Database Design for Learning Platform Portal

Tables:

Students:

id (INT, PRIMARY KEY): Unique identifier for each student.
name (VARCHAR(255)): Student's full name.
email (VARCHAR(255), UNIQUE): Student's email address (unique).
password (VARCHAR(255)): Secured password for login (hashed and salted).

Classes:

id (INT, PRIMARY KEY): Unique identifier for each class.
name (VARCHAR(255)): Name of the class.
description (TEXT): Description of the class content.
teacher_id (INT, FOREIGN KEY REFERENCES Students(id)): ID of the teacher in charge of the class (foreign key).

Dashboard:

id (INT, PRIMARY KEY): Unique identifier for each dashboard entry.
student_id (INT, FOREIGN KEY REFERENCES Students(id)): ID of the student the dashboard entry belongs to (foreign key).
class_id (INT, FOREIGN KEY REFERENCES Classes(id)): ID of the class the dashboard entry concerns (foreign key).
last_accessed (DATETIME): Timestamp of the last time the student accessed the dashboard for this class.

Tasks:

id (INT, PRIMARY KEY): Unique identifier for each task.
class_id (INT, FOREIGN KEY REFERENCES Classes(id)): ID of the class the task belongs to (foreign key).
title (VARCHAR(255)): Title of the task.
description (TEXT): Description of the task details and instructions.
due_date (DATETIME): Due date and time for the task.
completed (BOOLEAN): Flag indicating whether the task is completed (true) or not (false).

Doubts:

id (INT, PRIMARY KEY): Unique identifier for each doubt.
student_id (INT, FOREIGN KEY REFERENCES Students(id)): ID of the student who raised the doubt (foreign key).
class_id (INT, FOREIGN KEY REFERENCES Classes(id)): ID of the class the doubt relates to (foreign key).
question (TEXT): Text describing the student's doubt or question.
answer (TEXT): Teacher's response or answer to the doubt (optional).
answered (BOOLEAN): Flag indicating whether the doubt has been answered (true) or not (false).

LeaveApplications:

id (INT, PRIMARY KEY): Unique identifier for each leave application.
student_id (INT, FOREIGN KEY REFERENCES Students(id)): ID of the student applying for leave (foreign key).
class_id (INT, FOREIGN KEY REFERENCES Classes(id)): ID of the class the leave applies to (foreign key).
start_date (DATETIME): Start date of the requested leave.
end_date (DATETIME): End date of the requested leave.
reason (TEXT): Reason for requesting leave.
status (ENUM('pending', 'approved', 'rejected')): Status of the leave application (pending, approved, rejected).

Requirements:

id (INT, PRIMARY KEY): Unique identifier for each requirement.
class_id (INT, FOREIGN KEY REFERENCES Classes(id)): ID of the class the requirement belongs to (foreign key).
description (TEXT): Description of the requirement (e.g., software, hardware, resources).

Applications:

id (INT, PRIMARY KEY): Unique identifier for each application.
student_id (INT, FOREIGN KEY REFERENCES Students(id)): ID of the student applying (foreign key).
class_id (INT, FOREIGN KEY REFERENCES Classes(id)): ID of the class the application pertains to (foreign key).
application_date (DATETIME): Date and time the application was submitted.
application_text (TEXT): Text explaining the reason for applying (optional).
status (ENUM('pending', 'accepted', 'rejected')): Status of the application (pending, accepted, rejected).

LeaderBoard:

id (INT, PRIMARY KEY): Unique identifier for each leaderboard entry.
**student_id (INT)

Students:

```
CREATE TABLE Students (
  id INT PRIMARY KEY AUTO_INCREMENT,
  name VARCHAR(255) NOT NULL,
  email VARCHAR(255) UNIQUE NOT NULL,
  password VARCHAR(255) NOT NULL
);
```

Classes:

```
CREATE TABLE Classes (
  id INT PRIMARY KEY AUTO_INCREMENT,
  name VARCHAR(255) NOT NULL,
  description TEXT,
  teacher_id INT FOREIGN KEY REFERENCES Students(id)
);
```

Dashboard:

```
CREATE TABLE Dashboard (
  id INT PRIMARY KEY AUTO_INCREMENT,
  student_id INT FOREIGN KEY REFERENCES Students(id),
  class_id INT FOREIGN KEY REFERENCES Classes(id),
  last_accessed DATETIME
);
```

Tasks:

```
CREATE TABLE Tasks (
  id INT PRIMARY KEY AUTO_INCREMENT,
  class_id INT FOREIGN KEY REFERENCES Classes(id),
  title VARCHAR(255) NOT NULL,
  description TEXT,
  due_date DATETIME,
  completed BOOLEAN DEFAULT FALSE
);
```

Doubts:

```
CREATE TABLE Doubts (
  id INT PRIMARY KEY AUTO_INCREMENT,
  student_id INT FOREIGN KEY REFERENCES Students(id),
  class_id INT FOREIGN KEY REFERENCES Classes(id),
  question TEXT NOT NULL,
  answer TEXT,
  answered BOOLEAN DEFAULT FALSE
);
```

Leave Application:

```
CREATE TABLE LeaveApplications (
  id INT PRIMARY KEY AUTO_INCREMENT,
  student_id INT FOREIGN KEY REFERENCES Students(id),
  class_id INT FOREIGN KEY REFERENCES Classes(id),
  start_date DATETIME,
  end_date DATETIME,
  reason TEXT,
  status ENUM('pending', 'approved', 'rejected') DEFAULT 'pending'
);
```

Requirement:

```
CREATE TABLE Requirements (
  id INT PRIMARY KEY AUTO_INCREMENT,
  class_id INT FOREIGN KEY REFERENCES Classes(id),
  description TEXT NOT NULL
);
```

Applications:

```
CREATE TABLE Applications (
  id INT PRIMARY KEY AUTO_INCREMENT,
  student_id INT FOREIGN KEY REFERENCES Students(id),
  class_id INT FOREIGN KEY REFERENCES Classes(id),
  application_date DATETIME,
  application_text TEXT,
  status ENUM('pending', 'accepted', 'rejected') DEFAULT 'pending'
);
```
