

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Емельянов Сергей Геннадьевич

Должность: ректор

Дата подписания: 27.04.2023 09:18:33

Уникальный программный ключ:

9ba7d3e34c012eba476ffd2d064cf2781953be730df2374d16f3c0ce536f0c0

МИНОБРАЗОВАНИЯ И НАУКИ РОССИИ

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра программной инженерии

УТВЕРЖДАЮ

Проректор по учебной работе

О.Г. Локтионова

2020 г.



ВЫПОЛНЕНИЕ БАЗОВЫХ ПРОСТРАНСТВЕННЫХ ЗАПРОСОВ

Методические указания по выполнению лабораторной работы
по дисциплине «Пространственные базы данных»
для студентов направления подготовки 09.04.04 «Программная
инженерия»

Курск 2020

УДК 004.65

Составители: В.Г. Белов, Т.М. Белова

Рецензент

Кандидат технических наук, доцент кафедры программной инженерии ЮЗГУ И.Н. Ефремова

Выполнение базовых пространственных запросов: методические указания по выполнению лабораторной работы по дисциплине «Пространственные базы данных» для студентов направления подготовки 09.04.04 «Программная инженерия» / Юго-Зап. гос. ун-т; сост.: В.Г. Белов, Т.М. Белова, – Курск, 2020. – 19 с.: ил. 17.

Изложена последовательность действий с системой JDeveloper по созданию пространственных запросов при работе в СУБД Oracle.

Материал предназначен для студентов направления подготовки 09.04.04 «Программная инженерия», а также будет полезен студентам всех направлений подготовки, изучающим технологии разработки пространственных баз данных.

Текст печатается в авторской редакции.

Подписано в печать 18.04.20. Формат 60x84 1/16.
Усл. печ. л. 1,1. Уч.-изд. л. 1. Тираж 100 экз. Заказ 4449. Бесплатно.
Юго-Западный государственный университет.
305040, Курск, ул.50 лет Октября, 94.

Содержание

1 Цель лабораторной работы	4
2 Порядок выполнения лабораторной работы	5
3 Содержание отчета по лабораторной работе	17
4 Вопросы к защите лабораторной работы	18

1 Цель лабораторной работы

Целью лабораторной работы является приобретение знаний умений и навыков при создании и выполнении базовых пространственных запросов.

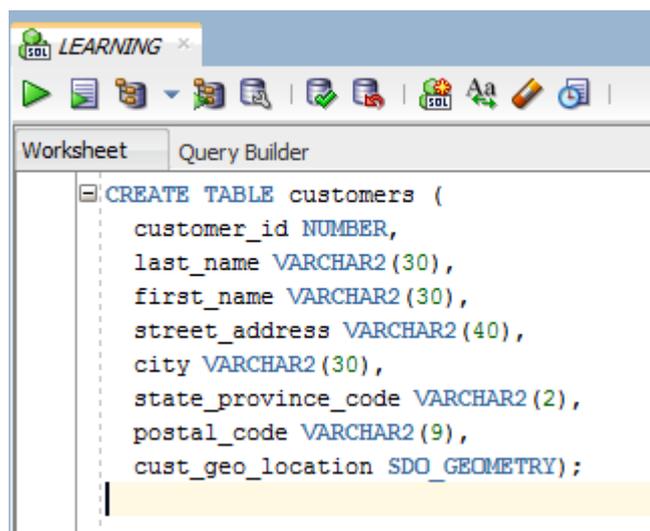
Данные действия осуществляет пользователь имеющих доступ к соответствующей базе данных. Работа выполняется с помощью программы JDeveloper, которая позволяет подключаться к СУБД Oracle.

2 Порядок выполнения лабораторной работы

Перед началом работы необходимо создать несколько таблиц, в которых будут храниться пространственные данные.

В качестве примера возьмём компанию с несколькими магазинами розничной торговли. Руководство компании хочет оповещать всех своих клиентов, находящихся поблизости от одного из магазинов, о специальных предложениях и действующих рекламных акциях. Для этого им необходимо хранить местоположение клиентов и магазинов.

