**courses\_schedule\_sections.php**

1. **Start in the view() function.**
   * Produce an abstracted ID for each course so that we can split multi-term courses into one-term courses. To do this:  
     1. Figure out how many terms each course will span based on course.term\_len and the number of terms in the school’s schedule. This should be programmatic but currently is not.
     2. Get all courses but exclude those that have already been entered in the abstracted table (courses\_split\_by\_term). Split the courses by inserting a row for each term of the course.
     3. **TO DO:** If this is a school scheduling using houses, When splitting the courses, we must do step ii for each of the possible houses for which there is the minimum number of students requesting the course from that house.
   * Get only the courses that have requests (from students whose enrollment status is current in the district) at least equal to the minimum section size. Suggest the number of sections of each course by dividing the number of requests by the max section size and rounding up.
   * Get courses that have requests totaling fewer than the minimum. Either delete those requests or lower the minimum section size so that they can be scheduled.
   * Check each course for violations and fix them, if any. The function **check\_violations()** runs through these checks on a per-course basis so it can be called on each course and an alert indicator added if any violations are found. **view\_details\_courses()** function will display the violations and details.
   * Violations include the following:  
     1. The course has not been assigned any possible teachers.
     2. The possible teachers have not been assigned any possible times.
     3. The course has not been assigned any possible times.
     4. There is no intersection between the course times and teacher times.  
        1. **TO DO:** There is a further level of checking in this point that has not yet been done. It is possible that, in aggregate, the possible teachers and the course have intersecting times but that the times are not sufficient to satisfy the term\_len without using different times and different teachers. The current check for this intersection is simply checking for any shared time slots and is quite dumb.
        2. **TO DO:** The way I’d like for this to work, visually, for the user is that term\_len for a course is displayed using blocks that fit into a grid. E.g. a course that meets for two of three trimesters that need not be consecutive has three distinct legal layout patterns:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

Once we have generated all possibilities, we should confirm that there is a teacher match to at least one of the layout patterns.

* + 1. The possible teachers have not been assigned any possible rooms.
    2. The course has not been assigned any possible rooms.
    3. There is no intersection between the course rooms and teacher rooms.  
       1. **TO DO:** The current script checks that at least one teacher has a legal room. We should throw an error if any of the teachers do not have at least one legal course room because this will make it impossible to assign those teachers any sections of the course. If we see this error, the user should either be required to add one of the offending teacher’s rooms to the course’s possible rooms or remove the teacher from the list of possible teachers.

1. **If all violations have been fixed, proceed to the create\_sections() function.**
   * Start by getting the courses we need to create sections for. This is either all of the courses with at least the minimum number of requests or the courses passed into the function. This function is called later from check\_fix\_fte as well, in which case, courses are passed in.  
     1. The function requires that you pass in courses with at least:   
        1. course\_id (the abstracted, split ID, not the original)
        2. the number of sections suggested [first time through, this will be CEIL(no\_requests / max\_enrollment). If we are calling the function to fix fte, the number of sections suggested will be higher in order to force the creation of more sections to fill FTE deficits.]
        3. term\_no out of the no\_terms that the course spans (e.g. if it is part 1 of 1 or part 1 of 3 makes a difference in determining which terms are legal, and this is going to be a subset of the legal terms for the original course\_id).
        4. [Optional]. You may specify teacher\_id
        5. [Optional]. You may specify term\_id
   * For each course, do the following to schedule sections of that course:  
     1. Get all possible teachers based on the original course\_id. Load them into a deck because later, we will pick teachers from the deck to fill sections. Order the teachers in terms of those who have the least number of sections assigned to them so we pick teachers with the least fulfilled FTE first.   
          
        **TO DO:** This should be more sophisticated and order by teachers with the greatest deficit between no\_sections\_assigned and min\_fte in the legal terms for this course.
     2. If there are at least some teachers found, loop from the number of sections of the course already created to the number suggested to create the additional required number of sections.  
        1. Create one section. Assign a **section\_num**; this is important because multi-part courses get stitched back together based on those that share a **course\_id** in **courses\_split\_by\_term** and a **section\_num** in **sections\_scheduling**.
        2. Assign a term to the section.  
           1. If a specific term has been passed in, use that term.
           2. If no term has been passed, get the possible terms for this section. Use the term\_no out of the no\_terms to determine the subset of legal terms for this course part. From that subset, pick the term that already has the fewest number of sections scheduled to it to try to achieve an even distribution of sections across the school year.
        3. Assign a teacher to the section.  
           1. If a specific teacher has been passed in, use that teacher.
           2. If no teacher has been passed, pick the teacher from the top of the deck.
           3. However, if this course is a part of a split-course, check if any other parts have been scheduled yet, and use the same teacher.
           4. Confirm that the teacher picked still has room to teach the section in the term selected and that the teacher has legal time slots in the term.   
                
              If yes, assign the teacher to the section and move to the next section, picking the next teacher in the deck for the next section. If all the teachers have been drawn, reload the deck.  
                
