Федеральное агентство связи

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательно учреждение высшего образования

«Московский технический университет связи и информатики»

Кафедра

Интеллектуальные системы в управлении и автоматизации

Лабораторная работа №2

По дисциплине

«Технологии баз данных»

Выполнила студентка группы:

БСТ1702

Нестерова Е.А

Москва

2020

Цель работы: Создание таблиц и формирование запросов в СУБД Oracle.

Задачи:

Построить таблицы, заполнить их информацией, сформировать по 2 запроса следующих типов:

- выборка всех данных
- запрос с условием (where)
- запрос с использованием синонимов
- с использованием подзапроса
- с использованием сортировки
- с использованием группировки
- с использованием конкатенации
- выборка неповторяющихся записей
- с использованием агрегатных функций
- с использованием NOT

Запросы сохранить в виде представлений.

Теоретические сведения:

Таблицы — основные единицы хранилища в базе данных Oracle. Таблица— это логическая сущность, которая делает чтение и манипуляции данных интуитивно понятными для пользователя. Таблица состоит из столбцов и строк, причем строка соответствует одиночной записи, которая состоит из набора полей. Когда вы создаете таблицу, то присваиваете ей имя и определяете набор столбцов, относящихся к ней. Каждый столбец имеет имя и определенный тип данных.

В базах данных Oracle можно создавать таблицы двух типов: реляционные и объектные.

Реляционные таблицы — это базовые табличные структуры, которые состоят из строк и столбцов, хранящих данные.

Объектные таблицы используют объектные типы для определений столбцов и служат для хранения экземпляров объектов определенного типа.

Представление (view) – это один из объектов БД, представляющий собой хранимый запрос, с которым можно работать, как с таблицей, т.е. добавлять, редактировать, удалять данные. Это удобно, т.к. чаще всего используются данные многих таблиц. Обычно сначала создаются представления, а потом на их основе выводятся данные для пользователя в клиентском приложении. В таком случае, пользователю можно дать полный доступ на какое-то представление, не давая доступа на таблицу.

Выполнение:

1. Создание и заполнение таблиц

Самостоятельно создаем 2 таблицы с данными при помощи функции create table и заполняем их. Одна таблица будет соответствовать перечню заказа магазина с полями «Идентификатор», «Тип товара», «Фирма производитель», «Количество в заказе» и «Цена всей партии», а другая - отображать расположение складов определенных производителей с полями «Идентификатор», «Фирма производитель», «Город расположения склада».

```
SQL> create table shop
  2  (ID number,
  3  TYPE varchar(20),
  4  FIRMA varchar(20),
  5  KOLVO number,
  6  PRICE number
  7  )
  8  /

Table created.
```

```
SQL> create table sklads
  2  (ID number,
  3  FIRMA varchar(20),
  4  CITY varchar(20)
  5  )
  6  /

Table created.
```

Заполняем таблицы соответствующими данными:

```
SQL> insert into sklads (ID, FIRMA, CITY)
  2  values(4, 'apple', 'Spb')
  3  /

1 row created.

SQL> insert into sklads (ID, FIRMA, CITY)
  2  values(5, 'lg', 'Kazan')
  3  /

1 row created.

SQL> select * from sklads
  2  /

        ID FIRMA                CITY
---------- -------------------- --------------------
         1 sony                 Moskva
         2 bosh                 Perm
         3 samsung              Spb
         4 apple                Spb
         5 lg                   Kazan
```

```
SQL> insert into shop (ID, TYPE, FIRMA, KOLVO, PRICE)
  2  VALUES(8,'4ainik','lg', 3, 9000)
  3  /

1 row created.

SQL> insert into shop (ID, TYPE, FIRMA, KOLVO, PRICE)
  2  VALUES(9,'monitor','apple', 1, 50000)
  3  /

1 row created.

SQL> insert into shop (ID, TYPE, FIRMA, KOLVO, PRICE)
  2  VALUES(10,'monitor','sony', 7, 35000)
  3  /

1 row created.

SQL> select * from shop
  2  /

        ID TYPE                FIRMA                KOLVO      PRICE
---------- ------------------- -------------------- ---------- ----------
         1 televizor           sony                          6     120000
         2 televizor           bosh                          2      80000
         3 telefon             samsung                      10     200000
         4 telefon             apple                        20     500000
         5 televizor           samsung                       1     100000
         6 4ainik              bosh                         30      30000
         7 monitor             lg                            5      20000
         8 4ainik              lg                            3       9000
         9 monitor             apple                         1      50000
        10 monitor             sony                          7      35000

10 rows selected.
```

2. Выборка всех данных

Для построения запроса используем оператор SELECT и сохраняем данные в виде представлений.

