

# **EXAM REGISTRATION SYSTEM**

**18CSC209J - Database Management System and Cloud  
Integration Services**

**Mini Project Report**

*Submitted by*

**OBUL REDDY. DUTTHALA**  
**[Reg. No.: RA2111028010137]**  
B.Tech. CSE - CLOUD COMPUTING

**SYED SAYEED**  
**[Reg. No.: RA2111028010135]**  
B.Tech. CSE - CLOUD COMPUTING



**DEPARTMENT OF NETWORKING AND COMMUNICATIONS  
SCHOOL OF COMPUTING  
COLLEGE OF ENGINEERING AND TECHNOLOGY  
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**  
**(Under Section 3 of UGC Act, 1956)**  
**S.R.M. NAGAR, KATTANKULATHUR – 603 203**  
**KANCHEEPURAM DISTRICT**  
**MAY 2023**

## **TABLE OF CONTENTS**

<b>Chapter No.</b>	<b>Title</b>	<b>Page No.</b>
1.	Abstract	4
2.	Introduction	5
3.	Problem Statement	6
4.	Module Description	6
5.	Use case diagram	7-8
6.	ER diagram	9-10
7.	Database Creation using DDL and DML	10-13
8.	Normalization of Database	14-16
9.	Implementation using Dynamo DB	17-23
10.	Conclusion	24
11	Appendix I - Screenshot	25-52
12	Appendix II - Github Profile	53-54
13	Appendix III - AWS Course Completion certificate	55

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**  
**KATTANKULATHUR – 603 203**

**BONAFIDE**

This is to certify that **18CSC209J - DATABASE MANAGEMENT SYSTEM AND CLOUD INTEGRATION SERVICES LABORATORY Mini Project report** titled "**EXAM REGISTRATION SYSTEM**" is the bonafide work of **OBUL REDDY.DUTTHALA(RA2111028010137)**, **SYED SAYEED (RA21110280100135)** who undertook the task of completing the project within the allotted time.

**SIGNATURE**

**Dr. Angayarkanni**  
Assistant Professor  
Department of Networking  
and Communications  
SRM Institute of Science  
Technology

**SIGNATURE**

**Dr. Annapurani Panaiyappan K**  
Professor and Head  
Department of Networking  
and Communications  
SRM Institute of Science and  
Technology

## **ABSTRACT**

The aim of Exam Registration System is to automate registering student details, maintaining the student detail, retrieving the specific details, when necessary, security from unauthorized access, functionalities such as insert, delete, update, etc, which must be independent of each other are to be made possible, issuing the hall ticket. This system will help in registering student details to issue hall tickets for the exam. The student can enter to register their details only with their valid username and password. The user can print the hall ticket immediately after registering their details. The administrator will issue the hall ticket only if the student has paid the fee. This Project mainly deals with Login form, Student details, Student report and Student database.

## **INTRODUCTION**

Exam Registration System is an interface between the Student and the Exam Controller responsible for the Issue of Hall Ticket. It aims at improving the efficiency in the Issue of Hall ticket and reduces the complexities involved in it to the maximum possible extent.

## **PURPOSE**

If the entire process of 'Issue of Hall ticket' is done in a manual manner, then it would take several days for the hall ticket to reach the student. Because the number of students for hall tickets is increasing every year, an Automated System becomes essential to meet the demand. So, this system uses several programming and database techniques to elucidate the work involved in this process. As this is a matter of National Security, the system has been carefully verified and validated in order to satisfy it.

## **SCOPE**

- The System provides an online interface to the user where they can fill in their personal details and register for the exam. With the issue of hall ticket can use this system to reduce his workload and process the application in a speedy manner.
- Provide a communication platform between the student and the admin.
- Students will come to know their status of application and the date of exam.

## **PROBLEM STATEMENT**

Exam Registration system.is used in the effective dispatch of registration form to all of the students. This system adopts a comprehensive approach to minimize the manual work and schedule resources, time in a cogent manner. The core of the system is to get the online registration form (with details such as name, reg.no etc.,) filled by the student whose testament is verified for its genuineness by the Exam Registration System with respect to the already existing information in the database. This forms the first and foremost step in the processing of exam applications. After the first round of verification done by the system, the information is in turn forwarded to the Exam Controller. The application is then processed manually based on the report given by the system. The system also provides the student with a list of exam dates. The controller will be provided with fee details to display the current status of application to the student, which they can view in their online interface. After all the necessary criteria has been met, the original information is added to the database and the hall ticket is sent to the student.

