Базы данных и SQL. Обучение в записи

Урок 8. Семинар: SQL – работа с несколькими таблицами

Ссылка на репозиторий, с выполненным Д/3: https://github.com/olgashenkel/Databases-and-SQL

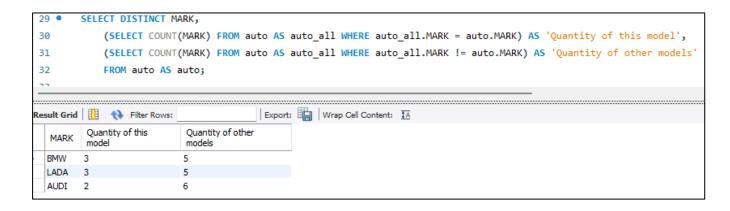
ЧАСТЬ І.

Домашнее задание Табличка: https://drive.google.com/file/d/1PQn576YVakvlWrIgIjSP9YEf5id4cqYs/view?usp=sharin 1.Вывести на экран, сколько машин каждого цвета для машин марок BMW и LADA 2.Вывести на экран марку авто(количество) и количество авто не этой марки. 100 машин, их них 20 - BMW и 80 машин другой марки , AUDI - 30 и 70 машин другой марки, LADA - 15, 85 авто другой марки Задание №3. Даны 2 таблицы, созданные следующим образом: create table test_a (id INT, test varchar(10)); create table test_b (id INT); 69 Домашнее задание insert into test_a(id, test) values (10, 'A'), (20, 'A'), (30, 'F'), (40, 'D'), (50, 'C'); insert into test_b(id) values (10),(30),(50);Напишите запрос, который вернет строки из таблицы test_a, id которых нет в таблице test_b, HE используя ключевого слова NOT. 63

1. Ход выполнения задания 1:

```
USE seminar 4;
 1 •
 2
       DROP TABLE IF EXISTS AUTO;
 3 •
       CREATE TABLE AUTO
4 •
    ⊖ (
5
           REGNUM VARCHAR(10) PRIMARY KEY,
6
7
           MARK VARCHAR(10),
           COLOR VARCHAR(15),
8
9
           RELEASEDT DATE,
           PHONENUM VARCHAR(15)
10
11
       );
12
13 •
       INSERT INTO AUTO (REGNUM, MARK, COLOR, RELEASEDT, PHONENUM)
14
       (111114, 'LADA', 'КРАСНЫЙ', date'2008-01-01', '9152222221'),
15
16
       (111116, 'BMW', 'СИНИЙ', date'2015-01-01', '9173333334'),
17
       (111121, 'AUDI', 'СИНИЙ', date'2009-01-01', '9173333332'),
       (111122, 'AUDI', 'СИНИЙ', date'2011-01-01', '9213333336'),
18
19
       (111113, 'BMW', 'ЗЕЛЕНЫЙ', date'2007-01-01', '9214444444'),
       (111126, 'LADA', 'ЗЕЛЕНЫЙ', date'2005-01-01', null),
20
       (111117, 'ВМW', 'СИНИЙ', date'2005-01-01', null),
21
       (111119, 'LADA', 'СИНИЙ', date'2017-01-01', 9213333331);
22
23
24 •
       SELECT MARK, COLOR, COUNT(COLOR) AS COUNT FROM auto
25
           WHERE MARK IN ('BMW', 'LADA')
       GROUP BY MARK, COLOR
26
27
       ORDER BY MARK;
esult Grid 🔢 🚷 Filter Rows:
                                        Export: Wrap Cell Content: IA
 MARK COLOR
                  COUNT
        ЗЕЛЕНЫЙ
 BMW
                 1
 BMW
        СИНИЙ
                 2
 LADA
        ЗЕЛЕНЫЙ
                 1
 LADA
        КРАСНЫЙ
                1
 LADA
        СИНИЙ
                 1
```

2. Ход выполнения задания 2:



