Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Дисциплина «Методы сбора, хранения, обработки и анализа данных»

Студент: Коренчук А.В.

ФИТ 3 курс 2 группа

Преподаватель: Блинова

Содержание

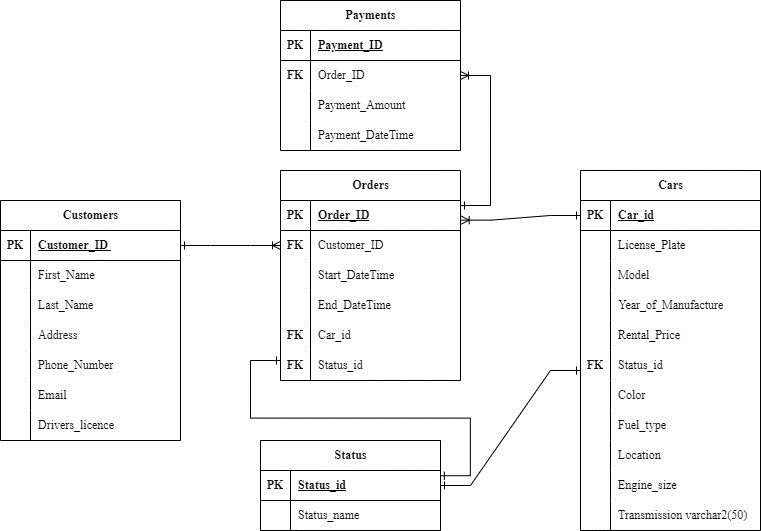
[**Лабораторная работа № 1 3**](#_Toc144645880)

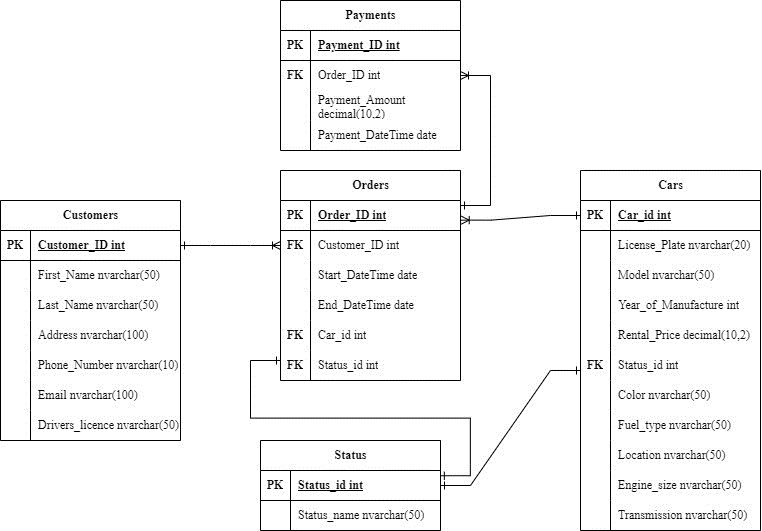
[**Лабораторная работа № 2 4**](#_Toc144645881)

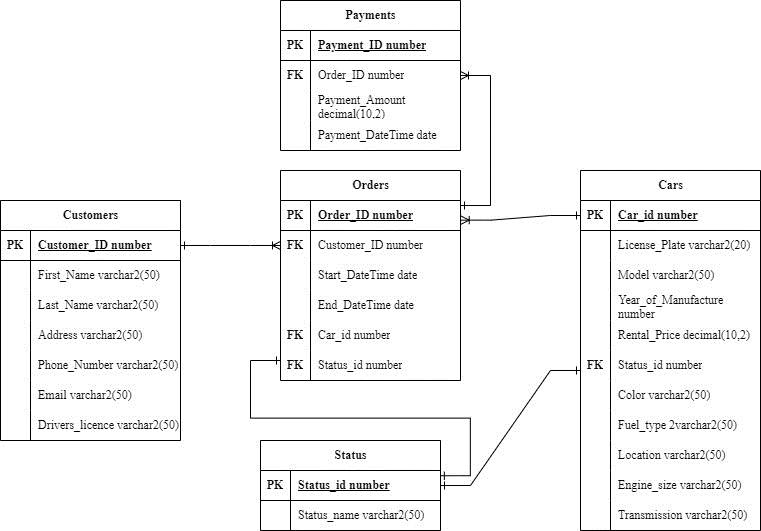
# **Лабораторная работа № 1**

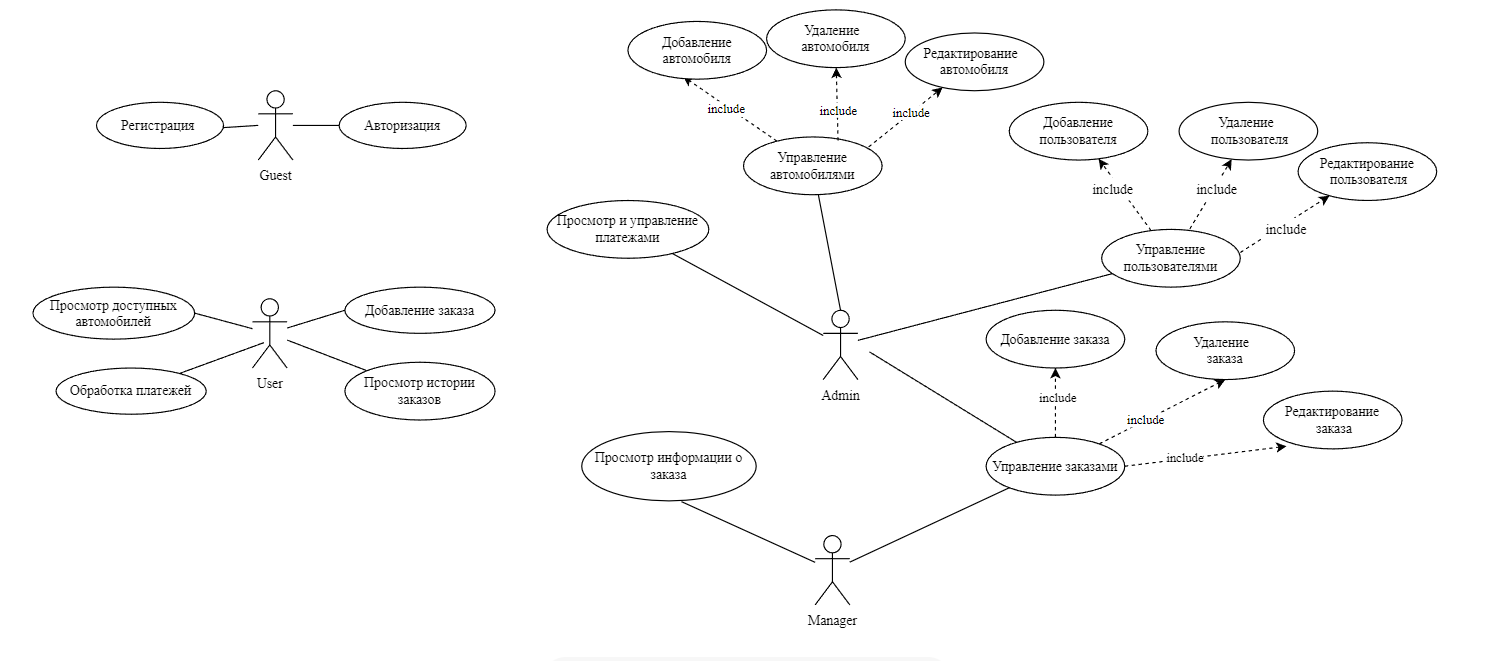
Задание, состоящий из: UML диаграммы; логической схемы БД; физических схем БД для СУБД SQL Server и Oracle.

