

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**)
6. 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**.
7. Выберите значения идентификатора заказчика и суммарный вес его отправок, если он превышает значение 30. Выведите данные по убыванию суммарного веса отправок. Используйте **GROUP BY, HAVING, SUM, DESC**.
8. Выберите значения всех столбцов в таблице **customer**, для которых выполняется условие: - значение штата равно **CA** во временную таблицу, отсортировывая по возрастанию столбца «название компании». Выберите все строки во временной таблице.
9. 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.
10. 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**.
 11. В предыдущем примере (10) вместо значения код производителя выведите полное название производителя (столбец **manu_name** в таблице **manufact**)
 12. Добавьте в таблицу **stock** строку со следующими значениями:
 - идентификатор товара = 11;
 - код производителя = **SMT**;
 - описание товара = **ski goggles**;
 - цена за единицу упаковки = 10.0
 13. 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**
 14. Удалите строки в таблице **stock**, для которых одновременно выполняются оба условия:
 - идентификатор товара = 11;
 - код производителя = **SMT**
 15. 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**.
 16. Самообъединение таблицы. Выберите из таблицы **stock** все описания товаров (**description**) и все пары кодов производителей (**manu_code**), производящих этот товар.
 17. 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.

18. 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**.
19. 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**.
20. Перепишите запрос при помощи **UNION** предложения:
SELECT * FROM stock
WHERE manu_code = “HRO” OR stock_num = 1
21. 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 «A» и «B» в предложениях **SELECT**. В результате выполнения должно получиться:

Sortkey	city	company
A	Redwood city	AA Athletics
A	Redwood city	Kids Korner
.		
.		
.		
B	другой город	
B	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_charge 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 = '—j'

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 goggles/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.DESCRPTION, S.MANU_CODE,C.DESCRPTION, 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