Для таблицы заказов магазина:

```
SQL> create view VShop as select* from shop
  2  /

View created.
```

```
SQL> select* from VShop
  2  /

        ID TYPE                 FIRMA                    KOLVO      PRICE
---------- -------------------- -------------------- ---------- ----------
         1 televizor            sony                          6     120000
         2 televizor            bosh                          2      80000
         3 telefon              samsung                      10     200000
         4 telefon              apple                        20     500000
         5 televizor            samsung                       1     100000
         6 4ainik               bosh                         30      30000
         7 monitor              lg                            5      20000
         8 4ainik               lg                            3       9000
         9 monitor              apple                         1      50000
        10 monitor              sony                          7      35000

10 rows selected.
```

Для таблицы складов:

```
SQL> create view VSklads as select* from sklads
  2  /

View created.

SQL> select* from VSklads
  2  /

        ID FIRMA                CITY
---------- -------------------- --------------------
         1 sony                 Moskva
         2 bosh                 Perm
         3 samsung              Spb
         4 apple                Spb
         5 lg                   Kazan
```

3. Запрос с условием (where)

Условия в предложении where может быть несколько. В таком случае используют ключевые слова:

И (and), или (or), как (like), между (between), выборка значений из списка (in), отсутствие записи/ не отсутствие (is null / is not null)

Создадим и выведем представления с запросами на вывод строк таблицы, в которых отображены телевизоры всех фирм, которые заказаны в магазине:

```
SQL> create view VShopT
  2  as select ID, TYPE, FIRMA from VShop
  3  where TYPE='televizor'
  4  /

View created.

SQL> select *from VShopT
  2  /

        ID TYPE                 FIRMA
---------- -------------------- --------------------
         1 televizor            sony
         2 televizor            bosh
         5 televizor            samsung
```

Аналогично выведем информацию о складах с идентификаторами от 3 до 5:

```
SQL> create view VSkladsId
  2  as select ID,FIRMA, CITY from VSklads
  3  where ID between 3 and 5
  4  /

View created.

SQL> select* from VSkladsId
  2  /

        ID FIRMA                CITY
---------- -------------------- --------------------
         3 samsung              Spb
         4 apple                Spb
         5 lg                   Kazan
```

4. Запрос с синонимом

Синонимы представляют собой альтернативное имя объекта, определяемое пользователем и служащее для более удобного использования при работе с именами объектов.

Создание синонимов sh и sk, для заказов магазина и перечня складов соответственно.

```
SQL> create synonym sh for VShop
  2  /

Synonym created.

SQL> create synonym sk for VSklads
  2  /

Synonym created.
```

Создадим запрос с использованием синонимов, который выведет: тип товара, фирму производителя, количество экземпляров товара, которые возьмем из таблицы магазина, и город, откуда они поставляются, который получим из таблицы складов, сопоставив значения «фирмы производителя» из двух таблиц.

```
SQL> select sh.TYPE, sk.FIRMA, sh.KOLVO, sk.CITY
  2  from VSklads sk, VShop sh
  3  where sk.FIRMA=sh.FIRMA
  4  /

TYPE                FIRMA                     KOLVO CITY
------------------- -------------------- ---------- --------------------
televizor           sony                          6 Moskva
televizor           bosh                          2 Perm
telefon             samsung                      10 Spb
telefon             apple                        20 Spb
televizor           samsung                       1 Spb
4ainik              bosh                         30 Perm
monitor             lg                            5 Kazan
4ainik              lg                            3 Kazan
monitor             apple                         1 Spb
monitor             sony                          7 Moskva

10 rows selected.
```

Создадим запрос используя синонимы, который выведет такую информацию, как: тип товара, количество и стоимость партии, о заказе, который будет доставлен со склада, расположенного в Санкт-Петербурге. Для этого сопоставим значения «фирмы производителя» из двух таблиц и поставим условие «Город = Санкт-Петербург».

```
SQL> select sh.TYPE, sh.KOLVO, sh.PRICE
  2  from VShop sh, VSklads sk
  3  where sh.FIRMA=sk.FIRMA and sk.CITY='Spb'
  4  /

TYPE                      KOLVO      PRICE
------------------- ---------- ----------
telefon                    10     200000
telefon                    20     500000
televizor                   1     100000
monitor                     1      50000
```

