Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

Кафедра вычислительной техники

Системы баз данных

Лабораторная работа № 1

Выполнили: Разумовская А.В.

Байрамов С.И.

Группа: Р3317

Преподаватель: Беликов П. А.

Санкт-Петербург

2016 год

Лабораторная работа по дисциплине

«Системы баз данных»

Этап 1

**1. Вывести содержимое всех столбцов таблицы Н\_СЕССИЯ, поля типа DATE представить в формате DD.MM.YYYY HH24:MI:SS**

select ИД, СЭС\_ИД, ЧЛВК\_ИД,

to\_char(ДАТА, 'DD.MM.YY') as ДАТА,

to\_char(ВРЕМЯ, 'HH24:MI:SS') as ВРЕМЯ,

АУДИТОРИЯ,

to\_char(ДАТА\_К, 'DD.MM.YY') as ДАТА\_К,

to\_char(ВРЕМЯ\_К, 'HH24:MI:SS') as ВРЕМЯ\_К,

АУДИТОРИЯ\_К,

УЧГОД,

ГРУППА,

СЕМЕСТР,

КТО\_СОЗДАЛ,

to\_char(КОГДА\_СОЗДАЛ, 'DD.MM.YY') as КОГДА\_СОЗДАЛ,

КТО\_ИЗМЕНИЛ,

to\_char(КОГДА\_ИЗМЕНИЛ, 'DD.MM.YY') as КОГДА\_ИЗМЕНИЛ

from Н\_СЕССИЯ;

**2. Вывести неповторяющиеся наименования дисциплин из таблицы Н\_ДИСЦИПЛИНЫ**

SELECT DISTINCT НАИМЕНОВАНИЕ

from Н\_ДИСЦИПЛИНЫ;

**3. Вывести округленное значение разницы между днем рождения произвольной персоны из Н\_ЛЮДИ\*\* и началом текущего учебного года.**

select ROUND((

select НАЧАЛО from Н\_УЧЕБНЫЕ\_ГОДА

where КОНЕЦ>=SYSDATE

AND НАЧАЛО<=SYSDATE)

-(

select ДАТА\_РОЖДЕНИЯ

from Н\_ЛЮДИ

where ИД=104990)) as РАЗНИЦА

from DUAL;

**4. Получить фамилии и инициалы людей (в виде «Иванов И.И.»), которые родились в том же месяце, что и произвольная персона из Н\_ЛЮДИ\*\*.**

select ФАМИЛИЯ || ' ' ||

SUBSTR(ИМЯ, 1, 1) || '. ' ||

SUBSTR(ФАМИЛИЯ, 1, 1) || '. ' as ФИО

from Н\_ЛЮДИ

where to\_Char(ДАТА\_РОЖДЕНИЯ, 'mon')=(

select to\_Char(ДАТА\_РОЖДЕНИЯ, 'mon')

from Н\_ЛЮДИ

where ИД=182790);

**5. Вывести фамилии, имена, отчества и номера (ИД) людей, фамилии которых начинаются на те же 2 буквы, что и у произвольной персоны из Н\_ЛЮДИ\*\*, упорядочить по убыванию ФИО, вывести не более 75 строк.**

select ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО, ИД

from Н\_ЛЮДИ

where SUBSTR(ФАМИЛИЯ,1,2)=SUBSTR((

select ФАМИЛИЯ

from Н\_ЛЮДИ

where ИД=182730)

,1,2)

AND ROWNUM<=75

order by ФАМИЛИЯ DESC;

**6. Вывести список персон (фамилия, имя, отчество, ИД), для которых инициалы не равны ‘А’, ‘Б’, ‘З’ и ‘К’, ‘У’. Представить не менее 2х вариантов запроса.**

select ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО, ИД

from Н\_ЛЮДИ

where SUBSTR(ИМЯ,1,1)<>'А' AND SUBSTR(ИМЯ,1,1)<>'Б' AND SUBSTR(ИМЯ,1,1)<>'З'

AND SUBSTR(ОТЧЕСТВО,1,1)<>'К' AND SUBSTR(ОТЧЕСТВО,1,1)<>'У';

select ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО, ИД

from Н\_ЛЮДИ

where SUBSTR(ИМЯ,1,1)<>ALL('А','Б','З')

AND SUBSTR(ОТЧЕСТВО,1,1)<>ALL('К','У');

**7. Вычислите количество персон, имя которых такое же, как у произвольной персоны из Н\_ЛЮДИ\*\*, учесть возможность наличия дубликатов.**

select COUNT(ИД) as КОЛИЧЕСТВО

from(select distinct ИД from(

select \*

from Н\_ЛЮДИ

where ИМЯ=(select ИМЯ from Н\_ЛЮДИ where ИД=152216)));

