LAB

Important notes

> Databases = ERD (Entity-Relationship Diagram)

> Tables = Entities

> Each each column/field = Attributes

> Each row/data = Record

Query to create database

> CREATE DATABASE database_name;

```
CREATE DATABASE lab;
```

Query to switch to the created database

> USE database_name;

```
USE lab;
```

Query to create the student table:

> CREATE TABEL table_name(attributes)

- > in this case the table name is **student** and the attributes are **sid**, **first_name**, **last_name**, **age**, **grade**, **major**; so...

> CREATE TABEL student(sid, first_name, last_name, age, grade, major);

```
CREATE TABLE student(
  sid INT PRIMARY KEY,
  first_name VARCHAR(20) NOT NULL,
  last_name VARCHAR(20) NOT NULL,
  age INT NOT NULL check(age > 18),
  grade VARCHAR(2) NOT NULL,
  major VARCHAR(20) NOT NULL
);
```

Constraints explanation

- `sid INT PRIMARY KEY` => `INT` because **id** has to be integer; `PRIMARY KEY` because **id** has to be unique and `NOT NULL`;
- `first_name VARCHAR(20) NOT NULL` => `VARCHAR(20)` because **first_name** must be a string and the maximum number of characters allowed is 20; `NOT NULL` because **first_name** must have a value, it cannot be left empty;
- `last_name VARCHAR(20) NOT NULL` => `VARCHAR(20)` because **last_name** must be a string and the maximum number of characters allowed is 20; `NOT NULL` because last_name must have a value, it cannot be left empty;
- `age INT NOT NULL check(age > 18)` => `INT` because **age** has to be integer; `NOT NULL` because **age** must have a value, it cannot be left empty; `check(age > 18)` ensures the value in the **age** column must be greater than 18, so only students aged 19 and above can be inserted;
- `grade VARCHAR(2) NOT NULL` => `VARCHAR(2)` because **grade** must be a string and the maximum number of characters allowed is 2; `NOT NULL` because **grade** must have a value, it cannot be left empty;
- `major VARCHAR(20) NOT NULL` => `VARCHAR(20)` because **major** must be a string and the maximum number of characters allowed is 20; `NOT NULL` because **major** must have a value, it cannot be left empty.

Query to insert data into the student table

> INSERT INTO name_table (attribuites_names) VALUES (data for each field/attribuite)

```sql
INSERT INTO student (sid, first_name, last_name, age, grade, major) VALUES
(1001, 'John', 'Doe', 20, 'B', 'Computer Sci'),
  -- for insert more then 1 row continue after the ','
  (1002, 'Jane', 'Smith', 21, 'A', 'Mathematics'),
  (1003, 'Emily', 'Johnson', 22, 'A-', 'Physics'),
  (1004, 'Michael', 'Brown', 19, 'B+', 'Chemistry'),
  (1005, 'Sarah', 'Davis', 23, 'A', 'Biology'),
  (1006, 'David', 'Wilson', 20, 'C+', 'English'),
  (1007, 'Laura', 'Martinez', 21, 'B', 'History'),
  (1008, 'Robert', 'Garcia', 22, 'B-', 'Economics'),
  (1009, 'Mary', 'Lee', 19, 'A', 'Political Sci'),
  (1010, 'James', 'Walker', 23, 'A-', 'Sociology'),
  (1011, 'Olivia', 'White', 20, 'B+', 'Psychology'),
  (1012, 'William', 'Harris', 21, 'A-', 'Philosophy'),
  (1013, 'Sophia', 'Clark', 22, 'A', 'Computer Sci'),
  (1014, 'Benjamin', 'Lewis', 19, 'B', 'Mathematics'),
  (1015, 'Isabella', 'Robinson', 23, 'A', 'Physics'),
  (1016, 'Ethan', 'Young', 20, 'C', 'Chemistry'),
  (1017, 'Mia', 'Hall', 21, 'B+', 'Biology'),
  (1018, 'Alexander', 'Allen', 22, 'B-', 'English'),
  (1019, 'Charlotte', 'King', 19, 'A-', 'History'),
  (1020, 'Daniel', 'Wright', 23, 'A', 'Economics');
```

Query to return all records inserted in the student table

> SELECT * FROM table_name

```
SELECT * FROM student;
```