

1. В базе данных хранятся 3 таблицы. Первая содержит данные о товарах. Вторая содержит данные о местах хранения. Третья таблица содержит данные о движении товара на местах хранения.

Поля таблицы 1, Items: ID, Name, Serial\_Number;


Поля таблицы 2, Warehouses: ID, Name, Address, MOL;

Поля таблицы 3, Details: ID, Item\_ID, Warehouse\_ID, Count;

```
create table Items(  
    ID int,  
    Name varchar2(1000),  
    Serial_number int  
)
```

```
insert into Items (ID, Name, Serial_number)  
VALUES
```

```
(1, 'iphone', 230),  
(2, 'watch', 244),  
(3, 'headphones', 150),  
(4, 'tv', 330),  
(5, 'vacuum', 111),  
(6, 'laptop', 222),  
(7, 'desktop', 333),  
(8, 'cleaner', 415);
```

	 id integer	 name character varying (1000)	 serial_number integer
1	1	iphone	230
2	2	watch	244
3	3	headphones	150
4	4	tv	330
5	5	vacuum	111
6	6	laptop	222
7	7	desktop	333
8	8	cleaner	415

```
create table Warehouses(  
    ID int,  
    Name varchar2(1000),  
    Address varchar2(1000),  
    MOL varchar2(1000)  
)
```

```
insert into Warehouses (ID, Name, Address, MOL)  
VALUES
```

```

(1, 'Manas', 'Manasa 30', ''),
(2, 'Zholdasbekova', 'Zholdasbekova 100', ''),
(3, 'Abay', 'Abay 15', ''),
(4, 'Dostyk', 'Dostyk 20', ''),
(5, 'ilias', 'ilias 13', '');

```

	id integer	name character varying (1000)	address character varying (1000)	mol character varying (1000)
1	1	Manas	Manasa 30	
2	2	Zholdasbekova	Zholdasbekova 100	
3	3	Abay	Abay 15	
4	4	Dostyk	Dostyk 20	
5	5	ilias	ilias 13	

```

create table Details(
  ID int,
  Item_id int,
  Warehouse_id int,
  Count int
)
insert into Details (ID, Item_id, Warehouse_id, Count)
VALUES
(1, 2, 2, 400),
(2, 1, 3, 20),
(3, 4, 5, 300),
(4, 6, 4, 150),
(1, 4, 5, 400),
(2, 2, 1, 20),
(3, 6, 4, 300),
(4, 7, 2, 150);

```

	id integer	item_id integer	warehouse_id integer	count integer
1	1	2	2	400
2	2	1	3	20
3	3	4	5	300
4	4	6	4	150
5	5	7	1	407
6	1	4	5	400
7	2	2	1	20
8	3	6	4	300
9	4	7	2	150
10	5	8	3	407

А) Написать скрипт, в результате выполнения которого будет выводиться информация об остатках товаров на складах, наименовании складов, наименовании товаров, серийный номер товаров и MOL складов.

SELECT i.Name, i.Serial\_number, w.Name, w.MOL, d.Count FROM Items i INNER JOIN Details d ON i.ID = d.Item\_id INNER JOIN Warehouses w ON w.ID = d.Warehouse\_id

	name character varying (1000)	serial_number integer	name character varying (1000)	mol character varying (1000)	count integer
1	watch	244	Zholdasbekova		400
2	iphone	230	Abay		20
3	tv	330	ilias		300
4	laptop	222	Dostyk		150
5	desktop	333	Manas		407
6	tv	330	ilias		400
7	watch	244	Manas		20
8	laptop	222	Dostyk		300
9	desktop	333	Zholdasbekova		150
10	cleaner	415	Abay		407

Б) По заданию А создать View.

Create View All\_information As SELECT i.Name as Item\_name, i.Serial\_number, w.Name as Warehouse\_name, w.MOL, d.Count FROM Items as i INNER JOIN Details as d ON i.ID = d.Item\_id INNER JOIN Warehouses w ON w.ID = d.Warehouse\_id

	item_name character varying (1000)	serial_number integer	warehouse_name character varying (1000)	mol character varying (1000)	count integer
1	watch	244	Zholdasbekova		400
2	iphone	230	Abay		20
3	tv	330	ilias		300
4	laptop	222	Dostyk		150
5	desktop	333	Manas		407
6	tv	330	ilias		400
7	watch	244	Manas		20
8	laptop	222	Dostyk		300
9	desktop	333	Zholdasbekova		150
10	cleaner	415	Abay		407

