Question Paper Generator Database

Submitted By :
Deepti Sharma (09)
Sandhya (32)
Shivani Tiwary (36)

Problem Statement :

A QUES_PAPER_DESIGN Software Database keeps track of all the types of Users, Courses and Questions Available on the topic.

I. **Users have a unique ID associated with them. The Database keeps track of their Login Name, First Name, Second Name, Password, Birth date, Sex, Address, and Contact Number.**

II. **The system has three possible types of user: Instructors, Learners (from any age group) and Admin.**

III. **Instructor has certain special attributes given as: Qualification, Specialization, Contribution to Question Paper, Review Pointer.**

IV. **While Learners have special properties like Degree, Grade Pointer, Attempts, Corrections Reported, and Level (defined in the system: beginner, expert).**

V. **Admin is a special type of privileged who can add questions to the questions table.**

VI. **Each and Every Instructor teaches many courses/subjects and each course can be taught by many instructors.**

VII. **Each Course has a special Id and Name. Each Course added in the system's database has at least one question of its own.**

VIII. **Questions are stored with the following properties saved: Unique serial number,, Question Statement, Answer, Four Options, Correct Submissions, Difficulty Level(Easy , Difficult, Medium).**

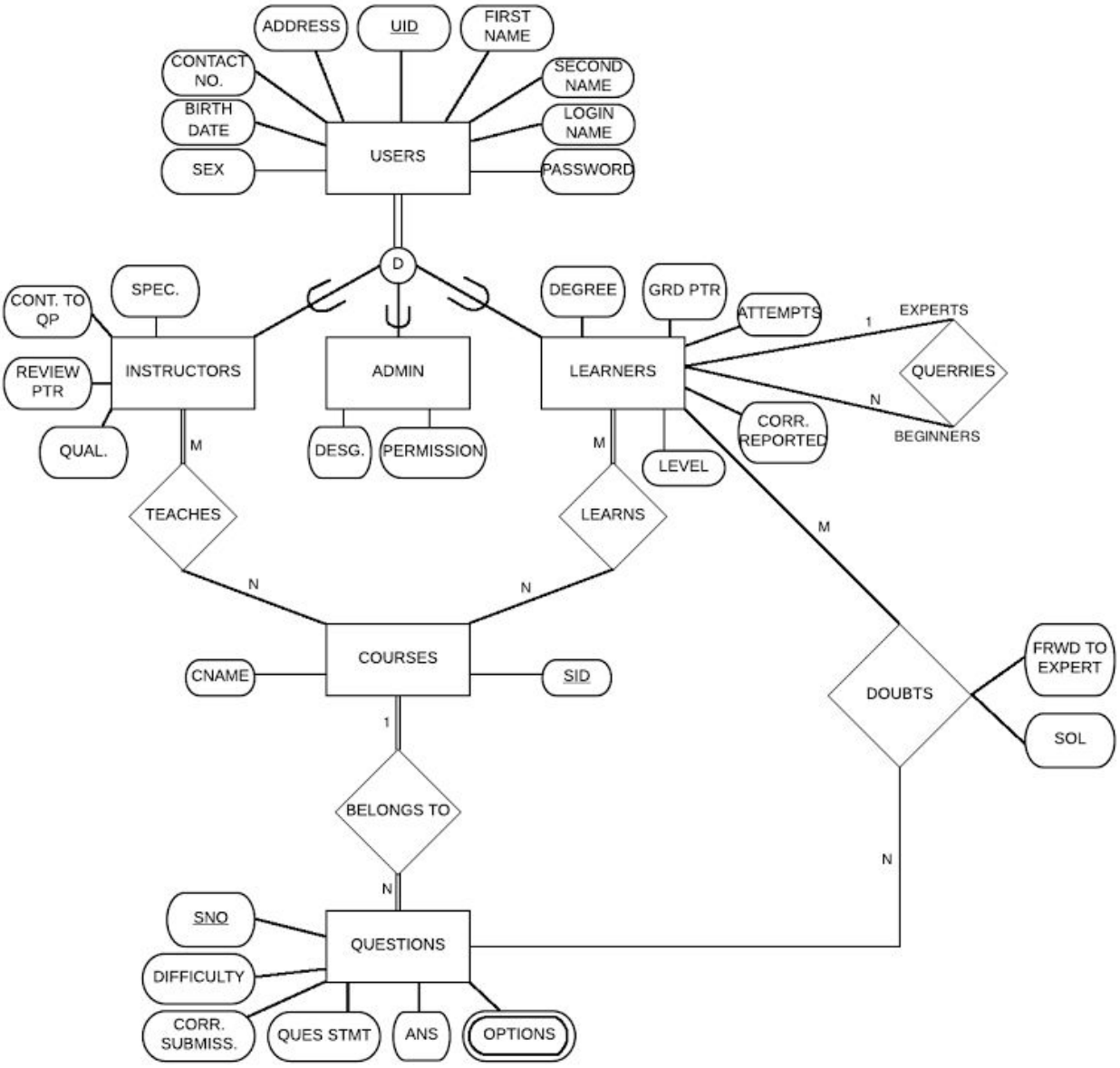
IX. **Each Question belongs to a particular course only.**

