**Updated Tables**

1. **Users**
   * user\_id (Primary Key)
   * username
   * password
   * role (Student, Admin, HR, Employee)
   * email
   * full\_name
   * is\_approved (Boolean, default false for students)
   * Other relevant user information (e.g., address, phone number)
2. **Students**
   * student\_id (Primary Key)
   * user\_id (Foreign Key referencing Users)
   * batch\_id (Foreign Key referencing Batches)
   * Other student-specific information (e.g., program, current courses)
3. **Admins**
   * admin\_id (Primary Key)
   * user\_id (Foreign Key referencing Users)
   * department\_id (Foreign Key referencing Departments)
   * Other admin-specific information (e.g., position, contact details)
4. **HR**
   * hr\_id (Primary Key)
   * user\_id (Foreign Key referencing Users)
   * department\_id (Foreign Key referencing Departments)
   * Other HR-specific information (e.g., role, contact details)
5. **Employees**
   * employee\_id (Primary Key)
   * user\_id (Foreign Key referencing Users)
   * designation
   * Other employee-specific information (e.g., hire date, salary, performance reviews)
6. **Batches**
   * batch\_id (Primary Key)
   * batch\_name (e.g., Year of enrollment, e.g., "2024")
   * start\_date
   * end\_date
   * department\_id (Foreign Key referencing Departments)
7. **Departments**
   * department\_id (Primary Key)
   * department\_name
   * head\_of\_department (Foreign Key referencing Admins)
   * Other department-specific information (e.g., location, contact details)
8. **Events**
   * event\_id (Primary Key)
   * event\_name
   * description
   * event\_date
   * location
   * organizer\_id (Foreign Key referencing Users who are Admins)
   * Other event-related information (e.g., registration deadline, maximum attendees)
9. **Registrations**
   * registration\_id (Primary Key)
   * event\_id (Foreign Key referencing Events)
   * student\_id (Foreign Key referencing Students)
   * registration\_date
   * Other registration-related information (e.g., status, payment details if applicable)

**Tables**

| **Table Name** | **Columns** | **Relationships and Notes** |
| --- | --- | --- |
| **Users** | user\_id (PK) <br> username <br> password <br> role <br> email <br> full\_name <br> is\_approved <br> ... | Stores all users (students, admins, HR, employees). role distinguishes roles. is\_approved for students' approval. |
| **Students** | student\_id (PK) <br> user\_id (FK to Users) <br> batch\_id (FK to Batches) <br> ... | Links students to their user profiles and assigned batches. |
| **Admins** | admin\_id (PK) <br> user\_id (FK to Users) <br> department\_id (FK to Departments) <br> ... | Represents administrative staff linked to departments. |
| **HR** | hr\_id (PK) <br> user\_id (FK to Users) <br> department\_id (FK to Departments) <br> ... | Represents HR personnel linked to departments. |
| **Employees** | employee\_id (PK) <br> user\_id (FK to Users) <br> designation <br> ... | Stores employee details and designations assigned by HR. |
| **Batches** | batch\_id (PK) <br> batch\_name <br> start\_date <br> end\_date <br> department\_id (FK to Departments) <br> ... | Manages batches associated with specific departments. |
| **Departments** | department\_id (PK) <br> department\_name <br> head\_of\_department (FK to Admins) <br> ... | Organizes departments with designated heads (admins). |
| **Events** | event\_id (PK) <br> event\_name <br> description <br> event\_date <br> location <br> organizer\_id (FK to Users) <br> ... | Represents events organized within the institute. organizer\_id links to admin organizing the event. |
| **Registrations** | registration\_id (PK) <br> event\_id (FK to Events) <br> student\_id (FK to Students) <br> registration\_date <br> ... | Tracks registrations of students for events. Links students to events they register for. |

### Tables Individual

#### **Users Table**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| user\_id | INT | PRIMARY KEY, AUTO\_INCREMENT | Unique identifier for each user. |
| username | VARCHAR(50) | NOT NULL | Username for login purposes. |
| password | VARCHAR(100) | NOT NULL | Hashed password for authentication. |
| role | VARCHAR(20) | NOT NULL | Role of the user (Student, Admin, HR, Employee, etc.). |
| email | VARCHAR(100) | NOT NULL | Email address of the user. |
| full\_name | VARCHAR(100) | NOT NULL | Full name of the user. |
| is\_approved | BOOLEAN | DEFAULT FALSE | Approval status for students. |
| Other relevant user information as needed. |  |  |  |

