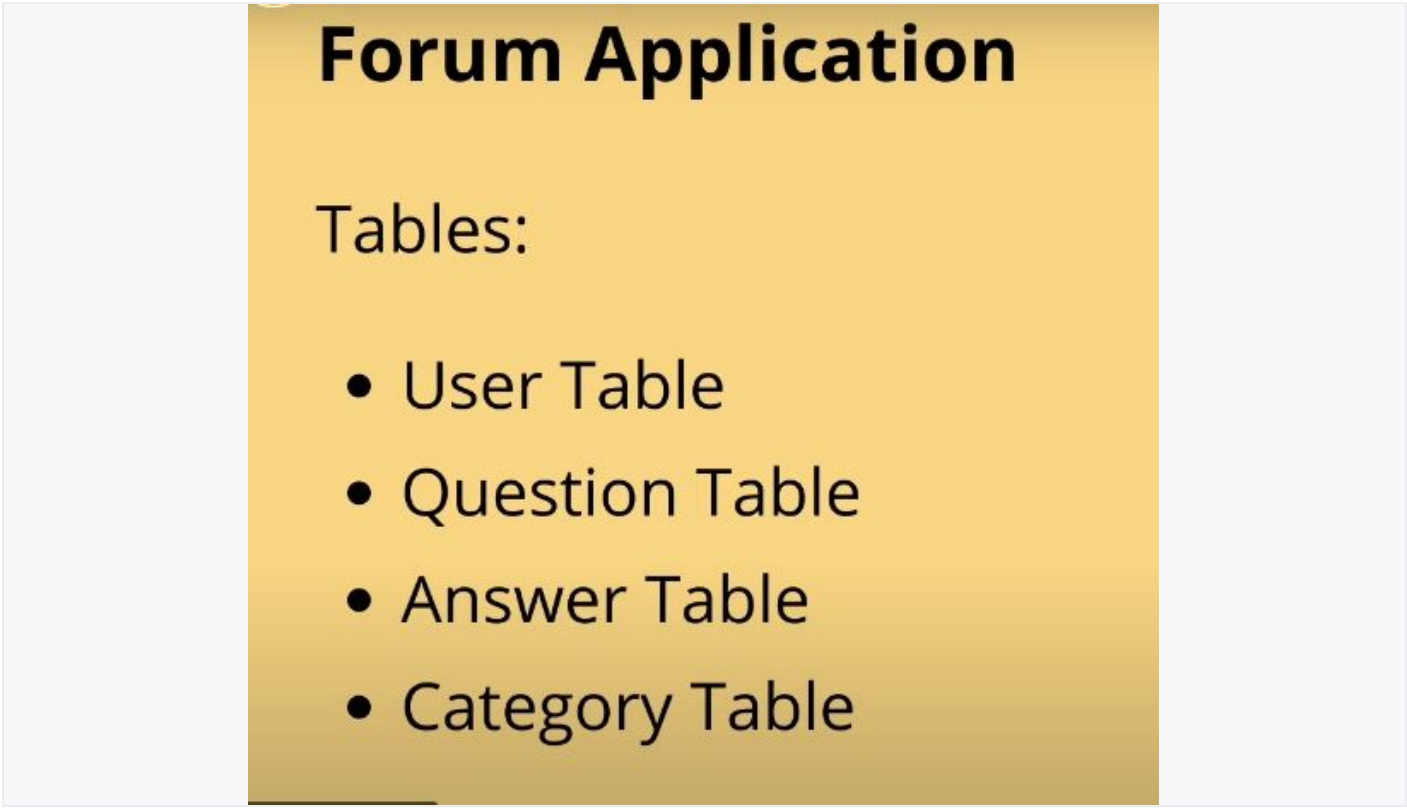
