```
show databases;

create database jid_thrillio;

drop database jid_thrillio;

create database jid_thrillio;

use jid_thrillio;

----------------

show tables;
```

Note on Conventions:
+ Use singular for table names.
+ For multiple words, you can separate them with underscores
+ Case does not matter

Integer: tinyint, smallint, mediumint, int, bigint
Fixed-point (exact): decimal ~ monetary
Floating-point (approx): float, double
Strings: char, varchar, blob, text
Dates: date, time, datetime, timestamp

Resources:
https://dev.mysql.com/doc/refman/5.7/en/data-types.html

```
CREATE TABLE Author(id bigint PRIMARY KEY
auto_increment,
name varchar(200) not null);

CREATE TABLE Publisher(id bigint PRIMARY KEY
auto_increment,
name varchar(200) not null);
```

```
CREATE TABLE Book(id bigint PRIMARY KEY
auto_increment,
title varchar(500) not null,
publication_year int,
publisher_id bigint,
book_genre_id TINYINT,
amazon_rating double,
kid_friendly_status TINYINT,
created_date datetime not null,
FOREIGN KEY (publisher_id) REFERENCES Publisher(id),
constraint UNIQUE (title (250), publication_year,
publisher_id));

CREATE TABLE Book_Author (id bigint PRIMARY KEY
auto_increment,
book_id bigint not null,
author_id bigint not null,
FOREIGN KEY (book_id) REFERENCES Book(id),
FOREIGN KEY (author_id) REFERENCES Author(id),
constraint UNIQUE(book_id, author_id));

INSERT INTO Author (name) VALUES ('Henry David
Thoreau');
INSERT INTO Author (name) VALUES ('Ralph Waldo
Emerson');
INSERT INTO Author (name) VALUES ('Lillian Eichler
Watson');
// Multiple rows in same query
INSERT INTO Author (name) VALUES ('Eric Freeman'),
('Bert Bates'), ('Kathy Sierra'), ('Elisabeth
Robson');
INSERT INTO Author (name) VALUES ('Joshua Bloch');

SELECT * FROM Author;

INSERT INTO Publisher (name) VALUES ('Wilder
Publications');
```

```
INSERT INTO Publisher (name) VALUES ('Dover
Publications');
INSERT INTO Publisher (name) VALUES ('Touchstone');
INSERT INTO Publisher (name) VALUES ("O'Reilly
Media");
INSERT INTO Publisher (name) VALUES ('Prentice
Hall');

INSERT INTO Book (title, publication_year,
publisher_id, book_genre_id, amazon_rating,
kid_friendly_status, created_date) VALUES ('Walden',
1854, 1, 6, 4.3, 2, NOW());
INSERT INTO Book (title, publication_year,
publisher_id, book_genre_id, amazon_rating,
kid_friendly_status, created_date) VALUES
('Self-Reliance and Other Essays', 1993, 2, 6, 4.5,
2, NOW());
INSERT INTO Book (title, publication_year,
publisher_id, book_genre_id, amazon_rating,
kid_friendly_status, created_date) VALUES ('Light
From Many Lamps', 1988, 3, 6, 5.0, 2, NOW());
INSERT INTO Book (title, publication_year,
publisher_id, book_genre_id, amazon_rating,
kid_friendly_status, created_date) VALUES ('Head
First Design Patterns', 2004, 4, 10, 4.5, 2, NOW());
INSERT INTO Book (title, publication_year,
publisher_id, book_genre_id, amazon_rating,
kid_friendly_status, created_date) VALUES
('Effective Java Programming Language Guide', 2007,
5, 10, 4.9, 2, NOW());

INSERT INTO Book_Author (book_id, author_id) VALUES
(1, 1), (2, 2), (3, 3), (4, 4), (4, 5), (4, 6), (4,
7), (5, 8);

DELETE from Book_Author where book_id = 5;

INSERT INTO Book_Author (book_id, author_id) VALUES
```

```
(5, 8);

--------------------

CREATE TABLE Actor(id bigint PRIMARY KEY
auto_increment,
name varchar (200));

CREATE TABLE Director(id bigint PRIMARY KEY
auto_increment,
name varchar (200));

CREATE TABLE Movie(id bigint PRIMARY KEY
auto_increment,
title varchar(500) not null,
release_year int,
movie_genre_id TINYINT,
imdb_rating double,
kid_friendly_status TINYINT,
created_date datetime not null,
CONSTRAINT UNIQUE(title (100), release_year));

CREATE TABLE Movie_Actor(id bigint PRIMARY KEY
auto_increment,
movie_id bigint not null,
actor_id bigint not null,
FOREIGN KEY (movie_id) REFERENCES movie(id),
FOREIGN KEY (actor_id) REFERENCES actor(id),
CONSTRAINT UNIQUE(movie_id, actor_id));

CREATE TABLE Movie_Director(id bigint PRIMARY KEY
auto_increment,
movie_id bigint not null,
director_id bigint not null,
FOREIGN KEY (movie_id) REFERENCES movie(id),
FOREIGN KEY (director_id) REFERENCES director(id),
CONSTRAINT UNIQUE(movie_id, director_id));
```

```
INSERT INTO actor (name) VALUES ('Orson Welles'),
('Joseph Cotten'), ('Henry Fonda'), ('Jane
Darwell'), ('Albert Cullum'), ('Kaley Cuoco'), ('Jim
Parsons'), ('Takashi Shimura'), ('Minoru Chiaki');

