

Пример создания базы данных в MySQL.

Создание пользователя БД

```
C:\Users\user>mysql -u root -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 8.4.2 MySQL Community Server - GPL

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>
```

Создание пользователя БД «my_library_owner»:

```
mysql> CREATE USER 'my_library_owner'@'localhost' IDENTIFIED BY '1111';
Query OK, 0 rows affected (0.02 sec)

mysql>
```

Смотрим наших пользователей:

```
mysql> SELECT User, Host FROM mysql.user;
+-----+-----+
| User           | Host       |
+-----+-----+
| my_library_owner | localhost  |
| mysql.infoschema | localhost  |
| mysql.session   | localhost  |
| mysql.sys       | localhost  |
| root            | localhost  |
+-----+-----+
```

Создание базы данных «my_library»:

```
mysql> CREATE DATABASE my_library;
Query OK, 1 row affected (0.02 sec)

mysql>
```

Проверим, что база создана:

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| my_library         |
| mysql              |
| performance_schema |
| students           |
| sys                |
+-----+
```

Назначим права (полный контроль) пользователю «my_library_owner» на базу данных «my_library»:

```
mysql> GRANT ALL PRIVILEGES ON my_library.* TO 'my_library_owner'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql>
```

Теперь посмотрим привилегии нашего пользователя:

```
mysql> SHOW GRANTS FOR 'my_library_owner'@'localhost';
+-----+
| Grants for my_library_owner@localhost |
+-----+
| GRANT USAGE ON *.* TO `my_library_owner`@`localhost` |
| GRANT ALL PRIVILEGES ON `my_library`.* TO `my_library_owner`@`localhost` |
+-----+
2 rows in set (0.00 sec)

mysql>
```

необходимо обновить таблицу прав пользователей MySQL в памяти:

```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

mysql>
```

Теперь выходим и заходим заново под новым пользователем:

```
mysql> exit
Bye

C:\Users\user>
```

```
C:\Users\user>mysql -u my_library_owner -p -h localhost my_library
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 18
Server version: 8.4.2 MySQL Community Server - GPL

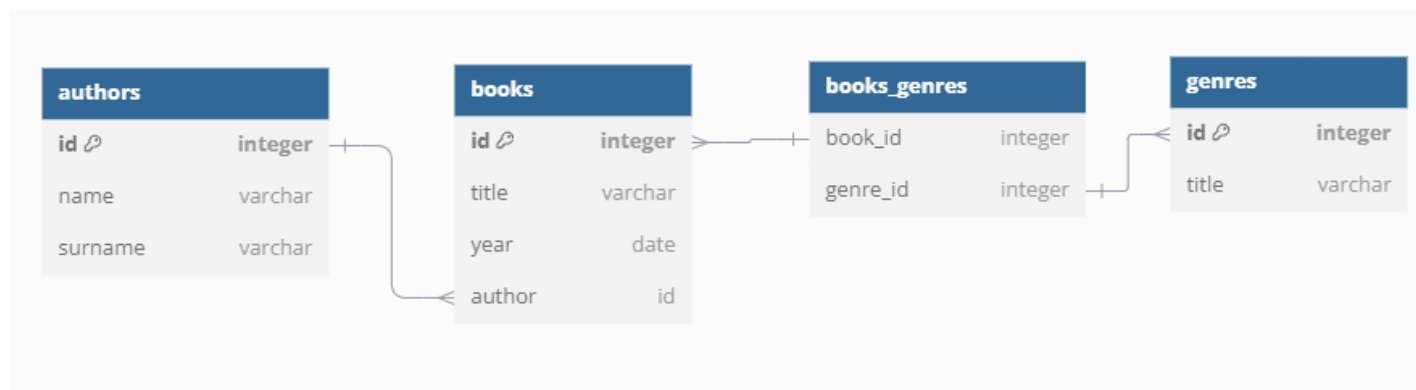
Copyright (c) 2000, 2024, Oracle and/or its affiliates.

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>
```

Создадим базу данных библиотеки согласно схемы:



```
mysql> CREATE TABLE authors (id INT PRIMARY KEY AUTO_INCREMENT,name VARCHAR(100) NOT NULL,surname VARCHAR(100) NOT NULL);
Query OK, 0 rows affected (0.06 sec)

mysql> CREATE TABLE genres (id INT PRIMARY KEY AUTO_INCREMENT, title VARCHAR(100) NOT NULL);
Query OK, 0 rows affected (0.06 sec)

mysql> CREATE TABLE books (id INT PRIMARY KEY AUTO_INCREMENT, title VARCHAR(255) NOT NULL, year YEAR, author INT, FOREIGN KEY
(author) REFERENCES authors(id));
Query OK, 0 rows affected (0.06 sec)

mysql> CREATE TABLE books_genres (book_id INT, genre_id INT, PRIMARY KEY (book_id, genre_id), FOREIGN KEY (book_id) REFERENCES
books(id), FOREIGN KEY (genre_id) REFERENCES genres(id));
Query OK, 0 rows affected (0.07 sec)

