Zaid Bhujwala, Andrew Figueroa, Jonathan Middleton, Bruno DaSilva Sean Murthy

Data Science & Systems Lab (DASSL), Western Connecticut State University (WCSU)

© 2017- DASSL. ALL RIGHTS RESERVED.

Licensed to others under CC 4.0 BY-SA-NC:

https://creativecommons.org/licenses/by-nc-sa/4.0/

PROVIDED AS IS. NO WARRANTIES EXPRESSED OR IMPLIED. USE AT YOUR OWN RISK.

## Logical Schema

**Course**(Name, Title\*)

**Season**(Order, Name\*, Code\*)

**Term**(ID, Year\*, Season\*, StartDate\*, EndDate\*)  
 Season → Season.Order

**Instructor**(ID, FirstName\*, MiddleName, LastName\*, Department\*, Email)

**Section**(ID, Term\*, Course\*, SectionNumber\*, CRN\*, Title\*, Schedule, StartDate, MidtermDate, EndDate, LocationTaught)  
 Term → Term.ID  
 Course → Course.Name

**Section\_Instructor**(Section, Instructor)  
 Section → Section.ID  
 Instructor → Instructor.ID

**Section**(ID, Term, Course\*, SectionNumber\*, CRN\*, Schedule, StartDate, EndDate, Location, Instructor1\*, Instructor2, Instructor3)  
 Term → Term.ID  
 Course → Course.Name  
 Instructor1 → Instructor.ID  
 Instructor2 → Instructor.ID  
 Instructor3 → Instructor.ID

**Grade**(LetterGrade, GPA\*)

**Section\_GradeTier**(Section, LetterGrade, LowPercent\*, HighPercent\*)  
 LetterGrade → Grade.LetterGrade  
 Section → Section.ID

**Student**(ID, FirstName\*, MiddleName, LastName\*, SchoolIssuedID\*, Email, Major, Standing\*)

**Enrollee**(Student, Section**,** EnrollmentDate\*, EnrollmentYear\*, EnrollmentMajor\* MidtermWeightedAggregate, MidtermGradeComputed, MidtermGradeAwarded, FinalWeightedAggregate, FinalGradeComputed, FinalGradeAwarded)  
 Student → Student.ID  
 Section → Section.ID   
 Section, MidtermGradeComputed → Section\_GradeTier.LetterGrade  
 Section, MidtermGradeAwarded → Section\_GradeTier.LetterGrade  
 Section, FinalGradeComputed → Section\_GradeTier.LetterGrade  
 Section, FinalGradeAwarded → Section\_GradeTier.LetterGrade

**AttendanceStatus**(Status, Description\*)

**AttendanceRecord**(Date, Student, Section, Status\*)  
 Student → Student.ID  
 Section → Section.ID  
 Status → AttendanceStatus.Status

**Section\_AssessmentComponent**(Name, Section, Weight\*, NumAssessments\*)  
 Section → Section.ID

**Section\_AssessmentItem**(Section, Component, AssessmentNumber, BasePointsPossible\*, ExtraCreditPointsPossible\*, AssignedDate, DueDate)  
 Section → Section\_AssessmentComponent.Section  
 Component → Section\_AssessmentComponent.Name

**Enrollee\_AssessmentItem**(Student, Section, Component, AssessmentNumber, BasePointsEarned, ExtraCreditPointsEarned, SubmissionDate, Penalty)  
 Student → Student.ID  
 Section → Section.ID  
 Component → Section\_AssessmentComponent.Name  
 AssessmentNumber → Section\_AssessmentItem.AssessmentNumber

## Relations in Detail

### Course

This table is used to hold a list of courses offered by the educational institution. This is distinct from sections of a course, which are actual instances of an offering of a course and are stored in a separate relation. Each course has a number that is used to identify the course. Formatting of this number is likely to vary from institution to institution, however, a common format is a value such as CS305, where the first part is an abbreviation of a department, and the second a number that indicates a numbering of courses within the department (with the first digit being a rough indicator of course difficulty). Title is a title that corresponds to the course that is used for sections of the course.