3. Ход выполнения задания 3:

```
CREATE TABLE test_a
           (id INT, test VARCHAR(10));
35
36
       CREATE TABLE test_b
37 •
38
          (id INT);
39
40 •
       INSERT INTO test_a (id, test)
41
       VALUES
42
       (10, 'A'),
43
      (20, 'A'),
44
       (30, 'F'),
       (40, 'D'),
45
       (50, 'C');
46
47
48
      INSERT INTO test_b (id)
49
       VALUES
50
       (10),
51
       (30),
52
       (50);
54 • SELECT * FROM test_a
55
           NATURAL LEFT JOIN test_b
           WHERE test_b.id IS NULL;
Result Grid 🔢 🚷 Filter Rows:
                                     Export
       test
  20
       Α
 40 D
```

ЧАСТЬ II.

1. Ход выполнения задания 0.

category_id INT PRIMARY KEY,

Запрос на создание начальной таблицы, выполнять задания на основе этих данных

Вы можете воспользоваться заготовкой fiddle https://dbfiddle.uk/oqvodW9a Или использовать код ниже для создания таблиц

```
USE seminar_4;
-- Создание таблицы Customers
CREATE TABLE Customers (
customer id INT PRIMARY KEY,
customer_name VARCHAR(255)
);
-- Создание таблицы Orders
CREATE TABLE Orders (
order_id INT PRIMARY KEY,
customer_id INT,
order_date DATE,
total_amount DECIMAL(10, 2),
shipper_id INT,
FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
);
-- Создание таблицы Shippers
CREATE TABLE Shippers (
shipper id INT PRIMARY KEY,
shipper_name VARCHAR(255)
);
-- Создание таблицы Products
CREATE TABLE Products (
product_id INT PRIMARY KEY,
product_name VARCHAR(255),
category id INT,
author VARCHAR(255),
price DECIMAL(10, 2)
);
-- Создание таблицы OrderDetails
CREATE TABLE OrderDetails (
order detail id INT PRIMARY KEY,
order_id INT,
product_id INT,
quantity INT,
FOREIGN KEY (order_id) REFERENCES Orders(order_id),
FOREIGN KEY (product_id) REFERENCES Products(product_id)
);
-- Создание таблицы Categories
CREATE TABLE Categories (
```

```
category_name VARCHAR(255)
);
-- Наполнение таблиц данными
INSERT INTO Customers (customer_id, customer_name)
VALUES
(1, 'Иван Иванов'), (2, 'Мария Смирнова'), (3, 'Алексей Попов'), (4,
'Наталья Кузнецова'), (5, 'Дмитрий Васильев'),
(6, 'Ольга Петрова'), (7, 'Андрей Сидоров'), (8, 'Елена Алексеева'),
(9, 'Сергей Морозов'), (10, 'Ирина Фёдорова'),
(11, 'Андрей Иванов'), (12, 'Екатерина Мартынова');
INSERT INTO Shippers (shipper_id, shipper_name)
VALUES
(1, 'СДЕК'), (2, 'Почта России'), (3, 'ПЭК');
INSERT INTO Categories (category_id, category_name)
VALUES
(1, 'Художественная литература'), (2, 'Наука'), (3, 'Мистика');
INSERT INTO Products (product_id, product_name, category_id, author, price)
VALUES
(1, '1984', 1, 'Джордж Оруэлл', 250),
(2, 'Убить пересмешника', 1, 'Харпер Ли', 300),
(3, 'Великий Гэтсби', 1, 'Фрэнсис Скотт Фицджеральд', 200),
(4, 'Краткая история времени', 2, 'Стивен Хокинг', 320),
(5, 'Собака Баскервилей', 3, 'Артур Конан Дойл', 350),
(6, 'Моби Дик', 1, 'Герман Мелвилл', 400),
(7, 'Скотный двор', 1, 'Джордж Оруэлл', 220),
(8, 'Похвала Каталонии', 1, 'Джордж Оруэлл', 180),
(9, 'Дневник Анны Франк', 1, 'Анна Франк', 300),
(10, 'Краткая история времени', 2, 'Стивен Хокинг', 320);
-- Добавление заказов и деталей заказов с реалистичным распределением
INSERT INTO Orders (order_id, customer_id, order_date, total_amount, shipper_id)
VALUES
(1, 1, '2023-01-10', 750, 1),
(2, 3, '2023-01-12', 820, 2),
(3, 2, '2023-01-15', 600, 3),
(4, 4, '2023-02-01', 670, 1),
(5, 6, '2023-02-05', 550, 2),
(6, 3, '2023-02-10', 400, 3),
(7, 7, '2023-03-01', 320, 1),
(8, 8, '2023-03-05', 500, 2),
(9, 1, '2023-03-10', 270, 3),
(10, 5, '2023-03-15', 350, 1),
(11, 4, '2023-03-20', 420, 2),
(12, 9, '2023-04-01', 300, 3),
(13, 5, '2023-04-05', 220, 1),
(14, 6, '2023-04-10', 500, 2),
(15, 8, '2023-04-15', 450, 3),
(16, 10, '2023-05-01', 350, 1),
(17, 4, '2023-05-05', 250, 2),
(18, 9, '2023-05-10', 300, 3),
(19, 7, '2023-05-15', 400, 1);
```

```
INSERT INTO OrderDetails (order_detail_id, order_id, product_id, quantity) VALUES
(1, 1, 1, 1), (2, 1, 2, 1), (3, 1, 4, 1),
(4, 2, 3, 1), (5, 2, 5, 1), (6, 2, 7, 1),
(7, 3, 6, 1), (8, 3, 9, 1),
(9, 4, 1, 1), (10, 4, 8, 1),
(11, 5, 2, 1), (12, 5, 5, 1),
(13, 6, 3, 1),
(14, 7, 4, 1), (15, 7, 9, 1),
(16, 8, 10, 1),
(17, 9, 5, 1), (18, 9, 6, 1),
(19, 10, 7, 1);
```

