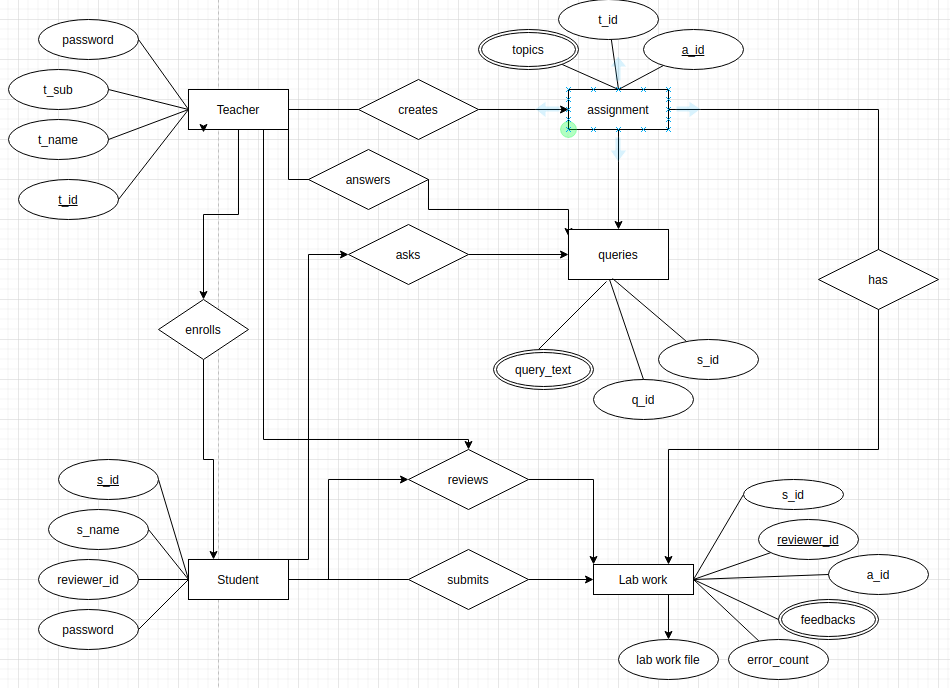
**Database Design and Implementation Phase 1**

**Project: ReviewerForLabWork**

**Entity Relationship Diagram:**

****

**Data Dictionary:**

**Teacher:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field name** | **Datatype** | **Not null** | **size** | **PK** | **FK** | **Description** |
| t\_id | int | 1 | 5 | 1 |  | Teacher id |
| t\_name | varchar | 1 | 40 |  |  | Teacher name |
| t\_sub | varchar | 1 | 15 |  |  | Subject of teaching |
| password | password | 1 | 15 |  |  | Password for login |

**Student:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field name** | **Datatype** | **Not null** | **size** | **PK** | **FK** | **Description** |
| s\_id | int | 1 | 5 | 1 |  | Student id |
| s\_name | varchar | 1 | 40 |  |  | Student name |
| reviewer\_id | int | 1 | 5 | 1 |  | Reviewer id |
| password | password | 1 | 15 |  |  | Password for login |

**Assignment:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field name** | **Datatype** | **Not null** | **size** | **PK** | **FK** | **Description** |
| t\_id | int | 1 | 5 |  | 1 | Teacher id |
| a\_id | int | 1 | 5 | 1 |  | Asignment id |
| topics | varchar | 1 | 15 |  |  | Topics of assignment |

**Lab work:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field name** | **Datatype** | **Not null** | **size** | **PK** | **FK** | **Description** |
| s\_id | int | 1 | 5 |  | 1 | Student id |
| Error count | int | 1 | 5 |  |  | Number of error found by reviewer in lab work |
| a\_id | int | 1 | 5 |  | 1 | Assignment id |
| reviewer\_id | int | 1 | 5 | 1 |  | Reviewer id |
| feedback | varchar | 1 | 15 |  |  | Feedback given by reviewer |
| Lab work file | file | 1 |  |  |  | File upload to submit lab work |

**Queries:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field name** | **Datatype** | **Not null** | **size** | **PK** | **FK** | **Description** |
| s\_id | int | 1 | 5 |  | 1 | student id |
| a\_id | int | 1 | 5 |  | 1 | Assignment id |
| q\_id | int | 1 | 5 | 1 |  | Query id |
| Query text | varchar | 1 | 15 |  |  | Queries of assignment |

**MongoDB**

MongoDB is a cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

MongoDB database stores its data in collections. A collection holds one or more documents, which corresponds to a record or a row in a relational database table, and each document has one or more fields, which corresponds to a column in a relational database table.

**Why MongoDB?**

- Data models are flexible and scalable

- NoSQL queries are faster

- Easy to use

- Supports non-structured data as it is required to store elements of html code.

**DDL for MongoDB in Node.js:**

1. Create Database:

var MongoClient = require('mongodb').MongoClient;

var url = "mongodb://localhost:27017/reviewerdb";

*// reviewerdb is the name of db*

MongoClient.connect(url, function(err, db) {

if (err) throw err;

console.log("Database created!");

db.close();

});

1. Create Collection:

mycol = mydb[*collectionName*]

1. Drop Collection:

mycol.drop()

1. Truncate Collection:

mycol.delete\_many( { } )

**DQL for MongoDB in Node.js:**

1. Query all document

mycol.find():

1. Find a document:

myquery = { "category": "sports" }

mydoc = mycol.find(myquery)

**DML for MongoDB in Node.js:**

1. Insert an assignment:

mydoc = { "assignment": "assignment1", "topics":"c++, programming" "file" : ""}

id = mycol.insert\_one(mydoc)

1. Update an assignment:

myquery = { "topic": "sdp" }

newvalues = { "$set": { "topic": "oose" } }

mycol.update\_one(myquery, newvalues)

1. Delete a document:

myquery = { "assignment": "assignment" }

mycol.delete\_one(myquery)