Государственный Университет Молдовы
Факультет Математики и Информатики
Департамент Информатики

# Индивидуальная работа
По курсу «Базы данных»
тема: “База данных для управления информацией в дилерских центрах Mercedes-Benz”

Выполнил:
студент группы I2302
Slavov Constantin

Проверил преподаватель:
Карчева Н.Ф.

Кишинёв, 2024

## Цель работы:

Создание реляционной базы данных «База данных для управления информацией в дилерских центрах Mercedes-Benz» в приложении “Oracle APEX” предназначено для управления данными о продажах, клиентах, доступных автомобилях и услугах, предоставляемых в автосалонах Mercedes-Benz. База данных необходима для учёта всех операций, включая регистрацию продаж, хранение информации о доступных автомобилях в автосалонах, клиентов, а также о предоставляемых услугах. Она также позволяет анализировать выручку по каждому автосалону и проводить инвентаризацию.

1. Все отношения находятся в нормальной форме.
2. Структура базы данных включает ограничения (уникальность ключа).
3. Используются реляционные операторы для выполнения операций с таблицами.

## Этапы создания базы данных:

1. Анализ предметной области.
2. Сбор необходимых данных.
3. Разработка проекта базы данных.
4. Физическая реализация базы данных.

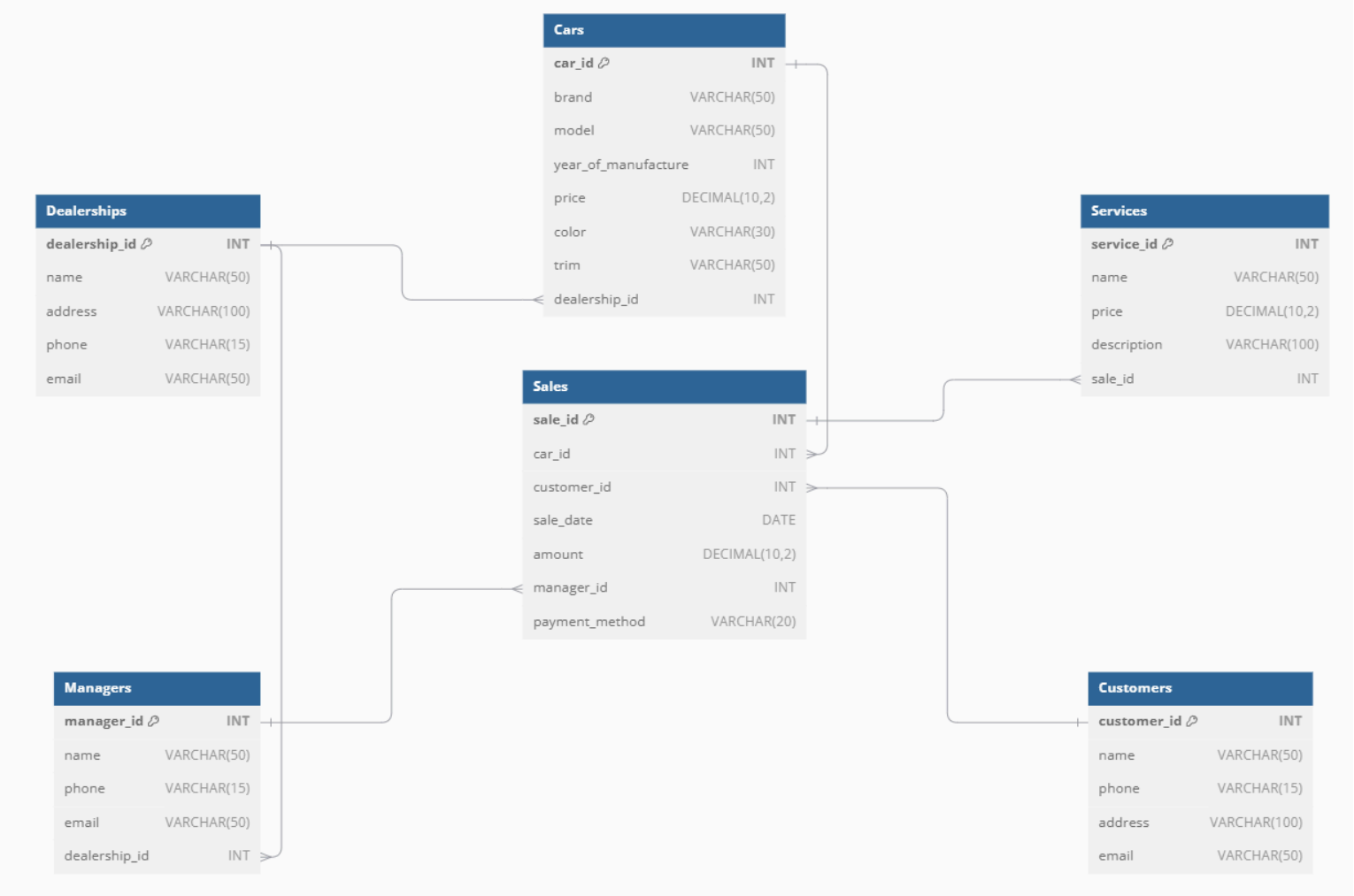
## Содержание работы:

1. Схема базы данных.
2. Примеры запросов для получения необходимой информации из базы данных.
3. Примеры представлений, обеспечивающих доступ к сводным данным.

## Схема данных включает в себя следующие таблицы:

1. **Cars** (car_id – ID автомобиля, brand – марка, model – модель, year_of_manufacture – год выпуска, price – цена, color – цвет, trim – комплектация, dealership_id – ID автосалона).
2. **Customers** (customer_id – ID клиента, name – имя клиента, phone – телефон клиента, address – адрес, email – электронная почта).
3. **Dealerships** (dealership_id – ID автосалона, name – название автосалона, address – адрес автосалона, phone – телефон, email – электронная почта).

4. **Sales** (sale_id – ID продажи, car_id – ID автомобиля, customer_id – ID клиента, sale_date – дата продажи, amount – сумма, manager_id – ID менеджера, payment_method – способ оплаты).
5. **Managers** (manager_id – ID менеджера, name – имя менеджера, phone – телефон менеджера, email – электронная почта, dealership_id – ID автосалона).
6. **Services** (service_id – ID услуги, name – название услуги, price – цена услуги, description – описание услуги, sale_id – ID продажи).



# Создание запросов (SELECT):

**1. Список всех продаж с информацией о клиенте, автомобиле и менеджере.**

```
SELECT
    Sales.sale_id,
    Cars.brand,
    Cars.model,
    Customers.name AS customer_name,
    Managers.name AS manager_name,
    Sales.sale_date,
    Sales.amount,
    Sales.payment_method
FROM
    Sales
JOIN Cars ON Sales.car_id = Cars.car_id
JOIN Customers ON Sales.customer_id = Customers.customer_id
JOIN Managers ON Sales.manager_id = Managers.manager_id;
```

**Объяснение:**
Этот SQL-запрос извлекает данные о продажах, включая информацию об автомобиле, клиенте и менеджере. Он объединяет таблицы Sales, Cars, Customers и Managers, используя ключи для связи между ними. Результат содержит идентификатор продажи, марку и модель автомобиля, имя клиента, имя менеджера, дату продажи, сумму и способ оплаты. Запрос позволяет получить полную информацию о каждой продаже, включая связанные с ней данные из других таблиц.

