

1. Определить многостолбцовые ограничения для каждой таблицы схемы.

Результат представить в виде:

Номер таблиц ы	Таблица	Номер ограничен ия	Имя ограничения	Тип ограничен ия	Кол-во столбцо в	Кол-во многостолбцов ых ограничений
1	JOB_HISTORY	1	JHIST_EMP_ID_ST_DATE _PK	Первичный ключ	2	2
		2	JHIST_DATE_INTERVAL	Ограничени е CHECK	2	
2	СОТРУДНИК_ПРО ЕКТ	1	ХРКСОТРУДНИК_ПРОЕК Т	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 cons AS (

/*отсекаем ограничения типа CHECK типа NOT NULL*/

SELECT t1.table_name, constraint_name, t2.column_name, t1.constraint_type

FROM user_constraints t1 LEFT OUTER JOIN user_cons_columns t2 USING (constraint_name)

MINUS

SELECT t1.table_name, constraint_name, t2.column_name, t1.constraint_type

FROM user_constraints t1 LEFT OUTER JOIN user_cons_columns t2 USING (constraint_name)

WHERE UPPER(constraint_type) = 'C' AND INSTR(UPPER(search_condition_vc), '*IS NOT NULL')

<> 0

),

/*считаем кол-во столбцов для каждого ограничения*/

cols_number AS (

SELECT L.table_name, L.constraint_name, L.constraint_type, COUNT(R.column_name)+1 as

col_number

FROM cons L LEFT OUTER JOIN cons R

ON (L.constraint_name = R.constraint_name AND L.column_name < R.column_name)

GROUP BY L.table_name, L.constraint_name, L.constraint_type

ORDER BY L.table_name, L.constraint_name

),

/*получаем многостолбцовые ограничения*/

many_cols_cons AS (

SELECT table_name, constraint_name, constraint_type, col_number

FROM cols_number

WHERE col_number > 1

```

ORDER BY table_name, constraint_name
),
/*нумеруем ограничения внутри каждой таблицы*/
cons_num AS (
    SELECT L.table_name, L.constraint_name, L.constraint_type, L.col_number,
COUNT(R.constraint_name)+1 as con_num
    FROM many_cols_cons L LEFT OUTER JOIN many_cols_cons R
    ON (L.table_name = R.table_name AND L.constraint_name > R.constraint_name)---> если по
алфавиту, < если как в примере
    GROUP BY L.table_name, L.constraint_name, L.constraint_type, L.col_number
    ORDER BY L.table_name, L.constraint_name
),
/*считаем кол-во многостолбцовых ограничений для каждой таблицы*/
cons_number AS (
    SELECT L.table_name, COUNT(R.constraint_name)+1 as con_number
    FROM many_cols_cons L LEFT OUTER JOIN many_cols_cons R
    ON (L.table_name = R.table_name AND L.constraint_name > R.constraint_name)
    GROUP BY L.table_name
    ORDER BY L.table_name
),

alltab AS (
    SELECT table_name FROM user_tables
    ORDER BY table_name
),

alltab_num AS (
    SELECT ROWNUM as n, table_name
    FROM alltab
),

finally AS (
    SELECT t1.n, t1.table_name, t2.con_num, t2.constraint_name, t2.constraint_type,
t2.col_number, t3.con_number
    FROM alltab_num t1 LEFT OUTER JOIN cons_num t2 ON (t1.table_name = t2.table_name) LEFT
OUTER JOIN cons_number t3
    ON (t1.table_name = t3.table_name)
    ORDER BY t1.n, t1.table_name, t2.con_num
)
SELECT CASE WHEN con_num = 1 OR con_number IS NULL THEN TO_CHAR(n) ELSE ' ' END as
"Номер таблицы",
    CASE WHEN con_num = 1 OR con_number IS NULL THEN table_name ELSE ' ' END as
"Таблица",
    NVL(TO_CHAR(con_num), ' ') as "Номер ограничения",
    NVL(constraint_name, 'Многостолбцовых ограничений нет') as "Имя ограничения",
    CASE constraint_type WHEN 'U' THEN 'Ограничение UNIQUE'
        WHEN 'P' THEN 'Первичный ключ'
        WHEN 'C' THEN 'Ограничение CHECK'
        WHEN 'R' THEN 'Внешний ключ'
        ELSE ' ' END as "Тип ограничения",

```

```
NVL(TO_CHAR(col_number), ' ') as "Кол-во столбцов",
CASE WHEN con_num = 1 OR con_number IS NULL
THEN NVL(TO_CHAR(con_number), '0') ELSE ' ' END as "Кол-во многостолбцовых огран."
```

FROM finally;

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

```
CREATE TABLE ZAD13(TEXT VARCHAR2(50)CONSTRAINT zad13_pk PRIMARY KEY);
INSERT INTO ZAD13 VALUES('Крокодил');
INSERT INTO ZAD13 VALUES('Колокол');
INSERT INTO ZAD13 VALUES('Крокодил и колокол');
INSERT INTO ZAD13 VALUES('Станок');
WITH REVERSE_STR AS (SELECT TEXT,
REVERSE(TEXT) AS REV_TEXT,
LENGTH(TEXT) AS LEN,
ROWNUM AS RN
FROM ZAD13),
PALINDROM AS (SELECT RN, 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)
SELECT TEXT, STRS AS "Palindrom"
FROM (SELECT RN, TEXT,
FIRST_VALUE(STR) OVER (PARTITION BY RN ORDER BY LENGTH(STR) DESC, 2 ASC)
AS STRS
FROM PALINDROM
WHERE STR IS NOT NULL
ORDER BY RN, TEXT)
GROUP BY TEXT, STRS;

```

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

--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 предыдущему
--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
--выпишем две крайние даты через дефис
a7 AS (
SELECT "Название товара", CASE WHEN mid = mad THEN
TO_CHAR(mid,'fmdd.fmmm.yyyy')
ELSE TO_CHAR(mid,'fmdd.fmmm.yyyy') || '-' || TO_CHAR(mad,'fmdd.fmmm.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 "Даты продажи", "Скидка %"
FROM a7;

```

- 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:

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 cons AS (

SELECT t1.table_name, t1.constraint_name, t2.column_name, t1.search_condition

FROM user_constraints t1 LEFT OUTER JOIN user_cons_columns t2 ON (t1.constraint_name = t2.constraint_name)

WHERE t1.constraint_type = 'C'

),

cols_number AS(

SELECT L.table_name, L.constraint_name, L.column_name, COUNT(R.column_name) + 1 AS

col_num

FROM cons L LEFT OUTER JOIN cons R ON

L.table_name = R.table_name AND L.constraint_name = R.constraint_name

AND

L.column_name < R.column_name

GROUP BY L.table_name, L.constraint_name, L.column_name

ORDER BY 1, 3 DESC

),

cons_number AS(

SELECT L.table_name, L.constraint_name, COUNT(R.constraint_name) + 1 AS cons_num

FROM cons L LEFT OUTER JOIN cons R ON

L.table_name = R.table_name

AND

L.constraint_name < R.constraint_name

GROUP BY L.table_name, L.constraint_name

ORDER BY 1

),

listed AS (

SELECT ROWNUM as i, t1.table_name, t3.cons_num, t1.constraint_name,

LTRIM(SYS_CONNECT_BY_PATH(t1.column_name, ',')) as cols,

t1.search_condition, t2.col_num

FROM cons t1 LEFT OUTER JOIN cols_number t2 ON

(t1.table_name = t2.table_name AND t1.constraint_name = t2.constraint_name AND

t1.column_name = t2.column_name)

```
LEFT OUTER JOIN cons_number t3 ON (t1.table_name = t3.table_name AND t1.constraint_name  
= t3.constraint_name)  
WHERE CONNECT_BY_ISLEAF = 1  
START WITH t2.col_num = 1  
CONNECT BY PRIOR t1.table_name = t1.table_name AND PRIOR t1.constraint_name =  
t1.constraint_name AND PRIOR t2.col_num = t2.col_num -1  
ORDER BY 1, 3  
)
```

```
SELECT CASE cons_num WHEN 1 THEN listed.table_name ELSE '' END AS "Имя таблицы",  
constraint_name as "Имя ограничения", cols as "Столбцы, входящие в ограничение",  
search_condition as "Ограничение CHECK"  
FROM listed, user_tables  
WHERE listed.table_name = user_tables.table_name  
ORDER BY listed.table_name, cons_num;
```

НЕПОНЯТНЫЕ СКРИПТЫ

1)

```
with f1 as(
select
t1.table_name,
t1.constraint_name,
rownum r,
t2.column_name,
t1.search_condition_vc search_condition
from user_constraints t1 join user_cons_columns t2
on(t1.constraint_name= t2.constraint_name and t1.constraint_type = 'C')
),
```

```
f2 as(
select
table_name,
constraint_name,
ltrim(sys_connect_by_path(column_name,',','') column_list,
search_condition
from f1
where search_condition <> '' || column_name || '' || ' IS NOT NULL'
connect by prior table_name = table_name and prior constraint_name = constraint_name and prior
r < r),
```

```
f3 as(
select
table_name,
constraint_name,
rownum r,
column_list,
search_condition
from(
select
t3.table_name,
t1.constraint_name,
t1.column_list,
t1.search_condition
from user_tables t3 left join (
select
table_name,
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)
) t1
```



```

        on(t3.table_name = t1.table_name
        )
    order by 1,2
    )
)

```

```

select
case when r = (select min(r) from f3 t2 where t1.table_name = t2.table_name) then table_name else
'' end "Имя таблицы",
nvl(constraint_name, 'Ограничений CHECK нет') "Имя ограничения",
nvl(column_list, ' ') "Столбцы, входящие в огр.",
nvl(search_condition, ' ') "Ограничение CHECK"
from f3 t1
order by table_name, r;

```

2)

```

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 union all
select
4, 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)

```

with f1 as(
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(
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
department_id,
last_name,
salary
from f2 where last_name is not null order by 1,3 desc;

```