## **MODULE DESCRIPTION**

**Module-1:** Is the login form in which the user or the admin has to enter the username and password to enter into the exam registration application system.

**Module-2:** Is the form where the student can enter in to register their details and in the same form the admin can enter to manage the student database and to retrieve it.

**Module-3:** Is the form where the student registers their details in order to receive their hall ticket only if they have paid the fee. As soon as the student submits their details the hall ticket will be issued by admin to print if else, the hall ticket will not be issued.

**Module-4:** Is the form where the admin maintains the student database to insert, delete and update. The complete data of the student can otherwise be viewed in grid form.

## USE CASE DIAGRAM

The Exam Registration use cases in our system are:

1. Login
2. View exam details
3. Register
4. Acknowledgement
5. Fee Processing

### **ACTORS INVOLVED:**

1. Student
2. System DB

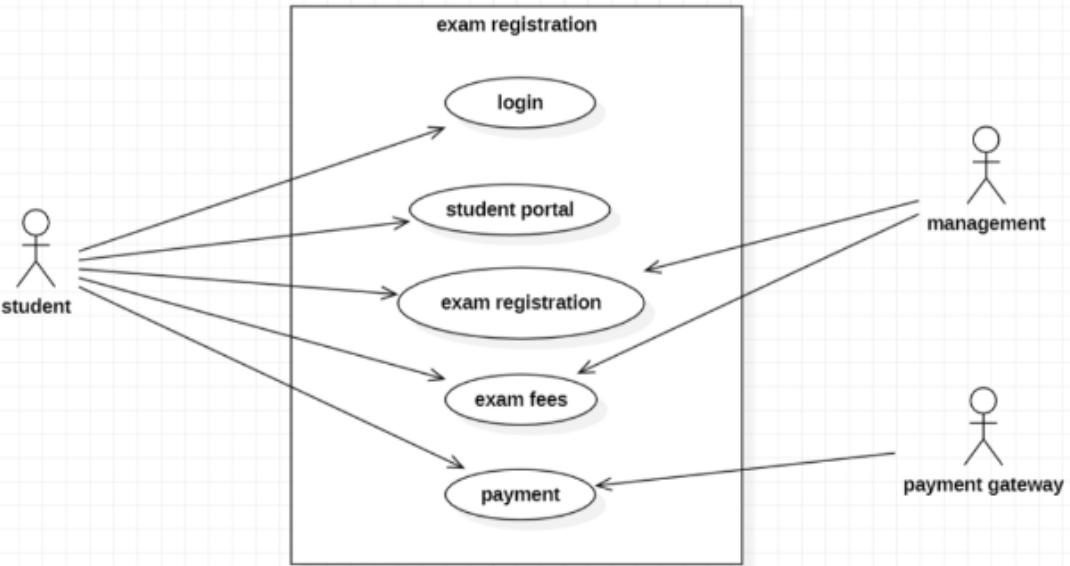
**USE-CASE NAME: LOGIN** -The student enters his username and password to login and retrieve the information'.

**USE-CASE NAME: VIEW EXAM DETAILS** - The student views the details about the exam schedule which contains Date, time, etc...

**USE-CASE NAME: REGISTER** - The student should notify the fee details that only the student can pay the correct amount.

**USE-CASE NAME: ACKNOWLEDGEMENT**- The exam fees should be paid by the student to get the hall ticket from the exam controller.