| SALE_ID | BRAND | MODEL | CUSTOMER_NAME | MANAGER_NAME | SALE_DATE | AMOUNT | PAYMENT_METHOD |
|---|---|---|---|---|---|---|---|
| 7 | Mercedes-Benz | GLE | Ivan Ivanov | Dmitry Sergeev | 05/01/2024 | 70000 | Credit |
| 1 | Mercedes-Benz | C-Class | Ivan Ivanov | Dmitry Sergeev | 01/15/2024 | 45000 | Credit |
| 6 | Mercedes-Benz | E-Class | Ivan Ivanov | Dmitry Sergeev | 04/10/2024 | 55000 | Cash |
| 12 | Mercedes-Benz | S-Class | Anna Smirnova | Olga Kuznetsova | 07/10/2024 | 80000 | Cash |
| 2 | Mercedes-Benz | S-Class | Anna Smirnova | Olga Kuznetsova | 02/10/2024 | 80000 | Cash |
| 8 | Mercedes-Benz | GLA | Peter Petrov | Nikolay Fedorov | 03/20/2024 | 40000 | Credit |
| 3 | Mercedes-Benz | E-Class | Peter Petrov | Dmitry Sergeev | 02/20/2024 | 55000 | Credit |
| 9 | Mercedes-Benz | GLC | Peter Petrov | Nikolay Fedorov | 05/15/2024 | 60000 | Cash |
| 13 | Mercedes-Benz | GLE | Alexey Romanov | Olga Kuznetsova | 08/01/2024 | 70000 | Credit |
| 5 | Mercedes-Benz | GLE | Alexey Romanov | Dmitry Sergeev | 03/15/2024 | 70000 | Credit |

**2. Общая выручка по каждому автосалону.**

```
SELECT
   Dealerships.name AS dealership_name,
   SUM(Sales.amount) AS total_revenue
FROM
   Sales
JOIN Managers ON Sales.manager_id = Managers.manager_id
JOIN Dealerships ON Managers.dealership_id = Dealerships.dealership_id
GROUP BY Dealerships.name;
```

**Объяснение:**
Этот SQL-запрос подсчитывает общую выручку для каждого автосалона. Он объединяет таблицы Sales, Managers и Dealerships, связывая их через соответствующие ключи. Результат содержит название автосалона и сумму всех продаж, связанных с ним. Группировка осуществляется по названиям автосалонов, а сумма выручки вычисляется с помощью агрегатной функции SUM. Это позволяет увидеть, сколько денег заработал каждый автосалон на продажах.

| DEALERSHIP_NAME | TOTAL_REVENUE |
|---|---|
| Mercedes-Benz Moscow | 295000 |
| Mercedes-Benz Kazan | 200000 |
| Mercedes-Benz St. Petersburg | 370000 |

**3. Список автомобилей, которые были проданы в кредит.**

```
SELECT
   Cars.brand,
   Cars.model,
   Cars.price,
   Sales.sale_date,
   Customers.name AS customer_name
FROM
   Sales
JOIN Cars ON Sales.car_id = Cars.car_id
JOIN Customers ON Sales.customer_id = Customers.customer_id
WHERE
   Sales.payment_method = 'Credit';
```

**Объяснение:**
Этот SQL-запрос извлекает данные о продажах автомобилей, которые

были оплачены в кредит. Он объединяет таблицы Sales, Cars и Customers, связывая их через ключи. Результат включает марку и модель автомобиля, его цену, дату продажи и имя клиента. Фильтр WHERE гарантирует, что в выборке будут только те продажи, где способ оплаты указан как "Credit". Это позволяет получить подробности о всех продажах, выполненных в кредит.

| BRAND | MODEL | PRICE | SALE_DATE | CUSTOMER_NAME |
|---|---|---|---|---|
| Mercedes-Benz | GLE | 70000 | 08/01/2024 | Alexey Romanov |
| Mercedes-Benz | GLE | 70000 | 03/15/2024 | Alexey Romanov |
| Mercedes-Benz | GLE | 70000 | 05/01/2024 | Ivan Ivanov |
| Mercedes-Benz | C-Class | 45000 | 01/15/2024 | Ivan Ivanov |
| Mercedes-Benz | C-Class | 45000 | 06/10/2024 | Maria Volkova |
| Mercedes-Benz | GLA | 40000 | 03/20/2024 | Peter Petrov |
| Mercedes-Benz | E-Class | 55000 | 02/20/2024 | Peter Petrov |
| Mercedes-Benz | E-Class | 55000 | 09/01/2024 | Maria Volkova |