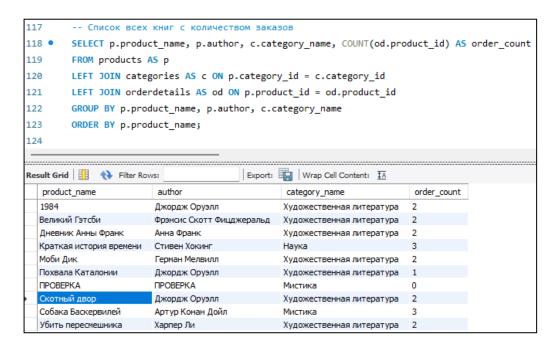
2. Ход выполнения задания 1.

Список всех книг с количеством заказов

Получите список всех книг вместе с именем автора, категорией и количеством заказов каждой книги.

В результате запроса будут столбцы:

- product name: название книги
- author: имя автора
- category name: название категории
- order count: количество заказов этой книги

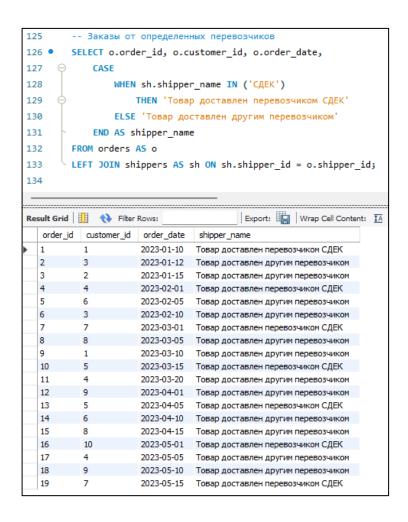


3. Ход выполнения задания 2.

Заказы от определенных перевозчиков

Получите список всех заказов, разделив их на те, которые были доставлены перевозчиком 'СДЕК', и те, которые были доставлены другими перевозчиками.

