

Базы данных и SQL. Обучение в записи

Урок 8. Семинар: SQL – работа с несколькими таблицами

Ссылка на репозиторий, с выполненным Д/З:

<https://github.com/olgashenkel/Databases-and-SQL>

ЧАСТЬ I.

Домашнее задание

Табличка:

<https://drive.google.com/file/d/1PQn576YVakvWIrIjSP9YEf5id4cqYs/view?usp=sharing>

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);
```



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, НЕ используя ключевого слова NOT.



1. Ход выполнения задания 1:

Вывести на экран сколько машин каждого цвета для марок BMW и LADA

```
1 • USE seminar_4;
2
3 • DROP TABLE IF EXISTS AUTO;
4 • CREATE TABLE AUTO
5 (
6     REGNUM VARCHAR(10) PRIMARY KEY,
7     MARK VARCHAR(10),
8     COLOR VARCHAR(15),
9     RELEASEDT DATE,
10    PHONENUM VARCHAR(15)
11 );
12
13 • INSERT INTO AUTO (REGNUM, MARK, COLOR, RELEASEDT, PHONENUM )
14 VALUES
15 (111114, 'LADA', 'КРАСНЫЙ', date'2008-01-01', '9152222221'),
16 (111116, 'BMW', 'СИНИЙ', date'2015-01-01', '9173333334'),
17 (111121, 'AUDI', 'СИНИЙ', date'2009-01-01', '9173333332'),
18 (111122, 'AUDI', 'СИНИЙ', date'2011-01-01', '9213333336'),
19 (111113, 'BMW', 'ЗЕЛЕНый', date'2007-01-01', '9214444444'),
20 (111126, 'LADA', 'ЗЕЛЕНый', date'2005-01-01', null),
21 (111117, 'BMW', 'СИНИЙ', date'2005-01-01', null),
22 (111119, 'LADA', 'СИНИЙ', date'2017-01-01', 9213333331);
23
24 • SELECT MARK, COLOR, COUNT(COLOR) AS COUNT FROM auto
25     WHERE MARK IN ('BMW', 'LADA')
26     GROUP BY MARK, COLOR
27     ORDER BY MARK;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

MARK	COLOR	COUNT
BMW	ЗЕЛЕНый	1
BMW	СИНИЙ	2
LADA	ЗЕЛЕНый	1
LADA	КРАСНЫЙ	1
LADA	СИНИЙ	1

2. Ход выполнения задания 2:

Вывести на экран марку авто (количество) и количество авто не этой марки.

Например: 100 машин, из них 20 – BMW и 80 машин другой марки; AUDI – 30 и 70 машин другой марки, LADA – 15 и 85 авто другой марки

```
29 • SELECT DISTINCT MARK,
30     (SELECT COUNT(MARK) FROM auto AS auto_all WHERE auto_all.MARK = auto.MARK) AS 'Quantity of this model',
31     (SELECT COUNT(MARK) FROM auto AS auto_all WHERE auto_all.MARK != auto.MARK) AS 'Quantity of other models'
32 FROM auto AS auto;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

MARK	Quantity of this model	Quantity of other models
BMW	3	5
LADA	3	5
AUDI	2	6

3. *Ход выполнения задания 3:*

Даны 2 таблицы, созданные следующим образом:

create table test_a (id number, data varchar2(1));

create table test_b (id number);

insert into test_a(id, data) 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, НЕ используя ключевого слова NOT.

```
34 • CREATE TABLE test_a
35     (id INT, test VARCHAR(10));
36
37 • CREATE TABLE test_b
38     (id INT);
39
40 • INSERT INTO test_a (id, test)
41     VALUES
42     (10, 'A'),
43     (20, 'A'),
44     (30, 'F'),
45     (40, 'D'),
46     (50, 'C');
47
48 • INSERT INTO test_b (id)
49     VALUES
50     (10),
51     (30),
52     (50);
53
54 • SELECT * FROM test_a
55     NATURAL LEFT JOIN test_b
56     WHERE test_b.id IS NULL;
```

Result Grid |  Filter Rows: | Export

	id	test
	20	A
	40	D

ЧАСТЬ II.

1. Ход выполнения задания 0.

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 таблицы

```
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_id INT PRIMARY KEY,
```

```
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 Баскервиль', 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

```

117 -- Список всех книг с количеством заказов
118 • SELECT p.product_name, p.author, c.category_name, COUNT(od.product_id) AS order_count
119 FROM products AS p
120 LEFT JOIN categories AS c ON p.category_id = c.category_id
121 LEFT JOIN orderdetails AS od ON p.product_id = od.product_id
122 GROUP BY p.product_name, p.author, c.category_name
123 ORDER BY p.product_name;
124

```

product_name	author	category_name	order_count
1984	Джордж Оруэлл	Художественная литература	2
Великий Гэтсби	Фрэнсис Скотт Фицджеральд	Художественная литература	2
Дневник Анны Франк	Анна Франк	Художественная литература	2
Краткая история времени	Стивен Хокинг	Наука	3
Моби Дик	Герман Мелвилл	Художественная литература	2
Похвала Каталонии	Джордж Оруэлл	Художественная литература	1
ПРОВЕРКА	ПРОВЕРКА	Мистика	0
Скотный двор	Джордж Оруэлл	Художественная литература	2
Собака Баскервилей	Артур Конан Дойл	Мистика	3
Убить пересмешника	Харпер Ли	Художественная литература	2

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.

