## Report on Database Entity-Relationship Diagram for CUETSphere

### Introduction

This report details the entities, attributes, and relationships depicted in the provided Entity-Relationship (ER) diagram for "CUETSphere." The ER diagram illustrates the structure of a database designed to manage various academic and communication aspects within a university environment. This report systematically describes each entity and its associated attributes, identifying primary keys (PK). Furthermore, it categorizes the relationships between entities as one-to-one (1:1), one-to-many (1:M), or many-to-many (M:M), based on the cardinality notation in the diagram.

### 1. Departments Entity

* **Attributes:**
  + dept\_id (PK)
  + dept\_name
* **Relationships:**
  + One-to-Many (1:M) with Courses through "offers": A Department offers many Courses.
  + One-to-Many (1:M) with Semesters through "contains": A Department contains many Semesters.

### 2. Semesters Entity

* **Attributes:**
  + semester\_id (PK)
  + semester\_name
* **Relationships:**
  + One-to-Many (1:M) with Courses through "belongs to": A Semester can have many Courses belonging to it.

### 3. Users Entity

* **Attributes:**
  + student\_id (PK)
  + u\_full\_name
  + u\_email
  + u\_password
  + u\_hall
  + u\_role
  + u\_profile\_img\_url
  + u\_bio
  + u\_is\_active
  + u\_created\_at
  + u\_updated\_at
* **Relationships:**
  + One-to-Many (1:M) with Resources through "uploads": A User uploads many Resources.
  + One-to-Many (1:M) with Notices through "sends": A User sends many Notices.
  + One-to-Many (1:M) with Notifications through "recieves": A User receives many Notifications.
  + One-to-Many (1:M) with Posts through "Creates": A User creates many Posts.
  + One-to-Many (1:M) with Comments through "makes": A User makes many Comments.
  + One-to-Many (1:M) with upvotes\_downvotes through "gives": A User gives many upvotes/downvotes.

### 4. Courses Entity

* **Attributes:**
  + course\_id (PK)
  + course\_code
  + course\_name
* **Relationships:**
  + Many-to-Many (M:M) with Semesters through "belongs to" and "contains": Courses belong to Semesters, and Semesters contain Courses.
  + One-to-Many (1:M) with Resources through "has": A Course has many Resources.

### 5. Resources Entity

* **Attributes:**
  + resource\_id (PK)
  + r\_batch
  + r\_resource\_type
  + r\_title
  + r\_file\_path
  + r\_created\_at
* **Relationships:**
  + Many-to-Many (M:M) with Users through "uploads": Resources are uploaded by Users, and Users upload Resources.
  + Many-to-Many (M:M) with Courses through "has": Resources are part of Courses, and Courses have Resources.

### 6. Posts Entity

* **Attributes:**
  + post\_id (PK)
  + p\_created\_at
  + p\_updated\_at
  + p\_attachment
* **Relationships:**
  + One-to-Many (1:M) with Comments through "has comment": A Post has many Comments.
  + One-to-Many (1:M) with post\_tag through "has\_tags": A Post can have many Post Tags.
  + Many-to-Many (M:M) with Users through "Creates": Posts are created by Users, and Users create Posts.

### 7. post\_tag Entity

* **Attributes:**
  + tag\_id (PK)
  + tag\_name{}
* **Relationships:**
  + Many-to-Many (M:M) with Posts through "has\_tags": Post Tags are associated with Posts, and Posts have Post Tags.

### 8. Comments Entity

* **Attributes:**
  + c\_text
  + c\_created\_at
* **Relationships:**
  + One-to-Many (1:M) with Replies through "has\_replies": A Comment has many Replies.
  + Many-to-Many (M:M) with Users through "makes": Comments are made by Users, and Users make Comments.
  + Many-to-Many (M:M) with Posts through "has comment": Comments are associated with Posts, and Posts have Comments.

### 9. Replies Entity

* **Attributes:**
  + par\_cmnt\_id
  + rep\_cmnt\_id
  + rep\_txt
* **Relationships:**
  + Many-to-Many (M:M) with Comments through "has\_replies": Replies are associated with Comments, and Comments have Replies.

### 10. upvotes\_downvotes Entity

* **Attributes:**
  + v\_vote\_type
  + v\_created\_at
* **Relationships:**
  + Many-to-Many (M:M) with Users through "gives": Upvotes/Downvotes are given by Users, and Users give Upvotes/Downvotes.
  + One-to-Many (1:M) with Posts through "has vote": A Post can have many Upvotes/Downvotes.

### 11. Notifications Entity

* **Attributes:**
  + notification\_id (PK)
  + NT\_notification\_type
  + NT\_notification\_msg
  + NT\_created\_at
* **Relationships:**
  + Many-to-Many (M:M) with Users through "recieves": Notifications are received by Users, and Users receive Notifications.

### 12. notices Entity

* **Attributes:**
  + notice\_id (PK)
  + N\_department
  + N\_batch
  + N\_message
  + N\_attachment
  + N\_notice\_type
  + N\_created\_at
  + N\_updated\_at
* **Relationships:**
  + Many-to-Many (M:M) with Users through "sends": Notices are sent by Users, and Users send Notices.