Создадим таблицу CUSTOMERS, в которой будет храниться информация о клиентах (рис. 1).



```
CREATE TABLE customers (  
  customer_id NUMBER,  
  last_name VARCHAR2(30),  
  first_name VARCHAR2(30),  
  street_address VARCHAR2(40),  
  city VARCHAR2(30),  
  state_province_code VARCHAR2(2),  
  postal_code VARCHAR2(9),  
  cust_geo_location SDO_GEOMETRY);
```

Рис. 1

А также создадим таблицу STORES, в которой будет храниться информация о магазинах (рис. 2).

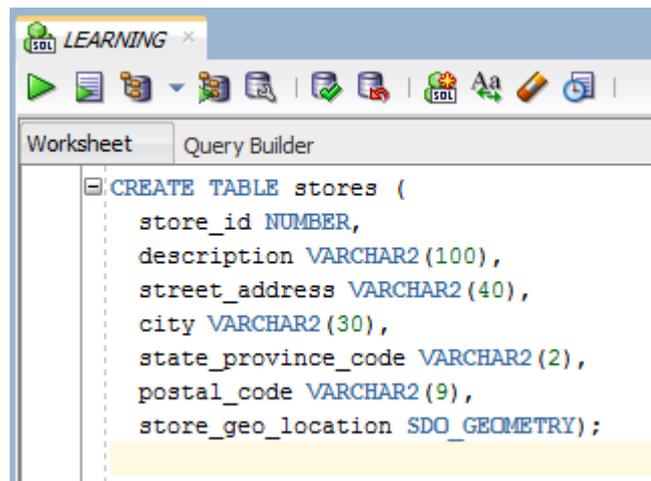


Рис. 2

Обе таблицы содержат колонки с данными типа SDO_GEOMETRY (cust_geo_location и store_geo_location), которые предназначены для хранения местоположения объектов в пространстве.

Важно помнить, что любая таблица должна содержать колонку или группу колонок, описывающую её первичных ключ (рис. 3).

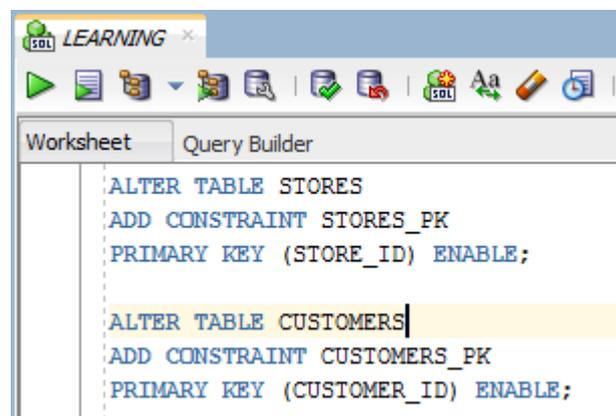


Рис. 3

Тип SDO_GEOMETRY описывается следующим образом (рис. 4):

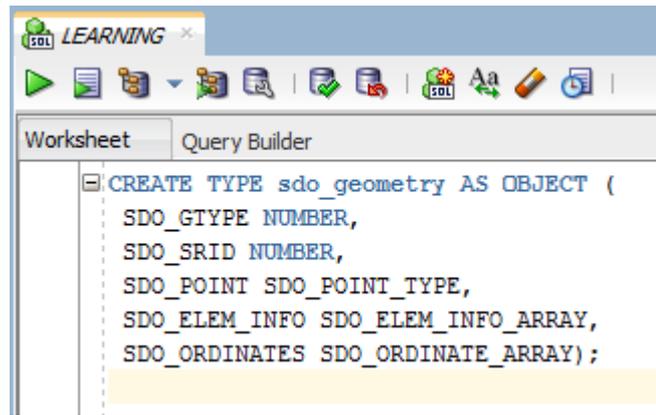


Рис. 4

`SDO_GTYPE` описывает тип геометрии объекта, представляя собой 4 цифры в формате `dltt`, где `d` - определяет количество измерений (dimensions), `l` - определяет систему измерения линейных координат (linear referencing), `tt` - определяет тип геометрии (от 00 до 07). Например, 2001 - точка в двумерном пространстве, 2002 - линия в двумерном пространстве, 2003 - многоугольник в двумерном пространстве.

