

## Homework\_Lesson17\_DB\_1

Цель: получить практический опыт написания SQL-запросов.

Задание 1:

Вводные данные

Есть таблица анализов Analysis:

- an\_id — ID анализа;
- an\_name — название анализа;
- an\_cost — себестоимость анализа;
- an\_price — розничная цена анализа;
- an\_group — группа анализов.

Есть таблица групп анализов Groups:

- gr\_id — ID группы;
- gr\_name — название группы;
- gr\_temp — температурный режим хранения.

Есть таблица заказов Orders:

- ord\_id — ID заказа;
- ord\_datetime — дата и время заказа;
- ord\_an — ID анализа.

Далее мы будем работать с этими таблицами.

Формулировка: вывести название и цену для всех анализов, которые продавались 5 февраля 2020 и всю следующую неделю.

Задание 2 (опционально): Используя left join, напишите запрос, который будет выводить список всех студентов и названий их курсов, которые они изучают. Если у студента нет курсов, то вместо названия курса нужно выводить NULL. Для этого вам необходимо связать таблицы "Студенты" и "Курсы".

Задание 3:

1. Создайте бэкап базы данных. Для этого используйте команду "mysqldump" для создания полного дампа базы данных. Сохраните файл дампа в безопасном месте, таком как внешний жесткий диск или облачное хранилище.

2. Измените какие-либо данные в базе данных, например, добавьте новую таблицу или обновите информацию в существующей таблице.

3. Восстановите базу данных из бэкапа, чтобы вернуть ее в исходное состояние. Для этого используйте команду "mysql" и укажите имя базы данных и файл дампа для восстановления.

4. Убедитесь, что база данных была восстановлена успешно, проверив данные и таблицы в базе данных.

5. Создайте скрипт, который будет автоматически создавать бэкап базы данных и отправлять его на удаленный сервер для хранения. Например, вы можете использовать инструмент "cron" для регулярного создания бэкапов и передачи их на удаленный сервер по расписанию.

**Задание 1:**

Установка MySQL:

\$ sudo apt install mysql-server	\\ установка СУБД MySQL
\$ sudo service mysql start	\\ старт сервера СУБД MySQL
\$ sudo service mysql status	\\ проверка состояния СУБД MySQL

```
makarov@makarov-VirtualBox:~$ sudo service mysql status
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2025-01-12 18:07:22 +03; 49s ago
     Process: 305609 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
    Main PID: 305617 (mysqld)
      Status: "Server is operational"
        Tasks: 38 (limit: 5678)
       Memory: 361.8M
          CPU: 1.448s
        CGroup: /system.slice/mysql.service
                └─305617 /usr/sbin/mysqld

cty 12 18:07:20 makarov-VirtualBox systemd[1]: Starting MySQL Community Server...
cty 12 18:07:22 makarov-VirtualBox systemd[1]: Started MySQL Community Server.
lines 1-14/14 (END)
```

### Проверка статуса mysql

\$ sudo mysql -u root -p                    \\ Заходим в среду управления MySQL  
если работаем из под root, то:  
\$ mysql

> CREATE DATABASE newdb\_analysis DEFAULT CHARACTER SET utf8  
DEFAULT COLLATE utf8\_general\_ci;  
> show databases;                        \\ выводит список БД  
> use newdb\_analysis                    \\ подключение к БД newdb\_analysis

```
mysql> CREATE DATABASE newdb_analysis DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
Query OK, 1 row affected, 2 warnings (0,02 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| newdb_analysis |
| performance_schema |
| sys |
+-----+
5 rows in set (0,00 sec)

mysql> use newdb_analysis
Database changed
mysql>
```

### Проверка и подключение к БД

Для работы с нашей БД создадим своего пользователя makarov:

> CREATE USER 'makarov'@'localhost' identified by 'makarov';  
> SELECT user, host FROM mysql.user;  
> GRANT ALL PRIVILEGES on mysql.user to 'makarov'@'localhost'; \\ даем все привилегии нашему созданному пользователю

```
mysql> create user 'makarov'@'localhost' identified by 'makarov';
Query OK, 0 rows affected (0,03 sec)

mysql> select user, host FROM mysql.user;
+-----+-----+
| user | host |
+-----+-----+
| debian-sys-maint | localhost |
| makarov | localhost |
| mysql.infoschema | localhost |
| mysql.session | localhost |
| mysql.sys | localhost |
| root | localhost |
+-----+-----+
6 rows in set (0,00 sec)

mysql> grant all privileges on mysql.user to 'makarov'@'localhost';
Query OK, 0 rows affected (0,01 sec)

mysql>
```

### Создание пользователя и назначение привилегий

Создаем SQL-скрипт на создание таблиц Analysis, Orders и Groups в нашей БД с необходимыми связями между ними:

```
CREATE TABLE Groups_1 (  
  gr_id INT PRIMARY KEY AUTO_INCREMENT,  
  gr_name VARCHAR(255),  
  gr_temp VARCHAR(50)  
);  
  
CREATE TABLE Analysis (  
  an_id INT PRIMARY KEY AUTO_INCREMENT,  
  an_name VARCHAR(255),  
  an_cost DECIMAL(10,2),  
  an_price DECIMAL(10,2),  
  an_group INT,  
  FOREIGN KEY (an_group) REFERENCES Groups_1(gr_id)  
);  
  
CREATE TABLE Orders (  
  ord_id INT PRIMARY KEY AUTO_INCREMENT,  
  ord_datetime DATETIME,  
  ord_an INT,  
  FOREIGN KEY (ord_an) REFERENCES Analysis(an_id)  
);
```

Создание таблицы с именем Groups выдавал синтаксическую ошибку “1064” поэтому пришлось переименовать таблицу как Groups\_1:

```
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your M  
ySQL server version for the right syntax to use near 'Groups (  
  gr_id INT PRIMARY KEY AUTO_INCREMENT,  
  gr_name VARCHAR(255),  
' at line 1  
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your M  
ySQL server version for the right syntax to use near 'Groups(gr_id)  
)' at line 7
```

> select database();                    \\ проверяем в какой БД находимся  
> source /mnt/shared\_folders/Create\_tables.sql    \\ выполнение нашего SQL-скрипта  
> show tables                            \\ просмотр созданных таблиц

```
mysql> select database();  
+-----+  
| database() |  
+-----+  
| newdb_analysis |  
+-----+  
1 row in set (0,00 sec)  
  
mysql> source /mnt/shared_folders/Create_table.sql  
Query OK, 0 rows affected (0,05 sec)  
  
Query OK, 0 rows affected (0,06 sec)  
  
Query OK, 0 rows affected (0,09 sec)  
  
mysql> show tables;  
+-----+  
| Tables_in_newdb_analysis |  
+-----+  
| Analysis |  
| Groups_1 |  
| Orders |  
+-----+  
3 rows in set (0,00 sec)  
  
mysql>
```

Наши таблицы необходимо заполнить. Добавление данных в таблицу выполняется с помощью команды INSERT. Создаем SQL-скрипт с INSERT для наших таблиц:

```
INSERT INTO Groups_1 (gr_name, gr_temp) VALUES
('Group A', 'Temp 1'),
('Group B', 'Temp 2'),
('Group C', 'Temp 3');
```

```
INSERT INTO Analysis (an_name, an_cost, an_price, an_group) VALUES
('Analysis 1', 40, 60, 1),
('Analysis 2', 80, 100, 2),
('Analysis 3', 120, 140, 3);
```

```
INSERT INTO Orders (ord_datetime, ord_an) VALUES
('2020-02-05 9:00:00', 1),
('2020-02-10 11:00:00', 2),
('2024-01-03 12:20:00', 3);
```