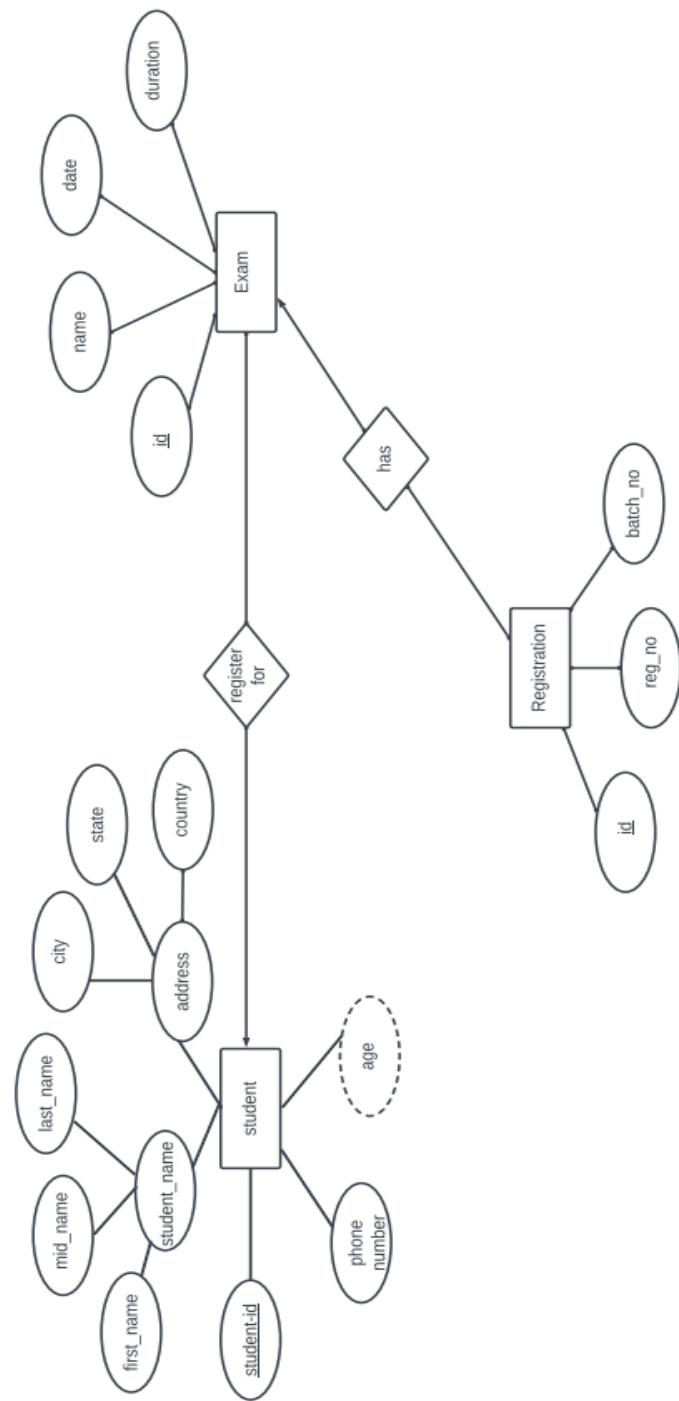
**USE-CASE NAME: FEE PROCESSING** - All the details should be viewed by both the student and the controller to verify whether all the entered details are correct.



USE CASE DIAGRAM FOR EXAM REGISTRATION SYSTEM

## ER DIAGRAM

Entity relationship diagram show the relation between different entities such as the relation between student and exam etc.



**ER DIAGRAM OF EXAM REGISTRATION SYSTEM**

## **Database Creation using DDL and DML**

For the Exam Registration System there are three database tables:  
**Students, Exams and Exam registrations.**

The **students** table stores information about students, the **exams** table stores information about exams, and the **exam registrations** table stores information about which students have registered for which exams and the marks they obtained.

### **USING DDL**

#### **For students**

```
CREATE TABLE students (
    student_id INT PRIMARY KEY,
    student_name VARCHAR(255) NOT NULL,
    student_email VARCHAR(255) NOT NULL,
    ssn VARCHAR(9) UNIQUE,
    student_phone VARCHAR(20) NOT NULL,
    student_address VARCHAR(255) NOT NULL
);
```

#### **For Exams**

```
CREATE TABLE exams (
    exam_id INT PRIMARY KEY,
    exam_name VARCHAR(255) NOT NULL,
    exam_date DATE NOT NULL, exam_time
    TIME NOT NULL, exam_duration INT
    NOT NULL
);
```