`SDO_DSIR` описывает используемую систему координат.

Если `SDO_SRID = null`, то геометрия не связана с системой координат.

Если `SDO_SRID != null`, то оно должно содержать значение из таблицы `SDO_COORD_REF_SYS` колонки `SRID`, а также это значение должно содержаться в таблице `USER_SDO_GEOM_METADATA` в колонке `SRID`.

Например, `SRID = 8307` означает использование стандарта WGS84, т.е. использование простых географических координат (широта и долгота).

`SDO_POINT` используется только для описания точек в пространстве. Использует тип `SDO_POINT_TYPE`, который содержит числа X, Y, Z .

Если поля `SDO_ELEM_INFO = null` и `SDO_ORDINATES = null`, а `SDO_POINT != null`, то значения X и Y используются как координаты точки.

Во всех остальных случаях значение `SDO_POINT` игнорируется. Рекомендуется хранить координаты точек именно в поле `SDO_POINT`.

`SDO_ELEM_INFO` – описывает то, как необходимо использовать массив чисел `SDO_ORDINATES`.

Использует тип `SDO_ELEM_INFO_ARRAY`, содержащий следующие значения:

- `SDO_STARTING_OFFSET` – описывает с какого элемента нужно начать обработку массива `SDO_ORDINATES`. Нумерация в массиве начинается с 1, не с 0.
- `SDO_ETYPE` – описывает тип элемента.
- `SDO_INTERPRETATION` – описывает механизм обработки массива `SDO_ORDINATES`. Если в `SDO_ETYPE` указан не составной элемент, то этот элемент описывает, как интерпретируется система ординат для данного элемента. Если в `SDO_ETYPE` указан составной элемент, то это поле описывает количество последовательностей из трёх чисел, описывающих фигуру.

`SDO_ORDINATES` описывает массив чисел (до 1048576), служащий для описания пространственного объекта. Значения в массиве делятся на группы в зависимости от размерности фигуры. Например в двумерном пространстве $\{ x_1, y_1, x_2, y_2, x_3, y_3, \dots \}$.

Теперь запишем данные о магазинах и пользователях в таблицы (рис. 5-6).



The screenshot shows the Oracle SQL Developer interface. The title bar reads 'LEARNING'. The main window is titled 'Worksheet' and 'Query Builder'. The query text is as follows:

```
INSERT INTO customers VALUES
(1001,'Nichols', 'Alexandra',
'17 Maple Drive', 'Nashua', 'NH','03062',
SDO_GEOMETRY(2001, 8307,
SDO_POINT_TYPE (-71.48923,42.72347,NULL), NULL, NULL));

INSERT INTO customers VALUES
(1002,'Harris', 'Melvin',
'5543 Harrison Blvd', 'Reston', 'VA', '20190',
SDO_GEOMETRY(2001, 8307,
SDO_POINT_TYPE(-70.120133,44.795766,NULL), NULL, NULL));

INSERT INTO customers VALUES
(1003,'Chang', 'Marian',
'294 Main St', 'Concord', 'MA','01742',
SDO_GEOMETRY(2001, 8307,
SDO_POINT_TYPE (-71.351,42.4598,NULL), NULL, NULL));

INSERT INTO customers VALUES
(1004,'Williams', 'Thomas',
'84 Hayward Rd', 'Acton', 'MA','01720',
SDO_GEOMETRY(2001, 8307,
SDO_POINT_TYPE (-71.4559,42.4748,NULL), NULL, NULL));

INSERT INTO customers VALUES
(1005,'Rodriguez', 'Carla',
'9876 Pine Lane', 'Sudbury', 'MA','01776',
SDO_GEOMETRY(2001, 8307,
SDO_POINT_TYPE (-71.4242,42.3826,NULL), NULL, NULL));

INSERT INTO customers VALUES
(1006,'Adnani', 'Ramesh',
'1357 Appletree Ct', 'Falls Church', 'VA','22042 ',
SDO_GEOMETRY(2001, 8307,
SDO_POINT_TYPE (-77.1745,38.88505,NULL),NULL,NULL));
```

