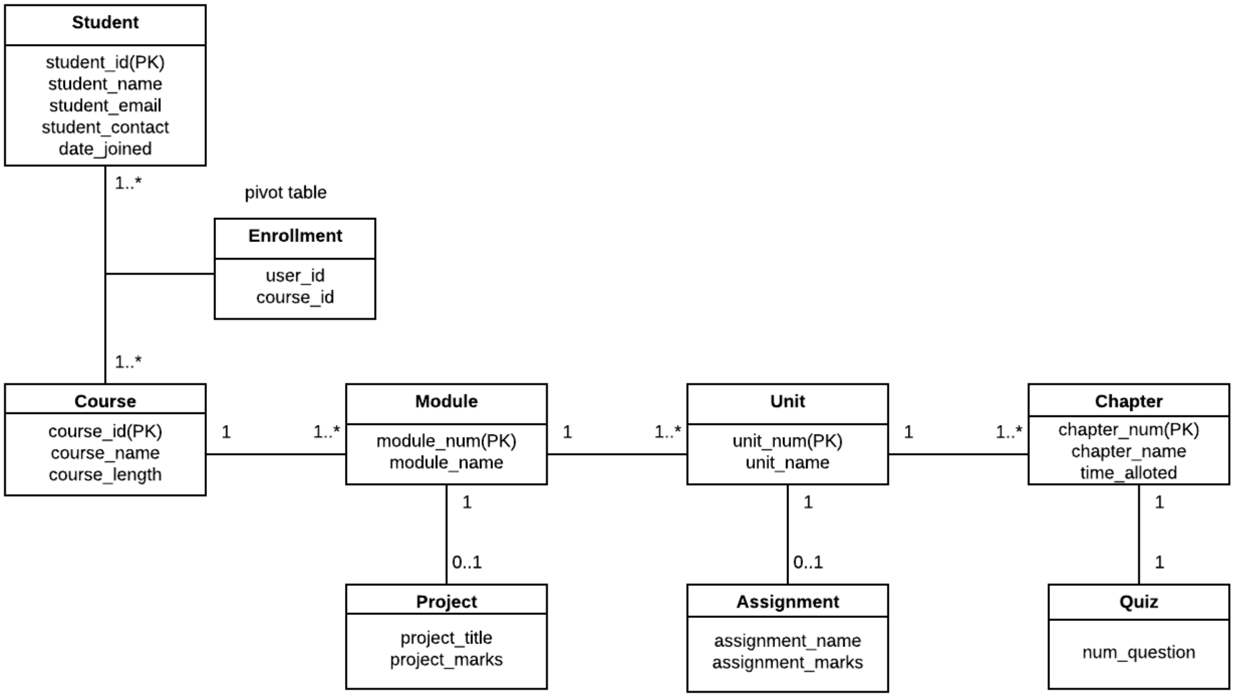
Recall: Fuse AI Database



Connecting to Sqlite Database

```
%load_ext sql
```

```
%sql sqlite://
```

```
'Connected: @None'
```

Minor Differences between sqlite and SQL

```
 * INT  ->  INTEGER
 * auto_increment -> AUTOINCREMENT
```

Create Table

Syntax

```
create table <relation_name>(
    <attribute_name>    <data_type>    <integrity_constraint>
    <attribute_name>    <data_type>    <integrity_constraint>
    <attribute_name>    <data_type>    <integrity_constraint>
    <integrity_constraint>
);
```

Double-click (or enter) to edit

> Create `Student` table

```
# Create table student
%%sql
CREATE TABLE student(
    student_id INTEGER PRIMARY KEY AUTOINCREMENT,
    student_name VARCHAR(50) NOT NULL,
    student_email VARCHAR(50),
    student_contact VARCHAR(15)
);
```

```
 * sqlite://
(sqlite3.OperationalError) table student already exists
[SQL: CREATE TABLE student(
    student_id INTEGER PRIMARY KEY AUTOINCREMENT,
    student_name VARCHAR(50) NOT NULL,
    student_email VARCHAR(50),
    student_contact VARCHAR(15)
);]
(Background on this error at: http://sqlalche.me/e/14/e3q8)
```

> Create `Course` table

```
# Create course table

%%sql
CREATE TABLE course(
    course_id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    course_name VARCHAR(50) NOT NULL,
    course_length INTEGER
);
```

```
 * sqlite://
Done.
```

```
[]
```

