Database Management Systems

BCPR203

Assignment one

Semester Two, 2018

Name : Tom Son (Dongwoo Son)

Student Number : 99182852

“

There are two folders in Zip file, Draft and Final, Yes I will also leave my github address to show the progress as well but I’d like to submit every files but will be messy.

So basically Draft folder will let you see how I started from scratch to final version

And Final folder is the one you will mainly see to give marks ☺

”

Report

1 & 4 & 5 Choice of entities & Connectivity / Relationships & Use of bridging entities:

So, I have 5 entities, Student, Course, Sessions, Enrolment and Attendance.

Student: So hence our project scope is all CFF staff to use to track and monitor students across all eight locations. From here we will need at least ‘Student’ and ‘Staff’. However, after I did all the queries, realized that I don’t really need to have entity ‘Staff’, Yes by having Staff with extra bridge table, I will do have more than 10 entities which is kind of mentioned on marking schedule says ‘Not enough entities (i.e. fewer than 10)’ but keeping ‘Staff’ entity just because to make it more than 10 entities didn’t make sense to me as you will eventually see those were useless data. So, I rather keep it minimum to be tidier, so I can manager easy and no issue of getting all the queries we need.

Course: From given spreadsheet, one of the tabs was Course, which contains all the information of course code, number of days, number of weeks and which program they are belonging to. So, if we did normalization, this will be a table we need to keep.

Sessions: From given spreadsheet, there were section for dates, based on my assumption, this will be pre-booked for school schedule, and this will be used to check attendance, but my cased I used this entity to make it as booked date, time and branch for each student.

**** Session is taken field on MySQL WorkBench, so using Sessions as this will be an exception**.

Now main entities are all there, time to get a relationship and bridge table.

Attendance: There is a connection between Student and Sessions, but Students will have many sessions and sessions will have many students as well. So, it is M-N relationship, needed to make a bridge table and named Attendance. From here, we can track the booking session that students made. I am using primary key, session ID to track unique date, time and branch for each student’s booked date.

Enrolment: This entity is obvious, there is an entity called ‘Student’ and ‘Course’, Student will have many courses but doing one at a time, Courses will have many students. So, we also need a bridge table between this and named enrolment which perfectly make sense.

2 & 3 & 5. Choice of attributes. & Choice of Keys & Use of composite

Student: There are three attributes, Peter says to use **student ID** as a primary key data type car(4) with NOT NULL, length of student ID can be changed depends on the size of business but at this point, we are doing it for school assignment so I just want to stick with the number I like to use. Also, student ID must be unique for each student, so no doubt to use as a primary key. **First Name** data type varchar(25) and **Last Name** data type varchar(25) which were given from Peter to use. Both name fields are NOT NULL as we cant have student without name.

Course: There are four attributes from excel, **Course Code** as a primary key data type char(7), all the given course code had exact length of 7, so I decided to use char(7) and its unique, which can be used as a primary key. **number of Days** data type int with unsigned so can't be a negative number, **number of sessions** data type int unsigned so can't be a negative number and **Program** ENUM. Yes, there is number of weeks, but this is calculated value based on number of days or vice versa, so I chose to use number of days instead of number of weeks, so I can calculate easy for my future query play. All my attributes are NOT NULL this case as courseCode is default NOT NULL by being a primary key and when there is a courseCode, it must have number of days, number of sessions and which program its belonged to.

Sessions: There are four different attributes this entity, **session ID** as a primary key data type int unsigned as it can't be negative. So, I can use this value toward bridge table and supply unique info, what date session is, what time that is and at which branch. **Session Date** data type date which is all the pre-booked dates like in spreadsheet, **Session Time** data type ENUM, was planning to use time but currently CFF only use 'Morning', 'Afternoon' and 'Evening' form, so stick with ENUM type, finally **branch name**, also ENUM as they are 7 different branes around.

Attendance: There are three attributes, **Student ID** as composite primary key which is connected to student ID in Student table data type char(4), **Session ID** as composite primary key, which is connected to Sessions's sessionID data type int, **attendance Date** data type TEXT which will be filled when they attended, if they haven't the field will be 'Not Attended'. So, we can track what are dates that student attended. But since it's a TEXT data type, when we run the queries, I also had to convert string into date form. By converting it, I can now compare this with dates values. All attributes are also NOT NULL as studentID and sessionID are composite primary key, so default NOT NULL and attendanceDate will have either the string form of date they attended or Not Attended.

