K. J. Somaiya College of Engineering, Mumbai-77
(Autonomous College Affiliated to University of Mumbai)

Batch: A1 Roll No.: 1611015

Experiment No. 01

Grade: AA / AB / BB / BC / CC / CD /DD

Title: Database system designing and implement Database System Life Cycle.

Objective:

- Database system designing ,Implement Database System Life Cycle.

Expected Outcome of Experiment:

CO1: Design and tune database.

Books/ Journals/ Websites referred:

1. Elmasri & Navathe " fundamentals of Database Systems" V edition. PEARSON Education.
2. Korth, Silberschatzsu darshan "Database systems, concepts" 5th edition McGraw Hill.
3. Raghu Ramkrishnan & Johannes Gehrke "Database Management System" Tata McGraw Hill. III edition.

Pre Lab/ Prior Concepts:

Database System, ER diagram and Relation mapping, SQL

Implementation Details:

Case Study of large database system

Virtual Classroom :

Online learning tool for Students.

Functionality:

1. A faculty creates course and uploads course contents to be available for enrolled students
2. A student can enroll in a course,view course details, access content
3. Faculty can create assignments and quiz based on the course
4. Student can answer quiz and upload answers to assignments
5. Student can interact and ask doubts to other students or faculty through discussion forum
6. System generates grade based on quiz answers and assignments graded by Faculty
7. Certificate is generated for successful completion of course

- Information System Life Cycle(Macro Life cycle)

Activities w.r.t to case study

1. Feasibility Analysis

- Operational feasibility and application areas:

Virtual Classroom is an online learning tool with all features of existing online learning softwares as well as some additional features like discussion forum.

It will provide platform to students and professors of different universities to interact and create an educational environment for students to learn any course in a friendly way.

The website will be user friendly, thus any user familiar with the nuances

of the internet will be able to easily navigate through our website unless there

is a communication barrier. Initially this website will work for desktop and

further modifications will include other devices.

- Economic feasibility of information gathering:

The development of this website has minimum cost requirements. Thus

it is economic in various ways since it uses open source frameworks for development.

- Complexity of data and processes :

Virtual Classroom will have two types of users : Student and faculty. Both users have different validation and processes. Also, data of both users are stored separately. Thus, our website has minimal complexity.

- Priority of features :

Features of our website include maintenance of different types of course content, evaluation of students' performances, discussion forum, etc.

Highest priority feature is different types of course content availability to students. Other features according to priority are evaluation of students' performances and certification, and discussion forum of each course.

2. Requirement collection and analysis

- Student requirements collection:

We collected detailed requirements from students by interacting with them and discussed various problems faced by them like unavailability of video lectures, outdated courses, restriction of available subjects, etc.

- Faculty requirements collection :

We collected detailed requirements from faculties by interacting with them and discussed various problems faced by them like no facility of providing video lectures, online test conduction, time restrictions, etc.

3. Database Design :

UML diagrams and ER diagram related to Virtual classroom system were created.

ER diagram was converted to relational model

Physical database tables were created using MySql.

Implementation :

- Phase 1 (User Interface of website) :

User interface will be developed using web technologies like HTML, CSS and framework like Bootstrap.

- Phase 2 (Database design and build) :

In phase2, complete database will be built using MariaDB.

- Phase 3 (Connection of frontend with backend) :

In phase 3, connection of frontend and backend will be done using PHP.

4. Validation

The user will be asked to verify his/her credentials before accessing the contents of the website. User will be prompt to enter his/her username and password and after checking the entered data with the database,he/she will be authenticated.

Testing

During the development of the website the plan is to perform four levels of testing:

1. Unit testing – Test individual aspects whether they are functioning correctly. These include examples like whether the data is correctly stored in the database, whether course content is available to enrolled students only, the website has correct orientation, whether discussion forum is working correctly,

etc. In short here each unit is tested separately.

2. Integration testing – Whether the frontend and backend are working together, the contents are displayed correctly on the webpage etc.

The units will be integrated (combined) together and their functionality with accuracy will be checked.

3. System testing – After all the parts of the website are connected and the final product is ready after design, the testing will be conducted on the entire system to check whether the output is correct. Here the team will test for any small defect during the normal functioning of the system as a whole and fix any problems.

4. Acceptance testing – the practical application of the website will be tested here. The testing team will check whether this application will be beneficial to the user or it will induce any complexity in operating as opposed to the traditional methods.

