

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

Кафедра информатики и прикладной математики

## **Системы баз данных**

Лабораторная работа 1

Этап 1

“Написание SQL-запросов к учебной базе данных”



**Старались:** Шкаруба Н.Е.  
Суханова В.А.

**Проверил:** Беликов П.А.  
Группа: Р3318  
2016 г

## Этап 1. SQL- запросы:

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

```
select
  ИД,
  СЭС_ИД,
  ЧЛВК_ИД,
  АУДИТОРИЯ_К,
  УЧГОД,
  ГРУППА,
  СЕМЕСТР,
  КТО_СОЗДАЛ,
  АУДИТОРИЯ,
  КТО_ИЗМЕНИЛ,
  to_char(ДАТА, 'YYYY.DD.MM HH24:MI:SS') as ДАТА,
  to_char(ВРЕМЯ, 'YYYY.DD.MM HH24:MI:SS') as ВРЕМЯ,
  to_char(ДАТА_К, 'YYYY.DD.MM HH24:MI:SS') as ДАТА_К,
  to_char(ВРЕМЯ_К, 'YYYY.DD.MM HH24:MI:SS') as ВРЕМЯ_К,
  to_char(КОГДА_СОЗДАЛ, 'YYYY.DD.MM HH24:MI:SS') as КОГДА_СОЗДАЛ,
  to_char(КОГДА_ИЗМЕНИЛ, 'YYYY.DD.MM HH24:MI:SS') 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(to_date('2016/09/01', 'yyyy/mm/dd') - ДАТА_РОЖДЕНИЯ) from (select
ДАТА_РОЖДЕНИЯ from Н_ЛЮДИ order by dbms_random.random) where rownum=1;
```

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(отчество, 1, 1) || '.'
from
  Н_люди
where
  extract(month from дата_рождения) = extract(month from (select дата_рождения from
  (select * from Н_люди order by dbms_random.random) where rownum = 1))
  and отчество is not null;
```

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, 0, 2) = (select substr(ФАМИЛИЯ, 0, 2) from (select ФАМИЛИЯ from
  Н_ЛЮДИ order by dbms_random.random) where rownum=1) AND rownum <= 75
order by
  ФАМИЛИЯ desc, ИМЯ desc, ОТЧЕСТВО 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.

```
--- Not in
select
  фамилия, имя, отчество, ид
from
  Н_люди
where
  substr(фамилия, 0, 1) not in ('А', 'Б', 'З', 'К', 'У')
  and substr(имя, 0, 1) not in ('А', 'Б', 'З', 'К', 'У')
  and substr(отчество, 0, 1) not in ('А', 'Б', 'З', 'К', 'У');

--- Regexp
select
  фамилия, имя, отчество, ид
from
  Н_люди
where
  not REGEXP_LIKE(имя, '^[АБЗКУ]') and
  not REGEXP_LIKE(фамилия, '^[АБЗКУ]') and
  not REGEXP_LIKE(отчество, '^[АБЗКУ]');
```

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

```
select
  count (distinct ИД)
from
  Н_ЛЮДИ
where
  ИМЯ = (select ИМЯ from (select * from Н_ЛЮДИ order by dbms_random.random) where
rownum=1);
```

8. Вывести, используя таблицу Н\_ВЕДОМОСТИ, удвоенные (значение\_оценки\* 2 ) оценки произвольной персоны из Н\_ЛЮДИ\*\*.

```
select
  (оценка * 2)
from
  н_ведомости
where
  члвк_ид = (select ид from (select * from н_люди order by dbms_random.random)
where rownum = 1)
and regexp_like(оценка, 'd');
```

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)
from
  Н_ВЕДОМОСТИ
where
  ЧЛВК_ИД in (select ИД from (select ИД from Н_ЛЮДИ order by dbms_random.random)
where rownum <= 7) AND REGEXP_LIKE(ОЦЕНКА, 'd');
```

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

```
select * from н_люди cross join н_ведомости;
```