X. **A learner can learn as many courses as (s) he wants.**

XI. **Each "Beginner" learner is assigned an "expert" for help.**

XII. **A learner can mark questions and forward doubts to expert. Learners doubt certain questions. A question can be doubted by many Learners.**

Entity Relationship Diagram



Functional Dependencies :

- **UserName/UID Of A User should uniquely identify**
    - **UID -> FirstName, SecondName, LoginName, Password, BirthDate, Sex, Address, ContactNo.**

- **Learners' UID uniquely identify it's attributes**
    - **UID -> Degree, Attempts , CorrectionReported**

- **Attempts performed determine the level of the user and their grade pointer**
    - **Attempts -> Level, GradePointer**

- **Instructors' UID uniquely identify it's attributes**
    - **UID -> Specialization, Qualification, ContributionToQP, ReviewPointer**

- **Admins' UID uniquely identify it's attributes**
    - **UID -> Designation, Permissions**

- **Serial Number uniquely identifies Question , its Ans and related attributes**
    - **Sno -> QuestionStatement, Answer, CorrectSubmission, Difficulty**

- **Serial Number uniquely identifies options of each question**
    - **Sno -> Options**

- **Questions and Number of Correct Submission Of that Question determine the difficulty level for that question**
    - **QuestionStatement, CorrectSubmission -> Difficulty**

- **SubjectID uniquely determines Course Name**
    - **C_SID -> Cname**

Mapping :

USER TABLE									
UID	FIRST NAME	SECOND NAME	LAST NAME	LOGIN NAME	PASS WORD	BIRTH DATE	SEX	CONTACT NO.	ADD

INSTRUCTOR TABLE				
UID	SPECIALIZATION	QUALIFICATION	CONTRIBUTION TO QP	REVIEW POINTER

ADMIN TABLE		
UID	DESIGNATION	PERMISSION

LEARNERS TABLE						
UID	GRADE POINTER	DEGREE	ATTEMPTS	CORRECTION REPORTED	LEVEL	EXPERT UID

COURSES TABLE	
C_SID	CNAME

LEARNS TABLE	
L_UID	C_SID

TEACHES TABLE	
I_UID	C_SID

DOUBTS TABLE			
L_UID	SNO	FORWARD TO EXPERT	SOLUTION

QUESTIONS TABLE					
SNO	QUESTION STATEMENT	ANSWER	DIFFICULTY	CORRECTION SUBMISSION	C_SID

QUESTION_OPTION TABLE	
SNO	OPTIONS

Normalisation :

Using the functional dependencies we identified , We can normalize the Question and Learners Table as follows :

1. **QUESTION TABLE :**

   We have :

   - **QuestionStatement, CorrectSubmission -> Difficulty**
   - **Sno -> QuestionStatement, Answer, CorrectSubmission**

   So we get :

   Questions_1(Sno, QuestionStatement, Answer)

   Questions_2(QuestionStatement, CorrectSubmission, Difficulty)

   This decomposition is in BCNF .

QUESTION_1 TABLE		
SNO	QUESTION STATEMENT	ANSWER

QUESTION_2 TABLE		
QUESTION STATEMENT	CORRECT SUBMISSION	DIFFICULTY

2. LEARNERS TABLE :

We were given :

- **Attempts -> Level, GradePointer**
- **UID -> Attempts , Degree, CorrectionReported**

So we get :

Learners (UID, Degree, Attempts, CorrectionReported)

Attempts (Attempts, GradePointer, Level)

This Decomposition is also in BCNF.

LEARNERS TABLE:			
UID	DEGREE	ATTEMPTS	CORRECTION REPORTED

ATTEMPTS TABLE		
ATTEMPTS	GRADE POINTER	LEVEL