**4. Клиенты, купившие более одного автомобиля.**

```
SELECT
   Customers.name,
   COUNT(Sales.sale_id) AS cars_bought
FROM
   Sales
JOIN Customers ON Sales.customer_id = Customers.customer_id
GROUP BY Customers.name
HAVING COUNT(Sales.sale_id) > 1;
```

**Объяснение:**
Этот SQL-запрос извлекает список клиентов, которые купили более одного автомобиля. Он объединяет таблицы Sales и Customers, связывая их по ключу customer_id. В результате возвращаются имя клиента и количество купленных им автомобилей. Группировка выполняется по имени клиента с использованием GROUP BY, а условие HAVING фильтрует результаты, оставляя только тех клиентов, у которых количество покупок превышает одну. Запрос помогает определить самых активных покупателей.

| NAME | CARS_BOUGHT |
|---|---|
| Ivan Ivanov | 3 |
| Peter Petrov | 3 |
| Elena Sidorova | 2 |
| Alexey Romanov | 2 |
| Anna Smirnova | 2 |
| Maria Volkova | 3 |

**5. Автосалоны с наибольшим количеством проданных автомобилей.**

```
SELECT
   Dealerships.name AS dealership_name,
   COUNT(Sales.sale_id) AS cars_sold
FROM
   Sales
JOIN Managers ON Sales.manager_id = Managers.manager_id
JOIN Dealerships ON Managers.dealership_id = Dealerships.dealership_id
GROUP BY Dealerships.name
ORDER BY cars_sold DESC;
```

**Объяснение:**
Этот SQL-запрос извлекает данные о количестве проданных автомобилей в каждом автосалоне. Он объединяет таблицы Sales, Managers и Dealerships, связывая их через ключи. Результат включает название автосалона и общее количество автомобилей, проданных через этот автосалон. Группировка осуществляется по названию автосалона с использованием GROUP BY, а сортировка по количеству проданных автомобилей выполняется в порядке убывания с помощью ORDER BY. Это позволяет определить, какой автосалон продал больше всего автомобилей.

| DEALERSHIP_NAME | CARS_SOLD |
|---|---|
| Mercedes-Benz St. Petersburg | 6 |
| Mercedes-Benz Moscow | 5 |
| Mercedes-Benz Kazan | 4 |

**6. Самые популярные дополнительные услуги.**

```
SELECT
   Services.name AS service_name,
   COUNT(Services.service_id) AS usage_count
FROM
   Services
GROUP BY Services.name
ORDER BY usage_count DESC;
```

**Объяснение:**
Этот SQL-запрос подсчитывает количество использований каждой услуги, предоставляемой в автосалонах. Он выполняет группировку по названию услуг с помощью GROUP BY, а для каждой группы подсчитывается количество записей в таблице Services с использованием агрегатной

функции COUNT. Результаты сортируются в порядке убывания частоты использования услуг с помощью ORDER BY. Запрос позволяет определить самые популярные услуги, предоставляемые в автосалонах.

| SERVICE_NAME | USAGE_COUNT |
|---|---|
| Extended Warranty | 2 |
| Maintenance | 2 |
| Winter Tires | 2 |
| Accessory Package | 2 |
| Insurance | 2 |

**7. Список автомобилей, проданных за указанный период (например, с января по март 2024 года)**

```
SELECT
   Cars.brand,
   Cars.model,
   Sales.sale_date,
   Sales.amount
FROM
   Sales
JOIN Cars ON Sales.car_id = Cars.car_id
WHERE
      Sales.sale_date BETWEEN TO_DATE('2024-01-01', 'YYYY-MM-DD')
AND TO_DATE('2024-03-31', 'YYYY-MM-DD');
```