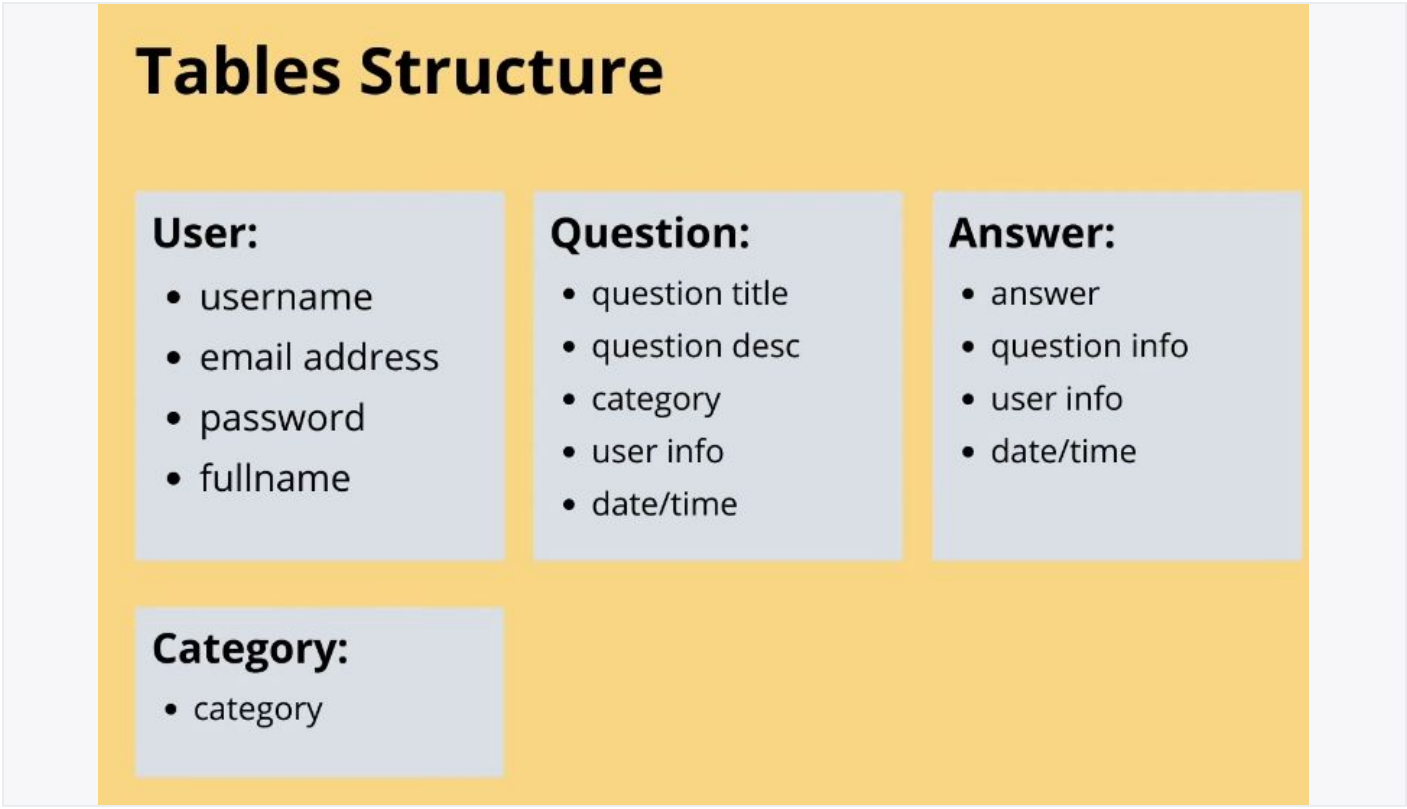
## Database Design

