# AD688 Assignment 2 - SQL

## Assignment Instructions

Assignment 2 contains 5 sections. You are required to try out all the SQL statements in all the questions, and take a screenshot of the resulting table for all questions (You do not need to paste screenshots for questions 1 and 2 in Section 1). Solutions to some of the problems have been provided for assistance.

Submit the SQL statements for each section in a word document, along with the corresponding screenshots. When taking screenshots, you will need to capture only the Output section.

Tip: For capturing screenshots in Mac, command+Shift+4. Use Snipping Tool in Windows.

**Section 1: BASIC SQL COMMANDS- INSERT, SELECT, ALTER**

1. Using SQL, create the following tables for a Town Library

- Book

- Patron

- Loan

Book:

**CREATE TABLE** Book (call\_no INT NOT NULL, title varchar(50) NOT NULL, subject varchar(50) NOT NULL);

Patron:

**CREATE TABLE** Patron (user\_id INT NOT NULL, name varchar(50) NOT NULL, age INT NOT NULL);

Loan:

**CREATE TABLE** Loan (call\_no INT NOT NULL, user\_id INT NOT NULL, fine INT, paid varchar(50));

2. Add the following sample records into the tables using **INSERT INTO**

**INSERT INTO** Book **VALUES** (100,'Physics Handbook','Physics'), (200,'Database Systems','Computing'), (300,'Modula-2','Computing'), (400,'Database Design','Computing'), (500,'Software Testing','Computing'), (600,'Business Society','Business'), (700,'Graphs','Mathematics'), (800,'Cell Biology','Biology'), (900,'Set Theory', 'Mathematics');

**INSERT INTO** Patron VALUES (100,'Wong',22), (150,'Colin',31), (200,'King',21), (250,'Das',67), (300,'Niall',17), (350,'Smith',72), (400,'Jones',41);

**INSERT INTO** Loan **VALUES** (100,100,null,'yes'), (300,100,null,null), (900,200,1.90,'yes'), (400,200,16.30,'yes'), (600,200,16.30,'yes'), (500,250,null, null), (600,250,36.5,'yes'), (700,300,null,null), (800,350,2.90,'yes'), (900,400,null,null);

3. Display the structure of the tables that you have just created using **SELECT** command. Paste a screenshot of all three tables.

4. Write SQL statements to modify the Patron table to include a new 'address' column which has character field of size 30. (Use ALTER and ADD commands)

5. Copying Tables: In one statement, create a table Seniors and copy the records from Patron where age > 65.

6. Write SQL statement to list all the titles of books in the database

7. Write SQL statement to list title and subject for each book

8. Write SQL statement to display the unique subjects

9. Write SQL statement to list titles of Mathematics books

10. List the book with call number 300

**Section 2: OPERATORS-ARITHMETIC, BOOLEAN, LIKE, BETWEEN, IS NULL, IN**

1. List patrons fines in British Pounds along with user\_id and call\_no (assume 1 pound = 2 dollars)

2. List loans where the fine is over 10 British pounds

3. List call numbers of books borrowed by patron 200 or patron 250 where the fine paid is greater than $2.

4. List books with 'Database' in the title (**LIKE**)

5. List books with title having an 'o' as second character.

6. List books with call numbers between 200 and 400 (**BETWEEN**)

7. To find the patrons who have not paid the fine. (**IS NULL**)

8. Listing the patrons along with the fine paid. (**IS NOT NULL**)

9. List names of patrons whose user\_id is 100, 200, 300 or 350.

(**IN)**

10. List all Computing and History titles using **IN** operator.

**SECTION 3: AGGREGATE FUNCTIONS - MAX, MIN, SUM, COUNT**

1. List the largest fine paid for an overdue book, using **MAX** function
2. List the least fine paid for an overdue book, using **MIN** function
3. How much has the library collected in fines? (**SUM**)
4. What is the average fine collected by the Library? (**AVG**)
5. How many books are there in the library? (**COUNT**)
6. How many times has a fine been collected? (**COUNT**)
7. Count the number of Computing books.
8. How many subject areas are there? (**DISTINCT**)

**SECTION 4: Ordering and Grouping- ORDER BY, GROUP BY**

1. List the books in alphabetical order by title. (**ORDER BY**)

ASC

DESC

2. List books in subject order, and, with each subject, order them by call number.

3. For each patron, list the total fines paid and group the list by GROUP BY function. (**GROUP BY**)

4. List the patron IDS for those who have paid more the $30 in fines on books with call numbers greater than 400. (**HAVING**)

**Section 5: JOINS**

1. List the names of patrons, their IDs, and the call numbers of the books they have borrowed. Capture a screenshot and paste it in your submission. (**JOINS**)

**SECTION 6: DROP TABLE, DATA MANIPULATION - UPDATE, DELETE, INSERT**

1. Write an SQL statement to create a sample table with the columns **():**

Sample Table:

Sample\_ID INT NOT NULL,

Sample\_Name varchar(60);

2. Create Write SQL statement that will delete the Sample table.

3. Write an SQL statement to increase the age of all patrons by 1. Select the new table and capture the screenshot. (**UPDATE**)

4. Delete all books with Computing as subject. Select the new table and capture the screenshot. (**DELETE**)

5. A patron named King has left the library. Using **DELETE**, remove all his loan records. Select the new table and capture the screenshot.

6. Add a new patron named Thomas with user\_id=900 and age=34 into Patron table. Select the new table and capture the screenshot. (**INSERT**)