#### Summary

|  |  |
| --- | --- |
| **Table** | Course |
| **ER origin** | Entity Course |
| **Primary key** | Number |
| **Foreign keys** | <None> |
| **Constraints** | <None> |

#### Attributes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Range** | **Required/Optional** | **Derived/Stored** |
| Name | VARCHAR(8) |  | Required | Stored |
| Title | VARCHAR(100) |  | Required | Stored |

### Season

This table represents a season during a time of year during which classes are held. A season may have many terms, such as 8-week classes during Spring and Fall, and Summer sessions. Order determines the order of the seasons. Name denotes the name of each season (Spring, Summer, Fall, Winter). Code is used as a shorthand for the season names. Since Spring and Summer both start with an ‘S’, ‘S’ would denote Spring, ‘M’ denotes Summer. Additionally, ‘F’ for Fall and ‘W” for Winter.

#### Summary

|  |  |
| --- | --- |
| **Table** | Season |
| **ER origin** | Entity Season |
| **Primary key** | Order |
| **Foreign keys** | <None> |
| **Constraints** | <None> |

#### Attributes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Range** | **Required/Optional** | **Derived/Stored** |
| Order | NUMERIC(1,0) | >0 | Required | Stored |
| Name | VARCHAR(20) |  | Required | Stored |
| Code | CHAR(1) |  | Required | Stored |

### Term

A Term represents the period of time during which classes occur. They are distinct from a season because more than one term may occur during a season, such as 8-week courses and summer sessions. ID acts as the primary key for the table. Each Term is associated with a particular season, and occurs within a particular year. A term is bound by a StartDate and EndDate. MidtermDate is the halfway point between the StartDate and EndDate, when midterm grades are due.

#### Summary

|  |  |
| --- | --- |
| **Table** | Term |
| **ER origin** | Entity Term |
| **Primary key** | ID |
| **Foreign keys** | Season references Season |
| **Constraints** | <None> |

#### Attributes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Range** | **Required/Optional** | **Derived/Stored** |
| ID | INTEGER | >0 | Required | Stored |
| Year | NUMERIC(4,0) | >0 | Required | Stored |
| Season | NUMERIC(1,0) |  | Required | Stored |
| StartDate | DATE |  | Required | Stored |
| EndDate | DATE |  | Required | Stored |
| MidtermDate | DATE |  | Required | Stored |

### Instructor

Instructor represents an actual instructor. ID acts as the primary key, and the FirstName, MiddleName, and LastName values provide a human-readable identifier for each row. Instructor.SchoolIssuedID is not only unique int the Instructor table, but is unique across Student.SchoolIssuedID as well. Instructors belong to a department and have a unique email address.

#### Summary

|  |  |
| --- | --- |
| **Table** | Instructor |
| **ER origin** | Entity Instructor |
| **Primary key** | ID |
| **Foreign keys** | <None> |
| **Constraints** | Unique(Email) |

#### Attributes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Range** | **Required/Optional** | **Derived/Stored** |
| ID | INTEGER | >0 | Required | Stored |
| FirstName | VARCHAR(50) |  | Required | Stored |
| MiddleName | VARCHAR(50) |  | Optional | Stored |
| LastName | VARCHAR(50) |  | Required | Stored |
| Department | VARCHAR(30) |  | Required | Stored |
| Email | VARCHAR(319) |  | Optional | Stored |

### Section

The Section table holds a listing of offerings of a course. An offering of a course is related to many other aspects of Gradebook, which can be seen in the conceptual diagram and the relations that reference this table. Each section is assigned an ID by Gradebook, though this is not conceptually necessary, as Term, Course, and SectionNumber together represent a unique offering of a course, However, a numeric identifier makes referencing this table much easier, and possibly leads to performance benefits).

SectionNumber is a value (like 01 or 73) that distinguishes offerings of a course among others in the same term. CRN (course registration number) is a numeric value that is used during student enrollment. Schedule represents the days of the week (stored as ‘MR’ for Monday and Thursday, ‘MWF’ for Monday Wednesday and Friday) that the section meets. It is used for both informational purposes and for automatically determining the days in a term that a section meets. Location is a description of where the section meets (like ‘WS103’ or ‘ONLINE’). StartDate and EndDate are the start and end of when a section meets (needs to be within, but not necessarily match the term start and end dates, though this constraint is not currently enforced).

Instructor1, Instructor2, and Instructor3 represent the instructors for a section, Instructor1 being the primary instructor. This is a workaround for the difficulties in enforcing the constraints on the instructors of a section (there must be at least one instructor, but all instructors must be distinct from one another). Conceptually, there may be more than 3 instructors for a course.

#### Summary

|  |  |
| --- | --- |
| **Table** | Section |
| **ER origin** | Associative entity Section (via relationship between Course and Term) |
| **Primary key** | ID |
| **Foreign keys** | Term references Term  Course references Course  Instructor1 references Instructor  Instructor2 references Instructor  Instructor3 references Instructor |
| **Constraints** | Unique(Term, Course, SectionNumber)  Distinct(Instructor1, Instructor2, Instructor3) |

#### Attributes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Range** | **Required/Optional** | **Derived/Stored** |
| ID | INTEGER | >0 | Required | Stored |
| Term | INTEGER | >0 | Required | Stored |
| Course | VARCHAR(8) |  | Required | Stored |
| SectionNumber | VARCHAR(3) |  | Required | Stored |
| CRN | VARCHAR(5) |  | Required | Stored |
| Schedule | VARCHAR(7) |  | Optional | Stored |
| Location | VARCHAR(25) |  | Optional | Stored |
| StartDate | DATE |  | Optional | Stored |
| EndDate | DATE |  | Optional | Stored |
| Instructor1 | INTEGER | >0 | Required | Stored |
| Instructor2 | INTEGER | >0 | Optional | Stored |
| Instructor3 | INTEGER | >0 | Optional | Stored |

### Section\_Instructor

Section\_Instructor establishes a relationship between class sections and the instructors who teach them. Sections may be taught by one or more instructors.

#### Summary

|  |  |
| --- | --- |
| **Table** | Section\_Instructor |
| **ER origin** | Relationship between associative entity Section and entity Instructor |
| **Primary key** | Section.ID, Instructor.ID |
| **Foreign keys** | Section references Section  Instructor references Instructor |
| **Constraints** | <None> |

#### Attributes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Range** | **Required/Optional** | **Derived/Stored** |
| Section | INTEGER | >0 | Required | Stored |
| Instructor | INTEGER | >0 | Required | Stored |

### Grade

The grade table provides a range of grades as LetterGrade, and their associated values as GPA.

#### Summary

|  |  |
| --- | --- |
| **Table** | Grade |
| **ER origin** | Entity Grade |
| **Primary key** | LetterGrade |
| **Foreign keys** | <None> |
| **Constraints** | LetterGrade must be one of the following: A+, A, A-, B+, B, B-, C+, C, C-, D+, D, D-, F, W, or SA. |

#### Attributes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Range** | **Required/Optional** | **Derived/Stored** |
| LetterGrade | VARCHAR(2) |  | Required | Stored |
| GPA | NUMERIC(4,3) | >0 | Required | Stored |

### Section\_GradeTier

Provides a mapping of scores to a letter grade. Allows individual sections to vary grade scoring. LetterGrade and Section together is the primary key. LowPercent and HighPercent specifies the low and high bound for the score mapping.