Рис. 5

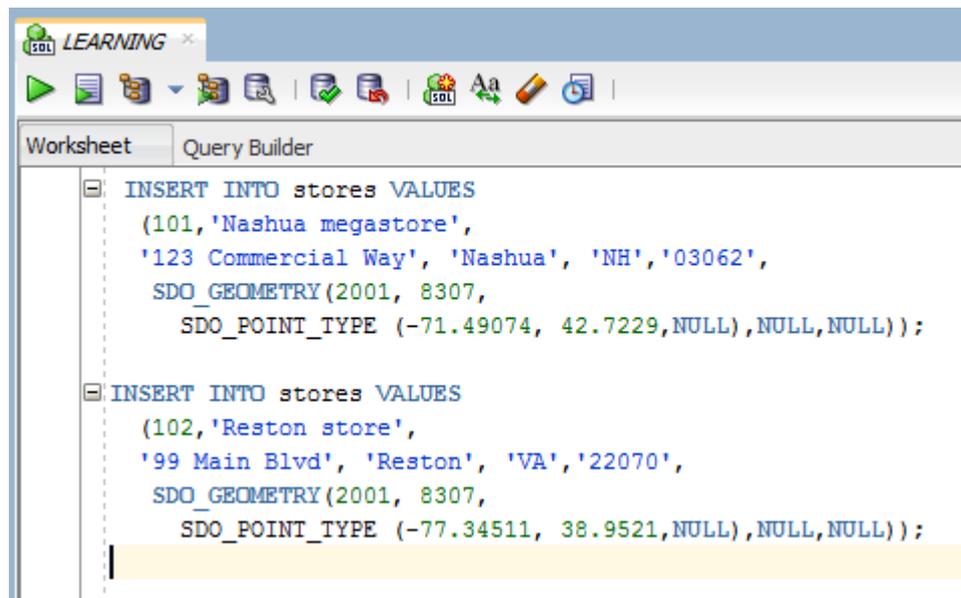


Рис. 6

Для каждой пространственной колонки (типа SDO_GEOMETRY), вы должны занести соответствующую строку таблицы USER_SDO_GEOM_METADATA (рис. 7), для обозначения пространственной информации о зоне, в которой находятся данные.

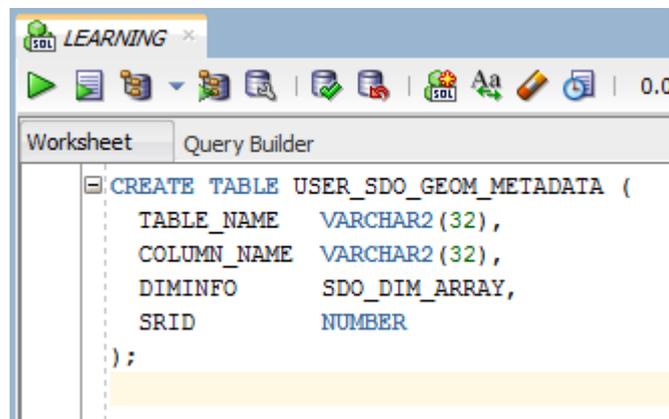


Рис. 7

USER_SDO_GEOM_METADATA содержит следующие колонки:

- TABLE_NAME – имя таблицы;
- COLUMN_NAME – имя колонки, содержащей пространственные данные, в таблице;

- SRID – значение, определяющее используемую систему координат;
- DIMINFO – служит для описания ограничений для каждого измерения.

