## Current (DASSL) Schema

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

**Course**(Name, Description\*)

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

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

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

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

**Section**(CRN, Year, Season, SectionNumber, Name\*, Schedule\*, MidtermDate\*, LocationTaught)  
 Name → Course.Name  
 Year → Term.Year  
 Season → Term.Season

**Section\_Grade\_Tier**(LetterGrade, CRN, LowPercent, HighPercent)  
 LetterGrade → Grade.LetterGrade  
 CRN → Section.CRN

**Grade**(LetterGrade)

**Attendance\_Record**(Date, ID, CRN, Status\*)  
 ID → Student.ID  
 CRN → Section.CRN

**Assessment\_Component**(Name, CRN, Weight\*, Planned Number of Assignments\*)  
 CRN → Section.CRN

**Assessment\_Item**(Assessment Number, CRN, BasePointsPossible\*, ExtraCreditPointsPossible\*)  
 CRN → Section.CRN

**Enrollee**(ID, CRN**,** EnrollmentDate\*, MidtermGradeComputed\*, MidtermGradeAwarded\*, FinalGradeComputed\*, FinalGradeAwarded\*)  
 ID → Student.ID  
 CRN → Section.CRN  
 MidtermGradeComputed → Section\_Grade\_Tier.LetterGrade  
 MidtermGradeAwarded → Section\_Grade\_Tier.LetterGrade  
 FinalGradeComputed → Section\_Grade\_Tier.LetterGrade  
 FinalGradeAwarded → Section\_Grade\_Tier.LetterGrade

**Submission**(ID, CRN, Name, BasePointsEarned\*, ExtraCreditEarned, HandoutDate\*, SubmissionDate\*, Penalty)  
 ID → Enrollee.ID  
 CRN → Enrollee.CRN  
 Name → Assessment\_Component

## New (Team Alpha) Schema

**Course**(Name, DefaultTitle\*)

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

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

**SignificantDate**(Term, Date, Name, ClassesHeld, Reason)  
 Term → Term.ID

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

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

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

**Section**(ID, Term, Course\*, SectionNumber\*, CRN\*, Title\*, 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, Standing\*)

**Major**(Name)

**Student\_Major**(Student, Major)  
 Student → Student.ID  
 Major → Major.Name

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

## Relational Schema Changes

This section describes the changes made from the existing Gradebook conceptual schema. Changes can be of one of three kinds:

* Modification – Changes that are a result of the changes and additions we plan to make as part of the project
* Fix – Addresses inconsistencies between Gradebook’s current conceptual, logical, and physical (implemented SQL code) schemas; related to Gradebook Issue 41
* Alteration – Suggests an alteration to Gradebook’s current schema that is not directly related to the project, but improves the product in a way that benefits system testing and the product as a whole

### Course

* Alteration – Attribute name Title (named Description in conceptual schema) was changed to DefaultTitle: this reflects a change we wish to make to Gradebook in terms of handling of course and section titles. Currently, a course always has the same title, and sections do not have an associated title. This leads to problems with “real-life” data, as course names change over time and different sections of a course offered in the same term can have different titles. Ideally, titles are associated with sections, however, each course has a default title that can be copied over to a section if a different title is not provided. This a part of resolving Gradebook Issue 67. See related changes in the Section associative entity.

### Season

No changes in any schema

### Term

* Fix – Add identifying ID attribute: the existing conceptual schema lists Year and Season together being the primary key. This seems to be enough for ensuring uniqueness, however the actual implemented primary key is a currently unlisted attribute ID. It is not clear if this change is intentional, as it is not documented anywhere (for example, using a numeric identifier in the physical schema instead of Year, Season may have been chosen for convenience and/or performance reasons). We are making an assumption that this is not the case and are considering ID as the intended primary key from the beginning, with a constraint that Year and Season must together be unique.
* Alteration – Add required MidtermDate attribute: currently, every section has a midterm date, however, this does not seem necessary as the midterm date is practically always determined at an organizational level. If an instructor wishes to have an assessment count before/after the midterm (useful when computing midterm grades), then the submission date of the assessment could be set before/after the midterm date.

### SignificantDate

Modification – The SiginificantDate relation is used to store information about holidays and closures and is a result of implementing Gradebook’s FP5.7 as a part of our project. See the description provided later in the document for more information about what each attribute represents.

### Instructor

* Fix – Add optional Email attribute: although present in the conceptual and physical schemas, the Email attribute was missing from the logical (relational) schema.
* Alteration – Add required and unique attribute SchoolIssuedID: this matches a similar attribute in the Student entity. We plan for this to serve as a friendly “username” used for interfacing with the product. This is distinct from IDs that the Gradebook product uses internally. SchoolIssuedIDs should be unique between all Students and Instructor rows.

### Section

* Fix – Add StartDate and EndDate attributes: these attributes were present in the implemented schema, but not the conceptual or logical schemas. It is necessary to keep these are section start and end dates do not necessarily match the term start and end dates.
* Fix – Add identifying ID attribute: the existing conceptual schema listed CRN as the primary key, which is wrong, even at the conceptual level (not unique). A possible primary key could be a combination of Term, Course, and SectionNumber (set as unique in the implemented schema). However, the implemented schema uses a numeric ID attribute (either for convenience and/or performance reasons). Like with Term.ID, we will consider ID as the intended primary key from the beginning, with a constraint that Term, Course, and SectionNumber must together be unique.
* Fix – Change Name attribute to Course: using Name as the attribute’s name may be misleading, as it actually refers to a course name (like CS305) rather than a section name.
* Fix – Mark SectionNumber as required: section number is conceptually a required attribute and is implemented as such in the physical schema.
* Alteration – Remove Midterm attribute: midterm dates could be stored at a term level rather than section level. See related change in Term entity for more information.
* Alteration – Add required Title attribute: each section may have its own title, which will allow for multiple sections of the same course to have different titles, and for historical title information to be stored. See related change in Course entity.
* Alteration – Remove InstructorX workaround: currently, a workaround exists to make it easier to enforce the constraints on instructors on a section, which avoids needing to have a Section\_Instructor relation. The gray-highlighted section relation above represents the current implementation that may be replaced by the two relations listed before it.

### Section\_Instructor

* Fix – Change reference to Section entity to ID: the correct attribute to reference in the Section relation is the ID attribute, not CRN.
* Alteration – Change attribute name of reference to Instructor entity to Instructor from ID: improves clarity by avoiding confusion with reference to Section.ID.

Note: We will investigate implementing this relationship entity in the physical schema, though it is a low-priority item. (Currently, a workaround in the Section entity exists, but limits a section to having 3 instructors.) See notes on Section entity.

### Grade

* Fix – Add GPA attribute: the implemented physical schema has a GPA attribute that needs to be stored for computing grades which is missing from the conceptual and logical schemas.

### Section\_GradeTier

* Fix – Change CRN to Section: ID is the correct reference to the Section entity, not CRN.
* Fix – Make LowPercentage and HighPercentage required: both attributes are conceptually and physically required.

### Student

* Modification – Remove Major attribute: a student can have 0 or more majors, meaning that an additional relation is needed to store this information (see Student\_Major and Major relations).
* Fix – Add SchoolIssuedID attribute: Gradebook’s physical schema implemented an attribute for maintaining identifiers issued by the educational institution. This may be of additional use to Gradebook and should be noted at all schema levels.
* Fix – Add optional Email attribute: although present in the physical schema, the Email attribute was missing from the logical (relational) schema.
* Alteration – Rename Standing attribute to Year: standing may be confused for another concept (such as “good” vs “bad” standing). Year is also the name used by the physical schema.

### Major

Modification – This is a newly added relation that is a result of the Major multi-valued attribute that is present in the Student entity due to a modification. Name refers to the institutionally-assigned name for a major (for example: “Computer Science”).

Unrelated to the logical schema, but this entity should also help normalize the Gradebook schema for future development.

### Student\_Major

Modification – This is a newly added relation that is a result of the Major multi-valued attribute that is present in the Student entity due to a modification. A Student is not required to have a major but is allowed to have any number of majors (however, most students will have one major, some two majors, and >2 only in very exceptional cases). Students without an entry in the relation could be considered to have an “undeclared” major.

### Enrollee

* Fix – Add EnrollmentMajor and EnrollmentYear attributes: both attributes hold information about the student at the time that they enrolled in a course, and were captured in the physical schema, but not the conceptual or logical schema.
* Fix – Add MidtermWeightedAggregate and FinalWeightedAggregate attributes: both attributed refer to grades that are computed with appropriate weighting assigned to them and were in the physical schema, but not the conceptual or logical schemas.
* Fix – Change CRN reference to Section.ID: ID is the appropriate attribute to reference in Section, not CRN. (Issue exists only in conceptual and logical schemas)
* Fix – Remove required status from MidtermGradeAwarded and FinalGradeAwarded attributes: both attributes were incorrectly marked as required in the logical schema. It is not necessary to have a value for these attributes at the time of enrollment or at some earlier points during the section.
* Fix – Remove requires status from MidtermGradeComputed and FinalGradeComputed attributes: again, at some point in a section these grades are not required to exist. Until these attributes are moved into a view or to computed columns, it does not make sense to require these attributes.
* Fix – Reference to Section\_GradeTier also includes Section: in the physical schema, the reference to the Section\_GradeTier relation correctly include section in the reference.
* Alteration – Change ID attribute name to Student: improves clarity by avoiding confusion with reference to Section.ID.

