U of S Course Planner Database

TABLE: **Courses**

- Contains all courses offered at the University of Saskatchewan

Column Name	Data Type	Details
SubjectCode	NCHAR(4)	- PRIMARY KEY 1/2 - 4 letter code for the course subject (ex. CMPT) - ZZZZ is used as a general stand-in subject code for courses that are undefined/currently outside the standard subject coding (particularly in CreditConflicts and Prerequisites) - HSCH is used as a stand-in for courses relating to highschool courses, as the current system is set up for University courses only
CourseCode	INTEGER	- PRIMARY KEY 2/2 - 3 number code for the specific course in the subject - Within the range of 1-999 - 999 is used as a general stand-in course code for courses that are undefined/currently outside the standard course coding (particularly in CreditConflicts and Prerequisites)
CourseName	TEXT	- Title name of the course
Description	TEXT	- A brief description of the course and what subject matter it will be covering

TABLE: **CreditConflicts**

- Contains all courses that have conflict courses in terms of earning credit for taking them, and what those conflicts are

Column Name	Data Type	Details
CreditConflictID	INTEGER	- PRIMARY KEY

		- Surrogate key, arbitrary increasing integers
SubjectCode	NCHAR(4)	- FOREIGN KEY; see Courses table - The course subject that has a conflict - (1) Upon update or deletion of the primary key referenced, any rows with that key will be updated/deleted from CreditConflicts accordingly to reflect the change
CourseCode	INTEGER	- FOREIGN KEY; see Courses table - The numerical code for a specific course that has a conflict - See (1) above
ConflictSubjectCode	NCHAR(4)	- FOREIGN KEY; see Courses table - The course subject that is a conflict - (2) Upon update or deletion of the primary key referenced, no change will be made, due to previous completion of the course still being a potentially valid conflict. May need to be changed manually if further action is required.
ConflictCourseCode	INTEGER	- FOREIGN KEY; see Courses table - The numerical code for a specific course that is a conflict - See (2) above
ConflictType	TEXT	- Brief categorization of the type of conflict - Entered values can only be 'Regular', 'Previous Name', 'Other', or 'Fluency' - 'Regular' -> Cannot take course for credit if this second course has been taken - 'Previous Name' -> a course name was changed, with the older course no longer being offered as it was - 'Fluency' -> for language courses that cannot be taken if fluent in that language, or certain high school courses of that language were taken - 'Other' -> any special cases not identified above, currently including any "Special Topics" courses, which may only apply depending on the specific topics covered the year it was taken

TABLE: **Degrees**

- Contains the degrees offered at the University of Saskatchewan, potentially with a backlog of discontinued courses

Column Name	Data Type	Details
ProgramName	TEXT	- PRIMARY KEY 1/2 - Name of the overall program/subject to be studied (ex. Computer Science, Accounting, etc.)
Degree	TEXT	- PRIMARY KEY 2/2 - Name of the specific degrees offered for a program (ex. Bachelor of Science Honours)

		- ‘All’ is used as a stand-in for courses common to all degrees in that program, for use in DegreeRequirements - Any Programs with 2+ Degrees that all have significant common overlap in required courses should have an ‘All’
SchoolingLevel	TEXT	- Level of schooling for the degree - Entered values must be 'Undergraduate', 'Graduate', or 'Non-Degree Certificate'
RequirementsType	TEXT	- How the requirements for the degree are organized - Entered values must be either 'ByYear' (ie. Year 1, Year 2, etc.) or 'ByCategory' (ie. C4 Major Requirement, C1, C2, etc.)
CurrentlyOffered	INTEGER	- A binary mark of whether or not the listed degree is the most current in regards to required courses - Must be not null, Default is 1 - Values must be either 0 (no, not currently offered) or 1 (yes, currently offered) - In progress: theoretically, old degree checklists could be updated with a date/semester timeframe to represent when it was current/they were introduced, and DegreeRequirements entries changing accordingly, then the changes entered

TABLE: **DegreeRequirements**

- Contains what courses are required for each degree in the database
- Shows lists of potential required courses for each given category of the degree's requirements
- Potentially with a backlog of previously offered requirements for that degree that have since been changed

Column Name	Data Type	Details
DegreeReqID	INTEGER	- PRIMARY KEY - Surrogate key, arbitrary increasing integers
ProgramName	TEXT	- FOREIGN KEY; see Degrees table - (1) Upon update of the primary key referenced, any rows with that key will be updated in DegreeRequirements accordingly - (2) Deletion of the primary key referenced is restricted, given the high volume of data associated with requirements, as well as how often older versions of degree requirements remain in use by continuing students for some years after
Degree	TEXT	- FOREIGN KEY; see Degrees table - See (1) and (2) above
SubjectCode	NCHAR(4)	- FOREIGN KEY; see Courses table