Сохраним запросы в виде представлений:

```
SQL> create view T1 as select sh.TYPE, sk.FIRMA, sh.KOLVO, sk.CITY
  2  from VSklads sk, VShop sh
  3  where sk.FIRMA=sh.FIRMA
  4  /

View created.

SQL> create view T2 as select sh.TYPE, sh.KOLVO, sh.PRICE
  2  from VShop sh, VSklads sk
  3  where sh.FIRMA=sk.FIRMA and sk.CITY='Spb'
  4  /

View created.
```

5. Запросы с использованием подзапросов

Запросы, в которых используются несколько предложений select. Могут быть однострочные (возвращают одну строку: в условии используются знаки равенства/ неравенства) и многострочные (подзапрос возвращает несколько строк: используется in)

Создадим представление запроса, который выведет типы товаров, которые заказаны из Санкт-Петербурга, для этого используем конструкцию, в которой определим какие фирмы расположены в Санкт-Петербурге, после чего сопоставим полученные фирмы с таблицей заказов.

```
SQL> create view t3
  2  as select TYPE from VShop
  3  where FIRMA in (select FIRMA from VSklads
  4  where CITY = 'Spb')
  5  /

View created.

SQL> select * from t3
  2  /

TYPE
--------------------
televizor
telefon
monitor
telefon
```

6. Запросы с использованием сортировки

Сортировка - последовательное расположение или разбиение на группы чего-либо в зависимости от выбранного критерия.

Создадим и выведем представление запроса, в котором выведем заказы магазина, отсортированные по количеству копий товаров.

Так же, создадим и выведем представление запроса таблицы с перечнем складов, отсортированную в алфавитном порядке по городам расположения.

```
SQL> create view t4 as select TYPE, FIRMA,KOLVO, PRICE from VShop
  2  order by KOLVO
  3  /

View created.

SQL> select* from t4
  2  /

TYPE                FIRMA                     KOLVO      PRICE
------------------- -------------------- ---------- ----------
televizor           samsung                      1     100000
monitor             apple                        1      50000
televizor           bosh                         2      80000
4ainik              lg                           3       9000
monitor             lg                           5      20000
televizor           sony                         6     120000
monitor             sony                         7      35000
telefon             samsung                     10     200000
telefon             apple                       20     500000
4ainik              bosh                        30      30000

10 rows selected.

SQL> create view t5 as select ID, FIRMA, CITY from VSklads
  2  order by FIRMA
  3  /

View created.

SQL> select* from t5
  2  /

        ID FIRMA                CITY
---------- -------------------- --------------------
         4 apple                Spb
         2 bosh                 Perm
         5 lg                   Kazan
         3 samsung              Spb
         1 sony                 Moskva
```

7. Запросы с использование группировки

В предложении group by указывается поле, котороеобязательно должно присутствовать в предложении select, в отличие отпредложения order by. Также, в предложении group by можно указывать несколько полей, в этом случае группировка будет последовательна сначала по одному полю, потом по второму.

Создадим представление запроса, который выведет стоимость заказов магазина у производителей, суммировав стоимость каждого типа товара, сгруппировав их по фирме.

```
SQL> create view t6 as select FIRMA, sum(PRICE) as ALLPRICE from VShop
  2  group by FIRMA
  3  /

View created.

SQL> select* from t6
  2  /

FIRMA                ALLPRICE
-------------------- ----------
bosh                     110000
lg                        29000
sony                     155000
samsung                  300000
apple                    550000
```

Создадим представление запроса, который выведет общее количество определенных типов товаров.

```
SQL> create view t7 as select TYPE, sum(KOLVO) as KOLVO from VShop
  2  group by TYPE
  3  /

View created.

SQL> select * from t7
  2  /

TYPE                    KOLVO
-------------------- ----------
televizor                    9
telefon                     30
4ainik                      33
monitor                     13
```

8. Запросы с использованием конкатенации(объедения)

Конкатенация позволяет нам объединять столбцы и строки в 1 предложение, добавляя новые слова.