**Объяснение:**
Этот SQL-запрос извлекает данные о продажах автомобилей, совершённых в определённый период. Он объединяет таблицы Sales и Cars через ключ car_id, чтобы отобразить марку и модель автомобиля, дату продажи и сумму продажи. Условие WHERE фильтрует записи, оставляя только те, где дата продажи находится в указанном диапазоне — с 1 января по 31 марта 2024 года. Запрос позволяет получить информацию о продажах за выбранный период.

| BRAND | MODEL | SALE_DATE | AMOUNT |
|---|---|---|---|
| Mercedes-Benz | S-Class | 02/10/2024 | 80000 |
| Mercedes-Benz | GLE | 03/15/2024 | 70000 |
| Mercedes-Benz | C-Class | 01/15/2024 | 45000 |
| Mercedes-Benz | GLA | 03/20/2024 | 40000 |
| Mercedes-Benz | GLA | 03/05/2024 | 40000 |
| Mercedes-Benz | E-Class | 02/20/2024 | 55000 |

**8. Список автомобилей, которые доступны в автосалоне, но ещё не были проданы.**

```
SELECT
    Cars.brand,
    Cars.model,
    Cars.price
FROM
    Cars
LEFT JOIN Sales ON Cars.car_id = Sales.car_id
WHERE
    Sales.car_id IS NULL;
```

**Объяснение:**
Этот SQL-запрос извлекает данные о доступных автомобилях, которые ещё не были проданы. Он использует LEFT JOIN для соединения таблиц Cars и Sales, чтобы найти все автомобили, независимо от того, есть ли у них связанные продажи. Условие WHERE Sales.car_id IS NULL фильтрует результаты, оставляя только те автомобили, для которых не было записей в таблице Sales. Результат включает марку, модель и цену автомобилей, которые остаются в наличии.

| BRAND | MODEL | PRICE |
|---|---|---|
| Mercedes-Benz | A-Class | 35000 |
| Mercedes-Benz | EQS | 100000 |
| Mercedes-Benz | EQB | 90000 |
| Mercedes-Benz | CLA | 45000 |
| Mercedes-Benz | V-Class | 75000 |
| Mercedes-Benz | GLB | 50000 |

**9. Средняя цена автомобилей по каждой комплектации.**

```
SELECT
    Cars.trim,
    ROUND(AVG(Cars.price), 2) AS average_price
FROM
    Cars
GROUP BY Cars.trim;
```

**Объяснение:**
Этот SQL-запрос вычисляет среднюю цену автомобилей для каждой комплектации. Группировка выполняется по полю trim (комплектация) с

использованием GROUP BY. Агрегатная функция AVG рассчитывает среднюю цену для каждой группы, а функция ROUND округляет результат до двух знаков после запятой. Результат отображает название комплектации и среднюю цену автомобилей в этой категории. Запрос позволяет сравнить среднюю стоимость автомобилей разных комплектаций.

| TRIM | AVERAGE_PRICE |
|---|---|
| Exclusive | 85000 |
| AMG Line | 45000 |
| Premium Plus | 77500 |
| Standard | 37500 |
| Electric | 90000 |
| Luxury | 55000 |

**10. Все клиенты, купившие автомобили в конкретном автосалоне (например, в "Mercedes-Benz Москва")**

```
SELECT
    Customers.name AS customer_name,
    Cars.brand,
    Cars.model,
    Dealerships.name AS dealership_name
FROM
    Sales
JOIN Cars ON Sales.car_id = Cars.car_id
JOIN Customers ON Sales.customer_id = Customers.customer_id
JOIN Managers ON Sales.manager_id = Managers.manager_id
JOIN Dealerships ON Managers.dealership_id = Dealerships.dealership_id
WHERE
    Dealerships.name = 'Mercedes-Benz Moscow';
```

