part (

part\_id **integer** **NOT** **NULL** **PRIMARY** **KEY**,

name **varchar**(100),

price **numeric**,

supplier\_id **integer**

);

**CREATE** **TABLE** maintenance\_work (

maintenance\_work serial **NOT** **NULL** **PRIMARY** **KEY**,

maintenance\_id **integer**,

work\_type\_id **integer**

);

**CREATE** **TABLE** worker (

worker\_id **integer** **NOT** **NULL** **PRIMARY** **KEY**,

name **varchar**(15),

position **varchar**(20),

phone **varchar**(15)

);

**CREATE** **TABLE** diagnosis (

diagnosis\_id serial **NOT** **NULL** **PRIMARY** **KEY**,

maintenance\_id **integer**,

result **text**

);

**ALTER** **TABLE** maintenance

**ALTER** **COLUMN** status **SET** **DEFAULT** 'planned';

**ALTER** **TABLE** maintenance

**ADD** **CONSTRAINT** chk\_dates **CHECK** (end\_date **IS** **NULL** **OR** end\_date >= start\_date);

Листинг 1 – создание таблиц

Далее заполним таблицу синтетическими данными (рис. 2). Для заполнения была выбрана таблица “product”

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 1 – заполненная таблица product

Теперь попробуем извлечь данные из некоторых столбцов таблиц.

-- 1. Все записи о техобслуживании

SELECT \*

FROM maintenance;

-- 2. Даты начала и статусы с псевдонимами

SELECT start\_date AS "Дата начала", status AS "Статус"

FROM maintenance;

-- 3. ID машины и длительность ремонта (кол-во дней)

SELECT

car\_id AS "Машина",

(end\_date - start\_date) AS "Длительность (дней)"

FROM maintenance

WHERE end\_date IS NOT NULL;

-- 4. Все уникальные статусы обслуживания

SELECT DISTINCT status

FROM maintenance;

-- 5. Работы, длившиеся больше 7 дней

SELECT maintenance\_id, car\_id, start\_date, end\_date

FROM maintenance

WHERE end\_date IS NOT NULL

AND (end\_date - start\_date) > 7;

-- 6. Работы, начавшиеся в сентябре 2025

SELECT maintenance\_id, start\_date

FROM maintenance

WHERE start\_date BETWEEN '2025-09-01' AND '2025-09-30';

-- 7. Работы, которые выполняли работники 2 или 3

SELECT maintenance\_id, worker\_id, status

FROM maintenance

WHERE worker\_id IN (2, 3);

-- 8. Работы со статусом, начинающимся на 'c' (например completed)

SELECT maintenance\_id, status

FROM maintenance

WHERE status LIKE 'c%';

-- 9. Работы без даты окончания (ещё не завершены)

SELECT \*

FROM maintenance

WHERE end\_date IS NULL;

-- 10. Работы по дате начала (от новых к старым)

SELECT maintenance\_id, car\_id, start\_date

FROM maintenance

ORDER BY start\_date DESC;

-- 11. Работы по дате окончания (от самых ранних)

SELECT maintenance\_id, car\_id, end\_date

FROM maintenance

ORDER BY end\_date ASC;

-- 12. Средняя длительность работ по каждому работнику

SELECT worker\_id, AVG(end\_date - start\_date) AS avg\_days

FROM maintenance

WHERE end\_date IS NOT NULL

GROUP BY worker\_id;

-- 13. Минимальная и максимальная длительность работ по каждой машине

SELECT car\_id, MIN(end\_date - start\_date) AS min\_days, MAX(end\_date - start\_date) AS max\_days

FROM maintenance

WHERE end\_date IS NOT NULL

GROUP BY car\_id;

-- 14. Работники, которые выполняли больше 2 работ

SELECT worker\_id, COUNT(\*) AS total\_jobs

FROM maintenance

GROUP BY worker\_id

HAVING COUNT(\*) > 2;

-- 15. Машины, у которых средняя длительность обслуживания больше 5 дней

SELECT car\_id, AVG(end\_date - start\_date) AS avg\_days

FROM maintenance

WHERE end\_date IS NOT NULL

GROUP BY car\_id

HAVING AVG(end\_date - start\_date) > 5;

Листинг 2 – Составление запросов на выборку

Результаты выполнения запросов представлены на рисунках ниже.

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 2 - Все записи о техобслуживании

Изображение выглядит как текст, снимок экрана, программное обеспечение, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 3 - Даты начала и статусы с псевдонимами

Изображение выглядит как текст, снимок экрана, программное обеспечение, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 4 - ID машины и длительность ремонта (кол-во дней)

Изображение выглядит как текст, снимок экрана, линия, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 5 - Все уникальные статусы обслуживания

Изображение выглядит как текст, снимок экрана, линия, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 6 - Работы, длившиеся больше 7 дней

Изображение выглядит как текст, снимок экрана, программное обеспечение, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 7 - Работы, начавшиеся в сентябре 2025

Изображение выглядит как текст, снимок экрана, программное обеспечение, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 8 - Работы, которые выполняли работники 2 или 3

Изображение выглядит как текст, снимок экрана, программное обеспечение, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 9 - Работы со статусом, начинающимся на 'c' (например completed)

Изображение выглядит как текст, снимок экрана, программное обеспечение, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 10 - Работы без даты окончания (ещё не завершены)

Изображение выглядит как текст, снимок экрана, программное обеспечение, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 11 - Работы по дате начала (от новых к старым)

Изображение выглядит как текст, снимок экрана, программное обеспечение, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 12 - Работы по дате окончания (от самых ранних)

Изображение выглядит как текст, снимок экрана, линия, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 13 - Cредняя длительность работ по каждому работнику

Изображение выглядит как текст, снимок экрана, линия, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 14 - Минимальная и максимальная длительность работ по каждой машине

Изображение выглядит как текст, снимок экрана, линия, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 15 - Работники, которые выполняли больше 2 работ

Изображение выглядит как текст, снимок экрана, программное обеспечение, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 16 - Машины, у которых средняя длительность обслуживания больше 5 дней

**ВЫВОД**

В ходе работы спроектирована и создана таблица с применением ограничений целостности, что обеспечило корректность и согласованность данных. Были вставлены тестовые записи и выполнены разнообразные SQL-запросы с фильтрацией, сортировкой, агрегацией и группировкой, что позволило закрепить навыки практической работы с Postgres Pro.