DIMINFO представляет собой массив элементов (рис. 8) типа SDO_DIM_ELEMENT, размер которого зависит от количества используемых измерений.

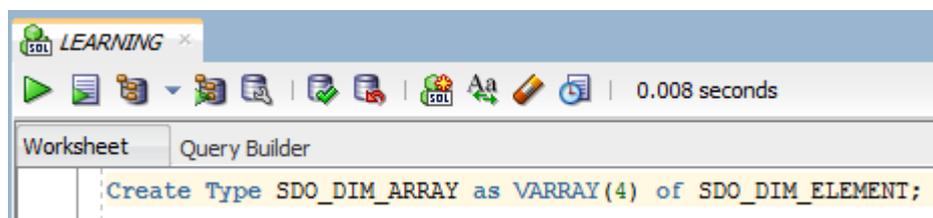


Рис. 8

Тип данных SDO_DIM_ELEMENT описывается следующим образом (рис. 9):

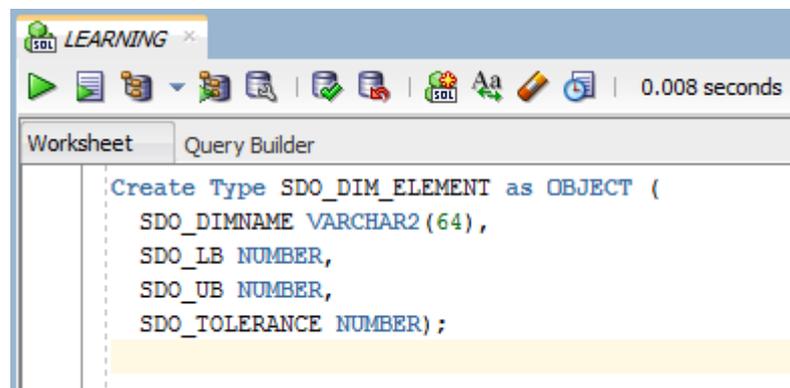
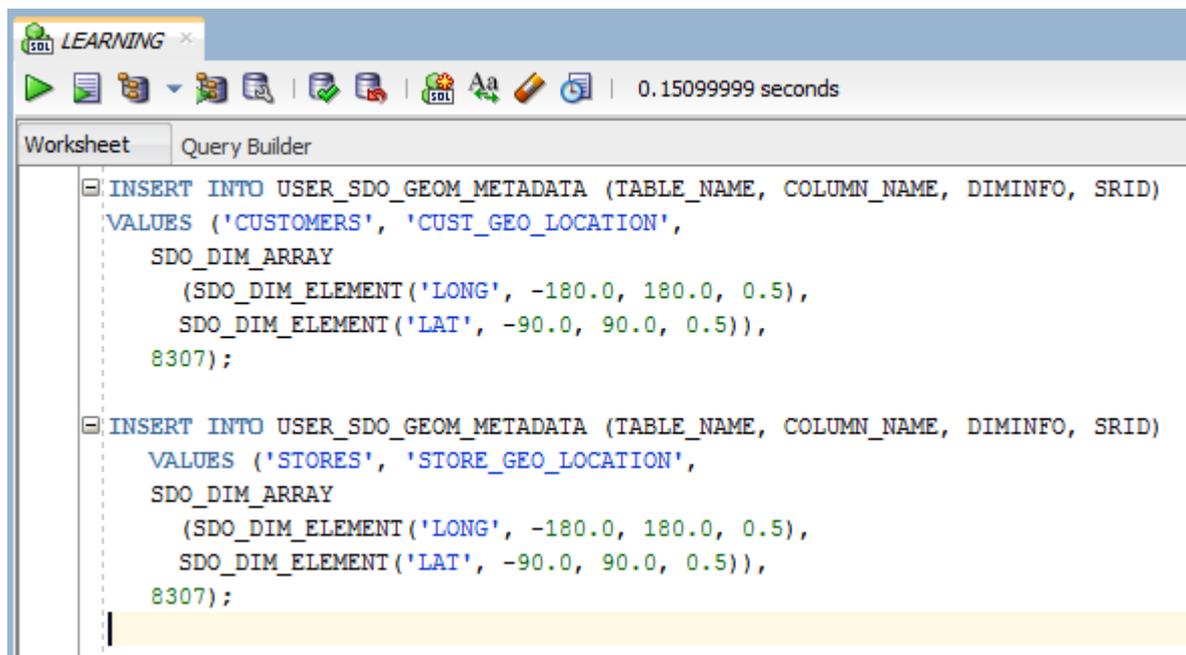


