

# CPE241 Database Systems

## Week-6 Assignment: SQL Data Definition Language

Submission Deadline: March 10 2024, 23:59

1. Open Command Prompt/Terminal and access MySQL/MariaDB using command line. Login with `root` or any user that has permission to create database.

2. Type `SHOW DATABASES;` to see what databases you have. Screenshot your result. If the database `conference` already exists, please delete it first using command below.

```
In [ ]: DROP DATABASE IF EXISTS conference;
```

3. Create a database named `conference` with `utf8` character set and `utf8_general_ci` collation using the following command. After the command is executed, take a screenshot of the result shown on your screen.

```
In [ ]: CREATE DATABASE conference
        DEFAULT CHARACTER SET utf8
        DEFAULT COLLATE utf8_general_ci;
```

4. Execute `USE conference` to select the database `conference` as the one you are going to operate with.

5. Create a table named `tb_account`. Then, show your result.

```
In [ ]: CREATE TABLE IF NOT EXISTS `tb_account` (
        `_id` int(11) NOT NULL AUTO_INCREMENT,
        `email` text NOT NULL,
        `password` text NOT NULL,
        `code` text NOT NULL,
        `title_name` text NOT NULL,
        `first_name` text NOT NULL,
        `middle_name` text,
        `last_name` text NOT NULL,
        `address` text,
        `city` text,
        `postal_code` text,
        `country` text NOT NULL,
        PRIMARY KEY (`_id`)
    );
```

6. Change the name of attribute `_id` to `account_id` by using the following command.

```
In [ ]: ALTER TABLE tb_account CHANGE _id account_id int(11);
```

7. Next, create another table named `tb_paper`. Set `_id` as the primary key of the table. Show your `CREATE` statement and screenshot of the result.

Field	Type	Null	Key	Default	Extra
_id	int(11)	NO	PRI	NULL	
account_id	int(11)	NO		NULL	
created	timestamp	YES		current_timestamp()	
modified	timestamp	YES		0000-00-00 00:00:00	
category	int(11)	NO		NULL	
title	text	NO		NULL	
author	text	NO		NULL	
present	int(11)	NO		NULL	
correspond	int(11)	NO		NULL	
emailcorr	text	NO		NULL	
abstract	text	NO		NULL	
status	int(11)	NO		NULL	

8. Next change the name of `_id` to `paper_id` . Show your statement and result.

9. Add `account_id` as another primary key by using an `ALTER TABLE` statement. What is the result? and why?  
For example, `ALTER TABLE` statement for adding a primary key is

```
In [ ]: ALTER TABLE table_name ADD PRIMARY KEY (column_name);
```

10. Next, try to use following command, show your result. What is it used for?

```
In [ ]: ALTER TABLE tb_paper DROP PRIMARY KEY, ADD PRIMARY KEY(paper_id, account_id);
```

11. Use a `SHOW COLUMNS` statement to show the list of columns in `tb_account` . Show your result.  
Example statement:

```
In [ ]: SHOW COLUMNS from table_name;
```

12. Use `SHOW COLUMNS` again to show the information of the columns in `tb_paper` . Show captured screen. What is the primary key of the table?

13. Delete `modified` , `present` , and `correspond` . Show your command and captured screen of the table after deleting columns.  
Example statement:

```
In [ ]: ALTER TABLE table_name DROP COLUMN column_name;
```

14. Change the data type of `status` from `int(11)` to `Boolean` . Show your command and captured screen. Note that the `status` column is used for storing either `0` (False) or `1` (True).  
Example statement:

```
In [ ]: ALTER TABLE table_name MODIFY column_name new_type NOT NULL;
```

15. After Step 14, what is the datatype of the `status` column? Why is the data type of the column not Boolean? Is it acceptable or valid for storing the status information?

16. Execute following statement. Show your result. What is the command used for?

```
In [ ]: ALTER TABLE tb_paper ADD comment VARCHAR(60) AFTER abstract;
```

17. Create a table named `tb_category` with two attributes; `category_id` and `category` . The primary key is `category_id` and its data type is `int(11)` with `AUTO_INCREMENT` . And the data type of `category` is `TEXT` . Show your `CREATE` statement.

18. Add the list of categories as shown in the figure below into `tb_category` using an `INSERT` statement. Show your command.

category
Chemistry
Biology
Physics
Mathematics
Computing and Information Technology

19. Show all the data in `tb_category` by using `SELECT * FROM tb_category;`

20. Add a foreign key constraint on the `category` column of `tb_paper` table which points to the `category_id` column of the `tb_category` table by using `ALTER TABLE` statement.  
Example statement:

```
In [ ]: ALTER TABLE table_name ADD FOREIGN KEY  
(column_name) REFERENCES  
reference_table(reference_column);
```

---

**21.** Add a foreign key constraint on the `account_id` column of `tb_paper` table which points to the `account_id` column of `tb_account` table. Show your statement and captured result.

**22.** Execute statement below. Show your result. What is it used for? What is the constraint name of the foreign key of `tb_paper` table?

```
In [ ]: SELECT COLUMN_NAME, CONSTRAINT_NAME, REFERENCED_COLUMN_NAME, REFERENCED_TABLE_NAME
        FROM information_schema.KEY_COLUMN_USAGE
        WHERE TABLE_NAME = 'tb_paper';
```

**23.** Use the following statement. Note that you have to replace `constraint_name` with the actual constraint name of the foreign key of `tb_paper` table found in Step 22. Show your result.

```
In [ ]: ALTER TABLE tb_paper
        DROP FOREIGN KEY constraint_name,
        ADD CONSTRAINT `fk_account_id` FOREIGN KEY (`account_id`) REFERENCES `tb_account` (`account_id`);
```

**24.** Then, repeat the same statement in Step 22 again. Do you see any changes? What is it? What is the statement in Step 23 used for?

**25.** Insert any dummy data into the tables `tb_account` and `tb_paper` by using `INSERT` statement.

```
In [ ]: INSERT INTO table_name (column1, column2, column3, ...) VALUES (value1, value2, value3, ...);
```

**26.** Then, use `SELECT * FROM tb_account;` Do you see your inserted data?

**27.** Delete the database. Show your captured screen

---