Pop quiz 1 (CSC 330) – 27 February 2024

[Ungraded + Anonymous + Memory-only + Self-graded]

1. Open an SQLite database `test.db` from the command line with `header` and `column` table display options enabled (5 points).

```
sqlite3 -header -column test.db
```

2. On the SQLite console, check which database you're using, and which tables the database contains (5 points).

```
.databases
.tables
```

```
main: /home/marcus/GitHub/admin/RoamNotes/test.db r/w
student
```

3. Which header arguments does an Emacs Org-mode code block need to execute SQLite commands and print tables with header and columns to the screen? Complete the block below (10 points).

```
#+begin_src sqlite :db test.db :header :column :results output

#+end_src
```

4. Create a table named `student` with an field `id` for the student ID, a field for the student's chosen major, and a field for enrolment (0 or 1). The ID should be a `PRIMARY KEY`, the student must choose a major, and the student is enrolled by default (1) - (10 points).

```
CREATE     TABLE student (
        id INTEGER PRIMARY KEY,
        major TEXT NOT NULL,
        enrolled INTEGER DEFAULT 1);
```

5. Check the table design (5 points).

```
.schema
```

```
CREATE TABLE student (
       id INTEGER PRIMARY KEY,
       major TEXT NOT NULL,
       enrolled INTEGER DEFAULT 1);
```

6. Insert three student records in `student` (10 points):
    1. The first student studies 'data science'.
    2. The second student studies 'computer science'.
    3. The third student studies 'Mathematik'.

```
INSERT INTO student (major) VALUES ('data science');
INSERT INTO student (major) VALUES ('computer science');
INSERT INTO student (major) VALUES ('Mathematik');
```

7. Print all rows of the column that shows the student's major and the student ID (in this order), as `MAJOR` and `ID` (5 points).

```
SELECT major AS 'MAJOR', id AS 'ID' FROM student;
```

```
MAJOR             ID
----------------  --
data science      1
computer science  2
mathematics       3
data science      4
computer science  5
Mathematik        6
```

8. You notice a mistake: a German must have done the data entry. The 3rd student's major is not 'Mathematik' but 'mathematics'. Correct the entry, and then print the `MAJOR` and `ID` columns again (10 points).

```
UPDATE student SET major='mathematics' WHERE id=3;
SELECT major AS 'MAJOR', id AS 'ID' FROM student;
```

```
MAJOR             ID
----------------  --
data science      1
computer science  2
mathematics       3
data science      4
computer science  5
Mathematik        6
```

9. In the SQLite console (10 points):
   1. create a directory `SQLite` using the command: `mkdir -v SQLite`
   2. check with the command: `ls - SQLite` (or `DIR SQLite` in Windows)

   Save the table data to a file `student.sql` in the current directory

```
.shell mkdir -v SQLite
.shell ls -l SQLite
```

```
total 4
-rw-rw-r-- 1 marcus marcus 320 Feb 26 22:38 student.sql
```

10. Save your database as `SQLite/student.sql`, then check success by using the `ls` (or the `DIR`) command again to list the file (10 points).

```
.output SQLite/student.sql
.dump
.output stdout
.shell ls -l SQLite
```

```
total 4
-rw-rw-r-- 1 marcus marcus 466 Feb 26 22:54 student.sql
```

11. Delete your table (if it exists) on the SQLite console and check that it was deleted (10 points).

```
.tables
DROP TABLE IF EXISTS student;
.tables
```

```
student
```

12. Import the SQL file from `SQLite/student.sql` and check that the table was recreated and that the content is intact (10 points).

```
.read SQLite/student.sql
.tables
SELECT * FROM student;
```

```
student
id  major             enrolled
--  ----------------  --------
1   data science      1
2   computer science  1
3   mathematics       1
4   data science      1
5   computer science  1
6   Mathematik        1
```

Once you're finished writing out all commands, create an Org-mode file and enter the code in code blocks (one for each section). Grade yourself and enter the total points here, then return the sheet:

____ / 100