INSERT INTO director (name) VALUES ('Orson Welles'),
('John Ford'), ('Leslie Sullivan'), ('Chuck Lorre'),
('Bill Prady'), ('Akira Kurosawa');

INSERT INTO movie (title, release_year,
movie_genre_id, imdb_rating, kid_friendly_status,
created_date) VALUES ('Citizen Kane', 1941, 0, 8.5,
2, NOW());
INSERT INTO movie (title, release_year,
movie_genre_id, imdb_rating, kid_friendly_status,
created_date) VALUES ('The Grapes of Wrath', 1940,
0, 8.2, 2, NOW());
INSERT INTO movie (title, release_year,
movie_genre_id, imdb_rating, kid_friendly_status,
created_date) VALUES ('A Touch of Greatness', 2004,
24, 7.3, 2, NOW());
INSERT INTO movie (title, release_year,
movie_genre_id, imdb_rating, kid_friendly_status,
created_date) VALUES ('The Big Bang Theory', 2007,
20, 8.7, 2, NOW());
INSERT INTO movie (title, release_year,
movie_genre_id, imdb_rating, kid_friendly_status,
created_date) VALUES ('Ikiru', 1952, 25, 8.4, 2,
NOW());

INSERT INTO movie_actor (movie_id, actor_id) VALUES
(1, 1), (1, 2), (2, 3), (2, 4), (3, 5), (4, 6), (4,
7), (5, 8), (5, 9);

INSERT INTO movie_director(movie_id, director_id)
VALUES (1, 1), (2, 2), (3, 3), (4, 4), (4, 5), (5,
6);
```

```
------------

CREATE TABLE WebLink(id bigint PRIMARY KEY
auto_increment,
title varchar(500) not null,
url varchar(250) not null,
host  varchar(250) not null,
kid_friendly_status TINYINT,
created_date datetime not null,
CONSTRAINT UNIQUE(url (200)));

INSERT INTO WebLink (title, url, host,
kid_friendly_status, created_date) VALUES ('Use
Final Liberally',
'http://www.javapractices.com/topic/TopicAction.do?I
d=23', 'http://www.javapractices.com', 2, NOW());
INSERT INTO WebLink (title, url, host,
kid_friendly_status, created_date) VALUES ('How do I
import a pre-existing Java project into Eclipse and
get up and running?',
'https://stackoverflow.com/questions/142863/how-do-i
-import-a-pre-existing-java-project-into-eclipse-and
-get-up-and-running', 'http://stackoverflow.com', 2,
NOW());
INSERT INTO WebLink (title, url, host,
kid_friendly_status, created_date) VALUES
('Interface vs Abstract Class',
'http://mindprod.com/jgloss/interfacevsabstract.html
', 'http://mindprod.com', 2, NOW());
INSERT INTO WebLink (title, url, host,
kid_friendly_status, created_date) VALUES ('NIO
tutorial by Greg Travis',
'http://cs.brown.edu/courses/cs161/papers/j-nio-ltr.
pdf', 'http://cs.brown.edu', 2, NOW());
INSERT INTO WebLink (title, url, host,
kid_friendly_status, created_date) VALUES ('Virtual
Hosting and Tomcat',
'http://tomcat.apache.org/tomcat-6.0-doc/virtual-hos
```

```
ting-howto.html', 'http://tomcat.apache.org', 2,
NOW());

------------

CREATE TABLE User(id bigint PRIMARY KEY
auto_increment,
email varchar(100) not null,
password varchar(50) not null,
first_name  varchar(100) not null,
last_name  varchar(100) not null,
gender_id TINYINT,
user_type_id TINYINT,
created_date datetime not null,
CONSTRAINT UNIQUE(email));

INSERT INTO User (email, password, first_name,
last_name, gender_id, user_type_id, created_date)
VALUES ('user0@semanticsquare.com', 'test', 'John',
'M', 0, 0, NOW());
INSERT INTO User (email, password, first_name,
last_name, gender_id, user_type_id, created_date)
VALUES ('user1@semanticsquare.com', 'test', 'Sam',
'M', 0, 0, NOW());
INSERT INTO User (email, password, first_name,
last_name, gender_id, user_type_id, created_date)
VALUES ('user2@semanticsquare.com', 'test', 'Anita',
'M', 1, 1, NOW());
INSERT INTO User (email, password, first_name,
last_name, gender_id, user_type_id, created_date)
VALUES ('user3@semanticsquare.com', 'test', 'Sara',
'M', 1, 1, NOW());
INSERT INTO User (email, password, first_name,
last_name, gender_id, user_type_id, created_date)
VALUES ('user4@semanticsquare.com', 'test',
'Dheeru', 'M', 0, 2, NOW());

-------------------------------------------
```

Storage Engines: Determines how MySQL stores data + which features are available

```
SHOW engines;
```

InnoDB: default since 5.5. Supports foreign keys and transactions (e.g., checking to saving)

MyISAM: default before 5.5. No support for foreign keys, no transactions, supports full-text search

```
SELECT table_name, engine FROM
information_Schema.tables where table_schema =
'jid_thrillio';

CREATE TABLE TestA(id int primary key)
ENGINE=MyISAM;

SELECT table_name, engine FROM
information_Schema.tables where table_schema =
'jid_thrillio';

DROP TABLE TestA;

exit;
```