|  |  |
| --- | --- |
| № | База данных |
| 5 | Услуги аренды |









# **Лабораторная работа № 2**

Отчет, состоящий из: списка таблиц с указанием атрибутов (название, тип данных, ограничения, назначение), списка прочих объектов (тип объекта, наименование, назначение, зависимые объекты) отдельно для СУБД SQL Server и Oracle.

Таблица 1.1– Сars

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип данных | Ограничения | Назначение |
| Car\_id | INT | PRIMARY KEY | Уникальный идентификатор автомобиля |
| License\_Plate | VARCHAR(20) |  | Номерной знак автомобиля |
| Model | VARCHAR(20) |  | Модель автомобиля |
| Year\_of\_Manufactory | INT |  | Год выпуска автомобиля |
| Rental\_price | DECIMAL(10,2) |  | Стоимость аренды в час |
| Color | VARCHAR(20) |  | Цвет машины |
| Fuel\_Type | VARCHAR(20) |  | Тип топлива |
| Location | VARCHAR(20) |  | Место машины |
| Status\_id | INT | FOREIGN KEY | Статус машины |

Таблица 1.2 - Customers

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип данных | Ограничения | Назначение |
| Customer\_id | INT | PRIMARY KEY | Уникальный идентификатор клиента |
| First\_Name | VARCHAR(20) |  | Имя клиента |
| Last\_Name | VARCHAR(20) |  | Фамилия клиента |
| Address | VARCHAR(20) |  | Адрес клиента |
| Phone\_number | VARCHAR(20) |  | Номер телефона клиента |
| Email | VARCHAR(20) |  | Адрес электронной почты клиента |
| Phone\_number | VARCHAR(20) |  | Номер телефона клиента |
| Email | VARCHAR(20) |  | Адрес электронной почты клиента |
| Drivers\_licence | VARCHAR(20) |  | Права клиента |

Таблица 1.3 - Orders

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип данных | Ограничения | Назначение |
| Order\_id | INT | PRIMARY KEY | Уникальный идентификатор заказа |
| Start\_DateTime | VARCHAR(20) |  | Дата начала аренды |
| End\_DateTime | VARCHAR(20) |  | Дата окончания аренды |
| Status\_id | VARCHAR(20) | FOREIGN KEY | Статус заказа (активен, закрыт). |
| Customer\_id | VARCHAR(20) | FOREIGN KEY | Связь с клиентом (внешний ключ) |
| Car\_id | VARCHAR(20) | FOREIGN KEY | Связь с автомобилем (внешний ключ) |

Таблица – 1.4 Payments

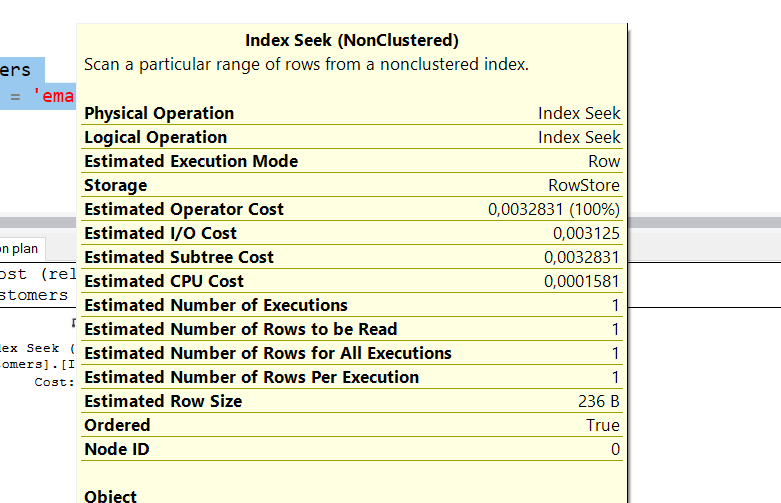
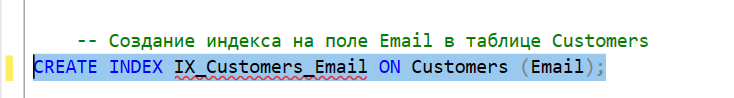
|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип данных | Ограничения | Назначение |
| Payment\_ID | INT | PRIMARY KEY | Уникальный идентификатор платежа. |
| Order\_ID | VARCHAR(20) | FOREIGN KEY | Связь с заказом (внешний ключ) |
| Payment\_Amount | VARCHAR(20) |  | Сумма платежа |
| Payment\_Date | VARCHAR(20) |  | Дата платежа |