#### Summary

|  |  |
| --- | --- |
| **Table** | Section\_GradeTier |
| **ER origin** | Associative entity Grade Tier List (via relationship between Section and Grade) |
| **Primary key** | Section, LetterGrade |
| **Foreign keys** | Section references Section  LetterGrade references Grade |
| **Constraints** | Unique(Section, LowPercent, HighPercent) |

#### Attributes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Range** | **Required/Optional** | **Derived/Stored** |
| Section | INTEGER | >0 | Required | Stored |
| LetterGrade | VARCHAR(2) |  | Required | Stored |
| LowPercent | NUMERIC(4,2) | >0 | Required | Stored |
| HighPercent | NUMERIC(5,2) | >0 | Required | Stored |

### Student

Student represents an actual Student. ID acts as the primary key, and the FirstName, MiddleName, and LastName values provide a human-readable identifier for each row. Students and Faculty both have a school-issued ID number which adheres to a similar format and Is used across other existing systems. As such, SchoolIssuedID provides a more familiar identifier to the user in comparison to the arbitrary ID attribute. Students have an email address, an optional major, and are associated to a given year (Freshman, Sophomore, Junior, Senior).

#### Summary

|  |  |
| --- | --- |
| **Table** | Student |
| **ER origin** | Entity Student |
| **Primary key** | ID |
| **Foreign keys** | <None> |
| **Constraints** | Unique(Student.SchoolIssuedID) |

#### Attributes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Range** | **Required/Optional** | **Derived/Stored** |
| ID | INTEGER | >0 | Required | Stored |
| FirstName | VARCHAR(50) |  | Required | Stored |
| MiddleName | VARCHAR(50) |  | Optional | Stored |
| LastName | VARCHAR(50) |  | Required | Stored |
| SchoolIssuedID | INTEGER | >0 | Required | Stored |
| Email | VARCHAR(319) |  | Optional | Stored |
| Major | VARCHAR(50) |  | Optional | Stored |
| Year | VARCHAR(30) |  | Required | Stored |

### Enrollee

Enrollee contains information about a student in context with a specific class section. The student’s major at the time of enrollment is captured in EnrollmentMajor. The date of enrollment is represented by EnrollmentDate and EnrollmentYear. Grades are represented by MidtermWeightedGrade, MidtermGradeComputed, MidtermGradeAwarded, FinalWeightedGrade, FinalGradeComputed, FinalGradeAwarded.

#### Summary

|  |  |
| --- | --- |
| **Table** | Enrollee |
| **ER origin** | Associative entity Enrollee (via relationship of Student and Section) |
| **Primary key** | Student, Section |
| **Foreign keys** | Student references Student  Section references Section  EnrollmentMajor references Major  Section, MidtermGradeComputed references Section\_GradeTier  Section, MidtermGradeAwarded references Section\_GradeTier  Section, FinalGradeComputed references Section\_GradeTier  Section, FinalGradeAwarded references Section\_GradeTier |
| **Constraints** | <None> |

#### Attributes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Range** | **Required/Optional** | **Derived/Stored** |
| Student | INTEGER | >0 | Required | Stored |
| Section | INTEGER | >0 | Required | Stored |
| EnrollmentDate | DATE |  | Required | Stored |
| EnrollmentYear | VARCHAR(30) | >0 | Required | Stored |
| EnrollmentMajor | VARCHAR(30) |  | Required | Stored |
| MidtermWeightedAggregate | NUMERIC(5,2) | >0 | Optional | Stored |
| MidtermGradeComputed | CHAR(1) |  | Optional | Stored |
| MidtermGradeAwarded | CHAR(1) |  | Optional | Stored |
| FinalWeightedAggregate | NUMERIC(5,2) | >0 | Optional | Stored |
| FinalGradeComputed | CHAR(1) |  | Optional | Stored |
| FinalGradeAwarded | CHAR(1) |  | Optional | Stored |

### AttendanceStatus

AttendanceStatus related a single-character Status with a particular status Description. For example, ‘P’ for present, ‘A’ for absent.

#### Summary

|  |  |
| --- | --- |
| **Table** | AttendanceStatus |
| **ER origin** | Entity Attendance Status |
| **Primary key** | Status |
| **Foreign keys** | <None> |
| **Constraints** | Unique(Description) |

#### Attributes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Range** | **Required/Optional** | **Derived/Stored** |
| Status | CHAR(1) |  | Required | Stored |
| Description | VARCHAR(20) |  | Required | Stored |

### AttendanceRecord

AttendanceRecord relates a student with an attendance status (present or absent), date, and class section. Gaps in attendance records when a class should have met are considered to be present by default (to conserve storage space).