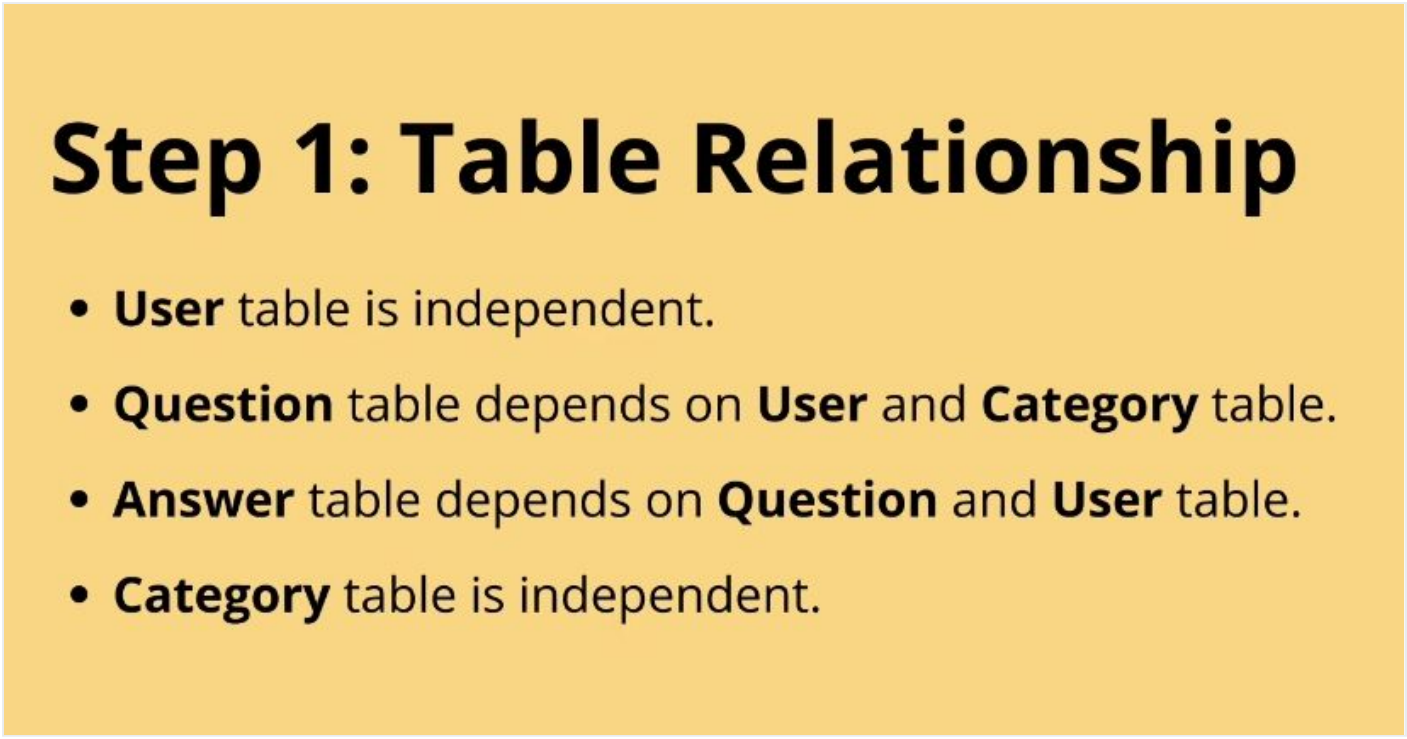



Let's create a database design for a forum.

1. Table Structure: what will be stored in which table, like the fields and data types of each need to be gathered in this stage.