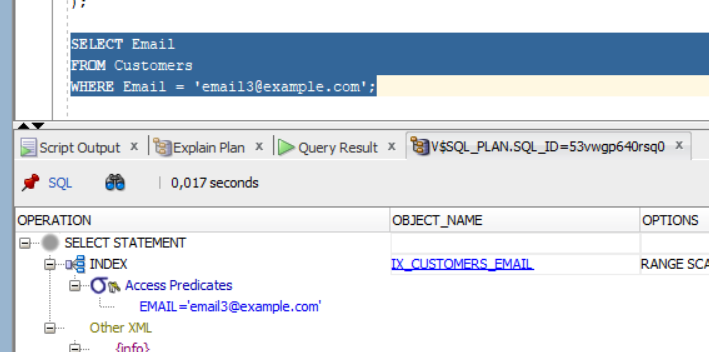
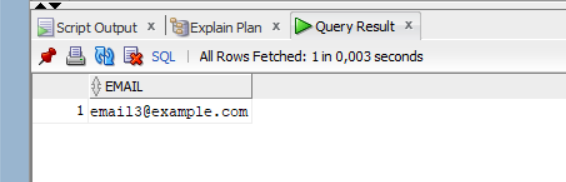
Таблица – 1.5 Status

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип данных | Ограничения | Назначение |
| Status\_id | INT | PRIMARY KEY | Уникальный идентификатор платежа. |
| Status\_name | VARCHAR(20) |  | Названия статуса |

Таблица – 1.6 Объекты

|  |  |  |  |
| --- | --- | --- | --- |
| Тип объекта | Наименование | Назначение | Зависимые объекты |
| Представление | UserOrderStatus | Представление, предоставляющее информацию о заказах клиентов. | Сustomers,  Order, Status |
| Представление | CarOrderStatus | Представление, предоставляющее информацию о заказах автомобилей. | Car, Order, Status |
| Индекс | IX\_Customers\_Email | Индекс для ускорения поиска клиентов по полю Email. | Customers |
| Процедура | InstertOrder | Процедура для добавление зкаказ | Order |





# **Лабораторная работа № 4**

**Расширенные группировки. Аналитические функции.**

|  |  |
| --- | --- |
| Услуги аренды | Вычисление итогов предоставленных услуг для определенного вида услуги помесячно, за квартал, за полгода, за год. |

Чтобы выполнить это задание, необходимо создать временную таблицу (CTE) DateIntervals, содержащую четыре временных интервала, а затем объединить данные из таблиц Orders, Payments, Order\_Services и Services, используя операторы LEFT JOIN с условиями, основанными на временных интервалах и услугах с именем "Аренда автомобиля". Затем выполнить агрегацию данных по времени и вывести результаты, включая временные интервалы и итоговую сумму оплаты. Результат работы будет на рисунке 4.1.

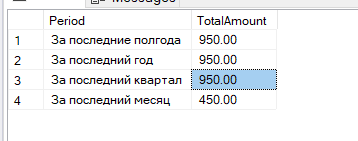


Рисунок 4.1 - Результат работы

|  |  |  |
| --- | --- | --- |
| 5 | Услуги аренды | Вычисление итогов предоставленных услуг для определенного вида услуги за период:   * объем услуг; * сравнение их с общим объемом услуг (в %); * сравнение с наибольшим объемом услуг (в %). |

Скрипт позволяет вычислить объем услуг для определенного вида услуги за указанный период, сравнить его с общим объемом услуг (в процентах) и с наибольшим объемом услуг (в процентах) для всех видов услуг. Результат работы будет на рисунке 4.2.

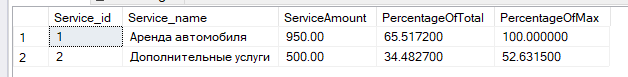


Рисунок 4.2 - Результат работы

Продемонстрируйте применение функции ранжирования ROW\_NUMBER() для разбиения результатов запроса на страницы (по 20 строк на каждую страницу). Результат работы будет на рисунке 4.3.

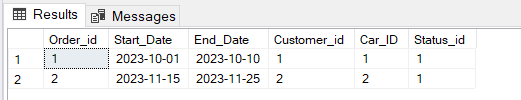


Рисунок 4.3 - Результат работы

Продемонстрируйте применение функции ранжирования ROW\_NUMBER() для удаления дубликатов. Результат работы будет на рисунке 4.4.

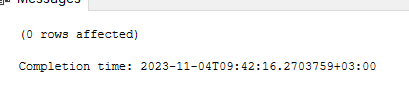


Рисунок 4.4 - Результат работы

|  |  |
| --- | --- |
| Услуги аренды | Вернуть для каждого вида клиентов суммы за аренду последних 6 месяцев помесячно. |

Для этого задания была создана временная таблица (CTE) CustomerRentData, которая содержит информацию о суммах аренды для каждого вида клиентов за последние 6 месяцев, разбитую помесячно. Затем результаты запроса выводятся в виде таблицы, включая данные о идентификаторе клиента, имени, фамилии, названии услуги, годе, месяце и сумме аренды, с сортировкой по идентификатору клиента, году, месяцу и названию услуги. Результат работы будет на рисунке 4.5.

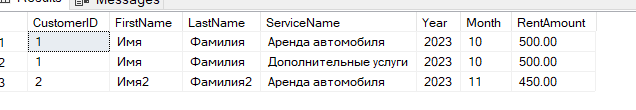


Рисунок 4.5 - Результат работы

|  |  |  |
| --- | --- | --- |
| 5 | Услуги аренды | Какая услуга была предоставлена наибольшее число раз для определенного вида? Вернуть для всех клиентов. |

Для задания создана временная таблица (CTE) `ServiceCounts`, которая содержит информацию о количестве предоставленных услуг разных видов для каждого клиента, а также их ранжирование в порядке убывания. Затем извлекаются данные о клиентах и услугах, для которых услуга была предоставлена наибольшее число раз, с сортировкой по названию услуги. Результат работы будет на рисунке 4.6.

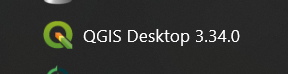


Рисунок 4.6 - Результат работы

# **Лабораторная работа № 5**

**Специальные типы данных в SQL Server – Пространственные данные.**

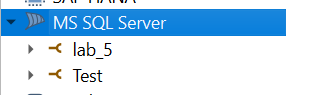
1. Установите приложение QGIS.



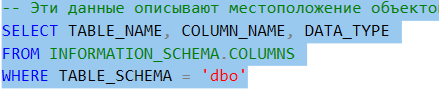
1. Загрузите данные в QGIS.

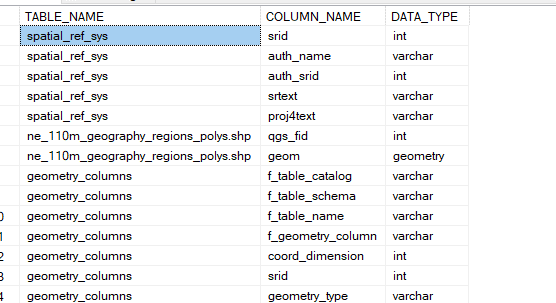


1. Установите подключение к своей СУБД из QGIS.