#### Summary

|  |  |
| --- | --- |
| **Table** | AttendanceStatus |
| **ER origin** | Weak entity Attendance Record |
| **Primary key** | Student, Section, Date |
| **Foreign keys** | Student references Student  Section references Section  Status references AttendanceStatus |
| **Constraints** | <None> |

#### Attributes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Range** | **Required/Optional** | **Derived/Stored** |
| Student | INTEGER | >0 | Required | Stored |
| Section | INTEGER | >0 | Required | Stored |
| Date | DATE |  | Required | Stored |
| Status | CHAR(1) |  | Required | Stored |

### Section\_AssessmentComponent

This entity relates AssessmentComponents to a particular Section. Name refers to the type of assessment component being stored (such as “exam” or “homework”). Weight assigns a percentage value to a given item, and NumAssessments represents the planned number of assessments.

#### Summary

|  |  |
| --- | --- |
| **Table** | Section\_AssessmentComponent |
| **ER origin** | Relationship between Section and AssessmentComponent |
| **Primary key** | Name, Section |
| **Foreign keys** | Section references Section |
| **Constraints** | <None> |

#### Attributes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Range** | **Required/Optional** | **Derived/Stored** |
| Section | INTEGER | >0 | Required | Stored |
| Name | VARCHAR(30) |  | Required | Stored |
| Weight | NUMERIC(3,2) | >0 | Required | Stored |
| NumAssessments | INTEGER | >0 | Required | Stored |

### Section\_AssessmentItem

This entity represents general information or an assignment in a particular section and section component. AssessmentNumber is … This entity also defines the total points possible as BasePointsPossible, extra credit as ExtraCreditPointsPossible. Date of assignment and turn in is captured as AssignedDate and DueDate.

#### Summary

|  |  |
| --- | --- |
| **Table** | Section\_AssessmentItem |
| **ER origin** | Relationship between Section and Assessment Item |
| **Primary key** | Section, Component, AssessmentNumber |
| **Foreign keys** | Section references Section\_AssessmentComponent  Component references Section\_AssessmentComponent |
| **Constraints** | <None> |

#### Attributes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Range** | **Required/Optional** | **Derived/Stored** |
| Section | INTEGER | >0 | Required | Stored |
| Component | VARCHAR(20) |  | Required | Stored |
| AssessmentNumber | INTEGER |  | Required | Stored |
| BasePointsPossible | NUMERIC(5,2) | >=0 | Required | Stored |
| ExtraCreditPointsPossible | NUMERIC(5,2) | >=0 | Required | Stored |
| AssignedDate | DATE |  | Optional | Stored |
| DueDate | DATE |  | Optional | Stored |

### Enrollee\_AssessmentItem

This table contains AssessmentNumber information in context with a specific Student, Section, and Component. BasePointsEarned, ExtraCreditPointsEarned, and Penalty determines the overall grade. Turn-in date is captured in SubmissionDate.

#### Summary

|  |  |
| --- | --- |
| **Table** | Enrollee\_AssessmentItem |
| **ER origin** | Relationship between Enrollee and Assessment Item |
| **Primary key** | Student, Section, Component, AssessmentNumber |
| **Foreign keys** | Student references Student  Section references Section  Component references Section\_AssessmentComponent  AssessmentNumber references Section\_AssessmentItem |
| **Constraints** | <None> |

#### Attributes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Range** | **Required/Optional** | **Derived/Stored** |
| Student | INTEGER | >0 | Required | Stored |
| Section | INTEGER | >0 | Required | Stored |
| Component | VARCHAR(30) |  | Required | Stored |
| AssessmentNumber | INTEGER |  | Required | Stored |
| BasePointsEarned | NUMERIC(5,2) | >=0 | Optional | Stored |
| ExtraCreditPointsEarned | NUMERIC(5,2) | >=0 | Optional | Stored |
| SubmissionDate | DATE |  | Optional | Stored |
| Penalty | NUMERIC(5,2) | >=0 | Optional | Stored |