#### **For Exam Registrations**

```
CREATE TABLE exam_registrations (
    registration_id INT PRIMARY KEY,
    exam_id INT NOT NULL,
    student_id INT NOT NULL,
```

```
registration_date DATE NOT NULL,  
registration_time TIME NOT NULL,  
FOREIGN KEY (exam_id) REFERENCES exams(exam_id),  
FOREIGN KEY (student_id) REFERENCES students(student_id)  
);
```

#### **For courses**

```
CREATE TABLE courses (  
course_id INT PRIMARY KEY,  
course_name VARCHAR(255) NOT NULL,  
instructor_id INT NOT NULL,  
instructor_name VARCHAR(255) NOT NULL,  
instructor_phone VARCHAR(20) NOT NULL,  
duration VARCHAR(20) NOT NULL  
);
```

### **USING DML**

#### **For Students**

```
INSERT INTO students (student_id, student_name,  
student_email, ssn, student_phone, student_address)  
VALUES (1, 'Sayeed', 'sayeed.m@example.com', 'CS032', '9177133972', 'Vijayawada'),  
(2, 'obul', 'obul.d@example.com', 'CS062', '9676197135', 'Chilakaluripet'),  
(3, 'Mahesh Reddy', 'mahesh.s@example.com', 'CS056', '9765435672', 'Guntur'),  
(4, 'Akhil', 'akhil.j@example.com', 'CS08', '9985450118', 'Guntur');
```

#### **For Exams**

```
INSERT INTO exams (exam_id, exam_name, exam_date, exam_time,  
exam_duration)  
VALUES (121, 'Mathematics', '2023-05-15', '09:00:00', 180),  
(221, 'Physics', '2023-05-17', '09:00:00', 200),  
(321, 'Chemistry', '2023-05-20', '09:00:00', 120),  
(421, 'French', '2023-05-21', '10:00:00', 180);
```

### **For Exam Registrations**

```
INSERT INTO exam_registrations (registration_id, exam_id, student_id,
registration_date, registration_time)

VALUES (1, 121, 1, '2023-04-20', '12:30:00'),

(2, 221, 2, '2023-04-21', '01:00:00'),

(3, 321, 3, '2023-04-22', '12:00:00'),

(4, 421, 4, '2023-04-22', '12:15:00');
```

### **For Course**

```
INSERT INTO courses (course_id, course_name, instructor_id, instructor_name,
instructor_phone, duration)

VALUES (101, 'Introduction to calculus and linear algebra', 1234, 'John Smith', '555-1234',
'60Days'),

(205, 'Introduction to Computer Science', 9112, 'Dr. Alice Brown', '522-6162',
'45Days'),

(303, 'A course in writing and critical thinking', 6534, 'Professor Jane Doe', '644-8769',
'50Days'),

(309, 'A course in Writing and Reading Paragraphs', 7657, 'Professor Ramalingam', '786-
9876', '45Days');
```

## Students

#	student_id	student_name	student_email	ssn	student_phone	student_address
1		Sayeed	sayeed.m@example.com	CS032	9177133972	Vijayawada
2		obul	obul.d@example.com	CS062	9676197135	Chilakaluripet
3		Mahesh Reddy	mahesh.s@example.com	CS056	9765435672	Guntur
4		Akhil	akhil.j@example.com	CS08	9985450118	Guntur

## Exams

#	exam_id	exam_name	exam_date	exam_time	exam_duration
1	121	Mathematics	2023-05-15	09:00:00	180
2	221	Physics	2023-05-17	09:00:00	200
3	321	Chemistry	2023-05-20	09:00:00	120
4	421	French	2023-05-21	10:00:00	180

## Exam Registration

#	registration_id	exam_id	student_id	registration_date	registration_time
1		121	1	2023-04-20	12:30:00
2		221	2	2023-04-21	01:00:00
3		321	3	2023-04-22	12:00:00
4		421	4	2023-04-22	12:15:00

## For Courses

#	course_id	course_name	instructor_id	instructor_name	instructor_phone	duration
1	101	Introduction to calc...	1234	John Smith	555-1234	60Days
2	205	Introduction to Com...	9112	Dr. Alice Brown	522-6162	45Days
3	303	A course in writing ...	6534	Professor Jane Doe	644-8769	50Days
4	309	A course in Writing ...	7657	Professor Ramaling...	786- 9876	45Days

## **NORMALIZATION OF DATABASE**

A database for an exam registration system would typically have multiple tables to store different types of information, such as students, courses, exam schedules, and registrations. To ensure efficient data management and eliminate data redundancies, normalization of the database is required. The normalization process involves breaking down a table into multiple smaller tables, with each table containing a single type of information. The goal is to minimize data redundancies and inconsistencies and ensure data integrity.

The normalization process typically involves the following steps:

**First Normal Form (1NF):** Each table should have a primary key, and each column should contain atomic values. This means that each column should hold a single value, rather than a list or set of values.

**Second Normal Form (2NF):** Each non-key column should be fully dependent on the primary key. This means that there should be no partial dependencies, where a non-key column depends on only a portion of the primary key.

**Third Normal Form (3NF):** Each non-key column should be dependent only on the primary key, and not on any other non-key columns. This means that there should be no transitive dependencies, where a non-key column depends on another non-key column.

For an exam registration system, the following tables may be required:

Student Table: Contains information about the students, such as name, student ID, and contact details.

Course Table: Contains information about the courses offered, such as course code, course name, and course instructor.

Exam Table: Contains information about the exam schedule, such as exam date, exam time, and exam location.

Registration Table: Contains information about the registration, such as registration ID, student ID and exam date.

Student ID	First Name	SSN	Course	Duration	Instructor	Contact
1	sayeed(S1)	CS032	Maths	60 Days	John Smith	555-1234
2	obul(S2)	CS062	Science	45 Days	Dr. Alice Brown	522-6162
3	Mahesh(S3)	CS056	English	50 Days	Professor Jane Doe	644-8769
4	Akhil(S4)	CS08	French	45 Days	Professor Ramalingam	786-9876

**Redundant Data Is Duplicate Data - Course Details And Instructor Details**

Student ID	First Name	SSN	Course ID
1	S1	CS032	101
2	S2	CS062	205
3	S3	CS056	303
4	S4	CS08	309

**STUDENT TABLE**

COURSE TABLE	Course ID	Course	Duration	Instructor ID
	101	Maths	60 Days	1234
	205	Science	45 Days	9112
	303	English	50 Days	6534
	309	French	45 Days	7657

Instructor ID	First Name	Contact
1234	John	555-1234
9112	Alice	522-6162
6534	Jane	644-8769
7657	Ramalingam	786-9876

**INSTRUCTOR TABLE**

**Student ID Is Primary Key For Student Table**

**Course ID Is Primary Key For Course Table And Foreign Key In Student Table**

**Instructor ID Is Primary Key For Instructor Table And Foreign Key In Course Table**

## **1.Student**

- As the attributes of this table does not have sub attributes, it is in first normal form.
- Because every non-primary key attribute is fully functionally dependent on the primary key of the table and it is already in first normal form, this table is now in second normal form.
- Since the table is in second normal form and no nonprimary key attribute is transitively dependent on the primary key, the table is now in 3NF.
- Since the table is in 3NF and no prime key is dependent on non-prime key, so the table is in BCNF.

$\{\text{student\_id}\} \rightarrow \{\text{first name}\}$  (functional dependency exists)

$\{\text{student\_id}\} \rightarrow \{\text{SSN}\}$  (functional dependency exists)

$\{\text{student\_id}\} \rightarrow \{\text{Course id}\}$  (functional dependency exists)

## **2.Course**

- As the attributes of this table does not have sub attributes, it is in first normal form.
- Because every non-primary key attribute is fully functionally dependent on the primary key of the table and it is already in first normal form, this table is now in second normal form.
- Since the table is in second normal form and no nonprimary key attribute is transitively dependent on the primary key, the table is now in 3NF.
- Since the table is in 3NF and no prime key is dependent on non-prime key, so the table is in BCNF.