mysql>
```

Разъяснение запросов:

CREATE TABLE: Команда для создания новой таблицы.

id INT PRIMARY KEY AUTO_INCREMENT: Определяет поле id как целочисленный первичный ключ, который автоматически увеличивается при добавлении новой записи.

VARCHAR(100): Определяет строковое поле с максимальной длиной 100 символов.

YEAR: Определяет поле для хранения года.

FOREIGN KEY: Определяет внешний ключ, который ссылается на первичный ключ другой таблицы, устанавливая связь между таблицами.

Объяснение структуры таблицы books_genres:

Составной первичный ключ: Использование book_id и genre_id в качестве составного первичного ключа гарантирует уникальность каждой связи между книгой и жанром.

Внешние ключи: Оба поля book_id и genre_id являются внешними ключами, ссылающимися на таблицы books и genres соответственно. Это обеспечивает целостность данных и предотвращает добавление некорректных связей.

Проверим, что таблицы созданы:

```
mysql> SHOW TABLES;
+-----+
| Tables_in_my_library |
+-----+
| authors               |
| books                 |
| books_genres          |
| genres                |
+-----+
4 rows in set (0.00 sec)

mysql>
```

Проверим структуру таблиц:

```
mysql> DESCRIBE authors;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
name	varchar(100)	NO		NULL	
surname	varchar(100)	NO		NULL	

```
3 rows in set (0.00 sec)
```



```
mysql> DESCRIBE books;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
title	varchar(255)	NO		NULL	
year	year	YES		NULL	
author	int	YES	MUL	NULL	

```
4 rows in set (0.00 sec)
```

```
mysql> DESCRIBE genres;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
title	varchar(100)	NO		NULL	

```
2 rows in set (0.00 sec)
```



```
mysql> DESCRIBE books_genres;
```

Field	Type	Null	Key	Default	Extra
book_id	int	NO	PRI	NULL	
genre_id	int	NO	PRI	NULL	

```
2 rows in set (0.00 sec)
```

```
mysql>
```

Заполним таблицы:

```
mysql> INSERT INTO authors (name, surname)VALUES ('Александр', 'Пушкин');
Query OK, 1 row affected (0.02 sec)
```

```
mysql> INSERT INTO authors (name, surname)VALUES ('Александр', 'Дюма');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> INSERT INTO authors (name, surname)VALUES ('Джорж', 'Оруэлл');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> INSERT INTO authors (name, surname)VALUES ('Стивен', 'Кинг');
Query OK, 1 row affected (0.01 sec)
```

```
mysql>
```

Посмотрим заполненную таблицу:

```
mysql> SELECT * FROM authors;
+-----+-----+-----+
| id | name      | surname |
+-----+-----+-----+
| 1  | Александр | Пушкин  |
| 2  | Александр | Дюма    |
| 3  | Джорж     | Оруэлл  |
| 4  | Стивен    | Кинг    |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

Аналогично заполняем таблицу genres:

```
mysql> INSERT INTO genres (title)VALUES ('Фантастика');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO genres (title)VALUES ('Антиутопия');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO genres (title)VALUES ('Поэзия');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO genres (title)VALUES ('Ужасы');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO genres (title)VALUES ('Приключения');
Query OK, 1 row affected (0.01 sec)

mysql>
```

```
mysql> SELECT * FROM genres;
+-----+-----+
| id | title      |
+-----+-----+
| 1  | Фантастика |
| 2  | Антиутопия |
| 3  | Поэзия     |
| 4  | Ужасы      |
| 5  | Приключения |
+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

В таблице books в поле author указываем ID автора:

```
mysql> INSERT INTO books (title, year, author)VALUES ('Евгений Онегин', 1823, 1);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO books (title, year, author)VALUES ('Граф Монтекристо', 1844, 2);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO books (title, year, author)VALUES ('1984', 1948, 3);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO books (title, year, author)VALUES ('Скотный двор', 1948, 3);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO books (title, year, author)VALUES ('Оно', 1986, 4);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO books (title, year, author)VALUES ('Сияние', 1977, 4);
Query OK, 1 row affected (0.01 sec)
```

```
mysql> SELECT * FROM books;
+-----+-----+-----+-----+
| id | title           | year | author |
+-----+-----+-----+-----+
| 3  | Евгений Онегин  | 1823 | 1       |
| 4  | Граф Монтекристо | 1844 | 2       |
| 5  | 1984            | 1948 | 3       |
| 6  | Скотный двор    | 1948 | 3       |
| 7  | Оно             | 1986 | 4       |
| 8  | Сияние          | 1977 | 4       |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

Добавление связи книги и жанра. Одна книга может принадлежать нескольким жанрам, например приключения и фантастика:

```
mysql> INSERT INTO books_genres (book_id, genre_id) VALUES (3, 3);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO books_genres (book_id, genre_id) VALUES (4, 5);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO books_genres (book_id, genre_id) VALUES (5, 2);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO books_genres (book_id, genre_id) VALUES (6, 2);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO books_genres (book_id, genre_id) VALUES (5, 1);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO books_genres (book_id, genre_id) VALUES (7, 4);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO books_genres (book_id, genre_id) VALUES (8, 4);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO books_genres (book_id, genre_id) VALUES (7, 5);
```

Проверяем:

```
mysql> SELECT * FROM books_genres;
+-----+-----+
| book_id | genre_id |
+-----+-----+
|      5 |        1 |
|      5 |        2 |
|      6 |        2 |
|      3 |        3 |
|      7 |        4 |
|      8 |        4 |
|      4 |        5 |
|      7 |        5 |
+-----+-----+
8 rows in set (0.00 sec)

mysql>
```

Запросы к БД.

Выборка данных с условиями (WHERE). Выбрать все книги, опубликованные после 1950 года:

```
mysql> SELECT * FROM books WHERE year > 1950;
+----+-----+-----+-----+
| id | title | year | author |
+----+-----+-----+-----+
|  7 | Оно   | 1986 |      4 |
|  8 | Сияние | 1977 |      4 |
+----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Найти все книги определенного автора:

```
mysql> SELECT * FROM books WHERE author = 3;
+----+-----+-----+-----+
| id | title       | year | author |
+----+-----+-----+-----+
|  5 | 1984        | 1948 |      3 |
|  6 | Скотный двор | 1948 |      3 |
+----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Выбрать книги, название которых начинается на букву "С":

```
mysql> SELECT * FROM books WHERE title LIKE 'С%';
+----+-----+-----+-----+
| id | title       | year | author |
+----+-----+-----+-----+
|  6 | Скотный двор | 1948 |      3 |
|  8 | Сияние       | 1977 |      4 |
+----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

Сортировка результатов (ORDER BY). Отсортировать книги по году публикации в порядке возрастания:

```
mysql> SELECT * FROM books ORDER BY year ASC;
+----+-----+-----+-----+
| id | title          | year | author |
+----+-----+-----+-----+
| 3  | Евгений Онегин | 1823 | 1       |
| 4  | Граф Монтекрис | 1844 | 2       |
| 5  | 1984           | 1948 | 3       |
| 6  | Скотный двор   | 1948 | 3       |
| 8  | Сияние         | 1977 | 4       |
| 7  | Оно            | 1986 | 4       |
+----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

И убывания:

```
mysql> SELECT * FROM books ORDER BY year DESC;
+----+-----+-----+-----+
| id | title          | year | author |
+----+-----+-----+-----+
| 7  | Оно            | 1986 | 4       |
| 8  | Сияние         | 1977 | 4       |
| 5  | 1984           | 1948 | 3       |
| 6  | Скотный двор   | 1948 | 3       |
| 4  | Граф Монтекрис | 1844 | 2       |
| 3  | Евгений Онегин | 1823 | 1       |
+----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

Отсортировать книги по названию в порядке возрастания:

```
mysql> SELECT * FROM books ORDER BY title ASC;
+----+-----+-----+-----+
| id | title          | year | author |
+----+-----+-----+-----+
| 5  | 1984           | 1948 | 3       |
| 4  | Граф Монтекрис | 1844 | 2       |
| 3  | Евгений Онегин | 1823 | 1       |
| 7  | Оно            | 1986 | 4       |
| 8  | Сияние         | 1977 | 4       |
| 6  | Скотный двор   | 1948 | 3       |
+----+-----+-----+-----+
```

Агрегатные функции (COUNT, SUM, AVG, MIN, MAX). Подсчитать общее количество книг:

```
mysql> SELECT COUNT(*) FROM books;
+-----+
| COUNT(*) |
+-----+
|         6 |
+-----+
1 row in set (0.00 sec)

mysql>
```


Найти самую древнюю книгу:

```
mysql> SELECT title, year FROM books WHERE year = (SELECT MIN(year) FROM books);
+-----+-----+
| title          | year |
+-----+-----+
| Евгений Онегин | 1823 |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Вычислить средний год публикации:

```
mysql> SELECT AVG(year) FROM books;
+-----+
| AVG(year) |
+-----+
| 1921.0000 |
+-----+
1 row in set (0.00 sec)

mysql>
```

Сгруппировать книги по авторам и посчитать количество книг каждого автора:

```
mysql> SELECT author, COUNT(*) AS count_book FROM books GROUP BY author;
+-----+-----+
| author | count_book |
+-----+-----+
| 1      | 1          |
| 2      | 1          |
| 3      | 2          |
| 4      | 2          |
+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

Сочетание условий, сортировки и агрегации. Найти средний год публикации книг Стивена Кинга, отсортированных по году:

```
mysql> SELECT AVG(year) FROM books WHERE author = 4 GROUP BY author ORDER BY AVG(year) DESC;
+-----+
| AVG(year) |
+-----+
| 1981.5000 |
+-----+
1 row in set (0.00 sec)

mysql>
```