School Database Management System

MEMBERS:

AAKASH SHRESTHA (THA076BEI001)

AAKASH SINGH THAPA (THA076BEI002)

BISHWAMBHAR DAHAL (THA076BEI009)

RAJAT DULAL (THA076BEI020)

DEPARTMENT OF COMPUTER AND ELECTRONICS ENGINEERING

INSTRUCTOR:

RAMA BASTOLA

Table of Contents

Motivation:

- We were motivated by the following problem:
  - Long Queue at entrance form at start of each academic year
  - Repetitive Process conducted at end of each academic year
  - Conflicts in communication between student and teacher
  - Tedious Marksheet report generation

Introduction

- The “School” word in our Project refers to the higher secondary school level
- This project helps to digitize the processes involved in storing information and generating useful data in schools.
- Our project uses tools such as:
- MySQL database
- Python applications(app) to add or modify data in database:
- Admin app: Given only to admin
- Student app: It will be distributed to students
- Teacher app: It will be distributed to teachers

FUNCTIONALITIES OF SCHOOL DATABASE MANAGEMENT SYSTEM

1)Easy Data Access to Authorized School Administrator/Admin

- Data regarding student and teacher can be stored and accessed by administrator in an easier, secure and convenient manner.

2)Student And Teacher Login:

- Login system allows the user to access limited resource from database which is defined by admin.
- Different functions are provided to student and teacher after they are logged in.

3) Data Security

-Every action by admin is recorded so that back tracing can be done incase data tempered situation occurs.

-Triggers are used to record updates in student teacher details.

4) Messaging System

- Facilitates the communication between teacher, student and admin.

- Features:

    i) seen/unseen status

    ii) shows inbox messages

    iii) admin could send message to group

    iv) student/teachers could be searched by keywords

5) Faster Admission Process

- has online registration feature which reduces the long queue of students and parent in front of school.

6) Marksheet Management

- Teachers can insert/update the marks of students form their home.
- Students can view their progress report simply by logging into their account.

7) User password Management

- Passwords are stored in binary hash form in DB.
- Forget password is managed by OTP system.

8) Dynamic Nature of System

- Marksheet table/view are created and deleted according to addition or deletion of category dynamically in real time.
- Provides longevity to the system.

9) Permission Control

- Has three user- student, teacher and root.

- Required permission is granted to user by admin on specific instances only.

- Some permissions are granted and revoked automatically.

10) Validation of data

- All entered data are validated before inserting into the database to avoid invalid data.

