

Catalyst Maria High School – Class Grid

Melissa Mosier

IST 659 Data Administration Concepts and Management

8/17/2022

**Summary**

Business case, high level rules about the data, stakeholders, and expectations of the final outcome

Catalyst Maria High School is creating a database for their class grid. This is how they will assign teachers and students to their classes for the coming semester. We will not get into Human Resources, teacher and admin relationships, additional staff, materials for classes, etc.; this is not meant to be a full mapping of relationships between everyone in this system. It is only to organize class assignments.

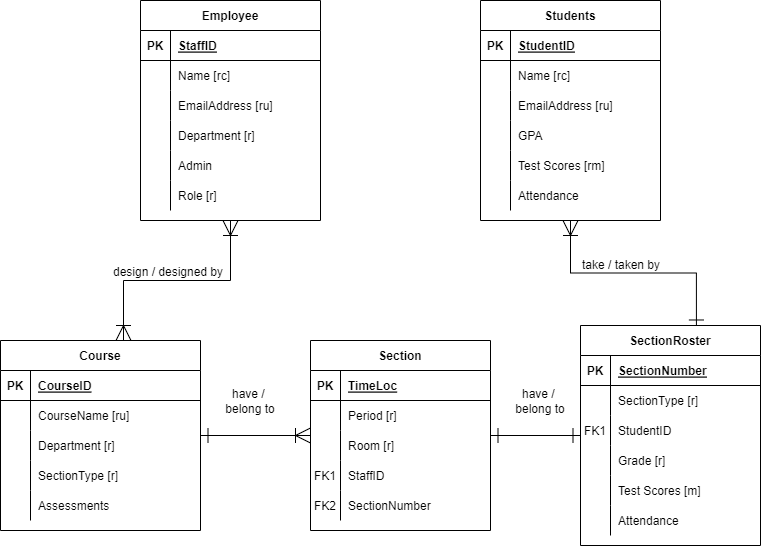
Teachers and administration from the same department will work together to create courses for their students. They will create a number of courses for each department. Each course will have at least one section but may have more. Each section will have many students.

Stakeholders: Administrators, Teachers, Students

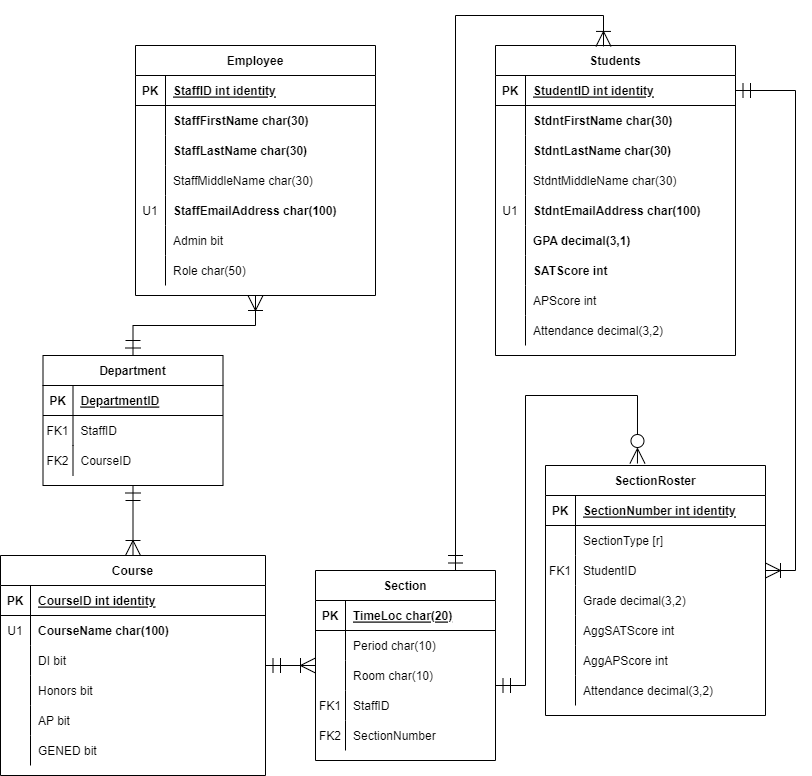
Business Rules:

* Many employees can design a course (this will usually include at least one teacher and one administrator). Many courses can be designed by an employee. Generally employees will design courses according to which department they belong.
* A course may have many sections. A section belongs to one course.
* Each section will have its own roster. A roster belongs to one section. Administrative information for each section (where it is, during which period, who teaches it) is kept separately from teacher-student information (section – DI, GENED, AP, etc.; grades; etc.)
* Many students will go on one roster (many students will take one section of a class together). A roster will only have a list of students for one class section (that class section will be taken by one roster of students).

**Conceptual Model**



**Normalized Logical Model**



Additional Notes / Glossary:

Employee Table

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Comments / Examples** |
| **StaffID** | **Int identity** | **Primary Key; randomly generated** |
| **StaffFirstName** | **Char(25)** |  |
| **StaffLastName** | **Char(25)** |  |
| StaffMiddleName | Char(25) |  |
| **StaffEmailAddress** | **Char(100)** | **Unique** |
| Admin | Bit | Ex: 0 or 1 |
| Role | Char(50) | Ex: “Algebra 2 Teacher” vs “Math Coach” (Coach is an admin role) |

Course Table

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Comments / Examples** |
| **CourseID** | **Int identity** | **Primary Key; randomly generated** |
| **CourseName** | **Char(100)** | **Unique. Ex: “Psychology”** |
| DI | Bit | DI = Direct Instruction |
| Honors | Bit |  |
| AP | Bit | AP = Advanced Placement |
| GENED | Bit | GENED = General Education |

Notes:

* A “1” for DI / Honors / AP / GENED indicates whether we are required to offer this type of section for this course. Later, we will indicate if a particular section is DI / Honors / AP / GENED.

Section Table

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Comments / Examples** |
| **TimeLoc** | **Char(20)** | **Primary Key. Derived from Period and Room.** |
| **Period** | **Char(10)** |  |
| **Room** | **Char(10)** |  |
| StaffID | Foreign Key | From Employee table |
| SectionNumber | Foreign Key | From SectionRoster table |

Notes:

* No two classes can be taught in the same room during the same period. This PK (TimeLoc) could be the same as the SectionNumber PK, but this information is administration-facing, whereas SectionRoster is student-teacher-facing.

SectionRoster Table

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Comments / Examples** |
| **SectionNumber** | **Int identity** | **Primary Key; randomly generated** |
| **SectionType** | **Char(25)** | **Categorical. Ex: “GENED”** |
| StudentID | Foreign Key | From Student table |
| Grade | Decimal (3,2) |  |
| AggSATScore | Int | Aggregated SAT Score |
| AggAPScore | Int | Aggregated AP Score |
| Attendance | Decimal (3,2) |  |

* This is a massive table that could be split into many smaller tables if each SectionNumber is its own table.
  + In 2NF, Section Number and SectionType would repeat while each StudentID gets their own row with Grade, TestScores, and Attendance.
  + In 3NF, each SectionNumber has its own table, and each StudentID still gets their own row.

Student Table

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Comments / Examples** |
| **StdntID** | **Int identity** | **Primary Key; randomly generated** |
| **StdntFirstName** | **Char(25)** |  |
| **StdntLastName** | **Char(25)** |  |
| StdntMiddleName | Char(25) |  |
| **StdntEmailAddress** | **Char(100)** |  |
| **GPA** | **Decimal (3,1)** | **Ex: “3.24”** |
| **SATScore** | **Int** |  |
| APScore | Int |  |
| Attendance | Decimal (3,2) | This is a percentage stored as a decimal. Ex: “98.2” |