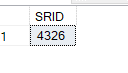
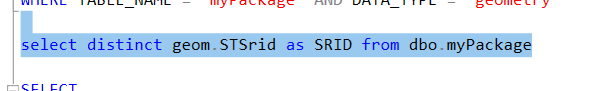


1. Определите тип пространственных данных во всех таблицах.



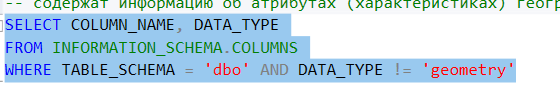


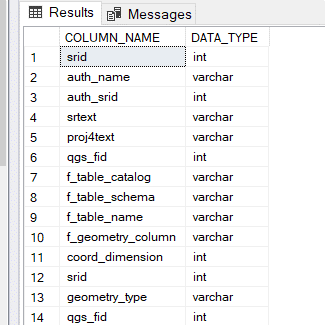
1. Определите SRID.



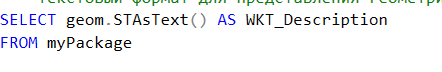
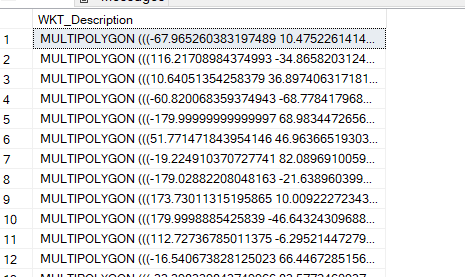
Идентификатор пространственной системы координат

1. Определите атрибутивные столбцы.

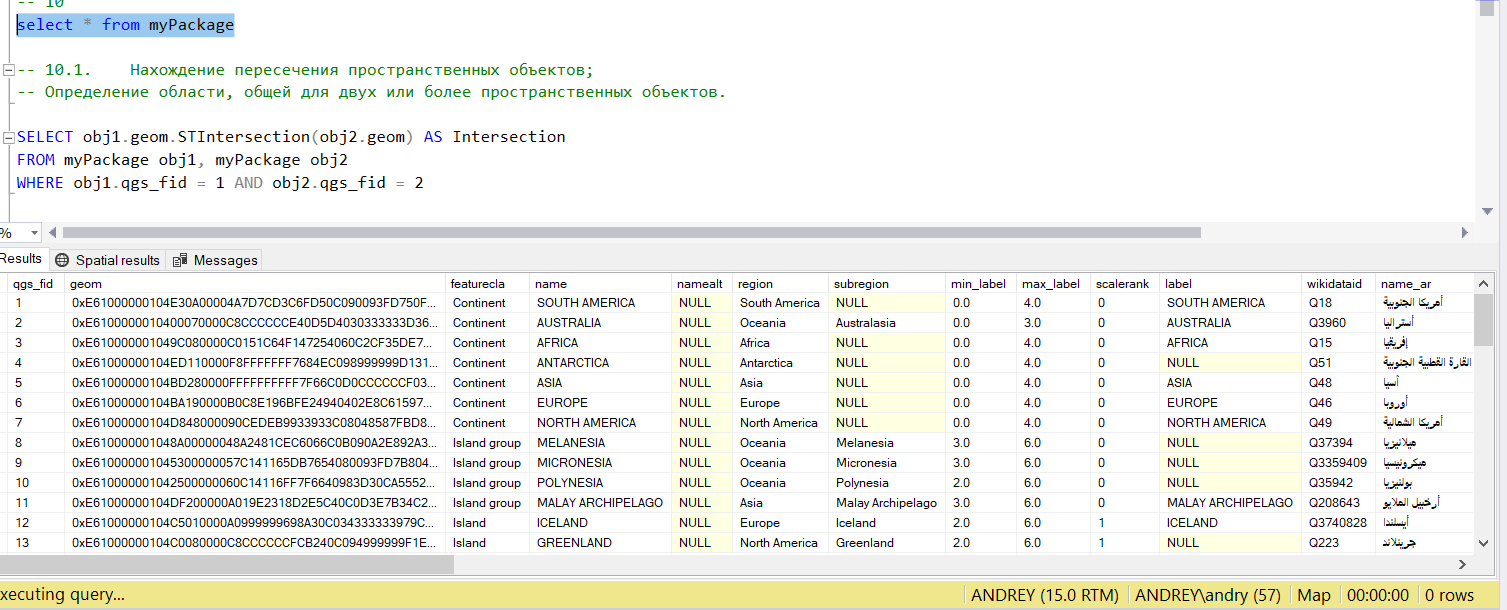


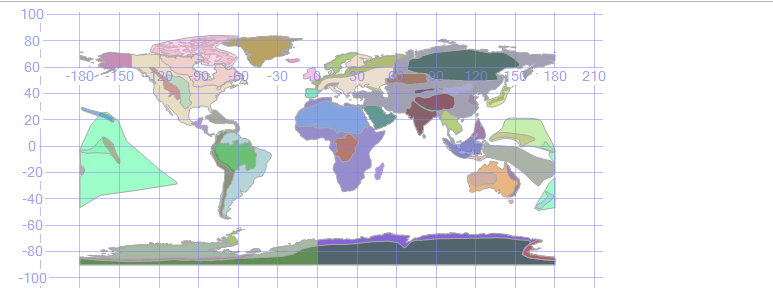


1. Верните описания пространственных объектов в формате WKT.

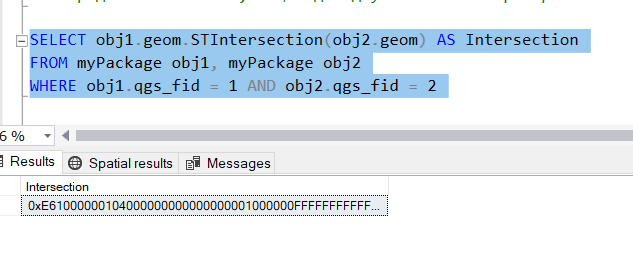
WKT - это текстовый формат для представления геометрических объектов в пространстве (точек, линий, полигонов и т. д.). 

