**Database Schema for CAPS**

**1. Roles Table**

* **roleID** (INT, PRIMARY KEY, AUTO\_INCREMENT) – Unique role identifier
* **roleName** (VARCHAR(50), UNIQUE, NOT NULL) – Role name (e.g., Student, Instructor, Program Chair, Dean)

**Indexing Strategy:**

* PRIMARY KEY (roleID) – Ensures fast lookup by role ID.
* UNIQUE INDEX (roleName) – Ensures role names are unique.

**2. Campus Table**

* **campusID** (INT, PRIMARY KEY, AUTO\_INCREMENT) – Unique campus identifier
* **campusName** (VARCHAR(100), NOT NULL) – Name of the campus

**3. Users Table**

* **userID** (INT, AUTO\_INCREMENT) – Unique user identifier
* **userCode** (VARCHAR(20), PRIMARY KEY, UNIQUE, NOT NULL) – Unique code for users
* **username** (VARCHAR(100), NOT NULL) – User’s full name
* **eMail** (VARCHAR(100), UNIQUE, NOT NULL) – User’s email address
* **password** (VARCHAR(255), NOT NULL) – Hashed password
* **roleID** (INT, FOREIGN KEY REFERENCES Roles(roleID)) – User’s role
* **campusID** (INT, FOREIGN KEY REFERENCES Campus(campusID)) – User’s associated campus

**Indexing Strategy:**

* PRIMARY KEY (userCode) – Optimizes searches by user code.
* UNIQUE INDEX (eMail) – Ensures email uniqueness and speeds up lookups.
* INDEX (roleID, campusID) – Speeds up role-based filtering and campus-related queries.

**4. Subjects Table**

* **subjectID** (INT, AUTO\_INCREMENT) – Unique subject identifier
* **subjectCode** (VARCHAR(20), PRIMARY KEY, UNIQUE, NOT NULL) – Unique subject code
* **subjectName** (VARCHAR(100), NOT NULL) – Name of the subject

**Indexing Strategy:**

* PRIMARY KEY (subjectCode) – Optimizes searches by subject code.

**5. Faculty\_Subjects Table**

* **facultySubjectsID** (INT, PRIMARY KEY, AUTO\_INCREMENT) – Unique ID for faculty-subject assignment
* **facultyCode** (VARCHAR(20), FOREIGN KEY REFERENCES Users(userCode)) – Faculty identifier
* **subjectCode** (VARCHAR(20), FOREIGN KEY REFERENCES Subjects(subjectCode)) – Subject identifier

**Indexing Strategy:**

* INDEX (facultyCode, subjectCode) – Optimizes faculty-subject lookups.

**6. Questionnaires Table**

* **questionnaireID** (INT, PRIMARY KEY, AUTO\_INCREMENT) – Unique questionnaire ID
* **title** (VARCHAR(255), NOT NULL) – Questionnaire title
* **subjectCode** (VARCHAR(20), FOREIGN KEY REFERENCES Subjects(subjectCode)) – Subject identifier
* **createdBy** (VARCHAR(20), FOREIGN KEY REFERENCES Users(userCode)) – User who created the questionnaire
* **status** (ENUM('Pending', 'Approved', 'Rejected')) – Approval status
* **coverage** (VARCHAR(50), NOT NULL) – Coverage type (e.g., Midterm, Final)
* **difficulty** (VARCHAR(20), NOT NULL) – Difficulty level
* **printable** (BOOLEAN, DEFAULT FALSE) – Indicates if it is printable

**Indexing Strategy:**

* INDEX (subjectCode) – Speeds up filtering by subject.
* INDEX (createdBy) – Optimizes queries related to questionnaire authors.

**7. Questions Table**

* **questionID** (INT, PRIMARY KEY, AUTO\_INCREMENT) – Unique question ID
* **questionnaireID** (INT, FOREIGN KEY REFERENCES Questionnaires(questionnaireID)) – Associated questionnaire
* **questionText** (TEXT, NOT NULL) – Question content
* **imagePath** (VARCHAR(255), NULL) – Optional image path
* **status** (ENUM('Pending', 'Approved', 'Rejected')) – Approval status
* **score** (INT, NOT NULL) – Designated score of each question

**Indexing Strategy:**

* FULLTEXT INDEX (questionText) – Enables fast full-text search.
* INDEX (questionnaireID) – Optimizes questionnaire-based queries.

**8. Exams Table**

* **examID** (INT, PRIMARY KEY, AUTO\_INCREMENT) – Unique exam ID
* **studentCode** (VARCHAR(20), FOREIGN KEY REFERENCES Users(userCode)) – Student taking the exam
* **subjectCode** (VARCHAR(20), FOREIGN KEY REFERENCES Subjects(subjectCode)) – Exam subject
* **type** (VARCHAR(50), NOT NULL) – Exam type (Practice, Qualifying, etc.)
* **score** (DECIMAL(5,2), DEFAULT 0) – Exam score

**Indexing Strategy:**

* INDEX (studentCode, subjectCode) – Improves performance for exam-related queries.

**9. Exam Questions Table**

* **examQuestionID** (INT, PRIMARY KEY, AUTO\_INCREMENT) – Unique exam-question mapping ID
* **examID** (INT, FOREIGN KEY REFERENCES Exams(examID)) – Associated exam
* **questionID** (INT, FOREIGN KEY REFERENCES Questions(questionID)) – Associated question
* **choiceID** (INT, FOREIGN KEY REFERENCES Choices(choiceID)) – Selected choice
* **isCorrect** (BOOLEAN) – Whether the answer was correct

**Indexing Strategy:**

* INDEX (examID, questionID) – Speeds up exam question retrieval.

**10. Choices Table**

* **choiceID** (INT, PRIMARY KEY, AUTO\_INCREMENT) – Unique choice ID
* **questionID** (INT, FOREIGN KEY REFERENCES Questions(questionID)) – Associated question
* **imagePath** (VARCHAR(255), NULL) – Optional image path
* **choiceText** (TEXT, NOT NULL) – Choice content
* **isCorrect** (BOOLEAN, NOT NULL) – Indicates if it is the correct answer

**Indexing Strategy:**

* INDEX (questionID) – Improves choice retrieval for questions.

**11. Print Table**

* **questionnaireID** (INT, FOREIGN KEY REFERENCES Questionnaires(questionnaireID)) – Associated questionnaire
* **userCode** (VARCHAR(20), FOREIGN KEY REFERENCES Users(userCode)) – User who requested the print

**12. Practice Settings Table**

* **settingsID** (INT, PRIMARY KEY, AUTO\_INCREMENT) – Unique practice settings ID
* **studentCode** (VARCHAR(20), FOREIGN KEY REFERENCES Users(userCode)) – Student using the settings
* **subjectCode** (VARCHAR(20), FOREIGN KEY REFERENCES Subjects(subjectCode)) – Associated subject
* **timerEnabled** (BOOLEAN, DEFAULT FALSE) – Whether timer is enabled
* **difficultyDistribution** (TEXT, NOT NULL) – Difficulty distribution settings

**13. Notifications Table**

* **notificationID** (INT, PRIMARY KEY, AUTO\_INCREMENT) – Unique notification ID
* **userCode** (VARCHAR(20), FOREIGN KEY REFERENCES Users(userCode)) – User receiving the notification
* **message** (TEXT, NOT NULL) – Notification content
* **type** (VARCHAR(50), NOT NULL) – Type of notification
* **isRead** (BOOLEAN, DEFAULT FALSE) – Whether the notification has been read
* **sentAt** (DATETIME, DEFAULT CURRENT\_TIMESTAMP) – Time notification was sent