**-- Выбрать всех заказчиков чьи имена попали в алфавитный диапазон 'A' - 'G'**

-- Будем считать, что надо выбрать всех включая граничных

-- Подключемся в базе данных orders

USE orders;

-- Получить все столбцы

SELECT \*

-- Из таблицы customers

FROM customers

-- Где значение поля cname при лексикографическом сравнении больше либо равно A и меньше H

WHERE cname >= N'A' AND cname < N'H';

-- Другой вариант:

SELECT \* FROM customers

-- Где значение cname лексикографически между A и H

WHERE cname BETWEEN N'A' AND N'H';

-- Другой вариант:

SELECT \* FROM customers

-- Где значение cname совпадает с шаблоном "А%"

WHERE cname LIKE N'A%'

-- или значение cname совпадает с шаблоном "B%"

OR cname LIKE N'B%'

-- или значение cname совпадает с шаблоном "C%"

OR cname LIKE N'C%'

-- или значение cname совпадает с шаблоном "D%"

OR cname LIKE N'D%'

-- или значение cname совпадает с шаблоном "E%"

OR cname LIKE N'E%'

-- или значение cname совпадает с шаблоном "F%"

OR cname LIKE N'F%'

-- или значение cname совпадает с шаблоном "G%"

OR cname LIKE N'G%';

-- Знак "%" в шаблоне означает "ноль и более символов"

-- Другой вариант:

SELECT \* FROM customers

-- Где значение cname совпадает с шаблоном "[ABCDEFG]%"

WHERE cname LIKE N'[ABCDEFG]%';

-- Шаблон читается, как "Любой символ из списка (A, B, C, D, E, F, G) и ноль или более символов после него"

**-- Найти всех заказчиков чьи имена начинаются с 'G'**

-- Подключемся в базе данных orders

USE orders;

-- Получить все столбцы

SELECT \*

-- Из таблицы customers

FROM customers

-- Где значение поля cname соответствует шаблону "G%"

WHERE cname LIKE N'G%';

-- Шаблон читается как: "Символ G и ноль или более символов после него"

**-- Найти все записи в таблице Заказчиков с NULL-значениями в столбце city**

-- Подключемся в базе данных orders

USE orders;

-- Получить все столбцы

SELECT \*

-- Из таблицы customers

FROM customers

-- Где значение поля city составляет NULL

WHERE city IS NULL;

**-- Найти сумму заказов в таблице Заказов**

-- Подключемся в базе данных orders

USE orders;

-- Получить сумму столбцов amt

SELECT SUM(amt)

-- Из таблицы orders

FROM orders

-- Где значение поля amt не равно NULL (с NULL нельзя складывать)

WHERE amt IS NOT NULL;

**-- Найти наибольшую сумму приобретений (amt) полученную каждым продавцом**

-- Подключемся в базе данных orders

USE orders;

-- Получить столбец snum и максимальное значение из столбца amt (которому будет задано имя max\_amt)

SELECT snum, MAX(amt) as max\_amt

-- Из таблицы orders

FROM orders

-- Сгруппированное по значениям из столбца snum

GROUP BY snum;

**-- Найти наибольшую сумму приобретений (amt) получаемую каждым продавцом каждый день**

USE orders;

SELECT snum, odate, MAX(amt) as max\_amt

FROM orders

GROUP BY snum, odate

ORDER BY snum, odate;

**-- Найти максимальную стоимость приобретений (amt) свыше $3000.00**

-- Подключемся в базе данных orders

USE orders;

-- Получить наибольшее значение из столбца amt

SELECT MAX(amt)

-- Из таблицы orders

FROM orders

-- Где поле amt больше 3000

WHERE amt > 3000.000;

**-- Найти наибольшие заказы продавцов Serres и Rifkin**

-- Подключемся в базе данных orders

USE orders;

-- Создаём "Обобщённое Табличое Выражение", грубо говоря временную таблицу, с именем sels

WITH sels AS (

-- Содержимым sels является результат выполнения подзапроса:

-- Получить столбцы snum и sname

SELECT snum, sname

-- Из salespeople

FROM salespeople

-- Где sname равно "Serres" или "Rifkin"

WHERE sname = N'Serres' OR sname = N'Rifkin'

)

-- Получить столбцы snum и sname из sels и максимальное значение столбца amt из orders, которому будет дано имя max\_amt

SELECT sels.snum, sels.sname, MAX(orders.amt) as max\_amt

-- Из таблиц sels и orders

FROM sels INNER JOIN orders

-- При этом, поля snum в sels и orders должны быть равны

ON sels.snum = orders.snum

-- Сгруппированное по паре snum и sname из sels

GROUP BY sels.snum, sels.sname;

