Table of Contents:

Sample Data and Instructions
Creating the Sample Database

```
CREATE DATABASE company;
USE company;

CREATE TABLE employees (
        id INT AUTO_INCREMENT PRIMARY KEY,
        first_name VARCHAR(50) NOT NULL,
        last_name VARCHAR(50) NOT NULL,
        email VARCHAR(100) UNIQUE NOT NULL,
        hire_date DATE NOT NULL
);

INSERT INTO employees (first_name, last_name, email, hire_date) VALUES
('John', 'Doe', 'john.doe@example.com', '2023-01-15'),
('Jane', 'Smith', 'jane.smith@example.com', '2023-02-20'),
('Alice', 'Johnson', 'alice.johnson@example.com', '2023-03-10'),
('Bob', 'Brown', 'bob.brown@example.com', '2023-04-05');
```

These commands create a `company` database and an `employees` table, and inserts sample data into the table.

Instructions for Set up:

1. Create and Use the Database:
   - Run the `CREATE DATABASE` and `USE` statements.
2. Create and Manage Tables:
   - Create the `employees` table using the provided SQL statement.
   - Insert sample data into the table.
3. Perform Data Operations:
   - Use `DELETE`, `ALTER`, `DROP`, `TRUNCATE`, and other statements to manipulate the data and table structure.
4. Temporary Tables and Backup:
   - Create temporary tables and backup tables to understand their usage.
5. Insert, Update, and Delete Operations:
   - Perform various insert, update, and delete operations to get hands-on experience.

1. Introduction to Tables in SQL

- Tables are the fundamental unit of storage in an SQL database. A table consists of rows and columns.

2. Creating a Table

```
CREATE TABLE employees (
        id INT AUTO_INCREMENT PRIMARY KEY,
        first_name VARCHAR(50) NOT NULL,
        last_name VARCHAR(50) NOT NULL,
        email VARCHAR(100) UNIQUE NOT NULL,
        hire_date DATE NOT NULL
);
```

This statement creates a table named `employees` with five columns: `id`, `first_name`, `last_name`, `email`, and `hire_date`.

3. Dropping a Table

SQL DROP TABLE statement deletes data and structure of a table from the database.DROP TABLE command is irreversible, so once a DROP TABLE command is executed for a table, all information about that table will be removed permanently.

DROP command in SQL is used to delete a whole database or a table.

```
DROP TABLE IF EXISTS employees;
```

This statement drops the `employees` table if it exists.

4. Using the DELETE Statement

DELETE can be used to **selectively remove records from a database table based on certain conditions.** Take note of the difference between DELETE and DROP.

The syntax is : DELETE FROM table_name WHERE some_condition;

- **table_name:** name of the table
- **some_condition :** condition to choose a particular record.

DELETE FROM employees WHERE id = 3;

This statement deletes the row from the `employees` table where the `id` is 3.

5. ALTER in SQL

ALTER TABLE command can add, delete, or modify columns of an existing table.
Sometimes we may want to rename our table to give it a more relevant name. For this purpose, we can use ALTER TABLE to rename the name of the table. SQL ALTER TABLE is a command used to modify the structure of an existing table in a database.
For example in our database we have an existing table names employees , lets rename it to staff.

ALTER TABLE employees RENAME TO staff;

This statement renames the `employees` table to `staff`.

ALTER TABLE employees ADD COLUMN phone_number VARCHAR(15);

This statement adds a new column `phone_number` to the `employees` table.

6. DROP and TRUNCATE in SQL

DROP and TRUNCATE in SQL remove data from the table. The main difference between DROP and TRUNCATE commands in SQL is that DROP removes the table or database completely, while TRUNCATE only removes the data, preserving the table structure.

-- Drop Table
DROP TABLE IF EXISTS employees;

-- Truncate Table
TRUNCATE TABLE employees;
```
`DROP TABLE` removes the table structure and data, whereas `TRUNCATE TABLE` removes all rows but keeps the table structure.
```

7. SQL Query to Copy, Duplicate, or Backup a Table

Sometimes, while performing operations on a table, we might need to create a table backup. We can create a backup of the table by creating its copy or duplicate.
We can create a backup of a table by creating a duplicate or copy of the original database.

```
CREATE TABLE employees_backup AS SELECT * FROM employees;
```

This statement creates a copy of the `employees` table and its data.

8. Temporary Tables in SQL

Temporary Tables are Created in a TempDB and are automatically deleted as soon as the last connection is terminated. Temporary Tables help us to store and process intermediate results. Temporary tables are very useful when we need to store temporary data.

There are 2 types of Temporary Tables:

1. Local Temporary Table
2. Global Temporary Table.

A **Local Temp Table** is available only for the session that has created it. It is automatically dropped (deleted) when the connection that has created it, is closed.

Unlike some other SQL databases (e.g., SQL Server), **MySQL does not support global temporary tables.** In databases that do support them, a global temporary table is available to all sessions but retains its data separately for each session.

```
CREATE TEMPORARY TABLE temp_employees (
        id INT,
        first_name VARCHAR(50),
        last_name VARCHAR(50)
);
```

This statement creates a temporary table `temp_employees` that exists only for the duration of the session.

11. SQL INSERT INTO Statement

The INSERT INTO statement in SQL is used to add new records to a table in a database. It is a fundamental command for data insertion and is used to insert new data into tables.

```
INSERT INTO employees (first_name, last_name, email, hire_date) VALUES
('Alice', 'Johnson', 'alice.johnson@example.com', '2023-03-10');
```

This statement inserts a single row into the `employees` table.

10. SQL Query to Insert Multiple Rows

Insertion in a table is a DML (Data manipulation language) operation in SQL. When we want to store data we need to insert the data into the database. We use the INSERT statement to insert the data into the database.

```
INSERT INTO employees (first_name, last_name, email, hire_date) VALUES
('John', 'Doe', 'john.doe@example.com', '2023-01-15'),
('Jane', 'Smith', 'jane.smith@example.com', '2023-02-20');
```

This statement inserts multiple rows into the `employees` table.

12. SQL UPDATE Statement

The UPDATE statement in SQL is used to update the data of an existing table in the database. We can update single columns as well as multiple columns using the UPDATE statement as per our requirement.

SQL commands(UPDATE and **DELETE**) are used to change the data that is already in the database. The SQL DELETE command uses a **WHERE** clause.

```
UPDATE employees SET email = 'john.doe@newdomain.com' WHERE first_name = 'John'
AND last_name = 'Doe';
```

This statement updates the email of John Doe in the `employees` table.