**Объяснение:**
Этот SQL-запрос извлекает данные о клиентах, которые купили автомобили в автосалоне "Mercedes-Benz Moscow". Он объединяет таблицы Sales, Cars, Customers, Managers и Dealerships, связывая их через ключи. Результат включает имя клиента, марку и модель автомобиля, а также название автосалона. Условие WHERE Dealerships.name = 'Mercedes-Benz Moscow' фильтрует записи, чтобы оставить только те, которые связаны с указанным автосалоном. Запрос позволяет получить информацию о покупателях и купленных ими автомобилях в конкретном автосалоне.

| CUSTOMER_NAME | BRAND | MODEL | DEALERSHIP_NAME |
|---|---|---|---|
| Alexey Romanov | Mercedes-Benz | GLE | Mercedes-Benz Moscow |
| Ivan Ivanov | Mercedes-Benz | GLE | Mercedes-Benz Moscow |
| Ivan Ivanov | Mercedes-Benz | C-Class | Mercedes-Benz Moscow |
| Ivan Ivanov | Mercedes-Benz | E-Class | Mercedes-Benz Moscow |
| Peter Petrov | Mercedes-Benz | E-Class | Mercedes-Benz Moscow |

## Создание представлений (CREATE VIEW):

**1. История продаж с подробной информацией о клиенте, автомобиле и менеджере.**

```
CREATE OR REPLACE VIEW SaleHistory AS
SELECT
    Sales.sale_id,
    Cars.brand,
    Cars.model,
    Customers.name AS customer_name,
    Managers.name AS manager_name,
    Sales.sale_date,
    Sales.amount,
    Sales.payment_method
FROM
    Sales
JOIN Cars ON Sales.car_id = Cars.car_id
JOIN Customers ON Sales.customer_id = Customers.customer_id
JOIN Managers ON Sales.manager_id = Managers.manager_id;
```

**Объяснение:**
Этот SQL-запрос создаёт или обновляет представление под названием SaleHistory, которое содержит информацию о продажах автомобилей. Представление включает идентификатор продажи, марку и модель автомобиля, имя клиента, имя менеджера, дату продажи, сумму и способ оплаты. Оно объединяет данные из таблиц продаж, автомобилей, клиентов и менеджеров, предоставляя полный и удобный доступ к информации о каждой продаже. Использование CREATE OR REPLACE VIEW позволяет создать новое представление или обновить существующее.

**Для проверки представления использую следующее:**

***- Проверка продаж по менеджеру Дмитрию Сергееву (Manager ID 1)***

```
SELECT *
FROM SaleHistory
WHERE manager_name = 'Dmitry Sergeev';
```

| SALE_ID | BRAND | MODEL | CUSTOMER_NAME | MANAGER_NAME | SALE_DATE | AMOUNT | PAYMENT_METHOD |
|---|---|---|---|---|---|---|---|
| 5 | Mercedes-Benz | GLE | Alexey Romanov | Dmitry Sergeev | 03/15/2024 | 70000 | Credit |
| 7 | Mercedes-Benz | GLE | Ivan Ivanov | Dmitry Sergeev | 05/01/2024 | 70000 | Credit |
| 1 | Mercedes-Benz | C-Class | Ivan Ivanov | Dmitry Sergeev | 01/15/2024 | 45000 | Credit |
| 6 | Mercedes-Benz | E-Class | Ivan Ivanov | Dmitry Sergeev | 04/10/2024 | 55000 | Cash |
| 3 | Mercedes-Benz | E-Class | Peter Petrov | Dmitry Sergeev | 02/20/2024 | 55000 | Credit |