**-- Выбрать данные из таблицы Продавцов, упорядочив в порядке убывания к наименьшему значению комиссионных** **(comm)**

-- Подключемся в базе данных orders

USE orders;

-- Получить все столбцы

SELECT \*

-- Из таблицы salespeople

FROM salespeople

-- Отсортировав по столбцу comm

ORDER BY comm

-- В порядке убывания

DESC;

**-- Подсчитать количество заказов каждого из продавцов, и вывести результаты в убывающем порядке**

-- Подключемся в базе данных orders

USE orders;

-- Создаём "Обобщённое Табличое Выражение", грубо говоря временную таблицу, с именем sels

WITH sels AS (

-- Содержимым sels является результат выполнения подзапроса:

-- Получить столбцы snum и sname

SELECT snum, sname

-- Из salespeople

FROM salespeople

)

-- Получить столбцы snum и sname из sels и количество записей, которому будет дано имя count

SELECT sels.snum, sels.sname, COUNT(\*) as count

-- Из таблиц sels и orders

FROM sels INNER JOIN orders

-- При этом, поля snum в sels и orders должны быть равны

ON sels.snum = orders.snum

-- Сгруппированное по паре snum и sname из sels

GROUP BY sels.snum, sels.sname

-- Отсортированное по полю count в порядке убывания

ORDER BY count DESC;

**-- Вывести пары продавцов - заказчиков живущих в одном городе**

-- Подключемся в базе данных orders

USE orders;

-- Получить столбец sname из таблицы salespeople и столбец cname из таблицы customers

SELECT salespeople.sname, customers.cname

-- Из таблиц customers и salespeople (утсановление внутреннего соединения)

FROM customers INNER JOIN salespeople

-- При этом city из customers должно быть равно city из salespeople

ON customers.city = salespeople.city;

**-- Вывести для каждого заказчика имя продавца который его обслуживает**

-- Подключемся в базе данных orders

USE orders;

-- Получить столбец cname из таблицы customers и столбец sname из таблицы salespeople

SELECT customers.cname, salespeople.sname

-- Из таблиц customers и salespeople (утсановление внутреннего соединения)

FROM customers INNER JOIN salespeople

-- При этом snum из customers должно быть равно snum из salespeople

ON customers.snum = salespeople.snum;

**-- Вывести все пары заказчиков имеющих один и тот же рейтинг (rating)**

-- Подключемся в базе данных orders

USE orders;

SELECT one.cnum, one.cname, two.cnum, two.cname, one.rating

FROM customers AS one,

customers AS two

WHERE one.rating = two.rating

AND one.cnum < two.cnum;

**-- Вывести все пары заказчиков: один с рейтингом (rating) 100, второй 200**

-- Подключемся в базе данных orders

USE orders;

WITH one AS (

SELECT cnum, cname, rating

FROM customers

WHERE rating = 100

)

SELECT one.\*, customers.cnum, customers.cname, customers.rating

FROM one INNER JOIN customers

ON customers.rating = 200;

**-- Вывести все тройки заказчиков: один с рейтингом (rating) 100, второй 200, третий 300**

-- Подключемся в базе данных orders

USE orders;

WITH one AS (

SELECT cnum, cname, rating

FROM customers

WHERE rating = 100

), two AS (

SELECT cnum, cname, rating

FROM customers

WHERE rating = 200

), three AS (

SELECT cnum, cname, rating

FROM customers

WHERE rating = 300

)

SELECT \*

FROM one, two, three;

**-- Вывести из таблицы Заказов все заказы продавца Motika**

-- Подключемся в базе данных orders

USE orders;

-- Получить все столбцы таблицы orders

SELECT orders.\*

-- Из таблиц orders и salespeople (с установлением внутреннего соединения)

FROM orders INNER JOIN salespeople

-- При этом snum из orders должно быть равно snum из salespeople

ON orders.snum = salespeople.snum

-- Где sname из salespeople равно "Motika"

WHERE salespeople.sname = N'Motika';

**-- Вывести все заказы продавца, который обслуживает Hoffman'а.**

-- Подключемся в базе данных orders

USE orders;

-- Получить все столбцы из orders

SELECT orders.\*

-- Из таблиц customers и orders (с установлением внутреннего соединения)

FROM customers INNER JOIN orders

-- При этом поля snum должны быть равны друг другу

ON orders.snum = customers.snum

-- Где поле cname из таблицы customers равно "Hoffman"

WHERE customers.cname = N'Hoffman';

**-- Вывести все заказы имеющие сумму приобретений (amt) выше средней на 4-е Октября**

-- Подключемся в базе данных orders

USE orders;

-- Не совсем понятно, что конкретно имеется ввиду в задании, поэтому решаем три задачи:

-- А) Берём среднее за 4 октября и выводим всё, что больше этого числа за все дни

-- Б) Берём среднее за все дни и выводим всё, что больше этого числа из заказов сделанных 4 октября