### AttendanceStatus

Fix – This relation was originally missing from both the conceptual and logical schemas. This entity is necessary in preventing data anomalies with attendance information. Based off the implemented physical schema, attendance status has two attributes:

* Status: an identifier that is used to reference the attendance status (also referred to as attendance code)
* Description: a unique word or short phrase that defines the attendance status (such as “Present”)

### AttendanceRecord

* Fix – Change relation name from Attendance\_Record to AttendanceRecord: Gradebook’s naming conventions use underscores to denote relations that are result of relationships but otherwise does not use underscores in relation names. The physical schema implementation does not use an underscore.
* Fix – Add reference to AttendanceStatus: the Status attribute is implemented in the physical schema as a reference to AttendanceStatus, however, this was not noted in the logical schema (likely related to the missing AttendanceStatus relation in the logical schema).
* Fix – Change attribute name of reference to Student entity to Student: improves clarity by avoiding confusion with reference to Section.ID.
* Fix – Change CRN reference to Section.ID: ID is the appropriate attribute to reference in Section, not CRN. (Issue exists only in conceptual and logical schemas)

### Section\_AssessmentComponent

* Fix – Change relation name from Assessment\_Component to Section\_AssessmentComponent: the relation is a result of the relationship between Section and AssementComponent, meaning that it should be renamed to indicate so.
* Fix – Change CRN reference to Section.ID: ID is the appropriate attribute to reference in Section, not CRN.
* Alteration – Rename Planned Number of Assignments to NumAssessments: the original attribute name did not fit into the naming convention of the relational schema.

### Section\_AssessmentItem

Fix – This relation was originally listed in the logical schema as Assessment\_Item. However, it is more correct refer to it as Section\_AssessmentItem, since it forms from the ternary relationship between Enrollee, AssessmentComponent, and AssessmentItem.

The relation was updated to match the implementation in the physical schema, while making sure that the changes were conceptually valid.

### Enrollee\_AssessmentItem

Fix – This relation was originally listed in the logical schema as Submission. However, like the previous relation, it is more correct to refer to it as Enrollee\_AssessmentItem, as it is part of the ternary relationship between Enrollee, AssessmentComponent, and AssessmentItem.

The relation was updated to match the implementation in the physical schema, while making sure that the changes were conceptually valid.

## 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). DefaultTitle is a title that corresponds to the course that is used for sections of the course when it is not overridden. (See the Section relation for more information)

#### 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 |
| DefaultTitle | 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 |

### SignificantDate

SignificantDate contains information about special days on the academic calendar. Term and Date together determines when the occasion occurs and which term it occurs in. ClosureStatus uses a single character to represent the kind of closure. For instance, a complete closure would be represented as ‘C’. If admin offices are closed, ‘A’ would be used. Null may be used to indicate that school is not closed. The Reason attribute describes the reason for an occasion (holiday, weather, and so forth). This disctinction between ClosureStatus and Reason is helpful for situations where a holiday may not be a day off, or when a day off may not be a holiday, and so forth.

#### Summary

|  |  |
| --- | --- |
| **Table** | SignificantDate |
| **ER origin** | Entity Significant Date |
| **Primary key** | Term, Date, Name |
| **Foreign keys** | Term references Term |
| **Constraints** | <None> |

#### Attributes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Range** | **Required/Optional** | **Derived/Stored** |
| Term | INTEGER | >0 | Required | Stored |
| Date | DATE |  | Required | Stored |
| Name | VARCHAR(30) |  | Required | Stored |
| ClassesHeld | BOOLEAN |  | Required | Stored |
| Reason | VARCHAR(30) |  | 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. 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. 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)  Unique(Student.SchoolIssuedID ∪ Instructor.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 |
| 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. Title is the actual title that the section is listed as, which may have been copied over from the corresponding Course’s DefaultTitle attribute. 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 |
| Title | VARCHAR(100) |  | 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. Student.SchoolIssuedID is not only unique int the Student table, but is unique across Instructor.SchoolIssuedID as well. Students have an email address 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 ∪ Instructor.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 |
| Year | VARCHAR(30) |  | Required | Stored |

### Major

The Major table contains a list of unique names of Majors.

#### Summary

|  |  |
| --- | --- |
| **Table** | Major |
| **ER origin** | Entity Major |
| **Primary key** | Name |
| **Foreign keys** | <None> |
| **Constraints** | <None> |

#### Attributes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Range** | **Required/Optional** | **Derived/Stored** |
| Name | VARCHAR(30) |  | Required | Stored |

### Student\_Major

Student\_Major relates students to one or more majors. Students not listed in this table are considered to be “undeclared”.

#### Summary

|  |  |
| --- | --- |
| **Table** | Student\_Major |
| **ER origin** | Relationship between Student and Major |
| **Primary key** | Student, Major |
| **Foreign keys** | Student references Student.ID  Major references Major.Name |
| **Constraints** | <None> |

#### Attributes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Range** | **Required/Optional** | **Derived/Stored** |
| Student | INTEGER | >0 | Required | Stored |
| Major | 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 |