

MySQL Walkthrough

Open MySQL command line client and create a database for your solution:

```
mysql> create database books;
Query OK, 1 row affected (0.02 sec)
```

```
mysql> show databases;
+-----+
| Database |
+-----+
| books    |
| crm      |
| information_schema |
| mysql    |
| performance_schema |
| sakila   |
| sys      |
| world    |
+-----+
8 rows in set (0.04 sec)
```

```
mysql> use books;
Database changed
```

Possibly you will need to change the default character set (if you already have utf8 or utf8mb4, you don't have to change):

```
mysql> show create database books;
+-----+-----+
| Database | Create Database |
+-----+-----+
| books    | CREATE DATABASE `books` /*!40100 DEFAULT CHARACTER SET latin1 */ |
+-----+-----+
```

```
mysql> alter database books default character set utf8;
```

Tables

For each entity type we need a table which will hold the items of that type:

- Table names should be in singular form*
- Each table should have primary key, don't use actual data item for that (don't specify email to be primary key for person)*

```
mysql> create table author(id int primary key not null auto_increment,
lastName varchar(32),firstName varchar(32));
Query OK, 0 rows affected (0.07 sec)
```

Insert data

```
mysql> insert into author values(null,'Tolkien','J.R.R.');
```

Query OK, 1 row affected (0.02 sec)

```
mysql> insert into author values(null,'Vonnegut','Kurt');
```

Query OK, 1 row affected (0.01 sec)

```
mysql> insert into author values(null,'Hemingway','Ernst');
```

Query OK, 1 row affected (0.01 sec)

```
mysql> insert into author(lastName,firstName) values('Kyrö','Tuomas');
```

Query OK, 1 row affected (0.01 sec)

Basic queries

```
mysql> select * from author;
```

id	lastName	firstName
1	Tolkien	J.R.R.
2	Vonnegut	Kurt
3	Hemingway	Ernst
4	Kyrö	Tuomas

4 rows in set (0.00 sec)

```
mysql> select * from author where lastName='Tolkien';
```

id	lastName	firstName
1	Tolkien	J.R.R.

1 row in set (0.00 sec)

```
mysql> select * from author where lastName like '%g%';
```

id	lastName	firstName
2	Vonnegut	Kurt
3	Hemingway	Ernst

2 rows in set (0.01 sec)

```
mysql> select * from author where lastName like 't%';
```

id	lastName	firstName
1	Tolkien	J.R.R.

1 row in set (0.00 sec)

Modifying table

```
mysql> create table book(id int primary key not null auto_increment, title
varchar(32));
```

Query OK, 0 rows affected (0.06 sec)

```
mysql> insert into book values(null,'Hobbit');
```

Query OK, 1 row affected (0.01 sec)

```
mysql> alter table book add author_id int after id;
```

Query OK, 0 rows affected (0.27 sec)

Records: 0 Duplicates: 0 Warnings: 0

```
mysql> select * from book;
```

id	author_id	title
1	NULL	Hobbit

1 row in set (0.00 sec)

```
mysql> update book set author_id=1 where id=1;
```

Query OK, 1 row affected (0.01 sec)

Rows matched: 1 Changed: 1 Warnings: 0

```
mysql> select * from book;
```

id	author_id	title
1	1	Hobbit

1 row in set (0.00 sec)

```
mysql> insert into book values(null,null,'Tuntematon sotilas');
```

Query OK, 1 row affected (0.01 sec)

```
mysql> select * from book;
```

id	author_id	title
1	1	Hobbit
2	NULL	Tuntematon sotilas

2 rows in set (0.00 sec)

Joins, data from several tables

Inner join

```
mysql> select * from book,author where author_id=author.id;
```

```
+-----+-----+-----+-----+-----+-----+
| id | author_id | title | id | lastName | firstName |
+-----+-----+-----+-----+-----+-----+
| 1 | 1 | Hobbit | 1 | Tolkien | J.R.R. |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Left (outer) join always returns all rows from the first table with nulls on columns of second table if the counterpart is not found:

```
mysql> select * from book left join author on author_id=author.id;
```

```
+-----+-----+-----+-----+-----+-----+
| id | author_id | title | id | lastName | firstName |
+-----+-----+-----+-----+-----+-----+
| 1 | 1 | Hobbit | 1 | Tolkien | J.R.R. |
| 2 | NULL | Tuntematon sotilas | NULL | NULL | NULL |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

What would this show?

```
mysql> select * from author left join book on author_id=author.id;
```

Exercises

Query the id and name of author (combine lastname, ' ', ' and firstname) to name column

Extend the previous query to also show number of books from that author. What if the author doesn't have any books?

Extend the previous query to display result set in descending order of number of books. Use name as secondary sort criteria

Create table person with id,firstname,lastname,email,password. Id is the primary key, the other fields may be varchar(32)

Create table loan. It is a join table for many-to-many -relation. Person may loan several books, book may be loaned by several persons. Add few loans and try to query

Modify the loan table, add column loandate.

Modify author-table. Add columns birthdate and deathdate.

Author id is actually a foreign key in the book-table but there is no definition yet that tells the database engine that fact. Can you figure out how to add the foreign key constraint:

<https://dev.mysql.com/doc/refman/5.6/en/create-table-foreign-keys.html>

Try to add a book with authorid pointing to a non-existing author.

Same person may loan the book couple of times. Add few rows like that and then try to make a query that shows persons name, title of the book and how many times he has loaned that book. How about not displaying the count but the date of latest loan.

Now create a view (loanview) that shows everything from loan-table+person's name and book's title

Final step: create a user

```
mysql> create user 'librarian'@'localhost' identified BY 'test123';  
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> grant all privileges on books.* to librarian@localhost;  
Query OK, 0 rows affected (0.01 sec)
```