5. Deployment, operation and maintenance

- Deployment :

Comparison between existing systems and our software:

Our Virtual Classroom will have all features provided by the existing online learning softwares as well as some additional features.

Our website also provides discussion forum for each course where students can post their queries and faculty/ex-students of that course can answer that query. This feature is not provided by many existing online learning systems.

- Training users

The users do not need any explicit training in order to use the website.

The website is user friendly and can be used by both naive and sophisticated

users. Any regular internet user (student/faculty) can use this website.

- Maintenance

Maintenance routine would include regular backup of data stored in database and deletion of inactive courses or courses that is no longer continued by faculty.

Maintenance will also include checking of consistency of data with website in case of course content uploaded by faculty.

- Database Application System Life Cycle(Micro Life Cycle)

Activities w.r.t to case study

1. System definition

The scope of the database system will be decided based on the rules governing the organization of the data in the database.

In the perspective of the considered case, the database system will consist of two users namely student and faculty, content of course, assignment and quiz related information and other information about the users.

Users:

Mainly consisting of two types:-

- Student: who will enroll in courses

- Faculty: who will create courses and assignments,quiz,etc

Applications:

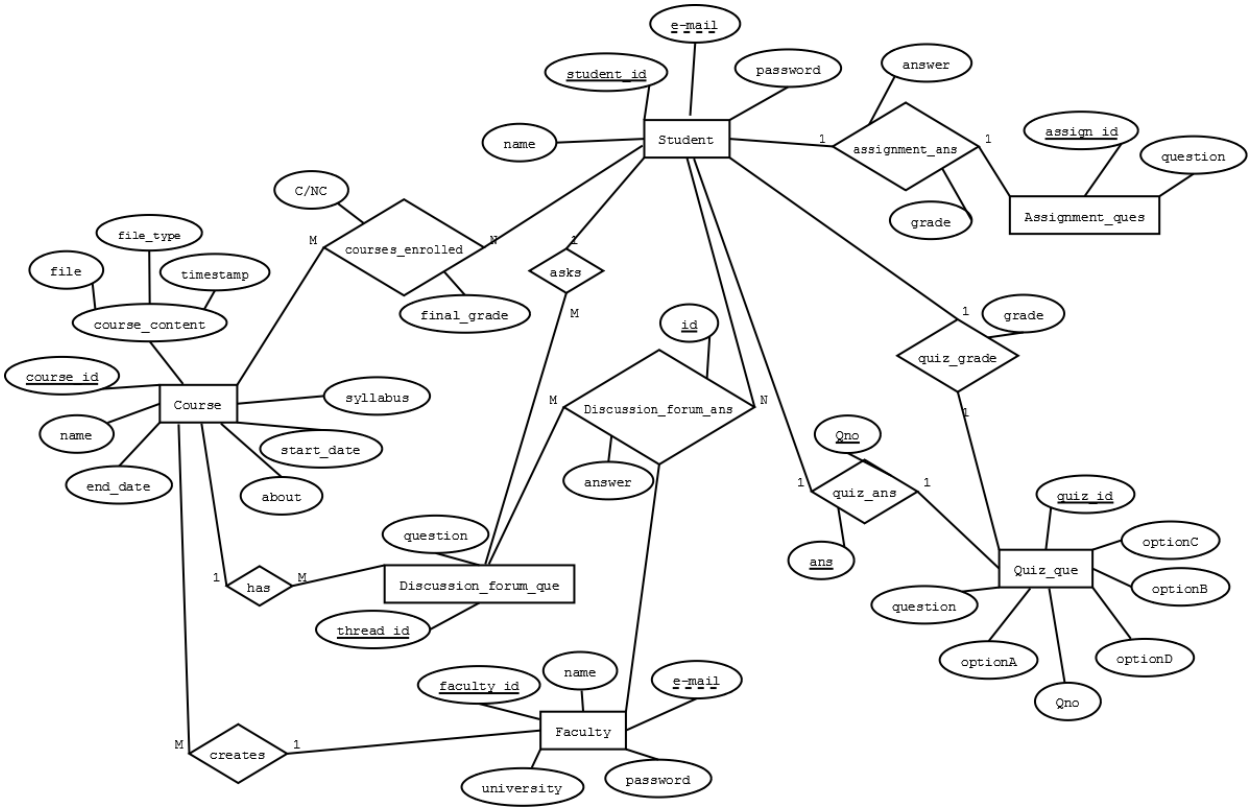
8. A faculty creates course and uploads course contents to be available for enrolled students
9. A student can enroll in a course,view course details, access content
10. Faculty can create assignments and quiz based on the course
11. Student can answer quiz and upload answers to assignments
12. Student can interact and ask doubts to other students or faculty through discussion forum
13. System generates grade based on quiz answers and assignments graded by Faculty
14. Certificate is generated for successful completion of course

2. Database design and implementation:

Design:

We will be using MySQL- a relational database design,We are going to have entity/tables for user, project, enquirer, accounts, documents and some extra tables will be added as per the new features that will be added. Logical design of the database will be shown in the UML diagrams and ERD diagrams.

