В базе данных развернута схема SH, используя которую необходимо выполнить задания.

Напомню, что при работе в схеме LEARN (как было на курсах), обращаться к таблицам в схеме SH следует с явным указанием схемы.

Например, select \* from sh.countries;

В приложении к письму находится схема таблиц их взаимосвязи, называемая также ER-диаграммой (от Entity-Relationship, Сущности-Связи)

Задание следующее:

Написать запрос, выводящий...

1. Имя и фамилию самых бедных клиентов - среди замужних женщин, не проживающих ни в Японии, ни в Бразилии, ни в Италии. Богатство определяется по кредитному лимиту. [Отсортировать по CUST\_LAST\_NAME].

SELECT CUST\_FIRST\_NAME, CUST\_LAST\_NAME FROM SH.CUSTOMERS JOIN SH.COUNTRIES USING(COUNTRY\_ID)

WHERE SH.COUNTRIES.COUNTRY\_NAME NOT IN ('Japan', 'Brazil', 'Italy')

AND SH.CUSTOMERS.CUST\_GENDER = 'F'

AND SH.CUSTOMERS.CUST\_MARITAL\_STATUS = 'married'

AND SH.CUSTOMERS.CUST\_CREDIT\_LIMIT = (SELECT MIN(CUST\_CREDIT\_LIMIT) FROM SH.CUSTOMERS)

ORDER BY CUST\_LAST\_NAME;

----------------------------------------------------------------------------

2. Клиента с самым длинным домашним адресом, чей телефонный номер заканчивается на 77. Вывести результат в одном столбце, в формате “Name: [cust\_first\_name] [cust\_last\_name]; city: [cust\_city]; address: [cust\_street\_address]; number:[cust\_main\_phone\_number]; email: [cust\_email]; ”. (всё, что обернуто в [] – названия полей (столбцов) таблицы).

WITH END77 AS (select \* from customers where (cust\_main\_phone\_number like'%77'))

select 'Name: '||cust\_first\_name||' '||cust\_last\_name||

'; city: '||cust\_city||'; address: '||cust\_street\_address||

'; number: '||cust\_main\_phone\_number||'; email:'||cust\_email||';' as DataName

from END77

where (length(cust\_street\_address) = (select max(length(cust\_street\_address)) from END77));

-----------------------------------------------------------------------------

3. Всех клиентов, которые купили самый дешевый продукт в категории ПО (Software/Other) или Фототехника (Photo) (связка таблиц CUSTOMERS -> SALES -> PRODUCTS), среди тех, кто родился позже 1980 года [самостоятельно выбрать категории, которые есть в таблице];

SELECT CUST\_ID, CUST\_FIRST\_NAME, CUST\_LAST\_NAME, CUST\_YEAR\_OF\_BIRTH, PROD\_CATEGORY

FROM SH.CUSTOMERS JOIN SH.SALES USING(CUST\_ID) JOIN SH.PRODUCTS USING(PROD\_ID)

WHERE CUST\_YEAR\_OF\_BIRTH > 1980

AND PROD\_LIST\_PRICE = (SELECT MIN(PROD\_LIST\_PRICE) FROM SH.PRODUCTS WHERE PROD\_CATEGORY IN ('Girls', 'Women'));

-------------------------------------------------------------------------

4. Всех клиентов-мужчин с уровнем дохода "D", у которых не заполнено поле "семейное положение" и которые проживают в США или Германии (с использованием EXISTS). Отсортировать по cust\_id.

SELECT \* FROM SH.CUSTOMERS

WHERE CUST\_GENDER = 'M' AND CUST\_INCOME\_LEVEL LIKE 'D%' AND CUST\_MARITAL\_STATUS IS NULL AND

EXISTS (SELECT COUNTRY\_ID FROM SH.COUNTRIES

WHERE COUNTRY\_NAME IN ('Germany', 'United States of America')

AND SH.COUNTRIES.COUNTRY\_ID = SH.CUSTOMERS.COUNTRY\_ID)

order by cust\_id;

--------------------------------------------------------------------------

5. Среднюю сумму покупки (является произведением цены товара на количество проданного товара (quantity\_sold)) в каждой стране, полное название страны. Отсортировать в порядке убывания средней суммы.

SELECT CTR.COUNTRY\_NAME, AVG(PRD.PROD\_LIST\_PRICE \* SLS.quantity\_SOLD) AS AVG\_PRICE

FROM SH.CUSTOMERS CUS

JOIN SH.SALES SLS ON SLS.CUST\_ID = CUS.CUST\_ID

JOIN SH.PRODUCTS PRD ON PRD.PROD\_ID = SLS.PROD\_ID

JOIN SH.COUNTRIES CTR ON CTR.COUNTRY\_ID = CUS.COUNTRY\_ID

GROUP BY CTR.COUNTRY\_NAME

ORDER BY AVG\_PRICE DESC;

------------------------------------------------------------------------

6. "популярность" почтовых доменов клиентов, т.е. количество клиентов с почтой в каждом из доменов.

SELECT DOMAIN, COUNT(\*) AS "DOMAIN\_COUNT"

FROM (SELECT SUBSTR(CUST\_EMAIL,INSTR(CUST\_EMAIL,'@')) AS "DOMAIN" FROM SH.CUSTOMERS)

GROUP BY DOMAIN;

-----------------------------------------------------------------------------

7. распределение общего количества покупок категории "Мужчины" (Peripherals and Accessories) по странам (т.е. распределение по странам, в которых проживают клиенты), в конечной выборке оставить те страны, в которых общее количество покупок выше, чем средняя сумма покупок в стране (по всему миру) [самостоятельно выбрать категорию, которая есть в таблице]. Упорядочить по полному названию стран.

with Table\_City as(select country\_name,sum(amount\_sold) as sum\_sold

from sh.sales sl

join sh.products pr on pr.prod\_id = sl.prod\_id

join sh.customers ct on sl.cust\_id=ct.cust\_id

join sh.countries coun on ct.country\_id = coun.country\_id

group by country\_name,prod\_category

having (prod\_category = 'Men'))

select country\_name,sum\_sold from Table\_City

where sum\_sold>(select avg(sum\_sold) from Table\_City);

-------------------------------------------------------------------------

8. Процентное соотношение мужчин и женщин, проживающих в каждой стране, отсортированное по названию страны в алфавитном порядке. Столбцы в выводе должны быть такими: «Страна», «% мужчин», «% женщин» [использовать WITH]. Упорядочить по полному названию стран.

WITH TBL AS (SELECT \* FROM

(SELECT COUNT(CUST\_GENDER) AS GEND\_M, COUNTRY\_ID FROM

SH.CUSTOMERS where CUST\_GENDER='M' GROUP BY COUNTRY\_ID)

INNER JOIN

(SELECT COUNT(CUS.CUST\_GENDER) AS GEND\_F, CUS.COUNTRY\_ID FROM

SH.CUSTOMERS CUS where CUS.CUST\_GENDER='F' GROUP BY CUS.COUNTRY\_ID) USING(COUNTRY\_ID))

----------------------------------------------------------------------------9. Максимальное суммарное количество проданных единиц товара за день для каждого продукта. Запрос должен выводить TOP 20 строк, отсортированных по убыванию количества покупок (Столбцы должны быть такими: "Макс покуп/день", prod\_name) [Под первым столбцом подразумевается объединение в одно поле количества покупок и последней даты, за которую сделаны эти покупки].

select mxn||'/'||time\_id as " Ìàêñ ïîêóï/äåíü ",prod\_subcategory as "Ïðîä Êàòåã"

from

(

select mxn,time\_id,prod\_subcategory,row\_number() over ( order by mxn desc) rn

from

(select max(quantity\_sold) as mxn,time\_id,prod\_subcategory

from sh.products p join sh.sales s on s.prod\_id=p.prod\_id

group by prod\_subcategory,time\_id)