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.

```
--- Красный - собирает все средние оценки
--- Зелёный - выбирает лучших из них
--- Фиолетовый - выбирает случайные результаты
--- Чёрный - Добавляет столбцы Фамилия, Имя и Отчество

select
  people.Фамилия, people.Имя, people.Отчество, result.avg_marks
from(
  select
    *
  from(
    select
      Фамилия, max(avg_marks) avg_marks
    from(
      select
        p.п.ФАМИЛИЯ Фамилия, p.ИМЯ Имя, p.ОТЧЕСТВО Отчество, avg(v.ОЦЕНКА)
                                                avg_marks
      from
        Н_ЛЮДИ p
      inner join
        Н_ВЕДОМОСТИ v
      on
        p.ИД = v.ЧЛВК_ИД
      where
        REGEXP_LIKE(v.ОЦЕНКА, 'd')
      group by
        p.ФАМИЛИЯ, p.ИМЯ, p.ОТЧЕСТВО)
    group by Фамилия)
  order by dbms_random.random) result
join
  Н_ЛЮДИ people
on
  result.Фамилия = people.Фамилия
where rownum <= 7;
```

12. Сформировать SQL-запрос для получения таблицы вида:

|  | Средняя оценка | Количество оценок |
|--|----------------|-------------------|
| Оценки 4 и 5 во всём университете  | 4.3            | 98                |
| Оценки "зачёт" в произвольном учебном году во всём университете                      | -              | 86                |
| Расстояние Левенштейна до вашей фамилии от фамилии 10 персон, имеющих оценки 3, 4, 5 | 2              | -                 |

```

select
  'ОЦЕНКИ 4 и 5 во всём университете' as " ",
  avg(ОЦЕНКА) as "Средняя оценка",
  count(ОЦЕНКА) as "Количество оценок"
from
  Н_ВЕДОМОСТИ
where
  REGEXP_LIKE(ОЦЕНКА, '[45]')
union
select
  'Оценки <Зачёт> в произвольном учебном году во всём университете' as " ",
  NULL as "Средняя оценка",
  count(ОЦЕНКА) as "Количество оценок"
from
  Н_ВЕДОМОСТИ
where
  REGEXP_LIKE(ОЦЕНКА, 'зачет')
union
select
  'Расстояние Левенштейна' as " ",
  avg(utl_match.edit_distance('Шкаруба', Фамилия)) as "Средняя оценка",
  NULL as "Количество оценок"
from(
  select
    фамилия
  from
    Н_люди
  where
    ид in(
      SELECT ЧЛВК_ИД FROM Н_ВЕДОМОСТИ WHERE ОЦЕНКА='3'
      UNION
      SELECT ЧЛВК_ИД FROM Н_ВЕДОМОСТИ WHERE ОЦЕНКА='4'
      UNION
      SELECT ЧЛВК_ИД FROM Н_ВЕДОМОСТИ WHERE ОЦЕНКА='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
    Фамилия, Имя, Отчество
from
    Н_ЛЮДИ
where
    ИД in (select ЧЛВК_ИД from Н_ВЕДОМОСТИ
    where
        ОЦЕНКА IN('3','4') AND DATA BETWEEN TO_DATE('2015.09.01', 'yyyy.mm.dd') AND
        TO_DATE('2016.07.20', 'yyyy.mm.dd'))
order by
    Фамилия, Имя, Отчество;
```

14. Получить список людей с наиболее частыми сочетаниями фамилии, имени и отчества, сумма оценок которых не превышает сумму цифр ИД произвольной персоны из таблицы Н\_ЛЮДИ.

```
-- Сдавалось по другому заданию.
select
    p.Фамилия, p.Имя, p.Отчество
from
    Н_ЛЮДИ p
where
    (substr(p.Фамилия, 0, 4) = substr(p.Имя, 0, 4)
    or substr(p.Фамилия, 0, 4) = substr(p.Отчество, 0, 4)
    or substr(p.Имя, 0, 4) = substr(p.Отчество, 0, 4))
and (select sum(ОЦЕНКА) from Н_ВЕДОМОСТИ where ЧЛВК_ИД = p.ИД AND
    REGEXP_LIKE(ОЦЕНКА, '\d')) < (select sum(val) from (select substr(col,level,1) val
    from (select ИД as col from Н_ЛЮДИ where ИД = ((select ИД from (select ИД from
    Н_ЛЮДИ order by dbms_random.random) where rownum = 1))) connect by level <=
    length(col)));
```