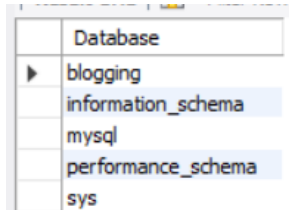


Lab3 MySQL Commands

```
CREATE DATABASE Blogging;
```

SHOW DATABASES;



A screenshot of a MySQL database interface showing a list of databases. The 'blogging' database is highlighted with a mouse cursor pointing to it. Other databases listed include information_schema, mysql, performance_schema, and sys.

Database
blogging
information_schema
mysql
performance_schema
sys

Navigate into the new database with: `use Blogging;`

Create a 'user' table with the following information:

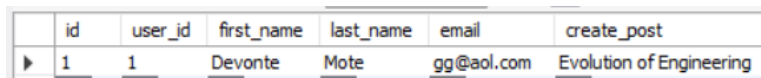
```
CREATE TABLE user (  
  id INT PRIMARY KEY,  
  user_id INT,  
  first_name VARCHAR(50),  
  last_name VARCHAR(50),  
  email VARCHAR(50),  
  create_post VARCHAR(50)  
);
```

Pass in test values:

```
INSERT INTO user VALUES (1, 1, "Devonte", "Mote", "gg@aol.com", "Evolution of Engineering")
```

Display the table to ensure values are updates:

```
SELECT * FROM blogging.user;
```



A screenshot of a MySQL query result showing a single row of data from the 'user' table. The columns are id, user_id, first_name, last_name, email, and create_post. The values are 1, 1, Devonte, Mote, gg@aol.com, and Evolution of Engineering.

id	user_id	first_name	last_name	email	create_post
1	1	Devonte	Mote	gg@aol.com	Evolution of Engineering

Create and update table 'posts' with the Foreign Key:

```
CREATE TABLE posts (  
  id INT PRIMARY KEY,  
  user_id INT,  
  post_id INT,  
  title VARCHAR (100),  
  description VARCHAR (1000),  
  post_time TIME,  
  published DATE,  
  image BLOB,  
  FOREIGN KEY (user_id) REFERENCES user(id)  
);
```



A screenshot of a MySQL query result showing a single row of data from the 'posts' table. The columns are id, user_id, post_id, title, description, post_time, published, and image. All values are NULL.

id	user_id	post_id	title	description	post_time	published	image
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Create and update table 'comments' with the Foreign Keys:

```

CREATE TABLE comments (
  id INT PRIMARY KEY,
  user_id INT,
  post_id INT,
  comment VARCHAR(500),
  parent_id INT,
  FOREIGN KEY (user_id) REFERENCES user(id),
  FOREIGN KEY (post_id) REFERENCES posts(id)
);

```

id	user_id	post_id	comment	parent_id
NULL	NULL	NULL	NULL	NULL

Create and update "reactions" table with the corresponding Foreign Keys:

```

CREATE TABLE reactions (
  id INT PRIMARY KEY,
  user_id INT,
  post_id INT,
  comment VARCHAR(500),
  action_type ENUM ('like', 'comment'),
  FOREIGN KEY (user_id) REFERENCES user(id),
  FOREIGN KEY (post_id) REFERENCES posts(id)
);

```

id	user_id	post_id	comment	action_type
NULL	NULL	NULL	NULL	NULL

The breakdown of the 'blogging' database layout:

