

9/11/19

CS157a HW1

1)

a)

i) Student

- 1) Name
- 2) StudentID
- 3) Class
- 4) Major

ii) Course

- 1) CourseName
- 2) CourseID
- 3) Credits
- 4) Dept

iii) Grade

- 1) StudentID
- 2) CourseID
- 3) Grade

b)

i) Student

- 1) (Brown, 8, 2, CS)
- 2) (Smith, 17, 1, MATH)

ii) Course

- 1) (Intro CS, 46, 3, CS)
- 2) (Discrete Math, 42, 3, MATH)

iii) Grade

- 1) (8, 46, A)
- 2) (8, 42, A-)
- 3) (17, 42, C)

c)

i) Student

- 1) (Brown, 8, 2, CS) \rightarrow Components = Brown,8,2,CS

ii) Course

- 1) (Intro CS, 46, 3, CS) \rightarrow Components = Intro to CS,46,3,CS

iii) Grade

- 1) (8, 46, A) \rightarrow Components = 8,46,A

d)

- i) Student(Name:string, StudentID:integer, Class:integer, Major:string)
- ii) Course(CourseName:string, CourseID:integer, Credits:integer, Dept:string)
- iii) Grade(StudentID:integer, CourseID:integer, Grade:string)

```
Student(
  Name:string,
  StudentID:integer,
  Class:integer,
  Major:string
)
Course(
  CourseName:string,
  CourseID:integer,
  Credits:integer,
  Dept:string
)
Grade(
  StudentID:integer,
  CourseID:integer,
  Grade:string
)
```

e)

f)

- i) Name \rightarrow string
 - ii) StudentID \rightarrow Integer
 - iii) Class \rightarrow integer
 - iv) Major \rightarrow string
 - v) CourseName \rightarrow string
 - vi) CourseID \rightarrow integer
 - vii) Credits \rightarrow integer
 - viii) Dept \rightarrow string
 - ix) StudentID \rightarrow integer
 - x) CourseID \rightarrow integer
 - xi) Grade \rightarrow string
- g) Each relation can be represented in JSON format

Student

```
{ Name:"Brown", StudentID:8, Class:2, Major:"CS" },
{ Name:"Smith", StudentID:17, Class:1, Major:"MATH" }
```

Course

```
{ CourseName:"Intro CS", CourseID:46, Credits:3, Dept:"CS" }
{ CourseName:"Discrete Math", CourseID:42, Credits:3, Dept:"MATH" }
```

Grade

```
{ StudentID:8, CourseID:46, Grade:"A" }  
{ StudentID:8, CourseID:42, Grade:"A-" }  
{ StudentID:17, CourseID:42, Grade:"C" }
```

2) There could be a relation called "Professor" that lists every professor along with the class they teach as well as other attributes such as EmployeeID and Dept

3) For Student, since ID should be unique, we could make that the key. For Course, since CourseID should be unique, we could use that as a key. For Grade, since there is no unique attribute, we would need to create an additional attribute for indexing and use that as a key

4)

- Student Input
 - CREATE TABLE Student(Name VARCHAR(20), StudentID INTEGER, Class INTEGER, Major VARCHAR(20));
 - INSERT INTO Student VALUES("Brown", 8, 2, "CS");
 - INSERT INTO Student(Name, StudentID, Class, Major) VALUES ("Smith", 17, 1, "MATH");
 - SELECT * FROM STUDENT WHERE true;
- Student Output
 - Brown|8|2|CS
 - Smith|17|1|MATH
- Course Input
 - CREATE TABLE Course(CourseName VARCHAR(20), CourseID INTEGER, Credits INTEGER, Dept VARCHAR(10));
 - INSERT INTO Course VALUES ("Intro CS", 46, 3, "CS");
 - INSERT INTO Course VALUES ("Discrete Math", 42, 3, "MATH");
 - select * from Course where true;
- Course Output
 - Intro CS|46|3|CS
 - Discrete Math|42|3|MATH
- Grade Input
 - CREATE TABLE Grade(StudentID INTEGER, CourseID INTEGER, Grade VARCHAR(2));
 - INSERT INTO Grade VALUES (8, 46, "A");
 - INSERT INTO Grade VALUES (8, 42, "A-");
 - INSERT INTO Grade VALUES (17, 42, "C");
 - select * from Grade where true;
- Grade Output
 - 8|46|A
 - 8|42|A-
 - 17|42|C

