1. Создать запрос для вывода списка всех столбцов представления ALL_TABLES. Имена столбцов должны быть отсортированы по алфавиту и разделяться запятыми. Список должен быть разбит на строки, при этом каждая строка должна содержать не более 50 символов. Имя каждого столбца должно размещаться целиком на одной строке. Если после имени столбца следует запятая, то она должна находиться на строке вместе с именем столбца.

   Результат запроса должен также содержать номер строк и количество столбцов в строке. Пример представления результата:

Имя представления	**Номер строки**	**Список столбцов**	**Количество столбцов**
ALL_TABLES	1	ACTIVITY_TRACKING, AVG_ROW_LEN, AVG_SPACE	3
	2	AVG_SPACE_FREELIST_BLOCKS, BACKED_UP, BLOCKS	3
	3	……	

```
with inp_col as (   --получаем все столбцы представления ALL_TABLES
select
row_number() over (order by column_name) r,
column_name
from all_tab_columns
where table_name = 'ALL_TABLES' order by 1),

find_rows as(
select
col_list,
rn, cnt
from inp_col
model
dimension by (r)
measures(column_name, cast('' as varchar2(1000)) as col_list, cast(0 as number(2)) as rn, cast(0 as number(2)) as cnt)
rules iterate (100) (
   --если строка не превышает размер 50, то к нему добавляется новый столбец. иначе - новая строка
   col_list[iteration_number + 1] = case when length(ltrim(col_list[iteration_number] || column_name[iteration_number + 1] || ',', ',')) > 50
                                 then column_name[iteration_number + 1] || ','
                                 else ltrim(col_list[iteration_number] || column_name[iteration_number + 1] || ',', ',') end,
   --если строка не превышает размер 50, то оставляется номер строки. иначе - предыдущий номер + 1
   rn[iteration_number + 1] = case when iteration_number + 1 = 1 then 1
                                when length(ltrim(col_list[iteration_number] || column_name[iteration_number + 1] || ',', ',')) > 50
                                then rn[iteration_number] + 1
                                else rn[iteration_number] end,
   --колчиество столбцов равно количеству запятых в строке
   cnt[iteration_number + 1] = regexp_count(col_list[iteration_number + 1], ',')
   )
```

```
),

find_rows2 as( --подчищаем лишние строки. убираем запятую у последней строки
select
case when rownum = (select max(r) from inp_col) then rtrim(col_list,',') else col_list end as
col_list,
rn, cnt
from find_rows
where rownum <= (select max(r) from inp_col))

select --выводим получившиеся строки, где длина строки каждого номера строки
максимальна
case when rownum = 1 then 'ALL_TABLES' else ' ' end "Имя представления",
rn "Номер строки",
col_list "Список столбцов",
cnt "Количество столбцов"
from find_rows2 f1
where length(col_list) = (select max(length(col_list)) from find_rows2 f2 where f1.rn = f2.rn);
```

1. Создать таблицу Пользователи, содержащую 2 столбца: Номер number(15,0) Primary Key и Фамилия varchar2(40). Создать запрос, который будет выводить значение столбца Номер и Номер группы последовательных целых значений с шагом 1. Например, для таблицы, содержащей значения:

Номер	**Фамилия**
1	Mougus
2	Green
3	Grase
7	Scott
8	Trumen
10	Kochhar
12	Drejk
13	Kook

результат должен быть:

Номер	**Номер группы**
1	1
2	1
3	1
7	2
8	2
10	3
12	4
13	4

```
create table b1e2(
   "Номер" number(2,0) Primary Key,
   "Фамилия" varchar2(40)
);
insert into b1e2("Номер","Фамилия")
select
1,         'Mougus'
from dual
```

```
union all
select
2,        'Green'
from dual
union all
select
3,        'Grase'
from dual
union all
select
7,        'Scott'
from dual
union all
select
8,        'Trumen'
from dual
union all
select
10,       'Kochhar'
from dual
union all
select
12,       'Drejk'
from dual
union all
select
13,       'Kook'
from dual;
with f1 as(
select
rownum as rn,
"Номер" as num
from b1e2
order by 1
),
f2 as (
select
rn,
num,
grnum
from f1
model
dimension by(rn)
measures(num, cast('' as number(2,0)) as grnum)
rules iterate(40)(
   grnum[iteration_number+1] = case
               when grnum[iteration_number] is null
               then 1
               when num[iteration_number+1] = num[iteration_number] + 1
               then grnum[iteration_number]
```

```
            else grnum[iteration_number] + 1
            end
    )
)
select
num as "Номер",
grnum as "Номер группы"
from f2
where num is not null;
```

2. В таблице Employees находится информация о фамилиях сотрудников, их зарплатах и номерах отделов, в которых они работают. Для каждого отдела вывести фамилии и зарплаты трех сотрудников, получающих самые высокие зарплаты в отделе. Если самую низкую зарплату у найденных трех сотрудников отдела получают и какие-то другие сотрудники этого отдела, они тоже должны попасть в список. Для отделов, в которых меньше трех сотрудников, информацию не выводить.

Примечание. Если в таблице имеются сотрудники с максимальной зарплатой и с одинаковой фамилией в одном и том же отделе, то необходимо вывести повторяющиеся значения.
Пример результата.

```
with f1 as(
select
department_id,
count(*) over (partition by department_id) as cnt,
last_name,
salary,
dense_rank() over (partition by department_id order by salary desc) as rnk
from employees
where salary is not null
order by 1),
f2 as(
select
row_number() over (order by department_id, rnk) as rn,
count(*) over (partition by department_id) as cntd,
department_id,
last_name,
salary,
rnk
from f1
where cnt >= 3 and rnk <= 3),
f3 as(
select
department_id,
last_name,
salary
from f2
model
dimension by(rn)
measures(cntd, department_id, last_name, salary, rnk)
rules iterate (100)(
```

```
   last_name[iteration_number+1] = case
      when cntd[iteration_number+1] > 3
      and rnk[iteration_number + 1] = 1
      and rnk[iteration_number] = 1
      then ''

      when cntd[iteration_number+1] > 3
      and rnk[iteration_number + 1] = 2
      and rnk[iteration_number] = 2
      then ''

      else last_name[iteration_number+1]
      end
   )
)

select *
from f3
where last_name is not null
order by 1,3 desc;
```

1. Создать запрос для разделения "задвоенных" данных. Например, из

CODE_OPERATION	ID_CLIENT
1000 1100	841000 841100
2000	6700 8967 5500

сделать

RN	CNT	CODE_OPERATION	ID_CLIENT
1	0	1000 1100	841000 841100
	1	1000	841000
	2	1100	841100
2	0	2000	6700 8967 5500
	1	2000	6700
	2		8967
	3		5500

```
with db as (
select
'1000 1100' as code_operation,
'841000 841100' as id_client
from dual
union all
select
'2000',
'6700 8967  5500'
from dual
),
f1 as(
select
rownum rn,
```

```
code_operation,
id_client
from db
),
f2 as(
select
rn,
rn2,
code_operation,
id_client
from f1
model
dimension by (rn, cast (0 as number(2,0)) as rn2)
measures (code_operation,id_client)
rules upsert all iterate (20)(
   code_operation[any, iteration_number] = case
                                    when iteration_number = 0
                                    then code_operation[cv(), iteration_number]
                                    else ltrim(regexp_substr(code_operation[cv(), 0], '\s?\d+\s?', 1,
iteration_number), ' ')
                                    end,
   id_client[any, iteration_number] = case
                                    when iteration_number = 0
                                    then id_client[cv(), iteration_number]
                                    else ltrim(regexp_substr(id_client[cv(), 0], '\s?\d+\s?', 1, iteration_number), '
')
                                    end
                                 )
   )
select
case when rn2 = 0 then to_char(rn) else ' ' end as ROWN,
rn2 as cnt,
nvl(code_operation, ' ') as code_operation,
nvl(id_client, ' ') as id_client
from f2
where
code_operation is not null or id_client is not null
order by rn, 2;
```

2. Используя словарь данных, получить информацию об ограничениях CHECK схемы.

В списках имена столбцов вывести через запятую. Имя таблицы не должно повторяться.

Задачу решить без использования функций Listagg и Wm_concat.

Пример представления результатов:

SQL | All Rows Fetched: 8 in 1,748 seconds

	Имя таблицы	Имя ограничения	Столбцы,входящие в ограничение	Ограничение CHECK
1	EMPLOYEES	EMP_SALARY_MIN	SALARY	salary > 0
2	JOB_HISTORY	JHIST_DATE_INTERVAL	END_DATE,START_DATE	end_date > start_date
3	TASK_13	COLUMN12345	COLUMN_1,COLUMN_2,COLUMN_3,COLUMN_4,COLUMN_5	column_1>ALL (column_2, column_3, column_4, column_5)
4	TRIP	CON_PER_BEG	PER_BEG	per_beg=TRUNC(per_beg)
5		CON_PER_END	PER_END	per_end=TRUNC(per_end)
6	ПРЕПАРАТ	ПРЕПАРАТ_CHK1	АББРЕВ_ФОРМЫ,ТИП	Аббрев_формы!=Тип
7	ПРЕПОДАВАТЕЛЬ	SYS_C00201812	КАФЕДРА	КАФЕДРА Like 'Кафедра%'
8	УСПЕВАЕМОСТЬ	SYS_C00201820	ОЦЕНКА	ОЦЕНКА between 1 and 5

```
with f1 as(
select
uc.table_name,
uc.constraint_name,
ucc.column_name
from user_constraints uc left join user_cons_columns ucc
on (uc.constraint_name = ucc.constraint_name)
where uc.constraint_type = 'C'
),
f2 as (
select
table_name,
dense_rank() over (partition by table_name order by constraint_name) rnk,
constraint_name,
dense_rank() over (partition by constraint_name order by column_name) rnk2,
column_name
from f1
order by 1, 3),
f3 as (
select
table_name, rnk, constraint_name, rnk2, column_name
from f2
model
dimension by (table_name, rnk, rnk2)
measures (constraint_name, column_name)
rules iterate (20) (
   column_name[any, any, iteration_number + 1] = ltrim (column_name[cv(), cv(), iteration_number] ||
',' || column_name[cv(), cv(), iteration_number + 1] ,','),
   column_name[any, any, iteration_number] = case when column_name[cv(), cv(), iteration_number +
1] is not null
                    then null
                     else column_name[cv(), cv(), iteration_number]
                                  end
)
order by 1,3
)
select
```

```
case when ff.rnk = 1 then ff.table_name else ' ' end as "Имя таблицы",
constraint_name as "Имя ограничения",
ff.column_name as "Столбцы, входящие в ограничения",
uc.search_condition as "Ограничение CHECK"
from f3 ff left join user_constraints uc
using (constraint_name)
where ff.column_name is not null
order by ff.table_name, constraint_name;
```

3. Определить список последовательностей подчиненности от преподавателей, не имеющих начальника, до преподавателей, не имеющих подчиненных. Если список состоит более, чем из четырех фамилий, то выводить только две первые и две последние фамилии, а вместо остальных фамилий поставить многоточие. У преподавателей, не имеющих подчиненных приписать – (не имеет подчиненных). Если в списке четыре или меньше фамилий, то список выводится полностью.
   Результат представить в виде:

   Костыркин-> Викулина-> …->Соколов->Казанко (не имеет подчиненных)

. . . . . .

```
with f1 as(
select
level as lvl,
ltrim(sys_connect_by_path("ФАМИЛИЯ", '->'),'->') as path
from "ПРЕПОДАВАТЕЛИ" p
where not exists (
        select 'X'
        from "ПРЕПОДАВАТЕЛИ"
        where "ПОДЧИНЯЕТСЯ" = p."НОМЕР_ПРЕПОДАВАТЕЛЯ")
start with "ПОДЧИНЯЕТСЯ" is null
connect by "ПОДЧИНЯЕТСЯ" = prior "НОМЕР_ПРЕПОДАВАТЕЛЯ"
)
select
case when lvl > 4 then
regexp_replace(path, '^(\w+\-\>\w+\-\>)(\w+\-\>)+(\w+\-\>\w+)$', '\1...->\3')
else path end || '(не имеет подчиненных)'as result
from f1;
```

2. Имеется таблица с двумя столбцами – дочерняя вершина и родительская вершина. Определить наборы вершин, образующих связанные множества. Например, для таблицы:

Дочерняя вершина	Родительская вершина	
1	2	
2	4	
4	5	
4	3	
7	6	

результат должен быть

Связанные множества
1,2,3,4,5
6,7

```
with tabl as(
select
1 as "Дочерняя вершина", 2 as "Родительская вершина"
from dual
union all
select
2 as "Дочерняя вершина", 4 as "Родительская вершина"
from dual
union all
select
4 as "Дочерняя вершина", 5 as "Родительская вершина"
from dual
union all
select
4 as "Дочерняя вершина", 3 as "Родительская вершина"
from dual
union all
select
7 as "Дочерняя вершина", 6 as "Родительская вершина"
from dual
),
tab1 as (
select
"Дочерняя вершина" as child_number,
"Родительская вершина" parent_number
from tabl
),
f1 as(
select
connect_by_root ( child_number ) ch,
parent_number
from tab1
connect by prior parent_number = child_number
order by ch),
f2 as (
select *
from f1
```

```
where ch not in (
   select parent_number
   from f1
   )
),
f3 as (
select
ch,
ch || sys_connect_by_path(parent_number,',') str
from f2
connect by prior ch = ch
and prior parent_number < parent_number)

select str as "Связанные множества"
from f3
where (ch, length(str)) in (
                        select
                        ch,
                        max(length(str))
                        from f3
                        group by ch);
```

1. Используя обращение только к таблице DUAL, построить SQL-запрос, возвращающий один столбец, содержащий календарь на заданный месяц заданного года:

   · номер дня в месяце (цифрами),

   · полное название месяца по-английски заглавными буквами (в верхнем регистре),

   · год (четыре цифры),

   · полное название дня недели по-английски строчными буквами (в нижнем регистре).

   Каждое "подполе" должно быть отделено от следующего одним пробелом. В результате не должно быть начальных и хвостовых пробелов. Количество возвращаемых строк должно точно соответствовать количеству дней в текущем месяце. Строки должны быть упорядочены по номерам дней в месяце по возрастанию.

   Календарь должен создаваться для любых допустимых значений дат Oracle.

   Задачу решить без использования разделов Model и рекурсивногоWith.

   Пример вывода результата:

   1 MAY 2020 friday

   2 MAY 2020 saturday

   ……

```
undefine x;

with impdate as(
select
to_char(trunc(to_date('&&X','DD.MM.SYYYY'), 'MONTH'),'DD.MM.SYYYY') dd
from dual
),
f1 as (
select
to_date(dd,'DD.MM.SYYYY') + level - 1 as daylist
from impdate
connect by level < 32
)
select
rtrim(regexp_replace(to_char(f1.daylist, 'fmDD MONTH SYYYY day',
'NLS_Date_Language=English'),'\s\s', ' '),' ') as result
from f1 join impdate impd
on (extract(month from f1.daylist) = extract(month from to_date(impd.dd,'DD.MM.SYYYY')))
order by daylist;
```

3. Написать запрос, который все пары прямых скобок в строке, внутри которых имеется две или более пары прямых скобок, заменит на фигурные скобки. Например, для строки

[[[98+77]-9]-1] => [[175-9]-[1]]=>165

результат должен быть

{[[98+77]-9]-1} =>{[175-9]-[1]}=>165

```
undefine x;

with impstr as(
select
'&&x' as str
from dual),
f1 as(
select
str,
curi_n
from impstr
model
dimension by (cast(0 as number(3)) as i_n)
measures (str, cast(0 as number(3)) as curi_n, cast(0 as number(3)) as maxi_n)
rules iterate(100) (
  str[iteration_number+1] = case when iteration_number + 1 = 0
                        then str[cv()]
                        else
regexp_replace(str[iteration_number],'\[(([^][]*\[[^][]*\[[^][]*\][^][]*\][^][]*)|([^][]*\[[^][]*\][^][]*\[[^][
]*\][^][]*))\]','{\1}')
                        end,
  str[iteration_number+1] = case when str[iteration_number+1] = str[iteration_number] then null else
str[iteration_number+1] end,
  curi_n[iteration_number]= iteration_number
                        )
                    )
select
str as rezult
from f1
where curi_n in (select max(curi_n) from f1 where str is not null);
```

1. Определить многостолбцовые ограничения для каждой таблицы схемы.

Результат представить в виде:

Номер таблицы	Таблица	Номер ограничения	Имя ограничения	Тип ограничения	Кол-во столбцов	Кол-во многостолбцовых ограничений
1	JOB_HISTORY	1	JHIST_EMP_ID_ST_DATE_PK	Первичный ключ	2	2
		2	JHIST_DATE_INTERVAL	Ограничение CHECK	2	
2	СОТРУДНИК_ПРОЕКТ	1	ХРКСОТРУДНИК_ПРОЕКТ	Первичный ключ	2	1

Данные должны быть отсортированы по названию таблиц по алфавиту.

Номера таблиц должны быть заданы в соответствии с указанной сортировкой.

Номер таблицы, название таблицы и количество многостолбцовых ограничений не должны повторяться для одной таблицы.
Для каждой таблицы данные должны быть отсортированы по именам ограничений.
Номера ограничений в таблицах должны быть заданы в соответствии с указанной сортировкой.
Информация о таблицах без многостолбцовых ограничений также должна выводиться с указанием, что таких ограничений нет (их количество равно нулю)

```
with f1 as(
select distinct
uc.table_name,
constraint_name,
uc.constraint_type,
count(ucc.column_name) over (partition by constraint_name) as col_cnt
from user_constraints uc join user_cons_columns ucc
using(constraint_name)
order by 1,3),
f2 as(
select
tab_num,
table_name,
cons_num,
constraint_name,
constraint_type2,
col_cnt,
conl_cnt_check,
cons_cnt
from f1
model
dimension by(dense_rank() over (order by table_name) as tab_num, dense_rank() over (partition by
table_name order by constraint_name) as cons_num)
measures(table_name, constraint_name, constraint_type, col_cnt, cast(0 as number(2)) as
conl_cnt_check, cast(0 as number(2)) as cons_cnt, cast('' as varchar2(40)) as constraint_type2)
rules (
```

```
   conl_cnt_check[any, any] = case when col_cnt[cv(), cv()] > 1 then conl_cnt_check[cv(), cv()] + 1 else
conl_cnt_check[cv(), cv()] end,
   cons_cnt[any, any] = sum(conl_cnt_check)[cv(), any],
   constraint_type2[any, any] = case when constraint_type[cv(), cv()] = 'C' then 'Ограничение CHECK'
                        when constraint_type[cv(), cv()] = 'P' then 'Первичный ключ'
                        when constraint_type[cv(), cv()] = 'U' then 'Ограничение UNIQUE'
                        when constraint_type[cv(), cv()] = 'R' then 'Ограничение FOREIGN KEY'
                        end
   )
order by 1,3
),
f3 as (
select distinct
table_name,
case when cons_cnt = 0 then 'Многостолбцовых ограничений нет' when cons_cnt<>0 and col_cnt = 1
then null else constraint_name end as cons_name,
case when cons_cnt = 0 then ' ' else constraint_type2 end as cons_type,
case when cons_cnt = 0 then ' ' else to_char(col_cnt) end as col_cnt,
case when cons_cnt = 0 then ' ' else to_char(cons_cnt) end as cons_cnt
from f2
order by 1, 2),
f4 as (
select
dense_rank() over (order by table_name) as tab_num,
table_name,
dense_rank() over (partition by table_name order by cons_name) as cons_num,
cons_name,
cons_type,
col_cnt,
cons_cnt
from f3
where cons_name is not null
order by 1,3
)
select
case when cons_num = 1 then to_char(tab_num) else ' ' end as "Номер таблицы",
case when cons_num = 1 then table_name else ' ' end as "Таблица",
case when cons_cnt = ' ' then ' ' else to_char(cons_num) end as "Номер ограничения",
cons_name as "Имя ограничения",
cons_type as "Тип ограничения",
col_cnt as "Кол-во столбцов",
case when cons_num = 1 then cons_cnt else ' ' end as "Кол-во многостолб. ограничений"
from f4;
```

2. Имеется таблица с символьным столбцом. Создать запрос для вывода тех значений, которые содержат в себе палиндромы, и самые длинные выражения, представляющие из себя палиндром.

Например, для таблицы с данными:

9. Text
10. Крокодил
11. Колокол
12. Станок

13. Результат должен быть:

14. Text	**15. Palindrom**
16. Крокодил	17. око
18. Колоколл	19. Колок, локол

Задачу решить без использования иерархических запросов и недокументированной функции Reverse.

Примечание: Палиндромом называется слово или фраза, которые одинаково читаются слева направо и справа налево

```
with mytable as(
select
'Крокодил' as text
from dual
union all
select
'Колокол'
from dual
union all
select
'Станок'
from dual
union all
select
'Шалаши'
from dual
union all
select
'Борода'
from dual
),
f1 as(
```

```
select
rn,
text,
st_sym,
st_sym_i
from mytable
model
dimension by(rownum as rn, cast(1 as number(2)) i_n)
measures(text, cast(0 as number(2)) st_sym, cast(0 as number(2)) st_sym_i)
rules upsert all iterate(20) (
   st_sym [any, iteration_number] = iteration_number + 1,
   st_sym_i [any, iteration_number] = iteration_number + 1,
   text[any, iteration_number] = text[cv(),1]
   )
order by 1,2),
f2 as(
select
text,
st_sym,
substr,
i_n
from f1
model
dimension by(rn, st_sym_i, cast(0 as number(2)) i_n)
measures(text, st_sym, cast('' as varchar2(20)) substr)
rules upsert all iterate(20) (
   text[any, any, iteration_number] = text[cv(),1,0],
   st_sym[any, any, iteration_number] = st_sym[cv(), cv(), 0],
   substr[any, any, iteration_number] = substr(text[cv(), cv(), cv()], st_sym[cv(), cv(), cv()],
iteration_number)
   )
),
f3 as(
select distinct
text,
substr
from f2
where substr is not null and length(substr) > 2
order by 1,2
),
f4 (text, substr, r_substr, num) as (
select
text,
substr,
'' as r_substr,
0 as num
from f3
union all
select
text,
```

```
substr,
r_substr || substr(substr, -(num + 1), 1) as r_substr,
num + 1 as num
from f4
where num + 1 <= length(substr)
),
f5 as(
select
text,
substr,
r_substr,
num,
max(num) over (partition by substr) as maxnum
from f4
),
f6 as (
select
text,
substr,
r_substr,
ultrastring,
rn1,
rn2
from f5
where num = maxnum
model
dimension by (dense_rank() over (order by text) as rn1, dense_rank() over (partition by text order by
substr) as rn2)
measures (text, substr, r_substr, cast('' as varchar2(20)) as ultrastring)
rules iterate(20)(
   ultrastring[any, 1] =
          case when lower(substr[cv(), iteration_number]) = lower(r_substr[cv(), iteration_number])
          then ltrim(ultrastring[cv(), 1] || ',' || substr[cv(), iteration_number], ',')
          else ultrastring[cv(), 1]
          end
   )
order by 5,6
)
select
text as "Text",
ultrastring as "Palindrom"
from f6
```

3. Имеется таблица Продажи (Номер, Название товара, Дата, Скидка %). Вывести отчет по продажам, который включает столбцы Название товара, Даты продажи, Скидка %, представив информацию таким образом, что если один и тот же товар продавался с одной и той же скидкой несколько дней, то эти даты должны выводиться через запятую. При этом если две или более даты отличаются друг от друга на один день, то они должны быть представлены в виде интервала с дефисом в качестве разделителя.

Название товара	**Даты продажи**	**Скидка, %**
Стул	1.02.2016, 5.02.2016, 7.02.2016-12.02.2016, 15.02.2016	5
Стол	2.02.2016, 4.02.2016	10
Кровать	2.02.2016, 6.02.2016 - 7.02.2016, 12.02.2016-15.02.2016	10
	……..	

```
with tabl as (
select
1 as id,
'Стул' as name,
to_date('01.02.2023','dd.mm.yyyy') as sdate,
5 as disc
from dual
union all
select
2,
'Стул',
to_date('02.02.2023','dd.mm.yyyy'),
5
from dual
union all
select
3,
'Стул',
to_date('04.02.2023','dd.mm.yyyy'),
5
from dual
union all
select
4,
'Стул',
to_date('02.02.2023','dd.mm.yyyy'),
10
from dual
union all
select
5,
'Стол',
to_date('01.02.2023','dd.mm.yyyy'),
5
from dual
```

```
union all
select
10,
'Стол',
to_date('02.02.2023','dd.mm.yyyy'),
5
from dual
union all
select
6,
'Стул',
to_date('31.01.2023','dd.mm.yyyy'),
5
from dual
union all
select
7,
'Стул',
to_date('05.02.2023','dd.mm.yyyy'),
5
from dual
),
f1 as (
select
id,
name, sdate,
lead(sdate, 1, sdate) over(partition by name, disc order by sdate asc) - sdate as lead_d,
sdate - lag(sdate, 1, sdate) over(partition by name, disc order by sdate asc) as lag_d,
disc
from tabl),
f2 as(
select id, name, disc, sdate,
case
when lead_d=1 then to_char(sdate, 'fmddfm.mm.yyyy-')
when lead_d=0 then to_char(sdate, 'fmddfm.mm.yyyy')
else to_char(sdate, 'fmddfm.mm.yyyy,')
end str
from f1
where not(lag_d=1 and lead_d=1)
)
select
name "Название товара",
listagg(str,'') within group (order by sdate) "Даты продажи",
disc "Скидка, %"
from f2
group by name, disc;
```

1. Используя словарь данных, получить информацию о первичных ключах и подчиненных таблицах всех таблиц схемы.

В списках имена столбцов и подчиненных таблиц вывести через запятую по алфавиту.

Задачу решить без использования функций Listagg и Wm_concat.

Имя таблицы	Список столб. перв. ключа	Список подчин. таблиц
EMPLOYEES2	ID	Подчиненных таблиц нет
JOB_SUM_SAL	Первичного ключа нет	Подчиненных таблиц нет
TASK	ID	Подчиненных таблиц нет
ВАКАНСИЯ	КОД_ПОЗИЦИИ,ДАТА_НАЧАЛА_ДОГОВОРА,КОД_КОМПАНИИ	СОБЕСЕДОВАНИЕ
КОМАНДА_ПРОЕКТА	НОМЕР_ДОГОВОРА_СОТРУДНИКА,НОМЕР_ПРОЕКТА	Подчиненных таблиц нет
НАПРАВЛЕНИЕ	КОД_НАПРАВЛЕНИЯ	СТУДЕНТ,ГРУППА,ПЛАН_ОБУЧЕНИЯ
ПРЕПОДАВАТЕЛИ	НОМЕР_ПРЕПОДАВАТЕЛЯ	ПРЕПОДАВАТЕЛИ,ДИСЦИПЛИНЫ
СТУДЕНТ	НОМЕР_ЗАЧЕТНОЙ_КНИЖКИ	БЮДЖЕТНИК,ВЕДОМОСТЬ,КОНТРАКТНИК
УЧЕБНОЕ_ЗАВЕДЕНИЕ	НАЗВАНИЕ_ЗАВЕДЕНИЯ	ОБРАЗОВАНИЕ
ERWIN_ИЗДАТЕЛЬСТВО	НАЗВАНИЕ_ИЗДАТЕЛЬСТВА	ERWIN_КНИГА
NEW_СОТРУДНИК	НОМЕР	NEW_УЧАСТНИКИ,NEW_СОТРУДНИК,NEW_ОТДЕЛ,NEW_ПРОЕКТ
SALES_SOURCE_DATA	Первичного ключа нет	Подчиненных таблиц нет
ГРУППЫ	НОМЕР_ГРУППЫ	СТУДЕНТЫ

```
with pkeys as(
select
uc.table_name,
ucc.column_name,
constraint_name
from user_constraints uc join user_cons_columns ucc
using (constraint_name)
where constraint_type = 'P'),

f1 as(
select
pk.table_name as p_table,
pk.column_name,
uc.table_name as f_table
from pkeys pk left join user_constraints uc
on (uc.r_constraint_name = pk.constraint_name)),

f2 as (
select
p_table,
column_name,
col_list,
f_table,
tab_list
from f1
model
dimension by (dense_rank() over (order by p_table) as tablrnk, dense_rank() over (partition by p_table order by column_name) as rn, dense_rank() over (partition by p_table, column_name order by f_table) as rn2)
measures (p_table, column_name, f_table, cast ('' as varchar2(100)) as col_list, cast ('' as varchar2(100)) as tab_list)
rules iterate(20) (
   col_list[any, 1, 1] = rtrim(ltrim(col_list[cv(), cv(), cv()] || ',' ||  column_name[cv(), iteration_number, cv()], ','), ','),
   tab_list[any, 1, 1] = rtrim(ltrim(tab_list[cv(), cv(), cv()] || ',' ||  f_table[cv(), cv(), iteration_number], ','), ',')
   )
)

select
```

```
p_table as "Имя таблицы",
col_list as "Список столб. перв. ключа",
tab_list as "Список подчин. таблиц"
from f2
where col_list is not null and tab_list is not null
order by 1;
```

3. Для произвольной строки, состоящей из цифр, определить все возможные наборы слов, получаемые при замене чисел на номер буквы в русском алфавите.

Например, для строки 211221 результатом должна быть строка:

баабба, бйбба,баафа,бакба,уку,……

```
WITH src as( --формирование исходной строки
   SELECT '211221' str
   FROM dual
),
--формируем все числа, которые встречаются в строке, так как нам нужен номер буквы алфавита, то это число либо одно-, либо дву-значное
all_nums AS (
--формируем однозначные числа
   SELECT str,to_number(substr(str, level, 1)) num,
      level AS pos,
      length(str) AS len
   FROM src
   CONNECT BY level <= length(str)
   UNION
--формируем двузначные числа
   SELECT str, to_number(substr(str, level, 2)),
      level,
      length(str) AS len
   FROM src
   CONNECT BY level <= length(str) - 1
),
--фильтруем номера, которые не соответствуют позиции буквы в алфавите и заодно добавляем позицию начала и конца номера в первоначальной строке
all_nums_filter AS (
  SELECT num, pos,
      CASE
         WHEN length(to_char(num))=1
            THEN pos
         ELSE pos+1
      END pos_end
  FROM all_nums
  WHERE num BETWEEN 1 AND 33
),
--собираем полученные числа в последовательность букв по правилу: конец предыдущего слова равен началу следующего
seq AS (
   SELECT SYS_CONNECT_BY_PATH(SUBSTR('абвгдеёжзийклмнопрстуфхцчшщъыьэюя', num, 1), ' ') path,
         CONNECT_BY_ISLEAF isleaf
   FROM all_nums_filter
   START WITH pos=1
   CONNECT BY PRIOR pos_end = pos-1
)
--вывод всех возможных наборов слов
SELECT LISTAGG(replace(path, ' ', ''),',') WITHIN GROUP(ORDER BY 1) res
FROM seq
WHERE isleaf = 1;
```

```
---------------------------------------------------------------------------------------------
--Дополненное решение

WITH src as( --формирование исходной строки
    SELECT '211221' str
    FROM dual
),
--формируем все числа, которые встречаются в строке, так как нам нужен номер буквы алфавита, то это число либо одно-, либо дву-значное
all_nums AS (
--формируем однозначные числа
    SELECT str, to_number(substr(str, level, 1)) num,
        level AS pos,
        length(str) AS len
    FROM src
    CONNECT BY level<=length(str)
    UNION
--формируем двузначные числа
    SELECT str, to_number(substr(str, level, 2)),
        level,
        length(str) AS len
    FROM src
    WHERE to_number(substr(str, level, 2)) >= 10 --01, 02 и т.д. - не двузначные числа, отсекаем их этим условием
    CONNECT BY level<=length(str)
),
--фильтруем номера, которые не соответствуют позиции буквы в алфавите и заодно добавляем позицию начала и конца номера в первоначальной строке
all_nums_filter AS (
  SELECT num, pos,
    (CASE WHEN length(to_char(num))=1
    THEN pos ELSE pos+1 END) pos_end
  FROM all_nums
  WHERE num BETWEEN 1 AND 33
),
--собираем полученные числа в последовательность букв по правилу: конец предыдущего слова равен началу следующего
seq AS (
    SELECT pos, pos_end, sys_connect_by_path(
        substr('абвгдеёжзийклмнопрстуфхцчшщъыьэюя', num, 1),
        ' ') path
    FROM all_nums_filter
    START WITH pos=1
    CONNECT BY prior pos_end=pos-1
),
--вывод всех возможных наборов слов
words AS (
    SELECT replace(path, ' ', '') res, ROWNUM rn
    FROM seq, src
    WHERE pos_end = length(to_char(str))
)
--собираем все значения в одну строку и перечисляем через запятую
```

```
SELECT ltrim(sys_connect_by_path(res, ', '), ', ') "Ответ"
   FROM words
   WHERE level = (SELECT COUNT(*) FROM words)
   CONNECT BY prior rn = rn - 1;
```

1. Создать запрос для определения среди таблиц Вашей схемы таких таблиц, названия которых получаются друг из друга циклическим сдвигом символов.

Пример результата:

Номер	Таблица 1	Таблица 2
1	BAA	ABA
2	AAB	ABA
3	ABA	BAA
4	BAA	AAB
5	ABA	AAB
6	AAB	BAA

```
create table FDF(strd varchar2(1));
create table FFD(strd varchar2(1));
create table DFF(strd varchar2(1));

with tab_names as(
select
table_name
from user_tables)
select
rownum,
f1.table_name as tab1,
f2.table_name as tab2
from tab_names f1 join tab_names f2
on(length(f1.table_name) = length(f2.table_name)
and f1.table_name != f2.table_name
and instr(f1.table_name || f1.table_name, f2.table_name) != 0);
```

1. Имеется таблица D_V с первым столбцом Dat типа DATE (первичный ключ) и вторым столбцом Val типа NUMBER.

Пример таблицы D_V (строки упорядочены по первому столбцу):

DAT	VAL
01-08-08	232
02-08-08	
10-08-08	182
11-08-08	
21-08-08	240
22-08-08	
22-08-08	

Требуется написать запрос для получения на основе таблицы D_V следующей таблицы:

DAT	MAX_VAL
01-08-08	232
02-08-08	232
10-08-08	182
11-08-08	182
21-08-08	240
22-08-08	240
22-08-08	240

Данная результирующая таблица должна быть упорядочена по Dat, но вместо пустых значений, которые присутствовали в столбце VAL отсортированной по DAT исходной таблицы, в столбце MAX_VAL результирующей таблицы, должны присутствовать значения столбца из предыдущей строки.

Задачу решить без использования аналитических функций, рекурсивного With и Model.

```
with D_V as(
select
to_date('01-08-08', 'DD-MM-YY') as dat,
'232' as val
from dual
union all
select
to_date('02-08-08', 'DD-MM-YY') as dat,
' ' as val
from dual
union all
select
to_date('10-08-08', 'DD-MM-YY') as dat,
'182' as val
from dual
union all
select
to_date('11-08-08', 'DD-MM-YY') as dat,
' ' as val
from dual
union all
select
to_date('21-08-08', 'DD-MM-YY') as dat,
```

```
'240' as val
from dual
union all
select
to_date('22-08-08', 'DD-MM-YY') as dat,
' ' as val
from dual
union all
select
to_date('22-08-08', 'DD-MM-YY') as dat,
' ' as val
from dual
)
select
dv_tab.dat,
case when val = ' ' then
   (select
   dv_tab2.val
   from d_v dv_tab2
   where dv_tab2.dat = (
            select
            max(dat)
            from d_v
            where dat < dv_tab.dat
            and val is not null
            )
   )
else val end as val
from d_v dv_tab
order by 1;
```

1. Для произвольной строки, состоящей из открывающих и закрывающих скобок написать запрос для вывода всех подстрок максимальной длины, представляющих правильные скобочные записи. Например, для строки (()))()() ответ должен быть:

Исходная строка	**Результат**
(()))()()	()()
	(())

```
with imp as( --ввод строки
select
'(()))(()))()))(()()' as str
from dual),

symbol as( --разбиение по символам
select
str,
level lvl,
substr(str, level, 1) sym
from imp
connect by level <= length(str)),

combs as(   --всевозможные комбинации
select distinct
str,
replace(sys_connect_by_path(sym, ','),',','') as substr
from symbol
connect by prior str = str and prior lvl < lvl),

findcombs (str,substr, s, num, cnt) as( --анализ подстрок в рекурсивной with. если (, то cnt+1. если ),то cnt-1
select
str,
substr,
'',
0,
0
```

```
from combs
where substr like '(%'
and substr like '%)'
and regexp_count(substr, '\(') = regexp_count(substr, '\)')
and instr(str, substr) > 0 --спросить насчет вхождения подстроки
union all
select
str,
substr,
substr(substr, num + 1, 1),
num + 1,
case when substr(substr, num + 1, 1) = '(' then cnt + 1 else cnt - 1 end
from findcombs
where num + 1 <= length(substr)
and cnt >= 0),

findcombs1 as( --каждая пара скобок должна давать 0. при этом в каждой паре ) должна быть после (
select * from findcombs
where num = length(substr) and cnt = 0)

select --оставим подстроки с максимальной длинной. вывод результатов
case when rownum = 1 then str else ' ' end as "Исходная строка",
substr as "Результат"
from findcombs1
where
length(substr) = (select max(length(substr)) from findcombs1);
```

2. Используя словарь данных, получить информацию о первичных ключах, внешних ключах, ограничениях уникальности и подчиненных таблицах всех таблиц схемы.

   Имена столбцов в составных первичных ключах, составных ограничениях уникальности и составных внешних ключах вывести в

порядке определенном описаниями таблиц.

Если таблица не имеет подчиненных таблиц, вывести – Подчиненных таблиц нет.

Если таблица не имеет первичного ключа, вывести – Первичного ключа не имеет.

Если таблица не имеет внешних ключей, вывести – Внешних ключей нет. Аналогично, если таблица не имеет ограничения уникальности, вывести – Ограничения уникальности нет.

Имена столбцов композитных ограничений уникальности и внешних ключей заключить в круглые скобки.

Задачу решить без использования аналитических функций, Listagg и Wm_concat.

Пример представления результата:

Имя таблицы	Столбцы первичного ключа	Столбцы с огр уникальности	Столбцы внешн ключей	Подчиненные таблицы
COUNTRIES	COUNTRY_ID	Ограничения уникальности нет	REGION_ID	LOCATIONS
DEPARTMENTS	DEPARTMENT_ID	Ограничения уникальности нет	MANAGER_ID,LOCATION_ID	EMPLOYEES, JOB_HISTORY
EMPLOYEES	EMPLOYEE_ID	EMAIL, (LAST_NAME,FIRST_NAME)	DEPARTMENT_ID, MANAGER_ID, JOB_ID	DEPARTMENTS, EMPLOYEES, JOB_HIST
JOB_HISTORY	EMPLOYEE_ID, START_DATE	Ограничения уникальности нет	EMPLOYEE_ID, DEPARTMENT_ID, JOB_ID	Подчиненных таблиц нет
JOBS	JOB_ID	Ограничения уникальности нет	Внешних ключей нет	EMPLOYEES, JOB_HISTORY
LISE	ID1, ID3, ID2	Ограничения уникальности нет	Внешних ключей нет	LISE_C
LISE_C	Первичного ключа нет	Ограничения уникальности нет	(ID1, ID2, ID3)	Подчиненных таблиц нет
LOCATIONS	LOCATION_ID	Ограничения уникальности нет	COUNTRY_ID	DEPARTMENTS

```
with f1 as( --обединение столбцов с одним именем ограничения в одну строку, start with position=1 потому что столбец position not null именно для ограничений ('P', 'R', 'U')

select

constraint_name,

ltrim(sys_connect_by_path(column_name, ','),',') as comp_columns

from user_cons_columns t1

where level = (select count(*) from user_cons_columns t2 where t1.constraint_name = t2.constraint_name)

start with position = 1

connect by prior constraint_name = constraint_name and prior position < position),


f2 as( --добавление скобок для составных (кроме P)

select

rownum r,

constraint_name,

case when instr(comp_columns, ',') > 0 and not exists (select 'X' from user_constraints t2 where t1.constraint_name = t2.constraint_name and constraint_type = 'P')
```

```
        then '('|| comp_columns || ')' else comp_columns end as cols
from f1 t1),

f3 as ( --Соединеине столбцов с их типами и табл., к которой относится
select
table_name,
constraint_type,
replace(ltrim(sys_connect_by_path(cols, '/'), '/', ',') cols
from f2 natural join user_constraints
where constraint_type in ('P','U','R')
connect by prior table_name = table_name and prior constraint_type = constraint_type and prior r < r),

f4 as (
select *
from f3 t1
where length(cols) = (select max(length(cols)) from f3 t2 where t1.constraint_type = t2.constraint_type and t1.table_name = t2.table_name)),

f5 as ( --соединение подчинённых таблиц с гланой
select
rownum r,
t1.table_name,
t2.table_name r_table_name
from user_constraints t1 join user_constraints t2
on(t2.constraint_type = 'R' and t2.r_constraint_name = t1.constraint_name)),

f6 as (
select
table_name,
ltrim(sys_connect_by_path(r_table_name, ','), ',') r_tables
from f5
connect by prior table_name = table_name and prior r < r),

f7 as(
select *
from f6 t1
```

```
where length(r_tables) = (select max(length(r_tables)) from f6 t2 where t1.table_name = t2.table_name))


select --вывод
t1.table_name "Имя таблицы",
nvl(t2.cols, 'Первичного ключа нет') "Столбцы первичного ключа",
nvl(t3.cols, 'Ограничения уникальности нет') "Столбцы с огр уникальности",
nvl(t4.cols, 'Внешних ключей нет') "Столбцы внешн ключей",
nvl(t5.r_tables, 'Подчиненных таблиц нет') "Подчиненные таблицы"
from user_tables t1 left outer join f4 t2
on (t1.table_name = t2.table_name and t2.constraint_type = 'P')
left outer join f4 t3
on (t1.table_name = t3.table_name and t3.constraint_type = 'U')
left outer join f4 t4
on (t1.table_name = t4.table_name and t4.constraint_type = 'R')
left outer join f7 t5
on (t1.table_name = t5.table_name);
```

1. Используя словарь данных, получить информацию о подчиненности таблиц в схеме в виде:

ИмяТаблицы1(ИмяFK1(Список столбцов) *ссылается на* ИмяТаблицы2/ИмяКлюча2(Список столбцов))

ИмяТаблицы2(ИмяFK2(Список столбцов) *ссылается на* ИмяТаблицы3/ИмяКлюча3(Список столбцов))

……

Пример представления результата:

	RES
1	COUNTRIES (COUNTR_REG_FK (REGION_ID) ссылается на REGIONS / REG_ID_PK (REGION_ID))
2	DEPARTMENTS (DEPT_LOC_FK (LOCATION_ID) ссылается на LOCATIONS / LOC_ID_PK (LOCATION_ID))
3	LOCATIONS (LOC_C_ID_FK (COUNTRY_ID) ссылается на COUNTRIES / COUNTRY_C_ID_PK (COUNTRY_ID))

```
select * from user_constraints; -- constraint_name, constraint_type, table_name, r_constraint_name


select * from user_cons_columns; -- constraint_name, table_name, column_name


with f1 as(
select
uc.table_name,
uc.constraint_name,
ucc.column_name as fk_col,
uc.r_constraint_name,
ucc2.table_name as pk_table,
```

```
ucc2.column_name pk_col
from user_constraints uc join user_cons_columns ucc
on(uc.constraint_name = ucc.constraint_name and uc.constraint_type = 'R')
join user_cons_columns ucc2
on(uc.r_constraint_name = ucc2.constraint_name)),
f2 as (
select
table_name,
constraint_name,
listagg(fk_col, ',') within group (order by fk_col) over (partition by table_name, constraint_name) as fk_col,
r_constraint_name,
pk_col,
pk_table
from f1
order by 1)
select
table_name || ' (' || constraint_name || ' (' || fk_col || ') ссылается на ' || pk_table || ' / ' || r_constraint_name || ' (' || pk_col || '))' as rez
from f2;
```

1. Создать запрос, который выведет из символьного столбца таблицы всю информацию за исключением значений, которые могут быть получены из другого значения столбца за счет циклической перестановки символов.

   Например, для столбца со значениями:

   acghjk
   rtrtr
   ghjkac
   agchjk

   результат должен быть:

   acghjk
   rtrtr
   agchjk

Для начала создаем таблицу:

```
CREATE table tabwords
(word VARCHAR2(20));
```

Заполняем ее значениями:

```
INSERT INTO tabwords(word)
VALUES ('acghjk');
INSERT INTO tabwords(word)
VALUES ('rtrtr');
INSERT INTO tabwords(word)
```

```
VALUES ('ghjkac');
INSERT INTO tabwords(word)
VALUES ('agchjk');
WITH
words (word, num) AS
(SELECT word, ROWNUM
FROM tabwords),
cycles_res (text) AS
(SELECT w1.word
FROM words w1, words w2
WHERE instr(w1.word || w1.word, w2.word) != 0
AND w1.num > w2.num AND LENGTH(w1.word) = LENGTH(w2.word))
SELECT word AS "Результат"
FROM tabwords
WHERE word NOT IN (SELECT text
FROM cycles_res);
```

4. Одной командой SELECT вывести список сотрудников компании, имеющих наименьший оклад среди сотрудников подразделения, в котором они работают.

Сведения о сотрудниках, для которых неизвестно подразделение компании, к которому они приписаны, выводить не нужно.

В результат вывести:

1.Идентификатор подразделения компании, к которому приписан сотрудник.

2. Фамилию сотрудника.

3.Оклад, установленный сотруднику.

В команде SELECT запрещается использовать:

- Фразы WITH, GROUP BY,HAVING, ORDER BY, CONNECT BY, START WITH,
- Условия IN, =ANY, =SOME, NOT IN, <> ALL, EXISTS, NOT EXISTS,
- Подзапросы (subqueries), в том числе подзапросы во фразе FROM,
- Иерархические запросы (hierarchical queries),

- Агрегатные функции (aggregate functions) – MIN,MAX, SUM,COUNT,AVG и др.
- Аналитические функции (analytic functions)

Пример результата.

Department_id	Last_name	Salary
10	Whalen	4400
20	Fay	6000
30	Himuro	2600

```
select-- Все сотрудники

department_id,
last_name,
salary
from employees
where department_id is not null and salary is not null -- За исключением тех, которые не приписаны ни к одному отделу
minus-- Вычитаем из всех сотрудиков тех, которые имеют зарплату большую, чем кто-то из коллег
select
e.department_id,
e.last_name,
e.salary
from employees e inner join employees d -- Делаем inner JOIN зарплат сотрудников внутри отделов
on (e.department_id = d.department_id and e.salary > d.salary); -- Делаем CROSS JOIN зарплат сотрудников внутри отделов
```

3. Создать запрос для получения информации об успеваемости студентов в виде:

ФИО	Дисциплина	Оценка	Дата	Примечания
Петров	Математика	5	20.1.2008	
	Физика	4	22.1.2008	
	Химия	2	25.1.2008	
	Химия	3	27.1.2008	Пересдача
Усов	Математика	5	12.06.99	
	Экономика	3	15.06.99	
	Менеджмент	2	17.06.99	
	Менеджмент	4	18.06.99	Пересдача
Судаков	Экзамены не сдавал			

В таблице должна быть представлена информация только по результатам сдачи экзаменов по дисциплинам, предусмотренным учебным планом для специальности, на которой учится студент.

Исходные данные в таблицах Студенты, Успеваемость, Группы, Учебный план, Дисциплины.

```
  /*Получение необходимого вида организации данных*/
SELECT DECODE(дата,минимальная_дата,фамилия, ' ') AS "Фамилия", nvl(название,
'Экзамены не сдавал')  AS "Дисциплина", nvl(to_char(оценка),' ') AS "Оценка",
nvl(to_char(дата),' ') AS "Дата",
/*Если количество пересдач не равно 1 и значение даты в текущей строке не равно дате
первой попытки, следовательно эта сдача экзамена - пересдача.*/
DECODE(количество,1,' ',DECODE(дата,дата_первой_попытки,' ', 'Пересдача')) AS
"Примечание"
FROM (
/*Нахождение списка студентов, их фамилий, дисциплины, даты экзамена.
Также определяется дата первого экзамена у студента, количество попыток сдачи
экзамена и дата первой попытки.*/
   SELECT
   фамилия,
   номер_студента,
   название,
   оценка,
   дата,
   MIN(дата) OVER (PARTITION BY номер_студента) AS минимальная_дата,
   MIN(дата) OVER (PARTITION BY номер_студента,номер_дисциплины) AS
дата_первой_попытки,
   COUNT(дата) OVER (PARTITION BY номер_студента,номер_дисциплины) AS
количество
   FROM успеваемость right JOIN студенты USING (номер_студента)
   left JOIN дисциплины USING (номер_дисциплины)
)

         ORDER BY номер_студента, дата;
```

БИЛЕТ 1

1. Создать запрос для вывода списка всех столбцов представления ALL_TABLES. Имена столбцов должны быть отсортированы по алфавиту и разделяться запятыми. Список должен быть разбит на строки, при этом каждая строка должна содержать не более 50 символов. Имя каждого столбца должно размещаться целиком на одной строке. Если после имени столбца следует запятая, то она должна находиться на строке вместе с именем столбца.

Результат запроса должен также содержать номер строк и количество столбцов в строке.

Пример представления результата:

Имя представления	**Номер строки**	**Список столбцов**	**Количество столбцов**
ALL_TABLES	1	ACTIVITY_TRACKING, AVG_ROW_LEN, AVG_SPACE,	3
	2	AVG_SPACE_FREELIST_BLOCKS, BACKED_UP, BLOCKS,	3
	3	……	

Решение 1 (Бровченко):

```
WITH
name_view AS
    (SELECT LISTAGG(column_name, ',') WITHIN GROUP (ORDER BY column_name ASC) f_str
    FROM all_tab_columns
    WHERE LOWER(table_name) = 'all_tables'),
term_tab(f_str, cols_list, str_length) AS
   (SELECT f_str,
    CASE
      WHEN REGEXP_COUNT(f_str, ',') = 0 THEN f_str
      ELSE REGEXP_SUBSTR(SUBSTR(f_str, 1, 50), '.*,')
    END cols_list,
    LENGTH(REGEXP_SUBSTR(SUBSTR(f_str, 1,50), '.*,'))+1 str_length
    FROM name_view
```

```
    UNION ALL
    SELECT f_str,
    REGEXP_SUBSTR(SUBSTR(f_str, str_length, 50), '.*,|.*[[:alnum:]]$'),
           str_length + LENGTH(REGEXP_SUBSTR(SUBSTR(f_str, str_length, 50),
'.*,|.*[[:alnum:]]$'))
    FROM term_tab
    WHERE str_length < LENGTH(f_str)),
output AS
   (SELECT ROW_NUMBER() OVER (ORDER BY cols_list) row_num,
        cols_list
    FROM term_tab)
SELECT
CASE
   WHEN row_num = 1 THEN 'ALL_TABLES'
   ELSE ' '
END "Имя представления",
row_num "Номер строки",
cols_list "Список столбцов",
CASE
   WHEN REGEXP_COUNT(cols_list, ',') = 0 THEN 1
   ELSE REGEXP_COUNT(cols_list, ',')
   END "Количество столбцов"
FROM output;
```

	Имя представления	Номер строки	Список столбцов	Количество столбцов
1	ALL_TABLES	1	ACTIVITY_TRACKING,AVG_ROW_LEN,AVG_SPACE,	3
2		2	AVG_SPACE_FREELIST_BLOCKS,BACKED_UP,BLOCKS,	3
3		3	BUFFER_POOL,CACHE,CELL_FLASH_CACHE,CHAIN_CNT,	4
4		4	CLUSTERING,CLUSTER_NAME,CLUSTER_OWNER,COMPRESSION,	4
5		5	COMPRESS_FOR,CONTAINER_DATA,DEGREE,DEPENDENCIES,	4
6		6	DML_TIMESTAMP,DROPPED,DURATION,EMPTY_BLOCKS,	4
7		7	FLASH_CACHE,FREELISTS,FREELIST_GROUPS,	3
8		8	GLOBAL_STATS,HAS_IDENTITY,INITIAL_EXTENT,	3
9		9	INI_TRANS,INSTANCES,IOT_NAME,IOT_TYPE,	4
10		10	LAST_ANALYZED,LOGGING,MAX_EXTENTS,MAX_TRANS,	4
11		11	MIN_EXTENTS,MONITORING,NESTED,NEXT_EXTENT,	4
12		12	NUM_FREELIST_BLOCKS,NUM_ROWS,OWNER,PARTITIONED,	4
13		13	PCT_FREE,PCT_INCREASE,PCT_USED,READ_ONLY,	4
14		14	RESULT_CACHE,ROW_MOVEMENT,SAMPLE_SIZE,SECONDARY,	4
15		15	SEGMENT_CREATED,SKIP_CORRUPT,STATUS,	3
16		16	TABLESPACE_NAME,TABLE_LOCK,TABLE_NAME,TEMPORARY,	4
17		17	USER_STATS	1

2. Создать таблицу Customers, содержащую 2 столбца: Id number(15,0) Primary Key

и Last_Name varchar2 (40). Создать запрос, который будет выводить значение столбца Id и Номер группы последовательных целых значений с шагом 1. Например, для таблицы, содержащей значения:

Id	Last_Name
1	Mougus
2	Green
3	Grase
7	Scott
8	Trumen
10	Kochhar
12	Drejk
13	Kook

результат должен быть:

Id	Номер группы
1	1
2	1
3	1
7	2
8	2
10	3
12	4
13	4

Задачу решить без использования аналитических функций и раздела Model.

Решение 1 (Бровченко):

Создаем таблицу Customers и заполняем информацию:

CREATE TABLE customers ("Id" number(15,0) Primary Key, "Last_Name" varchar2(40));

INSERT INTO customers VALUES (1, 'Mougus');

INSERT INTO customers VALUES (2, 'Green');

INSERT INTO customers VALUES (3, 'Grase');

INSERT INTO customers VALUES (7, 'Scott');

INSERT INTO customers VALUES (8, 'Trumen');

INSERT INTO customers VALUES (10, 'Kochhar');

INSERT INTO customers VALUES (12, 'Dreik');

INSERT INTO customers VALUES (13, 'Kook');

	Id	Last_Name
1	1	Mougus
2	2	Green
3	3	Grase
4	7	Scott
5	8	Trumen
6	10	Kochhar
7	12	Dreik
8	13	Kook

```
SELECT "Id",
  (SELECT MAX(gr_num)
          FROM (SELECT "Id",
          ROWNUM gr_num
          FROM customers
         WHERE "Id" - 1 NOT IN (SELECT "Id" FROM Customers)
          WHERE "Id" <= customers."Id") "Номер группы"
FROM customers;
```

Результат:

	Id	Номер группы
1	1	1
2	2	1
3	3	1
4	7	2
5	8	2
6	10	3
7	12	4
8	13	4

Поменяем значения в таблице:

```
UPDATE customers
SET "Id" = 4
WHERE "Id" = 7;
UPDATE customers
SET "Id" = 5
```

```
WHERE "Id" = 8;
UPDATE customers
SET "Id" = 20
WHERE "Id" = 13;
```

	Id	Last_Name
1	1	Mougus
2	2	Green
3	3	Grase
4	4	Scott
5	5	Trumen
6	10	Kochhar
7	12	Dreik
8	20	Kook

Результат:

	Id	Номер группы
1	1	1
2	2	1
3	3	1
4	4	1
5	5	1
6	10	2
7	12	3
8	20	4

Удалим несколько строк из таблицы:

```
DELETE FROM customers
WHERE "Id" = 2;
DELETE FROM customers
WHERE "Id" = 4;
```

	Id	Last_Name
1	1	Mougus
2	3	Grase
3	5	Trumen
4	10	Kochhar
5	12	Dreik
6	20	Kook

Результат:

	Id	Номер группы
1	1	1
2	3	2
3	5	3
4	10	4
5	12	5
6	20	6

3. Одной командой вывести все палиндромы, встречающиеся в произвольной символьной строке.

Например, для строки

aabacdca

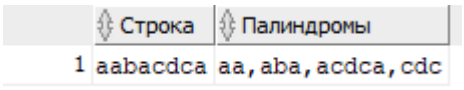
ответ должен быть:

aa,aba, cdc, acdca

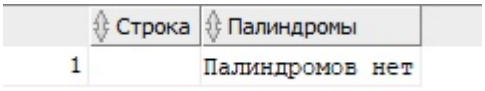
Решение 1:

```
DEFINE str = "aabacdca";
DEFINE str = "aA        bacdca          ";

WITH
    trimw AS (SELECT TRIM(both ' ' from replace(replace(replace('&STR', ' ', ' _'), '_ '), '_')) AS tstr
            FROM dual),
    source AS (SELECT substr(tstr,level,1) AS lt, ROW_NUMBER() OVER(ORDER BY level ASC) rn
            FROM trimw
            CONNECT BY level <= length(tstr)),
    tmp AS (SELECT replace(sys_connect_by_path(lt,','),',') word, level l, ROWNUM rn
        FROM source
        WHERE level > 1
        CONNECT BY PRIOR rn + 1 = rn),
                result AS (SELECT DISTINCT word w1, (SELECT LISTAGG(regexp_substr(word,'.',level,1), '') WITHIN GROUP(ORDER BY ROWNUM DESC)
                            FROM dual
```

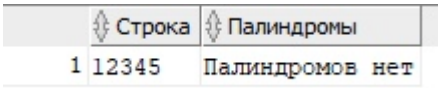
```
                    CONNECT BY level <= length(word)) AS w2
        FROM tmp)

SELECT '&STR' AS "Строка", nvl(LISTAGG(TRIM(w1), ',') WITHIN GROUP(ORDER BY w1), 'Палиндромов нет') AS "Палиндромы"
FROM result
WHERE LOWER(w1) = LOWER(w2);
--если учитываем регистр, то WHERE w1 = w2;

UNDEFINE str;
```



	Строка	Палиндромы
1	aabacdca	aa,aba,acdca,cdc



	Строка	Палиндромы
1		Палиндромов нет



	Строка	Палиндромы
1	12345	Палиндромов нет

4. Определить цифры, которые максимальное количество раз встречаются в столбце Phone_number таблицы Employees.

Пример результата:

MAX_CNT	NUM
212	1
212	4

Решение 1:

```
WITH nums AS (SELECT regexp_substr('0123456789', LEVEL, 1) AS num
        FROM dual
        CONNECT BY LEVEL <= 10),
    tabl AS (SELECT (SELECT SUM(regexp_count(phone_number, num))
            FROM employees) max_cnt, num
        FROM nums)
SELECT max_cnt, num
FROM tabl
WHERE max_cnt = (SELECT MAX(max_cnt)
            FROM tabl);
```

	MAX_CNT	NUM
1	212	1
2	212	4

5. Задана произвольная символьная строка, состоящая из двух частей, разделенных символами «=>». В левой и правой части выражения содержатся символьные строки, разделенные запятыми.

Требуется создать запрос, который будет выводить все возможные пары комбинаций из левой и правой частей.

Пример результата для строки a, fgf,yy=>uu,gh:

PATH
a=>uu
a=>gh
fgf=>uu
fgf=>gh
yy=>uu
yy=>gh

Решение 1 (ОЮ):

```
DEFINE &&STR;
SELECT A|| '=>' || B "PATH"
FROM(SELECT TRIM(regexp_substr(SUBSTR('&STR', 1, INSTR('&STR', '=>')-1), '[^,]+', 1, level)) A
    FROM DUAL
    CONNECT BY regexp_substr(SUBSTR('&STR', 1, INSTR('&STR', '=>')-1), '[^,]+', 1, level) IS NOT NULL)
CROSS JOIN (SELECT TRIM(regexp_substr(SUBSTR('&STR', INSTR('&STR',
```

'=>')+2), '[^,]+', 1, level)) B

FROM DUAL

CONNECT BY regexp_substr(SUBSTR('&STR', INSTR('&STR', '=>')+2), '[^,]+', 1, level) IS NOT NULL);

UNDEFINE STR;

	PATH
1	a=>uu
2	a=>gh
3	fgf=>uu
4	fgf=>gh
5	yy=>uu
6	yy=>gh

БИЛЕТ 2

1. Для каждого отдела вывести фамилии и зарплаты трех сотрудников, получающих самые высокие зарплаты в отделе. Если самую низкую зарплату у найденных трех сотрудников отдела получают и какие-то другие сотрудники этого отдела, они тоже должны попасть в список. *Для отделов, в которых меньше трех сотрудников, информацию не выводить.*

```
SELECT department_id, last_name, salary FROM
(SELECT department_id, last_name, salary,
    COUNT(employee_id) OVER (PARTITION BY department_id) emp_count,
    DENSE_RANK() OVER (PARTITION BY department_id ORDER BY salary DESC) sal_rank
  FROM employees
  WHERE department_id IS NOT NULL)
WHERE sal_rank <= 3 AND emp_count >= 3;
```

Результат:

	DEPARTMENT_ID	LAST_NAME	SALARY
1	30	Raphaely	11000
2	30	Khoo	3100
3	30	Baida	2900
4	50	Fripp	8200
5	50	Weiss	8000
6	50	Kaufling	7900
7	60	Hunold	9000
8	60	Ernst	6000
9	60	Austin	4800
10	60	Pataballa	4800
11	80	Russell	14000
12	80	Partners	13500
13	80	Errazuriz	12000
14	90	King	24000
15	90	Kochhar	17000
16	90	De Haan	17000
17	100	Greenberg	12008
18	100	Faviet	9000
19	100	Chen	8200

Переведём сотрудника De Haan в отдел 100, в 90 отделе количество работников получается меньше 3 человек:

```
UPDATE employees
SET department_id = 100
WHERE last_name = 'De Haan';
```

Результат:

	DEPARTMENT_ID	LAST_NAME	SALARY
1	30	Raphaely	11000
2	30	Khoo	3100
3	30	Baida	2900
4	50	Fripp	8200
5	50	Weiss	8000
6	50	Kaufling	7900
7	60	Hunold	9000
8	60	Ernst	6000
9	60	Austin	4800
10	60	Pataballa	4800
11	80	Russell	14000
12	80	Partners	13500
13	80	Errazuriz	12000
14	100	De Haan	17000
15	100	Greenberg	12008
16	100	Faviet	9000

90 отдел теперь не выводится.

Поменяем зарплату сотрудникам из 30 отдела:

```
UPDATE employees
SET salary = 10000
WHERE last_name = 'Raphaely';
UPDATE employees
SET salary = 10000
WHERE last_name = 'Khoo';
UPDATE employees
SET salary = 10000
WHERE last_name = 'Baida';
```

Результат:

	DEPARTMENT_ID	LAST_NAME	SALARY
1	30	Raphaely	10000
2	30	Khoo	10000
3	30	Baida	10000
4	30	Tobias	2800
5	30	Himuro	2600
6	50	Fripp	8200
7	50	Weiss	8000
8	50	Kaufling	7900
9	60	Hunold	9000
10	60	Ernst	6000
11	60	Austin	4800
12	60	Pataballa	4800
13	80	Russell	14000
14	80	Partners	13500
15	80	Errazuriz	12000
16	100	De Haan	17000
17	100	Greenberg	12008
18	100	Faviet	9000

2. Создать запрос для разделения "задвоенных" данных. Например, из

CODE_OPERATION	ID_CLIENT
1000 1100	841000 841100
2000	6700 8967 5500

сделать

RN	CNT	CODE_OPERATION	ID_CLIENT
1	0	1000 1100	841000 841100
	1	1000	841000
	2	1100	841100
2	0	2000	6700 8967 5500
	1	2000	6700
	2		8967
	3		5500

Задачу решить с использованием раздела Model.

БЕЗ РАЗДЕЛА MODEL

```
WITH
src AS (
   SELECT '1000 1100 900' AS code_operation, '841000 841100 841111' AS id_client
   FROM dual
   UNION ALL
   SELECT '700 500 400' AS code_operation, '923400 923411' AS id_client
   FROM dual
   UNION ALL
   SELECT '700 500 400 456' AS code_operation, '923400 923411 456 87 85' AS id_client
```

```
   FROM dual
   UNION ALL
   SELECT '700 500 400 456' AS code_operation, '923400 923411 456 87 8' AS id_client
   FROM dual),
search_reg (rn, cnt, part_act, part_code, code_operation, id_client) AS (
    SELECT TO_CHAR(ROWNUM) rn, 1 cnt, '' part_act, '' part_code, code_operation, id_client
   FROM src
   UNION ALL
        SELECT rn, cnt + 1, regexp_substr(code_operation,'[[:digit:]]+',1,cnt), regexp_substr(id_client,'[[:digit:]]+',1,cnt), code_operation, id_client
   FROM search_reg
        WHERE regexp_instr(code_operation,'[[:digit:]]+',1,cnt) <> 0 OR regexp_instr(id_client,'[[:digit:]]+',1,cnt) <> 0),
fin AS (
   SELECT rn, cnt, part_act, part_code, code_operation, id_client
   FROM search_reg
   ORDER BY rn, cnt)
SELECT (CASE WHEN part_act IS NULL AND part_code IS NULL THEN rn
     ELSE ' ' END) AS rn, cnt - 1 AS cnt, nvl((CASE WHEN part_act IS NULL AND part_code IS NULL THEN code_operation
                                                ELSE part_act END),' ') AS code_operation,
    nvl((CASE WHEN part_act IS NULL AND part_code IS NULL THEN id_client
       ELSE part_code END),' ') AS id_client
FROM fin;
```

3. Используя словарь данных, получить информацию об ограничениях CHECK схемы:

   В списках имена столбцов вывести через запятую. Имя таблицы не должно повторяться.

   Задачу решить без использования функций Listagg и Wm_concat.

   Пример представления результатов:

SQL | All Rows Fetched: 8 in 1,748 seconds

	Имя таблицы	Имя ограничения	Столбцы,входящие в ограничение	Ограничение CHECK
1	EMPLOYEES	EMP_SALARY_MIN	SALARY	salary > 0
2	JOB_HISTORY	JHIST_DATE_INTERVAL	END_DATE,START_DATE	end_date > start_date
3	TASK_13	COLUMN12345	COLUMN_1,COLUMN_2,COLUMN_3,COLUMN_4,COLUMN_5	column_1>ALL (column_2, column_3, column_4, column_5)
4	TRIP	CON_PER_BEG	PER_BEG	per_beg=TRUNC(per_beg)
5		CON_PER_END	PER_END	per_end=TRUNC(per_end)
6	ПРЕПАРАТ	ПРЕПАРАТ_CHK1	АББРЕВ_ФОРМЫ,ТИП	Аббрев_формы!=Тип
7	ПРЕПОДАВАТЕЛЬ	SYS_C00201812	КАФЕДРА	КАФЕДРА Like 'Кафедра%'
8	УСПЕВАЕМОСТЬ	SYS_C00201820	ОЦЕНКА	ОЦЕНКА between 1 and 5

Решение 1:

```
WITH
t AS (
    SELECT uc.table_name table_name, uc.constraint_name constraint_name,
  ucc.column_name cn, uc.search_condition sc,
    ROW_NUMBER() OVER(PARTITION BY uc.table_name, uc.constraint_name
  ORDER BY ucc.column_name) rn,
      ROW_NUMBER() OVER(PARTITION BY uc.table_name ORDER BY
  ucc.column_name) rn_t
    FROM user_constraints uc LEFT JOIN user_cons_columns ucc ON
  uc.constraint_name = ucc.constraint_name
  WHERE uc.constraint_type = 'C'),
result AS (
  SELECT table_name, constraint_name, ltrim(sys_connect_by_path(cn,','),',') cn, sc,
  rn, rn_t
  FROM t
  WHERE CONNECT_BY_ISLEAF = 1
  START WITH rn = 1
  CONNECT BY PRIOR table_name = table_name
      AND PRIOR constraint_name = constraint_name
      AND PRIOR rn + 1 = rn
  ORDER BY table_name, rn_t)
SELECT CASE WHEN rn_t = 1 OR rn != 1 THEN table_name
    ELSE ' ' END AS "Имя таблицы", constraint_name AS "Имя ограничения", cn
  AS "Столбцы, входящие в огр-е", sc "Ограничение CHECK"
FROM result;
```

4. Предполагая, что не существует зарплаты сотрудников (таблица Employees), большей 100000, для каждого сотрудника, имеющего более двух подчиненных, вывести представление зарплаты в десятичной и двоичной системах счисления (без ведущих нулей).

```
WITH Bin(num, salary, gr1, gr2, "list") AS(
SELECT r1.Employee_ID num, r1.Salary, r1.Salary gr1, floor(r1.Salary/2) gr2,
  TO_CHAR(mod(r1.Salary, 2)) "list"
FROM Employees r1
WHERE r1.Salary <= 100000
UNION ALL
SELECT num, Salary, gr2, floor(gr2/2),
mod(gr2, 2) || "list"
```

```
FROM Bin
WHERE gr1 >0
),
BinResult AS(
SELECT num Employee_ID, SALARY, SUBSTR("list", INSTR("list", '1', 1, 1)) AS
    "Binary salary"
FROM Bin
WHERE gr1=0
) SELECT b.Employee_ID, b.SALARY, "Binary salary"
FROM BinResult b
INNER JOIN Employees L ON (L.manager_id = b.Employee_ID)
GROUP BY b.Employee_ID, b.SALARY, "Binary salary"
HAVING count(L.employee_ID) > 2;
```

Алгоритм:

```
1) Сперва построим рекурсивный запрос Bin, выводящий порядок перевода
зарплаты каждого сотрудника в бинарную систему. Gr1=0 обозначает, что перевод
закончен, и строка list – итоговый результат, который нам и нужен
(Визуализируем этот промежуточный результат с помощью запроса SELECT * FROM
Bin)
2) Отберем эти строки в запросе BinResult. Обрежем результат, убрав ведущие
нули.
3)Наконец, в главном запросе сделаем INNER JOIN с Employees, получая тех
сотрудников, чьими менеджерами являются наши сотрудники из BinResult. Выберем
тех менеджеров, у которых в подчинении более двух сотрудников.
```

5. Используя обращение только к таблице DUAL, построить SQL-запрос, возвращающий один столбец, содержащий календарь на заданный месяц заданного года:
   - номер дня в месяце (две цифры),
   - полное название месяца по-английски заглавными буквами (в верхнем регистре),
   - год (четыре цифры),
   - полное название дня недели по-английски строчными буквами (в нижнем регистре).

   Каждое "подполе" должно быть отделено от следующего одним пробелом. В результате не должно быть начальных и хвостовых пробелов. Количество возвращаемых строк должно точно соответствовать количеству дней в текущем месяце. Строки должны быть упорядочены по номерам дней в месяце по возрастанию.

   Пример вывода результата:
   1 MAY 2020 friday
   2 MAY 2020 Saturday

   Решение 1:

```
WITH
t AS (
```

```
   SELECT TO_DATE('10.12.1998','dd.mm.syyyy') s
   FROM dual),
t1 AS (
   SELECT add_months(last_day(s) + 1,-1) ss, s
   FROM t),
t2 AS (
   SELECT ss + level - 1 sss
   FROM t1
   CONNECT BY ss + level - 1 != last_day(s) + 1)
SELECT TRIM(regexp_replace(TO_CHAR(sss,'fmdd MON syyyy
day','nls_date_language=english'),'  ',' ') ) kalendar
FROM t2;
```

БИЛЕТ 3

1. Для произвольной команды SELECT определить список входящих в нее таблиц (через запятую) с указанием имени схемы. Задачу решить одной командой SELECT.
Например, для команды:

```
WITH "СР ПО ОТД" AS (
SELECT DEPARTMENT_ID,AVG(SALARY) AS ASAL
FROM hr.EMPLOYEES
GROUP BY DEPARTMENT_ID),
"НАИБ БЛИЗ" AS (
SELECT DEPARTMENT_ID, MIN(ABS(SALARY - ASAL)) AS MINSAL
FROM EMPLOYEES JOIN "СР ПО ОТД" USING (DEPARTMENT_ID)
GROUP BY DEPARTMENT_ID)
SELECT EMPLOYEE_ID AS "Номер",LAST_NAME AS "Фамилия",JOB_ID
AS"Должность",
DEPARTMENT_ID AS "Отдел",SALARY AS "Оклад", TRUNC(ASAL) AS "Средний
оклад"
FROM EMPLOYEES JOIN "MY JOBS" USING (JOB_ID)
JOIN "СР ПО ОТД" USING (DEPARTMENT_ID) JOIN "НАИБ БЛИЗ"
USING (DEPARTMENT_ID)
WHERE (DEPARTMENT_ID, ABS(SALARY - ASAL)) IN
(SELECT DEPARTMENT_ID, MINSAL FROM "НАИБ БЛИЗ")
ORDER BY DEPARTMENT_ID, SALARY, LAST_NAME;
```

результат должен быть:
hr.EMPLOYEES,os.EMPLOYEES,os."MY JOBS"

```
--РЕШЕНИЕ С ЭКРАНИРОВАНИЕМ
--вводим исходую строку
WITH
input (str) AS (
SELECT '&&str'
FROM dual),
--все таблицы, которые фигурируют в строке-запросе
tabs AS
(SELECT
regexp_substr(str,'(".+?")|([[:alnum:]_.]*)',
regexp_instr(str,'[[:space:]](FROM|JOIN)[[:space:]]',1,level,1,'i'),1) AS table_name
FROM input
CONNECT BY level <= regexp_count(str,'[[:space:]](FROM|JOIN)[[:space:]]',1,'i')),
--если имя таблицы с кавычками, то оно регистрочуствительное, иначе - оно
--записано в словарях данных в высоком регистре
tabs_fix AS
(SELECT
(CASE
WHEN instr(tabs.table_name,"") = 0
AND tabs.table_name NOT LIKE '%.%' THEN upper(tabs.table_name)
ELSE tabs.table_name
END) table_name
```

```
FROM tabs),
--проверяем, чем является найденный объект и в зависимсоти от этого решаем, выводить его или нет
tables AS
(SELECT DISTINCT
CASE
WHEN tab.table_name LIKE '%.%'
OR upper(tab.table_name) = 'DUAL' THEN tab.table_name
WHEN syn.synonym_name IS NOT NULL THEN lower((SELECT user
FROM dual)) || '.' || syn.table_name
WHEN TRIM(BOTH '"' FROM tab.table_name) IN (SELECT table_name
FROM user_tables
UNION
SELECT table_name
FROM dictionary) THEN lower((SELECT user
FROM dual)) || '.' || tab.table_name
END table_name
FROM tabs_fix tab
LEFT JOIN user_synonyms syn ON tab.table_name = syn.synonym_name
ORDER BY TRIM(BOTH '"' FROM table_name)),
result AS
(SELECT table_name, ROWNUM
FROM tables
WHERE table_name IS NOT NULL)

SELECT LISTAGG(table_name, ',') WITHIN GROUP(ORDER BY ROWNUM) таблицы
FROM result;
UNDEFINE str;

--Решение
--вводим исходую строку
WITH
src AS
(SELECT q'{
WITH "СР ПО ОТД" AS (
SELECT DEPARTMENT_ID,AVG(SALARY) AS ASAL
FROM hr.EMPLOYEES
GROUP BY DEPARTMENT_ID),
"НАИБ БЛИЗ" AS (
SELECT DEPARTMENT_ID, MIN(ABS(SALARY - ASAL)) AS MINSAL
FROM EMPLOYEES JOIN "СР ПО ОТД" USING (DEPARTMENT_ID)
GROUP BY DEPARTMENT_ID)
SELECT EMPLOYEE_ID AS "Номер",LAST_NAME AS "Фамилия",JOB_ID AS "Должность",
DEPARTMENT_ID AS "Отдел",SALARY AS "Оклад", TRUNC(ASAL) AS "Средний оклад"
FROM EMPLOYEES JOIN "MY JOBS" USING (JOB_ID)
JOIN "СР ПО ОТД" USING (DEPARTMENT_ID) JOIN "НАИБ БЛИЗ"
USING (DEPARTMENT_ID)
```

```
WHERE (DEPARTMENT_ID, ABS(SALARY - ASAL)) IN
(SELECT DEPARTMENT_ID, MINSAL FROM "НАИБ БЛИЗ")
ORDER BY DEPARTMENT_ID, SALARY, LAST_NAME;
}' AS str
FROM dual),
--все таблицы, которые фигурируют в строке-запросе
tabs AS
(SELECT
regexp_substr(str,'(".+?")|([[:alnum:]_.]*)',
regexp_instr(str,'[[:space:]](FROM|JOIN)[[:space:]]',1,level,1,'i'),1) AS table_name
FROM src
CONNECT BY level <= regexp_count(str,'[[:space:]](FROM|JOIN)[[:space:]]',1,'i')),
--если имя таблицы с кавычками, то оно регистрочуствительное, иначе - оно
--записано в словарях данных в высоком регистре
tabs_fix AS
(SELECT
(CASE
WHEN instr(tabs.table_name,'"') = 0
AND tabs.table_name NOT LIKE '%.%' THEN upper(tabs.table_name)
ELSE tabs.table_name
END) table_name
FROM tabs),
--проверяем, чем является найденный объект и в зависимсоти от этого решаем, выводить
его или нет
tables AS
(SELECT DISTINCT
CASE
WHEN tab.table_name LIKE '%.%'
OR upper(tab.table_name) = 'DUAL' THEN tab.table_name
WHEN syn.synonym_name IS NOT NULL THEN lower((SELECT user
FROM dual)) || '.' || syn.table_name
WHEN TRIM(BOTH '"' FROM tab.table_name) IN (SELECT table_name
FROM user_tables
UNION
SELECT table_name
FROM dictionary) THEN lower((SELECT user
FROM dual)) || '.' || tab.table_name
END table_name
FROM tabs_fix tab
LEFT JOIN user_synonyms syn ON tab.table_name = syn.synonym_name
ORDER BY TRIM(BOTH '"' FROM table_name)),
result AS
(SELECT table_name, ROWNUM
FROM tables
WHERE table_name IS NOT NULL)

SELECT LISTAGG(table_name, ',') WITHIN GROUP(ORDER BY ROWNUM) таблицы
FROM result;
```

2. **Имеется таблица с колонкой, которая содержит множество значений, разделенных запятыми. Требуется создать запрос, который каждое значение выведет на отдельной строке. Например, дана таблица:**

Номер	Телефон
952240	2-78,2-89
952423	2-78,2-83,8-34

Результат:

Номер	Телефон
952240	2-78
	2-89
952423	2-78
	2-83
	8-34

Задачу решить с использованием раздела Model.

Решение 1:

```
CREATE TABLE phone_numbers (
   p_num     NUMBER(10,0),
   p_phone   VARCHAR2(100 CHAR),
   CONSTRAINT phone_numbers PRIMARY KEY (p_num));


INSERT INTO phone_numbers VALUES (952240, '2-78,2-89');
INSERT INTO phone_numbers VALUES (952423,'2-78,2-83,8-34');


--без модел


SELECT nvl(TO_CHAR(DECODE(lvl,1,p_num) ),' ') AS "Номер", phone_num AS "Телефон"
FROM (SELECT DISTINCT p_num, level lvl, regexp_substr(p_phone,'[^,]+',1,level) AS phone_num
    FROM phone_numbers
    CONNECT BY regexp_count(p_phone,'[^,]+') >= level
    ORDER BY p_num, phone_num);
```

3. **Для каждого отдела из таблицы Departments отобразить в виде одной строки с запятой в качестве разделителя фамилии сотрудников, работающих в нем. Фамилии сотрудников должны быть отсортированы по алфавиту. Задачу решить без использования функций Listagg и wm_concat.**

--1 способ

```
/*Определение номера отдела, фамилии и номера сотрудников, которые работают в
отделе.
Для дальнейшего использование иерархического запроса необходимо найти номер
служащего, который должен следовать в списке за текущим.*/
WITH sel AS (SELECT d.department_id,
last_name,
employee_id,
LAG(employee_id) OVER (PARTITION BY d.department_id ORDER BY last_name) AS lag1
FROM employees e RIGHT JOIN departments d ON e.department_id=d.department_id)

/*Организация списка департаментов и сотрудников, работающих в них.*/
SELECT
   department_id AS "Номер отдела",
   NVL(SUBSTR(SYS_CONNECT_BY_PATH(last_name, ','),2), ' ') AS "Фамилии
сотрудников"
FROM sel
WHERE CONNECT_BY_ISLEAF=1
START WITH lag1 IS NULL
CONNECT BY PRIOR employee_id=lag1
ORDER BY department_id;

--2 способ
WITH
--Выводим номера отделов, фамилии сотрудников, работающих в них, в алфавитном
порядке и их номера в пределах отдела
temp_tab AS (
   SELECT department_id, last_name, ROW_NUMBER() OVER(PARTITION BY
department_id ORDER BY last_name ASC) AS rn
   FROM employees
   WHERE department_id IS NOT NULL
   GROUP BY department_id, last_name)
--Основной запрос: выведем номера отделов и список фамилий сотрудников, работающих
в этих отделах, в строчку через запятую
SELECT tab.department_id,
--Определяем список сотрудников, работающих в заданном отделе, записав их в строчку
через запятую
     nvl((SELECT MAX(ltrim(sys_connect_by_path(last_name,', '),', ') )
         FROM temp_tab
         WHERE department_id = tab.department_id
--Начинаем цепочку фамилий с первой фамилии в заданном отделе
         START WITH department_id = tab.department_id AND rn = 1
         CONNECT BY NOCYCLE PRIOR rn = (rn - 1)
            AND department_id = tab.department_id
            AND level <= (SELECT COUNT(last_name)--Определяем количество
сотрудников в отделе
                     FROM temp_tab
                     WHERE department_id = tab.department_id)
         GROUP BY department_id),' ') AS last_names
```

```
FROM departments tab
ORDER BY tab.department_id ASC;
```

4. **Определить список сотрудников (таблица Employees), у которых в именах и фамилиях содержится, по крайней мере, по три совпадающие буквы.**

Результат представить в виде:

Сотрудник	Результат
Alberto Errazuriz	Совпадают три буквы (a,e,r)
Alexaner Hunold	Совпадают три буквы (d,l,n)
Elizabeth Bates	Совпадают четыре буквы (a,b,e,t)

```
WITH inf_l AS
(SELECT employee_id, LOWER(REPLACE(last_name, ' ', '')) last_name, NULL alph
FROM employees),

recur_l(employee_id, last_name, alph) AS
(SELECT employee_id, last_name, alph
FROM inf_l
UNION ALL
SELECT employee_id, REPLACE(last_name,RPAD(last_name, 1),''), RPAD(last_name, 1)
FROM recur_l
WHERE last_name IS NOT NULL),

las AS
(SELECT r.employee_id, e.last_name, r.alph
FROM recur_l r JOIN employees e ON(r.employee_id = e.employee_id)
WHERE alph IS NOT NULL
ORDER BY employee_id),

inf_f AS
(SELECT employee_id, LOWER(REPLACE(first_name, ' ', '')) first_name, NULL alph
FROM employees),

recur_f(employee_id, first_name, alph) AS
(SELECT employee_id, first_name, alph
FROM inf_f
UNION ALL
SELECT employee_id, REPLACE(first_name,RPAD(first_name, 1),''), RPAD(first_name, 1)
FROM recur_f
WHERE first_name IS NOT NULL),

fir AS
(SELECT r.employee_id, e.first_name, r.alph
```

```
FROM recur_f r JOIN employees e ON(r.employee_id = e.employee_id)
WHERE alph IS NOT NULL
ORDER BY employee_id),

res1 AS
(SELECT l.employee_id, f.first_name, f.alph , l.last_name
FROM las l JOIN fir f ON(l.employee_id = f.employee_id AND f.alph = l.alph)),

res2 AS
(SELECT employee_id , first_name, last_name, LISTAGG(alph, ',') WITHIN GROUP(ORDER
BY alph) alph
FROM res1
GROUP BY employee_id , first_name, last_name),

res3 AS
(SELECT employee_id , first_name||' '||last_name sotr, 'Совпадают
'||DECODE(REGEXP_COUNT(alph, '[^,]+'),3,'три',4,'четыре', 5, 'пять', 6, 'шесть',  7, 'семь',  8,
'восемь', 9, 'девять',
        10, 'десять', 11, 'одиннадцать', 12, 'двенадцать', 13, 'тринадцать', 14,
'четырнадцать', 15, 'пятнадцать', 16, 'шестандцать', 17, 'семнадцать', 18, 'восемнадцать',
        19, 'девятнадцать', 20, 'двадцать', 21, 'двадцать один', 22, 'двадцать два', 23,
'двадцать три', 24, 'двадцать четыре', 25, 'двадцать пять', 26, 'двадцать шесть')||'
буквы('||alph||')' alph
FROM res2
WHERE REGEXP_COUNT(alph, '[^,]+') >=3)

SELECT sotr as "Сотрудник", alph as "Результат"
FROM res3;
```

5.Определить список последовательностей подчиненности от преподавателей, не имеющих начальника, до преподавателей, не имеющих подчиненных.

Если список состоит более, чем из четырех фамилий, то выводить только две первые и две последние фамилии, а вместо остальных фамилий поставить многоточие.

Костыркин-> Викулина-> …->Соколов->Казанко (не имеет подчиненных)

```
WITH temp AS (SELECT substr(teachers, 3, length(teachers)-2) || '(не имеет подчиненных)' str
,lvl
FROM
(
   SELECT SYS_CONNECT_BY_PATH(фамилия,'->') as teachers, connect_by_isleaf as
isleaf,level lvl
   FROM Преподаватели
   START WITH подчиняется IS NULL
   CONNECT BY PRIOR номер_преподавателя = подчиняется
)
WHERE ISLEAF = 1)
SELECT
```

```
   CASE WHEN lvl>4
   THEN REPLACE(str,substr(str,instr(str,'->',1,2)+2,instr(str,'->',1,lvl-2)-instr(str,'->',1,1)-
2),'...')
   ELSE str
   END AS "Список подчиненностей"
FROM temp;
```

БИЛЕТ 4

1. Имеется таблица Продажи (Номер, Название товара, Дата, Скидка %). Вывести отчет по продажам, который включает столбцы Название товара, Даты продажи, Скидка %, представив информацию таким образом, что если один и тот же товар продавался с одной и той же скидкой несколько дней, то эти даты должны выводиться через запятую. При этом если две или более даты отличаются друг от друга на один день, то они должны быть представлены в виде интервала с дефисом в качестве разделителя.

Пример представления результата:

Название товара	**Даты продажи**	**Скидка, %**
Стул	1.02.2016, 5.02.2016, 7.02.2016-12.02.2016, 15.02.2016	5
Стол	2.02.2016, 4.02.2016	10
Кровать	2.02.2016, 6.02.2016 - 7.02.2016, 12.02.2016-15.02.2016	10

Решение 1:

```
CREATE TABLE "Продажи" ("Номер" NUMBER(8) NOT NULL, "Название товара" VARCHAR2(16), "Дата" DATE, "Скидка %" VARCHAR2(16));

INSERT INTO "Продажи" VALUES (1, 'Стул', '1.02.2016', '5%');
INSERT INTO "Продажи" VALUES (2, 'Стул', '5.02.2016', '5%');
INSERT INTO "Продажи" VALUES (3, 'Стул', '7.02.2016', '5%');
INSERT INTO "Продажи" VALUES (4, 'Стул', '8.02.2016', '5%');
INSERT INTO "Продажи" VALUES (5, 'Стул', '9.02.2016', '5%');
INSERT INTO "Продажи" VALUES (6, 'Стул', '10.02.2016', '5%');
INSERT INTO "Продажи" VALUES (7, 'Стул', '11.02.2016', '5%');
INSERT INTO "Продажи" VALUES (8, 'Стул', '12.02.2016', '5%');
INSERT INTO "Продажи" VALUES (9, 'Стул', '15.02.2016', '5%');
INSERT INTO "Продажи" VALUES (10, 'Стол', '2.02.2016', '10%');
INSERT INTO "Продажи" VALUES (11, 'Стол', '4.02.2016', '10%');
INSERT INTO "Продажи" VALUES (12, 'Кровать', '2.02.2016', '10%');
```

```
INSERT INTO "Продажи" VALUES (13, 'Кровать', '6.02.2016', '10%');
INSERT INTO "Продажи" VALUES (14, 'Кровать', '7.02.2016', '10%');
INSERT INTO "Продажи" VALUES (15, 'Кровать', '12.02.2016', '10%');
INSERT INTO "Продажи" VALUES (16, 'Кровать', '13.02.2016', '10%');
INSERT INTO "Продажи" VALUES (17, 'Кровать', '14.02.2016', '10%');
INSERT INTO "Продажи" VALUES (18, 'Кровать', '15.02.2016', '10%');


WITH
--выберем все данные и упорядочим их по названию позиции и размеру скидки, а внутри каждой такой группы - по дате
a1 AS (
   SELECT *
   FROM "Продажи"
   ORDER BY "Название товара", "Скидка %", "Дата"),
--сопоставим каждой позиции позицию, следующую за ней
a2 AS (
     SELECT "Номер", "Название товара", "Дата", "Скидка %", LEAD("Название товара",1) OVER(ORDER BY "Название товара", "Скидка %", "Дата") AS prev_position,
       LEAD("Дата",1) OVER(ORDER BY "Название товара", "Скидка %", "Дата") AS prev_date,
          LEAD("Скидка %",1) OVER(ORDER BY "Название товара", "Скидка %", "Дата") AS prev_discount
   FROM a1),
--посчитаем разницу дат между соседними позициями, и если она равна единице и значения POSITION/DISCOUNT соответственно совпадают в столбец GRUP запишем ноль(принадлежность тому же временному отрезку), иначе поставим единицу(переход в другой временной отрезок/другую группу)
a3 AS (
     SELECT ROWNUM rn, "Номер", "Название товара", "Дата", "Скидка %", prev_position, prev_date, prev_discount,
       prev_date - "Дата" AS days,
        CASE WHEN ( prev_date - "Дата" ) = 1 AND prev_discount = "Скидка %" AND prev_position = "Название товара" THEN 0
       ELSE 1 END AS grup
   FROM a2),
--найдем накапливающуюся сумму для столбца grup. Таким образом мы разобьем все записи на группы
```

```
a4 AS (
    SELECT rn, "Номер", "Название товара", "Дата", "Скидка %", grup, (SUM(grup) OVER(ORDER BY rn)) AS res
  FROM a3
),
--нам необходимо сдвинуть полученные группы на 1 запись, т.к. новая группа на данный момент начинается с последней записи, принадлежащей предыдщуему временному отрезку
a5 AS (
    SELECT rn, "Номер", "Название товара", "Дата", "Скидка %", nvl( (LAG(res,1) OVER(ORDER BY rn)),0) AS inlinegroup
  FROM a4
),
--теперь имея группы с их номером, выберем название это группы (Position & Discount) и две даты - максимальную и минимальную в этой группе
a6 AS (
   SELECT "Название товара", "Скидка %", MIN("Дата") AS mid, MAX("Дата") AS mad
  FROM a5
  GROUP BY "Название товара", "Скидка %", inlinegroup
),
--Теперь для каждой группы отобразим соответствующий промежуток времени: если минимальная и максимальная даты совпадают, то просто выведем ее, иначе выпишим две крайние даты через дефис
a7 AS (
        SELECT "Название товара", CASE WHEN mid = mad THEN TO_CHAR(mid,'dd.mm.yyyy')
            ELSE TO_CHAR(mid,'dd.mm.yyyy') || '-' || TO_CHAR(mad,'dd.mm.yyyy') END AS dates,
      mid, "Скидка %"
  FROM a6)
--С помощью listagg соединим группы с одинаковыми названиями товаром и скидкой в одну строчку и выведем требуемый результат
SELECT DISTINCT "Название товара", LISTAGG(dates, ',') WITHIN GROUP(ORDER BY mid) OVER(PARTITION BY "Название товара", "Скидка %") AS aa, "Скидка %"
FROM a7
ORDER BY "Название товара";
```

2. Используя словарь данных, получить информацию о первичных ключах и подчиненных таблицах всех таблиц в схеме HR:

Имя таблицы	**Список столбцов первичного ключа**	**Список подчиненных таблиц**

В списках имена столбцов и подчиненных таблиц вывести через запятую по алфавиту.

Задачу решить без использования функций Listagg и Wm_concat.

Решение 1:

```
WITH
t AS
   (SELECT ut.table_name, ucc.column_name,
   ROW_NUMBER() OVER(PARTITION BY ut.table_name ORDER BY
ucc.column_name) num
   FROM all_tables ut
   LEFT JOIN user_constraints uc
   ON ut.table_name = uc.table_name
   LEFT JOIN user_cons_columns ucc
   ON uc.constraint_name = ucc.constraint_name
   AND uc.constraint_type = 'P'
   WHERE ut.owner = 'HR'),
t1 AS
   (SELECT table_name,
   TRIM(',' FROM regexp_replace(substr(sys_connect_by_path(column_name,','),2),',,',',') )
pcol
   FROM t
   WHERE CONNECT_BY_ISLEAF = 1
   START WITH num = 1
   CONNECT BY NOCYCLE PRIOR num + 1 = num
   AND PRIOR table_name = table_name),
t2 AS
   (SELECT ut.table_name, ucc.table_name tn,
   ROW_NUMBER() OVER(PARTITION BY ut.table_name ORDER BY
ucc.table_name) num
   FROM all_tables ut
```

```
    LEFT JOIN user_constraints uc
    ON ut.table_name = uc.table_name
    AND uc.constraint_type = 'R'
    LEFT JOIN user_cons_columns ucc
    ON uc.r_constraint_name = ucc.constraint_name
    WHERE ut.owner = 'HR'),
t3 AS
    (SELECT table_name,
    TRIM(',' FROM regexp_replace(substr(sys_connect_by_path(tn,','),2),',,',') ) rtab
    FROM t2
    WHERE CONNECT_BY_ISLEAF = 1
    START WITH num = 1
    CONNECT BY NOCYCLE PRIOR num + 1 = num
    AND PRIOR table_name = table_name)

SELECT table_name "Имя таблицы",
nvl(pcol,' ') "Список столб. первичного ключа",
nvl(rtab,' ') "Список подчиненных таблиц"
FROM t3
JOIN t1 USING ( table_name );
```

3. Определить, сколько раз каждая из цифр от 0 до 9 встречается в столбце Phone_number таблицы Employees.

Пример результата:

Цифра	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Количество	15	12	23	45	24	33	45	12	30	15

```
    WITH
    counts AS
    (SELECT TO_CHAR(0) n, regexp_count(phone_number,0) cnt
    FROM employees
    UNION ALL
    SELECT TO_CHAR(1) n, regexp_count(phone_number,1) cnt
```

```
FROM employees
UNION ALL
SELECT TO_CHAR(2) n, regexp_count(phone_number,2) cnt
FROM employees
UNION ALL
SELECT TO_CHAR(3) n, regexp_count(phone_number,3) cnt
FROM employees
UNION ALL
SELECT TO_CHAR(4) n, regexp_count(phone_number,4) cnt
FROM employees
UNION ALL
SELECT TO_CHAR(5) n, regexp_count(phone_number,5) cnt
FROM employees
UNION ALL
SELECT TO_CHAR(6) n, regexp_count(phone_number,6) cnt
FROM employees
UNION ALL
SELECT TO_CHAR(7) n, regexp_count(phone_number,7) cnt
FROM employees
UNION ALL
SELECT TO_CHAR(8) n, regexp_count(phone_number,8) cnt
FROM employees
UNION ALL
SELECT TO_CHAR(9) n, regexp_count(phone_number,9) cnt
FROM employees)

SELECT*
FROM (SELECT 'Количество' "Цифра", n, cnt
   FROM counts)
PIVOT (SUM(cnt) FOR n IN ( 0,1,2,3,4,5,6,7,8,9 ));
```

4. Создать запрос для построения отчета по количеству студентов, сдававших экзамены в определенные дни в произвольно заданном интервале дат по различным дисциплинам. Результаты должны быть отсортированы по датам, количеству студентов и названию дисциплин. Одна и та же дата должна встречаться в отчете не более двух раз: на первой строке данной даты и на

отчетной строке даты. Отчет должен иметь следующий вид:

Дата	День недели	Дисциплина	Количество студентов
13.01.15	Вторник	Экономика	1
		Химия	2
13.01.15: Итого	Вторник		3
14.01.15: Итого	Среда		0
15.01.15	Четверг	Математика	1
		Физика	1
		Экономика	3
15.01.15: Итого	Четверг		5
16.01.15	Пятница	Менеджмент	1
16.01.15: Итого	Пятница		1
17.01.15	Суббота	Математика	4
17.01.15: Итого	Суббота		4
18.01.15:Итого	Воскресенье		0

19.01.15	Понедельник	Экономика	1
		Физика	2
19.01.15: Итого	Понедельник		3
20.01.15	Вторник	Химия	2
20.01.15: Итого	Вторник		2
		ОБЩИЙ ИТОГ	18

Задачу решить с использованием раздела Model.

данных за 15 год НЕТ, выведет пустые строчки, работает при ('10.06.1999') и ('20.06.1999') и тд.

Решение 1:

```
UNDEFINE date1;
UNDEFINE date2;


WITH
dates AS (
   SELECT TO_DATE('&&date1', 'dd.MM.yyyy') + level - 1 дата
   FROM dual
        CONNECT BY TO_DATE('&&date1', 'dd.MM.yyyy') + level - 1 <=
     TO_DATE('&&date2', 'dd.MM.yyyy')),
tmp AS (
   SELECT дата, to_char(дата, 'Day') день_недели, название, COUNT(номер_студента)
     количество,
        GROUPING(дата) gr1, GROUPING(название) gr2, ROW_NUMBER()
     OVER(PARTITION BY дата ORDER BY название) rnum
   FROM dates LEFT JOIN успеваемость USING ( дата )
        LEFT JOIN дисциплины USING ( номер_дисциплины )
   GROUP BY ROLLUP(дата, название))
SELECT CASE WHEN gr1 = 0 AND gr2 = 0 AND rnum = 1 THEN to_char(дата,
```

```
'dd.MM.yy')
    WHEN gr2 = 1 AND gr1 = 0 THEN to_char(дата, 'dd.MM.yy') || ': Итого'
ELSE ' ' END "Дата",
 CASE WHEN gr1 = 0 AND gr2 = 0 AND rnum = 1 OR gr2 = 1 AND gr1 = 0 THEN
день_недели
ELSE ' ' END "День недели",
CASE WHEN gr1 = 1 AND gr2 = 1 THEN 'ОБЩИЙ ИТОГ'
ELSE nvl(название, ' ') END "Дисциплина", количество  "Количество студентов"
FROM tmp
WHERE NOT (название IS NULL AND gr2 = 0 );
```

5. Для произвольной строки, состоящей из цифр, определить все возможные наборы слов, получаемые при замене чисел на номер буквы в русском алфавите. Например, для строки 211221 результатом должно быть: баабба, бйбба,баафа,бакба,уку,……

Решение 1:

```
WITH
src as(
   SELECT '211221' str
   FROM dual
),
--формируем все числа, которые встречаются в строке
--а так как нам нужен номер буквы алфавита, то одно- либо дву-значное
all_nums AS (
   SELECT to_number(substr(str, level, 1)) num, level AS pos, length(str) AS len
   FROM src
   CONNECT BY level<=length(str)
   UNION
   SELECT to_number(substr(str, level, 2)), level, length(str) AS len
   FROM src
   CONNECT BY level<=length(str)),
--фильтруем номера, которые не соответствуют позиции буквы в алфавите
--и заодно добавляем позицию начала и конца номера в первоначальной строке
all_nums_filter AS (
  SELECT num, pos, (CASE WHEN length(to_char(num))=1 THEN pos
                ELSE pos+1 END) pos_end, len
  FROM all_nums
```

```
  WHERE num BETWEEN 1 AND 33),
--собираем полученные числа в последовательность букв
--по правилу: конец предыдущего числа равен началу следующего
seq AS (
        SELECT  Sys_Connect_By_Path(substr('абвгдеёжзийклмнопрстуфхцчшщъыьэюя',
      num, 1), ' ') path, pos_end
    FROM all_nums_filter
    START WITH pos=1
    CONNECT BY prior pos_end=pos-1)
--выводим ответ
SELECT replace(path, ' ', '') res
FROM seq
WHERE pos_end=6;
```

БИЛЕТ 5

1. Напишите запрос к таблице Трафик (Дата и время начала транзакции, Дата время окончания транзакции, Объем в байтах), который на каждый день заданного месяца и года сосчитает количество переданного трафика. Если сессия началась и закончилась в разные дни, то трафик следут разделить пропорционально длительности в каждом дне. Если результат пропорционального деления дробный, то отбросить дробную часть для начального дня, а остаток относить к окончанию сессии. Продолжительность сессии не ограничена во времени.

вроде работает, но я могла неправильно понять условие

Решение 1:

```
CREATE TABLE traffic (tr_start DATE, tr_finish DATE, tr_bytes NUMBER);


INSERT INTO traffic VALUES (TO_DATE('10.1.2018 11:08', 'dd.mm.yyyy hh24:mi'), TO_DATE('12.1.2018 01:24', 'dd.mm.yyyy hh24:mi'), 3555);
INSERT INTO traffic VALUES (TO_DATE('11.1.2018 12:00', 'dd.mm.yyyy hh24:mi'), TO_DATE('11.1.2018 13:13', 'dd.mm.yyyy hh24:mi'), 1642);
INSERT INTO traffic VALUES (TO_DATE('12.1.2018 00:05', 'dd.mm.yyyy hh24:mi'), TO_DATE('12.1.2018 1:02', 'dd.mm.yyyy hh24:mi'), 655);
INSERT INTO traffic VALUES (TO_DATE('10.1.2018 23:07', 'dd.mm.yyyy hh24:mi'), TO_DATE('11.1.2018 14:11', 'dd.mm.yyyy hh24:mi'), 2555);


DEFINE inp_date = TO_DATE('&input', 'mm.yyyy');


WITH
session_info AS (
--Выведем для каждой сессии дату ее начала, дату конца, сколько байт передано,
--сколько байт передается за день(дробное число)
   SELECT tr_start, tr_finish, tr_bytes, tr_bytes / ( tr_finish - tr_start ) band
   FROM traffic),
--расчет для каждого трайика, сколько передано за целый день
traffic_per_day (tr_day, tr_bytes, sum_bytes, band, tr_finish, all_bytes) AS (
     SELECT trunc(tr_start, 'dd') + 1 AS tr_day, --текущий день (+1, так как на этот день рассчитано здесь)
        (CASE WHEN trunc(tr_start, 'dd') = trunc(tr_finish, 'dd') THEN tr_bytes
      --если трафик открылся и закрылся в один день, то за этот день передан весь трафик
           --иначе только часть от общего = разница между началом и следующим днем, умноженная
      --на band - скорость передачи
           ELSE trunc((trunc(tr_start, 'dd') + 1 - tr_start) * band, 0) END) AS tr_bytes, 0 AS sum_bytes, --сумма трафика за все дни
```

```
        band AS band, --скорость передачи байт
        tr_finish AS tr_finish, --дата окончания
        tr_bytes AS all_bytes --всего байт в трафике
    FROM session_info
    UNION ALL
    SELECT tr_day + 1, --увеличиваем текущий день на 1
        (CASE WHEN trunc(tr_finish, 'dd') > tr_day THEN band --если этот день не последний, то трафик за день = band
        --иначе - остатку трафика
        ELSE all_bytes - sum_bytes - tr_bytes END),
        sum_bytes + tr_bytes, --суммируем трафик за все дни с текущим
        band, tr_finish, all_bytes
    FROM traffic_per_day
    WHERE tr_day <= tr_finish --выходим, если достигли последней даты
), all_days AS ( --формируем дни за введенный месяц
    SELECT trunc(&inp_date, 'mm') + level - 1 AS tr_day
    FROM (SELECT &inp_date
        FROM dual)
    CONNECT BY trunc(&inp_date, 'mm') + level - 1 <= last_day(&inp_date))
SELECT ad.tr_day, nvl(to_char(SUM(tr.tr_bytes)), ' ') AS traffic
FROM all_days ad LEFT JOIN traffic_per_day  tr ON ( ad.tr_day = tr.tr_day - 1 )
--дни в таблице трафиков (traffic_per_day) оказались смещены на 1, так как
--мы сразу расчитывали трафик за первый день поэтому вычитаем этот день
GROUP BY ad.tr_day
ORDER BY ad.tr_day ASC;

UNDEFINE inp_date;

DROP TABLE TRAFFIC;
```

2. Имеется таблица со столбцом типа varchar2(2000), содержащем информацию о названиях документов, например:

Документоплатабанк

Сотрудниквыплата

Документналогбанксотрудник

и т.д.

Кроме того имеется таблица с сокращениями отдельных выражений. Например:

Полное выражение	Укороченное выражение
Документ	Док
Сотрудник	Сотр
Банк	Б
Оплата	Оп

Требуется вывести полные названия документов и сокращенные.

Пример результата:

Полное название документа	Сокращенное название
Документоплатабанк	ДокОпБ
Сотрудниквыплата	Сотрвыплата
Документналогбанксотрудник	ДокналогБСотр

Решение 1:

```
CREATE table ishodnik
(fulls VARCHAR2(2000));

CREATE table sokr
(fulls VARCHAR2(2000),
shorts VARCHAR2(100));

INSERT INTO ishodnik(fulls) VALUES ('Документоплатабанк');
INSERT INTO ishodnik(fulls) VALUES ('Сотрудниквыплата');
INSERT INTO ishodnik(fulls) VALUES ('Документналог банксотрудник');
INSERT INTO sokr(fulls, shorts) VALUES ('Документ', 'Док');
INSERT INTO sokr(fulls, shorts) VALUES ('Сотрудник', 'Сотр');
INSERT INTO sokr(fulls, shorts) VALUES ('Банк', 'Б');
INSERT INTO sokr(fulls, shorts) VALUES ('Оплата', 'Оп');

WITH
num_ishodnik AS (
   SELECT 0 AS x, fulls
   FROM ishodnik),
```

```
num_sokr AS (
   SELECT ROWNUM y, fulls,shorts
   FROM sokr),
tab (x, fulls, cutted) AS (
--выбираем все строки, для каждой рекурсия будет применять отдельно
   SELECT x, fulls, fulls AS cutted FROM num_ishodnik
   UNION ALL
--номер проверки увеличивается, все «длинные»вхождения заменяются на «короткие»
                SELECT x+1, tab.fulls, replace(replace(cutted,ns.fulls,ns.shorts), lower(ns.fulls),ns.shorts)
   FROM tab JOIN num_sokr ns ON tab.x+1 = ns.y
--остановимся, когда дойдем до последнего сокращения
   WHERE  x <= (SELECT MAX(y)
         FROM num_sokr))
SELECT fulls AS "Полное название документа",cutted "Сокращенное название"
FROM tab
WHERE x = (SELECT MAX(y)
      FROM num_sokr);
```

3. Имеется таблица Продажи (Номер, Название товара, Дата, Скидка %). Вывести отчет по продажам, который включает столбцы Название товара, Даты продажи, Скидка %, представив информацию таким образом, что если один и тот же товар продавался с одной и той же скидкой несколько дней, то эти даты должны выводиться через запятую. При этом если две или более даты отличаются друг от друга на один день, то они должны быть представлены в виде интервала с дефисом в качестве разделителя.

Пример представления результата:

Название товара	**Даты продажи**	**Скидка, %**
Стул	1.02.2016, 5.02.2016, 7.02.2016-12.02.2016, 15.02.2016	5
Стол	2.02.2016, 4.02.2016	10
Кровать	2.02.2016, 6.02.2016 - 7.02.2016, 12.02.2016-15.02.2016	10

Решение 1:

```
CREATE TABLE "Продажи" ("Номер" NUMBER(8) NOT NULL, "Название товара" VARCHAR2(16), "Дата" DATE, "Скидка %" VARCHAR2(16));


INSERT INTO "Продажи" VALUES (1, 'Стул', '1.02.2016', '5%');
INSERT INTO "Продажи" VALUES (2, 'Стул', '5.02.2016', '5%');
INSERT INTO "Продажи" VALUES (3, 'Стул', '7.02.2016', '5%');
INSERT INTO "Продажи" VALUES (4, 'Стул', '8.02.2016', '5%');
INSERT INTO "Продажи" VALUES (5, 'Стул', '9.02.2016', '5%');
INSERT INTO "Продажи" VALUES (6, 'Стул', '10.02.2016', '5%');
INSERT INTO "Продажи" VALUES (7, 'Стул', '11.02.2016', '5%');
INSERT INTO "Продажи" VALUES (8, 'Стул', '12.02.2016', '5%');
INSERT INTO "Продажи" VALUES (9, 'Стул', '15.02.2016', '5%');
INSERT INTO "Продажи" VALUES (10, 'Стол', '2.02.2016', '10%');
INSERT INTO "Продажи" VALUES (11, 'Стол', '4.02.2016', '10%');
INSERT INTO "Продажи" VALUES (12, 'Кровать', '2.02.2016', '10%');
INSERT INTO "Продажи" VALUES (13, 'Кровать', '6.02.2016', '10%');
INSERT INTO "Продажи" VALUES (14, 'Кровать', '7.02.2016', '10%');
INSERT INTO "Продажи" VALUES (15, 'Кровать', '12.02.2016', '10%');
INSERT INTO "Продажи" VALUES (16, 'Кровать', '13.02.2016', '10%');
INSERT INTO "Продажи" VALUES (17, 'Кровать', '14.02.2016', '10%');
INSERT INTO "Продажи" VALUES (18, 'Кровать', '15.02.2016', '10%');


WITH
--выберем все данные и упорядочим их по названию позиции и размеру скидки, а внутри каждой такой группы - по дате
a1 AS (
  SELECT *
  FROM "Продажи"
  ORDER BY "Название товара", "Скидка %", "Дата"),
--сопоставим каждой позиции позицию, следующую за ней
a2 AS (
    SELECT "Номер", "Название товара", "Дата", "Скидка %", LEAD("Название товара",1) OVER(ORDER BY "Название товара", "Скидка %", "Дата") AS
```

```
    prev_position,
      LEAD("Дата",1) OVER(ORDER BY "Название товара", "Скидка %", "Дата") AS prev_date,
        LEAD("Скидка %",1) OVER(ORDER BY "Название товара", "Скидка %", "Дата") AS prev_discount
  FROM a1),
--посчитаем разницу дат между соседними позициями, и если она равна единице и значения POSITION/DISCOUNT соответственно совпадают в столбец GRUP запишем ноль(принадлежность тому же временному отрезку), иначе поставим единицу(переход в другой временной отрезок/другую группу)
a3 AS (
    SELECT ROWNUM rn, "Номер", "Название товара", "Дата", "Скидка %", prev_position, prev_date, prev_discount,
      prev_date - "Дата" AS days,
       CASE WHEN ( prev_date - "Дата" ) = 1 AND prev_discount = "Скидка %" AND prev_position = "Название товара" THEN 0
      ELSE 1 END AS grup
  FROM a2),
--найдем накапливающуюся сумму для столбца grup. Таким образом мы разобьем все записи на группы
a4 AS (
    SELECT rn, "Номер", "Название товара", "Дата", "Скидка %", grup, (SUM(grup) OVER(ORDER BY rn)) AS res
  FROM a3
),
--нам необходимо сдвинуть полученные группы на 1 запись, т.к. новая группа на данный момент начинается с последней записи, принадлежащей предыдщуему временному отрезку
a5 AS (
    SELECT rn, "Номер", "Название товара", "Дата", "Скидка %", nvl( (LAG(res,1) OVER(ORDER BY rn)),0) AS inlinegroup
  FROM a4
),
--теперь имея группы с их номером, выберем название это группы (Position & Discount) и две даты - максимальную и минимальную в этой группе
a6 AS (
    SELECT "Название товара", "Скидка %", MIN("Дата") AS mid, MAX("Дата") AS mad
  FROM a5
```

```
    GROUP BY "Название товара", "Скидка %", inlinegroup
),
--Теперь для каждой группы отобразим соответствующий промежуток времени: если минимальная и максимальная даты совпадают, то просто выведем ее, иначе выпишим две крайние даты через дефис
a7 AS (
        SELECT "Название товара", CASE WHEN mid = mad THEN TO_CHAR(mid,'dd.mm.yyyy')
            ELSE TO_CHAR(mid,'dd.mm.yyyy') || '-' || TO_CHAR(mad,'dd.mm.yyyy') END AS dates,
        mid, "Скидка %"
    FROM a6)
--С помощью listagg соединим группы с одинаковыми названиями товаром и скидкой в одну строчку и выведем требуемый результат
SELECT DISTINCT "Название товара", LISTAGG(dates, ',') WITHIN GROUP(ORDER BY mid) OVER(PARTITION BY "Название товара", "Скидка %") AS aa, "Скидка %"
FROM a7
ORDER BY "Название товара";
```

4. Используя словарь данных, получить информацию о первичных ключах и подчиненных таблицах всех таблиц Вашей схемы :

Имя таблицы	**Список столбцов первичного ключа**	**Список подчиненных таблиц**

В списках имена столбцов и подчиненных таблиц вывести через запятую по алфавиту.

Задачу решить без использования функций Listagg и Wm_concat.

Решение 1:

```
WITH
t AS
   (SELECT ut.table_name, ucc.column_name,
   ROW_NUMBER() OVER(PARTITION BY ut.table_name ORDER BY ucc.column_name) num
   FROM user_tables ut
   LEFT JOIN user_constraints uc
   ON ut.table_name = uc.table_name
   LEFT JOIN user_cons_columns ucc
```

```
   ON uc.constraint_name = ucc.constraint_name
   AND uc.constraint_type = 'P'),
t1 AS
   (SELECT table_name,
   TRIM(',' FROM regexp_replace(substr(sys_connect_by_path(column_name,','),2),',,',',') ) pcol
   FROM t
   WHERE CONNECT_BY_ISLEAF = 1
   START WITH num = 1
   CONNECT BY NOCYCLE PRIOR num + 1 = num
   AND PRIOR table_name = table_name),
t2 AS
   (SELECT ut.table_name, ucc.table_name tn,
   ROW_NUMBER() OVER(PARTITION BY ut.table_name ORDER BY ucc.table_name)
num
   FROM user_tables ut
   LEFT JOIN user_constraints uc
   ON ut.table_name = uc.table_name
   AND uc.constraint_type = 'R'
   LEFT JOIN user_cons_columns ucc
   ON uc.r_constraint_name = ucc.constraint_name),
t3 AS
   (SELECT table_name,
   TRIM(',' FROM regexp_replace(substr(sys_connect_by_path(tn,','),2),',,',',') ) rtab
   FROM t2
   WHERE CONNECT_BY_ISLEAF = 1
   START WITH num = 1
   CONNECT BY NOCYCLE PRIOR num + 1 = num
   AND PRIOR table_name = table_name)

SELECT table_name "Имя таблицы",
nvl(pcol,'Первичного ключа нет') "Список столб. первичного ключа",
nvl(rtab,'Подчиненных таблиц нет') "Список подчининных таблиц"
FROM t3
JOIN t1 USING ( table_name );
```

5. Создать запрос для построения отчета по количеству студентов, сдававших экзамены в определенные дни в произвольно заданном интервале дат по различным дисциплинам. Результаты должны быть отсортированы по датам, количеству студентов и названию дисциплин. Одна и та же дата должна встречаться в отчете не более двух раз: на первой строке данной даты и на отчетной строке даты.

Задачу решить с использованием раздела Model.

Отчет должен иметь следующий вид:

Дата	**День недели**	**Дисциплина**	**Количество студентов**

13.01.15	Вторник	Экономика	1
		Химия	2
13.01.15: Итого	Вторник		3
14.01.15: Итого	Среда		0
15.01.15	Четверг	Математика	1
		Физика	1
		Экономика	3
15.01.15: Итого	Четверг		5
16.01.15	Пятница	Менеджмент	1
16.01.15: Итого	Пятница		1
17.01.15	Суббота	Математика	4
17.01.15: Итого	Суббота		4
18.01.15:Итого	Воскресенье		0
19.01.15	Понедельник	Экономика	1
		Физика	2

19.01.15: Итого	Понедельник		3
20.01.15	Вторник	Химия	2
20.01.15: Итого	Вторник		2
		ОБЩИЙ ИТОГ	18

данных за 15 год НЕТ, выведет пустые строчки, работает при ('10.06.1999') и ('20.06.1999') и тд.

Решение 1:

```
UNDEFINE date1;
UNDEFINE date2;

WITH
dates AS (
   SELECT TO_DATE('&&date1', 'dd.MM.yyyy') + level - 1 дата
   FROM dual
        CONNECT BY TO_DATE('&&date1', 'dd.MM.yyyy') + level - 1 <= TO_DATE('&&date2', 'dd.MM.yyyy')),
tmp AS (
   SELECT дата, to_char(дата, 'Day') день_недели, название, COUNT(номер_студента) количество,
        GROUPING(дата) gr1, GROUPING(название) gr2, ROW_NUMBER() OVER(PARTITION BY дата ORDER BY название) rnum
   FROM dates LEFT JOIN успеваемость USING ( дата )
        LEFT JOIN дисциплины USING ( номер_дисциплины )
   GROUP BY ROLLUP(дата, название))
SELECT CASE WHEN gr1 = 0 AND gr2 = 0 AND rnum = 1 THEN to_char(дата, 'dd.MM.yy')
       WHEN gr2 = 1 AND gr1 = 0 THEN to_char(дата, 'dd.MM.yy') || ': Итого'
    ELSE ' ' END "Дата",
    CASE WHEN gr1 = 0 AND gr2 = 0 AND rnum = 1 OR gr2 = 1 AND gr1 = 0 THEN день_недели
    ELSE ' ' END "День недели",
```

```
    CASE WHEN gr1 = 1 AND gr2 = 1 THEN 'ОБЩИЙ ИТОГ'
    ELSE nvl(название, ' ') END "Дисциплина", количество  "Количество студентов"
FROM tmp
WHERE NOT (название IS NULL AND gr2 = 0 );
```

БИЛЕТ 6

1. Одной командой вывести все палиндромы, встречающиеся в произвольной символьной строке.

Например, для строки aaabacdca ответ должен быть:

aa,aaa,aba, cdc, acdca

Решение 1:

```
DEFINE str = "aabacdca";
DEFINE str = "aA        bacdca          ";

WITH
    trimw AS (SELECT TRIM(both ' ' from replace(replace(replace('&STR', ' ', ' _'), '_ '), '_')) AS tstr
        FROM dual),
    source AS (SELECT substr(tstr,level,1) AS lt, ROW_NUMBER() OVER(ORDER BY level ASC) rn
        FROM trimw
        CONNECT BY level <= length(tstr)),
    tmp AS (SELECT replace(sys_connect_by_path(lt,','),',') word, level l, ROWNUM rn
        FROM source
        WHERE level > 1
        CONNECT BY PRIOR rn + 1 = rn),
    result AS (SELECT DISTINCT word w1, (SELECT LISTAGG(regexp_substr(word,'.',level,1), '') WITHIN GROUP(ORDER BY ROWNUM DESC)
        FROM dual
        CONNECT BY level <= length(word)) AS w2
    FROM tmp)

SELECT '&STR' AS "Строка", nvl(LISTAGG(TRIM(w1), ',') WITHIN GROUP(ORDER BY w1), 'Палиндромов нет') AS "Палиндромы"
FROM result
WHERE LOWER(w1) = LOWER(w2);
--если учитываем регистр, то WHERE w1 = w2;

UNDEFINE str;
```

2. Для произвольной строки, состоящей из открывающих и закрывающих скобок написать запрос для вывода всех слов максимальной длины, представляющих

правильные скобочные записи. Например, для строки (()(() ответ должен быть:

()()

(())

Решение 1:

```
WITH input AS (SELECT '(()(()(' AS str, '' AS s
            FROM dual),
    tmp1 AS (SELECT str, level AS lvl, substr(str,level,1) AS c
            FROM input
            CONNECT BY level <= length(str)),
    tmp2 AS (SELECT DISTINCT str, replace(sys_connect_by_path(c,'.'),'.','') s
            FROM tmp1
            CONNECT BY PRIOR str = str AND PRIOR lvl < lvl AND PRIOR sys_guid() IS
NOT NULL),
    tmp4 (s, lvl, c) AS (SELECT s, 1 AS lvl, 1 AS c
                    FROM tmp2
                    WHERE s LIKE '(%' AND regexp_count(s,'\(') = regexp_count(s,'\)')
                    UNION ALL
                    SELECT s, lvl + 1, CASE WHEN substr(s,lvl + 1,1) = '('
                                    THEN c + 1
                                    ELSE c - 1
                                END AS c
                    FROM tmp4
                    WHERE lvl < 1 + length(s) AND c >= 0),
     tmp5 AS (SELECT *
            FROM tmp4
            WHERE lvl = length(s))
SELECT s AS result
FROM tmp5
WHERE c = 0 AND   length(s) = (SELECT MAX(length(s) )
                        FROM tmp5
                        WHERE c = 0);
```

	RESULT
1	() ()
2	(( ))

3. Имеется таблица D_V с первым столбцом Dat типа DATE (первичный ключ) и вторым столбцом Val типа NUMBER. Пример (строки упорядочены по первому

столбцу):

DAT	VAL
01-08-08	232
02-08-08	
10-08-08	182
11-08-08	
21-08-08	240
22-08-08	
23-08-08	

Требуется написать запрос для получения на основе таблицы D_V следующей таблицы:

DAT	MAX_VAL
01-08-08	232
02-08-08	232
10-08-08	182
11-08-08	182
21-08-08	240
22-08-08	240
23-08-08	240

Данная результирующая таблица должна быть упорядочена по Dat, но вместо пустых значений, которые присутствовали в столбце VAL отсортированной по DAT исходной таблицы, в столбце MAX_VAL результирующей таблицы, должны присутствовать значения столбца из предыдущей строки.

Решение 1:

```
CREATE TABLE D_V (dat DATE PRIMARY KEY, val NUMBER);


INSERT INTO D_V VALUES('01-08-08', 232);
INSERT INTO D_V VALUES('02-08-08', NULL);
INSERT INTO D_V VALUES('10-08-08', 182);
INSERT INTO D_V VALUES('11-08-08', NULL);
INSERT INTO D_V VALUES('21-08-08', 240);
INSERT INTO D_V VALUES('22-08-08', NULL);
INSERT INTO D_V VALUES('23-08-08', NULL);

SELECT dat, CASE WHEN val IS NULL
            THEN (SELECT tmp1.val
```

```
                FROM D_V tmp1
                WHERE tmp1.dat = (SELECT MAX(dat)
                        FROM D_V
                        WHERE dat < tmp2.dat AND val IS NOT NULL))
            ELSE val END AS max_val
FROM D_V tmp2
ORDER BY dat;
```

	DAT	MAX_VAL
1	01.08.08	232
2	02.08.08	232
3	10.08.08	182
4	11.08.08	182
5	21.08.08	240
6	22.08.08	240
7	23.08.08	240

4. Создайте таблицу для хранения каталога товаров:

create table catalog(text varchar2(4000));

где text – информация о товарах, заданная в формате:

Код товара1/Тип товара1-Наименование товара1:Цена товара1;Код товара2/Тип товара2-Наименование товара2:Цена товара2;Код товара3/Тип товара3-Наименование товара3:Цена товара3;...;Код товараN/Тип товараN-Наименование товараN:Цена товараN

Требования к формату информации о товарах:

- Товары разделены точкой с запятой, после последнего товара точки с запятой нет;

- Код товара отделяется символом «слэш», имеет длину от 1 до 6 символов, допустимы только цифры;

- Тип товара имеет нефиксированную длину, отделяется символом «минус» (коротким тире), содержит любые символы;

- Наименование товара имеет нефиксированную длину, отделяется двоеточием, содержит любые символы;

- Цена товара может иметь дробное значение, при этом целая и дробная часть могут разделяться точкой или запятой;

- В коде, типе, наименовании, цене, недопустимо присутствие любого из перечисленных символов разделения (не допускаются символы точка с запятой, «слэш», «минус», двоеточие).

Заполните таблицу данными о товарах:

insert into catalog values ('125/refrigerator-Indesit SB200 T:17999.99;50/microwave-Samsung MT479:7499,99;103320/teakettle-Bosch TWK189:4890,32');

insert into catalog values ('05/pan-Tefal 040 80:849,00;125/pan-Tefal E20 60:3599,2;434031/iron-Braun Texstyle 535:5490,01');

commit;

Одним SQL-Запросом необходимо вывести таблицу товаров, отсортированную по возрастанию кода товара, затем по типу товара.

Формат результата

Код товара	**Тип товара**	**Наименование товара**	**Цена товара**
000005	Pan	Tefal 040 80	849
000050	Microwave	Samsung MT479	7499.99
000125	Pan	Tefal E20 60	3599.2
000125	Refrigerator	Indesit SB200 T	17999.99
...	...	...	...

В случае, когда длина кода товара меньше шести символов, необходимо дополнять код товара незначащими нулями слева до максимальной длины шесть символов.

Тип колонки цены товара в результирующем наборе должен быть числовым, тип остальных колонок – строковым.

Решение 1:

```
CREATE TABLE catalog (text VARCHAR2(4000));


INSERT INTO catalog VALUES ('125/refrigerator-Indesit SB200 T:17999.99;50/microwave-Samsung MT479:7499,99;103320/teakettle-Bosch TWK189:4890,32');

INSERT INTO catalog VALUES ('05/pan-Tefal 040 80:849,00;125/pan-Tefal E20 60:3599,2;434031/iron-Braun Texstyle 535:5490,01');

COMMIT;


WITH tmp AS
(SELECT DISTINCT REGEXP_SUBSTR(text, '[^;]+', 1, level) txt1
 FROM catalog CONNECT BY REGEXP_SUBSTR(text, '[^;]+', 1, level) IS NOT NULL)
SELECT LPAD(REGEXP_SUBSTR(txt1, '(\;)?\d{1,6}+'), 6, 0) "Код товара",
    INITCAP(REGEXP_SUBSTR(txt1, '\w+', 1, 2)) "Тип товара",
    LTRIM(REGEXP_SUBSTR(txt1, '-[^:]+'), '-') "Наименование товара",
    REPLACE(LTRIM(REGEXP_SUBSTR(txt1, ':\d+(\.|,)?\d+'), ':'), ',', '.') "Цена товара"
```

FROM tmp

ORDER BY "Код товара", "Тип товара";

	Код товара	Тип товара	Наименование товара	Цена товара
1	000005	Pan	Tefal 040 80	849.00
2	000050	Microwave	Samsung MT479	7499.99
3	000125	Pan	Tefal E20 60	3599.2
4	000125	Refrigerator	Indesit SB200 T	17999.99
5	103320	Teakettle	Bosch TWK189	4890.32
6	434031	Iron	Braun Texstyle 535	5490.01

5. Для произвольной команды SELECT определить список входящих в нее таблиц (через запятую) с указанием имени схемы. Если в команде указан синоним таблицы, то в результате должно быть приведено исходное имя таблицы. Задачу решить одной командой SELECT.

Например, для команды:

WITH "СР ПО ОТД" AS (

SELECT DEPARTMENT_ID,AVG(SALARY) AS ASAL

FROM hr.EMPLOYEES

GROUP BY DEPARTMENT_ID),

"НАИБ БЛИЗ" AS (

SELECT DEPARTMENT_ID, MIN(ABS(SALARY - ASAL)) AS MINSAL

FROM EMPLOYEES JOIN "СР ПО ОТД" USING (DEPARTMENT_ID)

GROUP BY DEPARTMENT_ID)

SELECT EMPLOYEE_ID AS "Номер",LAST_NAME AS "Фамилия",JOB_ID AS "Должность",

DEPARTMENT_ID AS "Отдел",SALARY AS "Оклад", TRUNC(ASAL) AS "Средний оклад"

FROM EMPLOYEES JOIN "MY JOBS" USING (JOB_ID)

JOIN "СР ПО ОТД" USING (DEPARTMENT_ID) JOIN "НАИБ БЛИЗ"

USING (DEPARTMENT_ID)

WHERE (DEPARTMENT_ID, ABS(SALARY - ASAL)) IN

(SELECT DEPARTMENT_ID, MINSAL FROM "НАИБ БЛИЗ")

ORDER BY DEPARTMENT_ID, SALARY, LAST_NAME;

результат должен быть:

hr.EMPLOYEES,os.EMPLOYEES,os."MY JOBS"

Решение 1:

--вводим исходую строку

WITH

input (str) AS (

```
SELECT '&&str'
FROM dual),
--все таблицы, которые фигурируют в строке-запросе
tabs AS
(SELECT
regexp_substr(str,'(".+?")|([[:alnum:]_.]*)',
regexp_instr(str,'[[:space:]](FROM|JOIN)[[:space:]]',1,level,1,'i'),1) AS table_name
FROM input
CONNECT BY level <= regexp_count(str,'[[:space:]](FROM|JOIN)[[:space:]]',1,'i')),
--если имя таблицы с кавычками, то оно регистрочуствительное, иначе - оно
--записано в словарях данных в высоком регистре
tabs_fix AS
(SELECT
(CASE
WHEN instr(tabs.table_name,'"') = 0
AND tabs.table_name NOT LIKE '%.%' THEN upper(tabs.table_name)
ELSE tabs.table_name
END) table_name
FROM tabs),
--проверяем, чем является найденный объект и в зависимсоти от этого решаем, выводить его или нет
tables AS
(SELECT DISTINCT
CASE
WHEN tab.table_name LIKE '%.%'
OR upper(tab.table_name) = 'DUAL' THEN tab.table_name
WHEN syn.synonym_name IS NOT NULL THEN lower((SELECT user
FROM dual)) || '.' || syn.table_name
WHEN TRIM(BOTH '"' FROM tab.table_name) IN (SELECT table_name
FROM user_tables
UNION
SELECT table_name
FROM dictionary) THEN lower((SELECT user
FROM dual)) || '.' || tab.table_name
END table_name
```

```
FROM tabs_fix tab
LEFT JOIN user_synonyms syn ON tab.table_name = syn.synonym_name
ORDER BY TRIM(BOTH '"' FROM table_name)),
result AS
(SELECT table_name, ROWNUM
FROM tables
WHERE table_name IS NOT NULL)

SELECT LISTAGG(table_name, ',') WITHIN GROUP(ORDER BY ROWNUM) таблицы
FROM result;
UNDEFINE str;
```

БИЛЕТ 7

1. Для произвольной команды SELECT определить список входящих в нее таблиц (через запятую) с указанием имени схемы. Задачу решить одной командой SELECT.

Например, для команды:

```
WITH "СР ПО ОТД" AS (
SELECT DEPARTMENT_ID,AVG(SALARY) AS ASAL
FROM hr.EMPLOYEES
GROUP BY DEPARTMENT_ID),
"НАИБ БЛИЗ" AS (
SELECT DEPARTMENT_ID, MIN(ABS(SALARY - ASAL)) AS MINSAL
FROM EMPLOYEES JOIN "СР ПО ОТД" USING (DEPARTMENT_ID)
GROUP BY DEPARTMENT_ID)
SELECT EMPLOYEE_ID AS "Номер",LAST_NAME AS "Фамилия",JOB_ID AS "Должность",
DEPARTMENT_ID AS "Отдел",SALARY AS "Оклад", TRUNC(ASAL) AS "Средний оклад"
FROM EMPLOYEES JOIN "MY JOBS" USING (JOB_ID)
JOIN "СР ПО ОТД" USING (DEPARTMENT_ID) JOIN "НАИБ БЛИЗ"
USING (DEPARTMENT_ID)
WHERE (DEPARTMENT_ID, ABS(SALARY - ASAL)) IN
(SELECT DEPARTMENT_ID, MINSAL FROM "НАИБ БЛИЗ")
ORDER BY DEPARTMENT_ID, SALARY, LAST_NAME;
```

результат должен быть:

hr.EMPLOYEES,os.EMPLOYEES,os."MY JOBS"

Примечание:

Задачу решить в предположении, что объекты, используемые в команде SELECT, существуют и пользователь имеет объектные привилегии на их использование.

РЕШЕНИЕ С ЭКРАНИРОВАНИЕМ

```
--вводим исходую строку
WITH
input (str) AS (
SELECT '&&str'
FROM dual),
--все таблицы, которые фигурируют в строке-запросе
tabs AS
```

```
(SELECT
regexp_substr(str,'(".+?")|([[:alnum:]_.]*)',
regexp_instr(str,'[[:space:]](FROM|JOIN)[[:space:]]',1,level,1,'i'),1) AS table_name
FROM input
CONNECT BY level <= regexp_count(str,'[[:space:]](FROM|JOIN)[[:space:]]',1,'i')),
--если имя таблицы с кавычками, то оно регистрочуствительное, иначе - оно
--записано в словарях данных в высоком регистре
tabs_fix AS
(SELECT
(CASE
WHEN instr(tabs.table_name,'"') = 0
AND tabs.table_name NOT LIKE '%.%' THEN upper(tabs.table_name)
ELSE tabs.table_name
END) table_name
FROM tabs),
--проверяем, чем является найденный объект и в зависимсоти от этого решаем, выводить его или нет
tables AS
(SELECT DISTINCT
CASE
WHEN tab.table_name LIKE '%.%'
OR upper(tab.table_name) = 'DUAL' THEN tab.table_name
WHEN syn.synonym_name IS NOT NULL THEN lower((SELECT user
FROM dual)) || '.' || syn.table_name
WHEN TRIM(BOTH '"' FROM tab.table_name) IN (SELECT table_name
FROM user_tables
UNION
SELECT table_name
FROM dictionary) THEN lower((SELECT user
FROM dual)) || '.' || tab.table_name
END table_name
FROM tabs_fix tab
LEFT JOIN user_synonyms syn ON tab.table_name = syn.synonym_name
ORDER BY TRIM(BOTH '"' FROM table_name)),
result AS
```

```
(SELECT table_name, ROWNUM
FROM tables
WHERE table_name IS NOT NULL)

SELECT LISTAGG(table_name, ',') WITHIN GROUP(ORDER BY ROWNUM) таблицы
FROM result;

undefine str;
```

2. Для каждого месяца, в котором принимались на работу сотрудники, найти 3 ближайших после данного "месяца-двойника".

"Двойниками" считать такие месяцы, которые и начинаются в один и тот же день недели, и заканчиваются в один и тот же день недели.

Выводить:

1. Фамилию сотрудника;

2. Месяц, в котором сотрудник был принят на работу в формате "mon year";

3.-5. Три ближайших "месяца-двойника" в формате "mon year

```
WITH
num (k) AS
   (SELECT 1 AS k
   FROM DUAL
   UNION ALL
   SELECT k+1
   FROM num
   WHERE k <= 2000),
days AS
   (SELECT last_name, hire_date,
   TO_CHAR(trunc(hire_date,'MM'),'DAY') AS first_day,
   TO_CHAR(last_day(hire_date),'DAY') AS last_day
   FROM employees),
twins AS
   (SELECT last_name, hire_date, first_day, last_day, k,
   dense_rank() OVER (PARTITION BY last_name, hire_date ORDER BY k) AS r
   FROM days, num
   WHERE TO_CHAR(trunc(ADD_MONTHS(hire_date, k),'MM'),'DAY') = first_day
```

```
   AND TO_CHAR(last_day(ADD_MONTHS(hire_date, k)),'DAY') = last_day),
result AS
   (SELECT last_name, hire_date,
   ADD_MONTHS(hire_date, k) AS next_d, r
   FROM twins
   WHERE r <= 3
   ORDER BY last_name, r)

SELECT last_name,
TO_CHAR(hire_date, 'mon year', 'NLS_DATE_LANGUAGE = ENGLISH') AS hire_date,
TO_CHAR((SELECT next_d
   FROM result res
   WHERE res.last_name = e.last_name
   AND res.hire_date = e.hire_date
   AND r = 1), 'mon year', 'NLS_DATE_LANGUAGE = ENGLISH') AS "TWIN-MONTH
     1",
TO_CHAR((SELECT next_d
   FROM result res
   WHERE res.last_name = e.last_name
   AND res.hire_date = e.hire_date
   AND r = 2), 'mon year', 'NLS_DATE_LANGUAGE = ENGLISH') AS "TWIN-MONTH
     2",
TO_CHAR((SELECT next_d
   FROM result res
   WHERE res.last_name = e.last_name
   AND res.hire_date = e.hire_date
   AND r = 3), 'mon year', 'NLS_DATE_LANGUAGE = ENGLISH') AS "TWIN-MONTH
     3"
FROM employees e
ORDER BY last_name, first_name;
```

3. Для заданного списка чисел найти все такие его разбиения на два непересекающихся подсписка, что модуль разности сумм чисел в первом и втором подсписке минимально отличаются друг от друга.

Например, для списка

1,-1,1,2,3,6.5

результат должен быть

Исходный список	Подсписок 1	Подсписок 2	Модуль разницы сумм
1,-1,1,2,3,6.5	6.5	-1,1,1,2,3	0,5
	1,2,3	-1,1,6.5	0,5

Результирующие списки должны быть отсортированы по возрастанию. Количество элементов Подсписка 1 должно быть меньше или равно количеству элементов Подсписка 2.

```
WITH
input AS
   (SELECT '1,-1,1,2,3,6.5' AS inpstr
   FROM DUAL),
numbers AS
   (SELECT REGEXP_SUBSTR(inpstr, '[^,]+', 1, LEVEL) AS nums,
      ROW_NUMBER() OVER(ORDER BY TO_NUMBER(REGEXP_SUBSTR(inpstr, '[^,]+',1,LEVEL),'999999.999999')) AS rn
   FROM input
   CONNECT BY REGEXP_SUBSTR(inpstr, '[^,]+', 1, LEVEL) IS NOT NULL),
counts(nums, summ, rn, lvl) AS
   (SELECT nums, TO_NUMBER(nums, '999999.99999') AS summ, rn, 1 AS lvl
   FROM numbers
   UNION ALL
   SELECT cnt.nums || ',' || nmbrs.nums,
   cnt.summ + TO_NUMBER(nmbrs.nums, '999999.99999'), nmbrs.rn, cnt.lvl + 1
   FROM counts cnt, numbers nmbrs
   WHERE cnt.RN < nmbrs.rn),
str AS
   (SELECT cnt1.nums AS n1, cnt2.nums AS n2,
   ABS(cnt1.summ - cnt2.summ) AS diff
   FROM counts cnt1, counts cnt2
   WHERE cnt1.lvl <= cnt2.lvl
   AND cnt1.lvl + cnt2.lvl = (SELECT MAX(rn)
      FROM numbers)
```

```
    AND cnt1.summ + cnt2.summ = (SELECT MAX(summ)
      FROM counts
      WHERE lvl = (SELECT MAX(rn)
        FROM numbers))),
rightstr AS
  (SELECT DISTINCT n1, n2, diff
  FROM str
  WHERE diff = (SELECT MIN(diff)
    FROM str)),
result AS
  (SELECT n1, n2, diff, LEVEL,
  ROW_NUMBER() OVER(PARTITION BY LEVEL ORDER BY LEVEL) AS rnum
  FROM (SELECT n1, n2, diff, ROWNUM rn
    FROM rightstr)
  CONNECT BY NOCYCLE PRIOR N1 = N2)

SELECT
CASE
  WHEN ROWNUM = 1 THEN (SELECT inpstr FROM input)
  ELSE ' '
END AS "Исходный список",
n1 AS "Подсписок 1", n2 AS "Подсписок 2",
diff AS "Модуль разницы сумм"
FROM result
WHERE RNUM = 1;
```

4. Создтть запрос для определения среди таблиц Вашей схемы таких таблиц, названия которых получаются друг из друга циклическим сдвигом символов.

Пример результата:

Номер	**Таблица 1**	**Таблица 2**
1	BAA	ABA
2	AAB	ABA

3	ABA	BAA
4	BAA	AAB
5	ABA	AAB
6	AAB	BAA

```
CREATE TABLE BAA(NUM NUMBER(2));
CREATE TABLE ABA(NUM NUMBER(2));
CREATE TABLE AAB(NUM NUMBER(2));
WITH TAB AS (
SELECT table_name
FROM user_tables )
SELECT ROWNUM AS "Номер", T1.table_name AS "Таблица 1", T2.table_name AS
    "Таблица 2"
FROM TAB T1 INNER JOIN TAB T2
ON (LENGTH(T1.table_name) = LENGTH(T2.table_name))
WHERE T1.table_name != T2.table_name AND
INSTR(T2.table_name||T2.table_name, T1.table_name) != 0;
```

	Номер	Таблица 1	Таблица 2
1	1	BAA	AAB
2	2	ABA	AAB
3	3	BAA	ABA
4	4	AAB	ABA
5	5	ABA	BAA
6	6	AAB	BAA

5. Определить список последовательностей подчиненности от преподавателей, не имеющих начальника, до преподавателей, не имеющих подчиненных.

Если список состоит более, чем из четырех фамилий, то выводить только две первые и две последние фамилии, а вместо остальных фамилий поставить многоточие.

Костыркин-> Викулина-> …->Соколов->Казанко (не имеет подчиненных)

```
WITH temp AS (SELECT substr(teachers, 3, length(teachers)-2) || '(не имеет
подчиненных)' str ,lvl
FROM
(
   SELECT SYS_CONNECT_BY_PATH(фамилия,'->') as teachers, connect_by_isleaf as
isleaf,level lvl
```

```
   FROM Преподаватели
   START WITH подчиняется IS NULL
   CONNECT BY PRIOR номер_преподавателя = подчиняется
)
WHERE ISLEAF = 1)
SELECT
   CASE WHEN lvl>4
    THEN REPLACE(str,substr(str,instr(str,'->',1,2)+2,instr(str,'->',1,lvl-2)-instr(str,'->',1,1)-2),'...')
   ELSE str
   END AS "Список подчиненностей"
FROM temp;
```

БИЛЕТ 8

1. Используя словарь данных, получить информацию о первичных ключах и подчиненных таблицах всех таблиц Вашей схемы:

Имя таблицы	Список столбцов первичного ключа	Список столбцов с ограничением уникальности	Список подчиненных таблиц

В списках имена столбцов первичного ключа и таблиц вывести через запятую по алфавиту. Если таблица не имеет подчиненных таблиц, вывести – Подчиненных таблиц не имеет. Если таблица не имеет первичного ключа, вывести – Первичного ключа не имеет. Аналогично, если таблица не имеет ограничения уникальности, вывести – Ограничения уникальности в таблице нет. Имена столбцов композитных ограничений уникальности заключить в круглые скобки.

Задачу решить без использования функций Listagg и Wm_concat.

Решение 1:

```
WITH
t AS
   (SELECT ut.table_name, ucc.column_name,
         ROW_NUMBER() OVER(PARTITION BY ut.table_name ORDER BY ucc.column_name) num
   FROM user_tables ut
   LEFT JOIN user_constraints uc
   ON ut.table_name = uc.table_name
```

```
   LEFT JOIN user_cons_columns ucc
   ON uc.constraint_name = ucc.constraint_name
   AND uc.constraint_type = 'P'),
t1 AS
   (SELECT table_name,
    TRIM(',' FROM regexp_replace(substr(sys_connect_by_path(column_name,','),2),',,',',') )
pcol
   FROM t
   WHERE CONNECT_BY_ISLEAF = 1
   START WITH num = 1
   CONNECT BY NOCYCLE PRIOR num + 1 = num
   AND PRIOR table_name = table_name),
t2 AS
   (SELECT ut.table_name, ucc.table_name tn,
           ROW_NUMBER() OVER(PARTITION BY ut.table_name ORDER BY
ucc.table_name) num
   FROM user_tables ut
   LEFT JOIN user_constraints uc
   ON ut.table_name = uc.table_name
   AND uc.constraint_type = 'R'
   LEFT JOIN user_cons_columns ucc
   ON uc.r_constraint_name = ucc.constraint_name),
t3 AS
   (SELECT table_name,
   TRIM(',' FROM regexp_replace(substr(sys_connect_by_path(tn,','),2),',,',',') ) rtab
   FROM t2
   WHERE CONNECT_BY_ISLEAF = 1
   START WITH num = 1
   CONNECT BY NOCYCLE PRIOR num + 1 = num
   AND PRIOR table_name = table_name)

SELECT table_name "Имя таблицы",
nvl(pcol,'Первичных ключей нет') "Список столб. первичного ключа",
nvl(rtab,'Подчиненных таблиц нет') "Список подчининных таблиц"
FROM t3
```

JOIN t1 USING (table_name);

	Имя таблицы	Столбцы первичного ключа	Список ст-цов с огран. уник	Список подчиненных таблиц
1	BOOK	Первичного ключа не имеет	Ограничения уникальности в таблице нет	Подчиненных таблиц не имеет
2	CATALOG	Первичного ключа не имеет	Ограничения уникальности в таблице нет	Подчиненных таблиц не имеет
3	COUNTRIES	COUNTRY_ID	Ограничения уникальности в таблице нет	LOCATIONS
4	DEPARTMENTS	DEPARTMENT_ID	Ограничения уникальности в таблице нет	EMPLOYEES, JOB_HISTORY

2. Задана таблица со столбцами Номер – Number и Сумма – Numданаber. Положительное значение во втором столбце обозначает сумму, которая пришла на счет, а отрицательное значение – корректировка (уменьшение) предыдущих поступлений. Требуется написать запрос, который определит суммы с учетом корректировок.

Например, для таблицы

Номер	**Сумма**
1	100
2	300
3	200
4	100
5	-350
6	100
8	100
9	-300
1	800
1	-600

Т.е. строка под номером 5 должна отменить сумму в строке 4, 3 и часть суммы из строки 2 и т.д.

Задачу решить с использованием раздела Model.

Результат должен быть:

Номер	**Сумма**	**Итог**

1	100	100
2	300	150
3	200	0
4	100	0
5	-350	0
6	100	0
8	100	0
9	-300	0
1	800	200
1	-600	0

Примечание: Столбец Номер содержит уникальные значение, но пропуски значений возможны.

Решение 1:

```
CREATE TABLE t (
n number(10,0),
summ number(10,2),
CONSTRAINT t PRIMARY KEY (n)
);


INSERT INTO t VALUES (1,100);
INSERT INTO t VALUES (2,300);
INSERT INTO t VALUES (3,200);
INSERT INTO t VALUES (4,100);
INSERT INTO t VALUES (5,-350);
INSERT INTO t VALUES (6,100);
INSERT INTO t VALUES (8,100);
INSERT INTO t VALUES (9,-300);
INSERT INTO t VALUES (10,800);
```

```
INSERT INTO t VALUES (11,-600);

WITH
tab0 AS
   (SELECT n, summ
   FROM t
   ORDER BY n),
tab1 AS
   (SELECT ROWNUM rn, n, summ
   FROM tab0),
tab2 (n, r, s, ost, results) AS
   (SELECT n, rn r, summ s,
   CASE
      WHEN REGEXP_LIKE(summ, '^(\d)+$') THEN 0
      ELSE summ
   END ost,
   CASE
      WHEN REGEXP_LIKE (summ, '^(\d)+$') THEN summ
      ELSE 0
   END results
   FROM tab1
   WHERE rn = (SELECT MAX(rn)
      FROM tab1)
   UNION ALL
   SELECT tab1.n, r - 1, summ,
   CASE
      WHEN summ + ost < 0 THEN summ + ost
      ELSE 0
   END,
   CASE
      WHEN summ + ost > 0   THEN summ + ost
      ELSE 0
   END
   FROM tab2 JOIN tab1
   ON tab2.r - 1 = tab1.rn
   WHERE r IS NOT NULL)
SELECT DISTINCT n "Номер", s "Сумма", results "Итог"
```

FROM tab2

ORDER BY n ASC;

3. Создать запрос для получения информации об успеваемости студентов в виде:

ФИО	Дисциплина	Оценка	Дата	Примечания
Петров	Математика	5	20.1.2008	
	Физика	4	22.1.2008	
	Химия	2	25.1.2008	
	Химия	3	27.1.2008	Пересдача
Усов	Математика	5	12.06.99	
	Экономика	3	15.06.99	
	Менеджмент	2	17.06.99	
	Менеджмент	4	18.06.99	Пересдача
Судаков	Экзамены не сдавал			

В таблице должна быть представлена информация только по результатам сдачи экзаменов по дисциплинам, предусмотренным учебным планом для специальности, на которой учится студент.

НАДО СОЗДАТЬ СВОЮ ТАБЛИЦУ, ГДЕ ЕСТЬ “ПЕРЕСДАЧА” В ПРИМЕЧАНИЯХ И ЛЮДИ, КОТОРЫЕ НЕ СДАВАЛИ ЭКЗАМЕНЫ И ПОСМОТРЕТЬ ЧЕ БУДЕТ

Решение 1:

SELECT CASE WHEN rw = 1 THEN фамилия

ELSE ' ' END AS "ФИО", дисциплина AS "Дисциплина", оценка AS "Оценка", дата AS "Дата",

CASE WHEN примечания IS NULL THEN ' '

ELSE примечания END AS "Примечания"

FROM (SELECT фамилия, ROW_NUMBER() OVER(PARTITION BY

студенты.номер_студента ORDER BY фамилия) rw,

название дисциплина, оценка, дата,

CASE WHEN ROW_NUMBER() OVER(PARTITION BY студенты.фамилия, дисциплины.название ORDER BY студенты."НОМЕР_СТУДЕНТА") = 2 THEN 'ПЕРЕСДАЧА'

ELSE TO_CHAR(NULL) END примечания

FROM студенты JOIN (SELECT *

FROM успеваемость

ORDER BY дата) x ON студенты."НОМЕР_СТУДЕНТА" = x."НОМЕР_СТУДЕНТА"

JOIN дисциплины ON x."НОМЕР_ДИСЦИПЛИНЫ" = дисциплины."НОМЕР_ДИСЦИПЛИНЫ");

-- в подзапросе соединяю таблицы и вычисляю номера строк, в основном запросе далее отображаю фамилию только для первых строк

4. Имеется таблица со столбцами Номер, Строка. Тип данных столбца Номер - Integer, тип данных столбца Строка – Varchar2(10). Первый столбец содержит порядковый номер записи, столбец Строка – символьные строки, состоящие из 0 и 1. Общее количество цифр во всех строках – одинаковое и равно 5. Написать запрос, который выведет номера максимального количества строк и позиции столбца Строка, образующие квадратную матрицу, состоящую только из единиц.

Например, для таблицы:

Номер	Строка
1	00101
2	10011
3	10101

ответ должен быть

Строки	Столбцы
1,3	3,5
2,3	1,5

Решение 1:

CREATE TABLE zad4 (id INTEGER CONSTRAINT pk4_pk PRIMARY KEY, "String" VARCHAR2(10));

```
INSERT INTO zad4 VALUES (1, '00101');
INSERT INTO zad4 VALUES (2, '10011');
INSERT INTO zad4 VALUES (3, '10101');

WITH
postol AS (
        SELECT id, "String", substr("String",1,1) AS a1, substr("String",2,1) AS a2, substr("String",3,1) AS a3,
        substr("String",4,1) AS a4, substr("String",5,1) AS a5
    FROM zad4),
perevorot AS (
    SELECT TO_CHAR(id) AS idd, TO_CHAR(colname) AS a
    FROM postol UNPIVOT(znach FOR colname IN ( a1 AS 1, a2 AS 2, a3 AS 3, a4 AS 4, a5 AS 5))
    WHERE znach = 1),
per2 (idd2, idd, digit) AS (
    SELECT idd AS idd2, idd, a AS digit
    FROM perevorot
    UNION ALL
    SELECT perevorot.idd, per2.idd || ',' || perevorot.idd, a
     FROM per2 INNER JOIN perevorot ON perevorot.idd > per2.idd2 AND perevorot.a = per2.digit),
per3 (len, idd, digit, a) AS (
    SELECT length(idd) AS len, idd, digit, digit AS a
    FROM per2
    UNION ALL
    SELECT length(per2.idd), per2.idd, per2.digit, per3.a || ',' || per2.digit
    FROM per3 INNER JOIN per2 ON per2.digit > per3.digit AND per2.idd = per3.idd),
matrs AS (
    SELECT len, idd, a
    FROM per3
    WHERE length(a) = len)
SELECT idd AS "Rows", a AS "Columns"
FROM matrs
WHERE len = (SELECT MAX(len)
            FROM matrs)
```

ORDER BY 1, 2;

5. Для каждого месяца, в котором принимались на работу сотрудники, найти 3 ближайших после данного "месяца-двойника".

"Двойниками" считать такие месяцы, которые и начинаются в один и тот же день недели, и заканчиваются в один и тот же день недели.

Выводить:

1. Фамилию сотрудника;

2. Месяц, в котором сотрудник был принят на работу в формате "Mon YYYY";

3.-5. Три ближайших "месяца-двойника" в формате "Mon YYYY"

Пример результата:

Last_Name	Hire_Date	Month 1	Month 2	Month 3
Bissot	Авг 2005	Май 2006	Янв 2007	Окт 2007
…….				

```
WITH
num (k) AS
   (SELECT 1 AS k
   FROM DUAL
   UNION ALL
   SELECT k+1
   FROM num
   WHERE k <= 2000),
days AS
   (SELECT last_name, hire_date,
   TO_CHAR(trunc(hire_date,'MM'),'DAY') AS first_day,
   TO_CHAR(last_day(hire_date),'DAY') AS last_day
   FROM employees),
twins AS
   (SELECT last_name, hire_date, first_day, last_day, k,
   dense_rank() OVER (PARTITION BY last_name, hire_date ORDER BY k) AS r
   FROM days, num
   WHERE TO_CHAR(trunc(ADD_MONTHS(hire_date, k),'MM'),'DAY') = first_day
```

```
   AND TO_CHAR(last_day(ADD_MONTHS(hire_date, k)),'DAY') = last_day),
result AS
   (SELECT last_name, hire_date,
   ADD_MONTHS(hire_date, k) AS next_d, r
   FROM twins
   WHERE r <= 3
   ORDER BY last_name, r)

SELECT last_name,
TO_CHAR(hire_date, 'mon year', 'NLS_DATE_LANGUAGE = ENGLISH') AS hire_date,
TO_CHAR((SELECT next_d
   FROM result res
   WHERE res.last_name = e.last_name
   AND res.hire_date = e.hire_date
   AND r = 1), 'mon year', 'NLS_DATE_LANGUAGE = ENGLISH') AS "TWIN-MONTH 1",
TO_CHAR((SELECT next_d
   FROM result res
   WHERE res.last_name = e.last_name
   AND res.hire_date = e.hire_date
   AND r = 2), 'mon year', 'NLS_DATE_LANGUAGE = ENGLISH') AS "TWIN-MONTH 2",
TO_CHAR((SELECT next_d
   FROM result res
   WHERE res.last_name = e.last_name
   AND res.hire_date = e.hire_date
   AND r = 3), 'mon year', 'NLS_DATE_LANGUAGE = ENGLISH') AS "TWIN-MONTH 3"
FROM employees e
ORDER BY last_name, first_name;
```

БИЛЕТ 9

1. Для произвольной строки, состоящей из открывающих и закрывающих скобок написать запрос для вывода всех слов максимальной длины, представляющих правильные скобочные записи. Например, для строки (()(() ответ должен быть:

()()

(())

Решение 1:

```
WITH input AS (SELECT '(()(()(' AS str, '' AS s
            FROM dual),
    tmp1 AS (SELECT str, level AS lvl, substr(str,level,1) AS c
            FROM input
            CONNECT BY level <= length(str)),
    tmp2 AS (SELECT DISTINCT str, replace(sys_connect_by_path(c,'.'),'.','') s
            FROM tmp1
            CONNECT BY PRIOR str = str AND PRIOR lvl < lvl AND PRIOR sys_guid() IS
NOT NULL),
    tmp4 (s, lvl, c) AS (SELECT s, 1 AS lvl, 1 AS c
                    FROM tmp2
                    WHERE s LIKE '(%' AND regexp_count(s,'\(') = regexp_count(s,'\)')
                    UNION ALL
                    SELECT s, lvl + 1, CASE WHEN substr(s,lvl + 1,1) = '('
                                    THEN c + 1
                                    ELSE c - 1
                                END AS c
                    FROM tmp4
                    WHERE lvl < 1 + length(s) AND c >= 0),
     tmp5 AS (SELECT *
            FROM tmp4
            WHERE lvl = length(s))
SELECT s AS result
FROM tmp5
WHERE c = 0 AND   length(s) = (SELECT MAX(length(s) )
                        FROM tmp5
                        WHERE c = 0);
```

	RESULT
1	() ()
2	(( ))

2. Имеется таблица D_V с первым столбцом Dat типа DATE (первичный ключ) и вторым столбцом Val типа NUMBER. Пример (строки упорядочены по первому столбцу):

DAT VAL

01-08-08 232

02-08-08

10-08-08 182

11-08-08

21-08-08 240

22-08-08

23-08-08

Требуется написать запрос для получения на основе таблицы D_V следующей таблицы:

DAT MAX_VAL

01-08-08 232

02-08-08 232

10-08-08 182

11-08-08 182

21-08-08 240

22-08-08 240

23-08-08 240

Данная результирующая таблица должна быть упорядочена по Dat, но вместо пустых значений, которые присутствовали в столбце VAL отсортированной по DAT исходной таблицы, в столбце MAX_VAL результирующей таблицы, должны присутствовать значения столбца из предыдущей строки.

Задачу решить без использования аналитических функций.

Решение 1:

```
CREATE TABLE D_V (dat DATE PRIMARY KEY, val NUMBER);


INSERT INTO D_V VALUES('01-08-08', 232);
INSERT INTO D_V VALUES('02-08-08', NULL);
INSERT INTO D_V VALUES('10-08-08', 182);
INSERT INTO D_V VALUES('11-08-08', NULL);
INSERT INTO D_V VALUES('21-08-08', 240);
```

```
INSERT INTO D_V VALUES('22-08-08', NULL);
INSERT INTO D_V VALUES('23-08-08', NULL);


SELECT dat, CASE WHEN val IS NULL
            THEN (SELECT tmp1.val
                FROM D_V tmp1
                WHERE tmp1.dat = (SELECT MAX(dat)
                                FROM D_V
                                WHERE dat < tmp2.dat AND val IS NOT NULL))
            ELSE val END AS max_val
FROM D_V tmp2
ORDER BY dat;
```

	DAT	MAX_VAL
1	01.08.08	232
2	02.08.08	232
3	10.08.08	182
4	11.08.08	182
5	21.08.08	240
6	22.08.08	240
7	23.08.08	240

3. Для произвольной команды SELECT определить список входящих в нее таблиц (через запятую) с указанием имени схемы. Если в команде указан синоним таблицы, то в результате должно быть приведено исходное имя таблицы. Задачу решить одной командой SELECT.

Например, для команды:

```
WITH "СР ПО ОТД" AS (
SELECT DEPARTMENT_ID,AVG(SALARY) AS ASAL
FROM hr.EMPLOYEES
GROUP BY DEPARTMENT_ID),
"НАИБ БЛИЗ" AS (
SELECT DEPARTMENT_ID, MIN(ABS(SALARY - ASAL)) AS MINSAL
FROM EMPLOYEES JOIN "СР ПО ОТД" USING (DEPARTMENT_ID)
GROUP BY DEPARTMENT_ID)
SELECT EMPLOYEE_ID AS "Номер",LAST_NAME AS "Фамилия",JOB_ID AS "Должность",
DEPARTMENT_ID AS "Отдел",SALARY AS "Оклад", TRUNC(ASAL) AS "Средний оклад"
FROM EMPLOYEES JOIN "MY JOBS" USING (JOB_ID)
```

JOIN "СР ПО ОТД" USING (DEPARTMENT_ID) JOIN "НАИБ БЛИЗ"

USING (DEPARTMENT_ID)

WHERE (DEPARTMENT_ID, ABS(SALARY - ASAL)) IN

(SELECT DEPARTMENT_ID, MINSAL FROM "НАИБ БЛИЗ")

ORDER BY DEPARTMENT_ID, SALARY, LAST_NAME;

результат должен быть:

hr.EMPLOYEES,os.EMPLOYEES,os."MY JOBS"

Решение 1:

```
--вводим исходую строку
WITH
input (str) AS (
SELECT '&&str'
FROM dual),
--все таблицы, которые фигурируют в строке-запросе
tabs AS
(SELECT
regexp_substr(str,'(".+?")|([[:alnum:]_.]*)',
regexp_instr(str,'[[:space:]](FROM|JOIN)[[:space:]]',1,level,1,'i'),1) AS table_name
FROM input
CONNECT BY level <= regexp_count(str,'[[:space:]](FROM|JOIN)[[:space:]]',1,'i')),
--если имя таблицы с кавычками, то оно регистрочуствительное, иначе - оно
--записано в словарях данных в высоком регистре
tabs_fix AS
(SELECT
(CASE
WHEN instr(tabs.table_name,'"') = 0
AND tabs.table_name NOT LIKE '%.%' THEN upper(tabs.table_name)
ELSE tabs.table_name
END) table_name
FROM tabs),
--проверяем, чем является найденный объект и в зависимсоти от этого решаем, выводить его или нет
tables AS
(SELECT DISTINCT
CASE
```

```
WHEN tab.table_name LIKE '%.%'
OR upper(tab.table_name) = 'DUAL' THEN tab.table_name
WHEN syn.synonym_name IS NOT NULL THEN lower((SELECT user
FROM dual)) || '.' || syn.table_name
WHEN TRIM(BOTH '"' FROM tab.table_name) IN (SELECT table_name
FROM user_tables
UNION
SELECT table_name
FROM dictionary) THEN lower((SELECT user
FROM dual)) || '.' || tab.table_name
END table_name
FROM tabs_fix tab
LEFT JOIN user_synonyms syn ON tab.table_name = syn.synonym_name
ORDER BY TRIM(BOTH '"' FROM table_name)),
result AS
(SELECT table_name, ROWNUM
FROM tables
WHERE table_name IS NOT NULL)

SELECT LISTAGG(table_name, ',') WITHIN GROUP(ORDER BY ROWNUM) таблицы
FROM result;


undefine str;
```

4. Имеется таблица с символьным столбцом. Создать запрос для вывода тех значений, которые содержат в себе палиндромы, и самые длинные выражения, представляющие из себя палиндром.

Например, для таблицы с данными:

Text
Крокодил
Колокол
Станок

Результат должен быть:

Text	**Palindrom**
Крокодил	око

Колокол	Колок, локол

СРОЧНО ТЕСТИТЬ!!!

```
CREATE TABLE tab13(TEXT VARCHAR2(50)CONSTRAINT zad13_pk PRIMARY KEY);


INSERT INTO tab13 VALUES('Крокодил');
INSERT INTO tab13 VALUES('Колокол');
INSERT INTO tab13 VALUES('Крокодил и колокол');
INSERT INTO tab13 VALUES('Станок');
INSERT INTO tab13 VALUES('  Крокодил  ');
INSERT INTO tab13 VALUES('  ');
INSERT INTO tab13 VALUES('Крокодил и  колокол');


WITH REVERSE_STR AS (SELECT text as inptext, TRIM(both ' ' from replace(replace(replace(text, ' ', ' _'), '_ '), '_')) as TEXT,
REVERSE(TRIM(both ' ' from replace(replace(replace(text, ' ', ' _'), '_ '), '_'))) AS REV_TEXT,
LENGTH(TEXT) AS LEN,
ROWNUM AS RN
FROM tab13),


PALINDROM AS (SELECT RN, inptext, TEXT,
CASE
WHEN SUBSTR(LOWER(TEXT), L1 - L2, L2) =
SUBSTR(LOWER(REV_TEXT), LENGTH(REV_TEXT) - L1 + 1 - L2, L2)
THEN SUBSTR(TEXT, L1 - L2, 2 * L2 + 1)
WHEN SUBSTR(LOWER(TEXT), L1 - L3, L3 + 1) =
SUBSTR(LOWER(REV_TEXT), LENGTH(REV_TEXT) - L1 - L3, L3 + 1)
THEN SUBSTR(TEXT, L1 - L3, (L3 + 1) * 2)
END STR
FROM REVERSE_STR A
INNER JOIN (SELECT ROWNUM L1 FROM USER_OBJECTS) B
ON(A.LEN >= B.L1)
INNER JOIN (SELECT ROWNUM L2 FROM USER_OBJECTS) C
```

```
ON(B.L1 - C.L2 > 0 AND B.L1 + C.L2 <= A.LEN)
INNER JOIN (SELECT ROWNUM - 1 L3 FROM USER_OBJECTS) D
ON(B.L1 - D.L3 > 0 AND B.L1 + D.L3 <= A.LEN)
ORDER BY RN, L1),
result AS (
SELECT DISTINCT RN, inptext, TEXT, str
FROM PALINDROM pr
WHERE STR IS NOT NULL
AND length(str) = ANY(SELECT MAX(length(str))
FROM palindrom
WHERE rn = pr.rn
GROUP BY rn)
ORDER BY RN, TEXT)

SELECT DISTINCT inptext, LISTAGG(str, ', ')
WITHIN GROUP (ORDER BY str)
OVER (PARTITION BY rn) AS "Palindrom"
FROM result
ORDER BY inptext;
```

5. Создать таблицу Города, в которой хранятся названия городов и расстояния между ними. Названия городов уникальны. Пример заполнения таблицы:

Город отправ	**Город назнач**	**Расстояние**
Москва	Казань	2000
Москва	Тула	200
Казань	Вологда	800
……	……	……

Написать команду SELECT, которая определит все пути и расстояния между двумя городами, имена которых задаются как параметры. Путь выводить в виде списка, например, Москва – Казань – Вологда. Расстояния в прямую и обратную сторону могут различаться.

ВЫВОДИТ ПУТЬ ТОЛЬКО В ОДНУ СТОРОНУ

```
CREATE TABLE goroda
(start_city   VARCHAR2(20),
end_city     VARCHAR2(20),
distance     NUMBER(6),
CONSTRAINT se_pk PRIMARY KEY ( start_city, end_city ));

INSERT INTO goroda VALUES ('Москва','Казань',2000);
INSERT INTO goroda VALUES ('Москва','Тула',200);
INSERT INTO goroda VALUES ('Казань','Вологда',800);
INSERT INTO goroda VALUES ('Москва','Санкт-Петербург',700);
INSERT INTO goroda VALUES ('Санкт-Петербург','Москва',735);
INSERT INTO goroda VALUES ('Москва','Иркутск',5200);
INSERT INTO goroda VALUES ('Санкт-Петербург','Иркутск',5777);
INSERT INTO goroda VALUES ('Казань','Москва',2000);

DEFINE city1 = '&&CITY1';
DEFINE city2 = '&&CITY2';

UNDEFINE city1;
UNDEFINE city2;

WITH
recursive (end_city, way, distance) AS
   (SELECT end_city, start_city || '*' || end_city, distance
   FROM goroda
   WHERE start_city = initcap('&CITY1')
   UNION ALL
   SELECT cp.end_city, rec.way || '*'|| cp.end_city, cp.distance + rec.distance
   FROM goroda cp
   INNER JOIN recursive rec
   ON rec.end_city = cp.start_city )
   CYCLE end_city SET cyclemark TO 'X' DEFAULT '-',
t1 AS
   (SELECT ROWNUM rn, '*' || way || '*' AS way, distance AS "Расстояние"
```

```
   FROM recursive
   WHERE substr(way,1,instr(way,'*',1) - 1) = initcap('&CITY1')
   AND   substr(way,instr(way,'*',-1) + 1) = initcap('&CITY2')),
t2 AS
   (SELECT rn,
    substr(way,instr(way,'*',1,level) + 1,instr(way,'*',1,level + 1) - instr(way,'*',1,level) - 1)
AS way
   FROM t1
   CONNECT BY level <= length(way) - length(replace(way,'*') )),
t3 AS
   (SELECT rn, COUNT(*) AS co
   FROM (SELECT DISTINCT rn, way
      FROM t2
      WHERE way IS NOT NULL)
   GROUP BY rn),
t4 AS
   (SELECT
   CASE
   WHEN ( length(way) - length(replace(way,'*') ) ) = co + 1 THEN way
   END way, "Расстояние"
   FROM t1 INNER JOIN t3
   ON ( t1.rn = t3.rn ))

SELECT substr(replace(way,'*','-'),2,length(way) - 2) AS "Путь", "Расстояние"
FROM t4
WHERE way IS NOT NULL;
```

БИЛЕТ 10

1. Вывести информацию о таблицах схемы в виде:

Имя таблицы	Столбцы первичного ключа	Столбцы с ограничением уникальности	Список подчиненных таблиц со столбцами вторичных ключей
Table 1	Col1, Col2	Col3, Col4	Table2 (Col5, Col6), Table3 (Col7,Col8)

Пример результата:

Имя таблицы	Столбцы первичного ключа	Столбцы с ограничением уникальности	Список подчиненных таблиц со столбцами вторичных ключей
DEPT3	DEPARTMENT_ID, DEPARTMENT_NAME	-	DEPT2(ID,NAME)
EMPLOYEES	EMPLOYEE_ID	EMAIL	DEPARTMENTS(MANAGER_ID), EMPLOYEES(MANAGER_ID), JOB_HISTORY(EMPLOYEE_ID)
EMPLOYEES	EMPLOYEE_ID	FIRST_NAME, LAST_NAME	DEPARTMENTS(MANAGER_ID), EMPLOYEES(MANAGER_ID), JOB_HISTORY(EMPLOYEE_ID)
…	…	…	…

```
WITH
src AS
  (SELECT all_t.table_name supp, c2.table_name cust,
      LAG(c2.table_name) OVER(PARTITION BY all_t.table_name ORDER BY all_t.table_name) one_more
  FROM user_constraints c1
  INNER JOIN user_constraints c2
  ON c2.r_constraint_name = c1.constraint_name
  RIGHT JOIN user_tables all_t
```

```
    ON all_t.table_name = c1.table_name),
t AS
    (SELECT supp, TRIM(LEADING ',' FROM(sys_connect_by_path(cust,',') ) ) AS listing,
      level AS l
    FROM src
    START WITH one_more IS NULL
    CONNECT BY NOCYCLE PRIOR cust = one_more
    AND PRIOR supp = supp),
res AS
    (SELECT supp, listing
    FROM t m
    WHERE l = (SELECT MAX(l)
       FROM t
       WHERE supp = m.supp
       GROUP BY supp)),
p_col AS
    (SELECT c3.table_name sys_table, column_name pk_col, LAG(column_name)
    OVER(PARTITION BY c3.table_name ORDER BY c3.table_name) col_col
    FROM user_constraints c3 JOIN user_cons_columns c4
    ON c3.constraint_name = c4.constraint_name
    WHERE c3.constraint_type = 'P'),
u_col AS
    (SELECT c3.table_name sys_table, column_name u_col, LAG(column_name)
    OVER(PARTITION BY c3.table_name ORDER BY c3.table_name) ucol_col
    FROM user_constraints c3 JOIN user_cons_columns c4
    ON c3.constraint_name = c4.constraint_name
    WHERE c3.constraint_type = 'U'),
p_col2 AS
    (SELECT sys_table,
    TRIM(LEADING ',' FROM(sys_connect_by_path(pk_col,',') ) ) listing_col,
    level lev
    FROM p_col
    START WITH col_col IS NULL
    CONNECT BY NOCYCLE PRIOR pk_col = col_col
    AND PRIOR sys_table = sys_table),
```

```
u_col2 AS
   (SELECT sys_table,
   TRIM(LEADING ',' FROM(sys_connect_by_path(u_col,',') ) ) ulisting_col, level lev
   FROM u_col
   START WITH ucol_col IS NULL
   CONNECT BY NOCYCLE PRIOR u_col = ucol_col
   AND PRIOR sys_table = sys_table),
p_col3 AS
   (SELECT sys_table, listing_col
   FROM p_col2 l
   WHERE lev = (SELECT MAX(lev)
      FROM p_col2
      WHERE sys_table = l.sys_table
      GROUP BY sys_table)),
u_col3 AS
   (SELECT sys_table, ulisting_col
   FROM u_col2 l
   WHERE lev = (SELECT MAX(lev)
      FROM u_col2
      WHERE sys_table = l.sys_table
      GROUP BY sys_table))
SELECT DISTINCT supp AS "Имя таблицы",
nvl(listing_col,'-') AS "Столбцы первич. ключа",
nvl(ulisting_col,'-') AS "Столбцы с огр. уникальности",
nvl(listing,'-') AS "Список подч. таблиц"
FROM res r
LEFT JOIN p_col3 p
ON r.supp = p.sys_table
LEFT JOIN u_col3 u
ON r.supp = u.sys_table;
```

2. Имеется произвольный набор косточек домино. Информация о них представлена в виде символьной строки, состоящей из четного числа цифр от 0 до 6. Цифры разделены запятыми. Цифры, находящиеся на соседних нечетном и четном местах относятся к одной косточке. Создать запрос для определения самых длинных последовательностей, которые можно составить из заданного набора.

Результат для каждой последовательности должен быть представлен в виде символьной строки.

```
WITH
tab (str) AS (SELECT '2,5,2,2,4,2,3,1,6,5,2,1'
   FROM dual),
tab2 (str) AS (SELECT substr(str,instr(',' || str
   || ',',',',1,2 * level - 1),3)
   FROM tab
   CONNECT BY level <= ( regexp_count(str,',') + 1 ) / 2),
tab3 (str, r) AS
   ((SELECT str, ROWNUM
   FROM tab2)
   UNION
   (SELECT substr(str,3,1) || ',' || substr(str,1,1), ROWNUM
   FROM tab2)),
tab4 (str) AS
   (SELECT sys_connect_by_path(str,'  ')
   FROM tab3
   CONNECT BY NOCYCLE PRIOR r <> r
   AND (substr(str,1,1) = substr(PRIOR str,3,1)
   AND substr(PRIOR str,1,1) <> substr(PRIOR str,3,1)))

SELECT str
FROM tab4
WHERE length(str) = (SELECT MAX(length(str) )
   FROM tab4);
```

3. Имеется таблица с двумя столбцами – дочерняя вершина и родительская вершина. Определить наборы вершин, образующих связанные множества. Например, для таблицы:

Дочерняя вершина	**Родительская вершина**
1	2
2	4

4	5
4	3
7	6

результат должен быть

Связанные множества
1,2,3,4,5
6,7

```
CREATE TABLE tab1
(child_number    NUMBER(7,0),
parent_number   NUMBER(7,0));

INSERT INTO tab1 VALUES (1,2);
INSERT INTO tab1 VALUES (2,4);
INSERT INTO tab1 VALUES (4,5);
INSERT INTO tab1 VALUES (4,3);
INSERT INTO tab1 VALUES (6,7);
INSERT INTO tab1 VALUES (6,17);
INSERT INTO tab1 VALUES (5,17);

WITH tmp AS
   (SELECT ch, ch || sys_connect_by_path(parent_number,',') str
   FROM (SELECT *
      FROM (SELECT CONNECT_BY_ROOT ( child_number ) ch,
         parent_number
         FROM tab1
         CONNECT BY PRIOR parent_number = child_number
         ORDER BY ch)
      WHERE ch NOT IN (SELECT parent_number
         FROM (SELECT CONNECT_BY_ROOT ( child_number ) ch,
            parent_number
```

```
        FROM tab1
        CONNECT BY PRIOR parent_number = child_number
        ORDER BY ch)))
    CONNECT BY PRIOR ch = ch
    AND PRIOR parent_number < parent_number)

SELECT str AS "Связанные множества"
FROM tmp
WHERE (ch, length(str)) IN (SELECT ch, MAX(length(str))
    FROM tmp
    GROUP BY ch);
```

4. Для произвольной строки, состоящей из чисел, разделенных указанным разделителем, получить строку, отображающую эти числа в обратном порядке. Например, для исходной строки:

0|0|1.45|2|1|2|10|22|34|15|0|-105|66|73

должна быть получена строка:

73|66|-105|0|15|34|22|10|2|1|2|1.45|0|0.

Задачу решить без использования иерархических запросов.

```
WITH
sour AS
    (SELECT '0|0|1|2|1|-2|10|22|34|15|0|-105|66|73' AS str
    FROM dual)

SELECT str AS "Строка",
LISTAGG(rs,'|') WITHIN GROUP(ORDER BY lvl DESC) AS "Строка наоборот"
FROM (SELECT regexp_substr(str,'[^|]+',1,level) AS rs, level AS lvl
    FROM sour
    CONNECT BY regexp_instr(str,'\|',1,level - 1) > 0), sour;
```

Алгоритм3:

```
 Разбиваем строку на числа с помощью регулярных выражений. Соединяем
получившиеся строки listaggom, но в обратном порядке. Так же нам нужно кол-во
строк, эквивалентное кол-ву чисел в строке.

БЕЗ МОДЕЛ И ИЕРАРХИЧЕСКИХ ЗАПРОСОВ
WITH
t AS (
SELECT '0|-83|-104|2|1|2' str_in
FROM dual),
rec(str, lead, num) AS (
SELECT '|' || str_in || '|', regexp_instr(str_in || '|', '[|]'),
```

```
substr(str_in || '|', 1, regexp_instr(str_in || '|', '[|]')-1)
FROM t
UNION ALL
SELECT str, regexp_instr(str, '[|]', lead+1), substr(str, lead+1,
regexp_instr(str, '[|]', lead+1)-lead-1)
FROM rec
WHERE lead<>length(str)),
res AS (
SELECT str, lead, num, rownum rr
FROM rec)
SELECT (listagg(num, '|') WITHIN GROUP (ORDER BY rr DESC)) "Result"
FROM res;
```

5. Для произвольного целого числа определить числа, полученные перестановками цифр в числе и имеющие максимальные суммы абсолютных разностей между соседними цифрами. Например, для числа 1239 результат должен быть:

3192

2913

Суммы абсолютных разностей равны:

|3-1| + |1-9| + |9-2| = 17

|2-9| + |9-1| + |1-3| = 17

Убедиться в работоспособности при 5, 10, 15 и 20 цифрах в числе.

```
DEFINE num = '19234';


UNDEFINE num;


WITH digits AS
   (SELECT substr('&NUM',level,1) d, level l
   FROM dual
   CONNECT BY level <= length('&NUM')),
transp_d AS
   (SELECT DISTINCT replace(sys_connect_by_path(d,' '),' ') nums
   FROM digits
   WHERE CONNECT_BY_ISLEAF = 1
   CONNECT BY NOCYCLE PRIOR l != l),
nums_and_sums AS
   (SELECT nums,
      (SELECT SUM(abs(to_number(substr(nums,level,1) ) - to_number(substr(nums,level
+ 1,1) ) ) )
      FROM dual
      CONNECT BY level <= length(nums) - 1) sums
```

```
   FROM transp_d)

SELECT nums
FROM nums_and_sums
WHERE sums = (SELECT MAX(sums)
   FROM nums_and_sums)
ORDER BY nums;
```

БИЛЕТ 11

1. Используя словарь данных, получить информацию об ограничениях CHECK схемы:

В списках имена столбцов вывести через запятую. Имя таблицы не должно повторяться.

Задачу решить без использования функций Listagg и Wm_concat.

Пример представления результатов:

SQL | All Rows Fetched: 8 in 1,748 seconds

	Имя таблицы	Имя ограничения	Столбцы,входящие в ограничение	Ограничение CHECK
1	EMPLOYEES	EMP_SALARY_MIN	SALARY	salary > 0
2	JOB_HISTORY	JHIST_DATE_INTERVAL	END_DATE,START_DATE	end_date > start_date
3	TASK_13	COLUMN12345	COLUMN_1,COLUMN_2,COLUMN_3,COLUMN_4,COLUMN_5	column_1>ALL (column_2, column_3, column_4, column_5)
4	TRIP	CON_PER_BEG	PER_BEG	per_beg=TRUNC(per_beg)
5		CON_PER_END	PER_END	per_end=TRUNC(per_end)
6	ПРЕПАРАТ	ПРЕПАРАТ_CHK1	АББРЕВ_ФОРМЫ,ТИП	Аббрев_формы!=Тип
7	ПРЕПОДАВАТЕЛЬ	SYS_C00201812	КАФЕДРА	КАФЕДРА Like 'Кафедра%'
8	УСПЕВАЕМОСТЬ	SYS_C00201820	ОЦЕНКА	ОЦЕНКА between 1 and 5

Решение 1:

```
WITH
t AS (
    SELECT uc.table_name table_name, uc.constraint_name constraint_name, ucc.column_name cn, uc.search_condition sc,
    ROW_NUMBER() OVER(PARTITION BY uc.table_name, uc.constraint_name ORDER BY ucc.column_name) rn,
        ROW_NUMBER() OVER(PARTITION BY uc.table_name ORDER BY ucc.column_name) rn_t
    FROM user_constraints uc LEFT JOIN user_cons_columns ucc ON uc.constraint_name = ucc.constraint_name
  WHERE uc.constraint_type = 'C'),
result AS (
  SELECT table_name, constraint_name, ltrim(sys_connect_by_path(cn,','),',') cn, sc, rn, rn_t
  FROM t
  WHERE CONNECT_BY_ISLEAF = 1
  START WITH rn = 1
  CONNECT BY PRIOR table_name = table_name
      AND PRIOR constraint_name = constraint_name
      AND PRIOR rn + 1 = rn
  ORDER BY table_name, rn_t)
SELECT CASE WHEN rn_t = 1 OR rn != 1 THEN table_name
    ELSE ' ' END AS "Имя таблицы", constraint_name AS "Имя ограничения", cn AS "Столбцы, входящие в огр-е", sc "Ограничение CHECK"
```

FROM result;

2. Одной командой SELECT вывести список сотрудников компании, имеющих наименьший оклад среди сотрудников подразделения, в котором они работают.

Сведения о сотрудниках, для которых неизвестно подразделение компании, к которому они приписаны выводить не нужно.

В результат вывести:

1.Идентификатор подразделения компании, к которому приписан сотрудник.

2. Фамилию сотрудника.

3.Оклад, установленный сотруднику.

В команде SELECT запрещается использовать:

- Фразы WITH, GROUP BY,HAVING, ORDER BY, CONNECT BY, START WITH,
- Условия IN, =ANY, =SOME, NOT IN, <> ALL, EXISTS, NOT EXISTS,
- Подзапросы (subqueries), в том числе подзапросы во фразе FROM,
- Иерархические запросы (hierarchical queries),
- Агрегатные функции (aggregate functions) – MIN,MAX, SUM,COUNT,AVG и др.
- Аналитические функции (analytic functions)

Решение (ОЮ):

SELECT department_id, last_name, salary -- Все сотрудники

FROM Employees

WHERE department_id IS NOT NULL -- За исключением тех, котороые не приписаны ни к одному отделу

MINUS -- Вычитаем из всех сотрудиков тех, которые имеют зарплату большую, чем кто-то из коллег

SELECT E.department_id, E.last_name, E.salary+

FROM Employees E INNER JOIN Employees D ON E.department_id = D.department_id -- Делаем CROSS JOIN зарплат сотрудников внутри отделов

WHERE E.salary > D.salary -- И исключаем тех, которые имеют зарплату больше чем один из коллег

3. Написать запрос, который все пары прямых скобок в строке, внутри которых имеется две или более пары прямых скобок, заменит на фигурные скобки. Например, для строки

[[[98+77]-9]-1] => [[175-9]-[1]]=>165

результат должен быть

{[[98+77]-9]-1} =>{[175-9]-[1]}=>165

Решение 1:

```
WITH tmp AS (SELECT '[[[98+77]-9]-1] => [[175-9]-[1]]=>165' str
        FROM dual)
SELECT str,
regexp_replace(str,'\[(([^][]*\[[^][]*\[[^][]*\][^][]*\][^][]*)|([^][]*\[[^][]*\][^][]*\[[^][]*\][^][]*))\]','{\1}') result
FROM tmp;
```

	STR	RESULT
1	[[[98+77]-9]-1] => [[175-9]-[1]]=>165	{[[98+77]-9]-1} => {[175-9]-[1]}=>165

4. В написанном выражении ((((1?2)?3)?4)?5)?6 вместо каждого знака ? вставить знак одной из 4 арифметических операций +,-,*,/ так, чтобы результат вычислений равнялся 35 (при делении дробная часть в частном отбрасывается). Найти все решения

Решение 1 (Кож):

```
WITH OPERATION AS ( SELECT '+' AS OPER FROM DUAL
UNION ALL
SELECT '-' FROM DUAL
UNION ALL
SELECT '*' FROM DUAL
UNION ALL
SELECT '/' FROM DUAL
),
EXPRESSION (E_LINE, EXPR, VAL) AS (SELECT '(((((1' AS E_LINE, 1
AS EXPR, 1 AS VAL FROM DUAL
UNION ALL
SELECT E_LINE || OP.OPER ||
(EXPR + 1) || ')' AS E_LINE,
EXPR + 1 AS EXPR,
CASE
WHEN OP.OPER = '+'
THEN VAL + (EXPR + 1)
WHEN OP.OPER = '-'
THEN VAL - (EXPR + 1)
WHEN OP.OPER = '*'
THEN VAL * (EXPR + 1)
```

```
WHEN OP.OPER = '/'
THEN TRUNC(VAL / (EXPR
+ 1),0)
END AS VAL
FROM EXPRESSION
JOIN OPERATION OP ON
EXPR < 7
--WHERE EXPR < 9
)
SELECT SUBSTR(E_LINE,2,LENGTH(E_LINE)-2) || ' = ' || VAL AS "RESULT"
FROM EXPRESSION
WHERE EXPR = 6 AND VAL = 35;
```

	RESULT
1	((((1+2)+3)*4)+5)+6 = 35
2	((((1+2)*3)*4)+5)-6 = 35
3	((((1*2)*3)*4)+5)+6 = 35

5. Создать запрос, который позволит оставить только одно из повторяющихся слов в тексте, идущих друг за другом и разделенных пробелами. Количество повторяющихся слов – произвольное.

Решение 1 :

```
DEFINE SOURCE_STRING = 'что-то что-то где-то где-то';
WITH SOURCE AS (
  SELECT * FROM (
                                                    SELECT
    TO_CHAR(REGEXP_SUBSTR(REGEXP_REPLACE('&SOURCE_STRING', '[\ ]+',' '),'(([^ ]+)([\ ]+(\2))+)|([^ ]?+)', 1, LEVEL,'i')) WORD, LEVEL L
     FROM DUAL
     CONNECT BY REGEXP_SUBSTR(REGEXP_REPLACE('&SOURCE_STRING', '[\ ]+',' '),'[^ ]+',1,LEVEL) IS NOT NULL)
     WHERE WORD IS NOT NULL),
RES AS ( SELECT TO_CHAR(REGEXP_SUBSTR(WORD, '[^ ]+', 1, 1)) AS WORD, L
      FROM SOURCE)
SELECT '&SOURCE_STRING' AS "Строка", LISTAGG(WORD, ' ') WITHIN GROUP (ORDER BY L) "Результат"
FROM RES;
UNDEFINE SOURCE_STRING;
```

	Строка	Результат
1	что-то что-то где-то где-то	что-то где-то

БИЛЕТ 12

1. Имеется таблица с символьным столбцом. Создать запрос для вывода тех значений, которые содержат в себе палиндромы, и самые длинные выражения, представляющие из себя палиндром.

Например, для таблицы с данными:

Text
Крокодил
Колокол
Станок

Результат должен быть:

Text	**Palindrom**
Крокодил	око
Колокол	Колок, локол

```
CREATE TABLE tab13(TEXT VARCHAR2(50)CONSTRAINT zad13_pk PRIMARY KEY);


INSERT INTO tab13 VALUES('Крокодил');
INSERT INTO tab13 VALUES('Колокол');
INSERT INTO tab13 VALUES('Крокодил и колокол');
INSERT INTO tab13 VALUES('Станок');
INSERT INTO tab13 VALUES('  Крокодил  ');
INSERT INTO tab13 VALUES('  ');
INSERT INTO tab13 VALUES('Крокодил и  колокол');


WITH REVERSE_STR AS (SELECT text as inptext, TRIM(both ' ' from replace(replace(replace(text, ' ', ' _'), '_ '), '_')) as TEXT,
REVERSE(TRIM(both ' ' from replace(replace(replace(text, ' ', ' _'), '_ '), '_'))) AS REV_TEXT,
LENGTH(TEXT) AS LEN,
ROWNUM AS RN
FROM tab13),
```

```
PALINDROM AS (SELECT RN, inptext, TEXT,
CASE
WHEN SUBSTR(LOWER(TEXT), L1 - L2, L2) =
SUBSTR(LOWER(REV_TEXT), LENGTH(REV_TEXT) - L1 + 1 - L2, L2)
THEN SUBSTR(TEXT, L1 - L2, 2 * L2 + 1)
WHEN SUBSTR(LOWER(TEXT), L1 - L3, L3 + 1) =
SUBSTR(LOWER(REV_TEXT), LENGTH(REV_TEXT) - L1 - L3, L3 + 1)
THEN SUBSTR(TEXT, L1 - L3, (L3 + 1) * 2)
END STR
FROM REVERSE_STR A
INNER JOIN (SELECT ROWNUM L1 FROM USER_OBJECTS) B
ON(A.LEN >= B.L1)
INNER JOIN (SELECT ROWNUM L2 FROM USER_OBJECTS) C
ON(B.L1 - C.L2 > 0 AND B.L1 + C.L2 <= A.LEN)
INNER JOIN (SELECT ROWNUM - 1 L3 FROM USER_OBJECTS) D
ON(B.L1 - D.L3 > 0 AND B.L1 + D.L3 <= A.LEN)
ORDER BY RN, L1),
result as(
SELECT DISTINCT RN, inptext, TEXT, str
FROM PALINDROM pr
WHERE STR IS NOT NULL
AND length(str) = ANY(SELECT MAX(length(str))
FROM palindrom
WHERE rn = pr.rn
group by rn)
ORDER BY RN, TEXT)

SELECT DISTINCT inptext, LISTAGG(str, ', ')
WITHIN GROUP (ORDER BY str)
OVER (PARTITION BY rn) AS "Palindrom"
FROM result
ORDER BY inptext;
```

2. Для произвольной строки, состоящей из открывающих и закрывающих скобок написать запрос для вывода всех слов максимальной длины, представляющих

правильные скобочные записи. Например, для строки (()(() ответ должен быть:

()()

(())

Решение 1:

```
WITH input AS (SELECT '(()(()(' AS str, '' AS s
            FROM dual),
    tmp1 AS (SELECT str, level AS lvl, substr(str,level,1) AS c
            FROM input
            CONNECT BY level <= length(str)),
    tmp2 AS (SELECT DISTINCT str, replace(sys_connect_by_path(c,'.'),'.','') s
            FROM tmp1
            CONNECT BY PRIOR str = str AND PRIOR lvl < lvl AND PRIOR sys_guid() IS
NOT NULL),
    tmp4 (s, lvl, c) AS (SELECT s, 1 AS lvl, 1 AS c
                    FROM tmp2
                    WHERE s LIKE '(%' AND regexp_count(s,'\(') = regexp_count(s,'\)')
                    UNION ALL
                    SELECT s, lvl + 1, CASE WHEN substr(s,lvl + 1,1) = '('
                                    THEN c + 1
                                    ELSE c - 1
                                END AS c
                    FROM tmp4
                    WHERE lvl < 1 + length(s) AND c >= 0),
     tmp5 AS (SELECT *
            FROM tmp4
            WHERE lvl = length(s))
SELECT s AS result
FROM tmp5
WHERE c = 0 AND   length(s) = (SELECT MAX(length(s) )
                        FROM tmp5
                        WHERE c = 0);
```

	RESULT
1	() ()
2	(( ))

3. Имеется таблица D_V с первым столбцом Dat типа DATE (первичный ключ) и вторым столбцом Val типа NUMBER. Пример (строки упорядочены по первому

столбцу):

DAT	VAL
01-08-08	232
02-08-08	
10-08-08	182
11-08-08	
21-08-08	240
22-08-08	
23-08-08	

Требуется написать запрос для получения на основе таблицы D_V следующей таблицы:

DAT	MAX_VAL
01-08-08	232
02-08-08	232
10-08-08	182
11-08-08	182
21-08-08	240
22-08-08	240
23-08-08	240

Данная результирующая таблица должна быть упорядочена по Dat, но вместо пустых значений, которые присутствовали в столбце VAL отсортированной по DAT исходной таблицы, в столбце MAX_VAL результирующей таблицы, должны присутствовать значения столбца из предыдущей строки.

Задачу решить без использования аналитических функций.

Решение 1:

```
CREATE TABLE D_V (dat DATE PRIMARY KEY, val NUMBER);


INSERT INTO D_V VALUES('01-08-08', 232);
INSERT INTO D_V VALUES('02-08-08', NULL);
INSERT INTO D_V VALUES('10-08-08', 182);
INSERT INTO D_V VALUES('11-08-08', NULL);
INSERT INTO D_V VALUES('21-08-08', 240);
INSERT INTO D_V VALUES('22-08-08', NULL);
INSERT INTO D_V VALUES('23-08-08', NULL);


SELECT dat, CASE WHEN val IS NULL
```

```
        THEN (SELECT tmp1.val
            FROM D_V tmp1
            WHERE tmp1.dat = (SELECT MAX(dat)
                    FROM D_V
                    WHERE dat < tmp2.dat AND val IS NOT NULL))
        ELSE val END AS max_val
FROM D_V tmp2
ORDER BY dat;
```

	DAT	MAX_VAL
1	01.08.08	232
2	02.08.08	232
3	10.08.08	182
4	11.08.08	182
5	21.08.08	240
6	22.08.08	240
7	23.08.08	240

4. Проверить наличие циклов в таблице подчиненностей. Вывести циклические зависимости в строчку в виде Номер1.Имя1->Номер2.Имя2->…Номер1.Имя1, начиная с первого по алфавиту имени.

Например, для таблицы подчинённостей:

Номер	**Имя**	**Номер_начальника**
1	Алексей	2
2	Пётр	3
3	Павел	4
4	Иван	2
5	Кристина	3
6	Андрей	5

Результат должен быть:

CYCLE
4.Иван->3.Павел->2.Пётр->4.Иван

Решение:

```
WITH temp_table AS (
  SELECT 1 empno, 'Алексей' ename, 2 mgr
  FROM DUAL
  UNION ALL
  SELECT 2, 'Пётр', 3
  FROM DUAL
  UNION ALL
  SELECT 3, 'Павел', 4
  FROM DUAL
  UNION ALL
  SELECT 4, 'Иван', 2
  FROM DUAL
  UNION ALL
  SELECT 5, 'Кристина', 3
  FROM DUAL
  UNION ALL
  SELECT 6, 'Андрей', 5
  FROM DUAL),
temp AS
(SELECT e.*, m.ename mname
FROM temp_table e
JOIN temp_table m ON e.mgr = m.empno)

SELECT substr(sys_connect_by_path(mgr || '.' || mname, '->') || '->' || empno || '.' || ename, 3)
    cycle
FROM temp
WHERE CONNECT_BY_ROOT(mgr) = empno
CONNECT BY NOCYCLE PRIOR empno = mgr AND PRIOR mgr > mgr;

--С СОЗДАНИЕМ ТАБЛИЦЫ
```

```
CREATE TABLE temp_table (
  empno  number(10,0),
  ename varchar2(50 char),
  mgr number(10,0),
  CONSTRAINT temp_table PRIMARY KEY (empno)
);

INSERT INTO temp_table VALUES (1, 'Алексей', 2);
INSERT INTO temp_table VALUES (2, 'Пётр', 3);
INSERT INTO temp_table VALUES (3, 'Павел', 4);
INSERT INTO temp_table VALUES (4, 'Иван', 2);
INSERT INTO temp_table VALUES (5, 'Кристина', 3);
INSERT INTO temp_table VALUES (6, 'Андрей', 5);

WITH temp AS
(SELECT e.*, m.ename mname
FROM temp_table e
JOIN temp_table m ON e.mgr = m.empno)

SELECT substr(sys_connect_by_path(mgr || '.' || mname, '->') || '->' || empno || '.' || ename, 3) cycle
FROM temp
WHERE CONNECT_BY_ROOT(mgr) = empno
CONNECT BY NOCYCLE PRIOR empno = mgr AND PRIOR mgr > mgr;
```

	CYCLE
1	4.Иван->3.Павел->2.Пётр->4.Иван

5. Вывести информацию о таблицах схемы в виде:

Имя таблицы	**Столбцы первичного ключа**	**Столбцы с ограничением уникальности**	**Список подчиненных таблиц со столбцами вторичных ключей**
Table 1	Col1, Col2	Col3, Col4	Table2 (Col5, Col6), Table3 (Col7,Col8)

Пример результата:

Имя таблицы	**Столбцы первичного ключа**	**Столбцы с ограничением уникальности**	**Список подчиненных таблиц со столбцами вторичных ключей**

DEPT3	DEPARTMENT_ID, DEPARTMENT_NAME	-	DEPT2(ID,NAME)
EMPLOYEES	EMPLOYEE_ID	EMAIL	DEPARTMENTS(MANAGER_ID), EMPLOYEES(MANAGER_ID), JOB_HISTORY(EMPLOYEE_ID)
EMPLOYEES	EMPLOYEE_ID	FIRST_NAME, LAST_NAME	DEPARTMENTS(MANAGER_ID), EMPLOYEES(MANAGER_ID), JOB_HISTORY(EMPLOYEE_ID)
…	…	…	…

```
WITH
src AS
   (SELECT all_t.table_name supp, c2.table_name cust,
          LAG(c2.table_name) OVER(PARTITION BY all_t.table_name ORDER BY all_t.table_name) one_more
   FROM user_constraints c1
   INNER JOIN user_constraints c2
   ON c2.r_constraint_name = c1.constraint_name
   RIGHT JOIN user_tables all_t
   ON all_t.table_name = c1.table_name),
t AS
   (SELECT supp, TRIM(LEADING ',' FROM(sys_connect_by_path(cust,',') ) ) AS listing, level AS l
   FROM src
   START WITH one_more IS NULL
   CONNECT BY NOCYCLE PRIOR cust = one_more
   AND PRIOR supp = supp),
```

```
res AS
   (SELECT supp, listing
   FROM t m
   WHERE l = (SELECT MAX(l)
      FROM t
      WHERE supp = m.supp
      GROUP BY supp)),
p_col AS
   (SELECT c3.table_name sys_table, column_name pk_col, LAG(column_name)
   OVER(PARTITION BY c3.table_name ORDER BY c3.table_name) col_col
   FROM user_constraints c3 JOIN user_cons_columns c4
   ON c3.constraint_name = c4.constraint_name
   WHERE c3.constraint_type = 'P'),
u_col AS
   (SELECT c3.table_name sys_table, column_name u_col, LAG(column_name)
   OVER(PARTITION BY c3.table_name ORDER BY c3.table_name) ucol_col
   FROM user_constraints c3 JOIN user_cons_columns c4
   ON c3.constraint_name = c4.constraint_name
   WHERE c3.constraint_type = 'U'),
p_col2 AS
   (SELECT sys_table,
   TRIM(LEADING ',' FROM(sys_connect_by_path(pk_col,',') ) ) listing_col,
   level lev
   FROM p_col
   START WITH col_col IS NULL
   CONNECT BY NOCYCLE PRIOR pk_col = col_col
   AND PRIOR sys_table = sys_table),
u_col2 AS
   (SELECT sys_table,
   TRIM(LEADING ',' FROM(sys_connect_by_path(u_col,',') ) ) ulisting_col, level lev
   FROM u_col
   START WITH ucol_col IS NULL
   CONNECT BY NOCYCLE PRIOR u_col = ucol_col
   AND PRIOR sys_table = sys_table),
p_col3 AS
```

```
    (SELECT sys_table, listing_col
    FROM p_col2 l
    WHERE lev = (SELECT MAX(lev)
        FROM p_col2
        WHERE sys_table = l.sys_table
        GROUP BY sys_table)),
u_col3 AS
    (SELECT sys_table, ulisting_col
    FROM u_col2 l
    WHERE lev = (SELECT MAX(lev)
        FROM u_col2
        WHERE sys_table = l.sys_table
        GROUP BY sys_table))
SELECT DISTINCT supp AS "Имя таблицы",
nvl(listing_col,'Первичного ключа нет') AS "Столбцы первичного ключа",
nvl(ulisting_col,'Ограничения уникальности в таблице нет') AS "Столбцы с огр. уникальности",
nvl(listing,'Подчиненных таблиц нет') AS "Список подчиненных таблиц"
FROM res r
LEFT JOIN p_col3 p
ON r.supp = p.sys_table
LEFT JOIN u_col3 u
ON r.supp = u.sys_table;
```

ДОПОЛНИТЕЛЬНЫЕ ЗАДАЧИ

1. Определить много-столбцовые ограничения для каждой таблицы схемы.

   Результат представить в виде:

Номер таблицы	Таблица	Номер ограничения	Имя ограничения	Тип ограничения	Кол-во столбцов	Кол-во многостолбцовых ограничений
1	JOB_HISTORY	1	JHIST_EMP_ID_ST_DATE_PK	Первичный ключ	2	2
		2	JHIST_DATE_INTERVAL	Ограничение CHECK	2	
2	СОТРУДНИК_ПРОЕКТ	1	ХРКСОТРУДНИК_ПРОЕКТ	Первичный ключ	2	1

ПРОВЕРИТЬ!!!

```
WITH
ex AS
   (SELECT table_name, constraint_name, COUNT(DISTINCT column_name) cnt
   FROM user_cons_columns
```

НЕПОНЯТНАЯ СТРОКА!!!

```
   --WHERE constraint_name NOT LIKE '%BIN%'
   GROUP BY table_name, constraint_name
   HAVING COUNT(DISTINCT column_name) > 1
   ORDER BY table_name),
tabs AS
   (SELECT DISTINCT table_name
   FROM ex
   ORDER BY table_name),
nums AS
   (SELECT table_name, ROWNUM rnm
   FROM tabs)
SELECT
```

```
CASE

   WHEN ROW_NUMBER() OVER(PARTITION BY c.table_name ORDER BY
c.constraint_name) = 1 THEN TO_CHAR(nums.rnm)

   ELSE ' '

END AS "Номер таблицы",

CASE

   WHEN ROW_NUMBER() OVER(PARTITION BY c.table_name ORDER BY
c.constraint_name) = 1 THEN c.table_name

   ELSE ' '

END AS "Таблица",

ROW_NUMBER() OVER(PARTITION BY c.table_name ORDER BY c.constraint_name) AS
"Номер ограничения",

c.constraint_name AS "Имя ограничения",

DECODE(c.constraint_type,'P','Первичный ключ','C','Ограничение CHECK','U','Уникальный
ключ','R','Внешний ключ',NULL) AS "Тип ограничения",

ex1.cnt AS "Кол-во столбцов",

CASE

   WHEN ROW_NUMBER() OVER(PARTITION BY c.table_name ORDER BY
c.constraint_name) = 1 THEN TO_CHAR(COUNT(DISTINCT total.constraint_name))

   ELSE ' '

END AS "Кол-во ограничений"

FROM user_constraints c INNER JOIN ex ex1

ON ( ex1.constraint_name = c.constraint_name )

INNER JOIN ex total

ON ( ex1.table_name = total.table_name )

INNER JOIN nums

ON ( ex1.table_name = nums.table_name )

INNER JOIN user_cons_columns cols

ON ( ex1.constraint_name = cols.constraint_name )
```

```
GROUP BY c.table_name, c.constraint_name, c.constraint_type,

ex1.cnt, nums.rnm

ORDER BY c.table_name;
```

2. Определить даты последних пятниц четырех високосных годов, ближайших к заданной дате. Результат вывести в виде списка дат по убыванию.

   Например, для заданной даты 10.01.2017 ответ должен быть:

   27-Дек-2024, 25-Дек-2020, 30-Дек-2016, 28-Дек-2012

Задачу решить без использования иерархических запросов, рекурсивного With и Model.

```
alter session set nls_language = 'russian';

undefine x;

define x = '01.07.2004';
define x = '02.07.2004';
define x = '10.01.2017';
define x = '10.01.0001';
define x = '10.01.9999';
define x = '10.01.-4712';

with inp as( --создаем 10 строк
select
to_date('&x', 'DD.MM.SYYYY') as dat
from dual),
f1 as(
select 1 as num from dual union all
select 2 as num from dual union all
select 3 as num from dual union all
select 4 as num from dual union all
select 5 as num from dual union all
select 6 as num from dual union all
select 7 as num from dual union all
select 8 as num from dual union all
select 9 as num from dual union all
select 10 as num from dual
),
f2 as(  --декартвово умножаем таблицы 5 раз чтобы получить 100 000 строк. оставляем номера от -4712 до 9999
select
rownum - 4713 as yrs
from f1 cross join f1 cross join f1 cross join f1 cross join f1
where rownum - 4713 <= 9999
```

```
),
f3 as (    --получаем високосные года
select
yrs
from f2
where yrs >= -4708 and mod(yrs,4) = 0 and yrs <> 0
order by 1),
f4 as(  --из вводимой даты полуаем float число года
select
to_number(extract(year from dat) +
to_char(dat, 'DDD')/
case when extract(year from dat) in (select * from f3)
then 366 else 365 end) as dat
from inp),
f5 as ( --получаем все рядом стоящие 4 високосные года
select
yrs yr1,
lead(yrs, 1) over (order by yrs) yr2,
lead(yrs, 2) over (order by yrs) yr3,
lead(yrs, 3) over (order by yrs) yr4
from f3
),
f6 as ( --оставляем те, расстояние до которых от заданной даты минимально. yr1 + 1 и yr2
+ 1, потому что мы ищем расстояние до самого года, а не до его начала
select *
from(
   select
   yr1,yr2,yr3,yr4,
   row_number() over(order by abs(((select * from f4) - (yr1 + 1)) * ((select * from f4) - (yr2 +
1)) * (yr3 - (select * from f4)) * (yr4 - (select * from f4)))) r
   from f5)
where r = 1
) --выводим
select
replace(to_char(next_day(to_date('24.12.'||yr4,'DD.MM.SYYYY'),'Пятница'), 'DD-Mon-
SYYYY'),' ','') || '  ' ||
replace(to_char(next_day(to_date('24.12.'||yr3,'DD.MM.SYYYY'),'Пятница'), 'DD-Mon-
SYYYY'),' ','') || '  ' ||
replace(to_char(next_day(to_date('24.12.'||yr2,'DD.MM.SYYYY'),'Пятница'), 'DD-Mon-
SYYYY'),' ','') || '  ' ||
replace(to_char(next_day(to_date('24.12.'||yr1,'DD.MM.SYYYY'),'Пятница'), 'DD-Mon-
SYYYY'),' ','') rezult
from f6;
```

2. В таблицу записана информация, об удачных и неудачных попытках подключения

к базе данных (Пользователь, Время, Удачно\Неудачно). Требуется, используя раздел Model, получить список пользователей, которые совершили подряд три неудачные попытки подключения наряду с зафиксированным временем третьей неудачной попытки. После трех подряд неудачных попыток отсчет попыток начинается сначала. Например, для таблицы:

Пользователь	Время	Статус
A	20.11.11 17:58:00	Неудачно
B	20.11.11 18:00:05	Удачно
C	20.11.11 18:10:03	Неудачно
A	20.11.11 18:12:20	Неудачно
B	20.11.11 18:18:00	Неудачно
B	20.11.11 18:20:01	Удачно
C	20.11.11 18:25:42	Неудачно
A	20.11.11 18:30:12	Неудачно
A	20.11.11 18:32:24	Неудачно
A	20.11.11 18:35:00	Удачно
B	20.11.11 18:41:30	Удачно
C	20.11.11 18:42:08	Неудачно
C	20.11.11 18:48:00	Удачно
A	20.11.11 18:52:00	Неудачно
A	20.11.11 18:53:13	Неудачно
B	20.11.11 18:54:30	Неудачно
A	20.11.11 18:55:19	Неудачно
A	20.11.11 18:55:58	Удачно

Результат должен быть такой:

Пользователь	Время
A	20.11.11 18:30:12
C	20.11.11 18:42:08
A	20.11.11 18:55:19

Решение 1:

```
drop table table1;

CREATE TABLE TABLE1 (пользователь varchar2(5), время date, статус varchar2(10));


INSERT INTO TABLE1 VALUES ('A', to_date('20.11.11 17:58:00', 'DD.MM.YY HH24:MI:SS'), 'Неудачно');

INSERT INTO TABLE1 VALUES ('B', to_date('20.11.11 18:00:05', 'DD.MM.YY HH24:MI:SS'), 'Удачно');

INSERT INTO TABLE1 VALUES ('C', to_date('20.11.11 18:10:03', 'DD.MM.YY HH24:MI:SS'), 'Неудачно');
```

```
INSERT INTO TABLE1 VALUES ('A', to_date('20.11.11 18:12:20', 'DD.MM.YY
HH24:MI:SS'), 'Неудачно');

INSERT INTO TABLE1 VALUES ('B', to_date('20.11.11 18:18:00', 'DD.MM.YY
HH24:MI:SS'), 'Неудачно');

INSERT INTO TABLE1 VALUES ('B', to_date('20.11.11 18:20:01', 'DD.MM.YY
HH24:MI:SS'), 'Удачно');

INSERT INTO TABLE1 VALUES ('C', to_date('20.11.11 18:25:42', 'DD.MM.YY
HH24:MI:SS'), 'Неудачно');

INSERT INTO TABLE1 VALUES ('A', to_date('20.11.11 18:30:12', 'DD.MM.YY
HH24:MI:SS'), 'Неудачно');

INSERT INTO TABLE1 VALUES ('A', to_date('20.11.11 18:32:24', 'DD.MM.YY
HH24:MI:SS'), 'Неудачно');

INSERT INTO TABLE1 VALUES ('A', to_date('20.11.11 18:35:00', 'DD.MM.YY
HH24:MI:SS'), 'Удачно');

INSERT INTO TABLE1 VALUES ('B', to_date('20.11.11 18:41:30', 'DD.MM.YY
HH24:MI:SS'), 'Удачно');

INSERT INTO TABLE1 VALUES ('C', to_date('20.11.11 18:42:08', 'DD.MM.YY
HH24:MI:SS'), 'Неудачно');

INSERT INTO TABLE1 VALUES ('C', to_date('20.11.11 18:48:00', 'DD.MM.YY
HH24:MI:SS'), 'Удачно');

INSERT INTO TABLE1 VALUES ('A', to_date('20.11.11 18:52:00', 'DD.MM.YY
HH24:MI:SS'), 'Неудачно');

INSERT INTO TABLE1 VALUES ('A', to_date('20.11.11 18:53:13', 'DD.MM.YY
HH24:MI:SS'), 'Неудачно');

INSERT INTO TABLE1 VALUES ('B', to_date('20.11.11 18:54:30', 'DD.MM.YY
HH24:MI:SS'), 'Неудачно');

INSERT INTO TABLE1 VALUES ('A', to_date('20.11.11 18:55:19', 'DD.MM.YY
HH24:MI:SS'), 'Неудачно');

INSERT INTO TABLE1 VALUES ('A', to_date('20.11.11 18:55:58', 'DD.MM.YY
HH24:MI:SS'), 'Удачно');
```

```
SELECT tab.пользователь, TO_CHAR(tab.время,'DD.MM.YY HH24:MI:SS') AS время

FROM (SELECT *

   FROM (SELECT счетчик_по_времени,
```

```
        ROWNUM AS счетчик_по_подключению,
        счетчик_по_времени - ROWNUM AS разница,
        пользователь, время, статус
        FROM (SELECT ROWNUM AS счетчик_по_времени,
          пользователь, время, статус
          FROM (SELECT пользователь, время, статус
            FROM table1
            ORDER BY пользователь, время, статус)
          ORDER BY пользователь, статус, время))
        WHERE статус = 'Неудачно') tab
JOIN (SELECT разница, пользователь,
    MIN(счетчик_по_времени) счетчик_по_времени
    FROM (SELECT *
      FROM (SELECT счетчик_по_времени,
        ROWNUM AS счетчик_по_подключению,
        счетчик_по_времени - ROWNUM AS разница,
        пользователь, время, статус
        FROM (SELECT ROWNUM AS счетчик_по_времени,
          пользователь, время, статус
          FROM (SELECT пользователь, время, статус
            FROM table1
            ORDER BY пользователь, время, статус)
          ORDER BY пользователь, статус, время))
        WHERE статус = 'Неудачно')
    GROUP BY разница, пользователь
    HAVING COUNT(*) >= 3) t
ON tab.пользователь = t.пользователь
```

```
AND tab.разница = t.разница

AND mod(1 + tab.счетчик_по_времени - t.счетчик_по_времени,3) = 0;


CREATE TABLE DB_CONNECTIONS (

user_name VARCHAR(20),

con_time DATE,

status VARCHAR(20),

CONSTRAINT DB_CONNECTIONS PRIMARY KEY(user_name, con_time)

);


--Решение2 (доп):

INSERT INTO DB_CONNECTIONS VALUES ('A', TO_DATE('20.11.11 17:58:00', 'dd.mm.rr hh24:mi:ss'), 'Неудачно');   --1

INSERT INTO DB_CONNECTIONS VALUES ('B', to_date('20.11.11 18:00:05', 'DD.MM.YY HH24:MI:SS'), 'Удачно');     --2

INSERT INTO DB_CONNECTIONS VALUES ('C', to_date('20.11.11 18:10:03', 'DD.MM.YY HH24:MI:SS'), 'Неудачно');   --3

INSERT INTO DB_CONNECTIONS VALUES ('A', to_date('20.11.11 18:12:20', 'DD.MM.YY HH24:MI:SS'), 'Неудачно');   --4

INSERT INTO DB_CONNECTIONS VALUES ('B', to_date('20.11.11 18:18:00', 'DD.MM.YY HH24:MI:SS'), 'Неудачно');   --5

INSERT INTO DB_CONNECTIONS VALUES ('B', to_date('20.11.11 18:20:01', 'DD.MM.YY HH24:MI:SS'), 'Удачно');     --6

INSERT INTO DB_CONNECTIONS VALUES ('C', to_date('20.11.11 18:25:42', 'DD.MM.YY HH24:MI:SS'), 'Неудачно');   --7

INSERT INTO DB_CONNECTIONS VALUES ('A', to_date('20.11.11 18:30:12', 'DD.MM.YY HH24:MI:SS'), 'Неудачно');   --8

INSERT INTO DB_CONNECTIONS VALUES ('A', to_date('20.11.11 18:32:24', 'DD.MM.YY HH24:MI:SS'), 'Неудачно');   --9

INSERT INTO DB_CONNECTIONS VALUES ('A', to_date('20.11.11 18:35:00', 'DD.MM.YY HH24:MI:SS'), 'Удачно');     --10

INSERT INTO DB_CONNECTIONS VALUES ('B', to_date('20.11.11 18:41:30', 'DD.MM.YY
```

HH24:MI:SS'), 'Удачно'); --11

INSERT INTO DB_CONNECTIONS VALUES ('C', to_date('20.11.11 18:42:08', 'DD.MM.YY HH24:MI:SS'), 'Неудачно'); --12

INSERT INTO DB_CONNECTIONS VALUES ('C', to_date('20.11.11 18:48:00', 'DD.MM.YY HH24:MI:SS'), 'Удачно'); --13

INSERT INTO DB_CONNECTIONS VALUES ('A', to_date('20.11.11 18:52:00', 'DD.MM.YY HH24:MI:SS'), 'Неудачно'); --14

INSERT INTO DB_CONNECTIONS VALUES ('A', to_date('20.11.11 18:53:13', 'DD.MM.YY HH24:MI:SS'), 'Неудачно'); --15

INSERT INTO DB_CONNECTIONS VALUES ('B', to_date('20.11.11 18:54:30', 'DD.MM.YY HH24:MI:SS'), 'Неудачно'); --16

INSERT INTO DB_CONNECTIONS VALUES ('A', to_date('20.11.11 18:55:19', 'DD.MM.YY HH24:MI:SS'), 'Неудачно'); --17

INSERT INTO DB_CONNECTIONS VALUES ('A', to_date('20.11.11 18:55:58', 'DD.MM.YY HH24:MI:SS'), 'Удачно'); --18

select user_name, TO_CHAR(con_time, 'dd.mm.rr hh24:mi:ss') as time, status from DB_CONNECTIONS;

```
WITH subquery_1 AS
(
SELECT *
FROM (
    SELECT total_number,
         ROWNUM as current_number,
         total_number - ROWNUM as difference,
         user_name,
         con_time,
         status
    FROM
       (
        SELECT ROWNUM as total_number, user_name, con_time, status
        FROM
          (
           SELECT user_name, TO_CHAR(con_time, 'dd.mm.rr hh24:mi:ss') as con_time, status
           FROM DB_CONNECTIONS
           ORDER BY user_name, con_time
          )
        ORDER BY user_name, status, con_time
       )
   )
WHERE status = 'Неудачно'
```

```
),

subquery_2 AS
(
   SELECT MIN(total_number) as total_number,
         difference,
         user_name
   FROM subquery_1
   GROUP BY difference, user_name
      HAVING COUNT(*) >= 3
)

SELECT sub1.user_name as "Пользователь",
     sub1.con_time as "Время"
FROM subquery_1 sub1 JOIN subquery_2 sub2
                ON
                  sub1.user_name = sub2.user_name
                AND
                  sub1.difference = sub2.difference
                AND
                  MOD((sub1.total_number - sub2.total_number)+1, 3) = 0
ORDER BY "Время";
```

```
--выключает &
SET DEFINE OFF;
select 'Coda & Sid' from dual;
--включает &
SET DEFINE ON;
```

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ 1

2. Создать запрос для вывода списка всех столбцов представления ALL_TABLES. Имена столбцов должны быть отсортированы по алфавиту и разделяться запятыми. Список должен быть разбит на строки, при этом каждая строка должна содержать не более 50 символов. Имя каждого столбца должно размещаться целиком на одной строке. Если после имени столбца следует запятая, то она должна находиться на строке вместе с именем столбца.

   Результат запроса должен также содержать номер строк и количество столбцов в строке.

   Пример представления результата:

Имя представления	Номер строки	Список столбцов	Количество столбцов
ALL_TABLES	1	ACTIVITY_TRACKING, AVG_ROW_LEN, AVG_SPACE,	3
	2	AVG_SPACE_FREELIST_BLOCKS, BACKED_UP, BLOCKS,	3
	3	…….	

Решение:

```
WITH cols(rn, col) AS (
SELECT ROW_NUMBER() OVER (ORDER BY column_name), column_name
FROM all_tab_columns
WHERE LOWER(table_name) = 'all_tables'
)
, R(grp, col, rn, cnt) AS (
SELECT 1, col, rn, 1
FROM cols
WHERE rn = 1
UNION ALL
SELECT CASE WHEN LENGTH(col || (SELECT col FROM cols WHERE rn = R.rn+1))+DECODE(rn, (SELECT
MAX(rn) FROM cols), 1, 2) <= 50 THEN grp ELSE grp+1 END,
— DECODE, потому что для последней записи запятая не нужна
CASE WHEN LENGTH(col || (SELECT col FROM cols WHERE rn = R.rn+1))+DECODE(rn, (SELECT MAX(rn)
FROM cols), 1, 2) <= 50 THEN col ||','|| (SELECT col FROM cols WHERE rn = R.rn+1) ELSE (SELECT col
FROM cols WHERE rn = R.rn+1) END,
R.rn+1,
CASE WHEN LENGTH(col || (SELECT col FROM cols WHERE rn = R.rn+1))+DECODE(rn, (SELECT MAX(rn)
FROM cols), 1, 2) <= 50 THEN cnt+1 ELSE 1 END
FROM R
WHERE R.rn < (SELECT MAX(rn) FROM cols)
), diffs (grp, col, len, ch, cnt) AS (
SELECT grp, col, LENGTH(col), LEAD(grp, 1, grp+1) OVER (ORDER BY grp, LENGTH(col)) - grp, cnt
FROM R
)
SELECT CASE WHEN grp = 1 THEN 'ALL_TABLES' ELSE ' ' END AS "Имя представления",
grp AS "Номер строки",
col || CASE WHEN grp = (SELECT MAX(grp) FROM diffs) THEN '' ELSE ',' END AS "Список столбцов",
cnt AS "Количество столбцов"
FROM diffs
WHERE ch = 1
ORDER BY 2;
```

Решение2:

```
WITH r1(STR) AS (
SELECT listagg(column_name, ',') WITHIN GROUP (ORDER BY COLUMN_NAME)
FROM all_tab_columns
WHERE LOWER(table_name) = 'user_tables'
),
r2(STR, "Список столбцов", LN) AS(
SELECT STR, REGEXP_SUBSTR(SUBSTR(STR, 1,50), '.*,') AS "Список столбцов",
LENGTH(REGEXP_SUBSTR(SUBSTR(STR, 1,50), '.*,'))+1 AS LN
FROM r1
UNION ALL
SELECT STR, REGEXP_SUBSTR(SUBSTR(STR, LN,50), '.*,|.*\w$'),
LN+LENGTH(REGEXP_SUBSTR(SUBSTR(STR, LN,50), '.*,|.*\w$'))
FROM R2
WHERE LENGTH(STR)>LN),

R3 AS(SELECT ROW_NUMBER() OVER (ORDER BY ("Список столбцов")) "Номер строки", "Список
столбцов"
FROM R2)

SELECT CASE WHEN "Номер строки" = 1 THEN 'ALL_TABLES' ELSE ' ' END AS "Имя представления",
"Номер строки", "Список столбцов",
CASE WHEN REGEXP_COUNT("Список столбцов", ',')=0 THEN 1
ELSE REGEXP_COUNT("Список столбцов", ',') END as "Количество столбцов"
FROM R3;
```

Решение3:

```
with tab(rown, nameT1, leg1) as(
select row_number() over(order by table_name), table_name, LENGTH(table_name)+1
from all_tables
order by table_name
),
tab2(row_num, nameT, leg, summa, flag) as(
select 1, (select nameT1 from tab where rown=1), (select leg1 from tab where rown=1), (select leg1
from tab where rown=1), 1
from dual
union all
select row_num+1, (select nameT1 from tab where rown=row_num+1), (select leg1 from tab where
rown=row_num+1),
case when (select leg1 from tab where rown=row_num+1)+summa<50 then (select leg1 from tab where
rown=row_num+1)+summa else (select leg1 from tab where rown=row_num+1) end,
case when (select leg1 from tab where rown=row_num+1)+summa<50 then flag else flag+1 end
from tab2
where row_num!=(select max(rown) from tab)
),
tab3(str) as(
SELECT distinct listagg(nameT, ',') within GROUP (order by row_num) over(PARTITION by flag)
FROM TAB2
order by 1)
select case when rownum=1 then 'ALL_TABLES' else ' ' end, rownum as "Номер строки", str||',' as
"Список столбцов", regexp_count(str, ',')+1 as "Количество столбцов"
from tab3;
```

Решение4:

```
select ROW_NUMBER() OVER (ORDER BY column_name) r, column_name || ',' col
from all_tab_columns
```

```
where table_name='ALL_TABLES';

with t as (
select ROW_NUMBER() OVER (ORDER BY column_name) r, column_name || ',' col
from all_tab_columns
where table_name='ALL_TABLES'),
rec(num, col, r, cnt) as (
select 1 , col, r, 1
from t
where r=1
union all
select
case when length(col || (select col from t where r=rec.r+1))<=50 then num else num+1 end,
case when length(col || (select col from t where r=rec.r+1))<=50 then (col || (select col from t where
r=rec.r+1)) else (select col from t where r=rec.r+1) end,
r+1,
case when length(col || (select col from t where r=rec.r+1))<=50 then cnt+1 else 1 end cnt
from rec
where rec.r<(select max(r) from t))
select case when num=1 then 'ALL_TABLES' else ' ' end "Имя представления", num "Номер строки",
case when num=(select max(num) from rec) then substr(col, 1, length(col)-1) else col end "Список
столбцов", cnt "Количество столбцов"
from rec r
where cnt = (select max(cnt)
from rec
where r.num=rec.num);
```

3. Создать таблицу Customers, содержащую 2 столбца: Id number(15,0) Primary Key и Last_Name varchar2 (40). Создать запрос, который будет выводить значение столбца Id и Номер группы последовательных целых значений с шагом 1. Например, для таблицы, содержащей значения:

Id	Last_Name
1	Mougus
2	Green
3	Grase
7	Scott
8	Trumen
10	Kochhar
12	Drejk
13	Kook

результат должен быть:

Id	Номер группы
1	1
2	1
3	1
7	2
8	2
10	3
12	4
13	4

Задачу решить без использования аналитических функций с помощью раздела Model.

РЕШЕНИЕ С МОДЕЛ от краснокутской

```
CREATE TABLE Customers (Id number(15,0) Primary Key, Last_Name varchar2(40));
INSERT INTO Customers VALUES (1, 'Mougus');
INSERT INTO Customers VALUES (2, 'Green');
INSERT INTO Customers VALUES (3, 'Grase');
INSERT INTO Customers VALUES (7, 'Scott');
INSERT INTO Customers VALUES (8, 'Trumen');
INSERT INTO Customers VALUES (10, 'Kochhar');
INSERT INTO Customers VALUES (12, 'Dreik');
INSERT INTO Customers VALUES (13, 'Kook');
--Вывод ID и номеров групп
WITH num_cust AS ( SELECT rownum n, id, last_name
FROM customers),
RES AS (SELECT n, id, gr
FROM num_cust
MODEL
DIMENSION BY (n)
MEASURES (CAST (0 AS number(3,0)) AS gr, id)
RULES ITERATE (1000) (
```

```
gr[0]=0,
gr[iteration_number+1]=case
WHEN id[iteration_number] = id[iteration_number+1]-1 THEN gr[iteration_number]
ELSE gr[iteration_number]+1
END
)
)
SELECT nc.id "Id", gr "Номер группы"
FROM res, num_cust nc
WHERE nc.id = res.id;
```

РЕШЕНИЕ С MODEL С НАШЕЙ ГРУППЫ

```
CREATE TABLE Customers (Id number(15,0) Primary Key, Last_Name varchar2(40));
INSERT INTO Customers VALUES (1, 'Mougus');
INSERT INTO Customers VALUES (2, 'Green');
INSERT INTO Customers VALUES (3, 'Grase');
INSERT INTO Customers VALUES (7, 'Scott');
INSERT INTO Customers VALUES (8, 'Trumen');
INSERT INTO Customers VALUES (10, 'Kochhar');
INSERT INTO Customers VALUES (12, 'Dreik');
INSERT INTO Customers VALUES (13, 'Kook');

select id, num
from (select rownum n, id, last_name from customers)
model unique SINGLE reference
dimension by(n)
measures(id, 1 num)
rules (
        num[any] =
        case
        when id[cv()] = (id[cv()-1]+1) then num[cv()-1]
        else presentv(num[cv()-1],num[cv()-1],0)+1
        end
);
```

Решение:

```
with task_1(id,last_name) as (select 1,'Mougus' from dual union
                select 2,'green' from dual union
                select 3,'grase' from dual union
                select 7,'scott' from dual union
                select 8,'trumen' from dual union
                select 10,'kochar' from dual union
                select 12,'drejk' from dual union
                select 13,'kook' from dual) ,
--находим разницу между текущим и следующим значением айди
tab_2 as (select id,(lead(id) over (order by id) - id) as next_id from task_1),
--сдвигаем на 1 назад для актуальности текущего значения
tab_3 as (select id,lag(next_id) over (order by id) as dif from tab_2),
--новый столбец: если значение разницы  равно 1, не меняем группу(записываем ноль), иначе группу меняем (записываем единицу)
```

tab_4 as (select id, case when dif=1 then 0 else 1 end as subgr from tab_3)
--записываем накопительную сумму для найденных значений
select id, sum(subgr) over(order by id) as Gruppa from tab_4

Решение: (ОЮ) 9 вариант

Создадим таблицу TASK_16, удовлетворяющую условиям задачи, и добавим в неё строки из примера.

Запрос:

```
--Создаём таблицу
CREATE TABLE TASK_16(ID NUMBER(15, 0) PRIMARY KEY, LAST_NAME VARCHAR2(40));
--Вводим данные в таблицу
INSERT INTO TASK_16 VALUES(1, 'Mougus');
INSERT INTO TASK_16 VALUES(2, 'Green');
INSERT INTO TASK_16 VALUES(3, 'Grase');
INSERT INTO TASK_16 VALUES(7, 'Scott');
INSERT INTO TASK_16 VALUES(8, 'Trumen');
INSERT INTO TASK_16 VALUES(10, 'Kochhar');
INSERT INTO TASK_16 VALUES(12, 'Drejk');
INSERT INTO TASK_16 VALUES(13, 'Kook');
WITH
TEMP_TAB AS (--Определим первые элементы цепочек элементов D1
SELECT REGEXP_SUBSTR(ID_LIST, '^[[:digit:]]+') AS START_ID, ID_LIST
FROM (--Получим все возможные цепочки ID1, начинающиеся с элементов с флагом "1", элементы которых отличаются на единицу (в порядке возрастания ID1
SELECT LTRIM(SYS_CONNECT_BY_PATH(ID1, ', '), ', ') AS ID_LIST
FROM (--Получаем ID и соответствующие им флаги начала группы
SELECT ID AS ID1, LAG(ID, 1) OVER (ORDER BY RN) AS ID2,
--Определяем первые элементы групп: устанавливаем флаг "1", если элемент - первый в группе, иначе - "0".
(CASE WHEN ID - LAG(ID, 1) OVER (ORDER BY RN) = 1 THEN 0
ELSE 1 END) AS START_POS
FROM (SELECT ROWNUM AS RN, ID --Получаем все ID и номера соответствующих строк
FROM (SELECT ID --Сортируем ID по возрастанию
FROM TASK_16
ORDER BY ID ASC)))
START WITH START_POS = 1
CONNECT BY PRIOR ID1 = ID1 - 1))
--Определяем номера групп и элементы, которые в них входят
SELECT DISTINCT TO_NUMBER(TRIM(REGEXP_SUBSTR(ID_LIST, '[^,]+', 1, LEVEL))) AS ID, GROUP_NUM
FROM (--Определяем номера групп
SELECT ROWNUM AS GROUP_NUM, (ID_LIST || ',') AS ID_LIST
FROM (--Определяем цепочки максимальной длинны для кажддого начального элемента (t.START_ID)
SELECT DISTINCT TO_NUMBER(t.START_ID) AS START_ID,
(SELECT ID_LIST
FROM TEMP_TAB
WHERE START_ID = t.START_ID AND LENGTH(ID_LIST) = (--Определяем цепочки максимальной длинны для кажддого начального элемента (t.START_ID)
SELECT MAX(LENGTH(ID_LIST)) AS MAX_LEN
FROM TEMP_TAB
GROUP BY START_ID
HAVING START_ID = t.START_ID)) AS ID_LIST
FROM TEMP_TAB t)
ORDER BY 1)
CONNECT BY LEVEL <= LENGTH(REGEXP_REPLACE(ID_LIST, '[^,]+'))
```

```
ORDER BY 1 ASC;
```

Решение (Кож) 39в:

```
CREATE TABLE Task_16(Id number(15,0) Primary Key, Last_Name varchar2 (40));
INSERT INTO Task_16 VALUES(1,'Mougus');
INSERT INTO Task_16 VALUES(2,'Green');
INSERT INTO Task_16 VALUES(3,'Grase');
INSERT INTO Task_16 VALUES(7,'Scott');
INSERT INTO Task_16 VALUES(8,'Trumen');
INSERT INTO Task_16 VALUES(10,'Kochhar');
INSERT INTO Task_16 VALUES(12,'Drejk');
INSERT INTO Task_16 VALUES(13,'Kook');
WITH table_1 AS (SELECT id, last_name, rownum r1, id-rownum+1 r2
from Task_16 order by id),
--таблица с данными и начальным идентификатором групп
table_2 AS (SELECT r2 from table_1 group by r2),
--таблица с перечислением всех групп
table_3 AS (SELECT r2, rownum r3 from table_2)
--таблица с перечислением всех групп и соответствующих
--им новых идентификаторов
SELECT table_1.id, table_3.r3 AS "Номер группы"
from table_3
join table_1
on table_1.r2 = table_3.r2;
```

Решение:

```
CREATE TABLE "Customers" (
"Id" NUMBER(15,0) PRIMARY KEY,
"Last_Name" VARCHAR2(40)
);
INSERT ALL
INTO "Customers" VALUES (1, 'Mougus')
INTO "Customers" VALUES(2, 'Green')
INTO "Customers" VALUES(3, 'Grase')
INTO "Customers" VALUES(7, 'Scott')
INTO "Customers" VALUES(8, 'Trumen')
INTO "Customers" VALUES(10, 'Kochhar')
INTO "Customers" VALUES(12, 'Drejk')
INTO "Customers" VALUES(13, 'Kook')
SELECT * FROM "Customers";
commit;

WITH R("Id", N) AS
(SELECT "Id", 1
FROM "Customers"
WHERE "Id"=(SELECT MIN("Id") FROM "Customers") AND ROWNUM=1
UNION ALL
SELECT (SELECT "Id" FROM "Customers" WHERE "Id">R."Id" AND ROWNUM=1), CASE WHEN
EXISTS(SELECT 1 FROM "Customers" WHERE R."Id"+1="Id") THEN N+1 ELSE N END
FROM R
WHERE "Id" < (SELECT MAX("Id") FROM "Customers"))
SELECT *
FROM R;
```

4. Одной командой вывести все палиндромы, встречающиеся в произвольной символьной строке.

Например, для строки

aabacdca

ответ должен быть:

aa,aba, cdc, acdca

Решение:

Находим длину строки, для нее строим таблицу с начальной позицией возможного палиндрома и длиной этого палиндрома. Вроде. 1-2, 1-3... 1-8,2-2...7-2. Для каждой строки этой таблицы находим подстроку с [первое число] длинной [второе число]. Для этой строки функцией reverse находим обратную запись этой же строки. Если строка и обратная запись идентичны - выводим ее в ответ

Решение4:

```
DEFINE STR='aabacdca'
WITH SOURCE AS (
SELECT SUBSTR('&STR', LEVEL, 1) AS LT, ROW_NUMBER() OVER (ORDER BY LEVEL ASC) rn
FROM DUAL
CONNECT BY INSTR('&STR', SUBSTR('&STR', LEVEL, 1), 1)<LENGTH('&STR')),

TMP(WORD, l, rn) AS (
SELECT REPLACE(SYS_CONNECT_BY_PATH(LT, '.'), '.'), LEVEL l, ROWNUM
FROM SOURCE
WHERE LEVEL>1
CONNECT BY rn= PRIOR rn+1),

RESULT AS (
SELECT WORD AS W1,
(SELECT LISTAGG(REGEXP_SUBSTR(word, '.', LEVEL, 1), '')WITHIN GROUP (ORDER BY ROWNUM DESC)
FROM DUAL
CONNECT BY LEVEL<=LENGTH(WORD))AS W2
FROM TMP)

SELECT LISTAGG(W1, ',') WITHIN GROUP (ORDER BY 1) AS "Полиндромы"
FROM RESULT
WHERE W1=W2;
```

Решение5:

```
UNDEFINE STR;
WITH REVERSE_STR AS (SELECT LISTAGG(SUB,'') WITHIN GROUP (ORDER BY LEV)
FROM(SELECT SUBSTR('&&STR', LENGTH('&STR') - LEVEL + 1, 1) AS SUB, LEVEL AS LEV
                                    FROM DUAL
                                    CONNECT BY LEVEL <= LENGTH('&STR'))),

TWO_STR AS(SELECT LOWER('&STR') AS S1,
LOWER((SELECT * FROM REVERSE_STR)) AS S2, LEVEL AS LEV
                                FROM DUAL
                                CONNECT BY LEVEL <= LENGTH('&STR')),

PALINDROM AS(SELECT
CASE
    WHEN SUBSTR(S1, L1 - L3, L3 + 1) = SUBSTR(S2, LENGTH(S2) - L1 - L3, L3 + 1) THEN SUBSTR(S1, L1 -
L3, (L3 + 1) * 2)
    WHEN SUBSTR(S1, L1 - L2, L2) = SUBSTR(S2, LENGTH(S2) - L1 + 1 - L2, L2)
```

```
THEN SUBSTR('&STR', L1 - L2, 2 * L2 + 1)
      END AS STR
      FROM
(SELECT LEVEL L1 FROM DUAL CONNECT BY LEVEL <= LENGTH('&STR')) A
   INNER JOIN
(SELECT LEVEL L2 FROM DUAL CONNECT BY LEVEL <= LENGTH('&STR')) B
   ON (A.L1 - B.L2 > 0 AND A.L1 + B.L2 <= LENGTH('&STR'))
   INNER JOIN
(SELECT LEVEL - 1 L3 FROM DUAL CONNECT BY LEVEL <= LENGTH('&STR')) D
   ON (A.L1 - D.L3 > 0 AND A.L1 + D.L3 <= LENGTH('&STR'))
   INNER JOIN
(SELECT S1, S2, LEV FROM TWO_STR) C
   ON (A.L1 = C.LEV) )

SELECT LISTAGG(STRS,',') WITHIN GROUP (ORDER BY STRS) AS PALINDROMS
FROM(SELECT DISTINCT LOWER(STR) AS STRS
           FROM PALINDROM
           WHERE STR IS NOT NULL AND
    LENGTH(STR) = (SELECT MAX(LENGTH(STR)) FROM PALINDROM));
```

РЕШЕНИЕ ОТ НАШЕЙ ГРУППЫ:

```
with t as ( select 'aabacdca' str from dual),
letter as
(select rownum n, substr(str,level,1),str from t
connect by level<=length(str)),
sub as (
select rownum n, s,str from
(select distinct substr(str,n,level) s, str from letter
connect by level<=length(str)/2)),
rev(n,lev,s,r)as
(select n, length(s) lev,s, substr(s,length(s),1) r from sub
union all
select n, lev-1,s, r||substr(s,lev-1,1) r from rev
where lev>0),
revers as(
select n,r from rev
where lev=1
union
select n,substr(r,2)from rev
where lev=1 and length(r)>1),
res as
(select sub.s||revers.r result
from revers join sub on revers.n=sub.n
where regexp_like(sub.str,sub.s||revers.r))
select listagg(result,', ') within group(order by 1) result
from res;
```

5. Определить цифры, которые максимальное количество раз встречаются в столбце Phone_number таблицы Employees.

Пример результата:

MAX_CNT	NUM
212	1
212	4

Решение1:

```
WITH TAB1 AS(SELECT SUM(REGEXP_COUNT(PHONE_NUMBER,'0')) A,'0' B
FROM EMPLOYEES
UNION ALL
SELECT SUM(REGEXP_COUNT(PHONE_NUMBER,'1')),'1'
FROM EMPLOYEES
UNION ALL
SELECT SUM(REGEXP_COUNT(PHONE_NUMBER,'2')),'2'
FROM EMPLOYEES
UNION ALL
SELECT SUM(REGEXP_COUNT(PHONE_NUMBER,'3')),'3'
FROM EMPLOYEES
UNION ALL
SELECT SUM(REGEXP_COUNT(PHONE_NUMBER,'4')),'4'
FROM EMPLOYEES
UNION ALL
SELECT SUM(REGEXP_COUNT(PHONE_NUMBER,'5')),'5'
FROM EMPLOYEES
UNION ALL
SELECT SUM(REGEXP_COUNT(PHONE_NUMBER,'6')),'6'
FROM EMPLOYEES
UNION ALL
SELECT SUM(REGEXP_COUNT(PHONE_NUMBER,'7')),'7'
FROM EMPLOYEES
UNION ALL
SELECT SUM(REGEXP_COUNT(PHONE_NUMBER,'8')),'8'
FROM EMPLOYEES
UNION ALL
SELECT SUM(REGEXP_COUNT(PHONE_NUMBER,'9')),'9'
FROM EMPLOYEES)
SELECT A MAX_CNT, B NUM
FROM TAB1
WHERE A = (SELECT MAX(A) FROM TAB1);
```

Решение2:

```
SELECT SUM(REGEXP_COUNT(PHONE_NUMBER,'0')) AS "0",
SUM(REGEXP_COUNT(PHONE_NUMBER,'1')) AS "1",
SUM(REGEXP_COUNT(PHONE_NUMBER,'2')) AS "2",
SUM(REGEXP_COUNT(PHONE_NUMBER,'3')) AS "3",
SUM(REGEXP_COUNT(PHONE_NUMBER,'4')) AS "4",
SUM(REGEXP_COUNT(PHONE_NUMBER,'5')) AS "5",
SUM(REGEXP_COUNT(PHONE_NUMBER,'6')) AS "6",
SUM(REGEXP_COUNT(PHONE_NUMBER,'7')) AS "7",
SUM(REGEXP_COUNT(PHONE_NUMBER,'8')) AS "8",
SUM(REGEXP_COUNT(PHONE_NUMBER,'9')) AS "9"
FROM EMPLOYEES;
2)
WITH TAB1 AS(SELECT SUM(REGEXP_COUNT(PHONE_NUMBER,'0')) A,'0' B
FROM EMPLOYEES
UNION ALL
SELECT SUM(REGEXP_COUNT(PHONE_NUMBER,'1')),'1'
FROM EMPLOYEES
UNION ALL
```

```
SELECT SUM(REGEXP_COUNT(PHONE_NUMBER,'2')),'2'
FROM EMPLOYEES
UNION ALL
SELECT SUM(REGEXP_COUNT(PHONE_NUMBER,'3')),'3'
FROM EMPLOYEES
UNION ALL
SELECT SUM(REGEXP_COUNT(PHONE_NUMBER,'4')),'4'
FROM EMPLOYEES
UNION ALL
SELECT SUM(REGEXP_COUNT(PHONE_NUMBER,'5')),'5'
FROM EMPLOYEES
UNION ALL
SELECT SUM(REGEXP_COUNT(PHONE_NUMBER,'6')),'6'
FROM EMPLOYEES
UNION ALL
SELECT SUM(REGEXP_COUNT(PHONE_NUMBER,'7')),'7'
FROM EMPLOYEES
UNION ALL
SELECT SUM(REGEXP_COUNT(PHONE_NUMBER,'8')),'8'
FROM EMPLOYEES
UNION ALL
SELECT SUM(REGEXP_COUNT(PHONE_NUMBER,'9')),'9'
FROM EMPLOYEES)
SELECT A MAX_CNT, B NUM
FROM TAB1
WHERE A = (SELECT MAX(A) FROM TAB1);
3)
WITH TAB1 AS(
SELECT SUBSTR('0123456789',LEVEL,1) A
FROM DUAL
CONNECT BY LEVEL <11
),
TAB2 AS(SELECT (SELECT SUM(REGEXP_COUNT(phone_number, a)) FROM EMPLOYEES) B,A
FROM TAB1)
SELECT B, A
FROM TAB2
WHERE B = (SELECT MAX(B) FROM TAB2);
```

РЕШЕНИЕ НАИБОЛЕЕ НОРМАЛЬНОЕ

```
with numb as
(select regexp_substr('0123456789',level,1) as numbs
from dual
connect by level<=10),
tabl as (select (select sum(regexp_count(phone_number,numbs)) from employees) rez,numbs
from numb)
select rez,numbs from tabl where rez=(select max(rez) from tabl);
```

6. Задана произвольная символьная строка, состоящая из двух частей, разделенных символами «=>». В левой и правой части выражения содержатся символьные строки, разделенные запятыми.

Требуется создать запрос, который будет выводить все возможные пары комбинаций из левой и правой частей.

Пример результата для строки a, fgf,yy=>uu,gh:

PATH
a=>uu
a=>gh
fgf=>uu
fgf=>gh
yy=>uu
yy=>gh

РЕШЕНИЕ С MODEL

```
define text = 'a, fgf,yy=>uu,gh';
WITH
/*Вырезаем левую часть*/
subtext1 AS(
SELECT trim(SUBSTR('&text', 1, INSTR('&text', '=>')-1)) subtext_1
from dual),
/*Вырезаем правую часть*/
subtext2 AS(
SELECT trim(SUBSTR('&text', INSTR('&text', '=>')+2, length('&text')-INSTR('&text', '=>')-1)) subtext_2
from dual),
/*определяем начало, конец, кол-во элементов для каждой подстроки*/
model_1 AS(
SELECT
subtext_1 AS orig_str1 ,
','|| subtext_1|| ',' AS mod_str1 ,
1 AS start_pos1 ,
Length(subtext_1) AS end_pos1 ,
(LENGTH(subtext_1) - LENGTH(REPLACE(subtext_1, ','))) + 1 AS element_count1 ,
0 AS element_no1
FROM subtext1),
model_2 AS(
SELECT
subtext_2 AS orig_str2 ,
','|| subtext_2|| ',' AS mod_str2 ,
1 AS start_pos2 ,
Length(subtext_2) AS end_pos2 ,
(LENGTH(subtext_2) - LENGTH(REPLACE(subtext_2, ','))) + 1 AS element_count2 ,
0 AS element_no2
FROM subtext2),
/*Каждую подстроку разбиваем на слова и заносим их по отдельности в таблицу*/
variants_1 AS
(SELECT
trim(Substr(mod_str1, start_pos1, end_pos1-start_pos1)) text
FROM (
SELECT *
FROM model_1
MODEL PARTITION BY ( orig_str1, mod_str1)
DIMENSION BY (element_no1)
MEASURES (start_pos1, end_pos1, element_count1)
RULES ITERATE (2000)
UNTIL (ITERATION_NUMBER+1 = element_count1[0])
```

```
( start_pos1[ITERATION_NUMBER+1] =
instr(cv(mod_str1), ',', 1, cv(element_no1)) + 1,
end_pos1[ITERATION_NUMBER+1] =
instr(cv(mod_str1), ',', 1, cv(element_no1) + 1) )
)
WHERE element_no1 != 0
ORDER BY mod_str1 , element_no1),
variants_2 AS
(SELECT
trim(Substr(mod_str2, start_pos2, end_pos2-start_pos2)) text
FROM (
SELECT *
FROM model_2
MODEL PARTITION BY ( orig_str2, mod_str2)
DIMENSION BY (element_no2)
MEASURES (start_pos2, end_pos2, element_count2)
RULES ITERATE (2000)
UNTIL (ITERATION_NUMBER+1 = element_count2[0])
( start_pos2[ITERATION_NUMBER+1] =
instr(cv(mod_str2), ',', 1, cv(element_no2)) + 1,
end_pos2[ITERATION_NUMBER+1] =
instr(cv(mod_str2), ',', 1, cv(element_no2) + 1) )
)
WHERE element_no2 != 0
ORDER BY mod_str2 , element_no2)
/*Составляем все возможные варианты*/
SELECT uno.text||'=>' || due.text AS PATH
FROM variants_1 uno, variants_2 due
ORDER BY PATH;
```

Решение2: ВСТАВИЛА НАША ГРУППА

```
with t as (
select 'a,fgf,yy=>uu,gh' str from dual),
t1 as (
select regexp_substr(str, '[^=>]+', 1, 1) s1, regexp_substr(str, '[^=>]+', 1, 2) s2
from t),
t2 as (
select regexp_substr(s1, '[^,]+', 1, level) ss1, s2, level l
from t1
connect by level <=regexp_count(s1, ',')+1),
t3  as (
select ss1, regexp_substr(s2, '[^,]+', 1, level) ss2
from t2
connect by level <=regexp_count(s2, ',')+1)
select distinct ss1 || '=>' || ss2
from t3
order by 1
```

Решение3: (ОЮ) НАША ГРУППА ЕГО СЕБЕ ВСТАВИЛА

```
DEFINE &&STR;
SELECT A|| '=>' || B "PATH"
FROM(
SELECT TRIM(regexp_substr(SUBSTR('&STR', 1, INSTR('&STR', '=>')-1), '[^,]+', 1, level)) A
FROM DUAL
```

```
connect by regexp_substr(SUBSTR('&STR', 1, INSTR('&STR', '=>')-1), '[^,]+', 1, level) is not null
)
CROSS JOIN (
SELECT TRIM(regexp_substr(SUBSTR('&STR', INSTR('&STR', '=>')+2), '[^,]+', 1, level)) B
FROM DUAL
CONNECT BY regexp_substr(SUBSTR('&STR', INSTR('&STR', '=>')+2), '[^,]+', 1, level) is not null
);
UNDEFINE STR;
```

Алгоритм:

Решение сводится к тому, чтобы отделить левую и правую части выражения, в каждой из них выделить подстроки, отделенные запятыми. Эти подстроки будут выводиться в подзапросах в виде одного столбика. Затем найдем Cortesion product двух столбцов и соединим значения в них через символы “=>”.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ 2

1. Для каждого отдела вывести фамилии и зарплаты трех сотрудников, получающих самые высокие зарплаты в отделе. Если самую низкую зарплату у найденных трех сотрудников отдела получают и какие-то другие сотрудники этого отдела, они тоже должны попасть в список.

Решение из группы:

```
with t as(
select department_id, last_name, salary,
count(employee_id) over (partition by department_id) cnt,
dense_rank() over (partition by department_id order by salary desc) dr
from employees
where department_id is not null
)
select department_id, last_name, salary from t
where cnt>=3 and dr<=3;
```

Решение с eksam(8):

Решение:

```
WITH FRST_3 AS ( —Первые 3
SELECT DEPARTMENT_ID, LAST_NAME, SALARY, RNK
FROM (
SELECT DEPARTMENT_ID, LAST_NAME, SALARY,
ROW_NUMBER() OVER (PARTITION BY DEPARTMENT_ID ORDER BY SALARY) AS RNK
FROM EMPLOYEES)
WHERE RNK <= 3),

LIKE_THIRD AS ( —Такие же, как третий
SELECT DISTINCT DEPARTMENT_ID, EMPLOYEES.LAST_NAME AS LST_NAME, SALARY
FROM
(SELECT DEPARTMENT_ID, LAST_NAME, SALARY
FROM FRST_3
WHERE RNK = 3) RNK_3 JOIN EMPLOYEES USING(DEPARTMENT_ID, SALARY))

SELECT DEPARTMENT_ID, LAST_NAME, SALARY
FROM FRST_3
UNION
SELECT DEPARTMENT_ID, LST_NAME, SALARY
FROM LIKE_THIRD
```

2. Создать запрос для разделения "задвоенных" данных. Например, из

CODE_OPERATION	ID_CLIENT
1000 1100	841000 841100
2000	6700 8967 5500

сделать

RN	CNT	CODE_OPERATION	ID_CLIENT
1	0	1000 1100	841000 841100
	1	1000	841000
	2	1100	841100
2	0	2000	6700 8967 5500
	1	2000	6700
	2		8967
	3		5500

Задачу решить с использованием раздела Model.

Решение без model:

```
create table DoubleValues(C1 VARCHAR2(20), C2 VARCHAR2(20));
insert into DoubleValues values('1000 1100', '841000 841100');

with DRN as (SELECT CODE_OPERATION, ID_CLIENT, rownum rn FROM DoubleValues),
t as
(SELECT rn, (rn - 1) as CNT, CODE_OPERATION, ID_CLIENT FROM DRN
UNION
SELECT rn, rn, rtrim(rtrim(CODE_OPERATION, '0123456789'), ' '), rtrim(rtrim(ID_CLIENT, '0123456789'), '
') FROM DRN
UNION
SELECT rn, rn + 1, ltrim(ltrim(CODE_OPERATION, '0123456789'), ' '), ltrim(ltrim(ID_CLIENT,
'0123456789'), ' ') FROM DRN)
SELECT rn, CNT, CODE_OPERATION, ID_CLIENT FROM t order by rn,CNT;
```

Решение2:

```
with rep as(
select rownum rn, code_operation, id_client
from(
select '1000 1100' code_operation, '841000 841100' id_client from dual
order by 2))
/*Ввод значений*/
select rn, row_number() over (partition by rn order by length(code_operation)desc)-1 cnt,
code_operation, id_client
from(
select code_operation, id_client, rn /*Первая строка – вывод исходных данных как и было*/
from rep
union all
select regexp_substr(code_operation, '\S*'), regexp_substr(id_client, '\S*'), rn
/*Отсекаем первую часть до пробела*/
from rep
union all
select
regexp_substr(code_operation, '\S*$'), regexp_substr(id_client, '\S*$'), rn
/*Отсекаем вторую часть с конца до пробела*/
from rep
order by 3);
```

Решение3 с ьщвуд:

```
WITH T AS (
   SELECT         '1000 1100' AS code_operation, '841000 841100'  AS id_client FROM dual
   UNION ALL SELECT '2000'                    , '6700 8967 5500' FROM dual
)
SELECT
   CASE WHEN cnt = 0 THEN rn || ' '
      ELSE ' ' END RN,
   cnt,
   nvl(cur_code, ' ') AS code_operation,
   NVL(clid, ' ') AS ID_CLIENT
FROM(
SELECT
   *
FROM
   (
      SELECT
         ROWNUM rn,
         T.*
      FROM T
   )
MODEL
   DIMENSION BY (rn, 0 AS cnt, code_operation, id_client)
   MEASURES(code_operation AS cur_code, id_client AS clid, 0 AS cur_row)
   RULES UPSERT ALL ITERATE (1000)
   (
      cur_code[ANY, iteration_number + 1, ANY, ANY] = regexp_substr(cur_code[cv(), 0, cv(), cv()], '\d+',
1, iteration_number + 1),
      clid[ANY, iteration_number + 1, ANY, ANY] = regexp_substr(clid[cv(), 0, cv(), cv()], '\d+', 1,
iteration_number + 1)
   )
ORDER BY rn, cnt)
WHERE cur_code IS NOT NULL OR clid IS NOT NULL OR CNT = 0;
```

Еще одно решение с модел из группы:

```
with t as(
select '1000 1100' CODE_OPERATION, '841000 841100' ID_CLIENT from dual
union all
select '2000 2000', '6700 8967 5500' from dual
union all
select '2000', '6700 8967 5500' from dual),
t1 as(
select rownum rn,
trim(both ' ' from CODE_OPERATION) CODE_OPERATION,
trim(both ' ' from  ID_CLIENT) ID_CLIENT from t),
t2 as(
select distinct rn, level cnt, regexp_substr(code_operation, '([^ ])+',1,level) cod from t1
    connect by level<=regexp_count(code_operation,' ')+1),
t3 as(
 select distinct rn, level cnt ,regexp_substr(ID_CLIENT, '([^ ])+',1,level) cid from t1
    connect by level<=regexp_count(ID_CLIENT,' ')+1),
joined as(
select t2.rn rn1, t3.rn rn2, t2.cnt cnt1, t3.cnt cnt2, t2.cod, t3.cid
from t2 full outer join t3 on t2.rn=t3.rn and t2.cnt=t3.cnt
union
select rn, 0, 0, 0, CODE_OPERATION, ID_CLIENT from t1),
```

```
mod as(
select rn, cnt, cod, cid
from joined
model
dimension by(rn1, rn2, cnt1, cnt2)
measures(rn1 rn, cnt1  cnt, cod,cid)
rules(
rn[null,any,any,any]=cv(rn2),
cnt[any,any,null,any]=cv(cnt2),
cod[any,any,any,any]=nvl(cod[cv(),cv(),cv(),cv()],' '),
cid[any,any,any,any]=nvl(cid[cv(),cv(),cv(),cv()],' ')
)
order by rn,cnt)
select decode(cnt,0,to_char(rn),' ') rn,cnt,cod code_operation,cid id_client
from mod;

with t as(
select '1000 1100' CODE_OPERATION, '841000 841100' ID_CLIENT from dual
union all
select '2000', '6700 8967 5500' from dual),
t1 as(
select rownum rn,
trim(both ' ' from CODE_OPERATION) CODE_OPERATION,
trim(both ' ' from  ID_CLIENT) ID_CLIENT from t),
mod as(
select distinct rn, r, cnt, cod code_operation, cid  id_client
from
   (select distinct rn, 1 tab, level cnt, regexp_substr(code_operation, '([^ ])+',1,level) cod,"cid
from t1
   connect by level<=regexp_count(code_operation,' ')+1
      union all
   select distinct rn, 2 tab, level cnt ,"cod, regexp_substr(ID_CLIENT, '([^ ])+',1,level) cid from t1
   connect by level<=regexp_count(ID_CLIENT,' ')+1
      union all
   select t1.rn, 0 tab, 0, t1.CODE_OPERATION, t1.ID_CLIENT from t1)
model return updated rows
partition by (rn)
dimension by(tab ,cnt)
measures (' ' r, cod,cid)
rules(
r[0,0]=to_char(cv(rn)),
cod[2,any]=nvl(cod[1,cv()],' '),
cid[1,any]=nvl(cid[2,cv()],' ')
)
order by rn, cnt)

select r rn, cnt, code_operation, id_client from mod;
```

3. Используя словарь данных, получить информацию об ограничениях CHECK схемы:
   В списках имена столбцов вывести через запятую. Имя таблицы не должно повторяться.
   Задачу решить без использования функций Listagg и Wm_concat.
   Пример представления результатов:

SQL | All Rows Fetched: 8 in 1,748 seconds

	Имя таблицы	Имя ограничения	Столбцы,входящие в ограничение	Ограничение CHECK
1	EMPLOYEES	EMP_SALARY_MIN	SALARY	salary > 0
2	JOB_HISTORY	JHIST_DATE_INTERVAL	END_DATE,START_DATE	end_date > start_date
3	TASK_13	COLUMN12345	COLUMN_1,COLUMN_2,COLUMN_3,COLUMN_4,COLUMN_5	column_1>ALL (column_2, column_3, column_4, column_5)
4	TRIP	CON_PER_BEG	PER_BEG	per_beg=TRUNC(per_beg)
5		CON_PER_END	PER_END	per_end=TRUNC(per_end)
6	ПРЕПАРАТ	ПРЕПАРАТ_CHK1	АББРЕВ_ФОРМЫ,ТИП	Аббрев_формы!=Тип
7	ПРЕПОДАВАТЕЛЬ	SYS_C00201812	КАФЕДРА	КАФЕДРА Like 'Кафедра%'
8	УСПЕВАЕМОСТЬ	SYS_C00201820	ОЦЕНКА	ОЦЕНКА between 1 and 5

Решение:

```
WITH T AS (
SELECT UC.TABLE_NAME TABLE_NAME, UC.CONSTRAINT_NAME CONSTRAINT_NAME,
UCC.COLUMN_NAME CN, UC.SEARCH_CONDITION SC,
ROW_NUMBER() OVER (PARTITION BY UC.TABLE_NAME, UC.CONSTRAINT_NAME ORDER BY
UCC.COLUMN_NAME) RN,
ROW_NUMBER() OVER (PARTITION BY UC.TABLE_NAME ORDER BY UCC.COLUMN_NAME) RN_T
FROM USER_CONSTRAINTS UC LEFT JOIN
USER_CONS_COLUMNS UCC ON UC.CONSTRAINT_NAME = UCC.CONSTRAINT_NAME
WHERE UC.CONSTRAINT_TYPE='C')
SELECT
CASE WHEN RN_T=1 THEN TABLE_NAME
ELSE ' ' END TABLE_NAME,
CONSTRAINT_NAME, LTRIM(SYS_CONNECT_BY_PATH(CN, ','), ',') CN, SC
FROM T
WHERE CONNECT_BY_ISLEAF=1
START WITH RN = 1
CONNECT BY PRIOR TABLE_NAME = TABLE_NAME AND PRIOR CONSTRAINT_NAME =
CONSTRAINT_NAME AND PRIOR RN + 1 = RN;)
```

Решение2:

```
WITH TNAME AS
    (SELECT TABLE_NAME
     FROM USER_CONSTRAINTS),
   CHK_COL AS
    (SELECT R1.TABLE_NAME, R2.COLUMN_NAME, LAG(R2.COLUMN_NAME) OVER (PARTITION BY
R1.TABLE_NAME, R2.CONSTRAINT_NAME ORDER BY R2.COLUMN_NAME) AS LAG_COL,
          R1.CONSTRAINT_NAME, R1.SEARCH_CONDITION_vc
     FROM USER_CONSTRAINTS R1 JOIN USER_CONS_COLUMNS R2 ON
(R1.CONSTRAINT_NAME=R2.CONSTRAINT_NAME AND R1.TABLE_NAME=R2.TABLE_NAME)
     WHERE CONSTRAINT_TYPE = 'C'),

   CHK_COL_LISTAGG AS (
    SELECT TABLE_NAME, TRIM(LEADING ',' FROM SYS_CONNECT_BY_PATH(COLUMN_NAME, ','))
COLUMN_NAME, LEVEL LEV,  CONSTRAINT_NAME, SEARCH_CONDITION_vc
    FROM CHK_COL
    START WITH LAG_COL IS NULL
    CONNECT BY NOCYCLE PRIOR COLUMN_NAME = LAG_COL AND PRIOR TABLE_NAME=TABLE_NAME
),

   CHK_COL_RES AS
    (SELECT TABLE_NAME, COLUMN_NAME, CONSTRAINT_NAME, SEARCH_CONDITION_vc
     FROM CHK_COL_LISTAGG R3
     WHERE LEV = (SELECT MAX(LEV)
            FROM CHK_COL_LISTAGG R4
            WHERE R3.TABLE_NAME=R4.TABLE_NAME
            GROUP BY TABLE_NAME))

SELECT distinct TABLE_NAME,CONSTRAINT_NAME, COLUMN_NAME, SEARCH_CONDITION_vc
     FROM CHK_COL_RES;
```

4. Предполагая, что не существует зарплаты сотрудников (таблица Employees), большей 100000, для каждого сотрудника, имеющего более двух подчиненных, вывести представление зарплаты в десятичной и двоичной системах счисления (без ведущих нулей).

Решение: (ОЮ) вставила группа

```
WITH Bin(num, salary, gr1, gr2, "list") AS(
SELECT r1.Employee_ID num, r1.Salary, r1.Salary gr1, floor(r1.Salary/2) gr2, TO_CHAR(mod(r1.Salary, 2))
"list"
FROM Employees r1
WHERE r1.Salary <= 100000
UNION ALL
SELECT num, Salary, gr2, floor(gr2/2),
mod(gr2, 2) || "list"
FROM Bin
WHERE gr1 >0
),
BinResult AS(
SELECT num Employee_ID, SALARY, SUBSTR("list", INSTR("list", '1', 1, 1)) AS "Binary salary"
FROM Bin
WHERE gr1=0
) SELECT b.Employee_ID, b.SALARY, "Binary salary"
FROM BinResult b
INNER JOIN Employees L ON (L.manager_id = b.Employee_ID)
GROUP BY b.Employee_ID, b.SALARY, "Binary salary"
HAVING count(L.employee_ID) > 2;
```

Алгоритм:

1) Сперва построим рекурсивный запрос Bin, выводящий порядок перевода зарплаты каждого сотрудника в бинарную систему. Gr1=0 обозначает, что перевод закончен, и строка list – итоговый результат, который нам и нужен (Визуализируем этот промежуточный результат с помощью запроса SELECT * FROM Bin)

2) Отберем эти строки в запросе BinResult. Обрежем результат, убрав ведущие нули.

3)Наконец, в главном запросе сделаем INNER JOIN с Employees, получая тех сотрудников, чьими менеджерами являются наши сотрудники из BinResult. Выберем тех менеджеров, у которых в подчинении более двух сотрудников.

Решение2:

```
WITH T AS (
SELECT E.Employee_ID, E.Salary
FROM Employees E JOIN Employees M
ON M.manager_id = E.Employee_ID
GROUP BY E.Employee_ID, E.Salary
HAVING count(E.employee_ID) > 2),
R(ID, SAL, PART, NUM) AS (
SELECT Employee_ID ID, Salary SAL, floor(Salary/2) PART, mod(Salary, 2) NUM
FROM T
WHERE Salary <= 100000
UNION ALL
SELECT ID, SAL, floor(PART/2), mod(PART, 2)
FROM R
WHERE floor(PART/2)>0),
RES AS (
SELECT ID, SAL, PART, NUM,
ROW_NUMBER() OVER (PARTITION BY ID ORDER BY PART) RN
FROM R
ORDER BY 1)  SELECT distinct ID, SAL,
```

```
(select min(part) from res where res.id=id) ||
listagg(NUM, '') WITHIN GROUP (order by rn) OVER (partition by ID) binary_sal
FROM RES;
```

5. Используя обращение только к таблице DUAL, построить SQL-запрос, возвращающий один столбец, содержащий календарь на заданный месяц заданного года:
   - номер дня в месяце (две цифры),
   - полное название месяца по-английски заглавными буквами (в верхнем регистре),
   - год (четыре цифры),
   - полное название дня недели по-английски строчными буквами (в нижнем регистре).

   Каждое "подполе" должно быть отделено от следующего одним пробелом. В результате не должно быть начальных и хвостовых пробелов. Количество возвращаемых строк должно точно соответствовать количеству дней в текущем месяце. Строки должны быть упорядочены по номерам дней в месяце по возрастанию.

Решение: Кож 33в

```
SELECT
TO_CHAR(LEVEL-1+TRUNC(SYSDATE, 'MM'), -- Каждый день это LEVEL + дата начала месяца
    'DD fmMONTH YYYY day', -- fm удаляет лишние пробелы спереди или сзади
    'NLS_DATE_LANGUAGE = American') -- Устанавливает язык Английский
    AS "Календарь"
FROM DUAL
CONNECT BY LEVEL<=EXTRACT(DAY FROM LAST_DAY(SYSDATE)); -- Пока не дойдём до последнего дня в месяце
```

Решение2: (ОЮ)

```
SELECT (TO_CHAR(s,'fmDD'))
  || ' '
  || TO_CHAR(s,'fmMONTH','NLS_DATE_LANGUAGE=AMERICAN')
  || ' '
  || TO_CHAR(s,'fmYYYY')
  || ' '
  || TO_CHAR(s,'fmday','NLS_DATE_LANGUAGE=AMERICAN') "Календарь"
FROM
 (SELECT TRUNC(sysdate,'MONTH')+level-1 s
 FROM dual
   START WITH level=0
   CONNECT BY level<1+extract( DAY FROM last_day(TRUNC(sysdate,'MONTH')))
 ) ;
```

Алгоритм2:

С помощью connect_by составим цикл до 1 числа следующего месяца (от текущей даты) и используя маски форматов и параметры nls формируем нужный вывод

Решение3:

```
SELECT TO_CHAR(CAL,
 'DD FMMONTH YYYY day',
 'NLS_DATE_LANGUAGE = ENGLISH') AS MONTHLY_CALENDAR
FROM(SELECT TRUNC(SYSDATE, 'MONTH') + LEVEL - 1 AS CAL
          FROM DUAL
          CONNECT BY LEVEL <= TO_CHAR(LAST_DAY(SYSDATE), 'DD'))
ORDER BY SUBSTR(MONTHLY_CALENDAR,1,2) ASC;
```

Решение4:

```
/*при помощи средств connect by level, выводится столбец чисел
от 1 до последнего дня в месяц, при помощи функции LPAD к цифрам с лева добавляется нолик*/
```

```
/*затем приклеивается месяц,год,день недели, все отформатированное должным образом.
*/
SELECT LPAD(TO_CHAR(ROWNUM),2,'0') ||' '||
RTRIM(TO_CHAR(sysdate,'MONTH','NLS_DATE_LANGUAGE = ENGLISH'))||'
'||TO_CHAR(sysdate,'YYYY')||' '||
      RTRIM(LOWER( TO_char(TO_DATE(rownum,'DD'),'DAY','NLS_DATE_LANGUAGE = ENGLISH'))
) "Day"
FROM DUAL
connect by level <= TO_CHAR(last_day(sysdate),'DD')
```

Решение5:

```
select
to_char(trunc(sysdate, 'mm')+level-1, 'dd FMMONTH yyyy day', 'nls_date_language=english') result
from dual
connect by level <= to_number(to_char(last_day(sysdate), 'dd'))
order by to_number(substr(result, 1,2));
```

Решение6: (ОЮ)

```
SELECT TRIM(EXTRACT(DAY FROM TRUNC(SYSDATE,'MONTH')+LEVEL-1)||' '||--номер дня, начиная с
первого в месяце
TO_CHAR(SYSDATE,'MONTH','NLS_DATE_LANGUAGE=ENGLISH')||' '||--определяем месяц
EXTRACT(YEAR FROM SYSDATE)||' '||--определяем год
TO_CHAR(TRUNC(SYSDATE,'MONTH')+LEVEL-1,'day','NLS_DATE_LANGUAGE=ENGLISH'))
"Календарь"--день недели
FROM DUAL
CONNECT BY LEVEL<=LAST_DAY(SYSDATE)-TRUNC(SYSDATE,'MONTH')+1;--определяем кол-о дней в
месяце
```

Решение: (ОЮ)

```
SELECT TO_CHAR(TRUNC(SYSDATE,'MON')+LEVEL-1,'ddfm MONTH yyyy
day','NLS_DATE_LANGUAGE=ENGLISH') AS CALENDAR
FROM DUAL
CONNECT BY LEVEL<=TO_CHAR(LAST_DAY(SYSDATE),'dd');
```

Решение (кож) 51в

```
alter session set nls_language = 'American';
select substr(to_date(level, 'dd'),1,2) ||' '||to_char(sysdate, 'fmMONTH')||' '||
to_char(sysdate, 'yyyy')||' '||to_char(to_date(level, 'dd'), 'day') "Календарь"
from dual
connect by level <= to_char(last_day(sysdate), 'dd');
```

ИЗ ГРУППЫ:

```
WITH T as(
select to_date('1.2020','mm.syyyy') d from dual)
SELECT
TO_CHAR(LEVEL-1+TRUNC(d, 'MM'),'DD fmMONTH YYYY day',
'NLS_DATE_LANGUAGE = American')
AS "Календарь"
FROM T
CONNECT BY LEVEL<=EXTRACT(DAY FROM LAST_DAY(d));
```

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ 3

2. Для произвольной команды SELECT определить список входящих в нее таблиц (через запятую) с указанием имени схемы. Задачу решить одной командой SELECT.
   Например, для команды:

```
WITH "СР ПО ОТД" AS (
SELECT DEPARTMENT_ID,AVG(SALARY) AS ASAL
```

```
FROM hr.EMPLOYEES
GROUP BY DEPARTMENT_ID),
"НАИБ БЛИЗ" AS (
SELECT DEPARTMENT_ID, MIN(ABS(SALARY - ASAL)) AS MINSAL
FROM EMPLOYEES JOIN "СР ПО ОТД" USING (DEPARTMENT_ID)
GROUP BY DEPARTMENT_ID)
SELECT EMPLOYEE_ID AS "Номер",LAST_NAME AS "Фамилия",JOB_ID AS "Должность",
DEPARTMENT_ID AS "Отдел",SALARY AS "Оклад", TRUNC(ASAL) AS "Средний оклад"
FROM EMPLOYEES JOIN "MY JOBS" USING (JOB_ID)
JOIN "СР ПО ОТД" USING (DEPARTMENT_ID) JOIN "НАИБ БЛИЗ"
USING (DEPARTMENT_ID)
WHERE (DEPARTMENT_ID, ABS(SALARY - ASAL)) IN
(SELECT DEPARTMENT_ID, MINSAL FROM "НАИБ БЛИЗ")
ORDER BY DEPARTMENT_ID, SALARY, LAST_NAME;
```

результат должен быть:

hr.EMPLOYEES,os.EMPLOYEES,os."MY JOBS"

Решение: ОЮ 26в

```
--вводим исходую строку
WITH src as(
  SELECT q'{
    WITH "СР ПО ОТД" AS (      SELECT DEPARTMENT_ID,AVG(SALARY) AS ASAL
    FROM hr.EMPLOYEES
    GROUP BY DEPARTMENT_ID),
    "НАИБ БЛИЗ" AS (
    SELECT DEPARTMENT_ID, MIN(ABS(SALARY - ASAL)) AS MINSAL
    FROM EMPLOYEES JOIN "СР ПО ОТД" USING (DEPARTMENT_ID)
    GROUP BY DEPARTMENT_ID)
    SELECT EMPLOYEE_ID AS "Номер",LAST_NAME AS "Фамилия",JOB_ID AS "Должность",
    DEPARTMENT_ID AS "Отдел",SALARY AS "Оклад", TRUNC(ASAL) AS "Средний оклад"
    FROM EMPLOYEES JOIN "MY JOBS" USING (JOB_ID)
    JOIN "СР ПО ОТД" USING (DEPARTMENT_ID) JOIN "НАИБ БЛИЗ"
    USING (DEPARTMENT_ID)
    WHERE (DEPARTMENT_ID, ABS(SALARY - ASAL)) IN
    (SELECT DEPARTMENT_ID, MINSAL FROM "НАИБ БЛИЗ")
    ORDER BY DEPARTMENT_ID, SALARY, LAST_NAME;
  }' AS str
  FROM dual
),
--все таблицы, которые фигурируют в строке-запросе
tabs AS (
  SELECT REGEXP_SUBSTR(str, '(".+?")|([[:alnum:]_.]*)',
    REGEXP_INSTR(str, '[[:space:]](FROM|JOIN)[[:space:]]', 1, level, 1, 'i'),
    1) AS table_name
  FROM src
  CONNECT BY level<=REGEXP_COUNT(str, '[[:space:]](FROM|JOIN)[[:space:]]', 1, 'i')
),
--если имя таблицы с кавычками, то оно регистрочуствительное, иначе - оно
--записано в словарях данных в высоком регистре
tabs_fix AS (
  SELECT (CASE WHEN instr(tabs.table_name,'"')=0 THEN upper(tabs.table_name)
       ELSE replace(tabs.table_name,'"','') END) table_name
  FROM tabs
```

```
),
 --проверяем, чем является найденный объект и в зависимсоти от этого решаем, выводить его
или нет
tables AS(
SELECT DISTINCT
   CASE
    WHEN tab.table_name LIKE '%.%' or upper(tab.table_name) = 'DUAL'
     THEN tab.table_name
    WHEN syn.synonym_name IS NOT NULL
     THEN
     (SELECT USER FROM dual)||'.'||syn.table_name
    WHEN tab.table_name IN
     (SELECT table_name FROM user_tables UNION
     SELECT table_name FROM dict)
    THEN
     (SELECT USER FROM dual)
     ||'.'
     ||tab.table_name
   END table_name
 FROM tabs_fix tab
 LEFT JOIN user_synonyms syn
 ON tab.table_name = syn.synonym_name)
SELECT listagg(table_name,',')  within GROUP(order by table_name ) Таблицы
FROM tables
WHERE table_name IS NOT NULL;
```

Решение:

```
WITH
VAL as (select '--&VAL' st from dual),
TAB1(st1) AS
(SELECT SUBSTR(St,regexp_INSTR(ST, '([[:space:]]JOIN[[:space:]|[:punct:]])', 1,1,1,'i'))
FROM VAL
UNION ALL
SELECT SUBSTR(ST1,regexp_INSTR(ST1, '([[:space:]]JOIN[[:space:]|[:punct:]])', 1,1,1,'i'))
FROM TAB1 where regexp_INSTR(ST1, '([[:space:]]JOIN[[:space:]|[:punct:]])', 1,1,1,'i') <> 0
),
TAB2(st1) as (
SELECT SUBSTR(St,regexp_INSTR(ST, '([[:space:]]FROM[[:space:]|[:punct:]])', 1,1,1,'i'))
FROM VAL
UNION ALL
SELECT SUBSTR(ST1,regexp_INSTR(ST1, '([[:space:]]FROM[[:space:]|[:punct:]])', 1,1,1,'i'))
FROM TAB2 where regexp_INSTR(ST1, '([[:space:]]FROM[[:space:]|[:punct:]])', 1,1,1,'i') <> 0
),
TAB3 as (
 select * from TAB1
 union all
 select * from TAB2
),
 TAB4 AS
 (SELECT
   CASE
    WHEN st1 LIKE '"%'
    THEN SUBSTR(st1,1,instr(st1, '"',1,2))
    ELSE (
     CASE
```

```
        WHEN regexp_instr(st1, '[[:space:]|;]') <>0
        THEN SUBSTR(st1,1,regexp_instr(st1, '[[:space:]|;]')-1 )
        ELSE st1
       END)
   END table_name
  FROM TAB3),
tables AS(
SELECT DISTINCT
   CASE
    WHEN TAB4.table_name LIKE '%.%' or upper(TAB4.table_name) = 'DUAL'
    THEN TAB4.table_name
    WHEN TAB4.table_name IN
     (SELECT synonym_name FROM user_synonyms)
    THEN
     user_synonyms.table_name
    WHEN upper(TAB4.table_name) IN
     (SELECT upper(table_name) FROM user_tables) or upper(TAB4.table_name) IN
     (SELECT upper(table_name) FROM dict)
    THEN
     TAB4.table_name
   END table_name
  FROM TAB4
  LEFT JOIN user_synonyms
  ON TAB4.table_name = user_synonyms.synonym_name)
SELECT listagg(table_name,',')  within GROUP(order by table_name ) Таблицы
FROM tables
WHERE table_name IS NOT NULL;
```

2. Имеется таблица с колонкой, которая содержит множество значений, разделенных запятыми. Требуется создать запрос, который каждое значение выведет на отдельной строке. Например, дана таблица:

Номер	Телефон
952240	2-78,2-89
952423	2-78,2-83,8-34

Результат:

Номер	Телефон
952240	2-78
	2-89
952423	2-78
	2-83
	8-34

Задачу решить с использованием раздела Model.

РЕШЕНИЕ

with src as (select

'952565' as action,'2-78,2-89' code from dual union all

```
select '952423' as action,'2-78,2-83,8-34' code from dual
), --задаем значения
SEARCH_REG(N_LIST, N_POS, PART_ACT, PART_CODE, ACTION, CODE) AS (
--выполняем поиск подстрок для последующего разделения
SELECT TO_CHAR(ROWNUM) N_LIST, 1 N_POS, '' PART_ACT, '' PART_CODE, ACTION, CODE
FROM SRC
UNION ALL
SELECT N_LIST, N_POS+1, REGEXP_SUBSTR(ACTION, '\d+-+\d+', 1, N_POS), REGEXP_SUBSTR(CODE, '\d+-+\d+', 1, N_POS), ACTION, CODE --собственно части строки
FROM SEARCH_REG
WHERE REGEXP_INSTR(ACTION, '\d+-+\d+', 1, N_POS)<>0 OR REGEXP_INSTR(CODE, '\d+-+\d+', 1, N_POS)<>0
),
FIN AS (  --записываем то что от нас хотят
SELECT PART_ACT, PART_CODE, ACTION, CODE
FROM SEARCH_REG
ORDER BY N_LIST, N_POS
)
SELECT
NVL((CASE WHEN PART_ACT IS NULL AND PART_CODE IS NULL
THEN ACTION ELSE PART_ACT END), ' ') AS ACTION,
NVL((CASE WHEN PART_ACT IS NULL AND PART_CODE IS NULL
THEN CODE ELSE PART_CODE END), ' ') AS CODE
FROM FIN;
```

-- var 3--task2--MODEL!!!!---- решение от катюши

```
with src as (select
'952565' as numb,'2-78,2-89' tel from dual union all
select '952423' as numb,'2-78,2-83,8-34' tel from dual
),
tmp as (select * from(select rownum r,0 cnt,numb,tel from src )
model
dimension by(r,cnt,numb,tel)
measures(numb cur_numb,tel cur_tel,0 cur_row)
rules upsert all iterate(1000)
(cur_tel[any,iteration_number+1,any,any]=regexp_substr(cur_tel[cv(),0,cv(),cv()],'\d+-
+\d+',1,iteration_number+1)
,cur_numb[any,iteration_number+1,any,any]=cur_numb[cv(),0,cv(),cv()]
)order by r,cnt)
select case when cnt=1 then nvl(to_char(cur_numb),' ') else ' ' end nomer,
case when cnt!=0 then cur_tel else ' ' end telefon from tmp
where cur_tel is not null and cnt!=0 —or cur_numb is not null
;
```

3. Для каждого отдела из таблицы Departments отобразить в виде одной строки с запятой в качестве разделителя фамилии сотрудников, работающих в нем. Фамилии сотрудников должны быть отсортированы по алфавиту. Задачу решить без использования функций Listagg и wm_concat.

Решение: (ОЮ) 52в

```
WITH Helper AS (
 SELECT
  department_id,
  last_name,
  ROW_NUMBER() OVER (PARTITION BY department_id ORDER BY last_name) row_num
 FROM Employees
)
SELECT
 department_id,
 LTRIM(sys_connect_by_path(last_name, ', '), ', ') name_list
FROM Helper
WHERE CONNECT_BY_ISLEAF = 1
START WITH row_num = 1
CONNECT BY PRIOR row_num = row_num - 1 AND PRIOR department_id = department_id
ORDER BY department_id
```

Алгоритм:

Создаем вспомогательное представление Helper из таблицы Employees, добавляя к каждому сотруднику номер в соответствии с местом, которое занимает его фамилия в упорядоченном списке фамилий сотрудников данного отдела.

С помощью иерархического запроса в каждом департаменте происходит выбор сотрудников и склеивание их с помощью sys_connect_by_path() по месту, которое занимает их фамилия.

С помощью WHERE CONNECT_BY_ISLEAF = 1 удаляются промежуточные результаты и с помощью LTRIM удаляются лишние знаки.

Решение2:

```
With tab As (
Select d.department_id, e.last_name,
          ( Select Count(*) + 1 num
            From employees ee Right Join departments dd On ee.department_id = dd.department_id
            Where e.employee_id > ee.employee_id And e.department_id = ee.department_id ) rn
From employees e Right Join departments d On e.department_id = d.department_id
Order by 1, 3
)
Select  department_id "Department",
      LTrim(Sys_Connect_By_Path(last_name,', '),', ') "Employees"
From tab
Where Connect_by_IsLeaf = 1
Start With rn = 1
Connect by Prior department_id = department_id
And Prior rn + 1 = rn
Order by 1;
```

Решение3:

```
with empl as /*отделы и фамилии их сотрудников, пронумерованные внутри отделов*/
( select department_id, last_name emp,
row_number() over (partition by department_id order by last_name) r
from employees ),
empl_str as /*максимальные деревья зависимостей служащих отделов при сортировке по
фамилии для каждого отдела*/
( select department_id, trim(',' from max(sys_connect_by_path(emp, ',' ))) str
```

```
from empl
start with r = 1
connect by prior r = r-1
and prior department_id = department_id
group by department_id )
select department_id dpt, department_name depatment, str employees_list
from departments left join empl_str using(department_id)
order by department_id;
```

- с использованием функции listagg.

```
with empl_str as
( select department_id,listagg(last_name,',') within group(order by last_name) str
from employees group by department_id)
select department_id dpt, department_name depatment, str employees_list
from departments left join empl_str using(department_id)
order by department_id;
```

Решение4: (Андреева)

```
/*Определение номера отдела, фамилии и номера сотрудников, которые работают в отделе.
Для дальнейшего использование иерархического запроса необходимо найти номер служащего,
который должен следовать в списке за текущим.*/
WITH sel AS (SELECT d.department_id,
last_name,
employee_id,
LAG(employee_id) OVER (PARTITION BY d.department_id ORDER BY last_name) AS lag1
FROM employees e RIGHT JOIN departments d ON e.department_id=d.department_id)
/*Организация списка департаментов и сотрудников, работающих в них.*/
SELECT department_id, SUBSTR(SYS_CONNECT_BY_PATH(last_name, ','),2) AS name_list
FROM sel
WHERE CONNECT_BY_ISLEAF=1
START WITH lag1 IS NULL
CONNECT BY PRIOR employee_id=lag1
ORDER BY 1;
```

Решение:

```
WITH Employees_Deps AS (
  SELECT Last_Name, Department_ID, ROWNUM AS TheOrder
  FROM (
     SELECT Last_Name, Department_ID
     FROM Employees
     ORDER BY Department_ID
   )
  WHERE Department_ID IS NOT NULL
),
Min_Employee AS (
  SELECT Department_ID, MIN(TheOrder) AS MinOrder
  FROM Employees_Deps
  GROUP BY Department_ID
)
SELECT
  Department_ID,
  SUBSTR(SYS_CONNECT_BY_PATH(Last_Name, ','), 2) AS Emp_List
FROM Employees_Deps JOIN Min_Employee USING (Department_ID)
  RIGHT JOIN Departments USING (Department_ID)
WHERE CONNECT_BY_ISLEAF = 1
START WITH TheOrder = MinOrder OR TheOrder IS NULL
```

```
CONNECT BY PRIOR Department_ID = Department_ID
  AND PRIOR TheOrder + 1 = TheOrder;
```
Решение2:
```
SELECT
        department_id,
        LTRIM(SYS_CONNECT_BY_PATH(last_name, ','), ',') employees_list
FROM (
   SELECT e1.department_id, e1.employee_id, e1.last_name, COUNT(e2.employee_id) cnt
   FROM employees e1
   LEFT JOIN employees e2 ON e1.department_id = e2.department_id AND e1.last_name > e2.last_name
   GROUP BY e1.department_id, e1.employee_id, e1.last_name
)
RIGHT JOIN (
   SELECT department_id
   FROM departments
) USING (department_id)
WHERE CONNECT_BY_ISLEAF = 1
START WITH last_name IN (SELECT MIN(last_name) FROM employees GROUP BY department_id)
CONNECT BY NOCYCLE PRIOR department_id = department_id AND PRIOR cnt = cnt - 1
UNION
SELECT department_id, NULL
FROM employees RIGHT JOIN departments USING(department_id)
GROUP BY department_id
HAVING COUNT(last_name) = 0;
```
Решение: (ОЮ) 9 вариант
```
WITH
--Выводим номера отделов, фамилии сотрудников, работающих в них, в алфавитном порядке и их номера в пределах отдела
TEMP_TAB AS (SELECT DEPARTMENT_ID, LAST_NAME, ROW_NUMBER() OVER(PARTITION BY DEPARTMENT_ID ORDER BY LAST_NAME ASC) AS RN
FROM EMPLOYEES
WHERE DEPARTMENT_ID IS NOT NULL
GROUP BY DEPARTMENT_ID, LAST_NAME)

--Основной запрос: выведем номера отделов и список фамилий сотрудников, работающих в этих отделах, в строчку через запятую
SELECT tab.DEPARTMENT_ID,
--Определяем список сотрудников, работающих в заданном отделе, записав их в строчку через запятую
NVL((SELECT MAX(LTRIM(SYS_CONNECT_BY_PATH(LAST_NAME, ', '), ', '))
FROM TEMP_TAB
WHERE DEPARTMENT_ID = tab.DEPARTMENT_ID
--Начинаем цепочку фамилий с первой фамилии в заданном отделе
START WITH DEPARTMENT_ID = tab.DEPARTMENT_ID AND RN = 1
CONNECT BY NOCYCLE PRIOR RN = (RN - 1) AND DEPARTMENT_ID = tab.DEPARTMENT_ID AND LEVEL <= (SELECT COUNT(LAST_NAME)--Определяем количество сотрудников в отделе
FROM TEMP_TAB
WHERE DEPARTMENT_ID = tab.DEPARTMENT_ID)
GROUP BY DEPARTMENT_ID), ' ') AS LAST_NAMES
FROM DEPARTMENTS tab
ORDER BY tab.DEPARTMENT_ID ASC;
```
Решение: (ОЮ) 22в
```
WITH Days AS --вычисляем перерывы в работе сотрудников
(SELECT last_name, hire_date, e.department_id, e.employee_id, start_date, end_date, start_date -
```

```
LAG(end_date)
 OVER (PARTITION BY e.employee_id ORDER BY end_date asc) AS days
 FROM job_history jh
 RIGHT JOIN employees e
 ON jh.employee_id = e.employee_id),
TDays AS --если находим общую сумму перерывов, если их больше одного
(SELECT DISTINCT d1.last_name, d1.hire_date, d1.department_id, d1.employee_id,
(NVL(d1.days,0)+NVL(d2.days,0)) tdays
 FROM Days d1
 LEFT JOIN Days d2
 ON d1.employee_id = d2.employee_id
 AND d1.start_date<>d2.start_date),
Exp AS -- вычисляем стаж сотрудника (кол-во дней с даты найма по нынешний минус перерыв)
(SELECT department_id, last_name, ((TRUNC(sysdate) - hire_date) - tdays) AS exp
 FROM TDays),
Helper AS
(SELECT department_id, last_name, exp,
  ROW_NUMBER() OVER (PARTITION BY department_id ORDER BY exp) row_num
  FROM Exp)
SELECT department_id, LTRIM(SYS_CONNECT_BY_PATH(h.last_name, ', '), ', ') name_list
FROM Helper h
WHERE CONNECT_BY_ISLEAF = 1
START WITH row_num = 1
CONNECT BY PRIOR row_num = row_num - 1 AND PRIOR department_id = department_id
ORDER BY department_id;
```

4. Определить список сотрудников (таблица Employees), у которых в именах и фамилиях содержится, по крайней мере, по три совпадающие буквы.
   Результат представить в виде:

Сотрудник	Результат
Alberto Errazuriz	Совпадают три буквы (a,e,r)
Alexaner Hunold	Совпадают три буквы (d,l,n)
Elizabeth Bates	Совпадают четыре буквы (a,b,e,t)

```
WITH inf_l AS
(SELECT employee_id, LOWER(REPLACE(last_name, ' ', '')) last_name, NULL
alph
FROM employees),

recur_l(employee_id, last_name, alph) AS
(SELECT employee_id, last_name, alph
FROM inf_l
UNION ALL
SELECT employee_id, REPLACE(last_name,RPAD(last_name, 1),''),
RPAD(last_name, 1)
FROM recur_l
WHERE last_name IS NOT NULL),

las AS
(SELECT r.employee_id, e.last_name, r.alph
FROM recur_l r JOIN employees e ON(r.employee_id = e.employee_id)
WHERE alph IS NOT NULL
ORDER BY employee_id),

inf_f AS
(SELECT employee_id, LOWER(REPLACE(first_name, ' ', '')) first_name, NULL
alph
```

```
FROM employees),

recur_f(employee_id, first_name, alph) AS
(SELECT employee_id, first_name, alph
FROM inf_f
UNION ALL
SELECT employee_id, REPLACE(first_name,RPAD(first_name, 1),''),
RPAD(first_name, 1)
FROM recur_f
WHERE first_name IS NOT NULL),

fir AS
(SELECT r.employee_id, e.first_name, r.alph
FROM recur_f r JOIN employees e ON(r.employee_id = e.employee_id)
WHERE alph IS NOT NULL
ORDER BY employee_id),

res1 AS
(SELECT l.employee_id, f.first_name, f.alph , l.last_name
FROM las l JOIN fir f ON(l.employee_id = f.employee_id AND f.alph =
l.alph)),

res2 AS
(SELECT employee_id , first_name, last_name, LISTAGG(alph, ',') WITHIN
GROUP(ORDER BY alph) alph
FROM res1
GROUP BY employee_id , first_name, last_name),

res3 AS
(SELECT employee_id , first_name||' '||last_name sotr, 'Совпадают
'||DECODE(REGEXP_COUNT(alph, '[^,]+'),3,'три',4,'четыре', 5, 'пять', 6,
'шесть',  7, 'семь',  8, 'восемь', 9, 'девять',
                10, 'десять', 11, 'одиннадцать', 12, 'двенадцать', 13,
'тринадцать', 14, 'четырнадцать', 15, 'пятнадцать', 16, 'шестандцать', 17,
'семнадцать', 18, 'восемнадцать',
                19, 'девятнадцать', 20, 'двадцать', 21, 'двадцать один',
22, 'двадцать два', 23, 'двадцать три', 24, 'двадцать четыре', 25,
'двадцать пять', 26, 'двадцать шесть')||' буквы('||alph||')' alph
FROM res2
WHERE REGEXP_COUNT(alph, '[^,]+') >=3)

SELECT sotr as "Сотрудник", alph as "Результат"
FROM res3;
```

Решение из группы:

```
with match as(
select employee_id eid, last_name, first_name, substr(last_name,level,1) let
from employees
where regexp_like(first_name, substr(last_name,level,1))
connect by level<=length(last_name)),
more3 as(
select eid , count(*) over (partition by eid) cnt, first_name||' '||last_name n, let from match
)
select n "Сотрудник", 'Совпадают '||case cnt
```

```
when 3 then 'три'
when 4 then 'четыре'
when 5 then 'пять'
when 6 then 'шесть'
when 7 then 'семь'
when 8 then 'восемь'
when 9 then 'девять'
else to_char(cnt)
end||' букв'||
case
when cnt<=4 then' ('
else 'ы ('
end||
listagg(let,',') within group (order by let) over (partition by eid) ||')' "Результат" from more3
where cnt>=3;
```

5. Определить список последовательностей подчиненности от преподавателей, не имеющих начальника, до преподавателей, не имеющих подчиненных.
   Если список состоит более, чем из четырех фамилий, то выводить только две первые и две последние фамилии, а вместо остальных фамилий поставить многоточие.

Костыркин-> Викулина-> …->Соколов->Казанко (не имеет подчиненных)

……

```
with tmp as (
select
regexp_replace(sys_connect_by_path("ФАМИЛИЯ", ' -> '), '^[ ->]+', '') as by_path,
"НОМЕР_ПРЕПОДАВАТЕЛЯ" as num_prep
from "ПРЕПОДАВАТЕЛИ"
start with "ПОДЧИНЯЕТСЯ" is null
connect by prior "НОМЕР_ПРЕПОДАВАТЕЛЯ" = "ПОДЧИНЯЕТСЯ"
)
select regexp_replace(by_path, '^\s?(\w+[ ->]+\w+ ).+\w+.+( \w+[ ->]+\w+)$', '\1 -> ... -> \2') as "Подчинение"
from tmp left join "ПРЕПОДАВАТЕЛИ" podch on num_prep = "ПОДЧИНЯЕТСЯ"
where podch."ПОДЧИНЯЕТСЯ" is null;
```

Рещение2

```
with source as(
SELECT cnt, substr(teachers, 3, length(teachers)-2) || '(не имеет подчиненных)' path
FROM
(
SELECT level cnt, SYS_CONNECT_BY_PATH(фамилия,'->') teachers
FROM Преподаватель
WHERE connect_by_isleaf = 1
START WITH подчиняется IS NULL
```

```
CONNECT BY PRIOR номер_преподавателя = подчиняется
))
select case
when            cnt>4            then            substr(path,1,instr(path,'-
>',1,2)+1)||'...'||substr(path,instr(path,'->',-1,2))
else path
end "Список"
from source;
```

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ 4

1. Имеется таблица Продажи (Номер, Название товара, Дата, Скидка %). Вывести отчет по продажам, который включает столбцы Название товара, Даты продажи, Скидка %, представив информацию таким образом, что если один и тот же товар продавался с одной и той же скидкой несколько дней, то эти даты должны выводиться через запятую. При этом если две или более даты отличаются друг от друга на один день, то они должны быть представлены в виде интервала с дефисом в качестве разделителя.

   Пример представления результата:

Название товара	Даты продажи	Скидка, %
Стул	1.02.2016, 5.02.2016, 7.02.2016-12.02.2016, 15.02.2016	5
Стол	2.02.2016, 4.02.2016	10
Кровать	2.02.2016, 6.02.2016 - 7.02.2016, 12.02.2016-15.02.2016	10

Решение:

```
with Sales as (
Select 1 as "ID", 'Chair' as "Position", to_date('1.02.2016','dd.mm.yyyy') as "Date", '5%' as "Discount"
from dual UNION ALL
Select 2 as "ID", 'Chair' as "Position", to_date('5.02.2016','dd.mm.yyyy') as "Date", '5%' as "Discount"
from dual UNION ALL
Select 3 as "ID", 'Chair' as "Position", to_date('7.02.2016','dd.mm.yyyy') as "Date", '5%' as "Discount"
from dual UNION ALL
Select 4 as "ID", 'Chair' as "Position", to_date('8.02.2016','dd.mm.yyyy') as "Date", '5%' as "Discount"
from dual UNION ALL
Select 5 as "ID", 'Chair' as "Position", to_date('9.02.2016','dd.mm.yyyy') as "Date", '5%' as "Discount"
from dual UNION ALL
Select 6 as "ID", 'Chair' as "Position", to_date('10.02.2016','dd.mm.yyyy') as "Date", '5%' as "Discount"
from dual UNION ALL
Select 7 as "ID", 'Chair' as "Position", to_date('11.02.2016','dd.mm.yyyy') as "Date", '5%' as "Discount"
from dual UNION ALL
Select 8 as "ID", 'Chair' as "Position", to_date('12.02.2016','dd.mm.yyyy') as "Date", '5%' as "Discount"
from dual UNION ALL
Select 9 as "ID", 'Chair' as "Position", to_date('15.02.2016','dd.mm.yyyy') as "Date", '5%' as "Discount"
from dual UNION ALL
Select 10 as "ID", 'Table' as "Position", to_date('2.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
from dual UNION ALL
Select 11 as "ID", 'Table' as "Position", to_date('4.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
```

```
from dual UNION ALL
Select 12 as "ID", 'Bed' as "Position", to_date('2.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
from dual UNION ALL
Select 13 as "ID", 'Bed' as "Position", to_date('6.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
from dual UNION ALL
Select 14 as "ID", 'Bed' as "Position", to_date('7.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
from dual UNION ALL
Select 15 as "ID", 'Bed' as "Position", to_date('12.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
from dual UNION ALL
Select 16 as "ID", 'Bed' as "Position", to_date('13.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
from dual UNION ALL
Select 17 as "ID", 'Bed' as "Position", to_date('14.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
from dual UNION ALL
Select 18 as "ID", 'Bed' as "Position", to_date('15.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
from dual )
--выберем все данные и упорядочим их по названию позиции и размеру скидки, а внутри каждой
такой группы - по дате
,a1 as (select * from Sales order by "Position","Discount","Date")
--сопоставим каждой позиции позицию, следующую за ней
,a2 as (select "ID","Position","Date","Discount",LEAD("Position",1) over (ORDER BY
"Position","Discount","Date") as Prev_Position,LEAD("Date",1) over (ORDER BY
"Position","Discount","Date") as Prev_Date,LEAD("Discount",1) over (ORDER BY
"Position","Discount","Date") as prev_discount from a1)
--посчитаем разницу дат между соседними позициями, и если она равна единице и значения
POSITION/DISCOUNT соответственно совпадают в столбец GRUP запишем ноль(принадлежность
тому же временному отрезку), иначе поставим единицу(переход в другой временной
отрезок/другую группу)
,a3 as (select rownum rn,
"ID","Position","Date","Discount",Prev_position,prev_date,prev_discount,prev_date-"Date" as
Days,case when (prev_date-"Date")=1 and prev_discount="Discount" and prev_Position="Position" then
0 else 1 end as Grup from a2)
--найдем накапливающуюся сумму для столбца grup. Таким образом мы разобьем все записи на
группы
,a4 as (select rn, "ID","Position","Date","Discount",Grup,(sum(grup) over(order by rn)) as res from a3)
--нам необходимо сдвинуть полученные группы на 1 запись, т.к. новая группа на данный момент
начинается с последней записи, принадлежащей предыдщуему временному отрезку
,a5 as (select rn, "ID","Position","Date","Discount",nvl((lag(res,1) over(order by rn)),0) as inlinegroup
from a4)
--теперь имея группы с их номером, выберем название это группы (Position & Discount) и две даты
- максимальную и минимальную в этой группе
,a6 as (select "Position","Discount",min("Date") as miD,max("Date") as maD from a5 group by
"Position","Discount",inlinegroup )
--Теперь для каждой группы отобразим соответствующий промежуток времени: если
минимальная и максимальная даты совпадают, то просто выведем ее, иначе выпишим две
крайние даты через дефис
,a7 as (select "Position", Case when Mid=MAD then to_char(MID,'dd.mm.yyyy') else
to_char(MID,'dd.mm.yyyy')||'-'||to_char(MAD,'dd.mm.yyyy') end as Dates,mid, "Discount" from a6)
--С помощью listagg соединим группы с одинаковыми названиями товаром и скидкой в одну
строчку и выведем требуемый результат
select distinct "Position","Discount", listagg(dates,',') within group (order by mid) over (partition by
"Position","Discount") as aa  from a7 order by "Position"
```

Решение2:

```
WITH SALES(ID, NAME, DT, DISCOUNT) AS (Select 1 as "ID", 'Chair' as "Position",
to_date('1.02.2016','dd.mm.yyyy') as "Date", '5%' as "Discount" from dual UNION ALL
```

```
Select 2 as "ID", 'Chair' as "Position", to_date('5.02.2016','dd.mm.yyyy') as "Date", '5%' as "Discount"
from dual UNION ALL
Select 3 as "ID", 'Chair' as "Position", to_date('7.02.2016','dd.mm.yyyy') as "Date", '5%' as "Discount"
from dual UNION ALL
Select 4 as "ID", 'Chair' as "Position", to_date('8.02.2016','dd.mm.yyyy') as "Date", '5%' as "Discount"
from dual UNION ALL
Select 5 as "ID", 'Chair' as "Position", to_date('9.02.2016','dd.mm.yyyy') as "Date", '5%' as "Discount"
from dual UNION ALL
Select 6 as "ID", 'Chair' as "Position", to_date('10.02.2016','dd.mm.yyyy') as "Date", '5%' as "Discount"
from dual UNION ALL
Select 7 as "ID", 'Chair' as "Position", to_date('11.02.2016','dd.mm.yyyy') as "Date", '5%' as "Discount"
from dual UNION ALL
Select 8 as "ID", 'Chair' as "Position", to_date('12.02.2016','dd.mm.yyyy') as "Date", '5%' as "Discount"
from dual UNION ALL
Select 9 as "ID", 'Chair' as "Position", to_date('15.02.2016','dd.mm.yyyy') as "Date", '5%' as "Discount"
from dual UNION ALL
Select 10 as "ID", 'Table' as "Position", to_date('2.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
from dual UNION ALL
Select 11 as "ID", 'Table' as "Position", to_date('4.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
from dual UNION ALL
Select 19 as "ID", 'Table' as "Position", to_date('5.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
from dual UNION ALL
Select 12 as "ID", 'Bed' as "Position", to_date('2.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
from dual UNION ALL
Select 13 as "ID", 'Bed' as "Position", to_date('6.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
from dual UNION ALL
Select 14 as "ID", 'Bed' as "Position", to_date('7.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
from dual UNION ALL
Select 15 as "ID", 'Bed' as "Position", to_date('12.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
from dual UNION ALL
Select 16 as "ID", 'Bed' as "Position", to_date('13.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
from dual UNION ALL
Select 17 as "ID", 'Bed' as "Position", to_date('14.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
from dual UNION ALL
Select 18 as "ID", 'Bed' as "Position", to_date('15.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
from dual UNION ALL
Select 20 as "ID", 'Bed' as "Position", to_date('05.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
from dual
)
,

-- Соединяем даты, которые отличаются на 1 с равными скидками и наименованиями через тире
MT AS (SELECT NAME, LTRIM (SYS_CONNECT_BY_PATH(DT, '-'), '-') AS DATES, LEVEL LV, DISCOUNT,
CONNECT_BY_ISLEAF IL
      FROM SALES
      CONNECT BY PRIOR DT = DT-1 AND PRIOR NAME = NAME AND PRIOR DISCOUNT = DISCOUNT
      ORDER SIBLINGS BY DT) ,

-- Из всех получившися последовательностей отберем только нужные нам (то есть самые
длинные). IL и SUBSTR нужны, чтобы отделять
-- несколько промежутков через тире у одного товара с определенной скидкой
PURE_DATES AS (SELECT NAME, DATES, DISCOUNT
       FROM MT
       WHERE LV > 1 AND
```

```
        LENGTH(DATES) = (SELECT MAX(LENGTH(DATES))
                        FROM MT T
                        WHERE LV > 1 AND IL = 1
                        GROUP BY NAME, DISCOUNT, SUBSTR(DATES, -8)
                        HAVING T.NAME = MT.NAME AND T.DISCOUNT = MT.DISCOUNT AND SUBSTR(T.DATES,
-8) = SUBSTR(MT.DATES, -8)
                        )ORDER BY NAME, DATES ),

-- Выделим из промежутков только последнюю и первую даты
SUBMT AS (SELECT NAME, REGEXP_SUBSTR(DATES,'^[^-]+-') || REGEXP_SUBSTR(DATES, '[^-]+$') AS
DATES, DISCOUNT
        FROM PURE_DATES),

-- Также из промежутков выберем все даты, которые в них входят, чтобы убрать их из всех
имеющихся
STUPID_DATES AS (SELECT DISTINCT NAME, regexp_substr(DATES, '[^-]+', 1, level) DT, DISCOUNT
            FROM PURE_DATES
            CONNECT BY INSTR(DATES, '-', 1, LEVEL - 1) >0),

-- Убираем даты из промежутков из первой таблицы и объединяем с нужными промежутками
NEEDED AS ((SELECT NAME, TO_CHAR(DT, 'DD.MM.RR') AS DT, DISCOUNT
        FROM SALES
        MINUS
        SELECT NAME, DT, DISCOUNT
        FROM STUPID_DATES)
        UNION
        SELECT *
        FROM SUBMT)

-- Просто преобразуем все в строку
SELECT DISTINCT NAME, DISCOUNT, LISTAGG(DT, ', ') WITHIN GROUP(ORDER BY DT) OVER (PARTITION
BY NAME, DISCOUNT) AS DATES
FROM NEEDED
```

Решение3:

```
-- Тоже CONNECT BY
WITH SALES(ID, NAME, DT, DISCOUNT) AS (Select 1 as "ID", 'Chair' as "Position",
to_date('1.02.2016','dd.mm.yyyy') as "Date", '5%' as "Discount" from dual UNION ALL
Select 2 as "ID", 'Chair' as "Position", to_date('5.02.2016','dd.mm.yyyy') as "Date", '5%' as "Discount"
from dual UNION ALL
Select 3 as "ID", 'Chair' as "Position", to_date('7.02.2016','dd.mm.yyyy') as "Date", '5%' as "Discount"
from dual UNION ALL
Select 4 as "ID", 'Chair' as "Position", to_date('8.02.2016','dd.mm.yyyy') as "Date", '5%' as "Discount"
from dual UNION ALL
Select 5 as "ID", 'Chair' as "Position", to_date('9.02.2016','dd.mm.yyyy') as "Date", '5%' as "Discount"
from dual UNION ALL
Select 6 as "ID", 'Chair' as "Position", to_date('10.02.2016','dd.mm.yyyy') as "Date", '5%' as "Discount"
from dual UNION ALL
Select 7 as "ID", 'Chair' as "Position", to_date('11.02.2016','dd.mm.yyyy') as "Date", '5%' as "Discount"
from dual UNION ALL
Select 8 as "ID", 'Chair' as "Position", to_date('12.02.2016','dd.mm.yyyy') as "Date", '5%' as "Discount"
from dual UNION ALL
Select 9 as "ID", 'Chair' as "Position", to_date('15.02.2016','dd.mm.yyyy') as "Date", '5%' as "Discount"
from dual UNION ALL
Select 10 as "ID", 'Table' as "Position", to_date('2.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
```

```
from dual UNION ALL
Select 11 as "ID", 'Table' as "Position", to_date('4.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
from dual UNION ALL
Select 19 as "ID", 'Table' as "Position", to_date('5.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
from dual UNION ALL
Select 12 as "ID", 'Bed' as "Position", to_date('2.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
from dual UNION ALL
Select 13 as "ID", 'Bed' as "Position", to_date('6.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
from dual UNION ALL
Select 14 as "ID", 'Bed' as "Position", to_date('7.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
from dual UNION ALL
Select 15 as "ID", 'Bed' as "Position", to_date('12.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
from dual UNION ALL
Select 16 as "ID", 'Bed' as "Position", to_date('13.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
from dual UNION ALL
Select 17 as "ID", 'Bed' as "Position", to_date('14.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
from dual UNION ALL
Select 18 as "ID", 'Bed' as "Position", to_date('15.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
from dual UNION ALL
Select 20 as "ID", 'Bed' as "Position", to_date('05.02.2016','dd.mm.yyyy') as "Date", '10%' as "Discount"
from dual
), days(name, discount, dt, lvl, root) AS ( -- От каждой даты строим цепочку следующих за ней дат
SELECT name, discount, dt, LEVEL, CONNECT_BY_ROOT(dt)
FROM sales
CONNECT BY name = PRIOR name AND discount = PRIOR discount AND dt = PRIOR dt + 1
), connected(name , discount, dt, lvl, root) AS (
SELECT DISTINCT name, discount, dt,
    MAX(lvl) OVER(PARTITION BY name, discount, dt), -- Для того, чтобы удалить ветви,
начинающиеся с дат в промежутках
    MIN(root) OVER(PARTITION BY name, discount, dt) -- Начальная дата
FROM days
), nice(name, discount, days) AS ( -- Получаем даты и группы дат
SELECT c1.name, c1.discount, CASE WHEN c1.dt = c2.dt THEN TO_CHAR(c1.dt) ELSE TO_CHAR(c1.dt||'-
'||c2.dt) END
FROM connected c1 INNER JOIN connected c2 ON (c1.name = c2.name AND c2.discount = c2.discount
AND c1.dt = c2.root)
WHERE c1.lvl = 1
  AND c2.lvl = (SELECT MAX(lvl) FROM connected WHERE root = c1.dt AND name = c1.name AND
discount = c1.discount)
ORDER BY 1, 2, 3
)
-- Записываем в строчку
SELECT name, discount, LISTAGG(days, ', ') WITHIN GROUP(ORDER BY days)
FROM nice
GROUP BY name, discount;
```

Решение4 :

```
with t1 as(
select * from(
select "Название товара", substr(sys_connect_by_path("Дата",~-~),2) path, CONNECT_BY_ISLEAF leaf,
connect_by_root "Дата" root, "Скидка %"
from ПРОДАЖИ
connect by prior "Название товара" = "Название товара" and prior "Скидка %"="Скидка %" and prior
"Дата" + 1 = "Дата"
order by "Название товара", path,"Скидка %")
```

```
where leaf <> 0),

t2 as(
select * from(
select "Название товара", substr(sys_connect_by_path("Дата",~-~),2) path, CONNECT_BY_ISLEAF leaf,
connect_by_root "Дата" root , "Скидка %"
from ПРОДАЖИ
connect by prior "Название товара" = "Название товара" and prior "Скидка %"="Скидка %" and prior
"Дата" - 1 = "Дата"
order by "Название товара", path,"Скидка %")
where leaf <> 1
)
select "Название товара", listagg(str,~,~) within group(order by "Название товара","Скидка %")as
Дата,"Скидка %"
from(
select "Название товара", replace(regexp_replace(replace(path,~-~,~--~),~(-)(\d{2}.\d{2}.\d{2})+(-
)~,~~),~--~,~-~) str, rownum rn, "Скидка %"
from t1 join
(select "Название товара",root,"Скидка %" from t1
minus
select "Название товара",root,"Скидка %" from t2)USING("Название товара",root,"Скидка %") )
group by "Название товара","Скидка %";
```

Решение5:

```
WITH SALES AS--ТАБЛИЦА
(SELECT 'Стул' "Название товара", TO_DATE('1.02.2016') "Даты продажи", 5 "Скидка, %"
FROM DUAL
UNION ALL
SELECT 'Стул', TO_DATE('5.02.2016'), 5
FROM DUAL
UNION ALL
SELECT 'Стул', TO_DATE('7.02.2016'), 5
FROM DUAL
UNION ALL
SELECT 'Стул', TO_DATE('8.02.2016'), 5
FROM DUAL
UNION ALL
SELECT 'Стул', TO_DATE('9.02.2016'), 5
FROM DUAL
UNION ALL
SELECT 'Стул', TO_DATE('10.02.2016'), 5
FROM DUAL
UNION ALL
SELECT 'Стул', TO_DATE('11.02.2016'), 5
FROM DUAL
UNION ALL
SELECT 'Стул', TO_DATE('12.02.2016'), 5
FROM DUAL
UNION ALL
SELECT 'Стул', TO_DATE('15.02.2016'), 5
FROM DUAL
UNION ALL
SELECT 'Стол', TO_DATE('2.02.2016'), 10
FROM DUAL
UNION ALL
```

```
SELECT 'Стол', TO_DATE('4.02.2016'), 10
FROM DUAL
UNION ALL
SELECT 'Кровать', TO_DATE('2.02.2016'), 10
FROM DUAL
UNION ALL
SELECT 'Кровать', TO_DATE('6.02.2016'), 10
FROM DUAL
UNION ALL
SELECT 'Кровать', TO_DATE('7.02.2016'), 10
FROM DUAL
UNION ALL
SELECT 'Кровать', TO_DATE('12.02.2016'), 10
FROM DUAL
UNION ALL
SELECT 'Кровать', TO_DATE('13.02.2016'), 10
FROM DUAL
UNION ALL
SELECT 'Кровать', TO_DATE('14.02.2016'), 10
FROM DUAL
UNION ALL
SELECT 'Кровать', TO_DATE('15.02.2016'), 10
FROM DUAL),

INT AS (
  SELECT "Название товара","Даты продажи", "Скидка, %", CONNECT_BY_ROOT("Даты продажи")
ROOT,
  CONNECT_BY_ROOT("Даты продажи")||'-'||REGEXP_SUBSTR(LTRIM(SYS_CONNECT_BY_PATH("Даты
продажи",'-'),'-'),'\d{2}.\d{2}.\d{2}',1,LEVEL) path, LEVEL L
  FROM SALES
  WHERE CONNECT_BY_ISLEAF=1
  CONNECT BY PRIOR "Скидка, %"="Скидка, %"
  AND PRIOR "Даты продажи"="Даты продажи"-1
  AND PRIOR "Название товара"="Название товара"
),

INT2 AS (
  SELECT INT.*, LTRIM(REGEXP_SUBSTR(PATH,'-.+'),'-') END
  FROM INT
  WHERE L>1
),

MINR AS (
  SELECT "Название товара", MIN(ROOT) MROT
  FROM INT2 I
  GROUP BY ("Название товара",END)
),

F_INT AS (
  SELECT INT."Название товара",ROOT,PATH FROM INT JOIN MINR ON MINR."Название
товара"=INT."Название товара"
  AND MINR.MROT=INT.ROOT
),
```

```
FF_INT AS (
  SELECT "Название товара", LISTAGG (PATH,', ') WITHIN GROUP (ORDER BY "Название товара") PATH
  FROM F_INT
  GROUP BY "Название товара"
),

PEND AS (
  SELECT "Название товара",TO_CHAR("Даты продажи") "Даты продажи" ,"Скидка, %"
  FROM SALES S
  WHERE NOT EXISTS (SELECT 'X' FROM
  INT2 WHERE "Название товара"=S."Название товара" AND S."Даты продажи" BETWEEN ROOT AND
END)
  UNION ALL
  SELECT "Название товара",PATH,(SELECT "Скидка, %" FROM SALES WHERE FF."Название
товара"=SALES."Название товара" AND SALES."Даты
продажи"=REGEXP_SUBSTR(FF.PATH,'\d{2}.\d{2}.\d{2}'))"Скидка, %"
  FROM FF_INT FF
)

SELECT "Название товара",LISTAGG ( "Даты продажи",', ') WITHIN GROUP (ORDER BY "Название
товара") "Даты продажи","Скидка, %"
FROM PEND
GROUP BY "Название товара","Скидка, %";
```

Решение:

```
WITH sales(t_id, t_name, t_date, t_disc) AS (
   SELECT 1, 'Chair', to_date('1.02.2016','dd.mm.yyyy'), '5%' FROM dual UNION ALL
   SELECT 2, 'Chair', to_date('5.02.2016','dd.mm.yyyy'), '5%' FROM dual UNION ALL
   SELECT 3, 'Chair', to_date('7.02.2016','dd.mm.yyyy'), '5%' FROM dual UNION ALL
   SELECT 4, 'Chair', to_date('8.02.2016','dd.mm.yyyy'), '5%' FROM dual UNION ALL
   SELECT 5, 'Chair', to_date('9.02.2016','dd.mm.yyyy'), '5%' FROM dual UNION ALL
   SELECT 6, 'Chair', to_date('10.02.2016','dd.mm.yyyy'), '5%' FROM dual UNION ALL
   SELECT 7, 'Chair', to_date('11.02.2016','dd.mm.yyyy'), '5%' FROM dual UNION ALL
   SELECT 8, 'Chair', to_date('12.02.2016','dd.mm.yyyy'), '5%' FROM dual UNION ALL
   SELECT 9, 'Chair', to_date('15.02.2016','dd.mm.yyyy'), '5%' FROM dual UNION ALL
   SELECT 10, 'Table', to_date('2.02.2016','dd.mm.yyyy'), '10%' FROM dual UNION ALL
   SELECT 11, 'Table', to_date('4.02.2016','dd.mm.yyyy'), '10%' FROM dual UNION ALL
   SELECT 12, 'Bed', to_date('2.02.2016','dd.mm.yyyy'), '10%' FROM dual UNION ALL
   SELECT 13, 'Bed', to_date('6.02.2016','dd.mm.yyyy'), '10%' FROM dual UNION ALL
   SELECT 14, 'Bed', to_date('7.02.2016','dd.mm.yyyy'), '10%' FROM dual UNION ALL
   SELECT 15, 'Bed', to_date('12.02.2016','dd.mm.yyyy'), '10%' FROM dual UNION ALL
   SELECT 16, 'Bed', to_date('13.02.2016','dd.mm.yyyy'), '10%' FROM dual UNION ALL
   SELECT 17, 'Bed', to_date('14.02.2016','dd.mm.yyyy'), '10%' FROM dual UNION ALL
   SELECT 18, 'Bed', to_date('15.02.2016','dd.mm.yyyy'), '13%' FROM dual
),
diff AS ( --формирование разности 2х ближайших дат в группе одного товара
   SELECT t_date - LAG(t_date) OVER (PARTITION BY t_name ORDER BY t_date ASC) t_diff,
      sales.*
   FROM sales
),
gouping AS ( --группировка продаж внутри одного товара. Если 2 продажи идут датами подряд,
--то они в одной группе, иначе - в разных
   SELECT nvl(sum(t_diff-1) OVER (PARTITION BY t_name ORDER BY t_date ASC), 0) AS t_group,
         t_id, t_name, t_date, t_disc
```

```
   FROM diff
),
first_last AS ( --выделение первой и последней продажи в группе
   SELECT t_id, t_name, t_disc, t_date, t_group,
      FIRST_VALUE(t_date) OVER(PARTITION BY t_name, t_group, t_disc ORDER BY t_date ASC ROWS
BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) t_first,
      LAST_VALUE(t_date) OVER(PARTITION BY t_name, t_group, t_disc ORDER BY t_date ASC ROWS
BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) t_last
   FROM gouping
),
with_range AS ( --если в одной группе несколько продаж, то они записываются через тире
   SELECT t_id, t_name, t_disc, t_date,
      (CASE WHEN t_first=t_last
      THEN to_char(t_first, 'dd.mm.yyyy')
      ELSE to_char(t_first, 'dd.mm.yyyy')||' - '||to_char(t_last, 'dd.mm.yyyy')
      END) t_chr_date
   FROM first_last
),
last_prepare AS ( --удаление дублирующихся интервалов продаж
   SELECT t_name, t_chr_date, t_disc
   FROM with_range
   GROUP BY t_chr_date, t_name, t_disc
)
SELECT t_name, LISTAGG(t_chr_date, ', ') WITHIN GROUP(ORDER BY 1) dates, t_disc
FROM last_prepare
GROUP BY t_name, t_disc;
```

3. Используя словарь данных, получить информацию о первичных ключах и подчиненных таблицах всех таблиц в схеме HR:

Имя таблицы	Список столбцов первичного ключа	Список подчиненных таблиц

В списках имена столбцов и подчиненных таблиц вывести через запятую по алфавиту.

Задачу решить без использования функций Listagg и Wm_concat.

Решение: 33в Кож

```
WITH primaries AS (
/** В этой таблице выбираем все первичные ключи для таблиц схемы HR
*  Так как данные об ограничениях хранятся в отображении USER_CONSTRAINTS,
*  то выбираем из неё ограничения типа 'P'
*  Учтём, что нам нужны лишь таблицы схемы HR.
*/
SELECT table_name, constraint_name
FROM user_constraints
WHERE UPPER(constraint_type) = 'P' -- Тип ограничения 'P' - Primary key
-- Только таблицы из HR
  AND LOWER(table_name) IN ('employees', 'departments', 'locations',
      'regions', 'countries', 'jobs', 'job_history', 'job_grades')
)
-- Все вторичные ключи у таблиц из схемы HR
```

```
, foreigns AS (
-- R_CONSTRAINT_NAME - столбец с именем ограничения, на которое ссылается FK
SELECT table_name, constraint_name, r_constraint_name
FROM user_constraints
WHERE UPPER(constraint_type) = 'R' -- Тип ограничения 'R' - Reference
          AND LOWER(table_name) IN ('employees', 'departments', 'locations',
      'regions', 'countries', 'jobs', 'job_history', 'job_grades')
)
/** В отображении USER_CONS_COLUMNS хранится информация
*   о столбцах, на которые наложены ограничения. Найдём с его помощью
*   информацию об именах столбцов в первичных ключах каждой таблицы.
*   Информация о POSITION нужна для последующих действий (записи имён ПК
*   в строку с помощью конструкции CONNECT BY).
*/
, prime_columns AS (
SELECT primaries.table_name, column_name, position, constraint_name
FROM primaries
/** Соединяем с USER_CONS_COLUMNS по равенству имени ограничения,
*   поскольку именя ограничений уникальны в рамках схемы
*   LEFT JOIN, поскольку должны присутствовать все ограничения
*   (RIGHT и INNER тоже подойдут)
*/
          LEFT JOIN user_cons_columns USING (constraint_name)
)
-- Вторичные ключи с именами столбцов и позицией
, ref_columns AS (
SELECT foreigns.table_name, column_name, position, r_constraint_name, constraint_name
FROM foreigns
-- Соединяем с USER_CONS_COLUMNS по равенству имени ограничения
-- LEFT JOIN, поскольку все ограничения должны присутствовать (RIGHT и INNER тоже подойдут)
          LEFT JOIN user_cons_columns USING (constraint_name)
)
/** Список первичных ключей для каждой таблице.
*   Как и было сказано выше, записываем первичные ключи в строку
*   с помощью конструкции CONNECT BY и функции SYS_CONNECT_BY_PATH.
*   Условие соединения в CONNECT BY - равенство имён ограничений и
*   последовательный порядок столбца POSITION
*   (от первого к последнему с шагом 1).
*/
, prime_lists_unordered (table_name, primary_keys, constraint_name) AS(
SELECT table_name, LTRIM(SYS_CONNECT_BY_PATH(column_name, ', '), ', '),
     constraint_name
FROM prime_columns
WHERE CONNECT_BY_ISLEAF = 1 -- Выводить лишь полные списки первичных ключей
START WITH position = 1          -- Начинать строить иерархию только с первых столбцов
CONNECT BY NOCYCLE constraint_name = PRIOR constraint_name AND position = PRIOR position+1
)
-- Список вторичных ключей для каждой таблицы
, ref_lists_unordered (table_name, foreign_keys, r_constraint_name) AS (
SELECT table_name, LTRIM(SYS_CONNECT_BY_PATH(column_name, ', '), ', '),
     r_constraint_name
FROM ref_columns
WHERE CONNECT_BY_ISLEAF = 1 -- Только полные списки
START WITH position = 1          -- Начинать строить иерархию только с первых столбцов
```

```
CONNECT BY NOCYCLE constraint_name = PRIOR constraint_name AND position = PRIOR position+1
)
/** Соединяем главные и зависимые таблицы по равенству R_CONSTRAINT_NAME у
*  подчинённой и CONSTRAINT_NAME у главной. Номеруем строки внутри каждой
*  совокупности строчек для главной таблицы. Позже заменим повторяющиеся
*  пробелами.
*/
, ranked (rnk, main_table, pkeys, sub_table, fkeys) AS (
SELECT  ROW_NUMBER() OVER (PARTITION BY pu.table_name ORDER BY ru.table_name,
ru.table_name),
     pu.table_name,
     pu.primary_keys,
     ru.table_name,
     ru.foreign_keys
FROM prime_lists_unordered pu
-- Все главные должны отобразиться, поэтому LEFT JOIN
        LEFT JOIN ref_lists_unordered ru ON (pu.constraint_name = ru.r_constraint_name)
)
/** Форматируем вывод. Если номер строки внутри группы для главной таблицы
*  не равен 1, то выводим пробел. Эту операцию провести для имени главной
*  таблицы и для списка столбцов первичного ключа
*/
SELECT CASE WHEN rnk != 1 THEN ' '
        ELSE INITCAP(main_table) END          AS "Имя таблицы",
        CASE WHEN rnk != 1 THEN ' '
        ELSE INITCAP(pkeys) END               AS "Список столбцов ПК",
        -- Если подчинённых таблиц нет, вывести сообщение
        NVL(INITCAP(sub_table), 'Подчинённых таблиц нет') AS "Подчинённые таблицы",
        NVL(INITCAP(fkeys), ' ')          AS "Список столбцов втор. ключа"
FROM ranked;
```

Решение2:

```
--Рабочий вариант,единственное что он не сортирует стоблцы РК и подчиненние таблицы
WITH SOURCE AS(
SELECT ALL_T.TABLE_NAME SUPP,C2.TABLE_NAME CUST,
LAG(C2.TABLE_NAME) OVER (PARTITION BY ALL_T.TABLE_NAME ORDER BY ALL_T.TABLE_NAME)
ONE_MORE
FROM USER_CONSTRAINTS C1 INNER JOIN USER_CONSTRAINTS C2
ON C2.R_CONSTRAINT_NAME=C1.CONSTRAINT_NAME
RIGHT JOIN USER_TABLES ALL_T
ON ALL_T.TABLE_NAME=C1.TABLE_NAME
WHERE ALL_T.TABLE_NAME IN
(~COUNTRIES~,~DEPARTMENTS~,~EMPLOYEES~,~JOB_HISTORY~,~JOBS~,~JOB_GRADES~,~LOCATIONS~,
~REGIONS~)),

MYTABLE AS(
SELECT SUPP,TRIM(LEADING ~,~ FROM(SYS_CONNECT_BY_PATH(CUST,~,~))) AS LISTING,LEVEL AS l
FROM SOURCE
START WITH ONE_MORE IS NULL
CONNECT BY NOCYCLE PRIOR CUST=ONE_MORE),

FINAL_TABLE_FOR_CUST AS(
SELECT SUPP,LISTING
FROM MYTABLE M
WHERE l=(SELECT MAX(l) FROM MYTABLE WHERE SUPP=M.SUPP GROUP BY SUPP)),
```

```
AN_TABLE AS(
SELECT C3.TABLE_NAME MYTABLE,COLUMN_NAME PK_COL, LAG(COLUMN_NAME) OVER (PARTITION
BY C3.TABLE_NAME ORDER BY C3.TABLE_NAME) COL_COL
FROM USER_CONSTRAINTS C3 JOIN USER_CONS_COLUMNS C4
ON C3.CONSTRAINT_NAME=C4.CONSTRAINT_NAME
WHERE C3.CONSTRAINT_TYPE=~P~ AND (C3.TABLE_NAME
IN(~COUNTRIES~,~DEPARTMENTS~,~EMPLOYEES~,~JOB_HISTORY~,~JOBS~,~JOB_GRADES~,~LOCATIONS
~,~REGIONS~))),
LISTS_COL AS
(SELECT MYTABLE,TRIM(LEADING ~,~ FROM (SYS_CONNECT_BY_PATH(PK_COL,~,~))) LISTING_COL,
LEVEL LEV
FROM AN_TABLE
START WITH COL_COL IS NULL
CONNECT BY NOCYCLE PRIOR PK_COL=COL_COL),

FINAL_TABLE_FOR_COLUMNS AS(
SELECT MYTABLE,LISTING_COL
FROM LISTS_COL L
WHERE LEV=(SELECT MAX(LEV) FROM LISTS_COL WHERE MYTABLE=L.MYTABLE GROUP BY MYTABLE))

SELECT DISTINCT SUPP AS "Имя таблицы",
NVL(LISTING_COL,~Первичного ключа нет~)AS "Столбцы первичного ключа",
NVL(LISTING,~Подчиненных таблиц нет~)AS "Список подчиненных таблиц"
FROM FINAL_TABLE_FOR_CUST F1 LEFT JOIN FINAL_TABLE_FOR_COLUMNS F2
ON F1.SUPP=F2.MYTABLE;

Решение3:
WITH sourpk AS
 (SELECT table_name,
   column_name,
   constraint_name,
   row_number() over (partition BY constraint_name order by column_name)rn
 FROM all_cons_columns all1
 WHERE owner='HR'
 AND (SELECT constraint_type
   FROM all_constraints allc
   WHERE all1.constraint_name=allc.constraint_name
   AND allc.table_name      =all1.table_name
   AND allc.owner           =all1.owner ) IN ('P')
 ),
 /*Здесь содержится информация о всех подчинённых таблицах, хозяином которых является HR*/
 sourdeptab AS
 (SELECT dep.table_name deptbname,
   row_number() over (partition BY dep.r_constraint_name order by dep.table_name) rn,
   dom.table_name dtname
 FROM all_constraints dep
 INNER JOIN all_constraints dom
 ON dep.r_constraint_name=dom.constraint_name
 WHERE dep.owner        ='HR'
 AND dom.owner          =dep.owner
 ),
 /*Правильно сформированная (В строчку) информация о подчинённых таблицах*/
 sourdepkon AS
```

```
(SELECT table_name,
  trim(','
FROM str)str
FROM
  (SELECT sys_connect_by_path(deptbname,',') str,
    level lvl,
    dtname table_name
  FROM sourdeptab
    CONNECT BY prior rn=rn+1
  AND prior dtname     =dtname
  ) s1
WHERE lvl=
  (SELECT MAX(lvl)
  FROM
    (SELECT sys_connect_by_path(deptbname,',') str,
      level lvl,
      dtname table_name
    FROM sourdeptab
      CONNECT BY prior rn=rn+1
    AND prior dtname     =dtname
    ) s2
  WHERE s1.table_name=s2.table_name
  )
) ,
/*Правильно сформированная (В строчку) информация о пк*/
sourpkkon AS
(SELECT trim(','
FROM strpk) finstr,
  table_name,
  constraint_name
FROM
  (SELECT sys_connect_by_path(column_name,',') strpk,
    level lvl,
    table_name,
    constraint_name
  FROM sourpk s1
    CONNECT BY prior rn    =rn+1
  AND prior table_name     =table_name
  AND prior constraint_name=constraint_name
  )s3
WHERE lvl=
  (SELECT MAX(lvl)
  FROM (
    (SELECT sys_connect_by_path(column_name,',') strpk,
      level lvl,
      table_name,
      constraint_name
    FROM sourpk s1
      CONNECT BY prior rn    =rn+1
    AND prior table_name     =table_name
    AND prior constraint_name=constraint_name
    )) s2
  WHERE s2.table_name   =s3.table_name
  AND s3.constraint_name=s2.constraint_name
```

```
   )
 )
SELECT up.table_name "Имя таблицы",
  NVL(spk.finstr,'Для таблицы нет ПК')"Список столбцов ПК",
  NVL(sdt.str,'Нет подчинённых таблиц') "Список подчинённых таблиц"
FROM all_tables up
LEFT JOIN sourpkkon spk
ON up.table_name=spk.table_name
LEFT JOIN sourdepkon sdt
ON sdt.table_name=up.table_name
WHERE owner      ='HR'
```

Алгоритм3:

Запрос состоит из нескольких основных частей. Во 1х это информация о всех столбцах первичных ключей для всех constraint, которые создал 'HR'. Получаем их путём соединения all_constraints и all_cons_columns с соблюдением ограничений на тип constraint и создателя. Вторая часть – выделение списка подчинённых (по каким-то constraint) таблиц из HR. Т.к. задачу требовалось решить без listagg в каждую из 2х описанных выше таблиц пришлось добавить столбец определяющий порядок для строк(или таблиц) в пределах 1 зависимости. Получен он с помощью аналитической функции row_number(). 2 завершающие таблицы – соединяют все варианты привязанные к 1 таблице в 1 строчку и обрезают лишнюю информацию (такую как лишние запятые возникшие из-за sys_connect_by_path). Итоговый же вывод получается путём соединения 2х последних таблиц и выборки из all_tables, чтобы пройтись по каждой таблице HR

Решение4:

```
WITH
T(RES) AS (
SELECT 'COUNTRIES'
FROM DUAL
UNION
SELECT 'DEPARTMENTS'
FROM DUAL
UNION
SELECT 'EMPLOYEES'
FROM DUAL
UNION
SELECT 'JOB_GRADES'
FROM DUAL
UNION
SELECT 'JOB_HISTORY'
FROM DUAL
UNION
SELECT 'REGIONS' FROM DUAL
UNION
SELECT 'JOBS' FROM DUAL),

TT AS(
SELECT UC.TABLE_NAME, UCC.COLUMN_NAME, UCC.CONSTRAINT_NAME, UC.CONSTRAINT_TYPE
FROM USER_CONSTRAINTS UC LEFT OUTER JOIN USER_CONS_COLUMNS UCC
ON UC.CONSTRAINT_NAME = UCC.CONSTRAINT_NAME AND UC.TABLE_NAME=UCC.TABLE_NAME
WHERE UC.CONSTRAINT_TYPE = 'P'),

TBLS2 AS (
SELECT  DISTINCT(UCC2.TABLE_NAME) PK_TABLE, UC.TABLE_NAME FK_TABLE, UC.CONSTRAINT_NAME
FROM USER_CONSTRAINTS UC LEFT OUTER JOIN USER_CONS_COLUMNS UCC1 ON
```

```
UC.CONSTRAINT_NAME = UCC1.CONSTRAINT_NAME LEFT OUTER JOIN USER_CONS_COLUMNS UCC2
ON
UC.R_CONSTRAINT_NAME = UCC2.CONSTRAINT_NAME
WHERE UC.CONSTRAINT_TYPE='R'),

TTT AS (
SELECT T.res TABLEE,
LISTAGG(TT.COLUMN_NAME, ',') WITHIN GROUP (ORDER BY 2) COL_NAME
FROM T LEFT OUTER JOIN TT ON T.RES = TT.TABLE_NAME
GROUP BY T.reS)

SELECT TTT.TABLEE, NVL(TTT.COL_NAME, '  NO PK  ') PK_NAMES,
NVL(listagg(TBLS2.FK_TABLE,',') within group (order by 1), '  NO FK TABLE  ')  FK_TABLES_NAMES
FROM TTT LEFT OUTER JOIN TBLS2 ON TTT.TABLEE = TBLS2.PK_TABLE
GROUP BY TTT.TABLEE,TTT.COL_NAME;
```

Решение и описание решения без listagg

```
WITH T(RES) AS (
SELECT 'COUNTRIES'
FROM DUAL
UNION
SELECT 'DEPARTMENTS'
FROM DUAL
UNION
SELECT 'EMPLOYEES'
FROM DUAL
UNION
SELECT 'JOB_GRADES'
FROM DUAL
UNION
SELECT 'JOB_HISTORY'
FROM DUAL
UNION
SELECT 'REGIONS' FROM DUAL
UNION
SELECT 'JOBS' FROM DUAL),
```

--аналогично предыдущему описанию

```
TT AS (
SELECT UC.TABLE_NAME, UCC.COLUMN_NAME, UCC.CONSTRAINT_NAME, UC.CONSTRAINT_TYPE
FROM USER_CONSTRAINTS UC LEFT OUTER JOIN USER_CONS_COLUMNS UCC
ON UC.CONSTRAINT_NAME = UCC.CONSTRAINT_NAME AND UC.TABLE_NAME=UCC.TABLE_NAME
WHERE UC.CONSTRAINT_TYPE = 'P'),
```

--аналогично предыдущему описанию

```
TBLS2 AS (
SELECT  DISTINCT(UCC2.TABLE_NAME) PK_TABLE, UC.TABLE_NAME FK_TABLE, UC.CONSTRAINT_NAME
FROM USER_CONSTRAINTS UC LEFT OUTER JOIN USER_CONS_COLUMNS UCC1 ON
UC.CONSTRAINT_NAME = UCC1.CONSTRAINT_NAME LEFT OUTER JOIN USER_CONS_COLUMNS UCC2
ON
UC.R_CONSTRAINT_NAME = UCC2.CONSTRAINT_NAME
WHERE UC.CONSTRAINT_TYPE='R'),
```

--подчиненные таблицы: rn1 нумерует подчиненные таблицы в пределах одной таблицы
--cnt1 считает сколько подчиненных таблиц у главной таблицы

```
TBL6 AS(
SELECT TBLS2.PK_TABLE TAB1,
TBLS2.FK_TABLE FK, row_number() OVER (PARTITION BY TBLS2.PK_TABLE ORDER BY TBLS2.FK_TABLE )
rn1,
count(TBLS2.FK_TABLE) OVER (PARTITION BY TBLS2.PK_TABLE) cnt1
FROM T LEFT OUTER JOIN TBLS2 ON T.RES = TBLS2.PK_TABLE),
```

--получает дерево подчиненных таблиц, к примеру:

```
TBL7 AS (SELECT TBL6.TAB1, sys_connect_by_path(TBL6.FK, ',') scbp1 FROM TBL6
START WITH rn1=1 CONNECT BY (PRIOR rn1) = rn1-1 AND (prior TAB1)=TAB1 AND LEVEL<=cnt1),
```

--аналогично tbl6 проделывает то же самое только для столбцов первичного ключа

```
T4 AS(
SELECT T.res TABLEE,
TT.COLUMN_NAME COL_NAME,row_number() OVER (PARTITION BY T.RES ORDER BY
TT.COLUMN_NAME) rn,
count(TT.COLUMN_NAME) OVER (PARTITION BY T.RES) cnt
FROM T LEFT OUTER JOIN TT ON T.RES = TT.TABLE_NAME),
```

--аналогично tbl7 получает дерево первичных ключей

```
T5 AS (
SELECT T4.TABLEE, sys_connect_by_path(T4.COL_NAME, ',') scbp FROM t4 START WITH
rn=1 CONNECT BY (PRIOR rn) = rn-1 AND (prior TABLEE)=TABLEE AND LEVEL<=cnt)
```

--со второй позиции, потому что первый символ это запятая

```
SELECT distinct (TAblee) TABLE_NAME, NVL(substr((max(scbp) OVER (PARTITION BY tablee)) ,2), '  NO PK
')PK_COL_NAMES,
NVL(substr((max(scbp1) OVER (PARTITION BY TAB1)) ,2),'  NO FK TABLE  ') FK_TABLE_NAMES
FROM t5 LEFT OUTER JOIN TBL7 ON T5.TABLEE = TBL7.TAB1;
```

Решение5:

```
with t1 as (
        select  s.table_name parent_table, --выводим название родительской таблицы
              s.constraint_name, --название ограничения
              ltrim(sys_connect_by_path(column_name,','),',') parent_table_pk_column_list --и лист
столбцов первичного ключа через ,
         from     user_constraints s
              join
                user_cons_columns c
               on (
                    s.constraint_type = 'P'
                 and
                    c.constraint_name = s.constraint_name
                )
         where connect_by_isleaf = 1
         start with position = 1
         connect by s.table_name = prior s.table_name
              and position = prior position + 1
       ),
   t2 as (
        select  parent_table, --берем имя родительской таблицы
```

```
            parent_table_pk_column_list, --лист столбцов первичного ключа
            f.table_name child_table, --зависимую таблицу
            row_number() over(partition by parent_table order by f.table_name) rn --присваиваем
rownum внутри parent_table
        from     t1
            left join
              user_constraints f
             on (
                f.r_constraint_name = t1.constraint_name
                )
       )
select  parent_table, --выбираем из подзапросов имя родительской таблицы
      parent_table_pk_column_list, --список столбцов PK
      ltrim(sys_connect_by_path(child_table,','),',') child_table_list --список подчиненных таблиц
 from  t2
 where connect_by_isleaf = 1 --выводим только листы дерева чтобы убрать наращивание в поле
child_table_list
 start with rn = 1
 connect by parent_table = prior parent_table
      and rn = prior rn + 1
 order by parent_table;
```

Решение6:

Без использования listagg

```
with q as
--в этой таблице выбираем таблицы из HR и список их первичных кючей
--в этом варианте используем иерархический запрос и создаем список с
--помощью sys_connect_by_path создаем список
(select table_name,ltrim(sys_connect_by_path(column_name,', '),', ') pk, level lvl
from (select table_name, column_name, rownum r
from (select distinct c1.table_name, c1.column_name
from all_cons_columns c1 join all_constraints c2
on c1.constraint_name=c2.constraint_name
--выбираем из схемы HR и типом ключа P
where c1.owner='HR' and c2.constraint_type='P'))
connect by prior table_name=table_name AND prior r<r),
t AS
--в этой таблице выбираем таблицы из схемы HR и списки их подчиненных таблиц
--также с помощью sysconnectbypath создаем список
(select tab, ltrim(sys_connect_by_path(owner||'.'||lower(table_name),', '),', ') FK, level lvl
from (select c1.table_name tab, c2.table_name, c2.owner,rownum r from all_constraints c1 join
all_constraints c2
on c1.constraint_name=c2.r_constraint_name
where c1.owner='HR')
connect by prior tab=tab and prior r<r)
--соединяе первые две таблицы и выводим имена таблиц и списки их первичных ключей и
--списки подчиненных таблиц
select table_name, pk, fk
from (select table_name, pk from q where lvl=(select max(lvl) from q q1 where
q1.table_name=q.table_name group by table_name)) q1
left join (select tab, fk from t where lvl=(select max(lvl) from t t1 where t1.tab=t.tab group by tab)) t1
on q1.table_name=t1.tab;
```

С использованием listagg

```
with q as
--аналогично как в первом варианте только с помощью listagg создание списка
--и без использование иерархии
--содержит имена таблиц и список первичных ключей
(select distinct table_name,
listagg(column_name,', ') within group(order by column_name) over(partition by table_name) PK
from (select distinct c1.table_name, c1.column_name from all_cons_columns c1 join all_constraints c2
on c1.constraint_name=c2.constraint_name
--выбираються из схемы HR и с типом ключа P
where c1.owner='HR' and c2.constraint_type='P')),
q1 AS
--содержит имена таблиц и список подчиненных таблиц
(select distinct c1.table_name,
listagg(c2.owner||'.'||lower(c2.table_name), ', ') WITHIN GROUP(order by c2.table_name)
over(partition by c1.table_name) FK
from all_constraints c1 join all_constraints c2
on c1.constraint_name=c2.r_constraint_name
--выбираются из схемы HR
where c1.owner='HR')
--итоговый запрос в котором соединяются первые две таблицы
--выводиться имя таблицы и списки перв ключей и подчиненных таблиц
select q.table_name, pk, fk
from q left join q1
on q.table_name=q1.table_name;
```

Решение (кож) 51в

```
WITH
TEMP_1 AS (SELECT S.TABLE_NAME PARENT_TABLE, --РОДИТЕЛЬСКАЯ
ТАБЛИЦА
S.CONSTRAINT_NAME, --ОГРАНИЧЕНИЕ
LTRIM(SYS_CONNECT_BY_PATH(COLUMN_NAME,','),',') PK_PTABLE_LIST
--СТОЛБЦЫ ПЕРВИЧНОГО КЛЮЧА, ЧЕРЕЗ ЗАПЯТУЮ
FROM ALL_CONSTRAINTS S JOIN ALL_CONS_COLUMNS C ON
(S.CONSTRAINT_TYPE = 'P' AND C.CONSTRAINT_NAME =
S.CONSTRAINT_NAME)
WHERE CONNECT_BY_ISLEAF = 1
AND S.OWNER = 'HR' AND C.OWNER = 'HR'
START WITH POSITION = 1
CONNECT BY S.TABLE_NAME = PRIOR S.TABLE_NAME AND POSITION = PRIOR
POSITION + 1),
TEMP_2 AS (
SELECT PARENT_TABLE, --ИМЯ РОДИТЕЛЯ
PK_PTABLE_LIST,
(SELECT DISTINCT TABLE_NAME FROM ALL_TABLES WHERE TABLE_NAME =
F.TABLE_NAME) CTABLE, --ИМЯ ЗАВИСИМОЙ ТАБЛИЦЫ
ROW_NUMBER() OVER(PARTITION BY PARENT_TABLE ORDER BY
F.TABLE_NAME) RN --ПРИСВАИВАЕМ ROWNUM
FROM TEMP_1 LEFT JOIN ALL_CONSTRAINTS F ON (F.R_CONSTRAINT_NAME =
TEMP_1.CONSTRAINT_NAME))
SELECT PARENT_TABLE "ИМЯ ТАБЛИЦЫ",
PK_PTABLE_LIST "СПИСОК СТОЛБЦОВ PK",
LTRIM(SYS_CONNECT_BY_PATH(CTABLE,','),',') "СПИСОК ПОДЧИНЕННЫХ
ТАБЛИЦ"
FROM TEMP_2
```

```
WHERE CONNECT_BY_ISLEAF = 1 --ВЫВОДИМ ТОЛЬКО ЛИСТЬЯ ДЕРЕВА
START WITH RN = 1
CONNECT BY PARENT_TABLE = PRIOR PARENT_TABLE AND RN = PRIOR RN + 1
        ORDER BY PARENT_TABLE;
```

3. Определить, сколько раз каждая из цифр от 0 до 9 встречается в столбце Phone_number таблицы Employees.

Пример результата:

Цифра	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Количество	15	12	23	45	24	33	45	12	30	15

```
with
Str as
(select listagg(phone_number, '') WITHIN GROUP (order by phone_number) as mainStr
from employees)

select  'Количество' as "Цифра",
      REGEXP_COUNT(mainStr, 0) as "0",
      REGEXP_COUNT(mainStr, 1) as "1",
      REGEXP_COUNT(mainStr, 2) as "2",
      REGEXP_COUNT(mainStr, 3) as "3",
      REGEXP_COUNT(mainStr, 4) as "4",
      REGEXP_COUNT(mainStr, 5) as "5",
      REGEXP_COUNT(mainStr, 6) as "6",
      REGEXP_COUNT(mainStr, 7) as "7",
      REGEXP_COUNT(mainStr, 8) as "8",
      REGEXP_COUNT(mainStr, 9) as "9"
from Str;
```

4. Создать запрос для построения отчета по количеству студентов, сдававших экзамены в определенные дни в произвольно заданном интервале дат по различным дисциплинам. Результаты должны быть отсортированы по датам, количеству студентов и названию дисциплин. Одна и та же дата должна встречаться в отчете не более двух раз: на первой строке данной даты и на отчетной строке даты. Отчет должен иметь следующий вид:

Дата	**День недели**	**Дисциплина**	**Количество студентов**
13.01.15	Вторник	Экономика	1
		Химия	2
13.01.15: Итого	Вторник		3
14.01.15: Итого	Среда		0
15.01.15	Четверг	Математика	1
		Физика	1
		Экономика	3
15.01.15: Итого	Четверг		5
16.01.15	Пятница	Менеджмент	1
16.01.15: Итого	Пятница		1
17.01.15	Суббота	Математика	4
17.01.15: Итого	Суббота		4
18.01.15:Итого	Воскресенье		0
19.01.15	Понедельник	Экономика	1
		Физика	2
19.01.15: Итого	Понедельник		3

20.01.15	Вторник	Химия	2
20.01.15: Итого	Вторник		2
		ОБЩИЙ ИТОГ	18

Задачу решить с использованием раздела Model.

```
ALTER SESSION SET NLS_DATE_FORMAT = 'DD.MM.YYYY';
ALTER SESSION SET NLS_DATE_LANGUAGE = 'Russian';

DEFINE start_date = TO_DATE('10.06.1999')
DEFINE end_date = TO_DATE('20.06.1999')

WITH Days AS (
  SELECT Day
  FROM (
    SELECT &end_date - &start_date AS Days_Cnt
    FROM dual
  )
  MODEL
    DIMENSION BY (1 AS ID)
    MEASURES (&start_date AS Day, Days_Cnt)
    RULES UPSERT ALL ITERATE(1048576) UNTIL (ITERATION_NUMBER + 1 =
Days_Cnt[1]) (
      Day[ITERATION_NUMBER + 2] = Day[ITERATION_NUMBER + 1] + 1
    )
),
The_Model AS (
  SELECT *
  FROM Days
    LEFT JOIN Успеваемость Grades
      ON Days.Day = Grades.Дата
    LEFT JOIN Дисциплины Sub
      ON Grades.Номер_Дисциплины = Sub.Номер_Дисциплины
  MODEL
    DIMENSION BY (Days.Day AS Day, Sub.Название AS Subj,
Grades.Номер_Студента AS Student)
    MEASURES (0 AS Students)
    RULES UPSERT ALL (
      Students[ANY, ANY, ANY] = COUNT(Students)[CV(), CV(), ANY],
      Students[ANY, NULL, ANY] = COUNT(Students)[CV(), Subj IS NOT NULL,
ANY],
      Students[NULL, 'ОБЩИЙ ИТОГ', NULL] = COUNT(Students)[ANY, Subj IS
NOT NULL, ANY]
    )
)
SELECT
  "Дата",
  "День недели",
  "Дисциплина",
  "Количество студентов"
FROM (
  SELECT DISTINCT
    CASE
      WHEN DRank = 1 OR Subj IS NULL
      THEN "Дата"
      ELSE ' '
    END AS "Дата",
    CASE
      WHEN DRank = 1 OR Subj IS NULL
      THEN "День недели"
      ELSE ' '
```

```
    END AS "День недели",
    "Дисциплина",
    "Количество студентов",
    Day, Students, Subj
  FROM (
    SELECT
      CASE
        WHEN Subj IS NULL
        THEN TO_CHAR(Day, 'DD.MM.YY') || ': Итого'
        ELSE NVL(TO_CHAR(Day, 'DD.MM.YY'), ' ')
      END AS "Дата",
      NVL(TO_CHAR(Day, 'Day'), ' ') AS "День недели",
      NVL(Subj, ' ') AS "Дисциплина",
      Students AS "Количество студентов",
      Day, Students, Subj,
      DENSE_RANK() OVER (PARTITION BY Day ORDER BY Students, Subj) AS
DRank
    FROM The_Model
  )
)
ORDER BY Day, Students, Subj;
```

Без model

```
undefine date1
undefine date2
WITH dates AS (
SELECT TO_DATE('&&date1', 'dd.MM.yyyy')+level-1 дата
FROM dual
CONNECT BY TO_DATE('&&date1', 'dd.MM.yyyy')+level-1 <=
TO_DATE('&&date2', 'dd.MM.yyyy')
),
tmp AS (
SELECT дата,
    TO_CHAR(дата, 'Day') день_недели,
    название,
    COUNT(номер_студента) количество,
    GROUPING(дата) gr1,
    GROUPING(название) gr2,
        ROW_NUMBER() OVER (PARTITION BY дата ORDER BY название)
RNUM
FROM dates LEFT JOIN успеваемость USING(дата)
        LEFT JOIN дисциплины USING(номер_дисциплины)
GROUP BY ROLLUP(дата, название))

SELECT CASE WHEN gr1=0 AND gr2=0 AND RNUM=1 THEN TO_CHAR(дата,
'dd.MM.yy')
```

```
                WHEN gr2=1 AND gr1=0 THEN TO_CHAR(дата, 'dd.MM.yy') || ': Итого'
                ELSE ' '
            END "Дата",
            CASE WHEN gr1=0 AND gr2=0 AND RNUM=1 OR gr2=1 AND gr1=0 THEN день_недели ELSE ' ' END "День недели",
            CASE WHEN gr1=1 AND gr2=1
                THEN 'ОБЩИЙ ИТОГ'
                ELSE NVL(название, ' ') END "Дисциплина",
            количество "Количество студентов"
        FROM tmp WHERE NOT (название IS NULL AND GR2 = 0);
```

5 Для произвольной строки, состоящей из цифр, определить все возможные наборы слов, получаемые при замене чисел на номер буквы в русском алфавите. Например, для строки 211221 результатом должно быть: баабба, бйбба,баафа,бакба,уку,……

Решение

```
DEFINE STR = '211221';

DEFINE CHR = 'абвгдеёжзийклмнопрстуфхцчшщъыьэюя';
WITH REC(I,J,L,NUM) AS
 (SELECT 1 I, 2 J, 1 L, SUBSTR('STR',1,1) NUM
 FROM DUAL
 UNION ALL
 SELECT CASE
       WHEN L+1 > 2
       THEN I+1
       ELSE I
      END,
      I+2,
      CASE
       WHEN L+1 > 2
       THEN 1
       ELSE 2
      END,
      CASE
       WHEN L+1 > 2
       THEN SUBSTR('STR',I+1,1)
       ELSE SUBSTR('STR',I,2)
      END
 FROM REC
 WHERE I < LENGTH('STR') OR J < LENGTH('STR')),
CHARS AS
 (SELECT I,J,
         CASE
          WHEN NUM>33
          THEN NULL
          ELSE SUBSTR('CHR',TO_NUMBER(NUM),1)
         END CHR
```

```
   FROM REC
   WHERE NUM<=33),
TREE AS
 (SELECT REPLACE(SYS_CONNECT_BY_PATH(CHR,' '),' ') WORDS
  FROM CHARS
  WHERE CONNECT_BY_ISLEAF = 1
  START WITH I = 1
  CONNECT BY PRIOR J = I)
SELECT LISTAGG(WORDS,', ') WITHIN GROUP(ORDER BY WORDS) WORDS
FROM TREE;
```

Решение2:

```
WITH st(s) as(
          SELECT '2117221' FROM dual
), all_types(a, b, lvl) AS (
SELECT SUBSTR(s, LEVEL, 1), SUBSTR(s, LEVEL, 2), LEVEL
FROM str
CONNECT BY SUBSTR(s, LEVEL) IS NOT NULL
), one_column AS (
SELECT a, lvl
FROM all_types
UNION ALL
SELECT b, lvl
FROM all_types
WHERE b <= ASCII('я') - ASCII('a')
), possible_strings(s) AS (
SELECT DISTINCT SYS_CONNECT_BY_PATH(a, '.')
FROM one_column
WHERE CONNECT_BY_ISLEAF = 1
START WITH lvl = 1
CONNECT BY lvl = PRIOR lvl + LENGTH(PRIOR a)
), R(s, letter, pos) AS (
SELECT s, CHR(ASCII('a')+REGEXP_SUBSTR(s, '\d+', 1, 1)-1), 1
FROM possible_strings
UNION ALL
SELECT s, CHR(ASCII('a')+REGEXP_SUBSTR(s, '\d+', 1, pos+1)-1), pos+1
FROM R
WHERE REGEXP_SUBSTR(s, '\d+', 1, pos+1) IS NOT NULL
)
SELECT REPLACE(LISTAGG(letter, ',') WITHIN GROUP (ORDER BY pos), ',', '')
FROM R
GROUP BY s;
```

Решение3:

```
WITH scr as(
   SELECT '211221' str
   FROM dual
),
--формируем все числа, которые встречаются в строке
--а так как нам нужен номер буквы алфавита, то одно- либо дву-значное
all_nums AS (
   SELECT to_number(substr(str, level, 1)) num,
      level AS pos,
      length(str) AS len
   FROM src
   CONNECT BY level<=length(str)
```

```
   UNION
   SELECT to_number(substr(str, level, 2)),
      level,
      length(str) AS len
   FROM src
   CONNECT BY level<=length(str)
),
--фильтруем номера, которые не соответствуют позиции буквы в алфавите
--и заодно добавляем позицию начала и конца номера в первоначальной строке
all_nums_filter AS (
  SELECT num, pos,
    (CASE WHEN length(to_char(num))=1
    THEN pos ELSE pos+1 END) pos_end,
    len
  FROM all_nums
  WHERE num BETWEEN 1 AND 33
),
--собираем полученные числа в последовательность букв
--по правилу: конец предыдущего числа равен началу следующего
seq AS (
   SELECT Sys_Connect_By_Path(
      substr('абвгдеёжзийклмнопрстуфхцчшщъыьэюя', num, 1),
      ' ') path, pos_end
   FROM all_nums_filter
   START WITH pos=1
   CONNECT BY prior pos_end=pos-1
)
--выводим ответ
SELECT replace(path, ' ', '') res
FROM seq
WHERE pos_end=6;
```

Решение4: (Кож) 18в

```
define str = '211221';

with
  str_repr(pos, curr_len, tmp, str, len) as(
     select 1 as pos, 1 as curr_len, substr('&&str',1,1) as tmp, '&&str' as str, length('&&str') as len
     from dual
     union all
     select case when ((to_number(substr(str,pos,curr_len+1)) <= 33)  and (pos + curr_len  <= len)) then
pos else pos+1 end as pos
          ,case when ((to_number(substr(str,pos,curr_len+1)) <= 33)  and (pos + curr_len  <= len)) then
curr_len + 1 else 1 end as curr_len
          ,case when (to_number(substr(str,pos,curr_len+1)) <= 33) and (pos + curr_len  <= len) then
substr(str,pos,curr_len+1) else substr(str,pos+1,1) end as tmp
          ,str
          ,len
     from str_repr
     where pos < len)
  CYCLE pos, curr_len SET cyclemark TO 'X' DEFAULT '-'
  ,alph1(n,letter) as(
     select 1 as n,chr(224) as letter
     from dual
     union all
```

```
    select n+1 as n, chr(n+224)
    from alph1
    where n < 6)
 ,alph2(n,letter) as(
    select 7 as n,'ё' as letter
    from dual
    union all
    select n+1 as n, chr(n+223)
    from alph2
    where n < 33)
 ,alph as(
    select *
    from alph1
    union all
    select *
    from alph2)
 ,tree as(
   select lpad(tmp, length(tmp) + 2*(level - 1), '_'), lpad((select letter from alph where n =
to_number(tmp)), 1 + 2*(level - 1), '_')
        , case  when connect_by_isleaf = 1 then replace(sys_connect_by_path((select letter from alph
where n = to_number(tmp)),' '),' ') end as res
   from str_repr
   start with pos = 1
   connect by (pos = prior pos+1 and prior curr_len = 1)
           or (pos = prior pos + 2 and prior curr_len = 2))
select listagg(res, ',') within group(order by res) as "Всевозможные комбинации"
from tree;
undefine str;
```

Алгоритм4:

С помощью подзапросов alph1 и alph2, которые присваивают порядковые номера буквам русского алфавита до буквы “ё” и после соответственно, в подзапросе alph получаем все буквы русского алфавита в нижнем регистре и их порядковые номера. Рекурсивный подзапрос str_repr служит для того, чтобы разобрать строку на всевозможные последовательности чисел, которые могут представлять буквы русского алфавита. Так как одна цифра точно является номером некоторой буквы, то она извлекается из строки и сохраняется в столбце tmp. Солбцы pos и curr_len служат для сохранения позиции первой цифры строки, с которой ищется подпоследовательность, которая так же может быть порядковым номером некоторой буквы. На каждом шаге рекурсии проверяется, является ли подстрока, которая является числом, проверенном на прошлом шаге, с добавленной в конец следующей цифрой подстроки, и можно ли вообще выделить еще одну цифру (не выходит ли подпоследовательность за пределы строки), порядковым номером некоторой буквы алфавита, и если оба условия выполняются, то позиция, с которой начинается поиск, не меняется, для проверки подстроки длинной на 1 больше, длина подстроки увеличивается на 1, подстрока сохраняется в tmp. Если же хотя бы одно из условий не выполняется, то pos необходимо увеличить на 1, длинну подстроки сбросить в 1 и выделить цифру на следующей позиции, так как одна цифра гарантированно является некоторой буквой русского алфавита. Алгоритм продолжается, пока pos не дойдет до длинны строки. Так как возможен цикл, используется параметр cycle, а место, где начинается цикл, помечается “X” для поиска места и наглядности. Далее, на основе результата str_repr в подзапросе tree строится иерархия, которая разбирает строку с учетом того, понимается ли одно число или два под следующей буквой. Первый столбец выборки используется для наглядного представления дерева, второй заменят числа соответствующими буквами и сохраняет путь от листа до корня. В основном запросе из подзапроса tree выбирается второй столбец, и все варианты собираются в одну строку с помощью агрегатной функции listagg.

Решение5:

```
WITH dat(s) AS (
--SELECT '121212' FROM dual
--SELECT '9999' FROM dual
--SELECT '191' FROM dual
--SELECT '242424' FROM dual
SELECT '211221' FROM dual
), divided(n, rn) AS (
SELECT SUBSTR(s, LEVEL, 1),  LEVEL
FROM dat
CONNECT BY SUBSTR(s, LEVEL, 1) IS NOT NULL
UNION ALL
SELECT SUBSTR(s, LEVEL, 2),  LEVEL
FROM dat
WHERE SUBSTR(s, LEVEL, 2) <=  ASCII('я') - ASCII('a') + 1
CONNECT BY SUBSTR(s, LEVEL, 2) IS NOT NULL
), all_combs(comb) AS (
SELECT DISTINCT SYS_CONNECT_BY_PATH(n, '.')
FROM divided
WHERE CONNECT_BY_ISLEAF = 1
START WITH rn = 1
CONNECT BY rn = CASE WHEN LENGTH(PRIOR n) = 2 THEN  PRIOR rn + 2
ELSE PRIOR rn + 1 END
), R(comb, pos, n) AS (
SELECT comb, 1, REGEXP_SUBSTR(comb, '\d+', 1, 1)
FROM all_combs
UNION ALL
SELECT comb, pos+1, REGEXP_SUBSTR(comb, '\d+', 1, pos+1)
FROM R
WHERE REGEXP_SUBSTR(comb, '\d+', 1, pos+1) IS NOT NULL
)


SELECT REPLACE(SYS_CONNECT_BY_PATH(CHR(ASCII('a')+n-1), '.'), '.', '')
FROM R
WHERE CONNECT_BY_ISLEAF = 1
START WITH pos = 1
CONNECT BY comb = PRIOR comb AND pos = PRIOR pos+1;
```

БИЛЕТ 5

1. Напишите запрос к таблице Трафик (Дата и время начала транзакции, Дата время окончания транзакции, Объем в байтах), который на каждый день заданного месяца и года сосчитает количество переданного трафика. Если сессия началась и закончилась в разные дни, то трафик следут разделить пропорционально длительности в каждом дне. Если результат пропорционального деления дробный, то отбросить дробную часть для начального дня, а остаток относить к окончанию сессии. Продолжительность сессии не ограничена во времени.

Решение:

```
CREATE TABLE траффик(
начало DATE,
конец DATE,
байт NUMBER
);
WITH sta AS (
SELECT начало, конец, байт, байт / (конец - начало) band, rownum rwn
```

```
FROM траффик
),
rec(st, en, enf, tr, b, rn) AS (
SELECT начало, CASE WHEN конец - начало > trunc(начало + 1) - начало THEN trunc(начало + 1) - 1 / 86400 ELSE конец END,конец,
CASE WHEN конец - начало > trunc(начало + 1) - начало THEN trunc(band * (trunc(начало + 1) - начало)) ELSE байт END, band, rwn
FROM sta
UNION ALL
SELECT trunc(st + 1), CASE WHEN enf - trunc(st + 1) > trunc(st + 2) - st THEN trunc(st + 2) - 1 / 86400 ELSE enf END, enf,
CASE WHEN enf - trunc(st + 1) > trunc(st + 2) - st THEN b ELSE b * (enf - trunc(st + 1)) END, b, rn
FROM rec
WHERE trunc(st + 1) < enf
),
dop AS (
SELECT rn, sum(tr) tr, байт
FROM rec rc
INNER JOIN sta st ON st.rwn = rc.rn
GROUP BY rn, байт
),
dif AS (
SELECT rn, байт - tr d
FROM dop
),
res AS (
SELECT trunc(st) st, rc.tr + nvl(d, 0) tr
FROM rec rc
LEFT JOIN dif df ON rc.rn = df.rn AND trunc(st) = trunc(enf)
),
dat AS (
SELECT to_date('01.&&in', 'DD.MM.SYYYY') + level - 1 st
FROM dual
CONNECT BY to_date('01.&in', 'DD.MM.SYYYY') + level - 1 < add_months(to_date('01.&in', 'DD.MM.SYYYY'), 1)
)
SELECT to_char(dt.st,'DD.MM.SYYYY') Дата, nvl(sum(tr), 0) Передано
FROM dat dt
LEFT JOIN res rs ON dt.st = rs.st
GROUP BY dt.st
ORDER BY dt.st;
undefine in;
```

Алгоритм:

1. Выведем для каждой сессии дату ее начала, дату конца, сколько байт передано, сколько байт передается за день и порядковый номер записи (будет использоваться позднее для правильного распределения трафика).

```
WITH sta AS (
SELECT начало, конец, байт, байт / (конец - начало) band, rownum rwn
FROM траффик
)
```

2. С помощью рекурсии разобьем каждую сессию на дни. Также для каждого дня посчитаем, сколько байт передано в этот день. Для этой цели умножим часть дня, которая входит в сессию на количество байт, передаваемых за день. Причем, если это первый день, то по условию задачи нужно откинуть дробную часть числа. Если сессия целиком лежит в одном дне, то сразу выводим

общее число байт без умножения. Также для каждого дня сохраняем порядковый номер сессии, который был получен на предыдущем шаге.

```
rec(st, en, enf, tr, b, rn) AS (
SELECT начало, CASE WHEN конец - начало > trunc(начало + 1) - начало THEN trunc(начало + 1) - 1 / 86400 ELSE конец END,конец,
CASE WHEN конец - начало > trunc(начало + 1) - начало THEN trunc(band * (trunc(начало + 1) - начало)) ELSE байт END, band, rwn
FROM sta
UNION ALL
SELECT trunc(st + 1), CASE WHEN enf - trunc(st + 1) > trunc(st + 2) - st THEN trunc(st + 2) - 1 / 86400 ELSE enf END, enf,
CASE WHEN enf - trunc(st + 1) > trunc(st + 2) - st THEN b ELSE b * (enf - trunc(st + 1)) END, b, rn
FROM rec
WHERE trunc(st + 1) < enf
)
```

3. Посчитаем для каждой сессии, сколько байт было передано фактически, при нашем разделении. Это нужно для того, чтобы позднее добавить к последнему дню недостающие байты, которые были отобраны у 1 дня или были потеряны из-за округления периодических дробей.

```
dop AS (
SELECT rn, sum(tr) tr, байт
FROM rec rc
INNER JOIN sta st ON st.rwn = rc.rn
GROUP BY rn, байт
)
```

4. Посчитаем для каждой сессии недостающие байты.

```
dif AS (
SELECT rn, байт - tr d
FROM dop
)
```

5. Добавим к последнему дню каждой сессии недостающие байты. Последний день определяется так, что дата начала текущего дня должна быть равна дате начала дня окончания сессии.

```
res AS (
SELECT trunc(st) st, rc.tr + nvl(d, 0) tr
FROM rec rc
LEFT JOIN dif df ON rc.rn = df.rn AND trunc(st) = trunc(enf)
)
```

6. Выведем все дни для заданного месяца и года (в условии сказано, что требуется вывести трафик для всех дней заданного месяца).

```
dat AS (
SELECT to_date('01.&&in', 'DD.MM.SYYYY') + level - 1 st
FROM dual
CONNECT BY to_date('01.&in', 'DD.MM.SYYYY') + level - 1 < add_months(to_date('01.&in', 'DD.MM.SYYYY'), 1)
)
```

7. Сформируем окончательный результат. Выведем для дней, в которые ничего не было передано нули. Отсортируем результат по дню месяца для удобства.

```
SELECT to_char(dt.st,'DD.MM.SYYYY') Дата, nvl(sum(tr), 0) Передано
FROM dat dt
LEFT JOIN res rs ON dt.st = rs.st
GROUP BY dt.st
ORDER BY dt.st;
```

Решение:

```
WITH TRAFFIC AS (
  SELECT TO_DATE('12.06.2016 20:14', 'DD.MM.YYYY HH24:MI') START_DATE,
```

```
  TO_DATE('12.06.2016 23:44', 'DD.MM.YYYY HH24:MI') END_DATE, 55 BYTES
  FROM DUAL
  UNION ALL
  SELECT TO_DATE('15.06.2016 12:23', 'DD.MM.YYYY HH24:MI') START_DATE,
  TO_DATE('15.06.2016 23:44', 'DD.MM.YYYY HH24:MI') END_DATE, 1000 BYTES
  FROM DUAL
  UNION ALL
  SELECT TO_DATE('18.06.2016 22:23', 'DD.MM.YYYY HH24:MI') START_DATE,
  TO_DATE('24.06.2016 23:44', 'DD.MM.YYYY HH24:MI') END_DATE, 2000 BYTES
  FROM DUAL
  UNION ALL
  SELECT TO_DATE('13.04.2016 18:00', 'DD.MM.YYYY HH24:MI') START_DATE,
  TO_DATE('15.04.2016 18:00', 'DD.MM.YYYY HH24:MI') END_DATE, 1000 BYTES
  FROM DUAL
),

TAB3 AS (
  SELECT *
  FROM TRAFFIC
  WHERE (END_DATE - START_DATE > 1)),

DAYS AS (
  SELECT DISTINCT START_DATE + LEVEL - 1 DAY
  FROM TAB3
  CONNECT BY LEVEL <= END_DATE - START_DATE + 1
  ORDER BY DAY)

SELECT *
FROM DAYS;
```

Решение:

```
CREATE TABLE трафик(
начало DATE,
конец DATE,
байт NUMBER
);
WITH src(tr_start, tr_finish, tr_bytes) AS (
   SELECT to_date('10.1.2018 11:08', 'dd.mm.yyyy hh24:mi'), to_date('12.1.2018 01:24', 'dd.mm.yyyy
hh24:mi'), 3555
      FROM dual UNION ALL
   SELECT to_date('11.1.2018 12:00', 'dd.mm.yyyy hh24:mi'), to_date('11.1.2018 13:13', 'dd.mm.yyyy
hh24:mi'), 1642
      FROM dual UNION ALL
   SELECT to_date('12.1.2018 00:05', 'dd.mm.yyyy hh24:mi'), to_date('12.1.2018 1:02', 'dd.mm.yyyy
hh24:mi'), 655
      FROM dual UNION ALL
   SELECT to_date('10.1.2018 23:07', 'dd.mm.yyyy hh24:mi'), to_date('11.1.2018 14:11', 'dd.mm.yyyy
hh24:mi'), 2555
      FROM dual
),
session_info AS (
--Выведем для каждой сессии дату ее начала, дату конца, сколько байт передано,
--сколько байт передается за день(дробное число)
   SELECT tr_start, tr_finish, tr_bytes, tr_bytes / (tr_finish - tr_start) band
   FROM src
```

```
),
--расчет для каждого трайика, сколько передано за целый день
traffic_per_day(tr_day, tr_bytes, sum_bytes, band, tr_finish, all_bytes) AS (
   SELECT trunc(tr_start, 'dd')+1 AS tr_day, --текущий день (+1, так как на этот день рассчитано здесь)
      (CASE WHEN trunc(tr_start, 'dd')=trunc(tr_finish, 'dd') THEN tr_bytes
      --если трафик открылся и закрылся в один день, то за этот день передан весь трафик
      --иначе только часть от общего = разница между началом и следующим днем, умноженная
      --на band - скорость передачи
      ELSE trunc( (trunc(tr_start, 'dd')+1-tr_start)*band, 0) END) AS tr_bytes,
      0 AS sum_bytes, --сумма трафика за все дни
      band AS band, --скорость передачи байт
      tr_finish AS tr_finish, --дата окончания
      tr_bytes AS all_bytes --всего байт в трафике
   FROM session_info
   UNION ALL
   SELECT tr_day+1, --увеличиваем текущий день на 1
      (CASE WHEN trunc(tr_finish, 'dd')>tr_day
      THEN band --если этот день не последний, то трафик за день = band
      --иначе - остатку трафика
      ELSE all_bytes-sum_bytes-tr_bytes END),
      sum_bytes+tr_bytes, --суммируем трафик за все дни с текущим
      band,
      tr_finish,
      all_bytes
   FROM traffic_per_day
   WHERE tr_day<=tr_finish --выходим, если достигли последней даты
),
all_days AS ( --формируем дни за введенный месяц
   SELECT trunc(inp_date, 'mm')+level-1 AS tr_day
   FROM (SELECT to_date('01.2018', 'mm.yyyy') inp_date FROM dual)
   CONNECT BY trunc(inp_date, 'mm')+level-1<=last_day(inp_date)
)
SELECT ad.tr_day, nvl(to_char(sum(tr.tr_bytes)), ' ') AS trafic
FROM all_days ad LEFT JOIN traffic_per_day tr ON (ad.tr_day=tr.tr_day-1)
--дни в таблице трафиков (traffic_per_day) оказались смещены на 1, так как
--мы сразу расчитывали трафик за первый день поэтому вычитаем этот день
GROUP BY ad.tr_day
   ORDER BY ad.tr_day ASC
```

2. Имеется таблица со столбцом типа varchar2(2000), содержащем информацию о названиях документов, например:

   Документоплатабанк
   Сотрудниквыплата
   Документналогбанксотрудник
   и т.д.

   Кроме того имеется таблица с сокращениями отдельных выражений. Например:

Полное выражение	**Укороченное выражение**
Документ	Док
Сотрудник	Сотр
Банк	Б

Оплата	Оп

Требуется вывести полные названия документов и сокращенные.

Пример результата:

Полное название документа	**Сокращенное название**
Документоплатабанк	ДокОпБ
Сотрудниквыплата	Сотрвыплата
Документналогбанксотрудник	ДокналогБСотр

Решение:

```
with
   ishodnik as (select 'Документоплатабанк' as some_text from dual
     union all select 'Сотрудниквыплата' as some_text from dual
     union all select 'Документналог банксотрудник' as some_text from dual),

    sokr as (select 'Документ' as fulls, 'Док' as shorts from dual
   union all select 'Сотрудник' as fulls, 'Сотр' as shorts from dual
   union all select 'Банк' as fulls, 'Б' as shorts from dual
   union all select 'Оплата' as fulls, 'Оп' as shorts from dual)

    ,num_ishodnik as (Select 0 as x, some_text from ishodnik)
    ,num_sokr as (Select rownum y, fulls,shorts from sokr)
    ,tab (x, some_text, cutted) AS (
--выбираем все строки, для каждой рекурсия будет применять отдельно
                SELECT x,some_text, some_text as cutted FROM num_ishodnik
                UNION ALL
--номер проверки увеличивается, все «длинные»вхождения заменяются на «короткие»
                SELECT x+1,some_text, replace(replace(cutted,ns.fulls,ns.shorts),lower(ns.fulls),ns.shorts)
                FROM tab join num_sokr ns on tab.x+1 = ns.y
--остановимся, когда дойдем до последнего сокращения
                WHERE  x <= (select max(y) from num_sokr))
SELECT some_text as "Полное название документа",cutted "Сокращенное название" FROM tab
where x = (select max(y) from num_sokr);
```

Решение2:

```
WITH dat(str) AS (
select 'Документоплатабанк' from dual
union all select 'Сотрудниквыплата' from dual
union all select 'Документналог банксотрудник' from dual
), abbrs(word, abbr) AS (
select 'Документ' , 'Док'  from dual
union all select 'Сотрудник' as fulls, 'Сотр' as shorts from dual
union all select 'Банк' as fulls, 'Б' as shorts from dual
union all select 'Оплата' as fulls, 'Оп' as shorts from dual
), R(s, str, abbred) AS (
-- Заменяем имеющиеся слова (если такие есть)
SELECT str, str, CASE WHEN word IS NOT NULL THEN REGEXP_REPLACE(str, word, abbr, 1, 1, 'i')
            ELSE NULL END
FROM dat LEFT JOIN abbrs ON (REGEXP_INSTR(dat.str, abbrs.word, 1, 1, 0, 'i')!= 0)
UNION ALL
-- От каждой замены заменяем ещё раз и ещё, пока есть, что заменять
SELECT R.s, R.abbred, CASE WHEN abbrs.word IS NOT NULL THEN REGEXP_REPLACE(R.abbred,
abbrs.word, abbrs.abbr, 1, 1, 'i')
```

```
            ELSE NULL END
FROM R  LEFT JOIN abbrs ON (REGEXP_INSTR(R.abbred, abbrs.word, 1, 1, 0, 'i')!= 0)
WHERE abbred IS NOT NULL -- Нечего больше заменять
)
SELECT DISTINCT s, str
FROM R
WHERE abbred IS NULL; -- Полностью заменённые аббревиатуры
```

Решение3:

Создадим таблицы как в примере. Таблицу с информацией:

```
create table inf (names varchar2(2000));
insert into inf values (Документоплатабанк);
insert into inf values (Сотрудниквыплата);
insert into inf values (Документналог банксотрудник);
```

И таблицу с сокращениями:

```
create table sok (poln varchar2(2000), sokr varchar2(2000));
insert into sok values (Документ,Док);
insert into sok values (Сотрудник,Сотр);
insert into sok values (Банк,Б);
insert into sok values (Оплата,Оп);
with input as ( select rownum n, names from inf) ,--берем из таблички с информацией значения плюс порядковый номер
t_sokr as ( select lower(poln) poln, sokr from sok ),--берем из таблицы сокращений полное(в нижнем регистре) и сокр
dop as ( select count(*) over() cnt,--подсчет кол-ва
row_number() over(order by 1) numb,
poln,sokr from t_sokr),--вспомогательная таблица
rec (cnt,numb,n,names,rep) as (select cnt,numb,n,names,--рекурсивный with
replace(lower(names),poln,sokr) --ищем сокращенное в полном
from input,dop where numb = 1
union all --объединяем с другими значениями
select dop.cnt,dop.numb,n, names,
replace(rep,poln,sokr)
from rec,dop where dop.numb = rec.numb + 1 )
select distinct initcap(names) "Полное название документа", rep "Сокращенное название документа"
from rec where numb = cnt"
```

Решение: (ОЮ) 17в

Создадим таблицы как в примере. Таблицу с информацией:

```
create table inf (names varchar2(2000));
insert into inf values ('Документоплатабанк');
insert into inf values ('Сотрудниквыплата');
insert into inf values ('Документналог банксотрудник');
```

И таблицу с сокращениями:

```
create table sok (poln varchar2(2000), sokr varchar2(2000));
insert into sok values ('Документ','Док');
insert into sok values ('Сотрудник','Сотр');
insert into sok values ('Банк','Б');
insert into sok values ('Оплата','Оп');
with input as ( select rownum n, names from inf) ,--берем из таблички с информацией значения плюс порядковый номер
t_sokr as ( select lower(poln) poln, sokr from sok ),--берем из таблицы сокращений полное(в нижнем регистре) и сокр
dop as ( select count(*) over() cnt,--подсчет кол-ва
```

```
row_number() over(order by 1) numb,
poln,sokr from t_sokr),--вспомогательная таблица
rec (cnt,numb,n,names,rep) as (select cnt,numb,n,names,--рекурсивный with
replace(lower(names),poln,sokr) —ищем сокращенное в полном
from input,dop where numb = 1
union all —объединяем с другими значениями
select dop.cnt,dop.numb,n, names,
replace(rep,poln,sokr)
from rec,dop where dop.numb = rec.numb + 1 )
select distinct initcap(names) "Полное название документа", rep "Сокращенное название
документа"
from rec where numb = cnt;
```

Решение:

```
WITH dat AS (
   SELECT 'Документоплатабанк' AS str FROM dual UNION ALL
   SELECT 'Сотрудниквыплата' FROM dual UNION ALL
   SELECT 'Документналогбанксотрудник' FROM dual
),
abbrs AS (
   SELECT 'Документ' AS str , 'Док' AS abbr  FROM dual UNION ALL
   SELECT 'Сотрудник', 'Сотр' FROM dual UNION ALL
   SELECT 'Банк', 'Б' FROM dual UNION ALL
   SELECT 'Оплата', 'Оп' FROM dual
),
--пронумерованные аббревиатуры
abbrs_nums AS (
   SELECT rownum rn, abbrs.*
   FROM abbrs
),
--проход по каждой исходной строке столько раз, сколько аббревиатур
--и замена при каждом проходе
repl_abs(dat_str, lvl) AS (
   SELECT str AS dat_str, 1 AS lvl
   FROM dat
   UNION ALL
   SELECT REGEXP_REPLACE(dat_str,
            (SELECT str FROM abbrs_nums WHERE ra.lvl=rn), --замена
            (SELECT abbr FROM abbrs_nums WHERE ra.lvl=rn),
            1, 1, 'i'
         ),
         lvl+1
   FROM repl_abs ra
   WHERE lvl<=(SELECT count(*) FROM abbrs)
)
SELECT dat_str
FROM repl_abs
WHERE lvl=(SELECT count(*) FROM abbrs)+1;
```

Решение 27 в (кож):

```
CREATE TABLE DOCNAME(NAME VARCHAR2(2000));
INSERT INTO DOCNAME VALUES('Документоплатабанк');
INSERT INTO DOCNAME VALUES('Сотрудниквыплата');
INSERT INTO DOCNAME VALUES('Документналог банксотрудник ');
WITH SHORTNAME AS(/*Таблица с сокращениями отдельных
```

```
выражений*/
SELECT 'Документ' NAME,'Док' SHNAME FROM DUAL
UNION ALL
SELECT 'Сотрудник','Сотр' FROM DUAL
UNION ALL
SELECT 'Банк','Б' FROM DUAL
UNION ALL
SELECT 'Оплата','Оп' FROM DUAL),
SHORTNAMEWITHRN AS(/*Таблица SHORTNAME с уникальным номером
позиции RN, для того, чтобы потом можно выбрать из них нужную нам
строку*/
SELECT NAME,SHNAME,ROWNUM RN
FROM SHORTNAME),
COUNTOFSHNAME AS(/*Таблица, содержащая количество строк
сокращенных имен*/
SELECT COUNT(*) COFN
FROM SHORTNAME),
DOCNAMEWITHCOFN AS(/*Таблица с именами документов с
количеством сток, которые содержат сокращенные названия*/
SELECT LOWER(NAME) NAME,COFN
FROM DOCNAME
CROSS JOIN COUNTOFSHNAME),
RESULT(NAME,COFN,NUM,NEWNAME) AS(/*Таблица с результатом, где
поочередно заменено полные названия в полном название документа на
скоращенные*/
SELECT NAME,COFN,1,REGEXP_REPLACE(NAME,LOWER((SELECT NAME
FROM SHORTNAMEWITHRN WHERE RN=1)),(SELECT SHNAME FROM
SHORTNAMEWITHRN WHERE RN=1)) NEWNAME
FROM DOCNAMEWITHCOFN
UNION ALL
SELECT NAME,COFN,NUM+1,REGEXP_REPLACE(NEWNAME,LOWER((SELECT
NAME FROM SHORTNAMEWITHRN WHERE RN=NUM+1)),(SELECT SHNAME FROM
SHORTNAMEWITHRN WHERE RN=NUM+1))
FROM RESULT
WHERE COFN>=NUM+1)
SELECT NVL(D.NAME,'Передано полное название документа, как пустая
строка') "Полное название документа",NVL(R.NEWNAME,' ') "Сокращенное
название"
FROM RESULT R
LEFT JOIN DOCNAME D
ON LOWER(D.NAME)=R.NAME
WHERE R.NUM=(SELECT COFN FROM COUNTOFSHNAME)
```

БИЛЕТ 6

1. Для произвольной строки, состоящей из открывающих и закрывающих скобок написать запрос для вывода всех слов максимальной длины, представляющих правильные скобочные записи. Например, для строки (()(() ответ должен быть:

()()

(())

Задачу решить с использованием раздела Model.

```
define str = '(())(())((()()('
with symb(s,lvl) as (
   select substr('&str',level,1), level
   from dual
   connect by level <= length('&str')
   ), all_comb(comb) as (
   select distinct replace(sys_connect_by_path(s, ' '), ' ')
   from symb
   connect by prior lvl < lvl
   ), mdl as (
   select comb, t, res
   from all_comb
   where length(comb)>1
   model
   dimension by (comb)
    measures (comb cb, cast (0 as number(2)) as t, cast (1 as number(1))
                        as res)
   rules iterate(100) (
      t[any] = case
         when substr(cb[cv()], iteration_number+1, 1) = '(' and res[cv()] !=
                        0 then t[cv()]+1
         when substr(cb[cv()], iteration_number+1, 1) = ')' and res[cv()] !=
                        0 then t[cv()]-1
         else t[cv()]
         end,
      res[any] = case
         when t[cv()] < 0 or res[cv()] = 0 then 0
         else 1
      end
      )
   ), res as (
   select comb, length(comb) len
   from mdl
   where t=0 and res=1
   )
```

```
select comb
from res
where len = (select max(len) from res);
```

SECOND VARIANT

```
WITH scr AS(
                    SELECT '((()(((()))((()' AS str
                    FROM dual
                    ),
                    divide as (SELECT SUBSTR(STR,LEVEL,1) PARTS, LEVEL AS
                    POS
                    FROM SCR
                    CONNECT BY LEVEL<=LENGTH(STR)),
                    ALL_COMBS AS (SELECT
                    REPLACE(SYS_CONNECT_BY_PATH(PARTS, '|'),'|') COMBS,
                    ROWNUM RN
                    FROM DIVIDE
                    CONNECT BY PRIOR POS<POS),
                    CHECKED AS ( SELECT COMBS,RN, CNT, FLAG
                    FROM ALL_COMBS
                    MODEL
                    PARTITION BY(RN)
                    DIMENSION BY (1 AS I)
                    MEASURES(COMBS, CAST(0 AS NUMBER(8)) AS CNT, CAST (0
                    AS NUMBER(2)) AS FLAG)
                    RULES UPSERT ALL(

                    CNT[FOR I FROM 1 TO LENGTH(COMBS[1]) INCREMENT 1]=
                    CASE WHEN SUBSTR(COMBS[1],CV(I),1)='(' THEN CASE WHEN
                    NVL(CNT[CV()-1],0)<0 THEN -1 ELSE NVL2(CNT[CV()-
                    1],CNT[CV()-1]+1,1) END
                    ELSE CASE WHEN NVL(CNT[CV()-1],-1)<0 THEN -1 ELSE
                    NVL2(CNT[CV()-1],CNT[CV()-1]-1,-1) END END,
                    FLAG[LENGTH(COMBS[1])]=1,
                    COMBS[LENGTH(COMBS[1])]=COMBS[1]
                    )
                    ORDER BY FLAG DESC NULLS LAST, RN)
                    SELECT COMBS
                    FROM (SELECT DISTINCT COMBS, LENGTH(COMBS) LC,
                    MAX(LENGTH(COMBS)) OVER() MC
                    FROM CHECKED
                    WHERE FLAG=1 AND CNT=0)
                    WHERE LC=MC;
```

2. Имеется таблица D_V с первым столбцом Dat типа DATE (первичный ключ) и вторым столбцом Val типа NUMBER. Пример (строки упорядочены по первому столбцу):

DAT	VAL
01-08-08	232
02-08-08	
10-08-08	182
11-08-08	

21-08-08	240
22-08-08	
23-08-08	

Требуется написать запрос для получения на основе таблицы D_V следующей таблицы:

DAT	MAX_VAL
01-08-08	232
02-08-08	232
10-08-08	182
11-08-08	182
21-08-08	240
22-08-08	240
23-08-08	240

Данная результирующая таблица должна быть упорядочена по Dat, но вместо пустых значений, которые присутствовали в столбце VAL отсортированной по DAT исходной таблицы, в столбце MAX_VAL результирующей таблицы, должны присутствовать значения столбца из предыдущей строки.

РЕШЕНИЕ 1

```
WITH D_V AS (
    SELECT to_date('01-08-08', 'dd-mm-yy') DAT, 232 val FROM dual UNION
    SELECT to_date('02-08-08', 'dd-mm-yy'), NULL FROM dual UNION
    SELECT to_date('10-08-08', 'dd-mm-yy'), 182  FROM dual UNION
    SELECT to_date('11-08-08', 'dd-mm-yy'), NULL FROM dual UNION
    SELECT to_date('21-08-08', 'dd-mm-yy'), 240  FROM dual UNION
    SELECT to_date('22-08-08', 'dd-mm-yy'), NULL FROM dual UNION
    SELECT to_date('23-08-08', 'dd-mm-yy'), NULL FROM dual
) select DAT, case when val is null then
       (select TMP.VAL
        from D_V TMP
        where TMP.DAT = (select max (DAT)
                         from D_V
                         where DAT < DV.DAT
                               and VAL is not null
                        )
       ) else val end as val
FROM D_V dv
ORDER BY DAT;
```

РЕШЕНИЕ 2

```
WITH dataval AS (
    SELECT to_date('01-08-08', 'dd-mm-yy') DAT, 232 val FROM dual UNION
    SELECT to_date('02-08-08', 'dd-mm-yy'), NULL FROM dual UNION
    SELECT to_date('10-08-08', 'dd-mm-yy'), 182  FROM dual UNION
    SELECT to_date('11-08-08', 'dd-mm-yy'), NULL FROM dual UNION
    SELECT to_date('21-08-08', 'dd-mm-yy'), 240  FROM dual UNION
    SELECT to_date('22-08-08', 'dd-mm-yy'), NULL FROM dual UNION
    SELECT to_date('23-08-08', 'dd-mm-yy'), NULL FROM dual ),
numered as (select dat,val, row_number() over(order by dat) as rn
                from dataval)
```

```
select dat,val
from numered
model
dimension by(rn)
measures(dat,val)
rules (val[any] = case when val[cv()] is null then val[cv()-1] else
val[cv()] end);
```

4. Создайте таблицу для хранения каталога товаров:

create table catalog(text varchar2(4000));

где text – информация о товарах, заданная в формате:

Код товара1/Тип товара1-Наименование товара1:Цена товара1;Код товара2/Тип товара2-Наименование товара2:Цена товара2;Код товара3/Тип товара3-Наименование товара3:Цена товара3;...;Код товараN/Тип товараN-Наименование товараN:Цена товараN

Требования к формату информации о товарах:

- Товары разделены точкой с запятой, после последнего товара точки с запятой нет;
- Код товара отделяется символом «слэш», имеет длину от 1 до 6 символов, допустимы только цифры;
- Тип товара имеет нефиксированную длину, отделяется символом «минус» (коротким тире), содержит любые символы;
- Наименование товара имеет нефиксированную длину, отделяется двоеточием, содержит любые символы;
- Цена товара может иметь дробное значение, при этом целая и дробная часть могут разделяться точкой или запятой;
- В коде, типе, наименовании, цене, недопустимо присутствие любого из перечисленных символов разделения (не допускаются символы точка с запятой, «слэш», «минус», двоеточие).

Заполните таблицу данными о товарах:

insert into catalog values ('125/refrigerator-Indesit SB200 T:17999.99;50/microwave-Samsung MT479:7499,99;103320/teakettle-Bosch TWK189:4890,32');

insert into catalog values ('05/pan-Tefal 040 80:849,00;125/pan-Tefal E20 60:3599,2;434031/iron-Braun Texstyle 535:5490,01');

commit;

Одним SQL-Запросом необходимо вывести таблицу товаров, отсортированную по возрастанию кода товара, затем по типу товара.

Формат результата

Код товара	**Тип товара**	**Наименование товара**	**Цена товара**
000005	Pan	Tefal 040 80	849
000050	Microwave	Samsung MT479	7499.99
000125	Pan	Tefal E20 60	3599.2

000125	Refrigerator	Indesit SB200 T	17999.99
...	...	...	...

В случае, когда длина кода товара меньше шести символов, необходимо дополнять код товара незначащими нулями слева до максимальной длины шесть символов.

Тип колонки цены товара в результирующем наборе должен быть числовым, тип остальных колонок – строковым.

```
create table catalog(text varchar2(4000));

INSERT INTO catalog(text)
VALUES ('Код товара1/Тип товара1-Наименование товара1:Цена товара1;
Код товара2/Тип товара2-Наименование товара2:Цена товара2;
Код товара3/Тип товара3-Наименование товара3:Цена товара3;
Код товара67/Тип товараN-Наименование товараN:Цена товараN;
25/Мебель-Кровать:20000;
28/Мягкая мебель-Односпальная Кровать:200.00');

WITH
temp AS (
SELECT DISTINCT REGEXP_SUBSTR(text,'[^;]+',1,level) str
FROM catalog
CONNECT BY REGEXP_SUBSTR(text,'[^;]+',1,level) IS NOT NULL
)
SELECT LPAD(REGEXP_SUBSTR(str,'\d{1,6}+'),6,0) "Код товара",
    REGEXP_SUBSTR(str,'(\w|\s)+',1,2) "Тип товара",
    LTRIM(REGEXP_SUBSTR(str,'-[^:]+'),'-') "Наименование товара",
    LTRIM(REGEXP_SUBSTR(str,':\d+(\.|,)?\d+'),':') "Цена товара"
FROM temp;
```

БИЛЕТ 7

6. Для каждого месяца, в котором принимались на работу сотрудники, найти 3 ближайших после данного "месяца-двойника".

"Двойниками" считать такие месяцы, которые и начинаются в один и тот же день недели, и заканчиваются в один и тот же день недели.

Выводить: 1. Фамилию сотрудника;
2. Месяц, в котором сотрудник был принят на работу в формате "mon year";
3.-5. Три ближайших "месяца-двойника" в формате "mon year"

Решение:

```
WITH
n (l) AS (
SELECT 1 l FROM DUAL
UNION ALL
SELECT l+1 FROM n WHERE l <= 3000
),
```

```
t AS
(
SELECT last_name, hire_date,
to_char(trunc(hire_date,'MM'),'DAY') first_day,
to_char(last_day(hire_date),'DAY') last_day
FROM employees
),
next_doubles AS
(
SELECT last_name, hire_date, first_day, last_day,
 l added_months,
 dense_rank() OVER (PARTITION BY last_name, hire_date ORDER BY l) r
FROM t, n
WHERE to_char(trunc(ADD_MONTHS(hire_date, l),'MM'),'DAY') = first_day AND
to_char(last_day(ADD_MONTHS(hire_date, l)),'DAY') = last_day
),
res AS
(
SELECT last_name, hire_date,
ADD_MONTHS(hire_date, added_months) next_d, r
FROM next_doubles
WHERE r<=3
ORDER BY LAST_NAME
)

SELECT last_name, to_char(hire_date, 'mon YYYY') hire_date,
  to_char((SELECT next_d FROM res
  WHERE res.last_name = e.last_name
  AND res.hire_date = e.hire_date AND r = 1), 'mon YYYY') as "Month 1",
   to_char((SELECT next_d FROM res
  WHERE res.last_name = e.last_name
  AND res.hire_date = e.hire_date AND r = 2), 'mon YYYY') as "Month 2",
   to_char((SELECT next_d FROM res
  WHERE res.last_name = e.last_name
  AND res.hire_date = e.hire_date AND r = 3), 'mon YYYY') as "Month 3"
FROM employees e;
```

Решение:

```
WITH RESULT(RN, N, DT, D, DNE, D1, DATE1E, DATE2E, DATE3E, FLAG) AS (
  SELECT 1 RN, LAST_NAME N,
       HIRE_DATE DT,
       TO_CHAR(TRUNC(HIRE_DATE,'MONTH'),'fmD') D,
       TO_CHAR(TRUNC(ADD_MONTHS(HIRE_DATE,1),'MONTH'),'fmD') DNE,
       ADD_MONTHS(HIRE_DATE,1) D1,
       NULL DATE1E,
       NULL DATE2E,
       NULL DATE3E,
       0 FLAG
  FROM EMPLOYEES
  UNION ALL
  SELECT RN+1, N, DT, D, DNE, ADD_MONTHS(D1,1) D1,
       CASE
         WHEN TO_CHAR(TRUNC(D1,'MONTH'),'fmD') = D AND
TO_CHAR(TRUNC(ADD_MONTHS(D1,1),'MONTH'),'fmD') = DNE AND FLAG = 0
         THEN TO_CHAR(D1,'mon year')
```

```
      ELSE DATE1E
    END,
    CASE
      WHEN TO_CHAR(TRUNC(D1,'MONTH'),'fmD') = D AND
TO_CHAR(TRUNC(ADD_MONTHS(D1,1),'MONTH'),'fmD') = DNE AND FLAG = 1
      THEN TO_CHAR(D1,'mon year')
      ELSE DATE2E
    END,
    CASE
      WHEN TO_CHAR(TRUNC(D1,'MONTH'),'fmD') = D AND
TO_CHAR(TRUNC(ADD_MONTHS(D1,1),'MONTH'),'fmD') = DNE AND FLAG = 2
      THEN TO_CHAR(D1,'mon year')
      ELSE DATE3E
    END,
    CASE
      WHEN TO_CHAR(TRUNC(D1,'MONTH'),'fmD') = D AND
TO_CHAR(TRUNC(ADD_MONTHS(D1,1),'MONTH'),'fmD') = DNE
      THEN FLAG+1
      ELSE FLAG
    END
  FROM RESULT
  WHERE RN < 200)
SELECT distinct n last_name, TO_CHAR(dt,'mon year') hire_date, DATE1E AS "Первый месяц",DATE2E
AS "Второй месяц",DATE3E AS "Третий месяц"
FROM RESULT
WHERE DATE1E is not null and DATE2E is not null and DATE3E is not null;
```

7. Для заданного списка чисел найти все такие его разбиения на два непересекающихся подсписка, что модуль разности сумм чисел в первом и втором подсписке минимально отличаются друг от друга.

   Например, для списка

   1,-1,1,2,3,6.5

   результат должен быть

Исходный список	Подсписок 1	Подсписок 2	Модуль разницы сумм
1,-1,1,2,3,6.5	6.5	-1,1,1,2,3	0,5
	1,2,3	-1,1,6.5	0,5

Результирующие списки должны быть отсортированы по возрастанию. Количество элементов Подсписка 1 должно быть меньше или равно количеству элементов Подсписка 2.

Решение:

```
WITH VAL (n) AS (
SELECT 1 FROM dual UNION
SELECT -1 FROM dual UNION
SELECT 2 FROM dual UNION
SELECT 3 FROM dual UNION
SELECT 6.5 FROM dual
),
VAL2 (n, rn) AS (
SELECT n, RANK() OVER (ORDER BY n)
FROM VAL
),
RES (mx, sm1, sm2, grp) AS (
```

```
SELECT
(SELECT MAX(n) FROM VAL2), (SELECT MAX(n) FROM VAL2), 0, 1
FROM dual
UNION ALL
SELECT
(SELECT MAX(n) FROM VAL2 WHERE n < mx),
CASE WHEN ABS(sm1 + (SELECT MAX(n) FROM VAL2 WHERE n < mx) - sm2) <= ABS(sm1 - (sm2 + (SELECT
MAX(n) FROM VAL2 WHERE n < mx)))
THEN sm1 + (SELECT MAX(n) FROM VAL2 WHERE n < mx)
ELSE sm1
END,
CASE WHEN ABS(sm1 + (SELECT MAX(n) FROM VAL2 WHERE n < mx) - sm2) <= ABS(sm1 - (sm2 + (SELECT
MAX(n) FROM VAL2 WHERE n < mx)))
THEN sm2
ELSE sm2 + (SELECT MAX(n) FROM VAL2 WHERE n < mx) END,
CASE WHEN ABS(sm1 + (SELECT MAX(n) FROM VAL2 WHERE n < mx) - sm2) <= ABS(sm1 - (sm2 + (SELECT
MAX(n) FROM VAL2 WHERE n < mx)))
THEN 1
ELSE 2 END
FROM RES
WHERE mx != (SELECT MIN(n) FROM VAL2)
)
, grps(list_of_numbers, grp, tgrp) AS (
SELECT LISTAGG(mx, ' ') WITHIN GROUP(ORDER BY mx), grp,
CASE WHEN (SELECT COUNT(*) FROM RES WHERE grp = r_.grp)
<= (SELECT COUNT(*) FROM RES WHERE grp != r_.grp) THEN 1 ELSE 2 END

FROM RES r_
GROUP BY grp)
SELECT TRANSLATE((SELECT LISTAGG(n, ' ') WITHIN GROUP (ORDER BY n) FROM VAL), ' ,', ',.') AS
"Исходный список",
TRANSLATE((SELECT list_of_numbers FROM grps WHERE tgrp = 1), ' ,', ',.') AS "Подсписок 1",
TRANSLATE((SELECT list_of_numbers FROM grps WHERE tgrp = 2), ' ,', ',.') AS "Подсписок 2",
(SELECT ABS((SELECT MAX(sm1) FROM RES) - (SELECT MAX(sm2) FROM RES)) FROM dual) AS "Модуль
разницы сумм"
FROM dual;
```

8. Создать запрос для определения среди таблиц Вашей схемы таких таблиц, названия которых получаются друг из друга циклическим сдвигом символов.

Пример результата:

Номер	Таблица 1	Таблица 2
1	BAA	ABA
2	AAB	ABA
3	ABA	BAA
4	BAA	AAB
5	ABA	AAB
6	AAB	BAA

```
CREATE TABLE BAA(NUM NUMBER(2));
CREATE TABLE ABA(NUM NUMBER(2));
WITH TAB AS (
SELECT table_name
FROM user_tables )
```

SELECT ROWNUM AS "Номер", T1.table_name AS "Таблица 1", T2.table_name AS "Таблица 2"
FROM TAB T1 INNER JOIN TAB T2
ON (LENGTH(T1.table_name) = LENGTH(T2.table_name))
WHERE T1.table_name != T2.table_name AND
INSTR(T2.table_name||T2.table_name, T1.table_name) != 0;

БИЛЕТ 8

2. Используя словарь данных, получить информацию о первичных ключах и подчиненных таблицах всех таблиц Вашей схемы:

Имя таблицы	Список столбцов первичного ключа	Список столбцов с ограничением уникальности	Список подчиненных таблиц

В списках имена столбцов первичного ключа и таблиц вывести через запятую по алфавиту. Если таблица не имеет подчиненных таблиц, вывести – Подчиненных таблиц не имеет. Если таблица не имеет первичного ключа, вывести – Первичного ключа не имеет. Аналогично, если таблица не имеет ограничения уникальности, вывести – Ограничения уникальности в таблице нет. Имена столбцов композитных ограничений уникальности заключить в круглые скобки.

Задачу решить без использования функций Listagg и Wm_concat.

```
WITH src AS (SELECT  ALL_T.TABLE_NAME SUPP,
C2.TABLE_NAME CUST,
LAG(C2.TABLE_NAME) OVER (PARTITION BY ALL_T.TABLE_NAME ORDER BY ALL_T.TABLE_NAME)
ONE_MORE
FROM USER_CONSTRAINTS C1 INNER JOIN
USER_CONSTRAINTS C2 ON C2.R_CONSTRAINT_NAME = C1.CONSTRAINT_NAME
RIGHT JOIN
USER_TABLES ALL_T  ON ALL_T.TABLE_NAME = C1.TABLE_NAME),
t AS (
SELECT  SUPP,
TRIM( LEADING ',' FROM (SYS_CONNECT_BY_PATH(CUST,',')) ) AS LISTING,
LEVEL AS l
FROM src
START WITH ONE_MORE IS NULL
CONNECT BY NOCYCLE PRIOR CUST = ONE_MORE AND PRIOR SUPP = SUPP),
res AS (
SELECT  SUPP,LISTING
FROM t M
WHERE l = (SELECT MAX(l)
FROM t
WHERE SUPP=M.SUPP
GROUP BY SUPP)),
p_col AS (
SELECT C3.TABLE_NAME SYS_TABLE,
COLUMN_NAME PK_COL,
LAG(COLUMN_NAME) OVER (PARTITION BY C3.TABLE_NAME ORDER BY C3.TABLE_NAME) COL_COL
FROM USER_CONSTRAINTS C3 JOIN
USER_CONS_COLUMNS C4 ON C3.CONSTRAINT_NAME = C4.CONSTRAINT_NAME
WHERE C3.CONSTRAINT_TYPE='P'),

u_col AS (
```

```
SELECT C3.TABLE_NAME SYS_TABLE,
COLUMN_NAME U_COL,
LAG(COLUMN_NAME) OVER (PARTITION BY C3.TABLE_NAME ORDER BY C3.TABLE_NAME) UCOL_COL
FROM USER_CONSTRAINTS C3 JOIN
USER_CONS_COLUMNS C4 ON C3.CONSTRAINT_NAME = C4.CONSTRAINT_NAME
WHERE C3.CONSTRAINT_TYPE='U'),
p_col2 AS (
SELECT SYS_TABLE,
TRIM( LEADING ',' FROM (SYS_CONNECT_BY_PATH(PK_COL,',')) ) LISTING_COL,
LEVEL LEV
FROM p_col
START WITH COL_COL IS NULL
CONNECT BY NOCYCLE PRIOR PK_COL = COL_COL AND PRIOR SYS_TABLE = SYS_TABLE),
u_col2 AS (
SELECT SYS_TABLE,
TRIM( LEADING ',' FROM (SYS_CONNECT_BY_PATH(U_COL,',')) ) ULISTING_COL,
LEVEL LEV
FROM u_col
START WITH UCOL_COL IS NULL
CONNECT BY NOCYCLE PRIOR U_COL = UCOL_COL AND PRIOR SYS_TABLE = SYS_TABLE),
p_col3 AS (
SELECT  SYS_TABLE,
LISTING_COL
FROM p_col2 L
WHERE LEV = (SELECT MAX(LEV)
FROM p_col2
WHERE SYS_TABLE = L.SYS_TABLE
GROUP BY SYS_TABLE)),
u_col3 AS (
SELECT  SYS_TABLE,
ULISTING_COL
FROM u_col2 L
WHERE LEV = (SELECT MAX(LEV)
FROM u_col2
WHERE SYS_TABLE = L.SYS_TABLE
GROUP BY SYS_TABLE))
SELECT DISTINCT SUPP AS "Имя таблицы",
NVL(LISTING_COL,'Первичного ключа нет') AS "Столбцы первичного ключа",
NVL(ULISTING_COL,'Ограничения уникальности в таблице нет') AS "Столбцы с огр. уникальности",
NVL(LISTING,'Подчиненных таблиц нет')AS "Список подчиненных таблиц"
FROM res r LEFT JOIN
p_col3 p ON r.SUPP = p.SYS_TABLE
LEFT JOIN
u_col3 u ON r.SUPP = u.SYS_TABLE;
```

Решение2:

```
WITH TPK AS (SELECT TN,
      SUBSTR(MAX(SYS_CONNECT_BY_PATH(CN, ', ')), 3) AS CN
            FROM
  (SELECT UC.TABLE_NAME AS TN,
UCC.COLUMN_NAME AS CN,ROW_NUMBER() OVER(PARTITION BY UC.TABLE_NAME ORDER BY
UCC.COLUMN_NAME) AS RN
                        FROM USER_CONSTRAINTS UC INNER JOIN
    USER_CONS_COLUMNS UCC
    ON UC.CONSTRAINT_NAME=UCC.CONSTRAINT_NAME
```

```
                    WHERE UC.CONSTRAINT_TYPE = 'P'
                    ORDER BY TN)
              START WITH RN = 1
              CONNECT BY PRIOR TN = TN AND PRIOR RN + 1 = RN
              GROUP BY TN),

TPT AS (SELECT UCC.TABLE_NAME AS TN, UC.TABLE_NAME AS S_TN
                    FROM USER_CONSTRAINTS UC INNER JOIN
  USER_CONS_COLUMNS UCC
  ON UC.R_CONSTRAINT_NAME = UCC.CONSTRAINT_NAME
                    WHERE UC.CONSTRAINT_TYPE = 'R'),


COL_N AS(SELECT TN, TNN,
 SUBSTR(MAX(SYS_CONNECT_BY_PATH(CN, ', ')), 3) CN
                      FROM
      (SELECT DISTINCT TN, TNN, CN, ROW_NUMBER() OVER(PARTITION BY TN, TNN ORDER BY TNN) RN
                       FROM(SELECT UC.TABLE_NAME AS TN,
UCC.TABLE_NAME AS TNN, UCC.COLUMN_NAME AS CN
FROM USER_CONSTRAINTS UC LEFT JOIN USER_CONSTRAINTS UC2
ON(UC.CONSTRAINT_NAME = UC2.R_CONSTRAINT_NAME)
                                  LEFT JOIN USER_CONS_COLUMNS UCC ON(UC2.CONSTRAINT_NAME =
UCC.CONSTRAINT_NAME)
                                  WHERE UC.CONSTRAINT_TYPE = 'P'))
                      START WITH RN = 1
                      CONNECT BY PRIOR TN = TN AND PRIOR RN + 1 = RN
                      GROUP BY TN, TNN)

SELECT CASE WHEN TB.TN = LAG(TB.TN,1 )OVER(ORDER BY TB.TN)
  THEN ' ' ELSE TB.TN END AS "Имя таблицы",
           CASE WHEN TPK.CN = LAG(TPK.CN,1)OVER(ORDER BY TB.TN)
  THEN ' ' ELSE TPK.CN END AS "Столбцы первичного ключа",
           NVL(TPT.S_TN,'Подчиненных таблиц нет') AS "Подчиненные таблицы",
           NVL(COL_N.CN,' ') AS "Столбцы вторичного ключа"
FROM (SELECT DISTINCT TABLE_NAME AS TN
          FROM ALL_TABLES
          WHERE OWNER = 'HR') TB LEFT JOIN TPK ON TB.TN=TPK.TN
                                       LEFT JOIN TPT ON TB.TN=TPT.TN
                                       LEFT JOIN COL_N ON
  (TB.TN = COL_N.TN AND TPT.S_TN = COL_N.TNN);
```

Решение3:

```
WITH TPK AS (SELECT TN, SUBSTR(MAX(SYS_CONNECT_BY_PATH(CN, ', ')), 3) CN
                    FROM (SELECT UC.TABLE_NAME TN,
      UCC.COLUMN_NAME CN,
      ROW_NUMBER() OVER(PARTITION BY UC.TABLE_NAME ORDER BY UCC.COLUMN_NAME) RN
                          FROM USER_CONSTRAINTS UC
    INNER JOIN USER_CONS_COLUMNS UCC
    ON UC.CONSTRAINT_NAME=UCC.CONSTRAINT_NAME
                          WHERE UC.CONSTRAINT_TYPE = 'P'
                          ORDER BY TN)
                    START WITH RN = 1
                    CONNECT BY PRIOR TN = TN AND PRIOR RN + 1 = RN
                    GROUP BY TN),
```

```
    TPT AS (SELECT TN, SUBSTR(MAX(SYS_CONNECT_BY_PATH(S_TN, ', ')), 3) S_TN
              FROM (SELECT TN, S_TN,
ROW_NUMBER() OVER(PARTITION BY TN ORDER BY S_TN) RN
FROM( SELECT DISTINCT UCC.TABLE_NAME TN, UC.TABLE_NAME S_TN
                                FROM USER_CONSTRAINTS UC
 INNER JOIN USER_CONS_COLUMNS UCC
ON UC.R_CONSTRAINT_NAME = UCC.CONSTRAINT_NAME
                                WHERE UC.CONSTRAINT_TYPE = 'R'))
            START WITH RN = 1
            CONNECT BY PRIOR TN = TN AND PRIOR RN + 1 = RN
            GROUP BY TN),

    TU AS (SELECT TN,
CASE WHEN MAX(RN)>1
THEN '('|| SUBSTR(MAX(SYS_CONNECT_BY_PATH(CN, ', ')), 3)|| ')'
ELSE SUBSTR(MAX(SYS_CONNECT_BY_PATH(CN, ', ')), 3) END UN
FROM (SELECT UC.TABLE_NAME TN, UCC.COLUMN_NAME CN, ROW_NUMBER() OVER(PARTITION BY
UC.TABLE_NAME ORDER BY UCC.COLUMN_NAME) RN
                        FROM USER_CONSTRAINTS UC
       INNER JOIN USER_CONS_COLUMNS UCC
       ON UC.CONSTRAINT_NAME=UCC.CONSTRAINT_NAME
                        WHERE UC.CONSTRAINT_TYPE = 'U'
                        ORDER BY TN)
              START WITH RN = 1
              CONNECT BY PRIOR TN = TN AND PRIOR RN + 1 = RN
              GROUP BY TN)

SELECT TB.TN AS "Имя таблицы",
    NVL(TPK.CN,'Первичного ключа не имеет') AS "Столбцы первичного ключа",
    NVL(UN,'Ограничения уникальности в таблице нет') AS "Список ст-цов с огран.уник",
    NVL(TPT.S_TN,'Подчиненных таблиц не имеет') AS "Список подчиненных таблиц"
FROM (SELECT TABLE_NAME TN
          FROM USER_TABLES) TB
 LEFT JOIN TPK ON TB.TN=TPK.TN
 LEFT JOIN TPT ON TB.TN=TPT.TN
 LEFT JOIN TU ON TB.TN=TU.TN
ORDER BY 1;
```

Решение4: (ОЮ)

```
WITH sourpk AS  (SELECT table_name,  column_name,  constraint_name,  row_number() over
(partition BY constraint_name order by column_name)rn  FROM all_cons_columns all1  WHERE
owner='HR'  AND (SELECT constraint_type  FROM all_constraints allc  WHERE
all1.constraint_name=allc.constraint_name  AND allc.table_name  =all1.table_name  AND
allc.owner  =all1.owner ) IN ('P')  ),  /*Здесь содержится информация о всех подчинённых
таблицах, хозяином которых является HR*/  sourdeptab AS  (SELECT dep.table_name deptbname,
row_number() over (partition BY dep.r_constraint_name order by dep.table_name) rn,
dom.table_name dtname  FROM all_constraints dep  INNER JOIN all_constraints dom  ON
dep.r_constraint_name=dom.constraint_name  WHERE dep.owner  ='HR'  AND dom.owner
=dep.owner  ),  /*Правильно сформированная (В строчку) информация о подчинённых таблицах*/
sourdepkon AS  (SELECT table_name,  trim(',' FROM str)str  FROM  (SELECT
sys_connect_by_path(deptbname,',') str,  level lvl,  dtname table_name  FROM sourdeptab
CONNECT BY prior rn=rn+1  AND prior dtname  =dtname  ) s1  WHERE lvl=  (SELECT MAX(lvl)
FROM  (SELECT sys_connect_by_path(deptbname,',') str,
```

```
level lvl, dtname table_name FROM sourdeptab CONNECT BY prior rn=rn+1 AND prior dtname =dtname ) s2 WHERE s1.table_name=s2.table_name ) ) , /*Правильно сформированная (В строчку) информация о пк*/ sourpkkon AS (SELECT trim(',' FROM strpk) finstr, table_name, constraint_name FROM (SELECT sys_connect_by_path(column_name,',') strpk, level lvl, table_name, constraint_name FROM sourpk s1 CONNECT BY prior rn =rn+1 AND prior table_name =table_name AND prior constraint_name=constraint_name )s3 WHERE lvl= (SELECT MAX(lvl) FROM ( (SELECT sys_connect_by_path(column_name,',') strpk, level lvl, table_name, constraint_name FROM sourpk s1 CONNECT BY prior rn =rn+1 AND prior table_name =table_name AND prior constraint_name=constraint_name )) s2 WHERE s2.table_name =s3.table_name AND s3.constraint_name=s2.constraint_name ) ) SELECT up.table_name "Имя таблицы", NVL(spk.finstr,'Для таблицы нет ПК')"Список столбцов ПК", NVL(sdt.str,'Нет подчинённых таблиц') "Список подчинённых таблиц" FROM all_tables up LEFT JOIN sourpkkon spk ON up.table_name=spk.table_name LEFT JOIN sourdepkon sdt ON sdt.table_name=up.table_name WHERE owner ='HR'
```

Решение5: (ОЮ)

```
WITH FKS AS(
SELECT CONNECT_BY_ROOT CM.TABLE_NAME TABL, CONNECT_BY_ROOT CR.TABLE_NAME AS FK,
SUBSTR(SYS_CONNECT_BY_PATH(CC.COLUMN_NAME, ', '),3) cols,
LEVEL THISLEVEL, MAX(LEVEL) OVER (PARTITION BY CM.TABLE_NAME, CR.TABLE_NAME) AS LVL
FROM USER_CONSTRAINTS CM
INNER JOIN USER_CONSTRAINTS CR ON CR.R_CONSTRAINT_NAME=CM.CONSTRAINT_NAME AND
CR.CONSTRAINT_TYPE = 'R'
INNER JOIN USER_CONS_COLUMNS CC ON CC.CONSTRAINT_NAME=CR.CONSTRAINT_NAME
WHERE CM.TABLE_NAME IN
('COUNTRIES','DEPARTMENTS','EMPLOYEES','JOB_HISTORY','JOBS','JOB_GRADES','LOCATIONS','REGIONS
')
AND CONNECT_BY_ISLEAF = 1
CONNECT BY PRIOR CC.COLUMN_NAME < CC.COLUMN_NAME AND PRIOR CM.TABLE_NAME =
CM.TABLE_NAME AND PRIOR CR.TABLE_NAME = CR.TABLE_NAME
),
PKS AS(
SELECT CONNECT_BY_ROOT CC.TABLE_NAME AS TABL,
SUBSTR(SYS_CONNECT_BY_PATH(CC.COLUMN_NAME, ', '), 3) AS PK,
LEVEL THISLEVEL, MAX(LEVEL) OVER (PARTITION BY CC.TABLE_NAME) AS LVL
FROM USER_CONS_COLUMNS CC
INNER JOIN USER_CONSTRAINTS C1 ON C1.CONSTRAINT_NAME = CC.CONSTRAINT_NAME AND
C1.CONSTRAINT_TYPE = 'P'
WHERE CC.TABLE_NAME IN
('COUNTRIES','DEPARTMENTS','EMPLOYEES','JOB_HISTORY','JOBS','JOB_GRADES','LOCATIONS','REGIONS
')
AND CONNECT_BY_ISLEAF = 1
CONNECT BY PRIOR CC.COLUMN_NAME < CC.COLUMN_NAME AND PRIOR CC.TABLE_NAME =
CC.TABLE_NAME
) ,
RESULTS AS(
SELECT DISTINCT ALL_T.TABLE_NAME t, NVL(PKS.PK, ' ') p,
NVL(FKS.FK, 'Подчиненных таблиц нет') f, NVL(FKS.COLS, ' ') c,
ROW_NUMBER() OVER (PARTITION BY ALL_T.TABLE_NAME ORDER BY FKS.FK, PKS.PK) rn
FROM USER_TABLES ALL_T
LEFT OUTER JOIN PKS ON (ALL_T.TABLE_NAME = PKS.TABL)
LEFT OUTER JOIN FKS ON (ALL_T.TABLE_NAME = FKS.TABL)
WHERE (FKS.THISLEVEL = FKS.LVL OR FKS.THISLEVEL+FKS.LVL IS NULL) AND (PKS.THISLEVEL = PKS.LVL OR
PKS.THISLEVEL+PKS.LVL IS NULL)
AND ALL_T.TABLE_NAME IN
```

```
('COUNTRIES','DEPARTMENTS','EMPLOYEES','JOB_HISTORY','JOBS','JOB_GRADES','LOCATIONS','REGIONS
')
ORDER BY 1, rn)
SELECT decode(rn, 1, t, ' ') AS "Имя таблицы", decode(rn, 1, p, ' ') AS "Список столбцов ПК",
f AS "Подчиненные таблицы", c AS "Список столбцов FK в подч"
FROM RESULTS;
```

Решение:

```
WITH src AS (SELECT ALL_T.TABLE_NAME SUPP, C2.TABLE_NAME CUST, LAG(C2.TABLE_NAME) OVER
(PARTITION BY ALL_T.TABLE_NAME ORDER BY ALL_T.TABLE_NAME) ONE_MORE FROM
USER_CONSTRAINTS C1 INNER JOIN USER_CONSTRAINTS C2 ON C2.R_CONSTRAINT_NAME =
C1.CONSTRAINT_NAME RIGHT JOIN USER_TABLES ALL_T ON ALL_T.TABLE_NAME = C1.TABLE_NAME),
t AS ( SELECT SUPP, TRIM( LEADING ',' FROM (SYS_CONNECT_BY_PATH(CUST,',')) ) AS LISTING, LEVEL AS l
FROM src START WITH ONE_MORE IS NULL CONNECT BY NOCYCLE PRIOR CUST = ONE_MORE AND
PRIOR SUPP = SUPP),
res AS ( SELECT SUPP,LISTING FROM t M WHERE l = (SELECT MAX(l) FROM t WHERE SUPP=M.SUPP
GROUP BY SUPP)),
p_col AS ( SELECT C3.TABLE_NAME SYS_TABLE, COLUMN_NAME PK_COL, LAG(COLUMN_NAME) OVER
(PARTITION BY C3.TABLE_NAME ORDER BY C3.TABLE_NAME) COL_COL FROM USER_CONSTRAINTS C3
JOIN USER_CONS_COLUMNS C4 ON C3.CONSTRAINT_NAME = C4.CONSTRAINT_NAME WHERE
C3.CONSTRAINT_TYPE='P'),
u_col AS ( SELECT C3.TABLE_NAME SYS_TABLE, COLUMN_NAME U_COL, LAG(COLUMN_NAME) OVER
(PARTITION BY C3.TABLE_NAME ORDER BY C3.TABLE_NAME) UCOL_COL FROM USER_CONSTRAINTS C3
JOIN USER_CONS_COLUMNS C4 ON C3.CONSTRAINT_NAME = C4.CONSTRAINT_NAME WHERE
C3.CONSTRAINT_TYPE='U'),
p_col2 AS ( SELECT SYS_TABLE, TRIM( LEADING ',' FROM (SYS_CONNECT_BY_PATH(PK_COL,',')) )
LISTING_COL, LEVEL LEV FROM p_col START WITH COL_COL IS NULL CONNECT BY NOCYCLE PRIOR
PK_COL = COL_COL AND PRIOR SYS_TABLE = SYS_TABLE),
u_col2 AS ( SELECT
SYS_TABLE, TRIM( LEADING ',' FROM (SYS_CONNECT_BY_PATH(U_COL,',')) ) ULISTING_COL, LEVEL LEV
FROM u_col START WITH UCOL_COL IS NULL CONNECT BY NOCYCLE PRIOR U_COL = UCOL_COL AND
PRIOR SYS_TABLE = SYS_TABLE),
p_col3 AS ( SELECT SYS_TABLE, LISTING_COL FROM p_col2 L WHERE LEV = (SELECT MAX(LEV) FROM
p_col2 WHERE SYS_TABLE = L.SYS_TABLE GROUP BY SYS_TABLE)),
u_col3 AS ( SELECT SYS_TABLE, ULISTING_COL FROM u_col2 L WHERE LEV = (SELECT MAX(LEV) FROM
u_col2 WHERE SYS_TABLE = L.SYS_TABLE GROUP BY SYS_TABLE))
SELECT DISTINCT SUPP AS "Имя таблицы", NVL(LISTING_COL,'Первичного ключа нет') AS "Столбцы
первичного ключа", NVL(ULISTING_COL,'Ограничения уникальности в таблице нет') AS "Столбцы с
огр. уникальности", NVL(LISTING,'Подчиненных таблиц нет')AS "Список подчиненных таблиц"
FROM res r LEFT JOIN
p_col3 p ON r.SUPP = p.SYS_TABLE LEFT JOIN u_col3 u ON r.SUPP = u.SYS_TABLE;
```

3. Задана таблица со столбцами Номер – Number и Сумма – Number. Положительное значение во втором столбце обозначает сумму, которая пришла на счет, а отрицательное значение – корректировка (уменьшение) предыдущих поступлений. Требуется написать запрос, который определит суммы с учетом корректировок.
   Например, для таблицы

Номер	Сумма
1	100
2	300
3	200
4	100

5	-350
6	100
8	100
9	-300
10	800
11	-600

Т.е. строка под номером 5 должна отменить сумму в строке 4, 3 и часть суммы из строки 2 и т.д.

Задачу решить с использованием раздела Model.

Результат должен быть:

Номер	**Сумма**	**Итог**
1	100	100
2	300	150
3	200	0
4	100	0
5	-350	0
6	100	0
8	100	0
9	-300	0
10	800	200
11	-600	0

Примечание: Столбец Номер содержит уникальные значение, но пропуски значений возможны.

```
with new as (select row_number() over (order by nomer desc) r, nomer as n, summ
from ttt1)
select n, ost
from new
model
dimension by (r)
measures(n,summ, cast(null as number(3)) as new_s, cast(null as number(3)) as ost)
rules iterate(200) until (n[iteration_number+1] is null)
    (new_s[iteration_number] = case when summ[iteration_number] +
            nvl(new_s[iteration_number-1],0) > 0 then 0 else summ[iteration_number] +
            nvl(new_s[iteration_number-1],0) end,
    ost[iteration_number] = case when summ[iteration_number] +
            nvl(new_s[iteration_number-1],0) > 0 then  summ[iteration_number] +
            nvl(new_s[iteration_number-1],0) else 0 end
    )
order by 1;
```

Вариант2

```
with t as(
select 1 n, 100 summ from dual
union all
select 2 n, 300 summ from dual
union all
select 3 n, 200 summ from dual
union all
select 4 n, 100 summ from dual
```

```
union all
select 5 n, -350 summ from dual
union all
select 6 n, 100 summ from dual
union all
select 8 n, 100 summ from dual
union all
select 9 n, -300 summ from dual
union all
select 10 n, 800 summ from dual
union all
select 11 n, -600 summ from dual
),
tab1 as (
select rownum rn, n, summ
from(select rownum r, n, summ from t
order by r desc)
),
TAB2 AS (
SELECT rn,n, SUMM S, ost, results from tab1
model
dimension by(rn)
measures(SUMM, cast(0 as number(10)) ost,n, cast(0 as number(10)) results)
rules(
ost[1]=
   CASE WHEN REGEXP_LIKE(SUMM[cv()], '^(\d)+$')
   THEN 0 ELSE SUMM[cv()] END,
results[1] =
   case WHEN REGEXP_LIKE(SUMM[cv()], '^(\d)+$')
   THEN SUMM[cv()] ELSE 0 end,
ost[rn>1]=
   CASE WHEN summ[cv()]+ost[cv()-1]<0
   THEN summ[cv()]+ost[cv()-1] ELSE 0 END,
results[rn>1] =
   CASE WHEN summ[cv()]+ost[cv()-1]>0
   THEN summ[cv()]+ost[cv()-1] ELSE 0 END
))
select "Номер", "Сумма", "Итог" from
(SELECT rn,  n "Номер", S "Сумма", RESULTS "Итог"
FROM TAB2
order by 1 desc);


--БЕЗ MODEL Вариант3
with t as(
select 1 n, 100 summ from dual
union all
select 2 n, 300 summ from dual
union all
select 3 n, 200 summ from dual
```

```
union all
select 4 n, 100 summ from dual
union all
select 5 n, -350 summ from dual
union all
select 6 n, 100 summ from dual
union all
select 7 n, 100 summ from dual
union all
select 8 n, -300 summ from dual
union all
select 10 n, 800 summ from dual
union all
select 11 n, -600 summ from dual
union all
select 13 n, 300 summ from dual
),
tab1 as (
select rownum rn, n, summ from t
),
TAB2(R,S,OST,RESULTS) AS
(
  SELECT rn R, SUMM S,
  CASE WHEN REGEXP_LIKE(SUMM, '^(\d)+$')
    THEN 0 ELSE SUMM
  END OST,
  CASE WHEN REGEXP_LIKE(SUMM, '^(\d)+$')
    THEN SUMM ELSE 0
  END RESULTS
  FROM TAB1
  WHERE RN = (SELECT MAX(RN) FROM TAB1)
  UNION ALL
  SELECT R-1, SUMM,
  CASE WHEN SUMM+OST < 0
    THEN SUMM+OST ELSE 0
  END,
  CASE WHEN SUMM+OST>0
    THEN SUMM+OST ELSE 0
  END
  FROM TAB2 JOIN TAB1 ON TAB2.R-1 = TAB1.RN
  WHERE R is not null
)
SELECT DISTINCT R "Номер", S "Сумма", RESULTS "Итог"
FROM TAB2
ORDER BY 1 ASC;
```

4. Создать запрос для получения информации об успеваемости студентов в виде:

ФИО	Дисциплина	Оценка	Дата	Примечания
Петров	Математика	5	20.1.2008	
	Физика	4	22.1.2008	
	Химия	2	25.1.2008	

	Химия	3	27.1.2008	Пересдача
Усов	Математика	5	12.06.99	
	Экономика	3	15.06.99	
	Менеджмент	2	17.06.99	
	Менеджмент	4	18.06.99	Пересдача
Судаков	Экзамены не сдавал			

В таблице должна быть представлена информация только по результатам сдачи экзаменов по дисциплинам, предусмотренным учебным планом для специальности, на которой учится студент.

ПРИМЕЧАНИЕ: Задача решена на westfold, название некоторых столбцов и таблиц могут отличаться.

Решение: решение с аналитическими

```
/*Нахождение списка студентов, их фамилий, дисциплины, даты экзамена.
Также определяется дата первого экзамена у студента, количество попыток сдачи
экзамена и дата первой попытки.*/
WITH sel AS (SELECT фамилия, номер_студента, название, оценка, дата,
MIN(дата) OVER (PARTITION BY номер_студента) AS минимальная_дата,
MIN(дата) OVER (PARTITION BY номер_студента,номер_дисциплины) AS
дата_первой_попытки,
COUNT(дата) OVER (PARTITION BY номер_студента,номер_дисциплины) AS количество
FROM успеваемость right JOIN студенты USING (номер_студента)
left JOIN дисциплины USING (номер_дисциплины))
/*Получение необходимого вида организации данных*/
SELECT DECODE(дата,минимальная_дата,фамилия, ' ') AS "Фамилия", nvl(название,
'Экзамены не сдавал')  AS "Дисциплина", nvl(to_char(оценка),' ') AS "Оценка",
nvl(to_char(дата),' ') AS "Дата",
/*Если количество пересдач не равно 1 и значение даты в текущей строке не
равно дате первой попытки, следовательно эта сдача экзамена - пересдача.*/
DECODE(количество,1,' ',DECODE(дата,дата_первой_попытки,' ', 'Пересдача')) AS
"Примечание"
FROM sel
ORDER BY номер_студента, дата
```

Решение3:

а) с использованием аналитических функций;

```
with exams as
--экзамены студентов
( select номер_студента s_id, фамилия s_name, название exam, оценка mark, дата e_dt,
--попытка сдачи экзамена
row_number() over (partition by номер_студента, номер_дисциплины order by дата) rn_d,
--номер экзамена для студента
row_number() over (partition by номер_студента order by дата) rn_n
from студенты join успеваемость using(номер_студента)
join дисциплины using(номер_дисциплины)
where номер_дисциплины in
--только экзамены, предусмотренные учебным планом
( select номер_дисциплины
from учебные_планы join группы using(код_специальности)
where номер_группы=студенты.номер_группы )
order by 2, 5)

select decode(rn_n, 1, s_name, '') ФИО,
exam "Дисциплина", mark "Оценка", e_dt "Дата",
decode(rn_d, 1, '', 'Пересдача') "Примечание"
```

```
from exams;

б) без использования аналитических функций.
with exams as
--экзамены студентов
( select st.номер_студента s_id, st.фамилия s_name, ob.название exam, ex.оценка mark, ex.дата
e_dt,
--дата первого экзамена студента
( select min(дата) from успеваемость where номер_студента = st.номер_студента ) name_dt,
--дата первой попытки сдачи экзамена
( select min(дата) from успеваемость where номер_студента = st.номер_студента and
номер_дисциплины = ex.номер_дисциплины) m_dt
from студенты st join успеваемость ex on ex.номер_студента=st.номер_студента
join дисциплины ob on ob.номер_дисциплины=ex.номер_дисциплины
where ex.номер_дисциплины in
--только предусмотренные экзамены
( select номер_дисциплины
from учебные_планы join группы using(код_специальности)
where номер_группы = st.номер_группы )
order by 2, 5)

select
decode(e_dt, name_dt, s_name, '') ФИО, exam "Дисциплина", mark "Оценка", e_dt "Дата",
decode(e_dt, m_dt, '', 'Пересдача') "Примечание"
from exams;
Решение4:
a)
SELECT
CASE WHEN RW = 1 THEN FAM ELSE NULL END "ФАМИЛИЯ",
ДИСЦИПЛИНА,
ОЦЕНКА,
ДАТА,
ПРИМЕЧАНИЯ
FROM (SELECT ФАМИЛИЯ FAM, ROW_NUMBER() OVER (PARTITION BY СТУДЕНТЫ.НОМЕР_СТУДЕНТА
ORDER BY ФАМИЛИЯ) RW,
НАЗВАНИЕ ДИСЦИПЛИНА, ОЦЕНКА, ДАТА,
CASE WHEN ROW_NUMBER() OVER (PARTITION BY СТУДЕНТЫ.ФАМИЛИЯ, ДИСЦИПЛИНЫ.НАЗВАНИЕ
ORDER BY СТУДЕНТЫ."НОМЕР_СТУДЕНТА") = 2 THEN 'ПЕРЕСДАЧА' ELSE TO_CHAR(NULL) END
ПРИМЕЧАНИЯ
FROM СТУДЕНТЫ JOIN (SELECT * FROM УСПЕВАЕМОСТЬ ORDER BY ДАТА) X ON
СТУДЕНТЫ."НОМЕР_СТУДЕНТА" = X."НОМЕР_СТУДЕНТА"
JOIN ДИСЦИПЛИНЫ ON X."НОМЕР_ДИСЦИПЛИНЫ" = ДИСЦИПЛИНЫ."НОМЕР_ДИСЦИПЛИНЫ");
-- в подзапросе соединяю таблицы и вычисляю номера строк, в основном запросе далее
отображаю фамилию только для первых строк



б)
WITH A AS
(SELECT T.НОМЕР_СТУДЕНТА N, T.ДАТА D, T.ОЦЕНКА O, T.НОМЕР_ДИСЦИПЛИНЫ ND, COUNT (*)
CNT1 FROM(
SELECT T.НОМЕР_СТУДЕНТА, T.ДАТА, T.ОЦЕНКА, T.НОМЕР_ДИСЦИПЛИНЫ FROM УСПЕВАЕМОСТЬ T
INNER JOIN УСПЕВАЕМОСТЬ T1 ON T.НОМЕР_СТУДЕНТА = T1.НОМЕР_СТУДЕНТА WHERE T.ДАТА >=
T1.ДАТА) T
```

```
GROUP BY T.НОМЕР_СТУДЕНТА, T.ДАТА, T.ОЦЕНКА, T.НОМЕР_ДИСЦИПЛИНЫ
ORDER BY T.НОМЕР_СТУДЕНТА, T.ДАТА, T.НОМЕР_ДИСЦИПЛИНЫ),
B AS (
SELECT A1.N,A1.D,A1.O,A1.ND,A1.CNT1, COUNT(*) CNT2 FROM (
SELECT A1.N, A1.D, A1.O, A1.ND, A1.CNT1 FROM A A1
INNER JOIN A A2 ON A1.N = A2.N AND A1.ND = A2.ND WHERE A1.D >= A2.D) A1
GROUP BY A1.N, A1.D, A1.O, A1.ND, A1.CNT1
ORDER BY A1.N, A1.D, A1.O)

SELECT
(CASE WHEN B.CNT1 = 1 THEN C.ФАМИЛИЯ ELSE NULL END) "ФАМИЛИЯ",
D.НАЗВАНИЕ ДИСЦИПЛИНА, B.O "ОЦЕНКА", B.D "ДАТА",
(CASE WHEN B.CNT2 = 2 THEN 'ПЕРЕСДАЧА' ELSE NULL END) "ПРИМЕЧАНИЕ"
FROM (
B INNER JOIN СТУДЕНТЫ C ON B.N = C.НОМЕР_СТУДЕНТА) INNER JOIN ДИСЦИПЛИНЫ D ON B.ND =
D.НОМЕР_ДИСЦИПЛИНЫ;
```

Решение5: (Андреева)

Решение с использованием аналитических функций

```
/*Нахождение списка студентов, их фамилий, дисциплины, даты экзамена.
Также определяется дата первого экзамена у студента, количество попыток сдачи экзамена и
дата первой попытки.*/
WITH sel AS (SELECT фамилия, номер_студента, название, оценка, дата,
MIN(дата) OVER (PARTITION BY номер_студента) AS минимальная_дата,
MIN(дата) OVER (PARTITION BY номер_студента,номер_дисциплины) AS дата_первой_попытки,
COUNT(дата) OVER (PARTITION BY номер_студента,номер_дисциплины) AS количество
FROM успеваемость JOIN студенты USING (номер_студента)
JOIN дисциплины USING (номер_дисциплины))
/*Получение необходимого вида организации данных*/
SELECT DECODE(дата,минимальная_дата,фамилия, NULL) AS "Фамилия", название AS
"Дисциплина", оценка AS "Оценка", дата AS "Дата",
/*Если количество пересдач не равно 1 и значение даты в текущей строке не равно дате
первой попытки, следовательно эта сдача экзамена - пересдача.*/
DECODE(количество,1,NULL,DECODE(дата,дата_первой_попытки,NULL, 'Пересдача')) AS
"Примечание"
FROM sel
ORDER BY номер_студента, дата;
```

Решение без использования аналитических функций

```
/*Нахождение списка студентов, их фамилий, дисциплины, даты экзамена.*/
WITH sel_1 AS (SELECT номер_студента, фамилия, название, оценка, дата
FROM успеваемость JOIN студенты USING (номер_студента)
JOIN дисциплины USING (номер_дисциплины)),
/*Определение даты первого экзамена у студента.*/
sel_2 AS (SELECT номер_студента,MIN(дата) AS минимальная_дата
FROM sel_1
GROUP BY номер_студента),
/*Определение количества попыток сдачи экзамена и дата первой попытки.*/
sel_3 AS (SELECT номер_студента, название, COUNT(*) AS количество, MIN(дата) AS
дата_первой_сдачи
FROM sel_1
GROUP BY номер_студента, название)
/*Получение необходимого вида организации данных*/
SELECT DECODE(дата,минимальная_дата,фамилия, NULL) AS "Фамилия", sel_1.название AS
"Дисциплина", оценка AS "Оценка", дата AS "Дата",
/*Если количество пересдач не равно 1 и значение даты в текущей строке не равно дате
```

```
*первой попытки, следовательно эта сдача экзамена - пересдача.*/
DECODE(количество,1,NULL,DECODE(дата,дата_первой_сдачи,NULL, 'Пересдача')) AS "Примечание"
FROM sel_1 JOIN sel_2 ON sel_1.номер_студента=sel_2.номер_студента
JOIN sel_3 ON sel_1.номер_студента=sel_3.номер_студента AND sel_1.название=sel_3.название
ORDER BY sel_1.номер_студента, дата;
```

5. Имеется таблица со столбцами Номер, Строка. Тип данных столбца Номер - Integer, тип данных столбца Строка – Varchar2(10). Первый столбец содержит порядковый номер записи, столбец Строка – символьные строки, состоящие из 0 и 1. Общее количество цифр во всех строках – одинаковое и равно 5. Написать запрос, который выведет номера максимального количества строк и позиции столбца Строка, образующие квадратную матрицу, состоящую только из единиц.

Например, для таблицы:

Номер	Строка
1	00101
2	10011
3	10101

ответ должен быть

Строки	Столбцы
1,3	3,5
2,3	1,5

Решение:

```
CREATE TABLE ZAD4(ID INTEGER CONSTRAINT pk4_pk PRIMARY KEY,
    "String" VARCHAR2(10));

INSERT INTO ZAD4 VALUES(1,'00101');
INSERT INTO ZAD4 VALUES(2,'10011');
INSERT INTO ZAD4 VALUES(3,'10101');

WITH POSTOL AS (SELECT ID, "String",
                       SUBSTR("String", 1, 1) AS A1,
                       SUBSTR("String", 2, 1) AS A2,
                       SUBSTR("String", 3, 1) AS A3,
                       SUBSTR("String", 4, 1) AS A4,
                       SUBSTR("String", 5, 1) AS A5
                      FROM ZAD4 ),

PEREVOROT AS (SELECT TO_CHAR(ID) AS IDD,
TO_CHAR(COLNAME) AS A
                 FROM POSTOL
                  UNPIVOT (ZNACH FOR COLNAME IN(A1 AS 1, A2 AS 2, A3 AS 3, A4 AS 4, A5 AS 5))
                  WHERE ZNACH = 1 ),

PER2(IDD2, IDD, DIGIT) AS
(SELECT IDD AS IDD2, IDD, A AS DIGIT
      FROM PEREVOROT
      UNION ALL
      SELECT PEREVOROT.IDD,    PER2.IDD||','||PEREVOROT.IDD, A
      FROM PER2 INNER JOIN PEREVOROT
    ON PEREVOROT.IDD > PER2.IDD2
    AND PEREVOROT.A = PER2.DIGIT),
```

```
PER3(LEN, IDD, DIGIT, A) AS
 (SELECT LENGTH(IDD) AS LEN, IDD, DIGIT, DIGIT AS A
          FROM PER2
UNION ALL
          SELECT LENGTH(PER2.IDD), PER2.IDD, PER2.DIGIT, PER3.A||','||PER2.DIGIT
          FROM PER3 INNER JOIN PER2
        ON PER2.DIGIT > PER3.DIGIT
        AND PER2.IDD = PER3.IDD ),

MATRS AS (SELECT LEN, IDD, A
                        FROM PER3
                        WHERE LENGTH(A) = LEN )

SELECT IDD AS "Rows", A AS "Columns"
FROM MATRS
WHERE LEN = (SELECT MAX(LEN)
              FROM MATRS)
ORDER BY 1,2;
```

Решение2:

```
with src as /*таблица с исходными данными*/
( select 1 id, '00101' string from dual
union all select 2, '10011' from dual
union all select 3, '10101' from dual ),
/*разнесём поле string на колонки, соответствующие позициям символов*/
t as ( select id, string,
substr(string, 1, 1) a,
substr(string, 2, 1) b,
substr(string, 3, 1) c,
substr(string, 4, 1) d,
substr(string, 5, 1) e
from src ),
/*развернём таблицу на строки для каждой позиции единицы и поля id*/
up as ( select to_char(id) rs, to_char(cl) cs from t
unpivot (col for cl in(a as 1, b as 2, c as 3, d as 4, e as 5))
where col = 1 ),
/*запишем через запятую все возможные id строк, где позиции единицы совпадают*/
r_cl(r, rs, c) as
( select rs r, rs, cs c from up union all
select up.rs r, r_cl.rs||','||up.rs rs, cs c
from r_cl join up on up.rs > r_cl.r and up.cs = r_cl.c ),
/*запишем через запятую все возможные позиции единиц, совпадающие для комплектов строк*/
c_cl(l, rs, c, cs) as
( select length(rs) l, rs, c, c cs from r_cl union all
select length(r_cl.rs) l, r_cl.rs, r_cl.c, c_cl.cs||','||r_cl.c
from c_cl join r_cl on r_cl.c > c_cl.c and r_cl.rs = c_cl.rs ),
/*выделим все квадратные матрицы*/
all_matrix as
( select l, rs, cs from c_cl
where length(cs) = l )

/*выберем матрицы максиммальной размерности*/
select rs "Rows", cs "Columns" from all_matrix
where l = (select max(l) from all_matrix)
```

```
order by rs, cs;
```

БЛИЛЕТ 9

4. Имеется таблица с символьным столбцом. Создать запрос для вывода тех значений, которые содержат в себе палиндромы, и самые длинные выражения, представляющие из себя палиндром.

Например, для таблицы с данными:

Text
Крокодил
Колокол
Станок

Результат должен быть:

Text	Palindrom
Крокодил	око
Колокол	Колок, локол

- with stroki as

```
(
select 'Крокодил' text
from dual
union all
select 'Колокол'
from dual
)
, stroki2 as
(
select text text1, text text2
from stroki
)
, temp as(
select distinct text1, txt, txt txt2
from stroki2
model
partition by (text1)
dimension by (row_number() over (partition by text1 order by text1) id1, 1 as id2)
measures (text2, cast(' ' as varchar2(100)) as txt)
rules iterate(1000) until(iteration_number > length(text2[1,1]) *length(text2[1,1]))
(
--mod(iteration_number+1,length(text2[1,1])),
trunc((iteration_number+1)/length(text2[1,1]))+1
txt[1,iteration_number+1]=
substr(text2[1,1],mod(iteration_number+1,length(text2[1,1])),
trunc((iteration_number+1)/length(text2[1,1]))+2)
))

, temp2 as
(
select text1, txt, txt2, txt3, id1, id2
from temp
model
```

```
partition by (text1,txt)
dimension by (row_number() over (partition by text1,txt order by text1,txt) id1, 1 as id2)
measures (txt2, cast(0 as number(2)) as txt3)
rules iterate(1000) until (iteration_number+1 >=length(txt2[1,1]))
(
txt3[1,iteration_number+1] = —substr(txt2[1,1],iteration_number+1,1) || ' ' ||
substr(txt2[1,1],length(txt2[1,1])-iteration_number,1)
case
when lower(substr(txt2[1,1],iteration_number+1,1)) = lower(substr(txt2[1,1],length(txt2[1,1])-
iteration_number,1)) then 1
else 0
end
)
)
, temp3 as
(
select distinct text1, txt, sum(txt3) over (partition by text1, txt) con
from temp2
)

, temp4 as(


select text1, txt, max(length(txt)) over (partition by text1) mx
from temp3
where con = length(txt))
, temp5 as
(
select text1, txt
from temp4
where length(txt) = mx)

select distinct text1, listagg(txt,',')
within group(order by text1)
over(partition by text1) list
from temp5;
```

Решение2

```
with t as(
select 'Крокодил'text from dual
union all
select 'Колокол'text from dual
union all
select 'Станок'text from dual
union all
select 'Станок и колокол'text from dual
),
--добавим id к словам
t1 as(select rownum num ,text from t),
--разделим все слова на буквы
t2 as(select num,cnt,part from t1
model
partition by(num)
dimension by(cast(1 as number(10)) cnt)
```

```
measures(text,cast(' ' as varchar2(20)) part)
rules iterate(100) until(iteration_number+1>=length(text[1]))(
part[iteration_number+1]=substr(text[1],iteration_number+1,1)
)),
--найдём все подслова из букв в исходном порядке
t3 as(select num,  row_number() over (order by num,level,cnt)
rn,replace(sys_connect_by_path(part,'#'),'#') path from t2
where level!=1
connect by prior cnt+1=cnt and prior num=num),
--найдём все подслова из букв в обратном порядке
t4 as( select  num, row_number() over (order by num,level,cnt)
rn,replace(sys_connect_by_path(part,'#'),'#') path from t2
where level!=1
connect by prior cnt-1=cnt and prior num=num),
--найдем из этих частей слова палиндромы (одинаковые в исходнои и в обратном
порядке)
t5 as(
select distinct t3.num,t3.path p
from t3 join t4 on t3.rn=t4.rn and lower(t3.path)=lower(t4.path)
),
--выберем палиндромы с максимальной длиной
t6 as(
select num,p from t5
where length(p)=
(select distinct max(length(p)) from t5 t
where t.num = t5.num group by num))
select distinct t1.text ,
listagg(t6.p,',') within group (order by t6.p) over (partition by t6.num) Palindrom
from t6 join t1 on t6.num = t1.num;
```

6. Создать таблицу Города, в которой хранятся названия городов и расстояния между ними. Названия городов уникальны. Пример заполнения таблицы:

Город отправ	Город назнач	Расстояние
Москва	Казань	2000
Москва	Тула	200
Казань	Вологда	800
……	……	……

Написать команду SELECT, которая определит все пути и расстояния между двумя городами, имена которых задаются как параметры. Путь выводить в виде списка, например, Москва – Казань – Вологда. Расстояния в прямую и обратную сторону могут различаться.

Решение:

```
CREATE TABLE ZAD14(START_CITY VARCHAR2(20),
       END_CITY VARCHAR2(20),
       DISTANCE NUMBER(6),
     CONSTRAINT se_pk PRIMARY KEY(START_CITY,END_CITY));

INSERT INTO ZAD14 VALUES('Москва','Казань',2000);
INSERT INTO ZAD14 VALUES('Москва','Тула',200);
INSERT INTO ZAD14 VALUES('Казань','Вологда',800);
```

```
INSERT INTO ZAD14 VALUES('Москва','Санкт-Петербург',700);
INSERT INTO ZAD14 VALUES('Санкт-Петербург','Москва',735);
INSERT INTO ZAD14 VALUES('Москва','Иркутск',5200);
INSERT INTO ZAD14 VALUES('Санкт-Петербург','Иркутск',5777);

UNDEFINE CITY1;
UNDEFINE CITY2;
WITH RECURSIVE (END_CITY, WAY, DISTANCE) AS (
SELECT END_CITY, START_CITY || '*' || END_CITY, DISTANCE
FROM ZAD14
WHERE START_CITY = '&&CITY1'
UNION ALL
SELECT CP.END_CITY,
   REC.WAY || '*' || CP.END_CITY,
   CP.DISTANCE + REC.DISTANCE
FROM ZAD14 CP INNER JOIN RECURSIVE REC
      ON (REC.END_CITY = CP.START_CITY ))
CYCLE END_CITY SET cyclemark TO 'X' DEFAULT '-' ,

T1 AS(SELECT ROWNUM RN, '*' || WAY || '*' AS WAY, DISTANCE AS "Расстояние"
         FROM RECURSIVE
         WHERE SUBSTR(WAY,1,INSTR(WAY,'*',1) - 1) = '&CITY1'
         AND SUBSTR(WAY,INSTR(WAY,'*',-1) + 1) = '&CITY2'),

T2 AS(SELECT RN, SUBSTR(WAY, INSTR(WAY,'*',1,LEVEL) + 1, INSTR(WAY,'*',1,LEVEL + 1) -
INSTR(WAY,'*',1,LEVEL) - 1) AS WAY
         FROM T1
         CONNECT BY LEVEL <= LENGTH(WAY) - LENGTH(REPLACE(WAY,'*'))),

T3 AS(SELECT RN, COUNT(*) AS CO
         FROM(SELECT DISTINCT RN, WAY
                  FROM T2
                  WHERE WAY IS NOT NULL)
GROUP BY RN),

T4 AS(SELECT CASE WHEN (LENGTH(WAY) - LENGTH(replace(WAY,'*'))) = CO + 1
   THEN WAY END WAY, "Расстояние"
         FROM T1 INNER JOIN T3 on (T1.RN = T3.RN))

SELECT SUBSTR(REPLACE(WAY,'*','-'), 2, LENGTH(WAY) - 2) AS "Путь", "Расстояние"
FROM T4

            WHERE WAY IS NOT NULL;
```

БИЛЕТ 10

1. Вывести информацию о таблицах схемы в виде:

Имя таблицы	Столбцы первичного ключа	Столбцы с ограничением уникальности	Список подчиненных таблиц со столбцами вторичных ключей
Table 1	Col1, Col2	Col3, Col4	Table2 (Col5, Col6), Table3 (Col7,Col8)

Пример результата:

Имя таблицы	Столбцы первичного	Столбцы с	Список подчиненных таблиц

	ключа	ограничением уникальности	со столбцами вторичных ключей
DEPT3	DEPARTMENT_ID, DEPARTMENT_NAME	-	DEPT2(ID,NAME)
EMPLOYEES	EMPLOYEE_ID	EMAIL	DEPARTMENTS(MANAGER_ID), EMPLOYEES(MANAGER_ID), JOB_HISTORY(EMPLOYEE_ID)
EMPLOYEES	EMPLOYEE_ID	FIRST_NAME, LAST_NAME	DEPARTMENTS(MANAGER_ID), EMPLOYEES(MANAGER_ID), JOB_HISTORY(EMPLOYEE_ID)
...	...	...	...

Решение:

```
WITH src AS (SELECT ALL_T.TABLE_NAME SUPP,
C2.TABLE_NAME CUST,
LAG(C2.TABLE_NAME) OVER (PARTITION BY ALL_T.TABLE_NAME ORDER BY
ALL_T.TABLE_NAME) ONE_MORE
FROM USER_CONSTRAINTS C1 INNER JOIN
USER_CONSTRAINTS C2 ON C2.R_CONSTRAINT_NAME = C1.CONSTRAINT_NAME
RIGHT JOIN
USER_TABLES ALL_T ON ALL_T.TABLE_NAME = C1.TABLE_NAME),
t AS (
SELECT SUPP,
TRIM( LEADING ',' FROM (SYS_CONNECT_BY_PATH(CUST,',')) ) AS LISTING,
LEVEL AS l
FROM src
START WITH ONE_MORE IS NULL
CONNECT BY NOCYCLE PRIOR CUST = ONE_MORE AND PRIOR SUPP = SUPP),
res AS (
SELECT SUPP,LISTING
FROM t M
WHERE l = (SELECT MAX(l)
FROM t
WHERE SUPP=M.SUPP
GROUP BY SUPP)),
p_col AS (
SELECT C3.TABLE_NAME SYS_TABLE,
COLUMN_NAME PK_COL,
LAG(COLUMN_NAME) OVER (PARTITION BY C3.TABLE_NAME ORDER BY C3.TABLE_NAME)
COL_COL
FROM USER_CONSTRAINTS C3 JOIN
USER_CONS_COLUMNS C4 ON C3.CONSTRAINT_NAME = C4.CONSTRAINT_NAME
WHERE C3.CONSTRAINT_TYPE='P'),

u_col AS (
SELECT C3.TABLE_NAME SYS_TABLE,
COLUMN_NAME U_COL,
LAG(COLUMN_NAME) OVER (PARTITION BY C3.TABLE_NAME ORDER BY C3.TABLE_NAME)
UCOL_COL
FROM USER_CONSTRAINTS C3 JOIN
USER_CONS_COLUMNS C4 ON C3.CONSTRAINT_NAME = C4.CONSTRAINT_NAME
WHERE C3.CONSTRAINT_TYPE='U'),
p_col2 AS (
SELECT SYS_TABLE,
TRIM( LEADING ',' FROM (SYS_CONNECT_BY_PATH(PK_COL,',')) ) LISTING_COL,
LEVEL LEV
FROM p_col
START WITH COL_COL IS NULL
CONNECT BY NOCYCLE PRIOR PK_COL = COL_COL AND PRIOR SYS_TABLE = SYS_TABLE),
u_col2 AS (
```

```
SELECT SYS_TABLE,
TRIM( LEADING ',' FROM (SYS_CONNECT_BY_PATH(U_COL,',')) ) ULISTING_COL,
LEVEL LEV
FROM u_col
START WITH UCOL_COL IS NULL
CONNECT BY NOCYCLE PRIOR U_COL = UCOL_COL AND PRIOR SYS_TABLE = SYS_TABLE),
p_col3 AS (
SELECT SYS_TABLE,
LISTING_COL
FROM p_col2 L
WHERE LEV = (SELECT MAX(LEV)
FROM p_col2
WHERE SYS_TABLE = L.SYS_TABLE
GROUP BY SYS_TABLE)),
u_col3 AS (
SELECT SYS_TABLE,
ULISTING_COL
FROM u_col2 L
WHERE LEV = (SELECT MAX(LEV)
FROM u_col2
WHERE SYS_TABLE = L.SYS_TABLE
GROUP BY SYS_TABLE))
SELECT DISTINCT SUPP AS "Имя таблицы",
NVL(LISTING_COL,'Первичного ключа нет') AS "Столбцы первичного ключа",
NVL(ULISTING_COL,'Ограничения уникальности в таблице нет') AS "Столбцы с огр. уникальности",
NVL(LISTING,'Подчиненных таблиц нет')AS "Список подчиненных таблиц"
FROM res r LEFT JOIN
p_col3 p ON r.SUPP = p.SYS_TABLE
LEFT JOIN
u_col3 u ON r.SUPP = u.SYS_TABLE;
```

Решение:

```
with pk as(
select user_cons_columns.table_name tn, user_cons_columns.column_name cn
from user_cons_columns
join user_constraints uc
using(constraint_name)
where uc.constraint_type = 'P'
),
un as (
select user_cons_columns.table_name tn, user_cons_columns.column_name cn
from user_cons_columns
join user_constraints uc
using(constraint_name)
where uc.constraint_type = 'U'
),
tab as(
select ucc.table_name tn, ucc.column_name cn
from user_cons_columns ucc
join user_constraints uc
using(constraint_name)
where uc.constraint_type = 'R'
),
tab2 as(select pk.tn t1, tab.cn n1, tab.tn t2
from tab
join pk
on tab.cn = pk.cn)

select pk.tn as "Имя таблицы",
```

```
listagg(nvl(pk.cn,' '),',') within GROUP(order by pk.cn ) as "Столбцы первичного ключа",
listagg(nvl(un.cn,' '),',') within GROUP(order by un.cn) as "Столбцы с огр уникальности",
listagg(nvl(tab2.n1,' '),',') within GROUP(order by tab2.n1) as "Столбцы вторичных ключе1"
from pk
full outer join un on pk.tn = un.tn
full outer join tab2 on pk.tn = tab2.t1
group by pk.tn;
```

2. Имеется произвольный набор косточек домино. Информация о них представлена в виде символьной строки, состоящей из четного числа цифр от 0 до 6. Цифры разделены запятыми. Цифры, находящиеся на соседних нечетном и четном местах относятся к одной косточке. Создать запрос для определения самых длинных последовательностей, которые можно составить из заданного набора. Результат для каждой последовательности должен быть представлен в виде символьной строки.

Решение:

```
WITH
TAB(STR) AS  (SELECT  '2,5,2,2,4,2,3,1,6,5' FROM DUAL),

TAB2(STR) AS (
SELECT SUBSTR(STR,INSTR(','||STR||',',',',1,2*LEVEL-1),3) FROM TAB
CONNECT BY LEVEL<=(REGEXP_COUNT(STR,',')+1)/2),

TAB3(STR,R) AS ((SELECT STR,ROWNUM  FROM TAB2)
UNION
(SELECT SUBSTR(STR,3,1)||','||SUBSTR(STR,1,1),ROWNUM FROM TAB2)),

TAB4(STR) AS (
SELECT SYS_CONNECT_BY_PATH(STR,'  ')
FROM TAB3
CONNECT BY NOCYCLE PRIOR R<>R and (SUBSTR(STR,1,1)=SUBSTR(PRIOR STR,3,1) AND SUBSTR(PRIOR
STR,1,1)<>SUBSTR(PRIOR STR,3,1) ))

SELECT STR FROM TAB4
WHERE LENGTH(STR)=(SELECT MAX(LENGTH(STR)) FROM TAB4);
```

Решение2:

```
--исходная строка
with t as (select '2,1,2,6,4,2,3,5,6,5' s from dual),
--из строки вытаскиваем по 2 цифры
t2 as (
select substr(s,instr(','||s||',',',',1,level+(level-1)),instr(','||s||',',',',1,level+2)-instr(','||s||',',',',1,level)-1)
s from t
connect by level<=(regexp_count(s,',')+1)/2),
--пронумеровываем строки+строчки в обратном порядке так же пронумерованые
t3 as ((select s,rownum r from t2)
union (select reverse(s),rownum r from t2)),
--соединяем по принципу: номер строки не равен предидущему и первая цифра строки равна
последней
--цифре предидущей строки. Получаем все варианты.
t4 as (
select sys_connect_by_path(s,'     ') s from t3
where (level<=(select count(*) from t3))
connect by nocycle prior r!=r and (substr(s,1,1)=substr(prior s,3,1)))
--выбираем варианты с максимальной длинной
select s from
(select s, max(length(s)) over() n from t4)
```

```
where length(s)=n;
```

Решение 43 в (кож)

```
DEFINE SOURCE_STRING = '0,0,0,1,1,0,2,1,2,3,3,4,6,4,2,4,2,6'
-- выделяем из строки косточки домино
WITH SOURCE AS (
SELECT REGEXP_SUBSTR('&SOURCE_STRING','[^,]+', 1,
LEVEL * 2 - 1) FIRST,
REGEXP_SUBSTR('&SOURCE_STRING','[^,]+', 1,
LEVEL * 2) SECOND,
LEVEL L,
LEVEL
FROM DUAL
CONNECT BY REGEXP_SUBSTR('&SOURCE_STRING','[^,]
+',1,LEVEL * 2) IS NOT
),
SORTED AS (
SELECT CASE WHEN FIRST > SECOND THEN FIRST ELSE SECOND
END AS FIRST,
CASE WHEN FIRST > SECOND THEN SECOND ELSE FIRST
END AS SECOND
FROM SOURCE
),
USE_TA AS (
SELECT FIRST, SECOND, FIRST || ':' || SECOND AS
DOMINO, TO_CHAR(ROWNUM) AS ID
FROM SORTED
),
RESULT(FIRST, SECOND, F, S, DOMINO, ID) AS (
SELECT FIRST, SECOND, '-' AS F, '-' AS S, DOMINO, ','
|| ID || ',' AS ID
FROM USE_TA
UNION ALL
SELECT U.FIRST AS FIRST,
U.SECOND AS SECOND,
R.FIRST AS F,
R.SECOND AS S,
R.DOMINO || ' + ' || U.DOMINO AS DOMINO,
R.ID || U.ID || ',' AS ID
FROM USE_TA U INNER JOIN
RESULT R ON (
(((U.FIRST = R.FIRST OR U.SECOND =
R.FIRST) AND
((R.FIRST != R.F AND
R.SECOND = R.S) OR (R.FIRST != R.S AND R.SECOND = R.F) OR (R.F = '-')
OR (R.F = R.FIRST AND R.FIRST = R.SECOND AND R.SECOND = R.S)))
OR
((U.FIRST = R.SECOND OR U.SECOND =
R.SECOND) AND
((R.SECOND != R.F AND
R.FIRST = R.S) OR (R.SECOND != R.S AND R.FIRST = R.F) OR (R.F = '-')
OR (R.F = R.FIRST AND R.FIRST = R.SECOND AND R.SECOND = R.S))))
AND
(REGEXP_INSTR(R.ID, ',' || U.ID ||
```

```
',') = 0)
)
)
SELECT DISTINCT DOMINO
FROM RESULT
WHERE LENGTH(DOMINO) = (SELECT MAX(LENGTH(DOMINO)) FROM RESULT);
```

3. Имеется таблица с двумя столбцами – дочерняя вершина и родительская вершина. Определить наборы вершин, образующих связанные множества. Например, для таблицы:

Дочерняя вершина	Родительская вершина
1	2
2	4
4	5
4	3
7	6

результат должен быть

Связанные множества
1,2,3,4,5
6,7

Решение:

```
Create Table Tab1(Child_Number Number(7,0),Parent_Number Number(7,0));
Insert Into Tab1 Values(1,2);
Insert Into Tab1 Values(2,4);
Insert Into Tab1 Values(4,5);
Insert Into Tab1 Values(4,3);
Insert Into Tab1 Values(6,7);
With
Tmp As
(Select Ch, Ch||Sys_Connect_By_Path(Parent_Number, ',') Str
from (select * from (select connect_by_root(Child_Number) Ch, Parent_Number
From Tab1
Connect By Prior Parent_Number = Child_Number
Order By 1)
Where Ch Not In
(Select Parent_Number From (Select Connect_By_Root(Child_Number) Ch, Parent_Number
From Tab1
Connect By Prior Parent_Number = Child_Number
Order By 1)))
Connect By Prior Ch = Ch
And Prior Parent_Number<Parent_Number)
Select Str "Связанные множества" From Tmp
Where (Ch, Length(Str)) In (Select Ch, Max(Length(Str)) From Tmp Group By Ch)
```

Решение (кож) 50 в:

```
WITH TAB AS(SELECT 1 CHILD_N,2 PARENT_N FROM DUAL
UNION ALL
SELECT 2,4 FROM DUAL
UNION ALL
SELECT 4,5 FROM DUAL
UNION ALL
SELECT 4,3 FROM DUAL
```

```
UNION ALL
SELECT 6,7 FROM DUAL
UNION ALL
SELECT 6,10 FROM DUAL
UNION ALL
SELECT 9,8 FROM DUAL
UNION ALL
SELECT 10,9 FROM DUAL
UNION ALL
SELECT 5,8 FROM DUAL
UNION ALL
SELECT 15,18 FROM DUAL),
TAB1 AS(SELECT CHILD_N CN, PARENT_N PN
FROM TAB
UNION ALL
SELECT PARENT_N,
FROM TAB
UNION ALL
SELECT PARENT_N, CHILD_N
FROM TAB)-- select * from tab1;
,
TAB2 AS(SELECT DISTINCT CONNECT_BY_ROOT(CN) ROOT, CN, PN FROM TAB1
WHERE CN IS NOT
START WITH PN IS
CONNECT BY NOCYCLE PN = PRIOR CN)-- select * from tab2 order by 1,2,3;
SELECT DISTINCT LISTAGG(replace(CN,',','.'),',') WITHIN GROUP (ORDER BY CN) OVER (PARTITION BY
ROOT) "Связанные множества"
FROM TAB2;
```

4. Для произвольной строки, состоящей из чисел, разделенных указанным разделителем, получить строку, отображающую эти числа в обратном порядке. Например, для исходной строки:

   0|0|1.45|2|1|2|10|22|34|15|0|-105|66|73

   должна быть получена строка:

   73|66|-105|0|15|34|22|10|2|1|2|1.45|0|0.

   Задачу решить без использования иерархических запросов.

   ЭКСКЛЮЗИВНОЕ РЕШЕНИЕ ЕКАТЕРИНЫ:

```
DEFINE STR='0|0|1.45|2|1|2|10|2.2|34|15|0|-105|66|73'
SELECT '&STR' "Исходная строка",
rtrim(lTRIM(RS,'|'),'|') "Обратный порядок"
FROM DUAL
MODEL
DIMENSION BY (0 i)
MEASURES (CAST('' AS VARCHAR2(1000))RS)
RULES ITERATE (1000) UNTIL (ITERATION_NUMBER=regexp_count('&str','|')+1)
(RS[0]=REGEXP_SUBSTR('&str', '[^|]+',1,ITERATION_NUMBER+1)||'|'||RS[0]);
```

Решение3

```
Create Table Tab1(Child_Number Number(7,0),Parent_Number Number(7,0));
Insert Into Tab1 Values(1,2);
```

```
Insert Into Tab1 Values(2,4);
Insert Into Tab1 Values(4,5);
Insert Into Tab1 Values(4,3);
Insert Into Tab1 Values(6,7);

With
Tmp As
(Select Ch, Ch||Sys_Connect_By_Path(Parent_Number, ',') Str
from (select * from (select connect_by_root(Child_Number) Ch, Parent_Number
From Tab1
Connect By Prior Parent_Number = Child_Number
Order By 1)
Where Ch Not In
(Select Parent_Number From (Select Connect_By_Root(Child_Number) Ch, Parent_Number
From Tab1
Connect By Prior Parent_Number = Child_Number
Order By 1)))
Connect By Prior Ch = Ch
And Prior Parent_Number<Parent_Number)
Select Str "Связанные множества" From Tmp
Where (Ch, Length(Str)) In (Select Ch, Max(Length(Str)) From Tmp Group By Ch);
```

5. Для произвольного целого числа определить числа, полученные перестановками цифр в числе и имеющие максимальные суммы абсолютных разностей между соседними цифрами. Например, для числа 1239 результат должен быть:

   3192
   2913

   Суммы абсолютных разностей равны:

   |3-1| + |1-9| + |9-2| = 17
   |2-9| + |9-1| + |1-3| = 17

   Убедиться в работоспособности при 5, 10, 15 и 20 цифрах в числе.
   НАШИ РЕШЕНИЯ КАТЯ -1 НАСТЯ -2
   РЕШЕНИЕ 1

```
DEFINE NUMB='12390';

WITH DAF
AS(SELECT  NNUM "Число",SUM "Результат"
      FROM (SELECT DISTINCT REPLACE (SYS_CONNECT_BY_PATH(NUM,'/'),'/')NNUM--Шаг 2)перестановка без повтора
      FROM (SELECT SUBSTR('&&NUMB',LEVEL,1)NUM,ROWNUM R--Шаг 1)определение кол-а цифр
         FROM DUAL CONNECT BY LEVEL<=LENGTH('&NUMB'))
      WHERE CONNECT_BY_ISLEAF=1
      CONNECT BY NOCYCLE PRIOR R<>R)
  MODEL--Шаг 3)расчет
  PARTITION BY (NNUM)
  DIMENSION BY (0 i)
  MEASURES (0 DAF,0 SUM)
           RULES (DAF[FOR i FROM 1 TO LENGTH('&NUMB')-1 INCREMENT 1]=ABS(SUBSTR(CV(NNUM),CV(i),1)-SUBSTR(CV(NNUM),CV(i)+1,1)),
  SUM[FOR i FROM 1 TO LENGTH('&NUMB')-1 INCREMENT 1]=SUM[CV(i)-1]+DAF[CV(i)]))
```

```
 SELECT DECODE(ROWNUM,1,'&NUMB',' ')"Исходное число",--Шаг 4)оформленный вывод результата
  DECODE(ROWNUM,1,TO_CHAR(LENGTH('&NUMB')),' ') "Кол-во цифр в числе",
  "Число","Результат"
   FROM DAF
 WHERE "Результат"=(SELECT MAX("Результат")
            FROM DAF);
```

РЕШЕНИЕ 2

```
WITH n as (SELECT '12345' NUMB from dual),
   gr as (select regexp_substr(numb,'\d',1,level) as numbs, level as l
         from n
         connect by level<=length(numb)),
   seq as (select replace(sys_connect_by_path(numbs, ' '),' ') nums
         from gr
         WHERE CONNECT_BY_ISLEAF=1
         connect by nocycle prior l != l),
    sums as (select nums as "ЧИСЛО",(SELECT SUM(ABS(TO_NUMBER(SUBSTR(NUMS, LEVEL,1))-
TO_NUMBER(SUBSTR(NUMS,LEVEL+1,1))))
                  FROM DUAL
                  CONNECT BY LEVEL <= LENGTH(NUMS) - 1) as sumsus
   from seq)
select "ЧИСЛО" from sums
where sumsus = (select max(sumsus) from sums);
```

БИЛЕТ 11

1. Используя словарь данных, получить информацию об ограничениях CHECK схемы:
   В списках имена столбцов вывести через запятую. Имя таблицы не должно повторяться.
   Задачу решить без использования функций Listagg и Wm_concat.
   Пример представления результатов:

SQL | All Rows Fetched: 8 in 1,748 seconds

	Имя таблицы	Имя ограничения	Столбцы,входящие в ограничение	Ограничение CHECK
1	EMPLOYEES	EMP_SALARY_MIN	SALARY	salary > 0
2	JOB_HISTORY	JHIST_DATE_INTERVAL	END_DATE,START_DATE	end_date > start_date
3	TASK_13	COLUMN12345	COLUMN_1,COLUMN_2,COLUMN_3,COLUMN_4,COLUMN_5	column_1>ALL (column_2, column_3, column_4, column_5)
4	TRIP	CON_PER_BEG	PER_BEG	per_beg=TRUNC(per_beg)
5		CON_PER_END	PER_END	per_end=TRUNC(per_end)
6	ПРЕПАРАТ	ПРЕПАРАТ_CHK1	АББРЕВ_ФОРМЫ,ТИП	Аббрев_формы!=Тип
7	ПРЕПОДАВАТЕЛЬ	SYS_C00201812	КАФЕДРА	КАФЕДРА Like 'Кафедра%'
8	УСПЕВАЕМОСТЬ	SYS_C00201820	ОЦЕНКА	ОЦЕНКА between 1 and 5

Решение:

```
WITH T AS (
SELECT UC.TABLE_NAME TABLE_NAME, UC.CONSTRAINT_NAME CONSTRAINT_NAME,
UCC.COLUMN_NAME CN, UC.SEARCH_CONDITION SC,
ROW_NUMBER() OVER (PARTITION BY UC.TABLE_NAME, UC.CONSTRAINT_NAME ORDER BY
UCC.COLUMN_NAME) RN,
ROW_NUMBER() OVER (PARTITION BY UC.TABLE_NAME ORDER BY UCC.COLUMN_NAME) RN_T
FROM USER_CONSTRAINTS UC LEFT JOIN
USER_CONS_COLUMNS UCC ON UC.CONSTRAINT_NAME = UCC.CONSTRAINT_NAME
WHERE UC.CONSTRAINT_TYPE='C')
SELECT
CASE WHEN RN_T=1 THEN TABLE_NAME
ELSE ' ' END TABLE_NAME,
CONSTRAINT_NAME, LTRIM(SYS_CONNECT_BY_PATH(CN, ','), ',') CN, SC
FROM T
WHERE CONNECT_BY_ISLEAF=1
START WITH RN = 1
```

```
CONNECT BY PRIOR TABLE_NAME = TABLE_NAME AND PRIOR CONSTRAINT_NAME =
CONSTRAINT_NAME AND PRIOR RN + 1 = RN;)
```

Решение2:

```
WITH TNAME AS
     (SELECT TABLE_NAME
      FROM USER_CONSTRAINTS),
   CHK_COL AS
     (SELECT R1.TABLE_NAME, R2.COLUMN_NAME, LAG(R2.COLUMN_NAME) OVER (PARTITION BY
R1.TABLE_NAME, R2.CONSTRAINT_NAME ORDER BY R2.COLUMN_NAME) AS LAG_COL,
           R1.CONSTRAINT_NAME, R1.SEARCH_CONDITION_vc
      FROM USER_CONSTRAINTS R1 JOIN USER_CONS_COLUMNS R2 ON
(R1.CONSTRAINT_NAME=R2.CONSTRAINT_NAME AND R1.TABLE_NAME=R2.TABLE_NAME)
      WHERE CONSTRAINT_TYPE = 'C'),

   CHK_COL_LISTAGG AS (
     SELECT TABLE_NAME, TRIM(LEADING ',' FROM SYS_CONNECT_BY_PATH(COLUMN_NAME, ','))
COLUMN_NAME, LEVEL LEV,  CONSTRAINT_NAME, SEARCH_CONDITION_vc
     FROM CHK_COL
     START WITH LAG_COL IS NULL
     CONNECT BY NOCYCLE PRIOR COLUMN_NAME = LAG_COL AND PRIOR TABLE_NAME=TABLE_NAME
),

   CHK_COL_RES AS
     (SELECT TABLE_NAME, COLUMN_NAME, CONSTRAINT_NAME, SEARCH_CONDITION_vc
      FROM CHK_COL_LISTAGG R3
      WHERE LEV = (SELECT MAX(LEV)
                   FROM CHK_COL_LISTAGG R4
                   WHERE R3.TABLE_NAME=R4.TABLE_NAME
                   GROUP BY TABLE_NAME))

SELECT distinct TABLE_NAME,CONSTRAINT_NAME, COLUMN_NAME, SEARCH_CONDITION_vc
      FROM CHK_COL_RES;
```

2. Одной командой сотрудников подразделения, в котором они работают.
   Сведения о сотрудниках, для которых неизвестно подразделение компании, к которому они приписаны выводить не нужно.
   В результат вывести:
   1.Идентификатор подразделения компании, к которому приписан сотрудник.
   2. Фамилию сотрудника.
   3.Оклад, установленный сотруднику.
   В команде SELECT запрещается использовать:
   - Фразы WITH, GROUP BY,HAVING, ORDER BY, CONNECT BY, START WITH,
   - Условия IN, =ANY, =SOME, NOT IN, <> ALL, EXISTS, NOT EXISTS,
   - Подзапросы (subqueries), в том числе подзапросы во фразе FROM,
   - Иерархические запросы (hierarchical queries),
   - Агрегатные функции (aggregate functions) – MIN,MAX, SUM,COUNT,AVG и др.
   - Аналитические функции (analytic functions)

Решение: (ОЮ) 52в

```
SELECT department_id, last_name, salary   -- Все сотрудники
FROM Employees
WHERE department_id IS NOT NULL          -- За исключением тех, котороые не приписаны ни к
одному отделу

MINUS                                    -- Вычитаем из всех сотрудиков тех, которые имеют зарплату большую,
```

чем кто-то из коллег

```
SELECT E.department_id, E.last_name, E.salary
FROM Employees E INNER JOIN Employees D ON E.department_id = D.department_id    -- Делаем CROSS JOIN зарплат сотрудников внутри отделов
WHERE E.salary > D.salary              -- И исключаем тех, которые имеют зарплату больше чем один из коллег
```

Решение2: (Андреева)

```
select e1.department_id, e1.last_name, e1.salary
from employees e1 left join employees e2 on e1.department_id=e2.department_id and e1.salary>e2.salary
where e2.salary is null and e1.department_id is not null;
```

Алгоритм:

Соединим таблицу employees (emp) с ней же (emp2) по условию **(emp2.department_id = emp.department_id) AND (emp.salary > emp2.salary),** гарантирующему наличие сотрудника с меньшей зарплатой в том же отделе. Таким образом, для каждого отдела найдём сотрудников с не минимальной зарплатой по отделу.

Теперь соединим результат с таблицей **employees (emp3)** ещё раз при помощи правого внешнего соединения (RIGHT OUTER JOIN) по условию **(emp.employee_id = emp3.employee_id).** Сотрудникам с не минимальными зарплатами в своих отделах будет соответствовать **not null** значение **emp.employee_id**, так как они попали в результат предыдущего соединения. В итоге, остаётся отобрать сотрудников с минимальными зарплатами и **null**-значением **emp.employee_id**, а также исключить сотрудников, не приписанных к какому-либо отделу при помощи раздела WHERE.