Рис. 9

- SDO_DIMNAME – название величины;
- SDO_LB – нижняя граница значения;
- SDO_UB – верхняя граница значения;

- SDO_TOLERANCE – расстояние, относительно которого две точки будут считаться одной или разными (должно быть больше 0);

Теперь занесём данные в таблицу USER_SDO_GEOM_METADATA (рис. 10).



```
LEARNING x
0.15099999 seconds
Worksheet Query Builder
INSERT INTO USER_SDO_GEOM_METADATA (TABLE_NAME, COLUMN_NAME, DIMINFO, SRID)
VALUES ('CUSTOMERS', 'CUST_GEO_LOCATION',
       SDO_DIM_ARRAY
       (SDO_DIM_ELEMENT('LONG', -180.0, 180.0, 0.5),
        SDO_DIM_ELEMENT('LAT', -90.0, 90.0, 0.5)),
       8307);
INSERT INTO USER_SDO_GEOM_METADATA (TABLE_NAME, COLUMN_NAME, DIMINFO, SRID)
VALUES ('STORES', 'STORE_GEO_LOCATION',
       SDO_DIM_ARRAY
       (SDO_DIM_ELEMENT('LONG', -180.0, 180.0, 0.5),
        SDO_DIM_ELEMENT('LAT', -90.0, 90.0, 0.5)),
       8307);
```

Рис. 10

Пространственные индексы необходимы для большинства пространственных запросов. Прежде чем использовать функции или выполнять запросы с пространственными данными создайте пространственный индекс для каждой пространственной колонки (рис. 11).

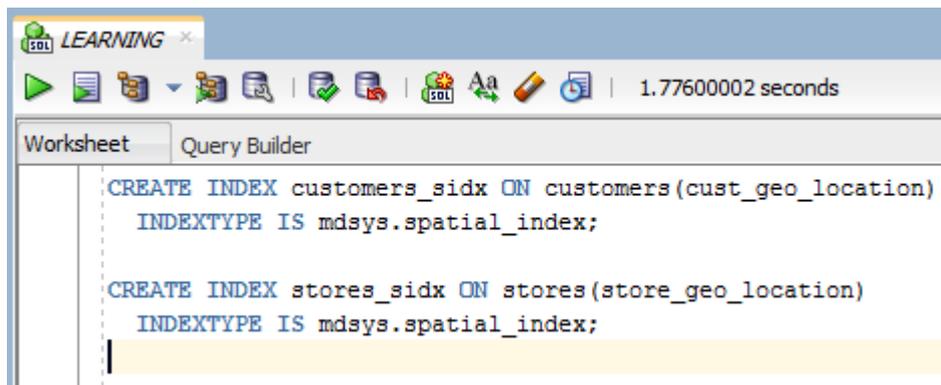


Рис. 11

После того как вы создали таблицы, заполнили таблицы, обновили метаданные и создали пространственные индексы, вы можете использовать операторы и функции, для выполнения пространственных запросов.

Запрос для получения трёх ближайших клиентов, находящихся рядом с магазином 101 (рис. 12).