$\{\text{course\_id}\} \rightarrow \{\text{Course}\}$  (functional dependency exists)

$\{\text{course\_id}\} \rightarrow \{\text{Duration}\}$  (functional dependency exists)

$\{\text{course\_id}\} \rightarrow \{\text{Instructor\_ID}\}$  (functional dependency exists)

## **3.Instructor**

- As the attributes of this table does not have sub attributes, it is in first normal form.
- Because every non-primary key attribute is fully functionally dependent on the primary key of the table and it is already in first normal form, this table is now in second normal form.
- Since the table is in second normal form and no nonprimary key attribute is transitively dependent on the primary key, the table is now in 3NF.
- Since the table is in 3NF and no prime key is dependent on non-prime key, the table is in BCNF.

$\{\text{Instructor\_id}\} \rightarrow \{\text{First Name}\}$  (functional dependency exists)

$\{\text{Instructor\_id}\} \rightarrow \{\text{Contact}\}$  (functional dependency exists)

## **IMPLEMENTATION USING DYNAMO DB**

To implement an exam registration system using DynamoDB, you can follow the following steps:

1. **Define the data model:** Determine the data you need to store for the exam registration system. For example, you may need to store information about students, exams, registrations, and results.
2. **Create a DynamoDB table:** Create a table in DynamoDB to store the data. Define the primary key for the table, which could be a single partition key or a combination of a partition key and a sort key.
3. **Define the table schema:** Define the attributes that you want to store in the table. You can also specify any secondary indexes that you want to create.

Here Is The Implement using Dynamo DB:

### **1.Data model:**

- **Students**

- ✓ ID
- ✓ Name
- ✓ Email

- **Exams**

- ✓ ID
- ✓ Name
- ✓ Date

- **Registrations**

- ✓ ID
- ✓ Student ID
- ✓ Exam ID

- ✓ Date

- **Results**

- ✓ ID
- ✓ Student ID
- ✓ Exam ID
- ✓ Score

### **2.DynamoDB table:**

- Table name: ExamRegistration
- Primary key: Partition key - ID (string)

### 3. Table schema: Attributes

ID (string)

- ✓ Name (string)
- ✓ Email (string)
- ✓ Exam ID (string)
- ✓ Exam Name (string)
- ✓ Exam Date (string)
- ✓ Registration Date (string)
- ✓ Score (number)

#### Secondary indexes:

- GSI1: Partition key - Exam ID (string), Sort key - Score (number)

The screenshot shows two views of the AWS DynamoDB console.

**Top View (Tables Overview):**

- Shows a green success message: "The ExamRegistration table was created successfully."
- Shows a table list with one item: ExamRegistration (Active, Exam ID (S) partition key, Score (N) sort key, Off deletion protection, Provisioned with auto scaling (1) read capacity mode).

**Bottom View (ExamRegistration Table Details):**

- Shows the table configuration page for ExamRegistration.
- Under "General information":
  - Partition key: Exam ID (String)
  - Sort key: Score (Number)
  - Capacity mode: Provisioned
  - Table status: Active
- Under "Alarms":
  - No active alarms
  - Point-in-time recovery (PITR) is off
- Under "Items summary":
  - DynamoDB updates the following information approximately every six hours.
  - Get live item count button

[Alt+S]

Oregon ▾ Obulreddy.D

Successfully submitted the request to create your index.

DB > Tables > ExamRegistration

## ExamRegistration

C Actions ▾ Explore table items

Overview Indexes Monitor Global tables Backups Exports and streams Additional settings

### Global secondary indexes (1) Info

Delete Create index

Find indexes

Name	Status	Partition key	Sort key	Read capacity	Write capacity	Project
Exam-id57	Creating	Exam ID (String)	Score (Number)	Range: 1 - 10 Auto scaling at 70% Current provisioned units: 1	Range: 1 - 10 Auto scaling at 70% Current provisioned units: 1	All

[Alt+S]