Проводим нами созданный скрипт с INSERT по БД:

```
mysql> source /mnt/shared_folders/insert_tables_analysis.sql
Query OK, 3 rows affected (0,06 sec)
Records: 3 Duplicates: 0 Warnings: 0

Query OK, 3 rows affected (0,01 sec)
Records: 3 Duplicates: 0 Warnings: 0

Query OK, 3 rows affected (0,03 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

SQL проведлся без ошибок по БД

Выполнение запроса в нашем задании: “Вывести название и цену для всех анализов, которые продавались 5 февраля 2020 и всю следующую неделю.” Для того, чтобы получить данные (сделать выборку данных) из наших таблиц, в SQL применяется команда “SELECT”.

Выполняем наш запрос:

```
> SELECT an_name, an_price FROM Analysis JOIN Orders ON an_id = ord_an
WHERE ord_datetime >= '2020-02-05 00:00:00' AND ord_datetime < '2020-02-12 00:00:00';
```

```
mysql> SELECT an_name, an_price
-> FROM Analysis
-> JOIN Orders ON an_id = ord_an
-> WHERE ord_datetime >= '2020-02-05 00:00:00'
-> AND ord_datetime < '2020-02-12 00:00:00';
+-----+-----+
| an_name | an_price |
+-----+-----+
| Analysis 1 | 60.00 |
| Analysis 2 | 100.00 |
+-----+-----+
2 rows in set (0,00 sec)
```

Наш запрос выполнялся верно, и вывел нам цену анализов, которые продавались с 5 февраля и всю следующую неделю

**Задание 2:** Используя left join, напишите запрос, который будет выводить список всех студентов и названий их курсов, которые они изучают. Если у студента нет курсов, то вместо названия курса нужно выводить NULL. Для этого вам необходимо связать таблицы "Студенты" и "Курсы".

```
$ CREATE DATABASE tms_courses DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci; \\\ создаем БД tms_courses
```

Создаем скрипт на создание таблиц Students и Courses, и данными вместе с ними. Для начала был создан файл Table\_student\_courses.sql. В нем мы создаем две таблицы Students и Courses. Таблица Courses соединена внешним ключом, student\_id.

```
CREATE TABLE Students (  
    student_id INT AUTO_INCREMENT PRIMARY KEY,  
    first_name VARCHAR(50) NOT NULL,  
    last_name VARCHAR(50) NOT NULL,  
    date_of_birth DATE,  
    email VARCHAR(100) UNIQUE,  
    enrollment_date DATE  
);  
  
CREATE TABLE Courses (  
    courses_id INT AUTO_INCREMENT PRIMARY KEY,  
    name_courses VARCHAR(100) NOT NULL,  
    student_id INT,  
    FOREIGN KEY (student_id) REFERENCES Students (student_id)  
);
```

Создаем скрипт insert\_std.sql на добавление данных в созданные таблицы:

```
INSERT INTO Students (student_id, first_name, last_name, date_of_birth, email, enrollment_date)  
VALUES  
(1, 'Александр', 'Макаров', '1995-02-28', 'sashamkr@mail.ru', '2024-10-20'),  
(2, 'Олег', 'Петров', '1993-09-20', 'user@gmail.com', '2024-08-10'),  
(3, 'Игорь', 'Иванов', '1994-05-06', 'qwerty@yandex.ru', '2023-05-25');  
  
INSERT INTO Courses (name_courses, student_id) VALUES  
( 'DevOps-инженер', 1),  
( 'Бизнес аналитика', 3),  
( 'UX/UI дизайнер', 1);
```

Выполняем скрипты на создание таблиц и добавления в них данных:

```
mysql> source /mnt/shared_folders/Table_student_courses.sql  
Query OK, 0 rows affected (0,10 sec)  
  
Query OK, 0 rows affected (0,08 sec)  
  
mysql> source /mnt/shared_folders/insert_std.sql  
Query OK, 3 rows affected (0,02 sec)  
Records: 3  Duplicates: 0  Warnings: 0  
  