**8. Вывести, используя таблицу Н\_ВЕДОМОСТИ, удвоенные (значение\_оценки\*2) оценки произвольной персоны из Н\_ЛЮДИ\*\*. Использовать NOT IN и регулярные выражения.**

select ОЦЕНКА\*2 as ДВОЙНАЯ\_ОЦЕНКА

from Н\_ВЕДОМОСТИ

where ЧЛВК\_ИД=152216 AND ОЦЕНКА NOT IN('зачет')

AND NOT REGEXP\_LIKE(ОЦЕНКА, 'незач');

**9. Вывести, используя таблицу Н\_ВЕДОМОСТИ, сумму оценок 7и произвольных персон из Н\_ЛЮДИ\*\*. 7 персон задаются условием к таблице Н\_ЛЮДИ.**

select SUM(ОЦЕНКА) as СУММА\_7

from (select \* from Н\_ВЕДОМОСТИ

where ЧЛВК\_ИД IN

(select ИД from Н\_ЛЮДИ

where to\_Char(ДАТА\_РОЖДЕНИЯ, 'mon')='apr' AND ИД>150000 AND ROWNUM<=7))

where ОЦЕНКА NOT IN('зачет','незач');

**10. Получить декартово соединение N таблиц, где N равно 3й цифре вашего табельного номера.**

select \*

FROM Н\_УЧЕБНЫЕ\_ГОДА, Н\_КВАЛИФИКАЦИИ;

**11. Вывести, используя таблицу Н\_ВЕДОМОСТИ, среднюю оценку 7и произвольных персон из Н\_ЛЮДИ\*\* , их фамилии, имена и отчества, при этом средняя оценка не должна быть наибольшей среди всех персон с такой же фамилией.**

select ТАБЛ\_МАКС.ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО, СРЕД, МАКС

from (

select ФАМИЛИЯ, MAX(СРЕД) МАКС

from (

select ЧЛВК\_ИД, AVG(ОЦЕНКА) СРЕД

from Н\_ВЕДОМОСТИ

where regexp\_like(ОЦЕНКА,'[0-9]')

group by ЧЛВК\_ИД) СРЕД\_ЗНАЧ,

Н\_ЛЮДИ

where ЧЛВК\_ИД=Н\_ЛЮДИ.ИД

group by ФАМИЛИЯ) ТАБЛ\_МАКС,

(select ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО, СРЕД

from (

select ЧЛВК\_ИД, AVG(ОЦЕНКА) СРЕД

from Н\_ВЕДОМОСТИ

where regexp\_like(ОЦЕНКА,'[0-9]')

group by ЧЛВК\_ИД) СРЕД\_ЗНАЧ,

Н\_ЛЮДИ

where ЧЛВК\_ИД=Н\_ЛЮДИ.ИД) ТАБЛ\_СРЕД

where ТАБЛ\_МАКС.ФАМИЛИЯ = ТАБЛ\_СРЕД.ФАМИЛИЯ

AND СРЕД<МАКС AND ROWNUM<=7;

**12. Сформировать SQL-запрос для получения таблицы вида:**

COLUMN " " FORMAT A20

COLUMN "Средняя оценка" FORMAT A20

COLUMN "Количество оценок" FORMAT A20

SELECT 'Оценки 4 и 5 во всем университете' as " ",

TO\_CHAR(ROUND(AVG(ОЦЕНКА),1)) as "Средняя оценка",

TO\_CHAR(COUNT(ОЦЕНКА)) as "Количество оценок"

FROM Н\_ВЕДОМОСТИ

WHERE ОЦЕНКА IN('4','5')

UNION

SELECT 'Оценки <<зачет>> в произвольном

учебном году во всем университете' as " ",

'-' as "Средняя оценка",

TO\_CHAR(COUNT(ОЦЕНКА)) as "Количество оценок"

FROM Н\_ВЕДОМОСТИ

WHERE ОЦЕНКА='зачет' AND TO\_CHAR(ДАТА,'YYYY')='2014'

UNION

SELECT 'Расстояние Левенштайна до вашей

фамилии от фамилий 10 персон,

имеющих оценки 3,4 и 5' as " ",

TO\_CHAR(AVG(utl\_match.edit\_distance('Байрамов', ФАМИЛИЯ))) as "Средняя оценка",

'-' as "Количество оценок"

FROM (SELECT ФАМИЛИЯ

FROM Н\_ЛЮДИ

WHERE ИД IN (

SELECT ЧЛВК\_ИД

FROM Н\_ВЕДОМОСТИ

WHERE ОЦЕНКА IN ('3','4','5') AND

ROWNUM <=10));

**13. Получить список персон, получивших оценки 3 и 4 с 01.09 прошлого календарного года по 20.07 текущего календарного года, упорядочить список по Ф.И.О. Использование объединений таблиц запрещено.**