Oregon ▾ Obulreddy.D ▾

Filters

Run Reset

Completed. Read capacity units consumed: 0.5

### Items returned (3)

C Actions ▾ Create item

Exam ID	Contact	Duration	Email ID	Exam Date	Exam Na...	Name	Re...
221	9676197135	200	obul.d@exa...	2023-05-17	Physics	obul	20...
121	9177133972	180 min	sayeed.m@...	2023-05-15	Mathematics	sayeed	20...
421	9985450118	180 min	akhil.j@exa...	2023-05-21	French	Akhil	20...

```
{  
    "Exam ID": {  
        "S": "121"  
    },  
    "Contact": {  
        "N": "9177133972"  
    },  
    "Duration": {  
        "S": "180 min"  
    },  
    "Email ID": {  
        "S": "sayeed.m@example.com"  
    },  
    "Exam Date": {  
        "S": "2023-05-15"  
    },  
    "Exam Name": {  
        "S": "Mathematics"  
    },  
    "Name": {  
        "S": "sayeed"  
    },  
    "Registration Date": {  
        "S": "2023-04-20\\t12:30:00"  
    }  
}
```

```
{  
    "Exam ID": {  
        "S": "421"  
    },  
    "Contact": {  
        "N": "9985450118"  
    },  
    "Duration": {  
        "S": "180 min"  
    },  
    "Email ID": {  
        "S": "akhil.j@example.com"  
    },  
    "Exam Date": {  
        "S": "2023-05-21"  
    },  
    "Exam Name": {  
        "S": "French"  
    }
```

```
},
  "Name": {
    "S": "Akhil"
  },
  "RegistrationDate": {
    "S": "2023-04-22\\t12:15:00"
  }
}

{
  "ExamID": {
    "S": "221"
  },
  "RegistrationTime": {
    "S": "2023-04-21\\t01:00:00"
  },
  "Contact": {
    "N": "967619713"
  },
  "Duration": {
    "N": "200"
  },
  "Email ID": {
    "S": "obul.d@example.com"
  },
  "Exam Date": {
    "S": "2023-05-17"
  },
  "Exam Name": {
    "S": "Physics"
  },
  "Name": {
    "S": "obul"
  }
}
```

```
{  
    "ExamID": {  
        "S": "321"  
    },  
    "RegistrationTime": {  
        "S": "2023-04-22\12:00:00"  
    },  
    "Contact": {  
        "S": "9765435672"  
    },  
    "Duration": {  
        "N": "120"  
    },  
    "Email ID": {  
        "S": "mahesh.s@example.com"  
    },  
    "Exam Date": { "S":  
        "2023-05-20"  
    },  
    "Exam Name": {  
        "S": "Chemistry"  
    },  
    "Name": {  
        "S": "Mahesh Reddy"  
    }  
}
```

**Create the table by running the following AWS CLI command:**

```
aws dynamodb create-table --cli-input-json file://table_schema.json
```

**To check the status of the table by running the following AWS CLI command:**

```
aws dynamodb describe-table --table-name my_table
```

**To add data:**

```
{"my_partition_key": {"S": "my_partition_key_value"},  
 "my_sort_key": {"N": "1"},  
 "attribute1": {"S": "value1"},  
 "attribute2": {"N": "2"} }
```

**Use the ‘AWS DynamoDB put item’ command to add the item to the table. Run the following command:**

```
AWS DynamoDB put-item --table-name my_table --item file://item.json
```

**Verify that the item has been added by using the aws DynamoDB get-item command. Run the following command:**

```
aws dynamodb get-item --table-name my_table --key '{"my_partition_key": {"S": "my_partition_key_value"}, "my_sort_key": {"N": "1"}}'
```

**You can add more items by repeating the steps above with different data in the ‘item.json file’**

**Scan Operations:**

**Scanning a table for all items:**

This command scans the my\_table table and returns all items in the table  
aws dynamodb scan --table-name my\_table

If the table has many items, you may need to paginate the results to retrieve all items. To do this, add the --page-size parameter to the aws dynamodb scan command and specify the maximum number of items to retrieve per page. For example, to retrieve 100 items per page, run the following command:

```
aws dynamodb scan --table-name my_table --page-size 100
```

**Scanning a table with filter expression:**

First run this command to scan with a filter expression.

```
aws dynamodb scan --table-name my_table --filter-expression "attribute1 = :value" --expression-attribute-values '{":value":{"S": "value1"}}'
```

**Next verify the results by this command**

```
aws dynamodb scan --table-name my_table --filter-expression "attribute1 = :value" --expression-attribute-values '{":value":{"S": "value1"}}' | jq '.Items[].{attribute1:.attribute1.S, attribute2:.attribute2.N}'
```