ER diagram:



Implementation:

MySQL Database is being used to create the required database for the application. Conceptual design is shown in the attached ER Diagram which gives a brief overview of the database structure. As for the external definition, the database will be hosted onto a localhost for testing purposes which will be then shifted onto a server for actual deployment. Internal implementations are done in the form of a table and its associated structure. Application implementation of the database is done with the help of PHP scripting language which will fetch the records from database and display them on the website as and when required in the format specified by the developer.

Mapping to relational model:

Relational Model (Virtual Classroom):

Student :

Student_id	First_Name	Last_Name	email	password

Faculty :

Faculty_id	First_Name	Last_Name	University	email	password

Course :

Course_id	Faculty_id	Name	Start_Date	End_Date

Course_enrolled :

Student_id	Course_id	Completion_Status	Final_Grade

Course_Content :

Course_id	File	TimeStamp	File_Type

Discussion_Forum_Question :

Course_id	Student_id	Thread_id	Question

Discussion_Forum_Answer :

Course_id	Id	Thread_id	Answer	User_Type

Assignment_Question :

Course_id	Assignment_id	Question

Assignment_Answer:

Assignment_id	Student_id	Answer	Grade

Quiz_Question :

Course_id	Quiz_id	Question_no	Question	Option_A	Option_B	Option_C	Option_D	Correct_Ans

Quiz_Answer :

Quiz_id	Course_id	Student_id	Question_no	Answer

Quiz_Grade :

Quiz_id	Student_id	Course_id	Grade

Primary Keys :

Table	**Foreign Key**
Course	Faculty → Faculty_id
Course_enrolled	Student → Student_id, Course → Course_id
Course_Content	Course → Course_id
Discussion_forum_question	Course → Course_id, Student → Student_id
Discussion_forum_ans	Course → Course_id, Depending on value of attribute 'UserType' , 'id' becomes foreign key of Student or Faculty. Student → Student_id, Faculty → Faculty_id
Assignment_Question	Course → Course_id
Assignment_Answer	Assignment → Assignment_id, Student → Student_id
Quiz_Question	Course → Course_id
Quiz_Answer	Student → Student_id, Quiz → Quiz_id, Course → Course_id
Quiz_Grade	Student → Student_id, Quiz → Quiz_id, Course → Course_id

Physical Tables Implementation:

Server: 127.0.0.1 » Database: brainfirst » Table: student

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	student_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	student_fname	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
3	student_lname	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
4	email	varchar(50)	latin1_swedish_ci		No	None			Change Drop More

Server: 127.0.0.1 » Database: brainfirst » Table: faculty

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	faculty_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	faculty_fname	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
3	faculty_lname	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
4	email	varchar(50)	latin1_swedish_ci		No	None			Change Drop More

Server: 127.0.0.1 » Database: brainfirst » Table: course

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	course_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	faculty_id	int(11)			No	None			Change Drop More
3	course_name	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
4	start_date	date			No	None			Change Drop More
5	end_date	date			No	None			Change Drop More
6	about	varchar(1000)	latin1_swedish_ci		No	None			Change Drop More
7	syllabus	blob			No	None			Change Drop More

Server: 127.0.0.1 » Database: brainfirst » Table: course_enrolled

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	student_id	int(11)			No	None			Change Drop More
2	course_id	int(11)			No	None			Change Drop More
3	whether_completed	varchar(4)	latin1_swedish_ci		Yes	None			Change Drop More
4	grade	varchar(5)	latin1_swedish_ci		Yes	None			Change Drop More

Server: 127.0.0.1 » Database: brainfirst » Table: course_content

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	course_id	int(11)			No	None			Change Drop More
2	file	longblob			No	None			Change Drop More
3	time	timestamp			No	CURRENT_TIMESTAMP			Change Drop More
4	file_type	varchar(10)	latin1_swedish_ci		No	None			Change Drop More