- order_id: идентификатор заказа
- customer_id: идентификатор клиента
- order date: дата заказа
- shipper name: имя перевозчика

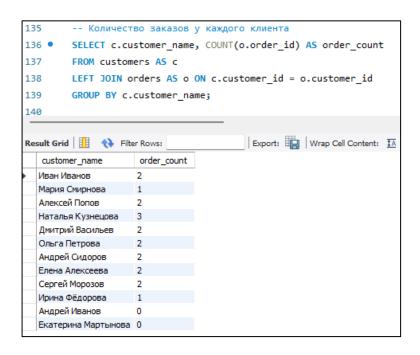


4. Ход выполнения задания 3.

Количество заказов у каждого клиента

Получите список всех клиентов и количество их заказов. Включите клиентов, которые не делали заказов.

- customer name: имя клиента
- order count: количество заказов



5. Ход выполнения задания 4.

Клиенты без заказов

Получите список всех клиентов, которые ничего не заказывали. В результате запроса будет один столбец:

– customer name: имя клиента

6. Ход выполнения задания 5.

Заказы с высокой стоимостью

Создайте новую таблицу HighValueOrders, которая будет содержать заказы на сумму более 500. Включите идентификатор заказа, идентификатор клиента и сумму заказа.

- order id: идентификатор заказа
- customer id: идентификатор клиента
- total amount: сумма заказа

```
-- Заказы с высокой стоимостью
        DROP TABLE IF EXISTS HighValueOrders;
149
150 •
       CREATE TABLE HighValueOrders
151
152
            HighVO_id INT PRIMARY KEY AUTO_INCREMENT NOT NULL,
            order_id INT,
            customer_id INT,
154
           total amount DECIMAL(10,2),
155
            FOREIGN KEY (order_id) REFERENCES orders(order_id),
            FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
157
158
        );
159
        INSERT INTO HighValueOrders (order_id, customer_id, total_amount)
160 •
161
            SELECT order_id, customer_id, total_amount
162
            FROM orders
163
               WHERE total_amount > 500;
        SELECT * FROM HighValueOrders;
Edit: 🚄 🖶 🖶 Export/Import: 📳 🚡
   HighVO_id order_id customer_id total_amount
                              750.00
           1
                    1
  2
           2
                   3
                              820.00
                              600.00
           4
                   4
                              670.00
                              550.00
```

7. Ход выполнения задания 6.

Анализ продаж книг по категориям

Получите список всех категорий книг с суммой продаж, средней ценой книги, минимальной и максимальной ценой книги, а также количеством уникальных заказов.

Отсортируйте по сумме продаж в порядке убывания и ограничьте результат первыми 5 строками.

- category name: название категории
- total sales: общая сумма продаж
- avg price: средняя цена книги
- min price: минимальная цена книги
- max price: максимальная цена книги
- unique orders: количество уникальных заказов

```
173
       -- Анализ продаж книг по категориям
      SELECT
174 •
175
         c.category_name,
         SUM(od.quantity * p.price) AS total_sales,
176
         ROUND(AVG(p.price),2) AS avg_price,
177
         MIN(p.price) AS min_price,
178
          MAX(p.price) AS max_price,
179
           COUNT(DISTINCT od.order_id) AS unique_orders
180
      FROM categories AS c
181
      LEFT JOIN
182
183
           products AS p ON p.category_id = c.category_id
      LEFT JOIN
184
185
           orderdetails AS od ON od.product_id = p.product_id
       GROUP BY c.category_name
186
187
       ORDER BY total_sales DESC
       LIMIT 5;
                                    Export: Wrap Cell Content: IA
total sales avg_price min_price max_price unique_orders
  category_name
                                 270.77
                                          180.00
                                                  400.00
  Художественная литература 3520.00
  Мистика
                1050.00 262.50 0.00 350.00 3
  Наука
                        960.00 320.00
                                         320.00 320.00
```