**2. Общая выручка по автосалонам.**

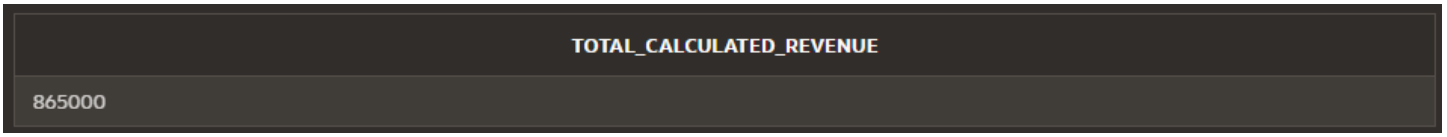
```
CREATE OR REPLACE VIEW DealershipRevenue AS
SELECT
    Dealerships.name AS dealership_name,
    SUM(Sales.amount) AS total_revenue
FROM
    Sales
JOIN Managers ON Sales.manager_id = Managers.manager_id
JOIN Dealerships ON Managers.dealership_id = Dealerships.dealership_id
GROUP BY Dealerships.name;
```

**Объяснение:**
Этот SQL-запрос создаёт или обновляет представление под названием DealershipRevenue, которое отображает информацию о выручке каждого автосалона. Представление включает название автосалона и общую сумму всех продаж, связанных с ним. Для вычисления выручки данные из таблицы Sales объединяются с таблицами Managers и Dealerships, чтобы установить связь между продажами и автосалонами. Сумма продаж подсчитывается с помощью функции SUM, а данные группируются по названию автосалона с использованием GROUP BY. Представление позволяет легко получить сводную информацию о доходах каждого автосалона.

**Для проверки представления использую следующее:**

***- Сравнение общей выручки всех автосалонов***

```
SELECT
   SUM(total_revenue) AS total_calculated_revenue
FROM
   DealershipRevenue;
```



| TOTAL_CALCULATED_REVENUE |
| --- |
| 865000 |

**3. Популярные услуги с количеством их использования.**

```
CREATE OR REPLACE VIEW PopularServices AS
SELECT
   Services.name AS service_name,
   COUNT(Services.service_id) AS usage_count
FROM
   Services
GROUP BY Services.name
ORDER BY usage_count DESC;
```

**Объяснение:**
Этот SQL-запрос создаёт или обновляет представление с названием PopularServices, которое предоставляет информацию о популярности услуг в автосалоне. Представление содержит название услуги и количество её использований. Данные группируются по названию услуг с использованием GROUP BY, а количество использований подсчитывается с помощью функции COUNT. Результаты сортируются в порядке убывания частоты использования услуг благодаря ORDER BY usage_count DESC. Это представление позволяет легко определить, какие услуги наиболее востребованы.

**Для проверки представления использую следующее:**

***- Сравнение количества использования услуг с таблицей Services***

```
SELECT
   Services.name AS service_name,
   COUNT(Services.service_id) AS usage_count
FROM
   Services
GROUP BY Services.name;
```

| SERVICE_NAME | USAGE_COUNT |
|---|---|
| Extended Warranty | 2 |
| Maintenance | 2 |
| Insurance | 2 |
| Accessory Package | 2 |
| Winter Tires | 2 |

**4. Клиенты с количеством купленных автомобилей.**

```
CREATE OR REPLACE VIEW CustomerCarCount AS
SELECT
   Customers.name AS customer_name,
   COUNT(Sales.sale_id) AS cars_bought
FROM
   Sales
JOIN Customers ON Sales.customer_id = Customers.customer_id
GROUP BY Customers.name;
```

**Объяснение:**
Этот SQL-запрос создаёт или обновляет представление под названием CustomerCarCount, которое предоставляет информацию о количестве автомобилей, купленных каждым клиентом. Представление объединяет данные из таблиц Sales и Customers, связывая их по идентификатору клиента. В результате для каждого клиента подсчитывается общее количество покупок автомобилей с использованием функции COUNT. Группировка данных выполняется по имени клиента с помощью GROUP BY. Это представление позволяет быстро получить сводную информацию о покупательской активности клиентов.