Решение:

```
SELECT emp3.department_id, emp3.last_name, emp3.salary
FROM hr.employees emp
JOIN hr.employees emp2
ON (emp2.department_id = emp.department_id) AND (emp.salary > emp2.salary)
RIGHT OUTER JOIN hr.employees emp3
ON (emp.employee_id = emp3.employee_id)
WHERE emp.employee_id is null AND emp3.department_id is not null;
```

Решение2:

```
Select E2.Department_Id, E2.Last_Name, E2.Salary
From Employees E0 Join Employees E1 On (E0.Department_Id=E1.Department_Id And E0.Salary>E1.Salary) ----отсекаем минимальные зарплаты
---связываем каждого с каждым с разными зарплатами
Right Outer Join Employees E2 On (E0.Employee_Id=E2.Employee_Id) -- свяжем соед.ROJ с полноценной таблицей
Where E0.Salary Is Null
----и у тех у кого E0 заполнится Nullми (т.е. у тех у кого минимальные зарплаты) выведем
Order By E2.Department_Id;
```

Решение3 (Кож) 39в:

```
SELECT department_id, last_name, salary
FROM employees
WHERE department_id IS NOT
--Все сотрудники, состоящие в каких-либо отделах
MINUS
-- Вычитаем из всех сотрудников тех, чей оклад больше чем чей-то еще в их отделе
SELECT DISTINCT e.department_id, e.last_name, e.salary
FROM employees e
```

```
INNER JOIN employees emp
ON e.department_id = emp.department_id
WHERE e.salary > emp.salary;
-- Сотрудники, чей оклад больше чем чей-то еще в их отделе
```

Решение: (ОЮ) 22в

```
SELECT department_id, last_name, salary
FROM employees
WHERE department_id IS NOT NULL -- Сотрудники, за исключением тех, которые не состоят ни в одном отделе
MINUS -- Вычитаем из всех сотрудиков тех, которые имеют зарплату больше, чем кто-то из коллег
SELECT e.department_id, e.last_name, e.salary
FROM employees e
INNER JOIN employees d
ON e.department_id = d.department_id
WHERE e.salary > d.salary; -- Те сотрудники, кто имеет зарплату больше, чем кто-то из коллег
```

3. Написать запрос, который все пары прямых скобок в строке, внутри которых имеется две или более пары прямых скобок, заменит на фигурные скобки. Например, для строки

[[[98+77]-9]-1] => [[175-9]-[1]]=>165

результат должен быть

{[[98+77]-9]-1} =>{[175-9]-[1]}=>165

```
with def as
(
select '[[[98+77]-9]-1] => [[175-9]-[1]]=>[[[165]yu]t]' str
from dual
)
select
str,regexp_replace(str,'\[(([^][]*\[[^][]*\[[^][]*\][^][]*\][^][]*)|([^][]*\[[^][]*\][^][]*\[[^][]*\][^][]*))\]','{\1}')
hueta
from def;
```

решение при помощи model:

```
with def as
(
select '[[[98+77]-9]-1] => [[175-9]-[1]]=>[[[165]yu]t]' str,rownum rn
from dual
)
select str
from def
model
dimension by(rn)
measures(regexp_replace(str,'\[(([^][]*\[[^][]*\[[^][]*\][^][]*\][^][]*)|([^][]*\[[^][]*\][^][]*\[[^][]*\][^][]*))\]','{\1}') str)
rules();
```

4. В написанном выражении ((((1?2)?3)?4)?5)?6 вместо каждого знака ? вставить знак одной из 4 арифметических операций +,-,*,/ так, чтобы результат вычислений равнялся 35 (при делении дробная часть в частном отбрасывается). Найти все решения

К Решение:

```
WITH T(N) AS (SELECT 2 FROM DUAL
UNION ALL
```

```
SELECT N+1
FROM T
WHERE N <6),

SIGNS(SN) AS (
SELECT '+' FROM DUAL
UNION
SELECT '-' FROM DUAL
UNION
SELECT '/' FROM DUAL
UNION
SELECT '*' FROM DUAL),

SN_FG(FIGURE) AS (
SELECT SN||N  FROM SIGNS CROSS JOIN T),

EXPRESSION(EXP,ROWN) AS (
SELECT '((((1'||T1.FIGURE||')'||T2.FIGURE||')'||T3.FIGURE||')'||T4.FIGURE|| ')'||T5.FIGURE ,
ROWNUM
FROM SN_FG T1 CROSS JOIN SN_FG T2 CROSS JOIN SN_FG T3 CROSS JOIN SN_FG T4 CROSS JOIN SN_FG
T5
WHERE SUBSTR(T1.FIGURE,2,1)=2
AND SUBSTR(T2.FIGURE,2,1)=3
AND SUBSTR(T3.FIGURE,2,1)=4
AND SUBSTR(T4.FIGURE,2,1)=5
AND SUBSTR(T5.FIGURE,2,1)=6),

SUMM(N_ROW,POS_F,EXP_S,RES,CNT) AS(
SELECT 1,1,EXP,1,1 FROM (SELECT EXP FROM EXPRESSION WHERE ROWN=1)
UNION ALL
SELECT
CASE POS_F WHEN 5 THEN N_ROW+1
ELSE N_ROW END,
CASE POS_F WHEN 6 THEN 1
ELSE POS_F+1 END,
(SELECT EXP FROM EXPRESSION WHERE ROWN=N_ROW),
CASE
WHEN SUBSTR(EXP_S,3+3*POS_F,1)='+' THEN RES+SUBSTR(EXP_S,3+3*POS_F+1,1)
WHEN SUBSTR(EXP_S,3+3*POS_F,1)='-' THEN RES-SUBSTR(EXP_S,3+3*POS_F+1,1)
WHEN SUBSTR(EXP_S,3+3*POS_F,1)='*' THEN RES*SUBSTR(EXP_S,3+3*POS_F+1,1)
WHEN SUBSTR(EXP_S,3+3*POS_F,1)='/' THEN TRUNC(RES/SUBSTR(EXP_S,3+3*POS_F+1,1),0)
ELSE 1 END, CNT+1
FROM  SUMM
WHERE CNT<1024*6
)
SELECT EXP_S EXPRESSION, RES RESULT FROM SUMM
WHERE RES = 35 and POS_F = 6;
```

Алгоритм:

В T мы создаем таблицу из 5 цифр (от 2 до 6 – от двух, потому что я явно задаю 1 в последующей таблице).

В SIGNS создаем таблицу из четырех знаков, которые используются в нашем выражении.

В SN_FG я соединяю по декартовому произведению таблицу с цифрами и знаками, чтобы получить всевозможные комбинации цифры и знака.

В EXPRESSION мы соединяем предыдущую таблицу саму с собой 4 раза и таким образом получаем

всевозможные выражения, которые могут быть получены подстановкой каждого знака вместо ? в исходном выражении, и пронумеровываем каждое такое выражение.
В SUMM происходит вычисление каждого выражения, по следующему принципу:

Сначала выбирается первая строчка с результатом 1, потому мы находимся на единичке. Затем POS_F определяет, в какой скобочке мы находимся (берется значение предыдущей строки) или на каком знаке операции (если судить по предыдущему значению, а если смотреть по значению для каждой строки, то это просто на какой циферке находимся), при этом, когда POS_F = 6, в RES новой строчки записывается единичка, потому что мы выходим за грани скобочек. Вычисления продолжаются до тех пор, пока CNT не будет равен 1024*6 (потому что 4 знака находится на 5 позициях, то есть 45=1024 и умножить на 6, потому что у нас 6 цифр для вычисления).
В основном запросе мы выбираем те строки, в которых POS_F = 6, то есть строка, содержащая ответ, и результат равен 35.
Пример вычисления:
WHEN SUBSTR(EXP_S,3+3*POS_F,1)='*' THEN RES*SUBSTR(EXP_S,3+3*POS_F+1,1)
Если POS_F = 1, то мы находимся в первой внутренней скобочке на первом знаке(6 позиция), то есть в (1*2) и тогда результат умножается на следующий символ после знака операции, то есть на 2 итд.
Решение2:

```
--создаем таблицу из 6 цифр
with t as (select rownum n from(
select level from dual
connect by level <7)),
--создаем таблицу из 4х арифметических знаков
t2 as (
select '+' z from dual
union
select '-' from dual
union
select '*' from dual
union
select '/' from dual),
--создаем таблицу из набора цифры и знака
t3 as (
select z||n s from t2,t),
--соединяем все наборы цифра и знака получая все возможные комбинации
t4 as (
select '((((1'||t32.s||')'||t33.s||')'||t34.s||')'||t35.s|| ')'||t36.s s, rownum r
from t3 t32,t3 t33,t3 t34,t3 t35, t3 t36
where substr(t32.s,2,1)=2
and substr(t33.s,2,1)=3
and substr(t34.s,2,1)=4
and substr(t35.s,2,1)=5
and substr(t36.s,2,1)=6),
--используем иерархический запрос для подсчета строки
t5(pos,crpos,str,res,coun) as(
select 1,1,s,1,1 from (select s from t4 where r=1)
union all
select
--pos флаг определяющий с какой строкой работаем
case crpos when 5 then pos+1
else pos end,
--crpos флаг определяющий в какой скобочке находимся
case crpos when 6 then 1
```

```
else crpos+1 end,
--исходная строка с которой осуществляем действие
(select s from t4 where r=pos),
--подсчет элементов
case
when crpos=6 then 1
when substr(str,3+3*crpos,1)='+' then res+substr(str,3+3*crpos+1,1)
when substr(str,3+3*crpos,1)='-' then res-substr(str,3+3*crpos+1,1)
when substr(str,3+3*crpos,1)='*' then res*substr(str,3+3*crpos+1,1)
when substr(str,3+3*crpos,1)='/' then trunc(res/substr(str,3+3*crpos+1,1),0)
else 1 end res,
--счетчик
coun+1
from t5
--всего 1024 строки (4^5  4 знака на 5 позициях)
--6 раз обходим каждую строчку
where coun<1024*6
)
--выводим результат
select str, res from t5
--нужна строка с результатом 35
where res=35 and crpos=6;
```

Решение 43 в (кож)

```
WITH OPERATION AS ( SELECT '+' AS OPER FROM DUAL
UNION ALL
SELECT '-' FROM DUAL
UNION ALL
SELECT '*' FROM DUAL
UNION ALL
SELECT '/' FROM DUAL
),
EXPRESSION (E_LINE, EXPR, VAL) AS (SELECT '(((((1' AS E_LINE, 1
AS EXPR, 1 AS VAL FROM DUAL
UNION ALL
SELECT E_LINE || OP.OPER ||
(EXPR + 1) || ')' AS E_LINE,
EXPR + 1 AS EXPR,
CASE
WHEN OP.OPER = '+'
THEN VAL + (EXPR + 1)
WHEN OP.OPER = '-'
THEN VAL - (EXPR + 1)
WHEN OP.OPER = '*'
THEN VAL * (EXPR + 1)
WHEN OP.OPER = '/'
THEN TRUNC(VAL / (EXPR
+ 1),0)
END AS VAL
FROM EXPRESSION
JOIN OPERATION OP ON
EXPR < 7
--WHERE EXPR < 9
)
```

```
SELECT SUBSTR(E_LINE,2,LENGTH(E_LINE)-2) || ' = ' || VAL AS "RESULT"
FROM EXPRESSION
WHERE EXPR = 6 AND VAL = 35;
```

5. Создать запрос, который позволит оставить только одно из повторяющихся слов в тексте, идущих друг за другом и разделенных пробелами. Количество повторяющихся слов – произвольное.

РЕШЕНИЕ 1

```
 UNDEFINE str
--объявляем переменную
DEFINE str = 'qq q qwe qwe qwe qwe dd dd ff ff ws dd ff ds'
--добавляем в начале и конце пробелы, также дублируем уже имеющиеся
with t as (select ' '||regexp_replace('&str', ' ', '  ')||' ' a from dual)
--с помощью regexp_replace удаляем повторы, регулярка удаляет одно из повторяющихся
--слов, слово – это набор символов ограниченный пробелами с двух сторон
select regexp_replace((trim(regexp_replace(a,'([[:space:]]{1}[[:alnum:]]+[[:space:]]{1})\1{1,}','\1',1))), '  ',
' ') n from t;
--в конце убираем лишнии пробелы по краям(trim) и в самом тексте regexp_replace
```

РЕШЕНИЕ 2

```
DEFINE SOURCE_STRING = 'мама мыла мыла мыла раму раму раму раму раму раму раму привет
мама Мама Name name';
WITH SOURCE AS (
   SELECT * FROM (
      SELECT TO_CHAR(REGEXP_SUBSTR(REGEXP_REPLACE('&SOURCE_STRING', '[\ ]+',' '),'(([^ ]+)([\
]+(\2))+)|([^ ]?+)', 1, LEVEL,'i')) WORD, LEVEL L
      FROM DUAL
      CONNECT BY REGEXP_SUBSTR(REGEXP_REPLACE('&SOURCE_STRING', '[\ ]+',' '),'[^ ]+',1,LEVEL) IS
NOT NULL)
      WHERE WORD IS NOT NULL),
RES AS ( SELECT TO_CHAR(REGEXP_SUBSTR(WORD, '[^ ]+', 1, 1)) AS WORD, L
       FROM SOURCE)
SELECT LISTAGG(WORD, ' ') WITHIN GROUP (ORDER BY L) "Результат"
FROM RES;
```

БИЛЕТ 12

4. Проверить наличие циклов в таблице подчиненностей. Вывести циклические зависимости в строчку в виде Номер1.Имя1->Номер2.Имя2->...Номер1.Имя1, начиная с первого по алфавиту имени.

Например, для таблицы подчинённостей:

Номер	Имя	Номер_начальника
1	Алексей	2
2	Пётр	3
3	Павел	4
4	Иван	2
5	Кристина	3
6	Андрей	5

Результат должен быть:

CYCLE
4.Иван->3.Павел->2.Пётр->4.Иван

Решение:

```
WITH POD AS(
SELECT '1' EI, 'Алексей' FN, '2' NN
FROM DUAL
UNION ALL
SELECT '2' EI, 'Петр' FN, '3'NN
FROM DUAL
UNION ALL
SELECT '3' EI, 'Павел' FN, '4' NN
FROM DUAL
UNION ALL
SELECT '4' EI, 'Иван' FN, '2' NN
FROM DUAL
UNION ALL
SELECT '5' EI, 'Кристина' FN, '3' NN
FROM DUAL
UNION ALL
SELECT '6' EI, 'Андрей' FN, '5' NN
FROM DUAL),
RES AS (SELECT LTRIM(SYS_CONNECT_BY_PATH(EI||'.'||FN, '->')||'-
>'||connect_by_root(EI||'.'||FN),'->') ANS
FROM POD
WHERE connect_by_root(ei) = NN
CONNECT BY NOCYCLE PRIOR NN = ei
ORDER BY connect_by_root(FN))
SELECT ANS FROM RES
WHERE ROWNUM =1;
```

Решение2:

```
WITH temp_table AS (
  SELECT 1 empno, 'Алексей' ename, 2 mgr
  FROM DUAL
  UNION ALL
  SELECT 2, 'Пётр', 3
  FROM DUAL
  UNION ALL
  SELECT 3, 'Павел', 4
  FROM DUAL
  UNION ALL
  SELECT 4, 'Иван', 2
  FROM DUAL
  UNION ALL
  SELECT 5, 'Кристина', 3
  FROM DUAL
  UNION ALL
  SELECT 6, 'Андрей', 5
  FROM DUAL),
temp AS
(SELECT e.*, m.ename mname
FROM temp_table e
JOIN temp_table m ON e.mgr = m.empno)

SELECT substr(sys_connect_by_path(mgr || '.' || mname, '->') || '->' || empno || '.' || ename, 3) cycle
FROM temp
WHERE CONNECT_BY_ROOT(mgr) = empno
```

```
CONNECT BY NOCYCLE PRIOR empno = mgr AND PRIOR mgr > mgr;
```

4. Создать запрос для определения всех примитивных пифагоровых троек x, y, z ($x^2+y^2=z^2$, x, y, z являются взаимно простыми числами) вплоть до некоторого z<=N.

Число N должно быть указано в подстановочной переменной.
Пример вывода для N=13:

x	y	z
3	4	5
5	12	13

Примечание: Взаимно простые числа — целые числа, не имеющие никаких общих делителей, кроме ±1.

Алгоритм:

1) получаем числа до n
2) находим x,y,z и их квадраты
3) оставляем лишь натуральные значения, чтобы x<>y
4) оставляем лишь взаимно простые значения

```
define n = 100;

with f1 as(
select
level as endp
from
dual
connect by
level <= &n),

f2(endp, n) as(
select
endp,
0 n
from
f1
union all
select
endp,
n + 1
from
f2
where
n + 1 <= endp),

f3 as(
select
endp as z,
n as x,
```

```
endp*endp as z_sq,
n*n as x_sq,
(endp*endp - n*n) as y_sq,
sqrt((endp*endp - n*n)) as y
from
f2
where n != 0
order by endp, n),

f4 as (
select
row_number() over(order by x) as num,
x, y, z, x_sq, y_sq, z_sq
from f3
where y = trunc(y)
and y >= 1
and x < y order by 1),

f5 as
(select
l.x, l.y, l.z
from
f4 l
left outer join
f4 r
on
l.num > r.num
and
r.x / l.x = r.y / l.y
and
r.x / l.x = r.z / l.z
where
r.num is null)

select * from f5 order by z;
```

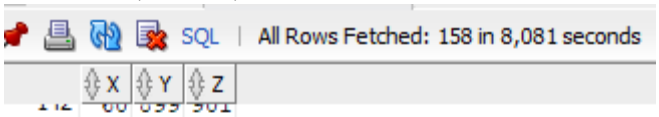
n=10

	X	Y	Z
1	3	4	5

n=100

	X	Y	Z
1	3	4	5
2	5	12	13
3	8	15	17
4	7	24	25
5	20	21	29
6	12	35	37
7	9	40	41
8	28	45	53
9	11	60	61
10	33	56	65
11	16	63	65
12	48	55	73
13	36	77	85
14	13	84	85
15	39	80	89
16	65	72	97

n=1000 (конец)



	X	Y	Z
143	464	777	905
144	616	663	905
145	43	924	925
146	533	756	925
147	129	920	929
148	215	912	937
149	580	741	941
150	301	900	949
151	420	851	949
152	615	728	953
153	124	957	965
154	387	884	965
155	248	945	977
156	473	864	985
157	696	697	985
158	372	925	997

5. Для таблицы Employees вывести тех сотрудников, для которых

результат побитовой операции OR, примененной к двоичным представлениям номеров сотрудников и их отделов, содержит последовательность из не менее четырех идущих подряд единичных битов.

Например, для King: employee_id=100 (01100100), department_id=90 (01011010), результат операции OR равен 126 (01111110) - содержит 6 идущих подряд 1.
Вывести - десятичные и двоичные значения номеров сотрудника, его отдела и результата операции OR.

Пример вывода:

DEC_emp	BIN_emp	DEC_dep	BIN_dep	DEC_or	BIN_or
100	01100100	90	01011010	126	01111110

Задачу решить без использования регулярных выражений, аналитических функций и MODEL.

Алгоритм:

1) Получаем номера сотрудников и отделов

2) Переводим номера сотрудников в двоичную с.с.

3) Переводим номера отделов в двоичную с.с.

4) Соединяем получившиеся двоичные значения и получаем значение or

5) Собираем все двоичные значения в одну строку и соединяем их с соответствующими десятичными значениями

6) Соединяем все значения и выводим только те, где в двоичном значении or встречается четыре единицы

```
with impemp as (
select
rownum rn,
employee_id emp_id,
department_id dep_id
from employees),

emptobin(rn, num, emp_id, bin) as(
select
rn,
1,
trunc(emp_id/2),
mod(emp_id, 2)
from impemp
union all
select
rn,
num+1,
trunc(emp_id/2),
mod(emp_id, 2)
from emptobin
```

```
where emp_id <> 0),

deptobin(rn, num, dep_id, bin) as(
select
rn,
1,
trunc(dep_id/2),
mod(dep_id, 2)
from impemp
union all
select
rn,
num+1,
trunc(dep_id/2),
mod(dep_id, 2)
from deptobin
where dep_id <> 0),

empdep(rn, num, emp_bin, dep_bin, or_) as(
select
emptobin.rn,
emptobin.num,
nvl(emptobin.bin, 0),
nvl(deptobin.bin, 0),
case when nvl(emptobin.bin, 0) = 1 and nvl(deptobin.bin, 0) = 1
then 1 else nvl(emptobin.bin, 0) + nvl(deptobin.bin, 0) end
from emptobin full outer join deptobin
on(emptobin.rn = deptobin.rn and emptobin.num = deptobin.num)),

bintostr (rn, num, emp_bin) as (
select
rn,
num,
replace(sys_connect_by_path(emp_bin, ','), ',', '')
from empdep
start with num = 1
connect by prior num+1=num and prior rn = rn),

emp_data as(
select
bintostr.rn,
bintostr.emp_bin,
impemp.emp_id
from bintostr join impemp
on (bintostr.rn = impemp.rn)
where num = (select max(num) from empdep e where e.rn = bintostr.rn group by rn)),

empdep(rn, num, emp_bin, dep_bin, or_) as(
select
emptobin.rn,
emptobin.num,
nvl(emptobin.bin, 0),
```

```
nvl(deptobin.bin, 0),
case when nvl(emptobin.bin, 0) = 1 and nvl(deptobin.bin, 0) = 1
then 1 else nvl(emptobin.bin, 0) + nvl(deptobin.bin, 0) end
from emptobin full outer join deptobin
on(emptobin.rn = deptobin.rn and emptobin.num = deptobin.num)),

bintostr (rn, num, emp_bin) as (
select
rn,
num,
replace(sys_connect_by_path(emp_bin, ','), ',', '')
from empdep
start with num = 1
connect by prior num+1=num and prior rn = rn),

emp_data as(
select
bintostr.rn,
bintostr.emp_bin,
impemp.emp_id
from bintostr join impemp
on (bintostr.rn = impemp.rn)
where num = (select max(num) from empdep e where e.rn = bintostr.rn group by rn)),

bintostr1 (rn, num, dep_bin) as (
select
rn,
num,
replace(sys_connect_by_path(dep_bin, ','), ',', '')
from empdep
start with num = 1
connect by prior num+1=num and prior rn = rn),

dep_data as(
select
bintostr1.rn,
bintostr1.dep_bin,
impemp.dep_id
from bintostr1 join impemp
on (bintostr1.rn = impemp.rn)
where num = (select max(num) from empdep e where e.rn = bintostr1.rn group by rn)),

bintostr2 (rn, num, or_) as (
select
rn,
num,
replace(sys_connect_by_path(or_, ','), ',', '')
from empdep
start with num = 1
connect by prior num+1=num and prior rn = rn),

or_val as(
```

```
select
rn,
sum(power(2, num - 1) * or_) or_value
from empdep
group by rn),

or_data as(
select
bintostr2.rn,
bintostr2.or_,
or_val.or_value
from bintostr2 join or_val
on (bintostr2.rn = or_val.rn)
where num = (select max(num) from empdep e where e.rn = bintostr2.rn group by rn)),

outp (DEC_emp,BIN_emp,DEC_dep,BIN_dep,DEC_or,BIN_or) as(
select
emp_data.emp_id,
reverse(emp_data.emp_bin),
dep_data.dep_id,
reverse(dep_data.dep_bin),
or_data.or_value,
reverse(or_data.or_)
from emp_data join dep_data
using (rn) join or_data
using (rn))

select * from outp where bin_or like '%1111%' order by 1,3;
```

	DEC_EMP	BIN_EMP	DEC_DEP	BIN_DEP	DEC_OR	BIN_OR
1	100	1100100	90	1011010	126	1111110
2	101	1100101	90	1011010	127	1111111
3	102	1100110	90	1011010	126	1111110
4	103	1100111	60	0111100	127	1111111
5	104	1101000	60	0111100	124	1111100
6	105	1101001	60	0111100	125	1111101
7	106	1101010	60	0111100	126	1111110
8	107	1101011	60	0111100	127	1111111
9	111	1101111	100	1100100	111	1101111
10	114	1110010	30	0011110	126	1111110
11	115	1110011	30	0011110	127	1111111
12	116	1110100	30	0011110	126	1111110
13	117	1110101	30	0011110	127	1111111
14	118	1110110	30	0011110	126	1111110
15	119	1110111	30	0011110	127	1111111
16	120	1111000	50	0110010	122	1111010

1. Вывести информацию о таблицах схемы в виде:

Имя таблицы	Столбцы первич. ключа	Столбцы с огран. уникальн.	Подч. табл. со столб. внешн. ключей
Table 1	Col1, Col2	Col3, Col4	Table2 (Col5, Col6), Table3 (Col7,Col8)

Пример результата:

Имя таблицы	Столбцы первич. ключа	Столбцы с огран. уникальн.	Подч. табл. со столб. внешн. ключей
DEPT3	DEPARTMENT_ID, DEPARTMENT_NAME	-	DEPT2(ID,NAME)
EMPLOYEES	EMPLOYEE_ID	EMAIL	DEPARTMENTS(MANAGER_ID), EMPLOYEES(MANAGER_ID), JOB_HISTORY(EMPLOYEE_ID)
EMPLOYEES	EMPLOYEE_ID	FIRST_NAME, LAST_NAME	DEPARTMENTS(MANAGER_ID), EMPLOYEES(MANAGER_ID), JOB_HISTORY(EMPLOYEE_ID)
…	…	…	…

Примечание: 1. Задачу решить без использования функций Listagg и Wm_concat.

2. Названия столбцов можно сократить.
3. Списки столбцов первичного ключа, столбцов с ограничением уникальности подчиненных таблиц отсортировать по алфавиту.

```
with pkeys as(
select
dense_rank() over (partition by ucc.table_name order by ucc.column_name) r,
ucc.table_name,
ucc.column_name p_column
from user_constraints uc join user_cons_columns ucc
on (uc.constraint_name = ucc.constraint_name and uc.constraint_type = 'P')),
pkeylist as(
select
table_name,
ltrim(sys_connect_by_path(p_column,','),',') p_column
from pkeys
connect by prior table_name = table_name and prior r < r),
pkeylist2 as(
select
table_name,
p_column
from pkeylist t1
where length(p_column) = (select max(length(p_column)) from pkeylist t2 where t1.table_name
= t2.table_name)),

ukeys as(
select
dense_rank() over (partition by ucc.table_name order by ucc.column_name) r,
ucc.table_name,
ucc.column_name u_column
from user_constraints uc join user_cons_columns ucc
on (uc.constraint_name = ucc.constraint_name and uc.constraint_type = 'U')),
ukeylist as(
select
table_name,
ltrim(sys_connect_by_path(u_column,','),',') u_column
from ukeys
```

```
connect by prior table_name = table_name and prior r < r),
ukeylist2 as(
select
table_name,
u_column
from ukeylist t1
where length(u_column) = (select max(length(u_column)) from ukeylist t2 where t1.table_name
= t2.table_name)),

r_columns as(
select
dense_rank() over (partition by uc.table_name order by ucc2.column_name) r,
ucc1.table_name p_table,
uc.table_name r_table,
ucc2.column_name r_column
from user_constraints uc join user_cons_columns ucc1
on (ucc1.constraint_name = uc.r_constraint_name)
join user_cons_columns ucc2
on (ucc2.constraint_name = uc.constraint_name)),
r_column_list as(
select
p_table,
r_table,
ltrim(sys_connect_by_path(r_column,','),',') r_column
from r_columns
connect by prior p_table = p_table and prior r_table = r_table and prior r < r),

r_tables as(
select
dense_rank() over (partition by p_table order by r_table) r,
p_table,
r_table,
r_column
from r_column_list t1
where length(r_column) = (select max(length(r_column)) from r_column_list t2 where t1.p_table
= t2.p_table and t1.r_table = t2.r_table)),
r_tablelist as(
select
p_table,
replace(ltrim(sys_connect_by_path(r_table || '(' || r_column || ')','/'),'/'), '/',',') r_table
from r_tables
connect by prior p_table = p_table and prior r < r),
r_tablelist2 as(
select *
from r_tablelist t1
where length(r_table) = (select max(length(r_table)) from r_tablelist t2 where t1.p_table =
t2.p_table))

select
t1.table_name "Имя таблицы",
nvl(t2.p_column, '-') "Столбцы первич. ключа",
nvl(t3.u_column, '-') "Столбцы с огран. уникальн.",
```

```
nvl(t4.r_table, '-') "Подч.табл. со столб.вн.ключей"
from user_tables t1 left outer join pkeylist2 t2
on(t1.table_name = t2.table_name)
left outer join ukeylist2 t3
on(t1.table_name = t3.table_name)
left outer join r_tablelist2 t4
on(t1.table_name = t4.p_table);
```

2. Создать запрос для определения всех примитивных пифагоровых троек x, y, z ($x^2+y^2=z^2$, x, y, z являются взаимно простыми числами) вплоть до некоторого z<=N.

   Число N должно быть указано в подстановочной переменной.
   Пример вывода для N=13:

x	y	z
3	4	5
5	12	13

Примечание: Взаимно простые числа — целые числа, не имеющие никаких общих делителей, кроме ±1.

```
define num = 100;

with inp_end as(    --Вводим числа от 1 до n для z
select
rownum endpoint
from all_objects
where rownum <= &num),

find_n (n, endpoint) as(    --Вводим числа от 1 до z для y
select
1,
endpoint
from inp_end
union all
select
n+1,
endpoint
from find_n
where n+1 < endpoint),

find_xyz as( --Находим x по y и z
select
sqrt(endpoint*endpoint - n*n) as x,
n as y,
endpoint as z
from find_n
where sqrt(endpoint*endpoint - n*n) = trunc(sqrt(endpoint*endpoint - n*n))
and sqrt(endpoint*endpoint - n*n) <> 0
and n <> 0
```

```
and sqrt(endpoint*endpoint - n*n) <= n),

find_xyz2 (x, y, z, od) as( --находим всевозможные общие делители
select
x,
y,
z,
2
from find_xyz
union all
select
x,
y,
z,
od + 1
from find_xyz2
where od + 1 < x),

find_xyz3 as( --получаем значения x, y, z с общим делителем
select
x,y,z
from find_xyz2
where
x/(od) = trunc(x/(od))
and y/(od) = trunc(y/(od))
and z/(od) = trunc(z/(od)))

select *    --выводим только те значения, которые не входят в список значений с обзим делителем
from find_xyz f1
where not exists (select 'X' from find_xyz3 f2 where f1.x = f2.x and f1.y = f2.y and f1.z = f2.z)
order by 1,2,3;
```

1. Создать запрос для определения всех ветвей от начальных вершин до всех конечных вершин по направленному дереву. В пределах одной ветви необходимо перемножить поле Количество и

просуммировать результат по веткам от одной начальной вершины до одной конечной вершины.

Пример исходной таблицы:

Номер дочерней вершины	Номер родительской вершины	Количество
2	1	2
5	1	2
3	2	1
4	2	2
4	6	2
6	5	3
4	3	2

Результат:

Начальная вершина	Конечная вершина	Количество ветвей	Сумма произведений
1	4	3	20

```
with tab (sv, pv, cnt) as(
select
2,      1,      2 from dual union all
select
5,      1,      2 from dual union all
select
3,      2,      1 from dual union all
select
4,      2,      2 from dual union all
select
4,      6,      2 from dual union all
select
6,      5,      3 from dual union all
select
4,      3,      2 from dual union all
select
7, 3, 2 from dual union all
select
2, 8, 2 from dual
),
begins as (
select distinct
pv begv
from tab
where connect_by_isleaf = 1
connect by prior pv = sv),
f1 as(
select
connect_by_root(pv) || sys_connect_by_path(sv,',') path
from tab
where connect_by_isleaf = 1
connect by prior sv = pv
order by length(connect_by_root(pv) || sys_connect_by_path(sv,',')) desc),
f2 as(
select
```

```
    rownum r,
    substr(path, 1, 1) beg,
    regexp_substr(path, '\d$') endv,
    path
    from f1
    where substr(path, 1, 1) in (select * from begins)),
    f3 (beg, endv, path, pv, sv, num) as(
    select
    beg,
    endv,
    path,
    '',
    '',
    0
    from f2
    union all
    select
    beg,
    endv,
    path,
    regexp_substr(path, '\d', 1, num + 1) pv,
    regexp_substr(path, '\d', 1, num + 2) sv,
    num + 1
    from f3
    where num + 1 <= regexp_count(path, ',')),

    f4 as (
    select
    beg,
    endv,
    path,
    num,
    cnt
    from f3 t1 left join tab t2 on (t1.pv = t2.pv and t1.sv = t2.sv)
    model
    dimension by (path, num)
    measures (beg, endv, cnt)
    rules iterate(10)(
       cnt[any, iteration_number] = case when iteration_number = 0 then 1
                else cnt[cv(), iteration_number] * cnt[cv(), iteration_number - 1]
                end
       )
    )

    select distinct
    beg "Начальная вершина", endv "Конечная вершина",
    (select count(path) from (select distinct path from f2 where f2.endv = t1.endv and f2.beg =
t1.beg)) "Количество ветвей" ,
    (select sum(cnt) from (
                select beg, endv, path, cnt
                from f4 t3
                where num = (
```

```
                select max(num)
                from f4 t4
                where t4.path = t3.path
                )
        ) t2
        where t1.beg = t2.beg and t1.endv = t2.endv) "Сумма произведений"
from f4 t1
```

3. Для произвольной строки, состоящей из цифр, определить все возможные наборы слов, получаемые при замене чисел на номер буквы в русском алфавите.

   Например, для строки 211221 результатом должна быть строка:

   баабба, бйбба,баафа,бакба,уку,……

```
define str = '211221'; --ввод строки
--разбиение алфавита на буквы
with alp as(
select
rownum r,
substr('абвгдеёжзийклмнопрстуфхцчшщъыьэюя', rownum, 1) as sym
from all_objects
where rownum <= 33),
--разбиение строки на цифры
inpstr as(
select
rownum r,
substr('&str', rownum, 1) as nums
from all_objects
where rownum <= length('&str')
),
--получим все возможные номера букв из строки, пронумернованные по порядку нахождения в этой строке
f1 as(
select
r,
replace(sys_connect_by_path(nums, ','), ',','') nums
from inpstr
connect by prior r < r),
--соединим номера букв с буквами из алфавита
f2 as(
select
f1.r,
nums,
sym
from f1 join alp
on(f1.nums = alp.r)
where nums <= 33 and instr(&str, nums) > 0),
--из всех возможных комбинаций букв выбираем только те, номера которых вместе образуют нашу строку
f3 as (
select distinct
replace(sys_connect_by_path(sym,','),',','') str,
ltrim(sys_connect_by_path(nums,','),',') numstr,
case when replace(sys_connect_by_path(nums,','),',','') = &str then 1 else 0 end chk
from f2
start with r in (1, 2)
connect by prior r < r
order by str),
```

```
f4 as(
select distinct
rownum r,
str
from f3
where chk = 1),
--выводим резльтат
f5 as (
select
ltrim(sys_connect_by_path(str, ','),',') rez
from f4
connect by prior r < r)
select
rez from f5
where length(rez) = (select max(length(rez)) from f5);
```

1. Имеется таблица

```
EXPERIMENTS (ID NUMBER(2) PRIMARY KEY,
START_DATE TIMESTAMP(3),
END_DATE TIMESTAMP(3),
CONSTRAINT exper_date_check CHECK(START_DATE <= END_DATE));
```

содержащая информацию о номерах, датах и временах начала и окончания экспериментов. Точность измерения времени 0.001с.
Требуется определить временные интервалы, в которые проводилось одновременно наибольшее количество экспериментов.
Вывести наибольшее количество одновременно проводившихся экспериментов и список соответствующих временных интервалов в виде символьной строки (временные интервалы разделять запятой).
Для исходных данных:

ID	START_DATE	END_DATE
1	01.03.2017 12:00:01.010	02.03.2017 06:55:20.253
2	01.03.2017 12:00:00.500	01.03.2017 22:47:00.499
3	03.03.2017 15:00:00.000	04.03.2017 15:00:00.000
4	01.03.2017 18:33:32.112	02.03.2017 19:56:17.000
5	01.03.2017 12:00:00.500	01.03.2017 18:33:32.111

результат должен быть:

МАХ КОЛ-ВО ЭКСПЕРИМЕНТОВ	ПЕРИОДЫ
3	01.03.2017 12:00:01.010 - 01.03.2017 18:33:32.111, 01.03.2017 18:33:32.112 - 01.03.2017 22:47:00.499

```
create table EXPERIMENTS (ID NUMBER(2) PRIMARY KEY,
START_DATE TIMESTAMP(3),
END_DATE TIMESTAMP(3),
CONSTRAINT exper_date_check CHECK(START_DATE <= END_DATE));

insert all
into experiments values(1, to_timestamp('01.03.2017 12:00:01.010', 'dd.mm.yyyy
hh24:mi:ss.FF3'),
   to_timestamp('02.03.2017 06:55:20.253', 'dd.mm.yyyy hh24:mi:ss.FF3'))
into experiments values(2, to_timestamp('01.03.2017 12:00:01.500', 'dd.mm.yyyy
hh24:mi:ss.FF3'),
   to_timestamp('01.03.2017 22:47:00.499', 'dd.mm.yyyy hh24:mi:ss.FF3'))
into experiments values(3, to_timestamp('03.03.2017 15:00:00.000', 'dd.mm.yyyy
```

```
hh24:mi:ss.FF3'),
        to_timestamp('04.03.2017 15:00:00.000', 'dd.mm.yyyy hh24:mi:ss.FF3'))
    into experiments values(4, to_timestamp('01.03.2017 18:33:32.112', 'dd.mm.yyyy
hh24:mi:ss.FF3'),
        to_timestamp('02.03.2017 19:56:17.000', 'dd.mm.yyyy hh24:mi:ss.FF3'))
    into experiments values(5, to_timestamp('01.03.2017 12:00:00.500', 'dd.mm.yyyy
hh24:mi:ss.FF3'),
        to_timestamp('01.03.2017 18:33:32.111', 'dd.mm.yyyy hh24:mi:ss.FF3'))
    select 1 from dual;

    select id, to_char(start_date, 'dd.mm.yyyy hh24:mi:ss.FF3') st,
        to_char(end_date, 'dd.mm.yyyy hh24:mi:ss.FF3') en
        from experiments;

    with f1 as( --выводим все возможные промежутки времени
    select
    rownum r, e1.start_date, e2.end_date
    from experiments e1 join experiments e2
    on (e1.start_date <= e2.end_date)
    ),
    f2 as(  --находим максимальное количество экспериметов всех периодов
    select max(count(*))
    from f1 e1 join experiments e2
    on(e1.start_date >= e2.start_date and e1.end_date <= e2.end_date)
    group by e1.start_date, e1.end_date
    ),
    f3 as( --находим эксперименты в которые входят периоды с максимальным количеством
    select
    to_char(e1.start_date, 'DD.MM.YYYY HH24:MI:SS.FF3') || ' - ' || to_char(e1.end_date,
'DD.MM.YYYY HH24:MI:SS.FF3') str, count(*) cnt
    from f1 e1 join experiments e2
    on(e1.start_date >= e2.start_date and e1.end_date <= e2.end_date)
    group by e1.start_date, e1.end_date having count(*) = (select * from f2)
    )
    select  --вывод
    cnt as "MAX КОЛ-ВО ЭКСПЕРИМЕНТОВ",
    listagg(str, ', ') within group (order by str) "ПЕРИОДЫ"
    from f3
    group by cnt;
```

1. Используя словарь данных, получить информацию о первичных

ключах и подчиненных таблицах всех таблиц схемы.

В списках имена столбцов и подчиненных таблиц вывести через запятую по алфавиту.

Если таблица не содержит подчиненных таблиц и\или первичного ключа, то вывести соответствующее сообщение.

Задачу решить без использования функций Listagg и Wm_concat.

Имя таблицы	Список столб. перв. ключа	Список подчин. таблиц
EMPLOYEES2	ID	Подчиненных таблиц нет
JOB_SUM_SAL	Первичного ключа нет	Подчиненных таблиц нет
TASK	ID	Подчиненных таблиц нет
ВАКАНСИЯ	КОД_ПОЗИЦИИ,ДАТА_НАЧАЛА_ДОГОВОРА,КОД_КОМПАНИИ	СОБЕСЕДОВАНИЕ
КОМАНДА_ПРОЕКТА	НОМЕР_ДОГОВОРА_СОТРУДНИКА,НОМЕР_ПРОЕКТА	Подчиненных таблиц нет
НАПРАВЛЕНИЕ	КОД_НАПРАВЛЕНИЯ	СТУДЕНТ,ГРУППА,ПЛАН_ОБУЧЕНИЯ
ПРЕПОДАВАТЕЛИ	НОМЕР_ПРЕПОДАВАТЕЛЯ	ПРЕПОДАВАТЕЛИ,ДИСЦИПЛИНЫ
СТУДЕНТ	НОМЕР_ЗАЧЕТНОЙ_КНИЖКИ	БЮДЖЕТНИК,ВЕДОМОСТЬ,КОНТРАКТНИК
УЧЕБНОЕ_ЗАВЕДЕНИЕ	НАЗВАНИЕ_ЗАВЕДЕНИЯ	ОБРАЗОВАНИЕ
ERWIN_ИЗДАТЕЛЬСТВО	НАЗВАНИЕ_ИЗДАТЕЛЬСТВА	ERWIN_КНИГА
NEW_СОТРУДНИК	НОМЕР	NEW_УЧАСТНИКИ,NEW_СОТРУДНИК,NEW_ОТДЕЛ,NEW_ПРОЕКТ
SALES_SOURCE_DATA	Первичного ключа нет	Подчиненных таблиц нет
ГРУППЫ	НОМЕР_ГРУППЫ	СТУДЕНТЫ

```
select * from user_constraints; --constraint_name, constraint_type, table_name,
r_constraint_name
select * from user_cons_columns; --constraint_name, table_name, column_name
select * from user_tables;

with f1 as( --получим все первичные ключи
select
t1.table_name,
t2.column_name
from user_constraints t1 join user_cons_columns t2
on(t1.constraint_name = t2.constraint_name)
where constraint_type= 'P'
order by 1,2
),
f2 as(   --объединим первичные ключи в строку для каждой таблицы
select
table_name,
ltrim(sys_connect_by_path(column_name,','),',') columns
from (select table_name, column_name, rownum r from f1)
connect by prior table_name = table_name and prior r < r),
f3 as( -оставим максимальную строку для каждой таблицы
select
table_name,
columns
from f2 t1
where length(columns) = (select max(length(columns)) from f2 t2 where t1.table_name =
t2.table_name)),
f4 as( --получим все подчиненные таблицы
select
t2.table_name table_name,
t1.table_name r_table_name
from user_constraints t1 join user_cons_columns t2
on(t1.r_constraint_name = t2.constraint_name)
order by 1,2
),
```

```
f5 as ( --объединим подчиненные таблицы в строку для каждой главной таблицы
select
table_name,
ltrim(sys_connect_by_path(r_table_name,','),',') tables
from (select table_name, r_table_name, rownum r from f4)
connect by prior table_name = table_name and prior r < r),
f6 as (
select
table_name,
tables
from f5 t1
where length(tables) = (select max(length(tables)) from f5 t2 where t1.table_name =
t2.table_name))
select
table_name "Имя таблицы",
nvl(t2.columns, 'Первичного ключа нет') "Список столб. перв. ключа",
nvl(t3.tables,'Подчиненных таблиц нет') "Список подчин. таблиц"
from user_tables t1 left join f3 t2
using(table_name)
left join f6 t3
using(table_name)
order by 1;
```

2. Имеется таблица с символьным столбцом. Создать запрос для вывода тех значений, которые содержат в себе палиндромы, и самые длинные выражения, представляющие из себя палиндром.

Например, для таблицы с данными:

Text
Крокодил
Колокол
Станок

Результат должен быть:

Text	**Palindrom**
Крокодил	око
Колокол	Колок, локол

Задачу решить без использования иерархических запросов и недокументированной функции Reverse.

Примечание: Палиндромом называется слово или фраза, которые одинаково читаются слева направо и справа налево

```
with inp as(
select 'Крокодил' as text from dual union all
select 'Колокол' as text from dual union all
select 'Станок' as text from dual union all
select 'Молодой' as text from dual
),
f1(text, sym) as(       --пронумеровываем симолы в каждой строке
select
text,
```

```
1
from inp
union all
select
text,
sym + 1
from f1
where sym + 1<= length(text)),

f2 (text, sym, num, str) as (    --извлекаем все подстроки из строки
select
text,
sym,
1,
upper(substr(text,sym,1))
from f1
union all
select
text,
sym,
num+1,
upper(substr(text, sym, num+1))
from f2 where num+1<=length(text)),

f3 (text, str, num, rev)as(       --реверсируем каждую подстроку
select distinct
text,
str,
0,
''
from f2
where length(str) >= 3
union all
select
text,
str,
num + 1,
substr(str, num + 1, 1) || rev
from f3
where num + 1 <= length(str))

select distinct  --выводим те подстроки, которые равны реверсивной
text "Text",
listagg(str,',') within group (order by str) over (partition by text) as "Palindrom"
from f3
where rev = str;
```

2. Имеется таблица Продажи (Номер, Название товара, Дата, Скидка %). Вывести отчет по продажам, который включает столбцы Название товара, Даты продажи, Скидка %, представив информацию таким образом, что если один и тот же товар

продавался с одной и той же скидкой несколько дней, то эти даты должны выводиться через запятую. При этом если две или более даты отличаются друг от друга на один день, то они должны быть представлены в виде интервала с дефисом в качестве разделителя.

Пример представления результата:

Название товара	**Даты продажи**	**Скидка, %**
Стул	1.02.2016, 5.02.2016, 7.02.2016-12.02.2016, 15.02.2016	5
Стол	2.02.2016, 4.02.2016	10
Кровать	2.02.2016, 6.02.2016 - 7.02.2016, 12.02.2016-15.02.2016	10
	……..	

```
with tabl (name, dat, disc) as (
select 'Стул', to_date('01.02.2016','DD.MM.YYYY'), 5 from dual union all
select 'Стул', to_date('05.02.2016','DD.MM.YYYY'), 5 from dual union all
select 'Стул', to_date('07.02.2016','DD.MM.YYYY'), 5 from dual union all
select 'Стул', to_date('08.02.2016','DD.MM.YYYY'), 5 from dual union all
select 'Стул', to_date('09.02.2016','DD.MM.YYYY'), 5 from dual union all
select 'Стул', to_date('10.02.2016','DD.MM.YYYY'), 5 from dual union all
select 'Стул', to_date('11.02.2016','DD.MM.YYYY'), 5 from dual union all
select 'Стул', to_date('12.02.2016','DD.MM.YYYY'), 5 from dual union all
select 'Стул', to_date('15.02.2016','DD.MM.YYYY'), 5 from dual union all

select 'Стол', to_date('02.02.2016','DD.MM.YYYY'), 10 from dual union all
select 'Стол', to_date('04.02.2016','DD.MM.YYYY'), 10 from dual union all

select 'Кровать', to_date('02.02.2016','DD.MM.YYYY'), 5 from dual union all
select 'Кровать', to_date('03.02.2016','DD.MM.YYYY'), 5 from dual),
f1 as(  --для каждой даты находим разницу с предыдущей и следуюущей датой
select
name, disc, dat,
dat - lag(dat) over (partition by name, disc order by dat) lag_d,
lead(dat) over (partition by name, disc order by dat) - dat lead_d
from tabl),
f2 as(
select
name,
disc,
case
when lead_d = 1 and lag_d = 1 then '' --если предыд. и следующ. даты есть то не выводим
when lead_d = 1 then to_char(dat,'DD.MM.YYYY-') --если есть следующая то выводим с тире
else to_char(dat,'DD.MM.YYYY,') end dates   --иначе выводим с запятой
from f1)
select distinct --объединяем в строку полученные даты и убираем запятую в конце
name "Название товара",
rtrim(listagg(dates) within group(order by dates) over(partition by name,disc),',') "Даты продажи",
disc "Скидка%"
from f2 order by 1;
```

1. В таблице Employees находится информация о фамилиях сотрудников, их зарплатах и номерах отделов, в которых они работают. Для каждого отдела вывести фамилии и зарплаты трех сотрудников, получающих самые высокие зарплаты в отделе. Если самую низкую зарплату у найденных трех сотрудников отдела получают и какие-то другие сотрудники этого отдела, они тоже должны попасть в список. Для отделов, в которых меньше трех сотрудников, информацию не выводить.

   Примечание. Если в таблице имеются сотрудники с максимальной зарплатой и с одинаковой фамилией в одном и том же отделе, то необходимо вывести повторяющиеся значения.

   Пример результата.

DEPARTMENT_ID	LAST_NAME	SALARY
30	Baida	11000
30	Khoo	11000
30	Raphaely	11000
30	Tobias	11000
50	Fripp	8200
50	Weiss	8000
50	Kaufling	7900
60	Hunold	9000
60	Ernst	6000
60	Austin	4800
60	Pataballa	4800
80	Russell	14000
80	Partners	13500
80	Errazuriz	12000
90	King	24000
90	De Haan	17000
90	Kochhar	17000
100	Greenberg	12008
100	Faviet	9000
100	Chen	8200

```
with f1 as( --выводим информацию о сотрудниках, пронумерованных в зависимости от их зрплаты, из отделов, где количество сотрудников больше или равно 3
select
department_id,
row_number() over (partition by department_id order by salary desc) as r,
last_name,
salary
from employees t1
where 3 <= (select count(*) from employees t2 where t1.department_id = t2.department_id
group by department_id)
and salary is not null
),
f2 as(  --оставляем сотрудников, которые попали в первую тройку или у которых зарплата равна зарплате сотрудника на 3 месте
select
```

```
department_id,
r,
case when r in (1,2,3) then last_name
when (select salary from f1 t2 where t1.department_id = t2.department_id and r = 3) = salary
then last_name
else ''
end as last_name,
salary
from f1 t1)
select  --выводим оставшихся сотрудников
department_id,
last_name,
salary
from f2 where last_name is not null order by 1,3 desc;
```

1. Создать запрос для разделения «задвоенных" данных. Например, из

CODE_OPERATION	ID_CLIENT
1000 1100	841000 841100
2000	6700 8967 5500

сделать

RN	CNT	CODE_OPERATION	ID_CLIENT
1	0	1000 1100	841000 841100
	1	1000	841000
	2	1100	841100
2	0	2000	6700 8967 5500
	1	2000	6700
	2		8967
	3		5500

```
with inp (code_operation, id_client) as(
select '1000 1100', '841000 841100' from dual union all
select '2000', '6700 8967  5500' from dual),
f2 as( --пронумеровываем каждую строку
select
rownum r,
code_operation, id_client
from inp),
f3 as(  --cnt присваеваем iter number. каждой строке code_operation, id_client кроме нулевой
присваеваем подстроку из чисел по порядку iter number
select
r, num, code_operation, id_client
from f2
model
dimension by (r, cast(0 as number(3)) as i_n)
measures (code_operation, id_client, cast(0 as number(3)) as num)
rules upsert all iterate (100)(
   num[any, iteration_number] = iteration_number,
   code_operation[any, iteration_number] = case when iteration_number = 0 then
code_operation[cv(), iteration_number]
   else regexp_substr(code_operation[cv(), 0], '\d+', 1, iteration_number) end,
   id_client[any, iteration_number] = case when iteration_number = 0 then id_client[cv(),
```

```
iteration_number]
   else regexp_substr(id_client[cv(), 0], '\d+', 1, iteration_number) end
   )
)
select     --выводим полученные данные в соответствии с заданием
case when num = 0 then to_char(r) else ' ' end rn,
num cnt,
nvl(code_operation, ' ') code_operation,
nvl(id_client, ' ') id_client
from f3
where not(code_operation is null and
id_client is null)
order by r,num;
```

1. Используя словарь данных, получить информацию об ограничениях CHECK схемы.

   В списках имена столбцов вывести через запятую. Имя таблицы не должно повторяться.

   Задачу решить без использования функций Listagg и Wm_concat.

   Пример представления результатов:

SQL | All Rows Fetched: 8 in 1,748 seconds

	Имя таблицы	Имя ограничения	Столбцы,входящие в ограничение	Ограничение CHECK
1	EMPLOYEES	EMP_SALARY_MIN	SALARY	salary > 0
2	JOB_HISTORY	JHIST_DATE_INTERVAL	END_DATE,START_DATE	end_date > start_date
3	TASK_13	COLUMN12345	COLUMN_1,COLUMN_2,COLUMN_3,COLUMN_4,COLUMN_5	column_1>ALL (column_2, column_3, column_4, column_5)
4	TRIP	CON_PER_BEG	PER_BEG	per_beg=TRUNC(per_beg)
5		CON_PER_END	PER_END	per_end=TRUNC(per_end)
6	ПРЕПАРАТ	ПРЕПАРАТ_CHK1	АББРЕВ_ФОРМЫ,ТИП	Аббрев_формы!=Тип
7	ПРЕПОДАВАТЕЛЬ	SYS_C00201812	КАФЕДРА	КАФЕДРА Like 'Кафедра%'
8	УСПЕВАЕМОСТЬ	SYS_C00201820	ОЦЕНКА	ОЦЕНКА between 1 and 5

```
with f1 as( --выводим все огранчения CHECK и соответствующие столбцы
select
t1.table_name,
t1.constraint_name,
rownum r,
t2.column_name,
t1.search_condition
from user_constraints t1 join user_cons_columns t2
on(t1.constraint_name= t2.constraint_name and constraint_type = 'C')
),
f2 as(  --объединяем стобцы в стоку по одинаковым таблицам и ограничениям
select
table_name,
constraint_name,
ltrim(sys_connect_by_path(column_name,','),',') column_list,
search_condition
from f1
connect by prior table_name = table_name and prior constraint_name = constraint_name and prior r <
            r),
f3 as(  -- оставляем строки столбцов с максимальной длиной
select
table_name,
row_number() over (partition by table_name order by constraint_name) r,
constraint_name,
```

```
column_list,
search_condition
from f2 t1
where length(column_list) = (select max(length(column_list)) from f2 t2 where t1.table_name =
        t2.table_name and t1.constraint_name = t2.constraint_name))
select  --выводим полученные результаты в соответствии с заданием
case when r = 1 then table_name else ' ' end "Имя таблицы",
constraint_name "Имя ограничения",
column_list "Столбцы, входящие в огр.",
search_condition "Ограничение CHECK"
from f3
order by table_name, r;
```

5. Определить список последовательностей подчиненности от преподавателей, не имеющих начальника, до преподавателей, не имеющих подчиненных. Если список состоит более, чем из четырех фамилий, то выводить только две первые и две последние фамилии, а вместо остальных фамилий поставить многоточие. У преподавателей, не имеющих подчиненных приписать – (не имеет подчиненных). Если в списке четыре или меньше фамилий, то список выводится полностью.

   Результат представить в виде:

   Костыркин-> Викулина-> …->Соколов->Казанко (не имеет подчиненных)

```
with f1 as( --выводим данные из таблицы преподаватели
select
номер_преподавателя as id,
фамилия as last_name,
подчиняется as man_id
from преподаватели),
f2 as( --объединяем фамилии в строку по порядку иерархии начиная от преподавателя без
начальника
select
level lvl,
ltrim(sys_connect_by_path(last_name,'->'),'->') paths
from f1
start with man_id is null
connect by prior id = man_id),
f3 as(  --оставляем строки с максимальной длиной
select
rownum r,
lvl,
paths
from f2 t1
where length(paths) = (select max(length(paths)) from f2 t2 where
regexp_substr(t1.paths,'(^\w+->)') = regexp_substr(t2.paths,'^(\w+->)')))
select  --если список состоит более чем из 4 преподавателей, то оставляем первые 2 и
последние 2 преподавателя, а остальных заменяем многоточием
case when lvl > 4 then regexp_replace(paths,'(^\w+->\w+->)(.+)(->\w+->\w+$)', '\1...\3') else
paths end || ' (Не имеет подчиненных)' as rezult
from f3;
```

1. Для месяца произвольно заданной даты найти 3 ближайших после данного "месяца-двойника".

   "Двойниками" считать такие месяцы, которые и начинаются в один и тот же день недели, и

заканчиваются в один и тот же день недели. Дата вводится при выполнении запроса и может быть любой допустимой в СУБД датой.

Выводить: 1. Заданная дата;
2.-4. Три ближайших "месяца-двойника" в формате "Mon YYYY"

Пример результата:

Date	Month 1	Month 2	Month 3
5 Авг 2005	Май 2006	Янв 2007	Окт 2007

```
define x = '05.08.2005';
define x = '03.01.-4712';
define x = '05.03.9998';
define x = '05.08.-0002';

alter session set nls_language = 'russian';

with inp_dat as( --ввод даты
select
to_date('&x', 'DD.MM.SYYYY') dat
from dual),

f1 (dat, fst, lst, num, mth1, mth2, mth3)as(    --найдем дни недели первого и последнего числа месяца даты
select
dat,
to_char(trunc(dat, 'MONTH'), 'day'),
to_char(last_day(dat), 'day'),
case when trunc(dat, 'MONTH') = to_date('01.12.9999', 'DD.MM.SYYYY') then null
else 0 end,
null,
null,
null
from inp_dat

union all

select
dat, fst, lst,

--если последний месяц до нашей эры, то к num добавляем 13 месяцев. если мы дошли до последнего месяца 9999 года то выходим из цикла
case when trunc(add_months(dat, num + 1), 'MONTH') = to_date('01.12.9999',
'DD.MM.SYYYY') then null
when  trunc(add_months(dat, num + 1), 'MONTH') = to_date('01.12.-0001', 'DD.MM.SYYYY')
then num + 13
else num + 1 end,

--если очередной месяц удовлетворяет условию, а mth1, mth2, mth3 равны null, то сохраняем первый месяц
case when mth1 is null and mth2 is null and mth3 is null
and to_char(trunc(add_months(dat, num + 1), 'MONTH'), 'day') = fst
and to_char(last_day(add_months(dat, num + 1)), 'day') = lst
```

```
and add_months(dat, num + 1) != dat
then to_char(add_months(dat, num + 1), 'Mon SYYYY')
else mth1 end,

--если очередной месяц удовлетворяет условию и не равен mth1, а mth2, mth3 равны null,
то сохраняем второй месяц
case when mth1 is not null and mth2 is null and mth3 is null
and to_char(trunc(add_months(dat, num + 1), 'MONTH'), 'day') = fst
and to_char(last_day(add_months(dat, num + 1)), 'day') = lst
and add_months(dat, num + 1) != dat
and to_char(add_months(dat, num + 1), 'Mon SYYYY') != mth1
then to_char(add_months(dat, num + 1), 'Mon SYYYY')
else mth2 end,

--если очередной месяц удовлетворяет условию и не равен mth1 и mth2, а mth3 равен null,
то сохраняем третий месяц
case when mth1 is not null and mth2 is not null and mth3 is null
and to_char(trunc(add_months(dat, num + 1), 'MONTH'), 'day') = fst
and to_char(last_day(add_months(dat, num + 1)), 'day') = lst
and add_months(dat, num + 1) != dat
and to_char(add_months(dat, num + 1), 'Mon SYYYY') != mth1
and to_char(add_months(dat, num + 1), 'Mon SYYYY') != mth2
then to_char(add_months(dat, num + 1), 'Mon SYYYY')
else mth3 end

from f1
where num is not null
)
select  --выводим в соответствии с заданием
ltrim(replace(to_char(dat, 'DD Mon SYYYY'),'  ', ' '),'0') "Date",
nvl(replace(mth1, '  ', ' '),' ') "Month 1",
nvl(replace(mth2, '  ', ' '),' ') "Month 2",
nvl(replace(mth3, '  ', ' '),' ') "Month 3"
from f1 where num is null;
```

1. Определить многостолбцовые ограничения для каждой таблицы схемы.

   Результат представить в виде:

Номер таблицы	Таблица	Номер ограничения	Имя ограничения	Тип ограничения	Кол-во столбцов	Кол-во многостолбцовых ограничений
1	JOB_HISTORY	1	JHIST_EMP_ID_ST_DATE_PK	Первичный ключ	2	2
		2	JHIST_DATE_INTERVAL	Ограничение CHECK	2	
2	СОТРУДНИК_ПРОЕКТ	1	ХРКСОТРУДНИК_ПРОЕКТ	Первичный ключ	2	1

Данные должны быть отсортированы по названию таблиц по алфавиту.
Номера таблиц должны быть заданы в соответствии с указанной сортировкой.

Номер таблицы, название таблицы и количество многостолбцовых ограничений не должны повторяться для одной таблицы.
Для каждой таблицы данные должны быть отсортированы по именам ограничений.
Номера ограничений в таблицах должны быть заданы в соответствии с указанной сортировкой.
Информация о таблицах без многостолбцовых ограничений также должна выводиться с указанием, что таких ограничений нет (их количество равно нулю)

```
with f1 as( --выводим данные ограничения и количество столбцов каждого ограничения
select
table_name,
constraint_name,
constraint_type,
(select count(column_name) from user_cons_columns t2 where t1.constraint_name = t2.constraint_name and t1.table_name = t2. table_name) as cnt_col
from user_constraints t1),
f2 as(  --оставляем только многостолбцовые ограничения. пронумеровывем таблицы и ограничения каждой таблицы. находим количество ограничений каждой таблицы
select
dense_rank() over (order by table_name) rn_t,
table_name,
dense_rank() over (partition by table_name order by constraint_name) rn_c,
constraint_name,
constraint_type,
cnt_col,
count(constraint_name) over (partition by table_name) cnt_con
from f1
where cnt_col > 1)
select  --выводим полученные данные в соответствии с заданием
case when rn_c = 1 then to_char(rn_t) else ' ' end "Номер таблицы",
case when rn_c = 1 then table_name else ' ' end "Таблица",
rn_c "Номер ограничения",
constraint_name "Имя ограничения",
case when constraint_type = 'C' then 'Ограничение CHECK'
when constraint_type = 'P' then 'Первичный ключ'
when constraint_type = 'U' then 'Уникальный ключ'
when constraint_type = 'R' then 'Ограничение FOREIGN KEY'
end "Тип ограничения",
cnt_col "Кол-во столбцов",
case when rn_c = 1 then to_char(cnt_con) else ' ' end "Кол-во многостолбц. огр."
from f2
order by rn_t, rn_c;
```

1. Используя обращение только к таблице DUAL, построить SQL-запрос, возвращающий один столбец, содержащий календарь на заданный месяц заданного года:

- номер дня в месяце (цифрами),

□ полное название месяца по-английски заглавными буквами (в верхнем регистре),
□ год (четыре цифры),
□ полное название дня недели по-английски строчными буквами (в нижнем регистре).

Каждое "подполе" должно быть отделено от следующего одним пробелом. В результате не должно быть начальных и хвостовых пробелов. Количество возвращаемых строк должно точно соответствовать количеству дней в текущем месяце. Строки должны быть упорядочены по номерам дней в месяце по возрастанию.
Календарь должен создаваться для любых допустимых значений дат Oracle.
Задачу решить без использования разделов Model и рекурсивногоWith.
Пример вывода результата:
1 MAY 2020 friday
2 MAY 2020 saturday
……

решение 2:

```
define x = '03.11.2001';
define x = '03.11.-0001';

with inpdate as (   --ввод даты
select
to_date('&x', 'DD.MM.SYYYY') dat
from dual),

f1 as(  --создадим таблицу из 6 строк
select 1 from dual union all
select 2 from dual union all
select 3 from dual union all
select 4 from dual union all
select 5 from dual union all
select 6 from dual
),

f2 as(  --чтобы декартово умножить ее на саму себя и получить 36 номеров строк
select
rownum r
from f1 cross join f1)

select --по полученным номерам находим дни соответствующего месяца и выводим результаты в соответствии с заданием
rtrim(ltrim(regexp_replace(rez,'  ',' '),'0'),' ') rez
from(
   select
     to_char(to_date(r || to_char(dat, '.MM.SYYYY'), 'DD.MM.SYYYY'),'DD MONTHSYYYY day', 'nls_date_language = english') rez
   from f2, inpdate
   where r <= to_char(last_day(dat), 'DD')
   );
```

3. Написать запрос, который все пары прямых скобок в строке, внутри которых имеется две или более пары прямых скобок, заменит на фигурные скобки. Например, для

строки

[[[98+77]-9]-1] => [[175-9]-[1]]=>165

результат должен быть **ДОДЕЛАТЬ**

{[[98+77]-9]-1} =>{[175-9]-[1]}=>165

```
define x = '[[[[98+77]-9]-1]] => [[175-9]-[1]]=>165';

with inpstr as(
select
'&x' as str
from dual),
f2 (str, num) as(
select
str,
0
from inpstr
union all
select
regexp_replace(str,
'\[(([^][]*\[[^][]*\[[^][]*\][^][]*\][^][]*)|([^][]*\[[^][]*\][^][]*(\[[^][]*\][^][]*)+)|([^][]*\[[^][]*(\[[
^][]*\])*[^][]*\][^][]*(\[[^][]*(\[[^][]*\])*[^][]*\][^][]*)+))\]', '{\1}'),
num + 1
from f2
where num <= 20)
select str from f2 where num = 20;
```

Используя словарь данных, получить информацию о подчиненности таблиц в схеме в виде:

ИмяТаблицы1(ИмяFK1(Список столбцов) *ссылается на* ИмяТаблицы2/ИмяКлюча2(Список столбцов))

ИмяТаблицы2(ИмяFK2(Список столбцов) *ссылается на* ИмяТаблицы3/ИмяКлюча3(Список столбцов))

……

```
with f1 as (    --получаем данные о таблицах, ограничениях и столбцах, пронумеровыванные по порядку сортировки по column_name
select
table_name,
rownum r,
constraint_name,
column_name
from (select table_name, constraint_name, column_name from user_cons_columns order by column_name)),
f2 as(   --объединяем столбцы в строку по ограничениям
select
table_name,
constraint_name,
ltrim(sys_connect_by_path(column_name, ','),',') col_list
from f1
connect by prior table_name = table_name and prior constraint_name = constraint_name and prior r < r),
f3 as(  --оставляем только строки с максимальной длиной
select
table_name,
constraint_name,
col_list
from f2 t1
where length(col_list) = ( select max(length(col_list)) from f2 t2 where t1.table_name = t2.table_name and t1.constraint_name = t2.constraint_name)),
f4 as(  --соединяем зависимые ограничения с ограничениями, на которые они ссылаются
select
t1.table_name r_tab,
t1.constraint_name r_con,
t2.col_list r_col,
t3.table_name p_tab,
t3.constraint_name p_con,
t3.col_list p_col
from user_constraints t1 join f3 t2
on(t1.table_name = t2.table_name and t1.constraint_name = t2.constraint_name)
join f3 t3
on(t1.r_constraint_name = t3.constraint_name))
select  --выводим в соответствии с заданием
r_tab||' ('||r_con||' ('||r_col||') ссылается на ' || p_tab || ' / ' || p_con || ' ('||p_col||'))' as rez
from f4 order by 1;
```

3. Создать запрос для получения информации об успеваемости студентов в виде:

ФИО	Дисциплина	Оценка	Дата	Примечания
Петров	Математика	5	20.1.2008	
	Физика	4	22.1.2008	
	Химия	2	25.1.2008	
	Химия	3	27.1.2008	Пересдача
Усов	Математика	5	12.06.99	
	Экономика	3	15.06.99	
	Менеджмент	2	17.06.99	
	Менеджмент	4	18.06.99	Пересдача
Судаков	Экзамены не сдавал			

В таблице должна быть представлена информация только по результатам сдачи экзаменов по дисциплинам, предусмотренным учебным планом для специальности, на которой учится студент.

Исходные данные в таблицах Студенты, Успеваемость, Группы, Учебный план, Дисциплины.

```
with f1 as( --получим данные о тех экзаменах, которые предусмотрены учебным планом для специальности студента
select
номер_студента,
номер_дисциплины,
оценка,
дата
from успеваемость t1
where номер_дисциплины in (select номер_дисциплины from учебные_планы where код_специальности = (select код_специальности from группы where номер_группы = (select номер_группы from студенты t2 where t1.номер_студента = t2.номер_студента)))
),
f2 as(      --если у студента уже был экзамен по этому предмету ранее, то в примечании указываем, что это пресдача
select
номер_студента,
номер_дисциплины,
оценка,
дата,
case when (select count(*) from f1 t2 where t1.номер_студента = t2.номер_студента and t1.номер_дисциплины = t2.номер_дисциплины and t2.дата < t1.дата) > 0 then 'Пересдача' else ' ' end per
from f1 t1)
select  --выводим полученные данные в соответствии с заданием
case when row_number() over (partition by номер_студента order by номер_дисциплины, дата) = 1 then фамилия || ' ' || имя || ' ' || отчество else ' ' end as ФИО,
nvl(название, 'Экзамены не сдавал') as "Дисциплина",
nvl(to_char(оценка), ' ') "Оценка",
nvl(case when extract(year from дата) >= 2000 then to_char(дата, 'DD.fmMM.YYYY') else to_char(дата, 'DD.MM.YY') end, ' ')  "Дата",
nvl(per, ' ')  "Примечание"
```

```
from студенты left join f2 t1
using (номер_студента)
left join дисциплины
using (номер_дисциплины)
order by фамилия,имя,отчество,номер_дисциплины,дата;
```