

# Этап 1. SQL запросы

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

ALTER SESSION SET NLS\_DATE\_FORMAT = 'dd-mm-yyyy hh24:mi:ss';

SELECT Н\_СЕССИЯ.\* FROM Н\_СЕССИЯ;

1. Вывести неповторяющиеся названия дисциплин из таблицы

Н\_ДИСЦИПЛИНЫ

SELECT DISTINCT НАИМЕНОВАНИЕ FROM Н\_ДИСЦИПЛИНЫ;

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

SELECT ROUND(TO\_DATE('2017/09/01', 'yyyy/mm/dd')

- ДАТА\_РОЖДЕНИЯ) AS РАЗНИЦА\_В\_ДНЯХ FROM

(SELECT ДАТА\_РОЖДЕНИЯ FROM Н\_ЛЮДИ ORDER BY dbms\_random.value)

WHERE rownum = 1;

1. 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) || '.' AS ИЗБРАННЫЙ

FROM Н\_ЛЮДИ WHERE TO\_CHAR(ДАТА\_РОЖДЕНИЯ,'mm') =

(SELECT TO\_CHAR(ДАТА\_РОЖДЕНИЯ,'mm') FROM

( SELECT ДАТА\_РОЖДЕНИЯ FROM Н\_ЛЮДИ ORDER BY dbms\_random.value )

WHERE rownum = 1);

1. 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 ФАМИЛИЯ LIKE(

SELECT SUBSTR((ФАМИЛИЯ),1,2) || '%' FROM

( SELECT ФАМИЛИЯ FROM Н\_ЛЮДИ ORDER BY dbms\_random.value )

WHERE rownum = 1)

AND ROWNUM <= 75

ORDER BY ФАМИЛИЯ DESC,ИМЯ DESC, ОТЧЕСТВО DESC;

1. 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) NOT IN ('А','Б','З')

AND SUBSTR(ОТЧЕСТВО,1,1) NOT IN ('К','У');

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 ИМЯ NOT LIKE 'А%'

AND ИМЯ NOT LIKE 'Б%'

AND ИМЯ NOT LIKE 'З%'

AND ОТЧЕСТВО NOT LIKE 'К%'

AND ОТЧЕСТВО NOT LIKE 'У%';

SELECT ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО, ИД

FROM Н\_ЛЮДИ

WHERE REGEXP\_LIKE (ИМЯ, '^[^АБЗ].\*$')

AND REGEXP\_LIKE (ОТЧЕСТВО, '^[^УК].\*$');

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

SELECT COUNT(DISTINCT ФАМИЛИЯ || ИМЯ || ОТЧЕСТВО || ДАТА\_РОЖДЕНИЯ) AS КОЛИЧЕСТВО

FROM Н\_ЛЮДИ

WHERE ИМЯ = (

SELECT ИМЯ FROM ( SELECT ИМЯ FROM Н\_ЛЮДИ ORDER BY dbms\_random.value )

WHERE rownum = 1);

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

SELECT ОЦЕНКА\*2

FROM Н\_ВЕДОМОСТИ

WHERE ЧЛВК\_ИД = (SELECT ИД FROM (

SELECT ИД FROM Н\_ЛЮДИ ORDER BY dbms\_random.value )

WHERE rownum = 1)

AND REGEXP\_LIKE(ОЦЕНКА, '^[0-9]$')

AND оценка NOT IN ('99');

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

SELECT Н\_ЛЮДИ.ФАМИЛИЯ, SUM(оценка) AS СУММА\_ОЦЕНОК FROM Н\_ВЕДОМОСТИ

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

WHERE ЧЛВК\_ИД IN (SELECT ид FROM (

SELECT ИД FROM Н\_ЛЮДИ

ORDER BY dbms\_random.value)

WHERE rownum <=7)

AND REGEXP\_LIKE(ОЦЕНКА,'^[0-9]+$')

AND ОЦЕНКА NOT IN ('99')

GROUP BY Н\_ЛЮДИ.ФАМИЛИЯ;

1. Получить декартово произведение N таблиц, где N равно 3й цифре вашего табельного номера.