		- The course subject that is on a list of (possible) required courses to complete the specifies degree - (3) Upon update or deletion of the primary key referenced, any rows with that key will be updated/deleted from DegreeRequirements accordingly to reflect the change
CourseCode	INTEGER	- FOREIGN KEY; see Courses table - The numerical code for a specific course that is on a list of (possible) required courses to complete the specifies degree - See (3) above
RequirementGroup	TEXT	- In what section of the requirements type the course belongs to (ex. C1, C2, C3, B1, Year 1, Year 2, etc.)
ReqStatus	TEXT	- A means to separate lists of requirements that ask for a certain number of courses to be selected from that list to fulfill the requirement, with letters at the end to separate different lists that ask for the same number of courses within a requirement group. (ex. 1ofa, 1ofb, 2Ofa)

TABLE: **Prerequisites**

- Contains all courses that have prerequisite courses, and what those prerequisites are

Column Name	Data Type	Details
PrerequisiteID	INTEGER	- PRIMARY KEY - Surrogate key, arbitrary increasing integers
SubjectCode	NCHAR(4)	- FOREIGN KEY; see Courses table - The course subject that has a prerequisite - (1) Upon update or deletion of the primary key referenced, any rows with that key will be updated/deleted from CreditConflicts accordingly to reflect the change
CourseCode	INTEGER	- FOREIGN KEY; see Courses table - The numerical code for a specific course that has a prerequisite - See (1) above
PrerequisiteSubjectCode	NCHAR(4)	- FOREIGN KEY; see Courses table - The course subject that is a prerequisite - (2) Upon update or deletion of the primary key referenced, no change will be made, due to previous completion of the course still being a potentially valid prerequisite. May need to be changed manually if further action is required.
PrerequisiteCourseCode	INTEGER	- FOREIGN KEY; see Courses table - The course subject that is a prerequisite - See (2) above

OneOf	INTEGER	- If prerequisite courses are only required to be one of a selection of two or more, they will be given the same integer in this column - May be slightly reworked as a 'NumberOf' col to accommodate different [number] of options, as two of is not as common as previously thought and larger options such as '3 of' and '4 of' exist, but not for now
TwoOf	INTEGER	- If prerequisite courses are only required to be two of a selection of three or more, they will be given the same integer in this column - May be slightly reworked as a 'Group' for NumberOf col, for the same reasons as in OneOf, but not for now
Outliers	TEXT	- Temporary, for identifying difficult to pigeonhole data and why, notes of a sort - Maybe not so temporary - Provides text data on outlier prerequisites that don't fit the above common schema

TABLE: **Users**

- A record of all users to have registered an account, with usernames and passwords

Column Name	Data Type	Details
UserID	INTEGER	- PRIMARY KEY - Surrogate key, arbitrary increasing integers
Username	TEXT	- The username of the user's account given at registration - Must be unique and not null - Can be changed
Password	TEXT	- The password of the user's account given at registration - Must be unique and not null - Can be changed

TABLE: **CompletedList**

- Holds all courses saved as completed by all users, identified to whom they belong by the UserID
- Has a UNIQUE constraint involving all columns but the CompletedCourseID so as to prevent a user from saving the same course as completed twice

Column Name	Data Type	Details
CompletedCourseID	INTEGER	- PRIMARY KEY - Surrogate key, arbitrary increasing integers
SubjectCode	NCHAR(4)	- FOREIGN KEY; see Courses table

		- The course subject that has been saved as a 'completed' by the student to whom this list belongs - (1) Upon update or deletion of the primary key referenced, any rows with that key will be updated/deleted from CompletedList accordingly to reflect the change
CourseCode	INTEGER	- FOREIGN KEY; see Courses table - The numerical code for a specific course that has been saved as a 'completed' by the student to whom this list belongs - See (1) above
UserID	INTEGER	- FOREIGN KEY - The numerical code representing the user who saved the given completed course to their list - See (1) above

TABLE: **FavouriteList**

- Holds all favourited courses by all users, identified to whom they belong by the UserID
- Has a UNIQUE constraint involving all columns but the FavouriteID so as to prevent a user from saving the same course as a favourite twice

Column Name	Data Type	Details
FavouriteID	INTEGER	- PRIMARY KEY - Surrogate key, arbitrary increasing integers
SubjectCode	NCHAR(4)	- FOREIGN KEY; see Courses table - The course subject that has been saved as a 'favourite' to be referenced later by the user to whom this list belongs - (1) Upon update or deletion of the primary key referenced, any rows with that key will be updated/deleted from FavouriteList accordingly to reflect the change
CourseCode	INTEGER	- FOREIGN KEY; see Courses table - The numerical code for a specific course that has been saved as a 'favourite' to be referenced later by the user to whom this list belongs - See (1) above
UserID	INTEGER	- FOREIGN KEY - The numerical code representing the user who saved the given favourited course to their list - See (1) above