1. CREATE TABLES

```
CREATE TABLE Student(
   sID INT AUTO_INCREMENT NOT NULL PRIMARY KEY,
   sName VARCHAR(50) NOT NULL,
   GPA TINYINT
   );

CREATE TABLE Module(
   mCode CHAR(6) NOT NULL PRIMARY KEY,
   mCredits TINYINT NOT NULL DEFAULT 10,
   mTitle VARCHAR(100) NOT NULL
   );
```

>> drop table and re-create it

```
CREATE TABLE Module(
   mCode CHAR(6) NOT NULL,
   mCredits TINYINT NOT NULL DEFAULT 10,
   mTitle VARCHAR(50) NOT NULL,
   CONSTRAINT mod_pk PRIMARY KEY(mCode)
   );
```

2. INSERT DATA

```
INSERT INTO Student
VALUES
(1, 'John', 18);
```

>> general form:

```
INSERT INTO Student
(sID, sName, GPA)
VALUES
(3, 'John', Null);
```

>>partial form:

```
INSERT INTO Student (sID, sName) VALUES (5, 'Mary');
```

>> illustrate auto_increment

```
INSERT INTO Student (sName) VALUES ('Mary');
```

>> more than 1 entries, drop table student;

```
INSERT INTO Student
(sID, sName, GPA)
VALUES
(1, 'John', 18.5),
(2, 'Mary', 19),
(3, 'James', 18),
```

```
(4, 'Amy', 17),
(5, 'John', 18.5),
(6, 'Amy', 18);

INSERT INTO Module
VALUES
('G51DBI', 10, 'Databases and Interfaces'),
('G51PRG', 20, 'Programming'),
('G51IAI', 10, 'Artificial Intelligence'),
('G52ADS', 10, 'Algorithms');
```

3. FOREIGN KEYS
>> we want to associate the data in Student with data in Module.
>> We know that students enrol to more than 1 table.
>> introduce Grade table

```
CREATE TABLE Grade(
   sID INT,
   mCode CHAR(6),
   gMark TINYINT,
   CONSTRAINT gr_pk PRIMARY KEY(sID, mCode),
   CONSTRAINT gr_fk1 FOREIGN KEY (sID)
   REFERENCES Student (sID)
   ON DELETE CASCADE,
   CONSTRAINT enr_fk2 FOREIGN KEY (mCode)
   REFERENCES Module (mCode)
   ON DELETE CASCADE
   );


INSERT INTO Grade
(sID, mCode, gMark)
VALUES
(1, 'G51DBI', 70),
(1, 'G51PRG', 60),
(1, 'G51IAI', 60),
(2, 'G51DBI', 80),
(2, 'G51PRG', 50),
(2, 'G51IAI', 60),
(3, 'G51DBI', 50),
(3, 'G51PRG', 50),
(3, 'G51IAI', 60),
(4, 'G51DBI', 75),
(4, 'G51PRG', 65),
(4, 'G51IAI', 55),
(5, 'G51DBI', 70),
(5, 'G51PRG', 50),
(5, 'G51IAI', 50),
(6, 'G51DBI', 70),
(6, 'G51PRG', 65),
```

(6, 'G51IAI', 55);

DELETE FROM Student WHERE sID = 1;
DELETE FROM Module WHERE mCode = 'G51DBI';

>> Repeat by changing CASCADE to RESTRICT : Now DELETE FROM Student WHERE sID = 1; won't work.

4. UPDATE

UPDATE Student
SET sName = 'Jane', GPA = 15
WHERE sID = 2;

UPDATE Grade
SET gMark = 1.05*gMark;

5. ALTER
>> Add column
ALTER TABLE Student
ADD COLUMN sDegree VARCHAR(64) NOT NULL;

>> Drop Column
ALTER TABLE Student
DROP sDegree;

>> Add column after specific column
ALTER TABLE Student
ADD COLUMN sDegree VARCHAR(64) NOT NULL AFTER sID;

>> Modify column
ALTER TABLE Student
CHANGE COLUMN sDegree
sDegreeNew VARCHAR(128);

>> modify constraint: first drop it, then add new
ALTER TABLE Grage
DROP FOREIGN KEY gr_fk1;

ALTER TABLE Grade
ADD CONSTRAINT gr_fk1 FOREIGN KEY (sID) REFERENCES Student (sID) ON DELETE RESTRICT;

>> now try

DELETE FROM Student WHERE sID = 1;

6. SELECT EXAMPLES

(6.1)

>> Basic Select. Select is really project :)

```
SELECT sName
FROM Student

SELECT sName, GPA
FROM Student
```

>> add Where. Where is select.

```
SELECT sName, GPA
FROM Student
WHERE GPA > 17;
```

>> remove GPA

```
SELECT sName
FROM Student
WHERE GPA > 17;
```

>> remove duplicate

```
SELECT DISTINCT sName
FROM Student
WHERE GPA > 17;
```

>> return all columns

```
SELECT *
FROM Student
WHERE GPA > 17;
```

(6.2) Cartesian Product

```
SELECT Student.sID, sName
FROM Student, Grade
```

%%%%%%%%%%%%%%%%
Data:

```
(1, 'John', 18.5),
(2, 'Mary', 19),
(3, 'James', 18),
(4, 'Amy', 17),
(5, 'John', 18.5),
(6, 'Amy', 18);

('G51DBI', 10, 'Databases and Interfaces'),
('G51PRG', 20, 'Programming'),
('G51IAI', 10, 'Artificial Intelligence'),
('G52ADS', 10, 'Algorithms'),
('G52APR', 10, 'Advanced Programming');
```

(1, 'G51DBI', 70),
(1, 'G51PRG', 60),
(1, 'G51IAI', 60),
(2, 'G51DBI', 80),
(2, 'G51PRG', 50),
(2, 'G51IAI', 60),
(3, 'G51DBI', 50),
(3, 'G51PRG', 50),
(3, 'G51IAI', 60),
(4, 'G51DBI', 75),
(4, 'G51PRG', 65),
(4, 'G51IAI', 55),
(5, 'G51DBI', 70),
(5, 'G51PRG', 50),
(5, 'G51IAI', 50),
(6, 'G51DBI', 70),
(6, 'G51PRG', 65),
(6, 'G51IAI', 55);