-- В) Берём среднее за 4 октября и выводим всё, что больше этого числа из заказов сделанных 4 октября

-- В принципе отличия между всеми этими тремя задачами минимальные

-- ===========================================================================

-- А) Берём среднее за 4 октября и выводим всё, что больше этого числа за все дни

-- Создаём "Обобщённое Табличое Выражение", грубо говоря временную таблицу, с именем avg

WITH avg as (

-- Содержимым avg является результат выполнения подзапроса:

-- Получить среднее значение столбца amt и записать под именем amt

SELECT AVG(amt) as amt

-- Из таблицы orders

FROM orders

-- Где поле odate соответствует 10 октября

WHERE orders.odate = '10/04/1990'

)

-- Получить все записи из orders

SELECT orders.\*

-- Из таблиц orders и avg ("временная таблица") (с использованием внутреннего связывания)

FROM orders INNER JOIN avg

-- При этом поле amt из orders должно быть больше поля amt из avg

ON orders.amt > avg.amt;

-- ===========================================================================

-- Б) Берём среднее за все дни и выводим всё, что больше этого числа из заказов сделанных 4 октября

-- Создаём "Обобщённое Табличое Выражение", грубо говоря временную таблицу, с именем avg

WITH avg as (

-- Содержимым avg является результат выполнения подзапроса:

-- Получить среднее значение столбца amt и записать под именем amt

SELECT AVG(amt) as amt

-- Из таблицы orders

FROM orders

)

-- Получить все записи из orders

SELECT orders.\*

-- Из таблиц orders и avg ("временная таблица") (с использованием внутреннего связывания)

FROM orders INNER JOIN avg

-- При этом поле amt из orders должно быть больше поля amt из avg

ON orders.amt > avg.amt

-- Где поле odate из orders соответствует 10 октября

WHERE orders.odate = '10/04/1990';

-- ===========================================================================

-- В) Берём среднее за 4 октября и выводим всё, что больше этого числа из заказов сделанных 4 октября

-- Создаём "Обобщённое Табличое Выражение", грубо говоря временную таблицу, с именем avg

WITH avg as (

-- Содержимым avg является результат выполнения подзапроса:

-- Получить среднее значение столбца amt и записать под именем amt

SELECT AVG(amt) as amt

-- Из таблицы orders

FROM orders

-- Где поле odate соответствует 10 октября

WHERE orders.odate = '10/04/1990'

)

-- Получить все записи из orders

SELECT orders.\*

-- Из таблиц orders и avg ("временная таблица") (с использованием внутреннего связывания)

FROM orders INNER JOIN avg

-- При этом поле amt из orders должно быть больше поля amt из avg

ON orders.amt > avg.amt

-- Где поле odate из orders соответствует 10 октября

WHERE orders.odate = '10/04/1990';

**-- Найти среднее значение комиссионных продавцов в Лондоне**

-- Подключемся в базе данных orders

USE orders;

-- Получить среднее значение столбца comm

SELECT AVG(comm)

-- Из таблицы salespeople

FROM salespeople

-- Где поле city равно "London"

WHERE city = N'London';

**-- Вывести все строки таблицы Заказов для продавцов в Лондоне**

-- Подключемся в базе данных orders

USE orders;

-- Получить все столбцы из orders

SELECT orders.\*

-- Из таблиц salespeople и orders (с установлением внутреннего соединения)

FROM salespeople INNER JOIN orders

-- При этом поля snum должны быть равны друг другу

ON orders.snum = salespeople.snum

-- Где поле city из таблицы salespeople равно "London"

WHERE salespeople.city = N'London';

**-- Вывести комиссионные всех продавцов обслуживающих заказчиков в Лондоне**

-- Подключемся в базе данных orders

USE orders;

-- Получить столбцы snum, sname и comm из salespeople

-- Но выбрать только РАЗЛИЧАЮЩИЕСЯ записи (дубли отбрасываются)

SELECT DISTINCT salespeople.snum, salespeople.sname, salespeople.comm

-- Из таблиц salespeople и customers (с установлением внутреннего соединения)

FROM salespeople INNER JOIN customers

-- При этом поля snum должны быть равны друг другу

ON customers.snum = salespeople.snum

-- Где поле city из таблицы customers равно "London"

WHERE customers.city = N'London';

**-- Подсчитать количество заказчиков в San Jose с оценками (rating) выше среднего**

-- Подключемся в базе данных orders

USE orders;

-- Создаём "Обобщённое Табличое Выражение", грубо говоря временную таблицу, с именем avg

WITH avg as (

-- Содержимым avg является результат выполнения подзапроса:

-- Получить среднее значение столбца rating и записать под именем rank

SELECT AVG(rating) as rank

-- Из таблицы customers

FROM customers

)