Server: 127.0.0.1 » Database: brainfirst » Table: discussion_forum_ques

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	course_id	int(11)			No	None			Change Drop More
2	student_id	int(11)			No	None			Change Drop More
3	thread_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
4	question	varchar(500)	latin1_swedish_ci		No	None			Change Drop More

Server: 127.0.0.1 » Database: brainfirst » Table: discussion_forum_ans

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Opera

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	course_id	int(11)			No	None			Change Drop More
2	thread_id	int(11)			No	None			Change Drop More
3	user_type	varchar(10)	latin1_swedish_ci		No	None			Change Drop More
4	id	int(11)			No	None			Change Drop More
5	answer	text	latin1_swedish_ci		No	None			Change Drop More

Server: 127.0.0.1 » Database: brainfirst » Table: assignment_ques

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	course_id	int(11)			No	None			Change Drop More
2	assignment_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
3	question	longblob			No	None			Change Drop More

Server: 127.0.0.1 » Database: brainfirst » Table: assignment_ans

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	student_id	int(11)			No	None			Change Drop More
2	assignment_id	int(11)			No	None			Change Drop More
3	answer	longblob			No	None			Change Drop More
4	grade	varchar(10)	latin1_swedish_ci		Yes	None			Change Drop More

Server: 127.0.0.1 » Database: brainfirst » Table: quiz_ques

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operatio

Table structure | Relation view

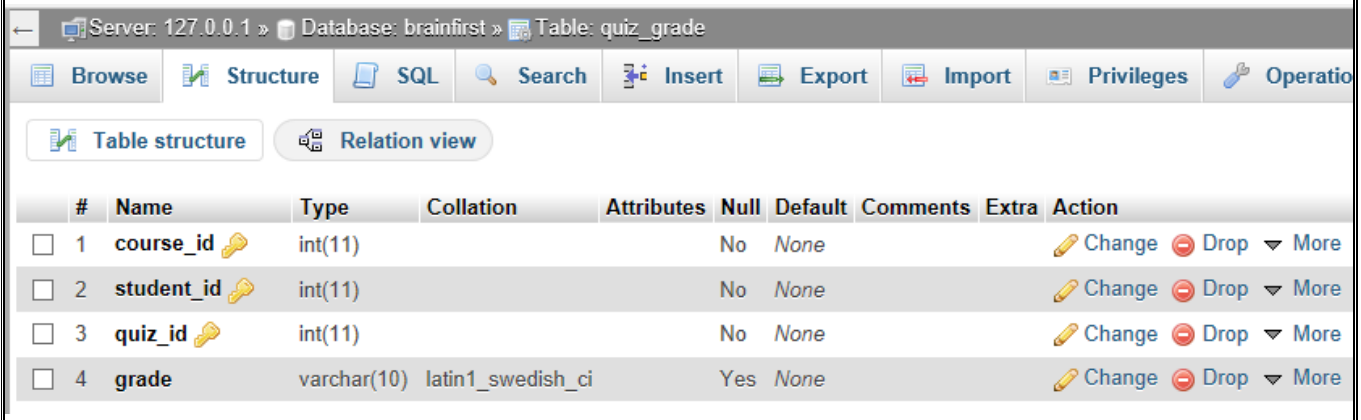
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	course_id	int(11)			No	None			Change Drop More
2	quiz_id	int(11)			No	None			Change Drop More
3	ques_num	int(11)			No	None			Change Drop More
4	question	varchar(200)	latin1_swedish_ci		No	None			Change Drop More
5	option_a	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
6	option_b	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
7	option_c	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
8	option_d	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
9	correct_answer	varchar(50)	latin1_swedish_ci		No	None			Change Drop More

Server: 127.0.0.1 » Database: brainfirst » Table: quiz_ans

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operatio

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	student_id	int(11)			No	None			Change Drop More
2	course_id	int(11)			No	None			Change Drop More
3	quiz_id	int(11)			No	None			Change Drop More
4	ques_num	int(11)			No	None			Change Drop More
5	answer	varchar(50)	latin1_swedish_ci		No	None			Change Drop More



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	course_id	int(11)			No	None			Change Drop More
2	student_id	int(11)			No	None			Change Drop More
3	quiz_id	int(11)			No	None			Change Drop More
4	grade	varchar(10)	latin1_swedish_ci		Yes	None			Change Drop More

Tuning:
Database tables were tuning using tools and optimized for better data access.

3. Database Loading:

The database will be populated through the data obtained from various Students and Faculties.All the information gathered will be put into the database in respective tables.