SELECT Н\_ЛЮДИ.\*, Н\_ВЕДОМОСТИ.\*, Н\_ОЦЕНКИ.\*, Н\_ИЗМ\_ЛЮДИ.\*, Н\_СЕССИЯ.\*, Н\_ОБУЧЕНИЯ.\*, Н\_ДИСЦИПЛИНЫ.\*

FROM Н\_ЛЮДИ, Н\_ВЕДОМОСТИ, Н\_ОЦЕНКИ, Н\_ИЗМ\_ЛЮДИ, Н\_СЕССИЯ, Н\_ОБУЧЕНИЯ, Н\_ДИСЦИПЛИНЫ;

1. Вывести, используя таблицу Н\_ВЕДОМОСТИ, среднюю оценку 7и произвольных персон из Н\_ЛЮДИ, их фамилии, имена и отчества.

WITH СР\_ОЦ7 AS (SELECT ФАМИЛИЯ AS ФАМИЛИЯ1, ИМЯ, ОТЧЕСТВО avg(ОЦЕНКА) AS СРЕДНЯЯ\_ОЦЕНКА1

FROM Н\_ВЕДОМОСТИ

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

WHERE ЧЛВК\_ИД IN (SELECT ИД FROM (

SELECT ИД FROM Н\_ЛЮДИ

ORDER BY DBMS\_RANDOM.RANDOM)

WHERE rownum <=7)

AND REGEXP\_LIKE(ОЦЕНКА,'^[0-9]+$')

AND ОЦЕНКА NOT IN ('99')

GROUP BY Н\_ЛЮДИ.ФАМИЛИЯ),

СР\_ОЦ\_ВСЕ AS (SELECT ФАМИЛИЯ AS ФАМИЛИЯ2, MAX(срроценка) AS СРЕДНЯЯ\_ОЦЕНКА2 FROM (

SELECT ФАМИЛИЯ, AVG(ОЦЕНКА) срроценка

FROM Н\_ВЕДОМОСТИ

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

WHERE REGEXP\_LIKE(ОЦЕНКА,'^[0-9]+$')

AND ОЦЕНКА NOT IN ('99')

GROUP BY Н\_ЛЮДИ.ФАМИЛИЯ)

GROUP BY ФАМИЛИЯ)

SELECT ФАМИЛИЯ1 ФАМИЛИЯ, СРЕДНЯЯ\_ОЦЕНКА1 СРЕДНЯЯ\_ОЦЕНКА FROM СР\_ОЦ7

JOIN СР\_ОЦ\_ВСЕ ON СР\_ОЦ\_ВСЕ.ФАМИЛИЯ2=СР\_ОЦ7.ФАМИЛИЯ1

WHERE СР\_ОЦ7.СРЕДНЯЯ\_ОЦЕНКА1<=СР\_ОЦ\_ВСЕ.СРЕДНЯЯ\_ОЦЕНКА2;

1. Сформировать SQL-запрос для получения таблицы вида:

|  |  |  |
| --- | --- | --- |
|  | Средняя оценка | Количество оценок |
| Оценки 4 и 5 во всем университете | 4.3 | 98 |
| Оценки «зачет» в произвольном учебном году во всем университете | - | 86 |
| Расстояние Левенштайна до вашей фамилии от фамилии  10 персон, имеющих оценки  3, 4 и 5 | 2 | - |

SELECT 'Оценки 4 и 5',

ROUND(AVG(ОЦЕНКА),2) Средняя\_оценка,

COUNT(ОЦЕНКА) Количество\_оценок

FROM Н\_ВЕДОМОСТИ

WHERE ОЦЕНКА IN('4','5')

UNION

SELECT 'Оценки ''зачет''',to\_number(NULL) Средняя\_оценка,

COUNT(ОЦЕНКА) Количество\_оценок

FROM Н\_ВЕДОМОСТИ

WHERE ДАТА BETWEEN TO\_DATE('2010/09/01', 'yyyy/mm/dd')

AND TO\_DATE('2011/07/20', 'yyyy/mm/dd')

AND ОЦЕНКА IN ('зачет')

UNION

SELECT 'Расстояние', avg(Средняя\_оценка),to\_number(NULL) Количество\_оценок FROM (

SELECT utl\_match.edit\_distance('Милешин', ФАМИЛИЯ) Средняя\_оценка

FROM (SELECT ФАМИЛИЯ

FROM Н\_ЛЮДИ

WHERE ИД IN (

SELECT DISTINCT A.ЧЛВК\_ИД

FROM Н\_ВЕДОМОСТИ A

INNER JOIN Н\_ВЕДОМОСТИ B ON A.ЧЛВК\_ИД = B.ЧЛВК\_ИД

INNER JOIN Н\_ВЕДОМОСТИ C ON A.ЧЛВК\_ИД = C.ЧЛВК\_ИД

WHERE A.ОЦЕНКА = '3'

AND B.ОЦЕНКА = '4'

AND C.ОЦЕНКА = '5'

)

AND ROWNUM <=10)

);

1. Получить список персон, получивших оценки 3 и 4 с 01.09 прошлого календарного года по 20.07 текущего календарного года, упорядочить список по ФИО. Использование объединений таблиц запрещено.

SELECT ИД, ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО

FROM Н\_ЛЮДИ

WHERE ИД IN ( SELECT A.ЧЛВК\_ИД FROM Н\_ВЕДОМОСТИ A, Н\_ВЕДОМОСТИ B

WHERE A.ЧЛВК\_ИД = B.ЧЛВК\_ИД

AND A.ОЦЕНКА = '3'

AND B.ОЦЕНКА = '4'

AND A.ДАТА BETWEEN

to\_date('2016/09/01', 'yyyy/mm/dd')

AND to\_date('2017/07/20', 'yyyy/mm/dd')

)

ORDER BY ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО;

1. 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.

SELECT СУММА\_ОЦЕНОК, Н\_ИД, ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО

FROM (

SELECT DISTINCT Н\_ВЕДОМОСТИ.ЧЛВК\_ИД Н\_ИД, SUM(ОЦЕНКА) OVER

(partition BY ЧЛВК\_ИД) СУММА\_ОЦЕНОК

FROM Н\_ВЕДОМОСТИ

WHERE ЧЛВК\_ИД IN(

SELECT ИД FROM Н\_ЛЮДИ WHERE ФАМИЛИЯ || ИМЯ || ОТЧЕСТВО IN (

SELECT ФАМИЛИЯ || ИМЯ || ОТЧЕСТВО

FROM(

SELECT ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО, COUNT(\*) a

FROM Н\_ЛЮДИ

GROUP BY ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО)

WHERE a=(

SELECT MAX(a)

FROM (

SELECT ФАМИЛИЯ, COUNT(ФАМИЛИЯ) a, ИМЯ,

COUNT(ИМЯ) b, ОТЧЕСТВО, COUNT(ОТЧЕСТВО) c

FROM Н\_ЛЮДИ

GROUP BY ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО

ORDER BY a, b, c

)

)

)

)

AND regexp\_like(ОЦЕНКА, '^[0-9]$')

AND ОЦЕНКА NOT IN ('99')

)

JOIN Н\_ЛЮДИ ON вч\_ИД =Н\_ЛЮДИ.ИД

WHERE СУММА\_ОЦЕНОК < (

SELECT 9 \* regexp\_count(a,'9') + 8 \* regexp\_count(a,'8') + 7 \*

regexp\_count(a,'7') + 6 \* regexp\_count(a,'6') + 5 \*

regexp\_count(a,'5') + 4 \* regexp\_count(a,'4') + 3 \*

regexp\_count(a,'3') + 2 \* regexp\_count(a,'2') + 1 \*

regexp\_count(a,'1')

FROM (

(SELECT ((SELECT ИД FROM (

SELECT ИД FROM Н\_ЛЮДИ ORDER BY dbms\_random.value )

WHERE rownum = 1)) a FROM dual)

)

);