-- Получить количество записей

SELECT COUNT(\*)

-- Из таблиц customers и avg ("временная таблица") (с использованием внутреннего связывания)

FROM customers INNER JOIN avg

-- При этом поле rating из customers должно быть больше поля rank из avg

ON customers.rating > avg.rank

-- Где поле city из customers равно "San Jose"

WHERE customers.city = N'San Jose';

**-- Вывести имена и номера всех продавцов которые имеют более одного заказчика**

-- Подключемся в базе данных orders

USE orders;

-- Создаём "Обобщённое Табличое Выражение", грубо говоря временную таблицу, с именем clients

WITH clients as (

-- Содержимым avg является результат выполнения подзапроса:

-- Получить столбец snum и количество записей под именем count

SELECT snum, COUNT(\*) as count

-- Из таблицы customers

FROM customers

-- Сгруппированное по столбцу snum

GROUP BY snum

)

-- Получить количество записей

SELECT salespeople.snum, salespeople.sname

-- Из таблиц salespeople и clients ("временная таблица") (с использованием внутреннего связывания)

FROM salespeople INNER JOIN clients

-- При этом поля snum из salespeople и clients должны быть равны

ON clients.snum = salespeople.snum

-- Где поле count из clients больше 1

WHERE clients.count > 1;

**-- Найти все пары заказчиков обслуживаемых одним продавцом.**

USE orders;

SELECT \* FROM customers;

SELECT one.snum, one.cnum, one.cname, two.cnum, two.cname

FROM customers AS one,

customers AS two

WHERE one.snum = two.snum

AND one.cnum < two.cnum;

**-- Найти всех заказчиков размещенных в городах где продавец Serres ( snum 1002 ) имеет заказчиков.**

USE orders;

WITH towns AS (

SELECT city

FROM customers

WHERE snum = 1002

GROUP BY city

)

SELECT customers.\*

FROM towns INNER JOIN customers

ON towns.city = customers.city;

**-- Найти всех продавцов, проживающих в Барселоне или Лондоне**

-- Подключемся в базе данных orders

USE orders;

-- Получить все столбцы

SELECT \*

-- Из таблицы salespeople

FROM salespeople

-- Где city равно "Barcelona" или "London"

WHERE city = N'Barcelona' OR city = N'London';

**-- Выбрать заказы в которых или сумма заказа > 2000.00, или сделанные 03-10-1990 продавцами с номерами > 1002**

-- Подключемся в базе данных orders

USE orders;

-- Получить все столбцы

SELECT \*

-- Из таблицы orders

FROM orders

-- Где amt > 2000

WHERE amt > 2000.000

--Или odate равно 3 октября 1990 года и snum > 1002

OR odate = '10/03/1990' AND snum > 1002;

**-- Выбрать заказы для которых НЕ выполняются следующие условия:**

**-- или сумма заказа > 2000.00, или сделаны 03-10-1990 продавцами с номерами > 1002**

-- Подключемся в базе данных orders

USE orders;

-- Получить все столбцы

SELECT \*

-- Из таблицы orders

FROM orders

-- Где не верно условие

WHERE NOT (

-- amt > 2000

amt > 2000.000

--Или odate равно 3 октября 1990 года и snum > 1002

OR odate = '10/03/1990' AND snum > 1002

);

-- Другой способ

-- Всё то же самое, просто мы "раскрыли скобки" у логического выражения

SELECT \*

FROM orders

WHERE amt <= 2000.000 AND (odate != '10/03/1990' OR snum <= 1002);

-- Другой способ

-- Опять то же самое, но ещё больше раскрываем скобки

SELECT \*

FROM orders

WHERE amt <= 2000.000 AND odate != '10/03/1990'

OR amt <= 2000.000 AND snum <= 1002;

**-- Выбрать заказчиков, сделавших заказы 3 октября 1990 года**

-- Подключемся в базе данных orders

USE orders;

-- Получить столбец cname таблицы orders

SELECT customers.cname

-- Из таблиц orders и customers (с установлением внутреннего соединения)

FROM orders INNER JOIN customers

-- При этом cnum из orders должно быть равно cnum из customers

ON orders.cnum = customers.cnum

-- Где odate из orders равно 3 октября 1990 года

WHERE orders.odate = '10/03/1990';

**-- Найти продавцов, заказчики которых имеют более чем один заказ.**

-- Подключемся в базе данных orders

USE orders;

WITH ords AS (

SELECT cnum, COUNT(\*) as count

FROM orders

GROUP BY cnum

), custs AS (

SELECT customers.snum

FROM customers INNER JOIN ords

ON ords.cnum = customers.cnum

WHERE ords.count > 1

)

SELECT salespeople.\*

FROM salespeople INNER JOIN custs

ON salespeople.snum = custs.snum;