В) Написать скрипт, в результате выполнения которого будет выводиться информация о товарах, остатки которого на складах превышают 500единиц

SELECT i.Name as Item\_name, i.Serial\_number, w.Name as Warehouse\_name, w.MOL, d.Count FROM Items as i INNER JOIN Details as d ON i.ID = d.Item\_id INNER JOIN Warehouses w ON w.ID = d.Warehouse\_id ORDER BY d.Count HAVING d.Count > 500

- Написать PL/SQL процедуру, которая с помощью команды dbms\_output.put\_line выводит содержание первых 2-х строк из таблицы первого задания.

```
create or replace procedure print_elem
as
CURSOR i_id is
SELECT id FROM Items;
type i_list is varray (200) of Items.id% type;
id_list i_list := i_list();
CURSOR n_name is
SELECT name FROM Items;
type n_list is varray (200) of Items.name% type;
```

```

name_list n_list := n_list();
CURSOR s_number is
SELECT serial_number FROM Items;
type s_list is varray (200) of Items.serial_number%type;
serial_list s_list := s_list();
i_counter integer := 0;
n_counter integer := 0;
s_counter integer := 0;
a_counter integer := 0;
begin
FOR n IN i_id LOOP
i_counter := i_counter + 1;
id_list.extend;
id_list(i_counter) := n.id;
END LOOP;
FOR n IN n_name LOOP
n_counter := n_counter + 1;
name_list.extend;
name_list(n_counter) := n.name;
END LOOP;
FOR n IN s_number LOOP
s_counter := s_counter + 1;
serial_list.extend;
serial_list(s_counter) := n.serial_number;
END LOOP;
FOR n IN s_number LOOP
a_counter := a_counter + 1;
dbms_output.put_line('ID: '||id_list(a_counter)||', '||'Name: '||
name_list(a_counter)||', Serial_number: '|| serial_list(a_counter));
END LOOP;
end;
/
begin
    print_elem
end;

```

3. Напишите два любых запроса, к таблице 3 первого задания, используя аналитический функции

```

SELECT DISTINCT FIRST_VALUE(Count) Over (ORDER BY Count DESC) as
Highest_value FROM Details

```

```

SELECT VARIANCE(Count) FROM Details

```

4. Создать таблицу MY\_TEST\_TABLE с 5 полями: MY\_CLOB(CLOB), MY\_STRING (VARCHAR2), ID (NUMBER), MY\_DATE (DATE), PARENT\_ID (NUMBER).

```
Create Table MY_TEST_TABLE(
    my_clob clob,
    my_string varchar2(200),
    id number,
    my_date date,
    parent_id number
)
```

Поле ID сделайте PRIMARY KEY.

```
ALTER TABLE MY_TEST_TABLE ADD PRIMARY KEY (id);
```

Создать индекс на поле MY\_STRING.

```
Create Index My_string ON MY_TEST_TABLE(my_string)
```

5. Создать триггер на таблицу MY\_TEST\_TABLE, в котором будет заполняться ID из созданного ранее Sequence.

```
CREATE OR REPLACE TRIGGER test_table_trigger
AFTER INSERT
    ON MY_TEST_TABLE
    FOR EACH ROW
DECLARE
    v_id number;
BEGIN
    -- Найти персону username, осуществляющего INSERT в таблицу
    SELECT dbms_random.value(1,10) INTO v_id
    FROM dual;
    -- Вставить строку в таблицу audit
    INSERT INTO MY_TEST_TABLE
    (id)
    VALUES
    (v_id);
END;
```

6. Заполнить таблицу MY\_TEST\_TABLE 10 записями.

```
INSERT INTO MY_TEST_TABLE (id, my_clob, my_string, my_date, parent_id)
VALUES
(1,'qwerty','qwerty','2004-01-22',2),
(2,'asd','asd','2005-01-12',2),
(3,'zxc','zxc','2008-03-22',2),
(4,'qwe','qwe','2010-04-02',2),
(5,'rty','rty','2020-05-12',2),
(6,'vbn','vbn','2021-02-22',2),
(7,'iop','iop','2015-08-22',2),
(8,'jkl','jkl','2017-09-22',2),
(9,'fgh','fgh','2018-10-22',2),
```

```
(10,'qwe','qwe','2019-12-22',2);
```

Выбрать все данные из таблицы MY\_TEST\_TABLE в виде отформатированного дерева с помощью иерархического запроса.

```
SELECT my_clob, my_string, my_date FROM MY_TEST_TABLE CONNECT BY PRIOR  
id = parent_id;
```