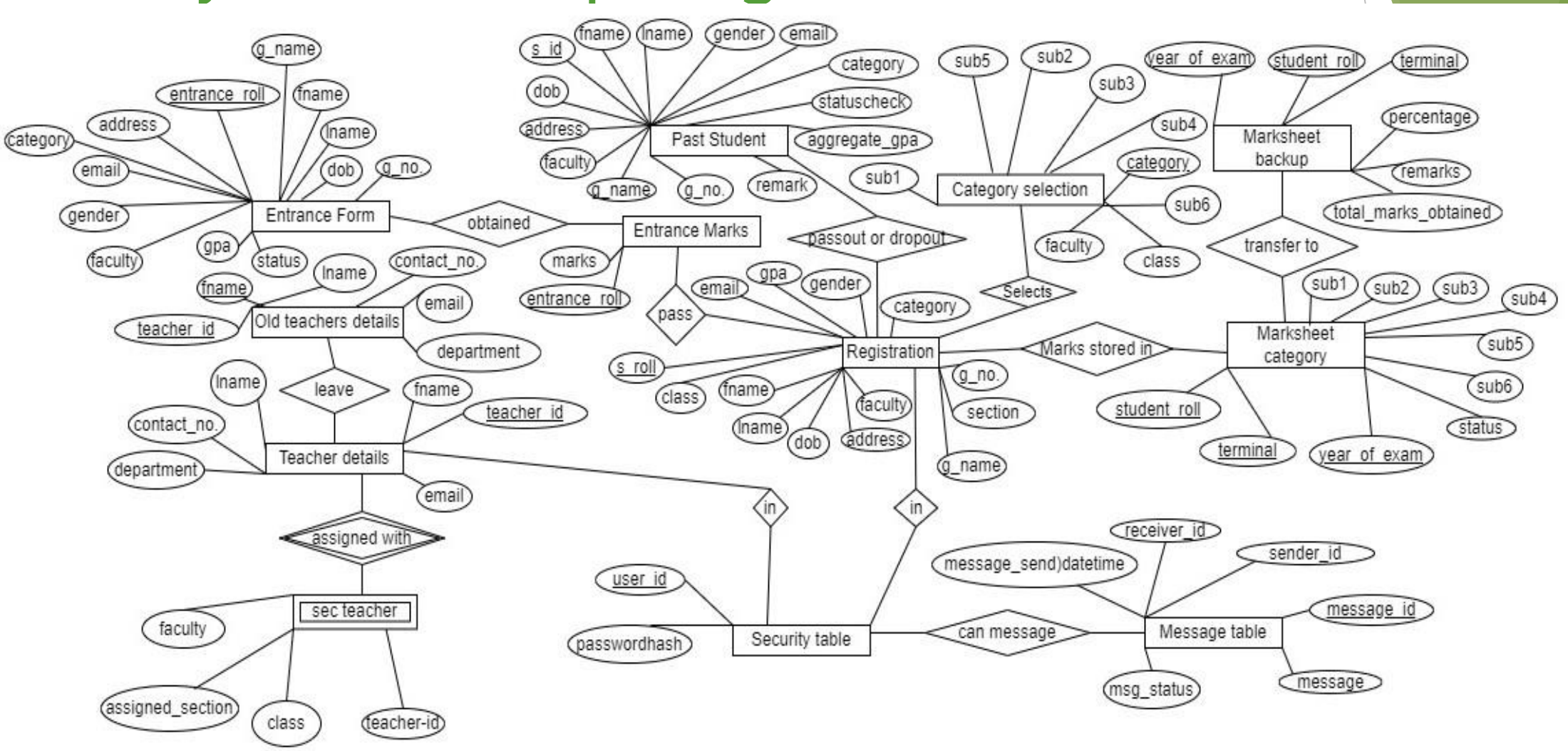
11) Data Recovery

- Commit and rollback is used for safe transaction of data.

LIMITATIONS OF OUR SCHOOL DATABASE MANAGEMENT SYSTEM

- No Proper UI Design (No Use of Front-End techniques)
- Account and Library Section which are also a part of a school are not given access to the database.
- In real colleges, Scholarships are provided to students on the basis of entrance marks. This feature hasn’t been added in our system.
- Since the entrance marks is only used by any college for admission purpose only, we have treated the entrance marks for all faculties in the same manner i.e., in a same table.

Entity Relationship Diagram



Schemas in our Project

- **Activity_record schema:**
  - Sn is the primary key with auto increment
  - The other attributes are: activity_by,activity,activity_status, activity_on and not null is assigned to all of them
  - Activity_on is of time datetime and value is generated automatically
- **Category_selection Schema:**
  - Adding a category in this table automatically generates marksheet table for the added category
  - The attributes are category,class,faculty,subject1,subject2 etc
  - Category is the primary key
  - Not null is assigned to all attributes except subject6 as it is optional subject

Continued

- **Entrance_form Schema:**
  - Its attributes are: entrance_roll,fname, dob,gpa etc
  - Entrance_roll is the primary key
  - Not null is assigned to all attributes
  - Gpa has domain constraint of 0 to 4
  - Statuscheck has domain constraint of one value between the set :(Waiting, Approved, Rejected)
- **Entrance_marks:**
  - Its attributes are entrance_roll and marks
  - Entrance roll is the primary key
  - Marks has domain constraint of 0 to 100
  - Not null is assigned to both of them

Continued…

- **Marksheet_backup**
  - It containes attributes like year_of_exam, student_roll_number,terminal,percentage, total, remark etc
  - Year_of_exam, student_roll_number and terminal are assigned primary key
  - Not null is assigned to all attributes
- **security_table**
  - Stores sensitive data like password(user_id, passwordhash)
  - User_id is primary key
  - Passwordhash has constrain not null

Continued…

- **student_update_rec**
  - Store trigger for update in student(sn,student_id,action_,action_on,address,email,category,aggregate_gpa,guatdain_name,guardain_number,remarks_further_study,action_date)
  - Sn is primary key with auto_increment
  - All other attributes are constrained to NOT NULL except aggregated_gpa,catagory, and remarks_futher_study