)

where rn between 1 and 20;

with

summAmount as

(select sum(quantity\_sold) as sum1, prod\_name, time\_id

from sh.sales inner join sh.products using (prod\_id) group by prod\_name, time\_id),

maxAmount as(select max(sum1) as maxx, prod\_name

from summAmount group by prod\_name)

select \* from

(select b.maxx || '/' || d.time\_id as "Max buys/day", b.prod\_name

from maxAmount b

inner join summAmount d

on d.sum1 = b.maxx and d.prod\_name = b.prod\_name order by maxx desc)

WHERE rownum < 21;

-----------------------------------------------------------------------------

10. Максимальное суммарное количество покупок за день для каждой категории продуктов. Отсортировать по убыванию количества. (Столбцы должны быть такими: "Макс за день", prod\_category). [Под первым столбцом подразумевается одно число].

with

qbd as (select count(prod\_category) as count\_q, prod\_category, truncated\_date

from (select TRUNC(sas.time\_id,'DD') as truncated\_date, Prods.prod\_category from SH.sales Sas

inner join SH.products Prods on Sas.prod\_id=Prods.prod\_id)

group by prod\_category, truncated\_date)

select max(count\_q), prod\_category from qbd group by prod\_category;

-------------------------------------------------------------------------

11. Создать таблицу с именем sales\_[имя пользователя в ОС]\_[Ваше имя]\_[Ваша фамилия], содержающую строки из таблицы sh.sales за пиковый месяц. (Т.е. месяц, за который получена максимальная выручка). Показать все поля таблицы в порядке возрастания дат.

CREATE TABLE sales\_User2\_Lianna\_Kuznetsova as

(SELECT \* FROM SH.sales

WHERE (SELECT month1

FROM ( SELECT TRUNC(time\_id, 'MM')AS month1 ,

sum(amount\_sold) AS sumsold

FROM SH.sales

GROUP BY TRUNC(time\_id, 'MM')

ORDER BY sumsold DESC)

WHERE ROWNUM <= 1) = TRUNC(time\_id, 'MM'));

-----------------------------------------------------------------------------

12. Для созданной в п.11 таблицы изменить значение поля time\_id на формат 'DD.MM.YYYY HH24:MI:SS' (см. NLS\_DATE\_FORMAT). Значение hh24:mm:ss должно выбираться случайным образом. Сохранить сделанные изменения. Показать все поля таблицы в порядке возрастания дат. time\_id=time\_id+dbms\_random.value;

ALTER TABLE SALES\_USER6\_Lianna\_Kuznetsova

ADD (TIME\_ID\_NEW CHAR(19));

UPDATE SALES\_USER6\_Lianna\_Kuznetsova

SET TIME\_ID\_NEW = TO\_CHAR (TIME\_ID + NUMTODSINTERVAL(MOD(ABS(DBMS\_RANDOM.RANDOM),24 \* 60 \* 60),'second'),'DD.MM.YYYY HH:MI:SS');

ALTER TABLE SALES\_USER6\_Lianna\_Kuznetsova

DROP COLUMN TIME\_ID;

ALTER TABLE SALES\_USER6\_ Lianna\_Kuznetsova

RENAME COLUMN TIME\_ID\_NEW TO TIME\_ID;

SELECT \* FROM SALES\_USER6 Lianna\_Kuznetsova;

alter session set nls\_date\_format = 'DD.MM.YYYY HH24:MI:SS';

update sales\_User14\_Kate\_Guledza

set time\_id=time\_id+dbms\_random.value;

SELECT \* FROM sales\_User14\_Kate\_Guledza;

------------------------------------------------------------------------------------------------------------------------------------------

13. Почасовую разбивку количества продаж для Вашей таблицы.

SELECT DATE(dt), COUNT(i) FROM leads WHERE lead = 'l' GROUP BY DATE(dt)

------------------------------------------------------------------------

14. Удалить созданную в п. 11 таблицу. Сохранить сделанные изменения.

Примечание: некоторые задания допускают неоднозначное толкование.

Если в задании явно не указаны какие-то нюансы, «додумайте» условие, как вам нравится.

В остальных случаях следует выводить в точности такой ответ, как требуется в задании.

|  |  |
| --- | --- |
| 2 | DROP TABLE sales\_User2\_Lianna\_Kuznetsova;  Query OK, 0 rows affected (0.01 sec) |