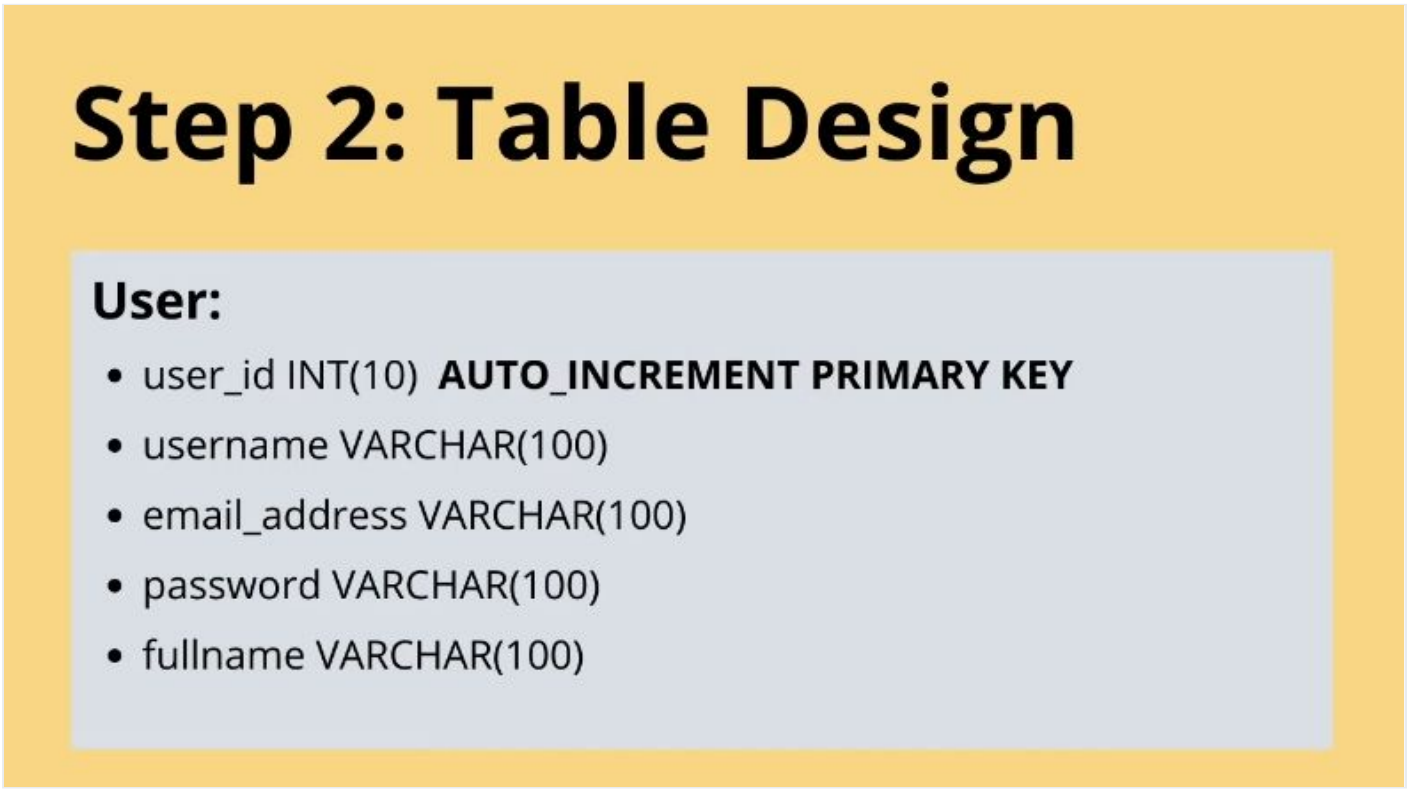


2. Now as we have the table structure ready comes the mapping or setting up **relationship** between the tables. Which table is dependent on another table for its existence.




3. We should start with independent tables first and then move to dependent tables.

4. Create the tables




**Note: when we have the password kindly hash them using some hashing technique hashing is different than encryption.**

# Step 2: Table Design

**Category**

- category_id INT(10) **AUTO_INCREMENT PRIMARY KEY**
- category VARCHAR(100)

# Step 2: Table Design

**Question:**

- que_id INT(10) **AUTO_INCREMENT PRIMARY KEY**
- user_id_fk INT(10) **FOREIGN KEY CONSTRAINT**
- cat_id_fk INT(10) **FOREIGN KEY CONSTRAINT**
- question VARCHAR(100)
- question_desc TEXT
- created INT(10) or DATE

# Step 2: Table Design

**Answer:**

- ans_id INT(10) **AUTO_INCREMENT PRIMARY KEY**
- que_id_fk INT(10) **FOREIGN KEY CONSTRAINT**
- user_id_fk INT(10) **FOREIGN KEY CONSTRAINT**
- answer TEXT
- created INT(10) or DATE

**Id of each table is preferred to be of number type and auto_incremented rather than a string because it is first of all generated by the database so it will always be unique and by number or int type the indexing becomes fast as compared to any other data type.**

**This database is normalized to an extent, sometimes more levels of normalization can even lead to more complexities.**

**The foreign key constraint is a very good way of validating whether that data exists or not, if while saving data in the database someone breaches the security and tries to save some data that does not exist in the table then this will help you a lot.**

**This is also called a fail-safe mechanism wrt to the database.**