> Add other tables

```
# Create some more tables

%%sql
CREATE TABLE enrollment(student_id INTEGER, course_id INTEGER, FOREIGN KEY(student_id) REFER
CREATE TABLE module(course_id INTEGER, module_num INTEGER, module_name VARCHAR(50) NOT NULL,
CREATE TABLE unit(course_id INTEGER, module_num INTEGER, unit_num INTEGER, unit_name VARCHAR
CREATE TABLE chapter(course_id INTEGER, module_num INTEGER, unit_num INTEGER, chapter_num IN
CREATE TABLE project(course_id INTEGER, module_num INTEGER, project_title VARCHAR(50) NOT NU
CREATE TABLE assignment(course_id INTEGER,module_num INTEGER,unit_num INTEGER, assignment_na
CREATE TABLE quiz(course_id INTEGER, module_num INTEGER, unit_num INTEGER, chapter_num INTEG
```

```
 * sqlite://
Done.
Done.
Done.
Done.
Done.
Done.
Done.
[]
```

Insertion

Syntax:

```
Insert into <table_name> Values('val1','val2',...,'valn')
```

> Insert values into `Student` table

```
# Insert values into Student table

%%sql
INSERT INTO student ( student_name, student_email)
VALUES ('John Doe', 'john@fusemachines.com'),
('Dummy Person', 'dummy@fusemachines.com'),
('Alison Burgers','alison@fusemachines.com'),
('Harka Bahadur','harke@fusemachines.com.np'),
('Ful Kumari','fulkumari@fusemachines.com.np'),
('Chulbul Pandey','chulbul@fusemachines.com'),
('Bir Kaji Sherchan','birkaji@fusemachines.com.np'),
('Maiya Gauchan','maiya@fusemachines.com.np');
```

```
 * sqlite://
8 rows affected.
[]
```

Insert values into `course` table

```
# Insert values into Course table

%%sql
INSERT INTO course (course_name,course_length)
VALUES ('Fundamentals of Computer Science',3),
('Fundamentals of Mathematics',4),
('Machine Learning',5),
('Deep Learning',5),
('Computer Vision',5),
('Natural Language Processing',5);
```

```
 * sqlite://
6 rows affected.
[]
```

Insert values into other tables

```
# Insert values into other tables

%%sql
INSERT INTO enrollment (student_id, course_id) VALUES (1,1),(1,2),(2,1),(3,1),(4,2),(5,1),(6
INSERT INTO module(course_id, module_num, module_name) VALUES (1,1,'Introduction to the Cour
INSERT INTO unit(course_id, module_num, unit_num, unit_name, time_allotted) VALUES (1,1,1,'I
INSERT INTO chapter(course_id, module_num, unit_num, chapter_num, chapter_name) VALUES (1,1,
INSERT INTO project(course_id, module_num, project_title, project_marks) VALUES (2,1,'None',
INSERT INTO assignment(course_id, module_num, unit_num, assignment_name, assignment_marks) V
```

```
 * sqlite://
8 rows affected.
12 rows affected.
31 rows affected.
10 rows affected.
3 rows affected.
15 rows affected.
[]
```

SELECT Statement

Syntax:

```
SELECT "column_name" FROM "table_name"
SELECT * FROM "table_name"
SELECT DISTINCT "column_name" FROM "table_name"
```

- An Asterisk(*) in the select clause denotes all attributes

> Q. Select all tuples from course relation

```
%sql SELECT * FROM course;
```

```
 * sqlite://
Done.
```

course_id	course_name	course_length
1	Fundamentals of Computer Science	3
2	Fundamentals of Mathematics	4
3	Machine Learning	5
4	Deep Learning	5
5	Computer Vision	5
6	Natural Language Processing	5

> Q. Select all tuples from module where course_id = 1

```
%sql SELECT * FROM module WHERE course_id = 1;
```

```
 * sqlite://
Done.
```

course_id	module_num	module_name
1	1	Introduction to the Course
1	2	Basics of Computer Systems
1	3	Python Programming
1	4	Data Structures and Algorithms
1	5	Database
1	6	Building Applications

> Q. Select all tuples from a unit of course_id = 1 and module_num = 2

```
%sql SELECT * FROM unit WHERE course_id = 1 AND module_num = 2;
```

```
 * sqlite://
Done.
course_id module_num unit_num           unit_name             time_allotted
1         2          1        Introduction to the Module          0.0
1         2          2        Digital Information & Digital Logic 1.25
1         2          3        Basics Computer Architecture        2.0
1         2          4        Basics of Linux Operating System    1.5
1         2          5        Basics of Computer Networks         1.5
1         2          6        Module Summary                      0.0
```

Start coding or generate with AI.