SELECT distinct ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО, ИД

FROM Н\_ЛЮДИ

WHERE ИД IN(

SELECT ЧЛВК\_ИД

FROM Н\_ВЕДОМОСТИ

WHERE ОЦЕНКА IN ('3','4')

AND ADD\_MONTHS(TRUNC(TRUNC(SYSDATE,'YYYY')-1,'YYYY'),8)<=ДАТА

AND ADD\_MONTHS(TRUNC(SYSDATE,'YYYY'),6)+19>=ДАТА)

ORDER BY ФАМИЛИЯ;

**14. Получить список людей с встречающимися более 1 раза [upd 20.09.2016] сочетаниями фамилии, имени и отчество, сумма оценок которых не превышает сумму цифр ИД произвольной персоны из таблицы Н\_ЛЮДИ.**

SELECT ФАМИЛИЯ, СУММ

FROM (

SELECT ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО, SUM(ОЦЕНКА) as СУММ

FROM Н\_ВЕДОМОСТИ

JOIN Н\_ЛЮДИ ON Н\_ЛЮДИ.ИД=Н\_ВЕДОМОСТИ.ЧЛВК\_ИД

WHERE ОЦЕНКА IN('3','4','5')

GROUP BY ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО)

WHERE ФАМИЛИЯ IN (

SELECT ФАМИЛИЯ

FROM(

SELECT ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО, COUNT(\*) as КОЛ

FROM Н\_ЛЮДИ

GROUP BY ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО)

WHERE КОЛ=1)

AND СУММ<(

SELECT ИД

FROM Н\_ЛЮДИ

WHERE ИМЯ='Сергей'

AND ROWNUM<2);

Лабораторная работа по дисциплине

«Системы баз данных»

этап 2

Вариант 1

Исполнители: 2 студента (Байрамов С.И. и Разумовская А.В.)

Примечания: Для выполнения 2 этапа обоими исполнителями должен быть успешно защищен 1 этап лабораторной работы.

Вариант 1.

Создайте базу данных с таблицами:

Таблица\_1, где хранятся данные о людях.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ид | фамилия | имя | отчество | дата рождения | пол | место рождения | адрес | телефон |

Таблица\_2, где хранится информация об отпусках, командировках, больничных, выходных и праздниках

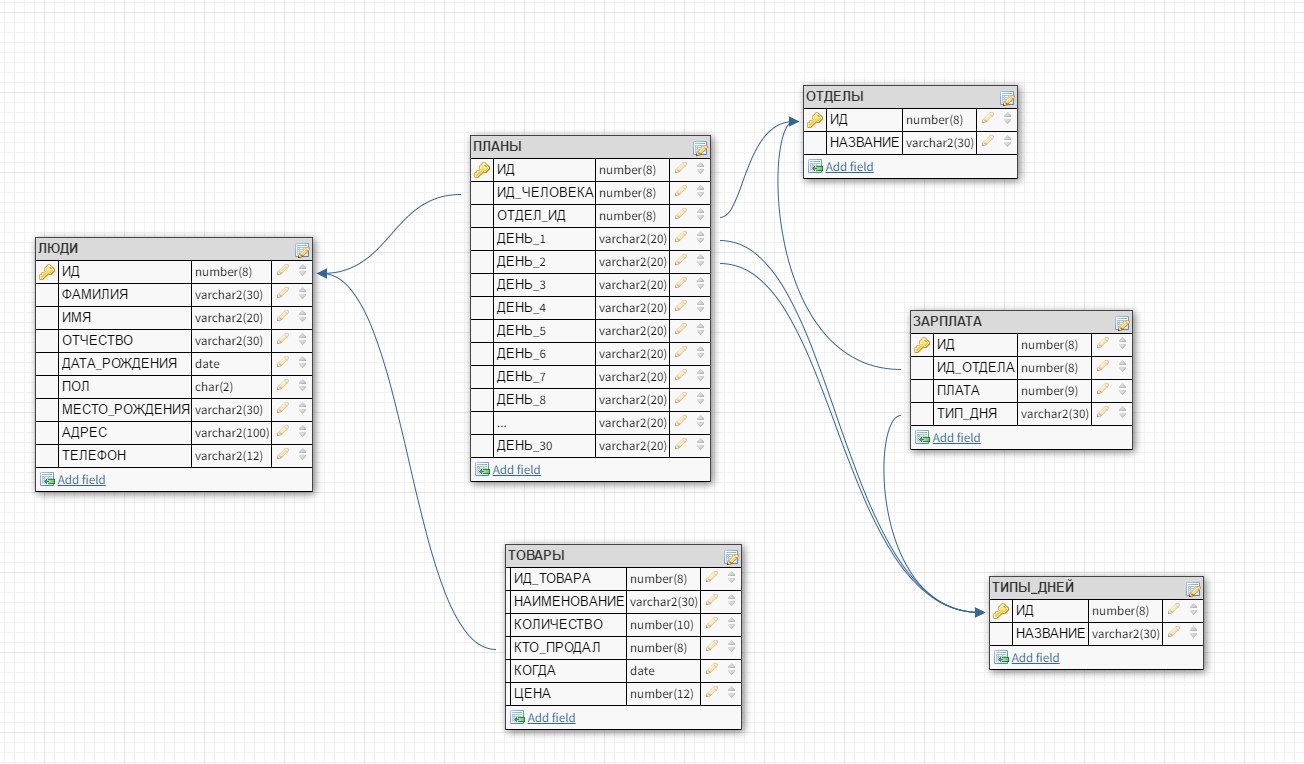
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ид | ид\_человека | отдел в котором работает человек (может работать в нескольких одновременно) | День\_1 | День\_2 | …. | День\_30 |