              If you have been through all the teachers in the deck and were unable to assign a teacher, scrap the section.
   * Check for multi-part course orphans and delete them.  
     1. Since we are creating sections one course at a time and some courses have been split, it is possible that we have created sections for part of a course but not the whole course (in case we were unable to find a teacher, etc.) So, check for any orphans and delete them.
2. **Check for and fix FTE deficits by creating more sections: check\_fix\_fte()**
   * Since we did not schedule any sections if doing so would exceed the teacher’s max\_fte, we should only have to fix deficits. There will likely be quite a few deficits because users set the max\_section\_size on the basis of the maximum number of students allowed in a class, not the size that will produce the correct number of sections to fulfill a teacher’s min\_fte.
   * Get the fte\_min and fte\_max for teachers who are in deficit.
   * For each teacher, figure out how many terms the deficit spans. If it is more than one term, it may be appropriate to add a section of a multi-term course. Else, we will pick a single-term course.
   * Pick the appropriate course based on:  
     1. Legal courses this teacher can teach
     2. Course term length is appropriate to fill the deficit
     3. Of courses that fit the first two criteria, pick a course that has the highest number of students above the minimum section size in the computed average\_class\_size after a new section has been added.  
        1. Pass the resulting course(s) (all course parts if it is a multi-term course) and term(s) to the create\_sections function to create the additional section(s).
   * **TO DO:** Call check\_fix\_fte function recursively until you have filled all deficits that are possible to do programmatically.
3. **Manually fix any remaining FTE violations or other violations.**
   * If there were no courses found that would fill the deficit, there may still be teachers whose FTE is outside the min\_fte or max\_fte either for a specific term or for the year.
   * **TO DO:** In the **view\_assigned\_fte()** function, there should be an alert on the full-year line if one of the terms is generating an alert. Currently, it is possible that full-year FTE is not in violation, but a single term is, and you wouldn’t see that until you manually expand each row.
   * If you create a new section or edit a section using **add\_edit\_section()**, you get to select a term (for a single-term course) and one or more teachers. If the course is a multi-term course, it will be distributed across the appropriate terms.
     1. **TO DO:** This only currently works for a full-year, multi-term course. We do not have a way for the user to specify multiple terms if the course is multi-term but less than a full year.
   * If the course is multi-term, the teacher will be added to all course parts.
   * If you are editing and changing the teacher or term, changes will apply to all parts of a multi-part course. If you delete part of a multi-part course, it deletes all parts.
   * **TO DO:** House assignment has not yet been done.
   * Other violations include:
     1. Courses with requests but no sections created or not enough sections created to satisfy all requests.  
        1. In this case, you need to delete the requests that won’t be filled. In the UI, if you delete the course, it will delete all associated requests, or you can delete individual requests. Or, you can increase the minimum section size so that it is possible to satisfy the requests that have been made.
     2. **TO DO:** Not currently checking for this, but do all rooms have non-zero capacity indicated.
     3. **TO DO:** Not currently checking for this, but do all students have requests totaling the minimum and maximum allowed FTE per student?
4. **Dump the data.**
   * If all violations have been resolved, the data is ready to be dumped. **schedule\_data.php** takes care of this and writes all the data to one file, **schedule\_data\_complete.txt**.
   * The dump process is relatively straightforward except for these things to keep in mind:  
     1. Need to map everything back from the abstracted course\_id to the original course\_id. This means:  
        1. Create a request for each course part
        2. Get the possible teachers, times, rooms, houses for each part
     2. Fill the **course\_sequence** table.  
        1. Get all courses from courses\_split\_by\_term that are more than one term in length. Put these in the sequence table. Since these courses have been arbitrarily split, they generally require same time/room/teacher.   
             
           **TO DO:** Allow the school to say whether or not tying these courses together in this way is required or not.
        2. Get all courses from courses\_multi\_part. The course\_id here is the course\_parent\_id, so get the course\_id for this year from the courses table; then find that id in the courses\_split\_by\_term table to get the abstracted id. Put these in the sequence table. Since these courses are actually separate courses, they generally do not require that we keep them in the same period throughout the year.
5. **Other TO DOs.**
   * **We don’t yet handle complex day patterns (or even very simple day patterns for that matter). We do not yet have encoding for a two-day schedule that indicates whether a course must meet every day or every other day.**
   * Allow a school to assign some students to a section in advance, e.g. certain special education students must be in a specific section with a certain teacher or teacher(s).
   * Some schools prefer that instead of assigning specific students to a section, you reserve spots for certain students, like, hold 5 special education spots in this section.
   * We do not yet have a way to require that several courses be taught in the same room by the same teacher at the same time. E.g. Latin I/II/III/IV might be taught by the school’s only Latin teacher in a one-room-school-house model.
   * We don’t have a way to indicate other time slot requirements, such as that courses must be in consecutive periods or terms for whatever reason.