1. Продемонстрируйте:

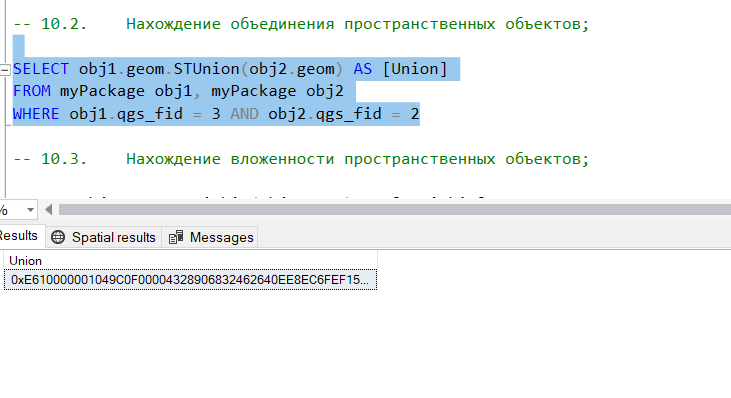


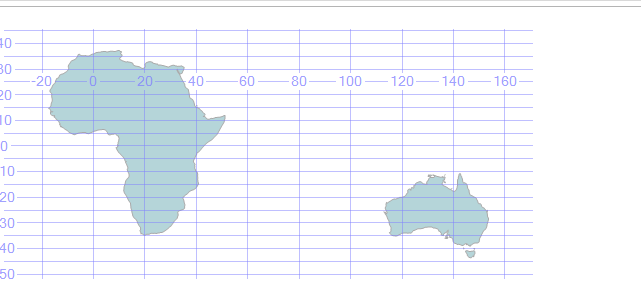


* 1. Нахождение пересечения пространственных объектов;

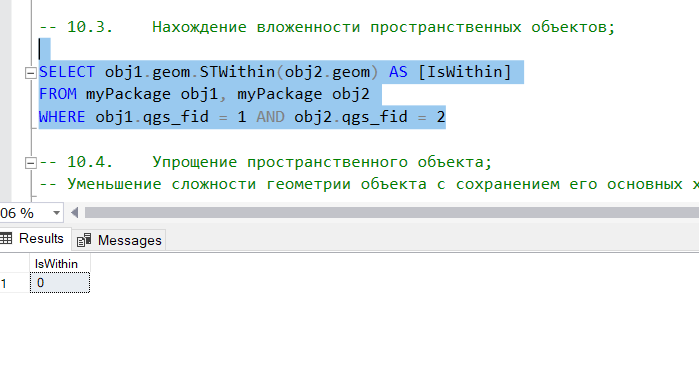


* 1. Нахождение объединения пространственных объектов;

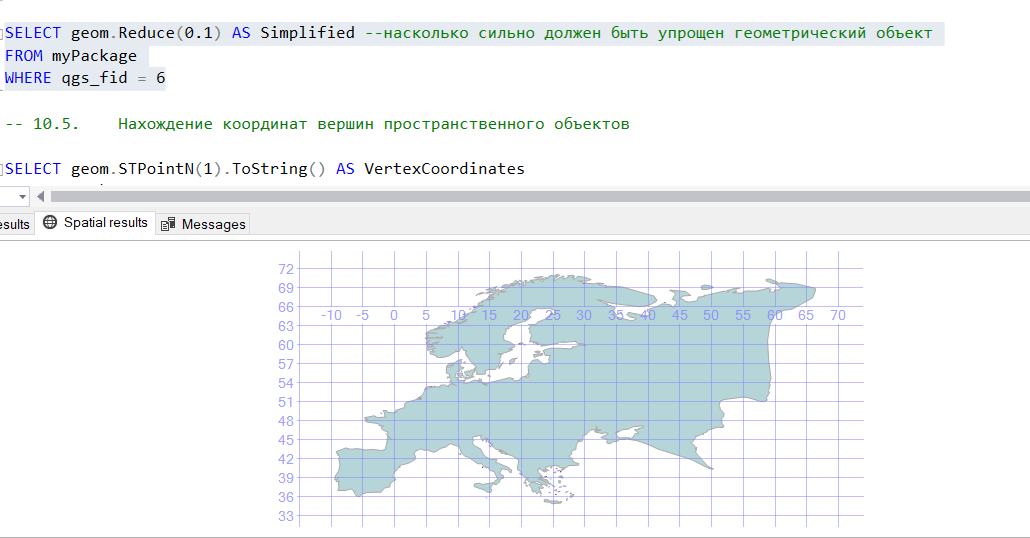




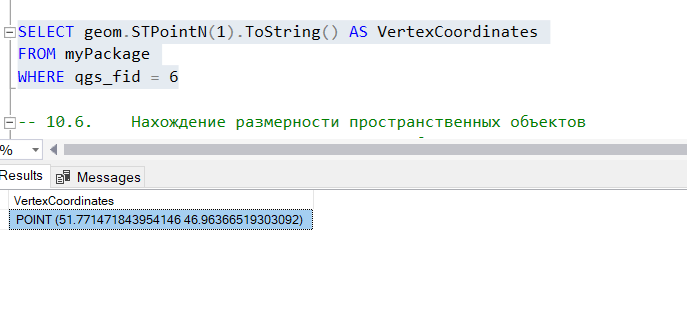
* 1. Нахождение вложенности пространственных объектов;



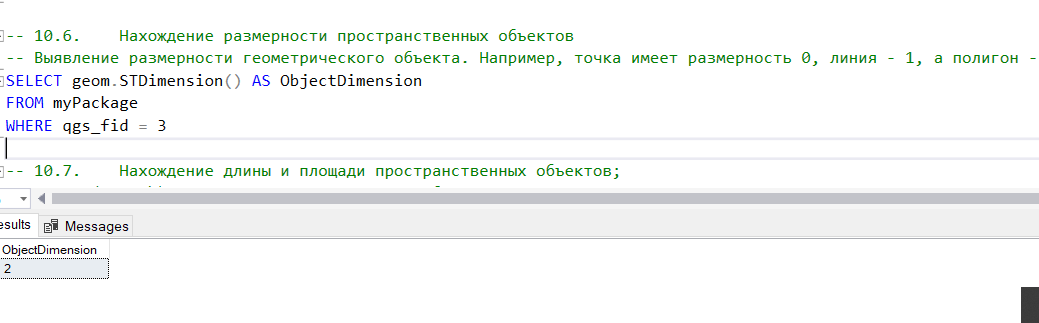
* 1. Упрощение пространственного объекта;



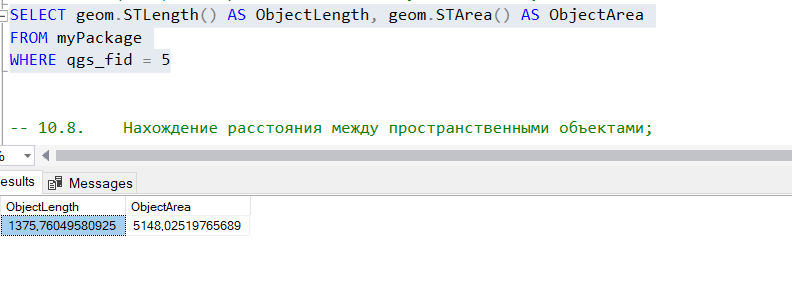
* 1. Нахождение координат вершин пространственного объектов;



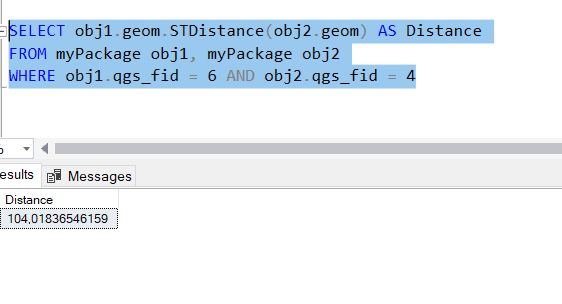
* 1. Нахождение размерности пространственных объектов;



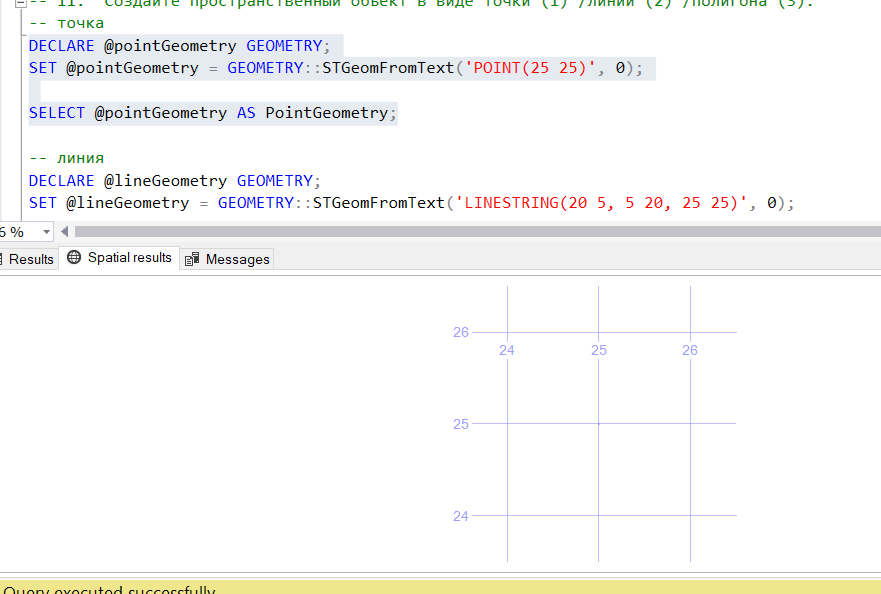
* 1. Нахождение длины и площади пространственных объектов;



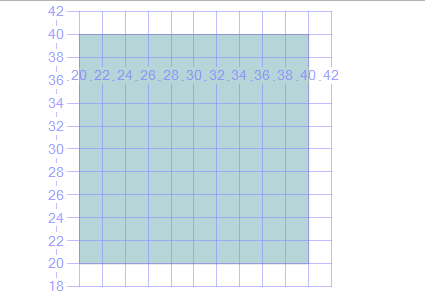
* 1. Нахождение расстояния между пространственными объектами;



1. Создайте пространственный объект в виде точки (1) /линии (2) /полигона (3).



1. Найдите, в какие пространственные объекты попадают созданные вами объекты.

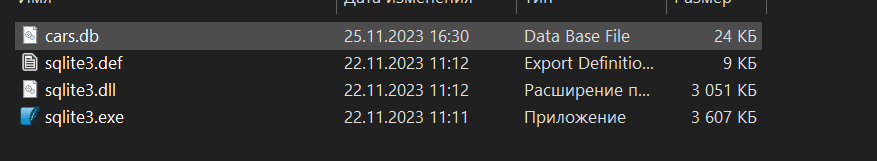


# **Лабораторная работа № 6**

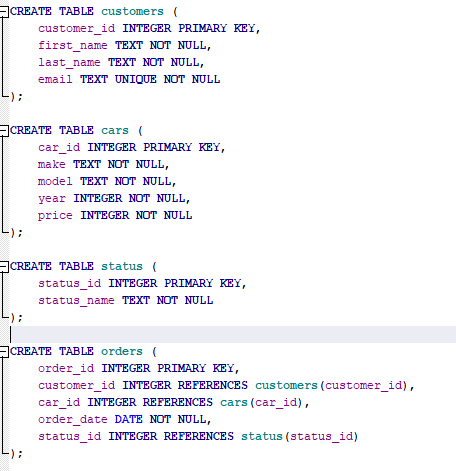
**SQLite**

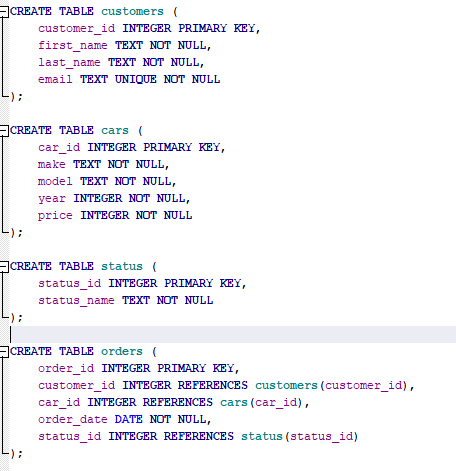
Задание:

* + - 1. Установить SQLite.

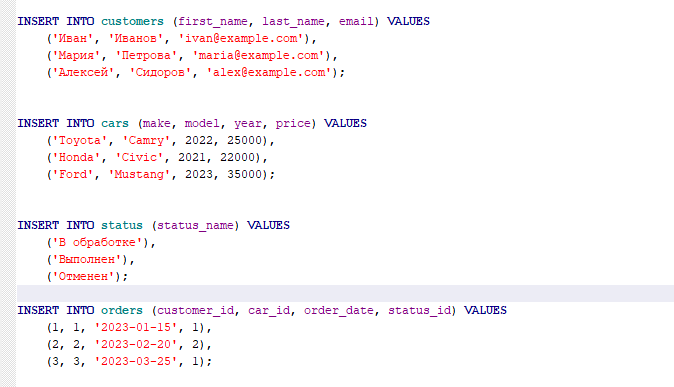


* + - 1. Создать базу данных SQLite, дополняющую вашу базу данных SQL Server по варианту.

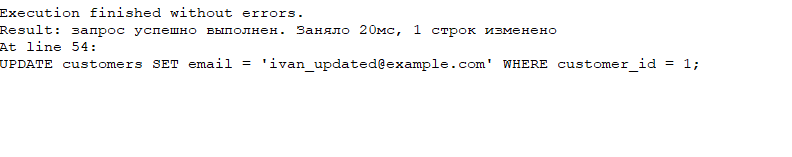


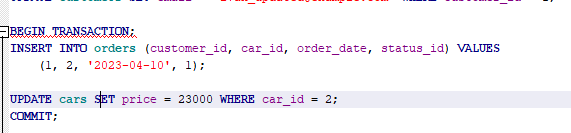


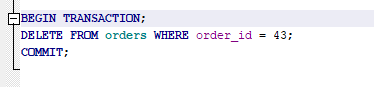
* + - 1. Добавить данные в базу данных SQLite.

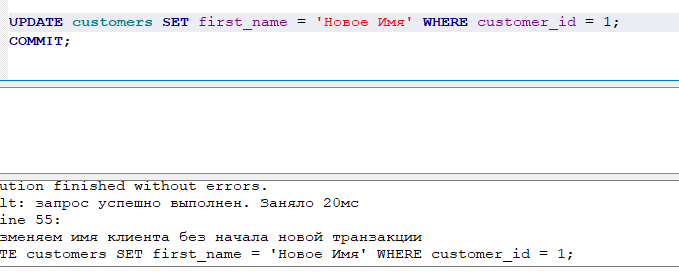


* + - 1. Продемонстрировать обновление, добавление и удаление данных в подчиненной таблице базы данных SQLite. Обработка должна демонстрировать особенности применения внешних ключей в базе данных SQLite и использовать транзакции.

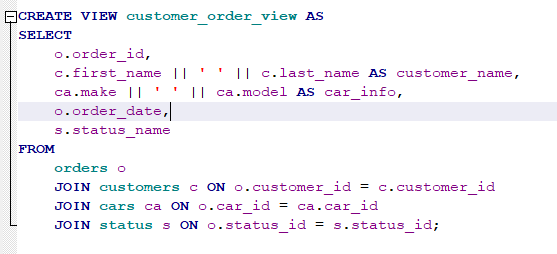
 

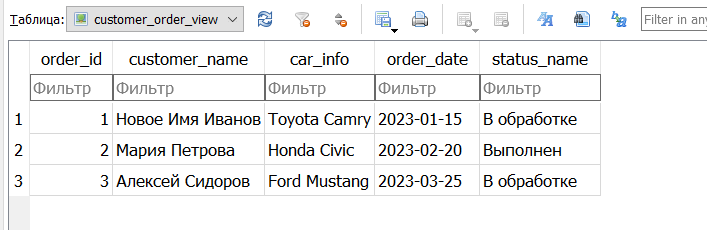




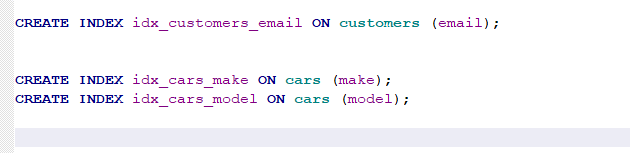
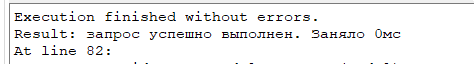


* + - 1. Создать представление в базе данных SQLite.

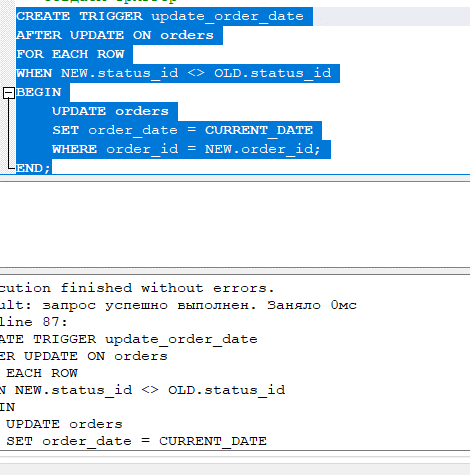




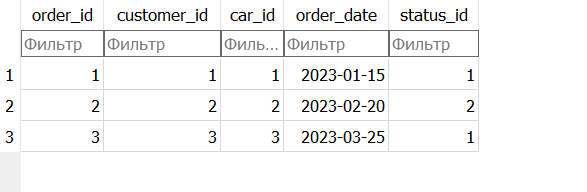
* + - 1. Создать необходимые индексы в базе данных SQLite.

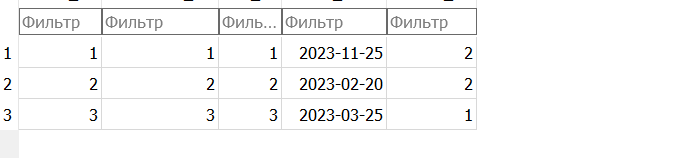
 

* + - 1. Создать триггер в базе данных SQLite.



* + - 1. Продемонстрировать использование созданных объектов в приложении.





Вопросы:

1. Что такое SQLite?
2. Что такое классы хранения? Какие классы хранения существуют в SQLite?
3. Что такое аффинированные типы? Какие аффинированные типы существуют в SQLite?
4. Как хранить дату в SQLite?
5. Поясните правила сравнения данных в SQLite.
6. Перечислите объекты SQLite.
7. Поясните порядок использования внешних ключей в SQLite.
8. Что такое PRAGMA?
9. Какие типы индексов существуют в SQLite?
10. Какие программные объекты SQLite вы знаете?

1. \*\*Что такое SQLite?\*\*

SQLite - это встраиваемая система управления базами данных (СУБД), которая является компактной, быстрой и легко доступной. Она предоставляет реляционную базу данных, которая не требует отдельного сервера и хранится в единственном файле.

2. \*\*Что такое классы хранения? Какие классы хранения существуют в SQLite?\*\*

Классы хранения в SQLite - это способы представления и хранения данных. В SQLite есть три основных класса хранения: INTEGER (целые числа), REAL (вещественные числа), и TEXT (текстовые строки). Кроме того, есть еще два специальных класса - BLOB (двоичные данные) и NULL (пустое значение).

3. \*\*Что такое аффинированные типы? Какие аффинированные типы существуют в SQLite?\*\*

В SQLite аффинированные типы связаны с классами хранения. Например, INTEGER аффинирован к целым числам, REAL аффинирован к вещественным числам, и так далее. Это свойство влияет на правила преобразования данных при выполнении операций.

4. \*\*Как хранить дату в SQLite?\*\*

В SQLite даты можно хранить с использованием типа данных TEXT, REAL или INTEGER. Обычно применяется тип данных TEXT, где дата представляется в виде текстовой строки в формате 'YYYY-MM-DD HH:MM:SS'. Используя функции преобразования, SQLite позволяет работать с датами.

5. \*\*Поясните правила сравнения данных в SQLite.\*\*

В SQLite сравнение данных осуществляется с учетом аффинированных типов. Строковые сравнения выполняются лексикографически, числовые - математически, а NULL считается меньшим чем любое другое значение.

6. \*\*Перечислите объекты SQLite.\*\*

Объекты SQLite включают в себя таблицы, представления, индексы, триггеры, триггерные процедуры и т. д.

7. \*\*Поясните порядок использования внешних ключей в SQLite.\*\*

В SQLite внешние ключи используются для создания связей между таблицами. Они определяют ссылочные ограничения между столбцами в двух таблицах, обеспечивая целостность данных. Внешний ключ определяется в таблице-потомке и ссылается на уникальный ключ в таблице-родителе.

8. \*\*Что такое PRAGMA?\*\*

PRAGMA - это SQL-инструкция в SQLite, которая предназначена для выполнения различных операций, таких как установка параметров сессии, получение информации о базе данных, определение настроек оптимизации и других.

9. \*\*Какие типы индексов существуют в SQLite?\*\*

В SQLite существует несколько типов индексов, включая B-tree (по умолчанию), R-tree (для пространственных данных), и индексы для полнотекстового поиска (FTS). Вы также можете создавать свои собственные пользовательские индексы с помощью команды `CREATE INDEX`.

10. \*\*Какие программные объекты SQLite вы знаете?\*\*

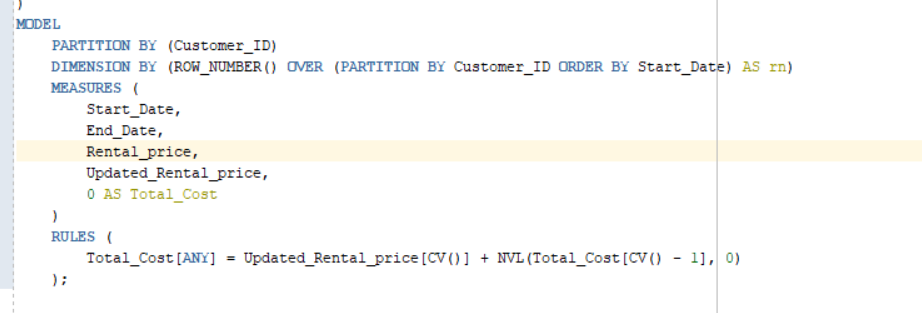
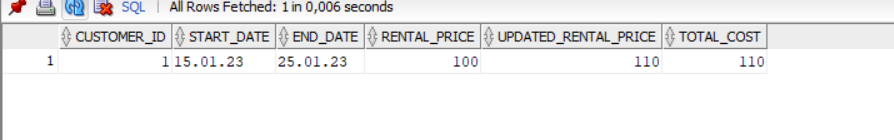
Программные объекты SQLite включают в себя библиотеки для различных языков программирования, такие как SQLite API для C, sqlite3 модуль для Python, SQLite.NET для C#, JDBC для Java и т. д. Также существуют графические инструменты управления базой данных SQLite, такие как DB Browser for SQLite.

# **Лабораторная работа № 7**

**Конструкции MODEL() и MATCH\_RECOGNIZE() для обработки данных в Oracle.**

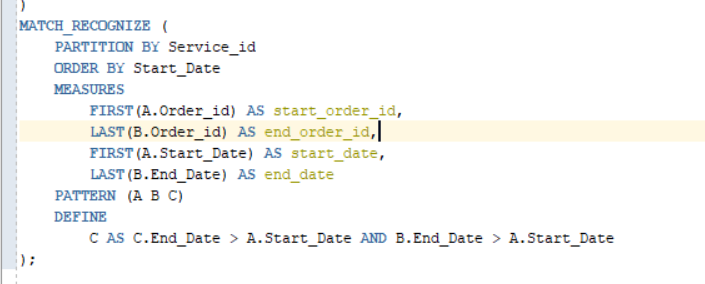
1. Постройте при помощи конструкции MODEL запросы, которые разрабатывают план:

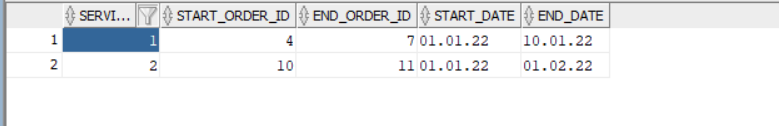
|  |  |  |
| --- | --- | --- |
| 5 | Услуги аренды | стоимости аренды для каждого клиента на следующий год, учитывая рост коммунальных затрат на 10% по сравнению с аналогичным месяцем прошлого года. |

1. Найдите при помощи конструкции **MATCH\_RECOGNIZE()** данные, которые соответствуют шаблону:

|  |  |  |
| --- | --- | --- |
| 5 | Услуги аренды | Рост, падение, рост предоставления для каждого вида услуг |





# **Лабораторная работа № 7**

*Лабораторная работа № 11*

**Миграция базы данных из MS SQL Server в Oracle**

Задание:

1. Произведите миграцию структуры, программного кода и данных базы данных из СУБД MS SQL Server в СУБД Oracle любым удобным для себя способом.
2. В отчете опишите процесс миграции, возникшие проблемы и способы их решения.