Таблица\_3,где хранится информация о товаре

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ид товара | наименование | количество | Кто продал (ид человека) | Когда (дата) | цена |

Дополнить другими таблицами, если необходимо. Написать пакет, который ежемесячно будет выдавать (подсчитывать) заработную плату с учётом отпусков, больничных, командировок, выходных и праздников. А также выводить количество товара, проданного всеми отделами, для определения лучшего отдела. Результат работы пакета показать на примерах. Написать промежуточные pipelined-функции, для возможной проверки на любом этапе обработки данных.

**Инфомодель:**



**SQL код:**

drop table ПЛАНЫ;

drop table ТОВАРЫ;

drop table ЗАРПЛАТА;

drop table ОТДЕЛЫ;

drop table ТИПЫ\_ДНЕЙ;

drop table ЛЮДИ;

CREATE TABLE ЛЮДИ (

ИД NUMBER(8) PRIMARY KEY CHECK(ИД > 0),

ФАМИЛИЯ VARCHAR2(30) NOT NULL,

ИМЯ VARCHAR2 (20) NOT NULL,

ОТЧЕСТВО VARCHAR2(30) NOT NULL,

ДАТА\_РОЖДЕНИЯ DATE NOT NULL,

ПОЛ CHAR(2) CHECK (ПОЛ IN ('М','Ж')) NOT NULL,

МЕСТО\_РОЖДЕНИЯ VARCHAR2(30) NOT NULL,

АДРЕС VARCHAR2(100) NOT NULL,

ТЕЛЕФОН VARCHAR2(12) CHECK(REGEXP\_LIKE(ТЕЛЕФОН, '^8[0-9]{10}|+7[0-9]{10}')) NOT NULL);

CREATE TABLE ОТДЕЛЫ(

ИД NUMBER(8) PRIMARY KEY CHECK(ИД>0),

НАИМЕНОВАНИЕ VARCHAR2(30) NOT NULL);

CREATE TABLE ТИПЫ\_ДНЕЙ(

ИД NUMBER(8) PRIMARY KEY CHECK(ИД>0),

НАЗВАНИЕ VARCHAR2(30) NOT NULL);

CREATE TABLE ТОВАРЫ(

ИД\_ТОВАРА NUMBER(8),

НАИМЕНОВАНИЕ VARCHAR2(30),

КОЛИЧЕСТВО NUMBER(10),

КТО\_ПРОДАЛ NUMBER(8), FOREIGN KEY (КТО\_ПРОДАЛ) REFERENCES ЛЮДИ(ИД),

КОГДА DATE,

ЦЕНА NUMBER(12));

CREATE TABLE ЗАРПЛАТА(

ИД NUMBER(8) PRIMARY KEY CHECK(ИД>0),

ИД\_ОТДЕЛА NUMBER(8), FOREIGN KEY (ИД\_ОТДЕЛА) REFERENCES ОТДЕЛЫ(ИД),

ПЛАТА NUMBER(9) CHECK(ПЛАТА >= 0),

ТИП\_ДНЯ NUMBER(8), FOREIGN KEY(ТИП\_ДНЯ) REFERENCES ТИПЫ\_ДНЕЙ(ИД));