> Q.Select only unit name and time allotted for a unit of first course second module

```
%sql SELECT unit_name, time_allotted FROM unit WHERE course_id = 1 AND module_num = 2;
```

```
 * sqlite://
Done.
           unit_name              time_allotted
Introduction to the Module          0.0
Digital Information & Digital Logic 1.25
Basics Computer Architecture        2.0
Basics of Linux Operating System    1.5
Basics of Computer Networks         1.5
Module Summary                      0.0
```

> Q. Display name of courses whose course length is between 3 to 4 months

```
%sql SELECT course_name, course_length FROM course WHERE course_length BETWEEN 3 AND 4;
```

```
 * sqlite://
Done.
           course_name              course_length
Fundamentals of Computer Science    3
Fundamentals of Mathematics         4
```

> A select clause can contain arithmetic expression involving +,-,*,/ operations. We can also rename a table or a column temporarily by giving another name using 'as' clause known as Alias.

```
%sql SELECT course_name, course_length*4 AS course_length_in_weeks FROM course;
```

```
 * sqlite://
Done.
          course_name              course_length_in_weeks
Fundamentals of Computer Science   12
Fundamentals of Mathematics        16
Machine Learning                   20
Deep Learning                      20
Computer Vision                    20
Natural Language Processing        20
```

> Q. List out the assignment names of all the assignments that have high weightage(greater than 100 marks)

```
%sql SELECT * FROM assignment WHERE assignment_marks > 100;
```

```
 * sqlite://
Done.
course_id module_num unit_num assignment_name assignment_marks
2         2          7        Eigen Faces     110
2         3          7        N/A             110
```

SQL includes a string matching operator for comparisions on character string.

- percent(%) character matches any substring
- underscore(_) matches any character

> Q. Find the name of all courses that are Fundamental(have Fundamental in their name)

```
%sql SELECT * FROM course WHERE course_name LIKE '%Fundamental%';
```

```
 * sqlite://
Done.
```

course_id	course_name	course_length
1	Fundamentals of Computer Science	3
2	Fundamentals of Mathematics	4

Students of fusemachines from Nepal have country domain `.np` in their email id. For Example: `buddha@fusemachines.com.np`

> Q.List down all the students from Nepal.

```
%sql SELECT * FROM student WHERE student_email LIKE '%.np%';
```

```
 * sqlite://
Done.
```

student_id	student_name	student_email	student_contact
4	Harka Bahadur	harke@fusemachines.com.np	None
5	Ful Kumari	fulkumari@fusemachines.com.np	None
7	Bir Kaji Sherchan	birkaji@fusemachines.com.np	None
8	Maiya Gauchan	maiya@fusemachines.com.np	None

> Alter Table

Syntax: `Alter table <table_name> add <attribute_name> <data_type>`

```
%sql ALTER TABLE course ADD course_difficulty VARCHAR(20)
```

```
 * sqlite://
Done.
[]
```

```
%sql SELECT * FROM course
```

```
 * sqlite://
Done.
```

course_id	course_name	course_length	course_difficulty
1	Fundamentals of Computer Science	3	None
2	Fundamentals of Mathematics	4	None
3	Machine Learning	5	None
4	Deep Learning	5	None
5	Computer Vision	5	None
6	Natural Language Processing	5	None

```
%sql UPDATE course SET course_difficulty = 'easy' WHERE course_id == 1 OR course_id == 2;
```

```
 * sqlite://
2 rows affected.
[]
```

```
%sql SELECT * FROM course
```

```
 * sqlite://
Done.
course_id            course_name            course_length course_difficulty
1         Fundamentals of Computer Science 3              easy
2         Fundamentals of Mathematics      4              easy
3         Machine Learning                 5              None
4         Deep Learning                    5              None
5         Computer Vision                  5              None
6         Natural Language Processing      5              None
```

Delete Table

Syntax: `Drop table <table_name>`

```
%sql drop table course;
```

```
 * sqlite://
Done.
[]
```

```
%sql select * from course;
```

```
 * sqlite://
(sqlite3.OperationalError) no such table: course
[SQL: select * from course;]
(Background on this error at: http://sqlalche.me/e/e3q8)
```

Start coding or generate with AI.