Continued…

- **teacher_details**
  - Store all teacher details (teacher_id,fname,laname,contact_number,email,department)
  - All attributes are constrained to NOT NULL
- **teacher_details_update_rec**
  - Store trigger for update in teacher(sn,teacher_id,action_,contact_number,email,action_date)
  - Sn is primary key with auto_increment
  - All attributes are constrained to NOT NULL

Continued…

- **technical_data**
  - Stores filter information from entrance and message table (sn, entrance_appear_stu, approved_in_entrance_stu, max_marks_in_entrance, cutoff_marks_in_entrance, avg_marks_in_entrance_of_approved_stu, avg_marks_in_entrance, total_msg, total_msg_stuTotec, total_msg_tecTostu, total_msg_tecToadmin, total_msg_adminTosec, total_msg_stuToadmin, Datetime_admin_declare_end_of_acedemic_year)
  - Sn is primary key with auto_increment

Continued…

- **Past students:**
  - Information of pass or dropout students(Student ID, Name, Address , DOB, etc.
  - Student ID is Primary Key
  - Remaining Attributes are constrained to NOT NULL except remark for further study
- **Registration:**
  - Information of students who passed entrance exam(Student roll, Name, Address, Gender, etc.
  - Student roll no.(s_roll) is Primary Key
  - Class value is default to 11 and section is default to “x”

Continued…

- **Section teacher:**
  - Information of section teacher(Teacher ID, Class, Faculty, Assigned section, etc.
  - Primary Key is Teacher ID
  - Remaining attributes are constrained to NOT NULL

Old teacher details:

- **Old teacher details:**
  - Information of teacher who left the school(Teacher ID , Name(first and last),contact no. ,email , department)
  - Teacher ID is the Primary Key
  - Remaining attributes are constrained to NOT NULL
- **Message table:**
  - Information of message (message id , sender id , receiver id,message send date and time , message status)
  - Message ID is the Primary Key and auto_increment
  - Remaining attributes are constrained to NOT NULL

Continued…

- Marksheet_category_{cat}
  - Store marks of student(year_of_exam, student_id, terminal, sub1,sub2,sub3, sub4, sub5)
  - Year_of_exam, student_id and terminal is assigned primary key

View used in our Project:

- Mark sheet calculation View
- Section teacher details view
- Mark sheet backup analysis view

Mark sheet calculation View

Field	Type	Null	Key	Default	Extra
year_of_exam	int(11)	NO		NULL	
student_roll_number	varchar(20)	NO		NULL	
terminal	varchar(1)	NO		NULL	
total_marks_obtained	bigint(15)	YES		NULL	
percentage	decimal(21,4)	YES		NULL	
remarks	varchar(4)	NO			

Remarks:

i) Dynamic View

ii) All marks from marksheet_catagory_{cat} table is added to give total marks

iii) Percentage is calculated from total marks.

iv) ‘pass’ is given in remarks if marks in all subject is >= 40 otherwise ‘fail’

Section teacher details view

Field	Type	Null	Key	Default	Extra
teacher_id	varchar(30)	NO		NULL	
full_name	varchar(61)	YES		NULL	
class	int(11)	NO		NULL	
faculty	varchar(15)	NO		NULL	
assigned_section	varchar(1)	NO		NULL	

Remarks:

i) Natural join is done between teacher_details and section_teacher.

ii) Fname and lname is concatenated to give full name.

iii) Teacher_id, full_name, class, faculty and assigned-section is displayed

Mark sheet backup analysis view

Field	Type	Null	Key	Default	Extra
year_of_exam	int(11)	NO		NULL	
faculty	varchar(20)	NO			
examTerminal	varchar(6)	YES		NULL	
max_percent	decimal(21,4)	YES		NULL	
avg_percent	decimal(25,8)	YES		NULL	
total_student	bigint(21)	NO		0	
total_passed_student	int(11)	YES		NULL	

Remarks:

i) Union is done between registration table and past_student table followed by inner join with marksheet_backup

ii) Record are grouped by year_of_examination,faculty,terminal

iii) Max_percent, avg_percent is calculated by using max() and avg ()

iv) Total number of student of particular group is calculated buy using count()

v) Function fn_totalpass is used to return total number of passed student.

vi) Terminals symbol is changed into respective word (like ‘1’ into first)

Program Flow: