CS-301:DATABASES

COURSE PROJECT , PHASE-1

SUBMITTED TO:

Dr. Viswanath Gunturi

Assistant Professor

Department of Computer Science and Engineering

Indian Institute of Technology, Ropar

SUBMITTED BY:

Name	Entry Number
Vishawam Datta	2019CSB1305
Manav Mago	2019EEB1274

Schema Details

SR NO	Table Name	Attributes	Primary key
1	Course_Catalog	a.Course_id b.List_Prerequisite(List data type) c.LTPS d.C(Representing credits)	Course_id
2	Academic_Semester	a.Year b.Semester	Year,Semester
3	Slots	a.Slot-id	Slot-id
4	Instructors	a.Instructor_id b.Name c.Department d.user_name E.batch_advisor_year	Instructor_id
5	Students	a.Start_Year b.student_name c.Dept_Name d.user_name e.Student_ID	Student_ID
6	Course_Offering_Year_Sem	a.Time_Slot b.CGPA c.Batches_Allowed_List(List Data type) d.Instructor_id e.Course_id	Course_id
7	Registration_Course_id_Sem_Year	a.Student_id	Student_id
8	transcript_student_id	a.Year b.Semester c.credits	Course_id

		d.Grade e.Course_id	
9	Grades_Course_id_sem_Year	a.Student_id b.Grade	Student_id
10	current_courses_student_id	a.course_id b.slot c.credits	course_id
11	tickets_Student_id	a.status b.Year c.Course_id d.Semester	Course_id, Year, Semester
12	tickets_instructor_id	a.student_id b.course_id c.status d.Year e.Semester	Course_id, Student_id, Year, Semester
13	batch_advisor_start_year	a.student_id b.course_id c.status d.Year e.Semester	Course_id, Student_id, Year, Semester
14	dean_tickets	a.student_id b.course_id c.status d.Year e.Semester	Course_id, Student_id, Year, Semester

Table 1:Details of the tables in schema

Schema

1.Course_Catalog: This table contains the details about all the existing courses in the university ,the details includes the pre-specified prerequisite courses,credit for that particular course ,LTPSC structure and it's ID as the primary key.This table will be hardcoded i.e to add any new course dean will be calling the procedure as described in the procedure section to add that particular course in the table.

2.Academic_Semester: Any new addition to this table marks the beginning of the new semester. This table will be handy in creating the new tables particular to that semester as well as starting the cycle of procedures for eg:instructor offering new courses and student registering for new courses etc.

3.Slots: Again this table would be hard coded i.e this table would not be declared dynamically and so it will be relevant for all the years and semesters.This table contains information on the type of slots the courses can be taught in for eg:'PCDE','HSME' etc

4.Instructors: This table would contain information about all the instructors present in the university.Information pertaining to their department ,their name,their id(which also be the primary key for this table) and most importantly information about them being the batch advisor or not .If the batch advisor contains value such as '2019' this will mean that the particular instructor will be batch advisor for 2019 batch students whereas if it states '0000' this will mean that the particular instructor will not be acting as advisor for any batch.

5.Students: This table would also be hardcoded and whenever a new student joins the university all the information regarding him/her would be present in this table.With his ID as the primary key for this table we will also be having information regarding the joining year ,department name and his user-id with which he would be able to access the relevant tables .

6.Course_Offering_Year_Sem:This table would be created dynamically which means that as soon as the new semester is declared this table would be created allowing instructors to offer the courses and the information about the offered course for that particular semester would be present in this table such as

information about the batches and department allowed for that particular course would be present in the form of a 2-d array For eg:([['2020','CSE'],['2019','EE']]) The table name for the year 2021 2nd semester would be written/declared as Course_offering_2021_2 etc.

7.Registration_Course_id_Sem_Year:This table would also be declared dynamically and this would be created for each of the offered courses.So if a student registers for a particular course and clears all the requirements like the cgpa,prerequisites courses,department and year and also if the course does not overlap with any of his/her registered course slot then the student would be registered in the course. For eg: The name of the table would be like Registration _CS301_2_2021.

8.Grades_Course_id_sem_Year:One such table for each course offered in the particular semester.This table would contain the grades which will be uploaded by the instructor for the particular course and for the particular student.

9.transcript_student_id:This table is created for each and every student The table pertains to the information of the courses which a particular student has taken in the past semesters.The information includes the courses taken ,the grade obtained by the student and the year and semester in which he took those courses.

10.current_courses_student_id:One such table for each student.This table would be containing the information regarding the courses taken by particular student in a semester and a year.This table is used for checking for the slots of the courses in the current semester and also for checking the 1.25 credit rule before registering a student in the course.

11.tickets_Student_id: This table is maintained for every student dynamically and whenever a student raises a ticket an entry is made corresponding to that particular course.

12.tickets_instructor_id:One such table for each instructor.This table allows the instructor to see the tickets related to his course and also records his action in the status column.

13.batch_advisor_start_year:This table is made for the batch advisor of the

students of the corresponding year ,table would be different for the batch advisor of the 2019 joining students and 2020 joining students.Similar to the instructor tickets table batch advisor can also see the tickets related to his batch and take appropriate action.

14.dean_tickets: This table would be hardcoded and all the tickets raised by the students will be present here as well so that the dean can take appropriate action .

IMPORTANT PROCEDURES : -

1.Insert_course_catalog -> This procedure will be used by the admin/dean who will insert a particular course in the course_catalog table.

2.insert_slot->This procedure will be used by the admin/dean to insert a new slot.

3.insert_academic_sem->This procedure will be used by the admin/dean to declare a new semester by inserting a row into this table.

4.create_student->The admin will call this procedure and this will create a new student user and set its username and password which it will receive as inputs along with other necessary student details such as the id , name , department etc. and makes an entry corresponding to this student into the students table.Along with this, the procedure also creates certain tables which are specific to a particular student , these tables are transcript_student_id , current_course_student_id and tickets_student_id.Along with this the student is given permission to view the course_catalog table , permission to view the slots table ,permission to view his/her own transcript table,permission to view and insert in the current_course_student_id table , permission to view and insert in the tickets_student_id table as well.The student also gets the permission to view the instructors as well as the students table.

5.create_instructor->The admin will call this procedure and this will create a new instructor user and set its username and password which it will receive as inputs along with other necessary instructor details such as the id , name , department etc. and the procedure makes an entry corresponding to this instructor into the instructors table. The procedure also creates the tickets_instructor_id table corresponding to this instructor .Along with this the instructor is given permission

to view the course_catalog table , permission to view the slots table ,the permission to view the instructors table , the permission to view the students table and the permission to update and view the tickets_instructor_id table.

6.create_batch_advisor->The admin will call the procedure after taking in the necessary parameters and this procedure will be used to make a particular instructor a batch advisor of a particular year and also give him/her the permission to update and view the particular batch_advisor_start_year table.

7.offer_course->This procedure will be called by the instructor to simply offer a course and this procedure inserts a row in the course_offering_year_sem table by taking the necessary attributes as input.

8.create_course_registration_tables->This procedure will be called by the admin to create the registration_course_id_sem_year and the grades_course_id_sem_year tables for each course which exists in the course_offering_year_sem table , do note that this must be called by admin only when all the courses have been offered by the instructor.This procedure gives all the students permission to insert into and view the registration_course_id_sem_year tables and the instructor of the course will be only given view permission on this table.Finally the instructor will be given view ,update,delete and insert access on the grades_course_id_sem_year table so that he can enter the grades corresponding to each student, the students themselves are not given any kind of permission for the grades_course_id_sem_year table.

9.register_for_course->This procedure will be called by the student to register for a particular offered course by inserting a row corresponding to his own student_id in the registration_course_id_sem_year table for the particular course in question.Note that the student cannot freely register himself in any course,a trigger is associated with this registration_course_id_sem_year table which will check whether the student who is inserting the row is satisfying all conditions , if so then only insertion will take place.

10.generate_student_ticket->This procedure will be called by a student to generate his or her own ticket for a particular course that is being offered in the current semester.This procedure will insert a row in the concerned student's tickets_student_id table by taking in the necessary attributes as input in the procedure.

11.propogate_all_tickets->This procedure will be called by the admin after all students have registered their tickets and will be used to propagate all the tickets in all the tickets_student_id tables of all students to the particular instructor tickets table, to the particular batch advisor tickets table and to the dean's tickets table.

12.update_instructor_ticket_status->This procedure will be used by an instructor to give his decision about the ticket raised by a student for his course.This procedure will simply update the status in the corresponding row of the tickets_instructor_id table.It takes in the necessary attributes as input.

13.update_batch_advisor_status->This procedure will be used by a batch advisor to give his decision about the ticket raised by a student.This procedure will simply update the status in the corresponding row of the batch_advisor_start_year table.It takes in the necessary attributes as input.

14.update_dean_tickets_status->This procedure will be used by the dean to update the final status for a student and this procedure updates the required student's tickets_student_id table row with the final status along with updating his own table i.e the dean_tickets table.Also this procedure will successfully register the student in the required course.

15.grade_submit->This procedure will be called by the instructor to submit the grades of the course that he has taught in the semester.

16.transcript_updation->To be called by the admin at the end of the semester which will take the grades from the grades table of the course and then update all the students' transcript.

NOTE -> In addition to these procedures there are some other trivial procedures such as procedures to view the data in different tables(but will work only for those users who actually have the permission to view a particular table) which may be used in the demo however are not mentioned here.

