МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «Национальный исследовательский ядерный университет МИФИ»

(НИЯУ МИФИ)

Институт интеллектуальных кибернетических систем Кафедра Кибернетики

**Лабораторная работа №2**

**Курс: "Основы корпоративных СУБД и SQL-программирование"**

Выполнил студент группы Б16-511:

Меркулов А.В.

Проверил:

Саманчук В.Н.

Москва 2020

**Курс: "Основы корпоративных СУБД и SQL-программирование"**

ЛАБОРАТОРНАЯ РАБОТА №2

1. Выберите значения идентификатора заказа, даты заказа, идентификатора заказчика, даты отпуска для строк, в которых одновременно выполнены условия:
   * дата оплаты отсутствует;
   * дата отпуска принадлежит интервалу первых 9 месяцев 1994 года.
2. Выберите значения идентификатора заказчика, названия компании в строках, для которых выполняется условие:
   * в названии компании присутствует символьная подстрока **Town** или **town**
3. Выберите значения названия компании, город, штат, индекс города в строках, для которых одновременно выполнены два условия:
   * в названии компании встречается подстрока **Sports**;
   * значение штата равно **CA** или **FL**
4. Выберите наибольшее, наименьшее и среднее значения отпускной цены ( таблица **orders** )
5. Выберите значения идентификатора заказа, даты заказа, даты отпуска для строк, в которых одновременно выполнены условия:
   * дата заказа - понедельник и пятница;
   * дата заказа и дата отпуска принадлежат одному месяцу

( используйте **WEEKDAY, IN, MONTH** )

1. Выберите значения идентификатора заказчика, общую сумму отпускной цены для этих заказчиков. Отсортируйте результат выборки по убыванию общей суммы отпускной цены ( таблица orders ). Используйте **GROUP BY, SUM**.
2. Выберите значения идентификатора заказчика и суммарный вес его отправлений, если он превышает значение 30. Выведите данные по убыванию суммарного веса отправлений. Используйте **GROUP BY, HAVING, SUM, DESC**.
3. Выберите значения всех столбцов в таблице **customer**, для которых выполняется условие: - значение штата равно СА во временную таблицу, отсортировывая по возрастанию столбца «название компании». Выберите все строки во временной таблице.
4. Выберите значения идентификатора заказчика, названия компании, идентификатора заказа для всех заказчиков, разместивших заказы ( связь по **customer\_num** таблиц **customer** и **orders** ). Отсортируйте результат по возрастанию идентификатора заказа.
5. Выберите значения идентификатора заказа, идентификатора счета, описания товара, кода производителя и общую цену при условии, что идентификатор заказа равен 1004. Используйте таблицы **items** и **stock**.
6. В предыдущем примере ( 10 ) вместо значения код производителя выведите полное название производителя ( столбец **manu\_name** в таблице **manufact** )
7. Добавьте в таблицу **stock** строку со следующими значениями:
   * идентификатор товара = 11;
   * код производителя = **SMT**;
   * описание товара = **ski goggles**;
   * цена за единицу упаковки = 10.0
8. Проведите изменение значений в строках таблицы **stock**, для которых идентификатор товара равен 11 и код производителя равен **SMT**. Новые значения столбцов:
   * цена за единицу упаковки = 85.0;
   * упаковка = **case**;
   * описание упаковки = 10 **goggless/case**
9. Удалите строки в таблице **stock**, для которых одновременно выполняются оба условия:
   * идентификатор товара = 11;
   * код производителя = **SMT**
10. Выберите все коды производителей, названия производителей и сумму общей цены ( столбец **total\_price** в таблице **items** ) для его товаров. Отсортируйте результат по коду производителя. Используйте **SUM, GROUP BY, ORDER BY**, таблицы **manufact, items.**
11. Самообъединение таблицы. Выберите из таблицы **stock** все описания товаров ( **description** ) и все пары кодов производителей ( **manu\_code** ), производящих этот товар.
12. Выберите все идентификаторы заказчика, имена ( **lname** и **fname** ) для всех заказчиков, которые имеют более одного заказа ( больше одной строки со значением **customer\_num** в таблице **orders** ).

Вариант 1: с использованием ключевого слова **IN**. ( используйте **GROUP BY**, **HAVING, COUNT**, таблицы **customer** и **orders** ).

Вариант 2: выборка через временную таблицу , с последующей выборкой из нее с условием.

1. Выберите идентификатор заказчика, идентификатор заказа, дату заказа для всех заказов, не включающих в описании товара символьную подстроку **“ baseball** **gloves**”. Отсортируйте результат по идентификатору заказа ( **order\_num** .). Используйте **NOT EXISTS, ORDER BY**, таблицы **orders, items, stock**.
2. Используя подзапрос в **HAVING** структуре предложения **SELECT**, найдите все идентификаторы заказов с общей ценой, меньшей, чем средняя общая цена всех товаров для всех заказов. Используйте **HAVING, SUM, AVG, GROUP BY**, таблицу **items**.
3. Перепишите запрос при помощи **UNION** предложения:

**SELECT \* FROM stock**

**WHERE manu\_code = “HRO” OR stock\_num = 1**

1. Используя **UNION**, выведите информацию из таблицы **customer**, отсортированную по столбцу **city**. Разместите строки с городом **Redwood City** в начале вывода. Для вывода **sortkey** оформляется в виде констант «А» и «В» в предложениях **SELECT**. В результате выполнения должно получиться:

**Sortkey city company**

**A Redwood city AA Athletics**

**A Redwood city Kids Korner**

**.**

**.**

**.**

**B другой город**

**В другой город**

**Запросы:**

1) SELECT MERKULOV\_ORDERS.ORDER\_NUM, MERKULOV\_ORDERS.ORDER\_DATE, MERKULOV\_ORDERS.CUSTOMER\_NUM,MERKULOV\_ORDERS.SHIP\_DATE FROM MERKULOV\_ORDERS

WHERE PAID\_DATE IS NULL AND SHIP\_DATE BETWEEN '1/1/1994' AND '9/30/1994'

2)SELECT CUSTOMER\_NUM, COMPANY FROM MERKULOV\_CUSTOMER WHERE COMPANY LIKE '%Town%' OR COMPANY LIKE '%town%'

3) SELECT COMPANY, CITY, STATE,ZIPCODE FROM MERKULOV\_CUSTOMER WHERE COMPANY LIKE '%Sports%' AND (STATE='CA' OR STATE='FL')

4) SELECT MAX(SHIP\_CHARGE), MIN(SHIP\_CHARGE), AVG(SHIP\_CHARGE) FROM MERKULOV\_ORDERS

5)SELECT ORDER\_NUM, ORDER\_DATE, SHIP\_DATE FROM MERKULOV\_ORDERS WHERE EXTRACT(WEEKDAY FROM ORDER\_DATE) IN ( 1,5) AND EXTRACT(MONTH FROM ORDER\_DATE)=EXTRACT(MONTH FROM SHIP\_DATE)

6)SELECT customer\_num, sum(ship\_charge) sum\_ship\_charge FROM MERKULOV\_ORDERS

GROUP BY sum\_ship\_sharge ORDER BY customer\_num ;

7)SELECT customer\_num, sum(ship\_weight) sum FROM MERKULOV\_ORDERS

HAVING sum() >= 30

GROUP BY customer\_num

ORDER BY sum DESC

8)CREATE customer\_tmp

AS

SELECT \* FROM MERKULOV\_CUSTOMER

WHERE state = '—ј'

9)SELECT c.customer\_num, c.company, o.order\_num FROM MERKULOV\_CUSTOMER c INNER JOIN MERKULOV\_ORDERS o ON c.customer\_num = o.customer\_num

ORDER BY o.order\_num

10)SELECT ORDER\_NUM, ITEM\_NUM, DESCRIPTION,MERKULOV\_ITEMS.MANU\_CODE, TOTAL\_PRICE FROM MERKULOV\_ITEMS I, MERKULOV\_STOCK S

WHERE I.MANU\_CODE = S.MANU\_CODE AND I.STOCK\_NUM=S.stock\_num AND ORDER\_NUM=1004

11)SELECT ORDER\_NUM, ITEM\_NUM, DESCRIPTION,MANU\_NAME, TOTAL\_PRICE FROM MERKULOV\_ITEMS I, MERKULOV\_STOCK S, MERKULOV\_MANUFACT M WHERE I.MANU\_CODE = S.MANU\_CODE AND I.STOCK\_NUM=S.stock\_num AND M.MANU\_CODE=S.MANU\_CODE AND ORDER\_NUM=1004

12) INSERT INTO MERKULOV\_STOCK(STOCK\_NUM, MANU\_CODE,DESCRIPTION,UNIT\_PRICE) VALUES(11, 'SMT', 'ski goggles',10.0)

13)UPDATE MERKULOV\_STOCK

SET unit\_price = 85.0,

unit = 'case',

unit\_deser = '10 goggless/case'

WHERE

stock\_num = 11 AND

manu\_code = 'SMT'

14)DELETE FROM MERKULOV\_STOCK

WHERE

stock\_num = 11 AND

manu\_code = 'SMT'

15)SELECT m.manu\_code, m.manu\_name, sum(i.total\_price) total

FROM MERKULOV\_MANUFACT m INNER JOIN MERKULOV\_ITEMS i ON m.manu\_code = i.manu\_code

GROUP BY m.manu\_code ORDER BY m.manu\_code

16)SELECT S.DESCRIPTION, S.MANU\_CODE,C.DESCRIPTION, C.MANU\_CODE FROM MERKULOV\_STOCK S, MERKULOV\_STOCK C

WHERE S.STOCK\_NUM=C.STOCK\_NUM

17)SELECT o.customer\_num, c.fname, c.lname FROM MERKULOV\_CUSTOMER c INNER JOIN MERKULOV\_ORDERS o ON

c.customer\_num = o.customer\_num

HAVING COUNT(o.order\_num) > 1

GROUP BY o.customer\_num

18)SELECT o.customer\_num, o.order\_num, o.order\_date FROM MERKULOV\_ORDERS o INNER JOIN MERKULOV\_ITEMS i

ON o.order\_num = i.order\_num ### stock s ON i.stock\_num = s.stock\_num

WHERE NOT EXISTS (

SELECT 1 FROM stock s

WHERE i.stock\_num = s.stock\_num

AND

s.description LIKE '%baseball gloves%'

)

ORDER BY o.order\_num

19)SELECT order\_num FROM MERKULOV\_ITEMS

HAVING total\_price < ALL(SELECT AVG(total\_price) FROM MERKULOV\_IEMS)

GROUP BY order\_num, total\_price

20)SELECT \* FROM MERKULOV\_STOCK

WHERE manu\_code = 'HRO'

UNION

SELECT \* FROM MERKULOV\_STOCK

WHERE stock\_num = 1

21)SELECT A sortkey, \* from MERKULOV\_CUSTOMER

WHERE city = 'Redwood City'

UNION

SELECT B sortkey, \* from MERKULOV\_CUSTOMER

WHERE city != 'Redwood City'

ORDER BY sortkey, city