CREATE TABLE ПЛАНЫ(

ИД NUMBER(8) PRIMARY KEY CHECK(ИД>0),

ИД\_ЧЕЛОВЕКА NUMBER(8), FOREIGN KEY (ИД\_ЧЕЛОВЕКА) REFERENCES ЛЮДИ(ИД),

ИД\_ОТДЕЛА NUMBER(8), FOREIGN KEY (ИД\_ОТДЕЛА) REFERENCES ОТДЕЛЫ(ИД),

ДЕНЬ\_1 NUMBER(8), FOREIGN KEY (ДЕНЬ\_1) REFERENCES ТИПЫ\_ДНЕЙ(ИД),

ДЕНЬ\_2 NUMBER(8), FOREIGN KEY (ДЕНЬ\_2) REFERENCES ТИПЫ\_ДНЕЙ(ИД),

ДЕНЬ\_3 NUMBER(8), FOREIGN KEY (ДЕНЬ\_3) REFERENCES ТИПЫ\_ДНЕЙ(ИД),

ДЕНЬ\_4 NUMBER(8), FOREIGN KEY (ДЕНЬ\_4) REFERENCES ТИПЫ\_ДНЕЙ(ИД),

ДЕНЬ\_5 NUMBER(8), FOREIGN KEY (ДЕНЬ\_5) REFERENCES ТИПЫ\_ДНЕЙ(ИД),

ДЕНЬ\_6 NUMBER(8), FOREIGN KEY (ДЕНЬ\_6) REFERENCES ТИПЫ\_ДНЕЙ(ИД),

ДЕНЬ\_7 NUMBER(8), FOREIGN KEY (ДЕНЬ\_7) REFERENCES ТИПЫ\_ДНЕЙ(ИД),

ДЕНЬ\_8 NUMBER(8), FOREIGN KEY (ДЕНЬ\_8) REFERENCES ТИПЫ\_ДНЕЙ(ИД),

ДЕНЬ\_9 NUMBER(8), FOREIGN KEY (ДЕНЬ\_9) REFERENCES ТИПЫ\_ДНЕЙ(ИД),

ДЕНЬ\_10 NUMBER(8), FOREIGN KEY (ДЕНЬ\_10) REFERENCES ТИПЫ\_ДНЕЙ(ИД),

ДЕНЬ\_11 NUMBER(8), FOREIGN KEY (ДЕНЬ\_11) REFERENCES ТИПЫ\_ДНЕЙ(ИД),

ДЕНЬ\_12 NUMBER(8), FOREIGN KEY (ДЕНЬ\_12) REFERENCES ТИПЫ\_ДНЕЙ(ИД),

ДЕНЬ\_13 NUMBER(8), FOREIGN KEY (ДЕНЬ\_13) REFERENCES ТИПЫ\_ДНЕЙ(ИД),

ДЕНЬ\_14 NUMBER(8), FOREIGN KEY (ДЕНЬ\_14) REFERENCES ТИПЫ\_ДНЕЙ(ИД),

ДЕНЬ\_15 NUMBER(8), FOREIGN KEY (ДЕНЬ\_15) REFERENCES ТИПЫ\_ДНЕЙ(ИД),

ДЕНЬ\_16 NUMBER(8), FOREIGN KEY (ДЕНЬ\_16) REFERENCES ТИПЫ\_ДНЕЙ(ИД),

ДЕНЬ\_17 NUMBER(8), FOREIGN KEY (ДЕНЬ\_17) REFERENCES ТИПЫ\_ДНЕЙ(ИД),

ДЕНЬ\_18 NUMBER(8), FOREIGN KEY (ДЕНЬ\_18) REFERENCES ТИПЫ\_ДНЕЙ(ИД),

ДЕНЬ\_19 NUMBER(8), FOREIGN KEY (ДЕНЬ\_19) REFERENCES ТИПЫ\_ДНЕЙ(ИД),

ДЕНЬ\_20 NUMBER(8), FOREIGN KEY (ДЕНЬ\_20) REFERENCES ТИПЫ\_ДНЕЙ(ИД),

ДЕНЬ\_21 NUMBER(8), FOREIGN KEY (ДЕНЬ\_21) REFERENCES ТИПЫ\_ДНЕЙ(ИД),

ДЕНЬ\_22 NUMBER(8), FOREIGN KEY (ДЕНЬ\_22) REFERENCES ТИПЫ\_ДНЕЙ(ИД),

ДЕНЬ\_23 NUMBER(8), FOREIGN KEY (ДЕНЬ\_23) REFERENCES ТИПЫ\_ДНЕЙ(ИД),

ДЕНЬ\_24 NUMBER(8), FOREIGN KEY (ДЕНЬ\_24) REFERENCES ТИПЫ\_ДНЕЙ(ИД),

ДЕНЬ\_25 NUMBER(8), FOREIGN KEY (ДЕНЬ\_25) REFERENCES ТИПЫ\_ДНЕЙ(ИД),

ДЕНЬ\_26 NUMBER(8), FOREIGN KEY (ДЕНЬ\_26) REFERENCES ТИПЫ\_ДНЕЙ(ИД),

ДЕНЬ\_27 NUMBER(8), FOREIGN KEY (ДЕНЬ\_27) REFERENCES ТИПЫ\_ДНЕЙ(ИД),

ДЕНЬ\_28 NUMBER(8), FOREIGN KEY (ДЕНЬ\_28) REFERENCES ТИПЫ\_ДНЕЙ(ИД),

ДЕНЬ\_29 NUMBER(8), FOREIGN KEY (ДЕНЬ\_29) REFERENCES ТИПЫ\_ДНЕЙ(ИД),

ДЕНЬ\_30 NUMBER(8), FOREIGN KEY (ДЕНЬ\_30) REFERENCES ТИПЫ\_ДНЕЙ(ИД));