Query OK, 3 rows affected (0,02 sec)  
Records: 3  Duplicates: 0  Warnings: 0
```

Выполнение скриптов успешное

Проверяем созданные таблицы:

```
mysql> select * from Students;
+-----+-----+-----+-----+-----+-----+
| student_id | first_name | last_name | date_of_birth | email | enrollment_date |
+-----+-----+-----+-----+-----+-----+
| 1 | Александр | Макаров | 1995-02-28 | sashamkr@mail.ru | 2024-10-20 |
| 2 | Олег | Петров | 1993-09-20 | user@gmail.com | 2024-08-10 |
| 3 | Игорь | Иванов | 1994-05-06 | qwerty@yandex.ru | 2023-05-25 |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0,00 sec)

mysql> select * from Courses;
+-----+-----+-----+
| courses_id | name_courses | student_id |
+-----+-----+-----+
| 1 | DevOps-инженер | 1 |
| 2 | Бизнес аналитика | 3 |
| 3 | UX/UI дизайнер | 1 |
+-----+-----+-----+
3 rows in set (0,00 sec)
```

Таблицы Students и Courses с данными

Запрос с использованием LEFT JOIN:

> SELECT Students.student\_id, Students.first\_name, Students.last\_name,  
Courses.name\_courses FROM Students LEFT JOIN Courses ON  
Students.student\_id=Courses.student\_id;

```
mysql> SELECT Students.student_id, Students.first_name, Students.last_name, Courses.name_courses FROM Students L
LEFT JOIN Courses ON Students.student_id=Courses.student_id;
+-----+-----+-----+-----+
| student_id | first_name | last_name | name_courses |
+-----+-----+-----+-----+
| 1 | Александр | Макаров | DevOps-инженер |
| 1 | Александр | Макаров | UX/UI дизайнер |
| 2 | Олег | Петров | NULL |
| 3 | Игорь | Иванов | Бизнес аналитика |
+-----+-----+-----+-----+
4 rows in set (0,00 sec)

mysql>
```

Таблицы связаны и выполненный запрос с использованием LEFT JOIN

### Задание 3:

1. Для создания бэкапов в MySQL используется команда “mysqldump”. Выгружаем дамп в общую директорию хостовой машины и гостевой VM.

```
$ mysqldump newdb_analysis > /mnt/shared_folders/expdump_10012025.sql
```

```
root@makarov-VirtualBox:~# mysqldump newdb_analysis > /mnt/shared_folders/expdump_10012025.sql
root@makarov-VirtualBox:~#
```

2. Делаем изменения в нашей БД. Добавляем пример в нашу БД таблицу клиентов:

2.1 Добавляем таблицу в БД.

```
CREATE TABLE clients (
  cl_id INT PRIMARY KEY AUTO_INCREMENT,
  cl_name VARCHAR(255) NOT NULL,
  cl_email VARCHAR(255) NOT NULL
);
```

```
mysql> CREATE table clients ( cl_id INT PRIMARY KEY AUTO_INCREMENT, cl_name VARCHAR(255) NOT NULL, cl_email VARCHAR(255) NOT NULL );
Query OK, 0 rows affected (0,07 sec)

mysql>
```

Наша таблица появилась в БД.

> show tables;

```
mysql> show tables;
+-----+
| Tables_in_newdb_analysis |
+-----+
| Analysis                  |
| Groups_1                  |
| Orders                    |
| clients                   |
+-----+
4 rows in set (0,00 sec)

mysql>
```

2.2 Также сделаем правки в таблице Analysis с помощью команды “UPDATE”. Для начала посмотрим нашу таблицу:

> select \* from Analysis;