The credentials obtained from the user such as name, etc will also populate the database. Course content, assignments related data, quiz questions and answers will also populate the database.

Course content can be updated or removed or moved by a faculty Hence, there will be change in data in database. Tables created allow all the required data types and size of data. Also, any major system changes, if made, should be such that the users are easily able to access the website in the new system.

4. Application conversion:

The old conventional system can upload and download content about course and scope was limited.
New system allows multiple functions along with conventional functions, such as assignments, grading, discussion forum
It will easy for student to keep track of his course, and clear his/her doubts.

5. Testing and Validation:

During the development of the website the plan is to perform four levels of testing:
1. Unit testing – Test individual aspects whether they are functioning correctly. These include examples like whether the data is correctly stored in the database, the regular expression etc. In short here each unit is tested separately.
2. Integration testing – Whether the frontend and backend are working together, the contents are displayed correctly on the webpage etc. The units will be integrated (combined) together and their functionality with accuracy will be checked.
3. System testing – After all the parts of the website are connected and the final product is ready after design, the testing is conducted on the entire system to check whether the output is correct. Here the team tests for any small defect during the normal functioning of the system as a whole and fix any problems.
4. Acceptance testing – the practical application of the website will be tested here. The testing team will check whether this application will be beneficial to the user or it will induce any complexity in operating as opposed to the traditional methods.

The system will be thoroughly tested with the constraints and made sure that it won't give expected results for faulty inputs.
Form validations will be done through PHP ie. on server side, PHP validation Is maintained whenever required in system.
Database queries will be tuned and processed in the most efficient way to make sure that server is not loaded with unnecessary query processing.

6. Operation:

The new system will be executed through an organized process.
All the functionalities of the system are successfully running without bugs.
And all the errors and exceptions are handled before had.
This way the operation of the new system will be successful

7. Monitoring and Maintenance:

System maintenance:
To maintain the overall system integrity, the database should always be checked and updated when necessary while making changes to the website itself. Hence during every maintenance cycle both should be accessed together. The

system should also be maintained regularly and should be checked for any errors hence they can be spotted before they cause a disruption. For example, only faculties have option to create a course,this property should be maintained.

Performance monitoring:
The website will need to be monitored as it might slow down if it is being accessed by a lot of users at a time which might hinder its performance. Due to large amount of data and its usage at same time,performance should be monitored for better throughput.

Conclusion:

Information life cycle related to Virtual classroom system was analysed and implemented.

Post Lab Descriptive Questions:

1. What are the strategies used schema design

   Ans:
   Following are strategies used in schema design

   i) Naming conventions:
   a) Avoid using just ID as the PK of each table. It will lead to lots of aliasing when joining other tables and returning multiple IDs from several tables.
   b) Beware of using SQL Server reserved words (User, Date, etc.) in table names, column names and elsewhere. Use of a reserved word will give a syntax error unless you specify

[square brackets] around the value, making development slower and the statements longer.

c) Don't use hyphens, spaces, quotes, etc. Because they will be invalid or require [square brackets]. e.g. SELECT [category-id] FROM [custom-category]

d) Name the tables in the singular, not plural. For example, name the table Customer and Order rather than Customers and Orders. It is obvious that a table contains multiple customers and hopefully not a single row, so the plurality is somewhat redundant and may introduce inconsistency issues with some table names.

ii) Using proper constraints:

Constraints such as required fields, unique values, allowed values, etc., at the database level can perform additional validation to ensure the integrity of the data. These checks should not be the only place where validation occurs. Validation should be baked into the front end application as well. If the application catches a validation issue, a "pretty" error can be displayed to the end user.

2. What are the strategies used for View Integration explain w.r.t your case study

Ans:
View integration used in our table is N-ary integration for as multiple user schemas merge into one. Views are used in our case study to show only the required tables as per the user type. Like, assignment answer of students shouldn't be viewed by other students but only the faculty.

3. Why it is important to design the schema and applications in Parallel

Ans:
Database design coordinates with the actual view of the database in the system and how it is going to look on the server side. Making factual views of database in the form of various diagrams mentioned above is a process of database design. This is an equally important step apart

from implementation because it simplifies the efforts and confusions created while actual implementation. Database implementation is actual database creation on the physical level with the help of database querying languages like SQL or MongoDB. For implementing the database one needs to have a clear view of the system he/she is going to build thus database implementation and design should go hand in hand. When the system is live and we need to add some transactions or constraints onto the database, we again need to make the changes in the physical view.

Date: 28/08/18