

Уже сделаны полностью:

Блок 1: 1, 2, 3, 5,7,8,9,10(без кода),11

Блок 2: 1,2,3,4,6,7,8,9,12(доделать),13,14,15

Блок 3:

Блок 4:1,2,3,4,5,6,7,8

Блок

5:1,2,3,4,5,6,7,8(доправить),9(задумка),10,13  
и 14(идеи без кода)

Базы данных для самых маленьких и тупых

1 блок - однострочные функции, разделы  
Select, группировка и групповые (агрегатные)  
функции

1.1. Определить сумму цифр в произвольной символьной строке.

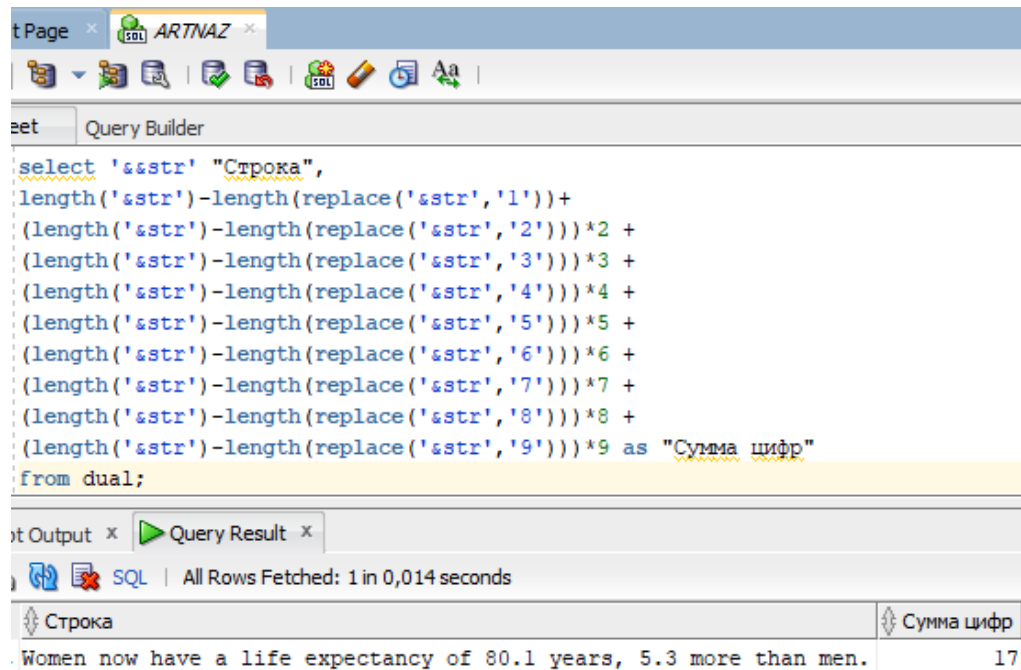
Пример результата:

Строка	Сумма цифр
Women now have a life expectancy of 80.1 years, 5.3 more than men.	17

1.1

```
select '&&str' "Строка",  
length('&str')-length(replace('&str','1'))+  
(length('&str')-length(replace('&str','2')))*2 +  
(length('&str')-length(replace('&str','3')))*3 +  
(length('&str')-length(replace('&str','4')))*4 +  
(length('&str')-length(replace('&str','5')))*5 +  
(length('&str')-length(replace('&str','6')))*6 +  
(length('&str')-length(replace('&str','7')))*7 +  
(length('&str')-length(replace('&str','8')))*8 +  
(length('&str')-length(replace('&str','9')))*9 as "Сумма цифр"  
from dual;  
undefine str;
```

Объявляем переменной входную строку (&& - чтобы не запрашивать ввод каждый раз), далее находим разницу длин при замене соответствующей цифры пробелом - так получим, сколько каждая цифра раз встречается в строке. Далее домножаем это на саму взятую цифру, чтобы грамотно суммировать, и, собственно, суммируем.



1.2. В 1845 году в США была установлена традиция, согласно которой выборы президента проводятся во вторник после первого понедельника ноября в год, делящийся на 4 без остатка. Определить, дату ближайших к заданной дате президентских выборов в США.

Примеры результата:

Заданная дата	Дата выборов
25.06.2012	12.11.2012
23.02.2003	08.11.2004

# 1.2

```
UNDEFINE current_date;
SELECT '&current_date' "Заданная дата",
CASE WHEN -- дата норм
  TO_NUMBER(SUBSTR('&current_date', 7)) >= -4712 AND TO_NUMBER(SUBSTR('&current_date', 7)) <= 9999
THEN
CASE WHEN -- если дата выборов до последних в 10000
  (TO_DATE('&current_date', 'DD.MM.SYYYY') <= TO_DATE('05.11.9996', 'DD.MM.YYYY'))
THEN
CASE WHEN -- до первых выборов
  (TO_DATE('&current_date', 'DD.MM.SYYYY') >= TO_DATE('01.01.1845', 'DD.MM.YYYY'))
THEN
CASE WHEN
  abs(TO_NUMBER(NEXT_DAY(TO_DATE((CONCAT('31.10.', TO_NUMBER(SUBSTR('&current_date', 7, 4))
+ 4 - MOD(TO_NUMBER(SUBSTR('&current_date', 7, 4)), 4))), 'DD.MM.YYYY'), 'MONDAY') + 1
- TO_DATE('&current_date', 'DD.MM.YYYY')) <=
abs(TO_NUMBER(NEXT_DAY(TO_DATE((CONCAT('31.10.', TO_NUMBER(SUBSTR('&current_date', 7, 4))
- MOD(TO_NUMBER(SUBSTR('&current_date', 7, 4)), 4))), 'DD.MM.YYYY'), 'MONDAY') + 1 - TO_DATE('&current_date',
'DD.MM.YYYY')))
THEN
```

```

        TO_CHAR(NEXT_DAY(TO_DATE((CONCAT('31.10.',TO_NUMBER(SUBSTR('&current_date', 7, 4)) + 4 -
MOD(TO_NUMBER(SUBSTR('&current_date', 7, 4)), 4))), 'DD.MM.YYYY'), 'MONDAY') + 1, 'DD.MM.YYYY')
    ELSE
        TO_CHAR(NEXT_DAY(TO_DATE((CONCAT('31.10.',TO_NUMBER(SUBSTR('&current_date', 7, 4)) -
MOD(TO_NUMBER(SUBSTR('&current_date', 7, 4)), 4))), 'DD.MM.YYYY'), 'MONDAY') + 1, 'DD.MM.YYYY')
    END
    ELSE
        TO_CHAR(NEXT_DAY(TO_DATE((CONCAT('31.10.',TO_NUMBER(SUBSTR('01.11.1848', 7, 4))), 'DD.MM.YYYY'), 'MONDAY') + 1,
'DD.MM.YYYY')
    END
    ELSE
        TO_CHAR(NEXT_DAY(TO_DATE((CONCAT('31.10.',TO_NUMBER(SUBSTR('&current_date', 7, 4)) -
MOD(TO_NUMBER(SUBSTR('&current_date', 7, 4)), 4))), 'DD.MM.YYYY'), 'MONDAY') + 1, 'DD.MM.YYYY')
    END
    ELSE
        'Валидные даты имеют год в промежутке между -4712 до 9999 года'
    END "Дата выборов"
FROM DUAL;

```

1.3. В произвольной символьной строке убрать все лидирующие и конечные пробелы, а между словами оставить только по одному пробелу.

Пример результата:

1.3

Исходная строка	Результат
.....This.....is the.....solution..... <sup>1</sup>	This is the solution

“...” => “()()()” => “()” => ‘ ’

```

UNDEFINE string_;
SELECT '&string_',
    LTRIM(RTRIM(REPLACE(REPLACE(REPLACE('&string_' , ' ' , '()'), '(',')', '(',')', ' ')))
FROM DUAL;

```

1.4. Определить дату начала ближайшего к заданной дате Уимблдонского турнира, который начинается за шесть недель до первого понедельника августа. Если заданная дата совпадает с датой начала турнира в этом году, вывести её.

Примеры результата:

1.4

Заданная дата	Дата начала турнира
25.06.2012	25.06.2012
26.06.2012	24.06.2013
27.12.2010	20.06.2011

1.5. Определить временной интервал между заданной и текущей датами. Результат вывести в виде: ## лет<sup>1</sup> ## мес ## дн, где # обозначает цифру, лидирующие нули и отсутствующие значения не выводить.

Примеры результата:

Заданная дата	Текущая дата	Временной интервал
25.06.2012	30.09.2012	3 мес 5 дн
04.07.2015	30.09.2012	2 года 9 мес 4 дн

1.5

with

input as ( -- Входные значения

select

to\_date('&&d1', 'dd.mm.syyy') as x,

trunc(sysdate) as y

from dual),

tmp1 as ( -- Сортируем x и y по возрастанию, вычисляем промеж. значения и вводим константу

select

case when x < y then x else y end as f, -- f < s

case when x < y then y else x end as s, -- f < s

abs(trunc(months\_between(x, y))) as diffmf, -- полные месяцы

abs(trunc(months\_between(x, y) / 12)) as diffyf, -- полные года

abs(months\_between(x, y)) as diffmd, -- дробные месяцы

trunc(to\_date('01.01.0001', 'DD.MM.YYYY')) as zero

from input),

tmp2 as ( -- Вычисляем значения разницы

select

case when diffyf = 0 or f > zero or s < zero -- если всё в одной эре

then diffyf -- то года и так высчитаны верно

else diffyf - 1 -- вычитаем фантомный нулевой год

end as years,

mod(diffmf, 12) as months, -- целые месяцы

case when to\_char(f, 'dd') <= to\_char(s, 'dd') -- Если даты и так "по порядку"

then abs(to\_char(f, 'dd') - to\_char(s, 'dd')) -- то просто выводим разницу

**else trunc( -- Если нет, то делаем вид, будто они в соседних  
месяцах:**

```
    add_months(to_date(to_char(s, 'dd.')  
    || to_char(f, 'mm.syyyy'), 'dd.mm.syyyy'), 1) - f -- и вычитаем  
    )  
end as days  
from tmp1)
```

```

select -- Отображаем вычисленные значения:
case when years + months + days = 0 --
then '0 дн'
else case -- Определяем нужное слово
    when mod(years, 10) = 1 and mod(years, 100) != 11
    then years || ' год'
    when mod(years, 10) in (2, 3, 4) and mod(years, 100) not in (12, 13, 14)
    then years || ' года'
    when years = 0
    then ''
    else
        years || ' лет'
    end
    || -- Ставить ли пробел после наименования года:
case when years > 0 and (days > 0 or months > 0)
then ''
else ''
end
|| -- Месяцы:
case when months > 0
then months || ' мес'
else ''
end
|| -- ставить ли пробел перед днём:
case when days > 0 and (years > 0 or months > 0)
then ''
else ''
end
|| -- Дни:
case when days > 0
then days || ' дн'
else ''
end
end
as result
from tmp2;

```

Суть в том, что мы находим разницу между датами в месяцах и делим на 12 с отрезанием, чтобы получить целое количество лет, далее вычисляем остаток от разности месяцев на 12, чтобы найти количество месяцев, оставшихся после вычленения всех целых лет, далее просто вычитаем дни (знаем, что они в пределах месяца, так как месяцы уже посчитаны). Далее много приведения типов (к дате, к чару, к числу), а также нужно не выводить пустые значения (например, если целых лет нет - не выводить 0 лет), а также для каждого значения лет подбирать конец "лет, год или года".

1.7. В названии отдела вывести только второе слово, если название состоит из двух или более слов. Иначе вывести первое слово.

Пример результата:

DEPARTMENT_NAME	RESULT
Accounting	Accounting
...	...
Control And Credit	And
...	...
Government Sales	Sales
...	...

1.7

```
select department_name,
case
when length(department_name)-length(replace(department_name,' ',''))>0
then ltrim(substr(department_name||' ',instr(department_name||' ',' '),instr(department_name||' ',' ',1,2)-instr(department_name||' ',' ')))
else department_name
end result
from departments;
```

Используем тот факт, что если слов 2 и более, то между ними есть минимум один пробел. Тогда мы дополняем в конце строку пробелом (для случая, когда слов ровно 2), и взимаем подстроку, начиная с 1ого вхождения пробела, до 2ого вхождения. Иначе возвращаем то, что и было.

Page	ARTNAZ
Query Builder	
<pre> select department_name, case when length(department_name)-length(replace(department_name,' ',''))&gt;0 then ltrim(substr(department_name  ' ',instr(department_name  ' ',' '),instr(department_name  ' ',' ',1,2)-instr(department_name  ' ',' '))) else department_name end result from departments;</pre>	
y Result x	
SQL   All Rows Fetched: 28 in 0,005 seconds	
DEPARTMENT_NAME	RESULT
Administration	Administration
Marketing	Marketing
Purchasing	Purchasing
Human Resources	Resources
Shipping	Shipping
IT	IT
Public Relations	Relations
Sales	Sales
Executive	Executive
Finance	Finance
Accounting	Accounting
Treasury	Treasury
Corporate Tax	Tax
Control And Credit	And

1.11. Написать команду, которая позволит выводить фамилию сотрудника через запятую столько раз, сколько букв в его фамилии.

Пример результата:

LAST_NAME
OConnell, OConnell, OConnell, OConnell, OConnell, OConnell, OConnell, OConnell
Grant, Grant, Grant, Grant, Grant
...

1.11

```

select
rpad(last_name,length(last_name)*length(last_name)+2*(length(last_name)-1),
'||last_name) last_name
from employees;
```

Буду дополнять строку справа ей же самой с запятой и пробелом до тех пор, пока длина не станет равна квадрату длины фамилии + еще два символа(запятая и пробел) между словами.



Page	ARTNAZ
Query Builder	
select	rpadd(last_name,length(last_name)*length(last_name)+2*(length(last_name)-1),' '  last_name) last_name
from	employees;
Result	SQL   Fetched 100 rows in 0,057 seconds
LAST_NAME	
Abel, Abel, Abel, Abel	
Ande, Ande, Ande, Ande	
Atkinson, Atkinson, Atkinson, Atkinson, Atkinson, Atkinson, Atkinson, Atkinson	
Austin, Austin, Austin, Austin, Austin, Austin	
Baer, Baer, Baer, Baer	
Baida, Baida, Baida, Baida, Baida	
Banda, Banda, Banda, Banda, Banda	
Bates, Bates, Bates, Bates, Bates	
Bell, Bell, Bell, Bell	
Bernstein, Bernstein, Bernstein, Bernstein, Bernstein, Bernstein, Bernstein, Bernstein, Bernstein	

1.9. Найти все (вещественные и/или комплексные) корни квадратного уравнения  $ax^2 + bx + c = 0$  для заданных коэффициентов  $a$ ,  $b$  и  $c$ . Результат округлить до двух цифр после запятой.

Примеры результата:

a	b	c	Корни квадр. уравнения
2	7	6	Корень 1: -1.50; корень 2: -2.00
1.5	14	8	Корень 1: -0.61; корень 2: -8.72
3	9	9	Корень 1: -1,5+0.87*i; корень 2: -1,5-0.87*i
0	7	4	Не является квадратным уравнением

1.9

```

SELECT A "a", B "b", C "c",
CASE
WHEN A=0 THEN 'Не является квадратным уравнением'
WHEN (B*B-4*A*C) = 0 THEN 'Корень: ' || TO_CHAR(ROUND(-B/(2*A),2),'FM90.00')
WHEN (B*B-4*A*C) > 0 THEN 'Корень 1: ' || TO_CHAR(ROUND((-B + SQRT(B*B - 4*A*C))/(2*A),2),'FM90.00') ||
'; корень 2: ' || TO_CHAR(ROUND((-B - SQRT(B*B - 4*A*C))/(2*A),2),'FM90.00')
WHEN (B*B-4*A*C) < 0 THEN 'Корень 1: ' || -B/(2*A) || TO_CHAR(ROUND(SQRT(-(B*B -
4*A*C))/(2*A),2),'SFM90.00')
|| '*i; корень 2: ' || -B/(2*A) || TO_CHAR(-ROUND(SQRT(-(B*B - 4*A*C))/(2*A), 2), 'SFM90.00') || '*i'
END AS "Корни квадр. уравнения"
FROM squares;

```

Единственное, что - как в маске учесть, если цифр очень много?

1.8. Создать таблицу Test, состоящую из двух столбцов ID и Text. Столбец ID — ключевой, а столбец Text содержит символьную информацию, включающую цифры и символы латинского алфавита, при этом либо буквенная, либо цифровая части могут отсутствовать. Требуется создать запрос, который выведет информацию из второго столбца в отсортированном виде. Правила сортировки следующие:

- выражения, начинающиеся с буквы, вывести первыми;
- символьные выражения, начинающиеся с чисел, должны быть отсортированы в порядке увеличения чисел, а в случае наличия выражений с одинаковыми числами — по алфавиту той части выражения, которая следует за числом, считая число менее приоритетным;
- в случае наличия выражений, начинающихся с одинакового числа, некоторые из которых не содержат буквенной части, первыми вывести числа.

1.8

```
SELECT TEXT
FROM TEST
ORDER BY CASE WHEN
LTRIM(SUBSTR(LOWER(TEXT),1,1),'abcdefghijklmnopqrstuvwxyz') IS NULL
THEN TEXT END NULLS LAST,
TO_NUMBER(SUBSTR(TEXT,1,LENGTH(TEXT)-NVL(LENGTH(LTRIM(TEXT,'0123456789')),0))),
CASE WHEN (TRIM(LTRIM(TEXT,'0123456789')) IS NOT NULL) THEN TEXT END
NULLS FIRST;
```

Первый CASE отвечает за то, что первый символ - один из алфавитных, то есть буква, далее мы указываем, что такие (с буквой) нужно вывести первыми.

Далее мы рассматриваем число, то есть состоящее только из цифр и сортируем их, впоследствии мы смотрим числа с символами и сортируем их.

Нужно обсудить и потестируй!!! Здесь LTRIM используется интересно, для убиения любых символов, входящих далее в строку в кавычках

1.10. Создать запрос для определения произведения количеств сотрудников в отделах.

Пример результата:

Intersection
3304800

1.10

Поскольку агрегатной функции умножения нет, то для этого используется представление произведения как суммы логарифмов, например,  $\ln 2 + \ln 3 = \ln 6$ , и далее от результата берется экспонента, чтобы получить  $\exp(\ln 2 + \ln 3) = \exp(\ln 6) = 6$ . На этом построено все решение.