```
mysql> select * from Analysis;
+-----+-----+-----+-----+-----+
| an_id | an_name | an_cost | an_price | an_group |
+-----+-----+-----+-----+-----+
| 1     | Analysis 1 | 40.00 | 60.00 | 1 |
| 2     | Analysis 2 | 80.00 | 100.00 | 2 |
| 3     | Analysis 3 | 120.00 | 140.00 | 3 |
+-----+-----+-----+-----+-----+
3 rows in set (0,00 sec)
```

Изменяем цену анализа с ID 1 себестоимость с 40 на 60 и цену анализа с 60 на 90.

UPDATE Analysis

SET an\_cost = 60, an\_price = 90

WHERE an\_id = 1;

```
mysql> select * from Analysis;
+-----+-----+-----+-----+-----+
| an_id | an_name | an_cost | an_price | an_group |
+-----+-----+-----+-----+-----+
| 1     | Analysis 1 | 40.00 | 60.00 | 1 |
| 2     | Analysis 2 | 80.00 | 100.00 | 2 |
| 3     | Analysis 3 | 120.00 | 140.00 | 3 |
+-----+-----+-----+-----+-----+
3 rows in set (0,00 sec)

mysql> UPDATE Analysis SET an_cost=60, an_price=90 WHERE an_id=1;
Query OK, 1 row affected (0,01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from Analysis;
+-----+-----+-----+-----+-----+
| an_id | an_name | an_cost | an_price | an_group |
+-----+-----+-----+-----+-----+
| 1     | Analysis 1 | 60.00 | 90.00 | 1 |
| 2     | Analysis 2 | 80.00 | 100.00 | 2 |
| 3     | Analysis 3 | 120.00 | 140.00 | 3 |
+-----+-----+-----+-----+-----+
3 rows in set (0,00 sec)

mysql>
```

Наглядное изменение значений с помощью UPDATE в таблице Analysis

3. Восстанавливаем нашу БД с помощью команды mysql:

```
$ mysql newdb_analysis < /mnt/shared_folders/expdump_10012025.sql
```

После загрузки нашего дампа БД, мы убеждаемся, что все значения вернулись в исходные.

```
Database changed
mysql> select * from Analysis;
+-----+-----+-----+-----+-----+
| an_id | an_name   | an_cost | an_price | an_group |
+-----+-----+-----+-----+-----+
| 1     | Analysis 1 | 40.00   | 60.00   | 1        |
| 2     | Analysis 2 | 80.00   | 100.00  | 2        |
| 3     | Analysis 3 | 120.00  | 140.00  | 3        |
+-----+-----+-----+-----+-----+
3 rows in set (0,00 sec)

mysql>
```

Но, при этом, таблица, созданная нами, осталась, скорее всего в дампе не дропается сама БД.

```
mysql> show tables;
+-----+
| Tables_in_newdb_analysis |
+-----+
| Analysis                  |
| Groups_1                  |
| Orders                    |
| clients                   |
+-----+
4 rows in set (0,02 sec)

mysql>
```

Попробуем удалить полностью БД и заново её загрузим:

```
> drop database newdb_analysis;
```

```
mysql> drop database newdb_analysis;
Query OK, 4 rows affected (0,11 sec)

mysql>
```

БД newdb\_analysis удалена

4. Восстанавливаем БД. Перед созданием БД, нужно сделать CREATE DATABASE newdb\_analysis. Только после создания, можно будет делать загрузку дампа в БД, т.к. при загрузке, мы указываем, как называется сама БД.

```
> CREATE DATABASE newdb_analysis DEFAULT CHARACTER SET utf8
DEFAULT COLLATE utf8_general_ci;
> exit
```

```
$ mysql newdb_analysis < /mnt/shared_folders/expdump_10012025.sql
```



```
mysql> CREATE DATABASE newdb_analysis DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
Query OK, 1 row affected, 2 warnings (0,02 sec)

mysql> exit
Bye
root@makarov-VirtualBox:~# mysql newdb_analysis < /mnt/shared_folders/expdump_10012025.sql
root@makarov-VirtualBox:~# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 25
Server version: 8.0.40-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| newdb_analysis |
| performance_schema |
| sys |
+-----+
5 rows in set (0,00 sec)

mysql>
```

БД восстановлена