**Для проверки представления использую следующее:**

***- Проверка всех данных из представления***

```
SELECT * FROM CustomerCarCount;
```

| CUSTOMER_NAME | CARS_BOUGHT |
|---|---|
| Ivan Ivanov | 3 |
| Peter Petrov | 3 |
| Elena Sidorova | 2 |
| Alexey Romanov | 2 |
| Anna Smirnova | 2 |
| Maria Volkova | 3 |

**5. Доступные автомобили в автосалонах.**

```
CREATE OR REPLACE VIEW AvailableCars AS
SELECT
    Cars.brand,
    Cars.model,
    Cars.price,
    Dealerships.name AS dealership_name
FROM
    Cars
LEFT JOIN Sales ON Cars.car_id = Sales.car_id
JOIN Dealerships ON Cars.dealership_id = Dealerships.dealership_id
WHERE
    Sales.car_id IS NULL;
```

**Объяснение:**
Этот SQL-запрос создаёт или обновляет представление под названием AvailableCars, которое отображает информацию о доступных автомобилях, ещё не проданных в автосалонах. Представление включает марку, модель, цену автомобиля и название автосалона. Запрос использует LEFT JOIN для объединения таблиц Cars и Sales, чтобы найти автомобили, не имеющие записей о продажах. Связь между автомобилями и автосалонами устанавливается через таблицу Dealerships. Условие WHERE Sales.car_id IS NULL фильтрует только те автомобили, которые остаются доступными. Это представление позволяет легко увидеть, какие автомобили в наличии в каждом автосалоне.

**Для проверки представления использую следующее:**

***- Проверка всех данных из представления***

```
SELECT * FROM AvailableCars;
```

| BRAND | MODEL | PRICE | DEALERSHIP_NAME |
|---|---|---|---|
| Mercedes-Benz | A-Class | 35000 | Mercedes-Benz Moscow |
| Mercedes-Benz | EQS | 100000 | Mercedes-Benz St. Petersburg |
| Mercedes-Benz | EQB | 90000 | Mercedes-Benz Kazan |
| Mercedes-Benz | CLA | 45000 | Mercedes-Benz St. Petersburg |
| Mercedes-Benz | V-Class | 75000 | Mercedes-Benz Kazan |
| Mercedes-Benz | GLB | 50000 | Mercedes-Benz Kazan |

## Вывод по выполненной работе:

В рамках данной работы была разработана реляционная база данных под названием **«База данных для управления информацией в дилерских центрах Mercedes-Benz»**, предназначенная для организации данных о продажах, клиентах, автомобилях, услугах и автосалонах. Работа охватывала все ключевые этапы проектирования базы данных: анализ предметной области, сбор необходимых данных, проектирование структуры и её реализация в приложении Oracle APEX.

Созданная база данных обладает хорошо организованной структурой, включающей шесть взаимосвязанных таблиц: **Cars** (автомобили), **Customers** (клиенты), **Dealerships** (автосалоны), **Sales** (продажи), **Managers** (менеджеры) и **Services** (услуги). Таблицы находятся в нормальной форме, что обеспечивает целостность данных и минимизирует избыточность. В базе данных реализованы все необходимые ограничения, включая первичные и внешние ключи, для обеспечения точности и связности информации.

Для анализа и работы с данными были разработаны 10 сложных и составных запросов, позволяющих получать разнообразную информацию, такую как общая выручка автосалонов, доступные автомобили, популярные услуги, активность клиентов и другие ключевые показатели. Это обеспечивает удобство доступа к данным и возможность глубокого анализа операций.

Дополнительно созданы 5 представлений, которые упрощают доступ к наиболее востребованным данным. Среди них — история продаж, популярные услуги, доступные автомобили и данные о покупках клиентов. Эти представления служат инструментами для быстрого получения агрегированной информации, облегчая выполнение рутинных операций и повышая эффективность управления данными.

В ходе работы были добавлены тестовые данные, имитирующие реальную деятельность автосалонов. Это позволило не только проверить корректность работы базы данных, но и продемонстрировать её функциональность. Система охватывает несколько автосалонов в разных городах, учитывает разные категории автомобилей, а также отображает активность клиентов и востребованность услуг.

Созданная база данных позволяет эффективно управлять данными, связанными с работой дилерских центров Mercedes-Benz.