IMPORTANT TRIGGERS :-

1.Procedure name - course_offeredtable, Trigger name - initialise_course->

This trigger works on the Academic_semester table ,so whenever a row is inserted In the Academic_semester table the trigger dynamically creates the Course_Offering_Year_Sem table by extracting the current year and sem using the new row.It is also responsible for granting relevant and required permissions on the Course_Offering_Year_Sem table to the instructors and simultaneously gives view permission to all students present in the students table on the Course_Offering_Year_Sem table.Furthermore ,this trigger will also dynamically create the batch_advisor_start_year table as we possess the information regarding the new semester and new year being declared.

2.Procedure name - trigger_register_course , Trigger name - course_id_sem_year_trigger (Example - CS301_2_2021_trigger) ->This trigger runs before insert on the registration_course_id_sem_year table and this trigger is mainly implemented so as to prevent ineligible students to register in any course they want and also prevent a student from registering a different student in a course(For example I can maliciously register my friend in a difficult course).Hence this trigger checks whether the registering student is the same one who is inserting the column in registration_course_id_sem_year table.After that the trigger checks whether the student is satisfying the prerequisites , whether he/she is from an eligible batch , whether the credit limit is being exceeded or not , time slot is clashing or not , CGPA is beyond the requirement limit or not,etc., if and only if all the above conditions are satisfied , the row is allowed to be inserted (which means that student is now registered) .In case any one of the conditions above is not met then the row insertion is denied.

3.Procedure name - trigger_instructor_validity , Trigger name - course_offering_trigger_sem_year (Example course_offering_trigger_2_2021) ->This trigger is activated before insert on the table course_offering_year_sem.This trigger basically checks that the person who is trying to offer a course is inserting a row only corresponding to himself / herself , i.e a particular instructor say with instructor_id=1 cannot offer a course posing as an instructor with id=2 because this trigger will cross check the username of the currently logged in user and the username corresponding to the inserted instructor_id.In case the above conditions don’t satisfy insertion will be denied.

Security Aspects and Permissions

SR NO	Table Name	Student Permissions	Instructor Permissions
1	Course_Catalog	VIEW-To all students	VIEW-To all instructors
2	Academic_Semester	NONE	NONE
3	Slots	VIEW-To all students	VIEW-To all instructors
4	Instructors	VIEW-To all students	VIEW-To all instructors
5	Students	VIEW-To all students	VIEW-To all instructors
6	Course_Offering_Year_Sem	VIEW-To all students	VIEW-To all instructors INSERT-To all instructors
7	Registration_Course_id_Sem_Year	VIEW-To all students INSERT-To all students	VIEW-Permission given only to the instructor who is offering the course.
8	Grades_Course_id_sem_Year	NONE	VIEW-Permission given only to the instructor who has offered the course. INSERT-Permission given only to the instructor who has offered the course UPDATE-Permission given only to the instructor who has offered the course DELETE-Permission given only to the instructor who has offered the course

9	transcript_student_id	VIEW - Only to the student whose id is student_id as mentioned in the table name.	NONE
10	current_courses_student_id	VIEW - Only to the student whose id is student_id as mentioned in the table name. INSERT - Only to the student whose id is student_id as mentioned in the table name.	NONE
11	tickets_Student_id	VIEW - Only to the student whose id is student_id as mentioned in the table name. INSERT - Only to the student whose id is student_id as mentioned in the table name.	NONE
12	tickets_instructor_id	NONE	UPDATE & VIEW - Only to the instructor whose id is instructor_id as mentioned in the table name.
13	batch_advisor_start_year	NONE	UPDATE & VIEW - Only to the instructor who is the batch advisor of the year mentioned in the table name as start_year.
14	dean_tickets	NONE	NONE

Table 2:Permissions Details of each table

NOTE -> The admin/dean has all kinds of access to all the relations and hence their permissions are not mentioned in the above table.

Some key aspects enhancing the security:

1.Trigger associated with the course offering table allows the logged in instructor to offer the course only with his own Instructor_id.If this would not have been put in place then any logged in instructor will also have been able to offer course maliciously on the behalf of some other instructors.(Example instructor A logs in and then offers a course on behalf of instructor B , such a scenario can never occur in our design since a trigger checks before such an insertion).

2.The main reason behind keeping separate tables for maintaining the grades of a course and the course registration information of students was to prevent students from seeing the grades of other student.Since all the students have insert and view access on the registration table, if grades were to be mentioned in the same table then students would also have been able to view grades of other students.

3.Tables where instructors maintain grades are different for each course so as to prevent the other instructors from potentially performing unnecessary actions if they are given view as well as insert access to the same table.Hence only the instructor who is offering the course is given edit and view privileges on the grades table of that course.

4.A student can only register himself in a particular course and cannot register other students in a course since the trigger associated with registration tables checks that such an insertion is not being performed.(Hence in our design student A cannot register student B in CS301).

5.A student cannot view other student's transcripts nor can a student raise a ticket on behalf of other students since there are separate ticket and transcript tables for each student where other students do not have any kind of privileges.Neither do instructors have the privileges to view a student's transcript.