Например:

```
SQL> select 'ТИП ТОВАРА'||' '||TYPE||' '||'ОТ ФИРМЫ'||' '||FIRMA||' '||'СТОИТ'||' '||PRICE/KOLVO
  2  from VShop
  3  /

'ТИПТОВААА'||''||TYPE||''||'ОТФИАМЫ'||''||FIRMA||''||'СТОИТ'||''||PRICE/KOLVO
--------------------------------------------------------------------------------
ТИП ТОВАРА televizor ОТ ФИРМЫ sony СТОИТ 20000
ТИП ТОВАРА televizor ОТ ФИРМЫ bosh СТОИТ 40000
ТИП ТОВАРА telefon ОТ ФИРМЫ samsung СТОИТ 20000
ТИП ТОВАРА telefon ОТ ФИРМЫ apple СТОИТ 25000
ТИП ТОВАРА televizor ОТ ФИРМЫ samsung СТОИТ 100000
ТИП ТОВАРА 4ainik ОТ ФИРМЫ bosh СТОИТ 1000
ТИП ТОВАРА monitor ОТ ФИРМЫ lg СТОИТ 4000
ТИП ТОВАРА 4ainik ОТ ФИРМЫ lg СТОИТ 3000
ТИП ТОВАРА monitor ОТ ФИРМЫ apple СТОИТ 50000
ТИП ТОВАРА monitor ОТ ФИРМЫ sony СТОИТ 5000
```

9. Запрос с выборкой неповторяющихся записей

Для наглядности добавим в таблицу заказа магазина повторяющуюся запись. После чего сначала выведем всю таблицу целиком, а потом с выборкой.

```
SQL> insert into shop (ID, TYPE, FIRMA, KOLVO, PRICE)
  2  VALUES(4,'telefon','apple', 20, 500000)
  3  /

1 row created.

SQL> select * from VShop
  2  /

        ID TYPE                 FIRMA                    KOLVO      PRICE
---------- -------------------- -------------------- ---------- ----------
         1 televizor            sony                          6     120000
         2 televizor            bosh                          2      80000
         3 telefon              samsung                      10     200000
         4 telefon              apple                        20     500000
         5 televizor            samsung                       1     100000
         6 4ainik               bosh                         30      30000
         7 monitor              lg                            5      20000
         8 4ainik               lg                            3       9000
         9 monitor              apple                         1      50000
        10 monitor              sony                          7      35000
         4 telefon              apple                        20     500000

11 rows selected.

SQL> select distinct * from VShop
  2  /

        ID TYPE                 FIRMA                    KOLVO      PRICE
---------- -------------------- -------------------- ---------- ----------
         1 televizor            sony                          6     120000
         4 telefon              apple                        20     500000
         6 4ainik               bosh                         30      30000
         5 televizor            samsung                       1     100000
         7 monitor              lg                            5      20000
         8 4ainik               lg                            3       9000
         2 televizor            bosh                          2      80000
         3 telefon              samsung                      10     200000
         9 monitor              apple                         1      50000
        10 monitor              sony                          7      35000

10 rows selected.
```

10.Запросы с применением агрегатных функций

COUNT - считает число значений в данном столбце, или число строк в таблице.

SUM – арифметическая сумма всех выбранных значений данного поля.

AVG – усредненное значение всех выбранных значений данного поля.

MAX - наибольшее из всех выбранных значений данного поля.

MIN - производит наименьшее из всех выбранных значений данного поля.

Создадим представление запроса, в котором выведем общее количество товаров, сгруппировав их по типу.

```
SQL> create view t7 as select TYPE, sum(KOLVO) as KOLVO from VShop
  2  group by TYPE
  3  /

View created.

SQL> select * from t7
  2  /

TYPE                      KOLVO
-------------------- ----------
televizor                  9
telefon                   30
4ainik                    33
monitor                   13
```

Создадим представление запроса, в котором выведется тип товара с наибольшим количеством экземпляров.

```
SQL> create view t8 as select TYPE, KOLVO, PRICE from VShop
  2  where KOLVO=(select max(KOLVO) from VShop)
  3  /

View created.

SQL> select * from t8
  2  /

TYPE                      KOLVO      PRICE
-------------------- ---------- ----------
4ainik                        30      30000
```

11.Запросы с применением NOT

Not позволяет отсортировать строчки с незаполненными полями при выводе.

Для наглядности создадим пустую строку в таблице складов. После чего создадим представление, в котором выведем таблицу фирм, исключив строку, в которой не указан город расположения склада, данной фирмы.

```
SQL> insert into sklads (ID, FIRMA, CITY)
  2  values('6','','')
  3  /

1 row created.

SQL> create view t9 as select ID, FIRMA from VSklads where CITY is not NULL
  2  /

View created.

SQL> select * from t9
  2  /

        ID FIRMA
---------- --------------------
         1 sony
         2 bosh
         3 samsung
         4 apple
         5 lg
```

Вывод: в ходе выполнения лабораторной работы были получены навыки создания, заполнения и форматирования таблиц данных в SQL, знания о создании представлений и ввода сложных запросов, изучены функции сортировки, группировки и объединения информации в базах данных. Полученный опыт облегчит дальнейшую работу в СУБД Oracle.