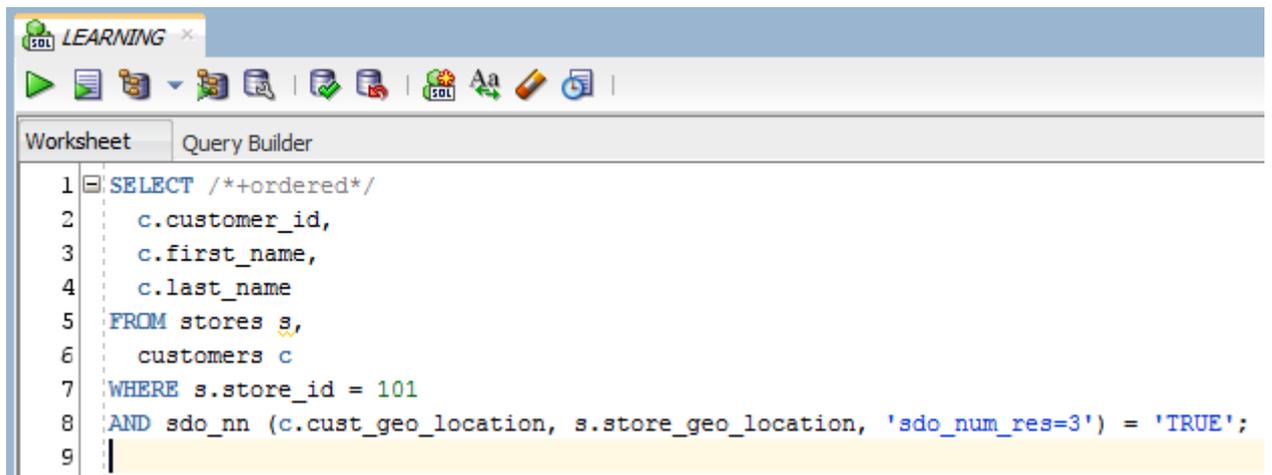


Рис. 12

Результат выполнения запроса представлен на рисунке 13.

CUSTOMER_ID	FIRST_NAME	LAST_NAME
1	1001 Alexandra	Nichols
2	1003 Marian	Chang
3	1004 Thomas	Williams

Рис. 13

Описание запроса:

- `/*+ordered*/` - пометка для оптимизатора, что поиск будет производиться по таблице `STORES` в первую очередь.
- Первый аргумент функции `SDO_NN` (`c.cust_geo_location`) это колонка для поиска.
- Второй аргумент функции `SDO_NN` (`s.storeh_geo_location`) это значение по отношению к которому вы производите поиск.
- Третий аргумент функции `SDO_NN` (`'sdo_num_res=3'`) в данном примере указывает на максимальное количество результатов.

Запрос для получения ближайших к магазину 101 клиентов, упорядоченных по дальности (рис. 14).

```

SELECT /*+ordered*/
  c.customer_id,
  c.first_name,
  c.last_name,
  sdo_nn_distance (1) distance
FROM stores s,
  customers c
WHERE s.store_id = 101
AND sdo_nn
  (c.cust_geo_location, s.storeh_geo_location, 'sdo_num_res=3', 1) = 'TRUE'
ORDER BY distance;

```

Рис. 14

Результат выполнения запроса представлен на рисунке 15.

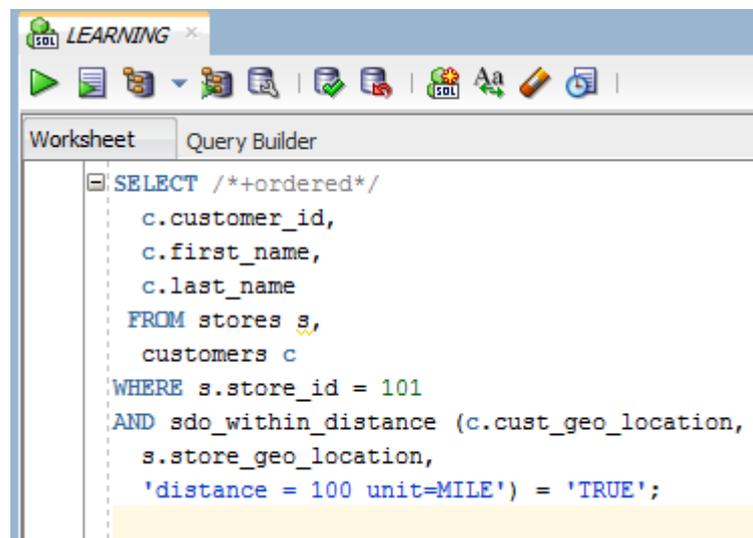
	CUSTOMER_ID	FIRST_NAME	LAST_NAME	DISTANCE
1	1001	Alexandra	Nichols	138.944860336828
2	1004	Thomas	Williams	27708.0945853743
3	1003	Marian	Chang	31396.4521207523

Рис. 15

Описание запроса:

- SDO_NN_DISTANCE это дополнительный оператор SDO_NN, и который используется только вместе с ним. Единственный аргумент (в примере это 1) это тэг, соответствующий четвёртому параметру функции SDO_NN, который служит для вывода вычисленного расстояния. Если данные о местоположении хранятся в виде широты и долготы, то по умолчанию расстояние будет в метрах;
- SDO_NN может принимать значение единиц измерения расстояния дополнительным параметром (в данном примере не рассматривается);
- команда ORDER BY distance указывает что вывод должен быть упорядочен по расстоянию, от меньшего к большему.

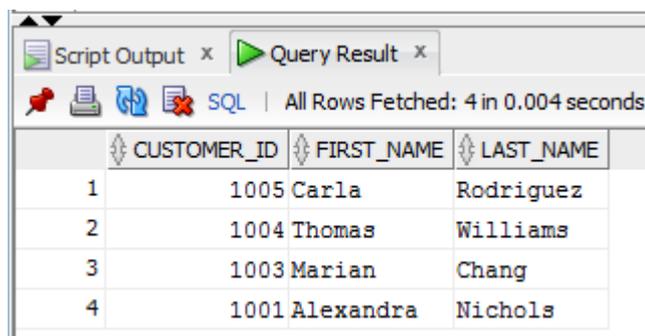
Запрос для получения клиентов, находящихся на расстоянии не более 100 миль (рис. 16).



```
SELECT /*+ordered*/
  c.customer_id,
  c.first_name,
  c.last_name
FROM stores s,
  customers c
WHERE s.store_id = 101
AND sdo_within_distance (c.cust_geo_location,
  s.store_geo_location,
  'distance = 100 unit=MILE') = 'TRUE';
```

Рис. 16

Результат выполнения запроса представлен на рисунке 17.



CUSTOMER_ID	FIRST_NAME	LAST_NAME
1	1005 Carla	Rodriguez
2	1004 Thomas	Williams
3	1003 Marian	Chang
4	1001 Alexandra	Nichols

Рис. 17

Описание запроса:

- функция `SDO_WITHIN_DISTANCE` возвращает клиентов находящихся в районе 100 миль от магазина 101. Первый параметр (`c.cust_geo_location`) колонка для поиска. Вторым параметром (`s.store_geo_location`) это место относительно, которого необходимо определить расстояние. Результат запроса не отсортирован;
- значение `distance` определяет расстояние для поиска;
- значение `unit` определяет единицы измерения расстояния `distance`. По умолчанию используются метры, в данном примере это мили.

3 Содержание отчета по лабораторной работе

В сводный отчет по лабораторным работам в качестве одного из разделов или подразделов включаются скриншоты, показывающие созданные таблицы, их содержимое, а также результаты выполнения пространственных запросов.

4 Вопросы к защите лабораторной работы

1. Какой тип данных используется для хранения пространственных данных?
2. Каким требованиям должна удовлетворять таблица, для выполнения пространственных запросов.
3. Какие аргументы принимает функция SDO_NN?
4. Какие аргументы принимает функция SDO_WITHIN_DISTANCE?