В результате запроса будут столбцы:

- *order_id*: идентификатор заказа
- *customer_id*: идентификатор клиента
- *order_date*: дата заказа
- *shipper_name*: имя перевозчика

125	-- Заказы от определенных перевозчиков
126	• SELECT o.order_id, o.customer_id, o.order_date,
127	CASE
128	WHEN sh.shipper_name IN ('СДЕК')
129	THEN 'Товар доставлен перевозчиком СДЕК'
130	ELSE 'Товар доставлен другим перевозчиком'
131	END AS shipper_name
132	FROM orders AS o
133	LEFT JOIN shippers AS sh ON sh.shipper_id = o.shipper_id;
134	

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
order_id	customer_id	order_date	shipper_name
1	1	2023-01-10	Товар доставлен перевозчиком СДЕК
2	3	2023-01-12	Товар доставлен другим перевозчиком
3	2	2023-01-15	Товар доставлен другим перевозчиком
4	4	2023-02-01	Товар доставлен перевозчиком СДЕК
5	6	2023-02-05	Товар доставлен другим перевозчиком
6	3	2023-02-10	Товар доставлен другим перевозчиком
7	7	2023-03-01	Товар доставлен перевозчиком СДЕК
8	8	2023-03-05	Товар доставлен другим перевозчиком
9	1	2023-03-10	Товар доставлен другим перевозчиком
10	5	2023-03-15	Товар доставлен перевозчиком СДЕК
11	4	2023-03-20	Товар доставлен другим перевозчиком
12	9	2023-04-01	Товар доставлен другим перевозчиком
13	5	2023-04-05	Товар доставлен перевозчиком СДЕК
14	6	2023-04-10	Товар доставлен другим перевозчиком
15	8	2023-04-15	Товар доставлен другим перевозчиком
16	10	2023-05-01	Товар доставлен перевозчиком СДЕК
17	4	2023-05-05	Товар доставлен другим перевозчиком
18	9	2023-05-10	Товар доставлен другим перевозчиком
19	7	2023-05-15	Товар доставлен перевозчиком СДЕК

4. Ход выполнения задания 3.

Количество заказов у каждого клиента

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

В результате запроса будут столбцы:

- *customer_name*: имя клиента
- *order_count*: количество заказов

135	-- Количество заказов у каждого клиента
136	• SELECT c.customer_name, COUNT(o.order_id) AS order_count
137	FROM customers AS c
138	LEFT JOIN orders AS o ON c.customer_id = o.customer_id
139	GROUP BY c.customer_name;
140	

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
customer_name	order_count		
Иван Иванов	2		
Мария Смирнова	1		
Алексей Попов	2		
Наталья Кузнецова	3		
Дмитрий Васильев	2		
Ольга Петрова	2		
Андрей Сидоров	2		
Елена Алексеева	2		
Сергей Морозов	2		
Ирина Фёдорова	1		
Андрей Иванов	0		
Екатерина Мартынова	0		

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

```
141 -- Количество заказов у каждого клиента
142 • SELECT c.customer_name
143 FROM customers AS c
144 LEFT JOIN orders AS o ON c.customer_id = o.customer_id
145 WHERE o.order_id IS NULL;
146
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

customer_name
Андрей Иванов
Екатерина Мартынова

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.

В результате запроса будут столбцы:

- `order_id`: идентификатор заказа
- `customer_id`: идентификатор клиента
- `total_amount`: сумма заказа

```
147 -- Заказы с высокой стоимостью
148 • DROP TABLE IF EXISTS HighValueOrders;
149
150 • CREATE TABLE HighValueOrders
151 (
152     HighVO_id INT PRIMARY KEY AUTO_INCREMENT NOT NULL,
153     order_id INT,
154     customer_id INT,
155     total_amount DECIMAL(10,2),
156     FOREIGN KEY (order_id) REFERENCES orders(order_id),
157     FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
158 );
159
160 • INSERT INTO HighValueOrders (order_id, customer_id, total_amount)
161     SELECT order_id, customer_id, total_amount
162     FROM orders
163     WHERE total_amount > 500;
164
165 • SELECT * FROM HighValueOrders;
```

Result Grid | Filter Rows: | Edit: | Export/Import: |

HighVO_id	order_id	customer_id	total_amount
1	1	1	750.00
2	2	3	820.00
3	3	2	600.00
4	4	4	670.00
5	5	6	550.00
*	NULL	NULL	NULL

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.

В результате запроса будут столбцы:

- *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
174 • SELECT
175     c.category_name,
176     SUM(od.quantity * p.price) AS total_sales,
177     ROUND(AVG(p.price),2) AS avg_price,
178     MIN(p.price) AS min_price,
179     MAX(p.price) AS max_price,
180     COUNT(DISTINCT od.order_id) AS unique_orders
181 FROM categories AS c
182 LEFT JOIN
183     products AS p ON p.category_id = c.category_id
184 LEFT JOIN
185     orderdetails AS od ON od.product_id = p.product_id
186 GROUP BY c.category_name
187 ORDER BY total_sales DESC
188 LIMIT 5;
```

	category_name	total_sales	avg_price	min_price	max_price	unique_orders
►	Художественная литература	3520.00	270.77	180.00	400.00	9
	Мистика	1050.00	262.50	0.00	350.00	3
	Наука	960.00	320.00	320.00	320.00	3