#### **Admins Table**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| admin\_id | INT | PRIMARY KEY, AUTO\_INCREMENT | Unique identifier for each admin. |
| user\_id | INT | FOREIGN KEY (references Users) | Links to the corresponding user. |
| department\_id | INT | FOREIGN KEY (references Departments) | Department administered by the admin. |
| Other admin-specific information. |  |  |  |

#### **HR Table**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| hr\_id | INT | PRIMARY KEY, AUTO\_INCREMENT | Unique identifier for each HR personnel. |
| user\_id | INT | FOREIGN KEY (references Users) | Links to the corresponding user. |
| department\_id | INT | FOREIGN KEY (references Departments) | Department managed by the HR personnel. |
| Other HR-specific information. |  |  |  |

#### **Departments Table (for reference)**

| **Column Name** | **Data Type** |  | **Constraints** | **Description** |
| --- | --- | --- | --- | --- |
| department\_id | INT |  | PRIMARY KEY, AUTO\_INCREMENT | Unique identifier for each department. |
| department\_name | VARCHAR(100) |  | NOT NULL | Name of the department. |
| head\_of\_department | INT |  | FOREIGN KEY (references Admins) | Admin who heads the department. |
| Other department-specific information. |  |  |  |  |

#### **Events Table**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| event\_id | INT | PRIMARY KEY, AUTO\_INCREMENT | Unique identifier for each event. |
| event\_name | VARCHAR(100) | NOT NULL | Name of the event. |
| description | TEXT |  | Description of the event. |
| event\_date | DATE | NOT NULL | Date of the event. |
| location | VARCHAR(100) | NOT NULL | Location of the event. |
| organizer\_id | INT | FOREIGN KEY (references Users) | User ID of the admin organizing the event. |
| Other event-related information as needed. |  |  |  |

#### **Registrations Table**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| registration\_id | INT | PRIMARY KEY, AUTO\_INCREMENT | Unique identifier for each registration. |
| event\_id | INT | FOREIGN KEY (references Events) | Event ID of the event being registered for. |
| student\_id | INT | FOREIGN KEY (references Students) | Student ID of the student registering. |
| registration\_date | DATETIME | NOT NULL, DEFAULT CURRENT\_TIMESTAMP | Date and time of registration. |
| Other registration-related information as needed. |  |  |  |

### Example SQL Queries

Here are simplified examples of queries you might use with this schema:

* **Inserting a new event**:

INSERT INTO Events (event\_name, description, event\_date, location, organizer\_id)

VALUES ('Orientation Day', 'Welcome new students to campus', '2024-09-01', 'Main Auditorium', <admin\_user\_id>);

* **Retrieving events organized by a specific admin**:

SELECT event\_id, event\_name, event\_date, location

FROM Events

WHERE organizer\_id = <admin\_user\_id>;

* **Registering a student for an event**:

INSERT INTO Registrations (event\_id, student\_id, registration\_date)

VALUES (<event\_id>, <student\_id>, NOW());

* **Listing students registered for a specific event**:

SELECT Users.full\_name, Events.event\_name, Registrations.registration\_date

FROM Registrations

JOIN Students ON Registrations.student\_id = Students.student\_id

JOIN Users ON Students.user\_id = Users.user\_id

JOIN Events ON Registrations.event\_id = Events.event\_id

WHERE Events.event\_id = <event\_id>;

### Explanation of Columns

1. **user\_id**: This is a unique identifier for each user, typically generated automatically using AUTO\_INCREMENT. It serves as the primary key for the Users table, ensuring each user record can be uniquely identified.
2. **username**: Users use their username along with a password to log into the portal. It must be unique and is often limited in length to ensure compatibility with login systems.
3. **password**: This column stores a hashed version of the user's password. Storing passwords securely as hashes (not plain text) ensures security even if the database is compromised.
4. **role**: Indicates the role of the user within the educational institute's online portal. Examples include Student, Admin, HR, Employee, etc. This column helps in distinguishing different levels of access and functionalities.
5. **email**: The user's email address, used for communication and verification purposes. It must be unique and valid.
6. **full\_name**: The user's full name, which provides personal identification within the system. This information is often displayed in user interfaces and reports.
7. **is\_approved**: This column is a boolean flag (TRUE/FALSE) that indicates whether a student account has been approved. Defaulted to FALSE, it can be updated by admins during the approval process.
8. **Other relevant user information**: Depending on the specific requirements of the educational institute's online portal, additional columns might include:
   * **Address**: Physical address of the user.
   * **Phone number**: Contact number of the user.
   * **Date of birth**: Date of birth of the user.
   * **Emergency contact**: Contact information for emergencies.
   * **Profile picture URL**: Link to the user's profile picture.
   * And more, based on application needs.