INSERT INTO ЛЮДИ VALUES(1, 'Байрамов', 'Сергей', 'Игоревич', TO\_DATE('23.08.1996', 'DD.MM.YYYY'), 'М', '-', '-', '89323252522');

INSERT INTO ЛЮДИ VALUES(2, 'Разумовская', 'Анастасия', 'Владимировна', TO\_DATE('23.08.1996', 'DD.MM.YYYY'), 'Ж', '-', '-', '+78884445551');

INSERT INTO ЛЮДИ VALUES(3, 'Гулямова', 'Ситора', 'Икромовна', TO\_DATE('20.07.1996', 'DD.MM.YYYY'), 'Ж', '-', '-', '+78884645155');

INSERT INTO ЛЮДИ VALUES(4, 'Байрамова', 'Наталья', 'Николаевна', TO\_DATE('01.01.1996', 'DD.MM.YYYY'), 'Ж', '-', '-', '89327778800');

INSERT INTO ЛЮДИ VALUES(5, 'Яркеев', 'Александр', 'Сергеевич', TO\_DATE('14.11.1996', 'DD.MM.YYYY'), 'М', '-', '-', '+78883200522');

INSERT INTO ЛЮДИ VALUES(6, 'Сердюк', 'Евгений', 'Юрьевич', TO\_DATE('17.11.1994', 'DD.MM.YYYY'), 'М', '-', '-', '88883252522');

INSERT INTO ОТДЕЛЫ VALUES(1, 'Верхний');

INSERT INTO ОТДЕЛЫ VALUES(2, 'Средний');

INSERT INTO ОТДЕЛЫ VALUES(3, 'Нижний');

INSERT INTO ТИПЫ\_ДНЕЙ VALUES(1, 'Рабочий');

INSERT INTO ТИПЫ\_ДНЕЙ VALUES(2, 'Отпуск');

INSERT INTO ТИПЫ\_ДНЕЙ VALUES(3, 'Больничный');

INSERT INTO ТИПЫ\_ДНЕЙ VALUES(4, 'Командировка');

INSERT INTO ТИПЫ\_ДНЕЙ VALUES(5, 'Выходной');

INSERT INTO ТИПЫ\_ДНЕЙ VALUES(6, 'Праздник');

INSERT INTO ЗАРПЛАТА VALUES(1, 1, 100, 1);

INSERT INTO ЗАРПЛАТА VALUES(2, 1, 100, 2);

INSERT INTO ЗАРПЛАТА VALUES(3, 1, 100, 3);

INSERT INTO ЗАРПЛАТА VALUES(4, 1, 100, 4);

INSERT INTO ЗАРПЛАТА VALUES(5, 1, 50, 5);

INSERT INTO ЗАРПЛАТА VALUES(6, 1, 50, 6);

INSERT INTO ЗАРПЛАТА VALUES(7, 2, 70, 1);

INSERT INTO ЗАРПЛАТА VALUES(8, 2, 70, 2);

INSERT INTO ЗАРПЛАТА VALUES(9, 2, 30, 3);

INSERT INTO ЗАРПЛАТА VALUES(10, 2, 70, 4);

INSERT INTO ЗАРПЛАТА VALUES(11, 2, 20, 5);

INSERT INTO ЗАРПЛАТА VALUES(12, 2, 0, 6);

INSERT INTO ЗАРПЛАТА VALUES(13, 3, 50, 1);

INSERT INTO ЗАРПЛАТА VALUES(14, 3, 50, 2);

INSERT INTO ЗАРПЛАТА VALUES(15, 3, 20, 3);

INSERT INTO ЗАРПЛАТА VALUES(16, 3, 50, 4);

INSERT INTO ЗАРПЛАТА VALUES(17, 3, 0, 5);

INSERT INTO ЗАРПЛАТА VALUES(18, 3, 0, 6);

INSERT INTO ПЛАНЫ VALUES(1, 1, 1,

1,1,1,1,1,1,5,1,1,1,1,1,6,5,1,1,1,1,2,2,2,2,2,2,1,1,1,5,1,1);

INSERT INTO ПЛАНЫ VALUES(2, 2, 1,

1,1,3,3,3,1,5,1,1,4,4,4,4,4,1,1,1,1,1,4,4,4,4,4,1,1,1,5,1,1);

INSERT INTO ПЛАНЫ VALUES(3, 3, 2,

3,3,3,3,3,5,5,1,1,1,1,1,6,5,2,2,2,2,2,2,5,1,1,1,1,1,5,5,1,1);

INSERT INTO ПЛАНЫ VALUES(4, 3, 3,

3,3,3,3,3,1,5,1,1,1,1,1,6,5,4,4,4,4,4,4,5,1,1,1,1,1,1,5,1,1);

INSERT INTO ПЛАНЫ VALUES(5, 4, 1,

1,1,1,1,1,1,5,1,1,1,1,1,6,5,1,1,1,1,1,1,5,1,1,1,1,1,1,5,1,1);

INSERT INTO ПЛАНЫ VALUES(6, 4, 3,

1,1,1,1,1,1,5,1,1,1,1,1,6,5,1,1,1,1,1,1,5,1,1,1,1,1,1,5,1,1);

INSERT INTO ПЛАНЫ VALUES(7, 5, 1,

1,1,1,1,1,1,5,1,1,1,1,1,6,3,3,3,3,3,3,3,3,2,2,2,2,2,2,2,2,2);

INSERT INTO ПЛАНЫ VALUES(8, 6, 3,

1,1,1,1,1,1,5,1,1,1,1,1,6,5,1,1,1,1,1,1,5,1,1,1,1,1,1,5,1,1);

INSERT INTO ПЛАНЫ VALUES(9, 6, 2,

4,4,4,4,4,4,4,4,4,2,2,2,2,2,2,2,1,1,1,5,5,1,1,1,1,1,5,5,1,1);

INSERT INTO ТОВАРЫ VALUES(1, 'Молоко', 200, 1, TO\_DATE('01.10.2016','DD.MM.YYYY'), 10000);

INSERT INTO ТОВАРЫ VALUES(2, 'Кефир', 100, 1, TO\_DATE('01.10.2016','DD.MM.YYYY'), 4000);

INSERT INTO ТОВАРЫ VALUES(1, 'Молоко', 500, 4, TO\_DATE('03.10.2016','DD.MM.YYYY'), 25000);

INSERT INTO ТОВАРЫ VALUES(3, 'Сметана', 100, 3, TO\_DATE('03.10.2016','DD.MM.YYYY'), 7000);

INSERT INTO ТОВАРЫ VALUES(4, 'Творог', 100, 1, TO\_DATE('03.10.2016','DD.MM.YYYY'), 10000);

INSERT INTO ТОВАРЫ VALUES(4, 'Творог', 300, 2, TO\_DATE('04.10.2016','DD.MM.YYYY'), 30000);

INSERT INTO ТОВАРЫ VALUES(5, 'Бифидок', 250, 5, TO\_DATE('04.10.2016','DD.MM.YYYY'), 15000);

INSERT INTO ТОВАРЫ VALUES(6, 'Кумыс', 50, 1, TO\_DATE('04.10.2016','DD.MM.YYYY'), 3000);

INSERT INTO ТОВАРЫ VALUES(7, 'Йогурт', 400, 2, TO\_DATE('04.10.2016','DD.MM.YYYY'), 20000);

INSERT INTO ТОВАРЫ VALUES(1, 'Молоко', 1000, 6, TO\_DATE('04.10.2016','DD.MM.YYYY'), 50000);

INSERT INTO ТОВАРЫ VALUES(2, 'Кефир', 400, 4, TO\_DATE('05.10.2016','DD.MM.YYYY'), 16000);

INSERT INTO ТОВАРЫ VALUES(4, 'Творог', 150, 6, TO\_DATE('05.10.2016','DD.MM.YYYY'), 15000);

INSERT INTO ТОВАРЫ VALUES(5, 'Бифидок', 200, 5, TO\_DATE('06.10.2016','DD.MM.YYYY'), 12000);

INSERT INTO ТОВАРЫ VALUES(6, 'Кумыс', 300, 1, TO\_DATE('07.10.2016','DD.MM.YYYY'), 18000);

DROP PACKAGE staff\_pkg;

DROP TYPE personal\_salary\_tab;

DROP TYPE personal\_salary\_row;

DROP TYPE salary\_tab;

drop TYPE salary\_row;

drop type sales\_count\_tab;

drop type sales\_count\_row;

CREATE TYPE salary\_row AS OBJECT (

salary NUMBER,

department NUMBER(9),

name VARCHAR2(50),

surname VARCHAR2(50),

id NUMBER(9)

);

/

CREATE TYPE salary\_tab IS TABLE OF salary\_row;

/

CREATE TYPE sales\_count\_row AS OBJECT (

id NUMBER(9),

name VARCHAR2(50),

sales\_count NUMBER(9)

);

/

CREATE TYPE sales\_count\_tab IS TABLE OF sales\_count\_row;

/

CREATE TYPE personal\_salary\_row AS OBJECT (

salary NUMBER,

name VARCHAR2(50),

surname VARCHAR2(50),

id NUMBER(9)

);

/

CREATE TYPE personal\_salary\_tab IS TABLE OF personal\_salary\_row;

/

CREATE OR REPLACE PACKAGE staff\_pkg IS

FUNCTION get\_tab\_salary RETURN salary\_tab PIPELINED;

FUNCTION get\_sales\_count RETURN sales\_count\_tab;

FUNCTION get\_personal\_salary (all\_salary IN salary\_tab)

RETURN personal\_salary\_tab;

END staff\_pkg;

/

CREATE OR REPLACE PACKAGE BODY staff\_pkg IS

FUNCTION get\_tab\_salary RETURN salary\_tab PIPELINED

IS

salary number := 0;

curr\_salary number;

BEGIN

FOR j IN (SELECT ЛЮДИ.ИД as id, ИМЯ, ФАМИЛИЯ, ЗАРПЛАТА.ИД\_ОТДЕЛА FROM ЗАРПЛАТА, ЛЮДИ, ПЛАНЫ

WHERE ЗАРПЛАТА.ИД\_ОТДЕЛА = ПЛАНЫ.ИД\_ОТДЕЛА AND ПЛАНЫ.ИД\_ЧЕЛОВЕКА = ЛЮДИ.ИД

GROUP BY ЛЮДИ.ИД, ИМЯ, ФАМИЛИЯ, ЗАРПЛАТА.ИД\_ОТДЕЛА)

LOOP

salary:= 0;

FOR i IN 1 .. 30 LOOP

SELECT ЗАРПЛАТА.ПЛАТА INTO curr\_salary

FROM ЗАРПЛАТА

WHERE j.ИД\_ОТДЕЛА = ЗАРПЛАТА.ИД\_ОТДЕЛА AND

ТИП\_ДНЯ LIKE (SELECT case

when i = 1 then День\_1

when i = 2 then День\_2

when i = 3 then День\_3

when i = 4 then День\_4

when i = 5 then День\_5

when i = 6 then День\_6

when i = 7 then День\_7

when i = 8 then День\_8

when i = 9 then День\_9

when i = 10 then День\_10

when i = 11 then День\_11

when i = 12 then День\_12

when i = 13 then День\_13

when i = 14 then День\_14

when i = 15 then День\_15

when i = 16 then День\_16

when i = 17 then День\_17

when i = 18 then День\_18

when i = 19 then День\_19

when i = 20 then День\_20

when i = 21 then День\_21

when i = 22 then День\_22

when i = 23 then День\_23

when i = 24 then День\_24

when i = 25 then День\_25

when i = 26 then День\_26

when i = 27 then День\_27

when i = 28 then День\_28

when i = 29 then День\_29

when i = 30 then День\_30

END FROM ПЛАНЫ WHERE ПЛАНЫ.ИД\_ЧЕЛОВЕКА = j.id AND j.ИД\_ОТДЕЛА = ПЛАНЫ.ИД\_ОТДЕЛА);

salary := salary + curr\_salary;

END LOOP;

PIPE ROW(salary\_row(salary, j.ИД\_ОТДЕЛА, j.ИМЯ, j.ФАМИЛИЯ, j.id));

END LOOP;

RETURN;

END ;

FUNCTION get\_personal\_salary (all\_salary IN salary\_tab)

RETURN personal\_salary\_tab IS

l\_tab personal\_salary\_tab := personal\_salary\_tab();

BEGIN

FOR i IN (SELECT id, sum(salary) as sum\_salary, name, surname FROM table(all\_salary) GROUP BY id, name, surname)

LOOP

l\_tab.extend;

l\_tab(l\_tab.last) := personal\_salary\_row(i.sum\_salary, i.name, i.surname, i.id);

END LOOP;

RETURN l\_tab;

END ;

FUNCTION get\_sales\_count RETURN sales\_count\_tab IS

l\_tab sales\_count\_tab := sales\_count\_tab();

BEGIN

FOR i IN (select sum(цена) as sales\_count, отделы.ид as id, отделы.наименование as name

FROM отделы, товары, планы

WHERE товары.кто\_продал = планы.ид\_человека

AND планы.ид\_отдела = отделы.ид

GROUP BY отделы.ид, отделы.наименование ORDER BY sum(цена) DESC)

LOOP

l\_tab.extend;

l\_tab(l\_tab.last) := sales\_count\_row(i.id, i.name, i.sales\_count);

END LOOP;

RETURN l\_tab;

END ;

END staff\_pkg;

/

SELECT \* FROM table(staff\_pkg.get\_personal\_salary(staff\_pkg.get\_tab\_salary));

SELECT \* FROM table(staff\_pkg.get\_sales\_count);