Enrolment: By doing normalization with given spreadsheet, we can see that this enrolment table will have attributes of **Student ID** as composite primary key which is connected to student ID in Student table data type char(4), **course Code** composite primary key which is connected to course code from Course table data type char(7), **start date** data type date determines each students start date as they can enroll at any time they want, **last moodle active** data type date, track when was their last active on moodle, **completed** data type boolean, will tell whether students finished

their course or not and **Note** data type text, any notes that need to be in.

startDate, completed, studentID and CourseCode are NOT NULL as studentID and courseCode are composite primary key, no doubt. For startDate it the date student start, so can't be null for a date and completed will have either FALSE or TRUE depends on whether they finished course or not.

6. Extent of normalisation

Since I did this in Visio, I will upload it as a separate file.

Assumptions / Decisions I made

1. As there were many more entities over drafts but when I was working on queries, because of (i.e. fewer than 10) from marking guide, to make useless entities don't make sense to me to avoid it as if it's useless, don't even think it will be counted. So I decided to keep them minimize so easy to maintain, reduce complexity, keep core entities we need for our queries.

2. Reason I made auto increment value for sessionID in Sessions, is because session Table has attributes Session Date, Session Time and Branch name. However, unlike normal industry, CFF offers same course, at same time at 7 different branches. E.g BCPR203 can happen at City, Hornby, Bishopdale on 2018-08-27 at 9am simultaneously. Which none of those values can be unique to determine a single row. So, I decided to generate auto increment integer to specify a set of data. E.g 1 will be BCPR203, 9AM at City, 2 will be BCPR203, 9AM at Hornby, 3 will be BCPR203, 9AM at Bishopdale… etc.

3. sessionID in Attendance, this is used to determine specific date, time and branch name for each student's booked information, so I used this value as 'Booked date'. Only reason I kept it as sessionID is because it's a composite key that's connected to Sessions which has a primary key sessionID, if table Sessions have a primary key called… something like bookedDateKey, doesn't make sense for table Sessions. So, I kept using sessionID to determine booked class.

4. Since Booking is based on SessionID which is a primary key that can't be NULL, so for my assumption / rule is, if student decided to do drop-in instead of doing booking and attend, their default info to be saved for bookedDate (sessionID info) will be random year that has no data for session Time and Session branch, (sessionID = 1 from Sessions table). It will be changed every time they make a booking, otherwise default date as we are focusing on student engagement based on attendance as they

can still do drop-in.

5. Booking must be made when attendance is still on their choice. However, based on our business rule they still need to show up more than two times a week.

6. For Query 2, I assume when percentage of their progress in terms of time frame, is higher than 75% then it means students are near the end of their course duration. As 75% of classes saying they are ready to sit their assessment. This query is asking me for course duration not number of sessions attended but still stick with 75%. E.g 100 days for course A and its been 75 days since student B started course A, then it notifies staff as its 75% >=.

7. I mentioned above as well but to clear them out, as for this assignment, we are only focusing on student engagement. Also, Peter cleared out for us that having Branch is not relevant and not required.

8. My queries didn't really need some tables. Based on assignment guideline, there were no reasons to keep Staff entity as nowhere to use, it was there to make my number of entities more than ten and it make my EDR looks messy, ugly. Hence, I chose to get rid of them to be tidier.

9. I added ANY_VALUE for every queries SELECT, for some reasons, if I don't put ANY_VALUE on Widnwos Workbench 8.0, giving me an error code that there is an issue by using GROUP_BY without ANY_VALUE. On the otherside, on MAC Workbench asks me not to use ANY_VALUE. But my last test was done with Ara PC at lab which is based on Windows. So I am keeping ANY_VALUE and provide reason I used ANY_VALUE.

Links

https://youtu.be/DALkbkXuem8

-- Just in case code is not running on Ara's 6.3 Workbench,
I just found out I am running same 8.0 Workbench for both my mac and Windows but my code is running well on Windows but Mac.
I am sure you will still be able to read code and understand that they should work but just in case if you wanted to play but not working on other version of Workbench, just example of what are the outputs like.

Github

--- for Final (also have draft)

https://github.com/SnowNooDLe/BCPR203_2018_PART1_FINAL