## **CONCLUSION**

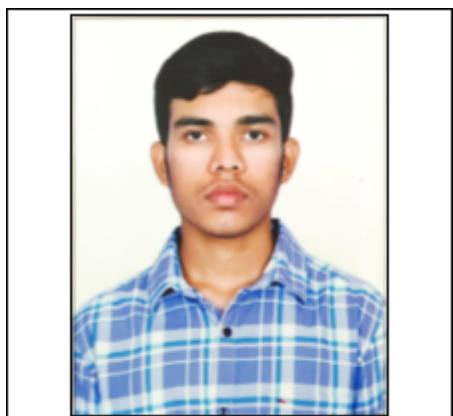
In conclusion, the exam registration system project is an essential tool for managing student registration and exam schedules. It helps to simplify the registration process by providing a user-friendly interface and ensures that students can register for their exams without any complications. The system can handle large amounts of data and store them securely. The system can also generate reports and analyse data to provide useful insights to the management team. Overall, the exam registration system can improve the efficiency of the registration process and enhance the overall student experience.

## Appendix -I

### **DATABASE MANAGEMENT SYSTEMS AND CLOUD INTEGRATION SERVICES - 18CSC209J**

**B.Tech CSE (Cloud Computing)**

#### **Student Portfolio**



Name :Obul Reddy . Dutthala

Reg No :RA2111028010137

Year :2nd year

Section :cse (cc) :q1

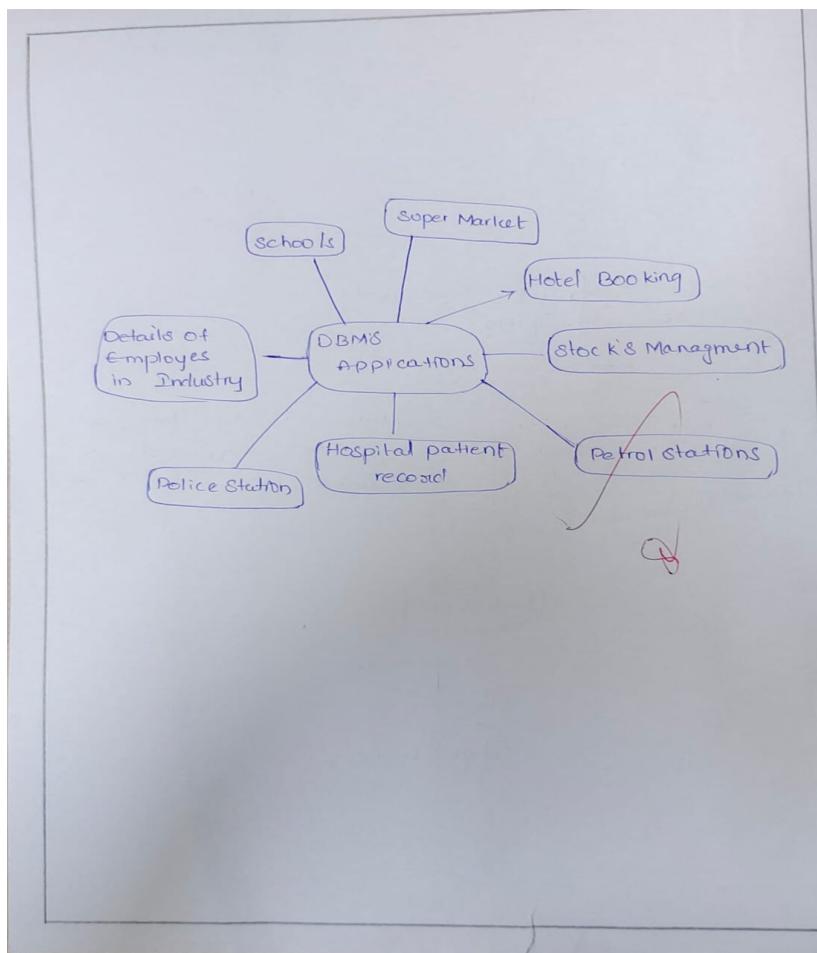
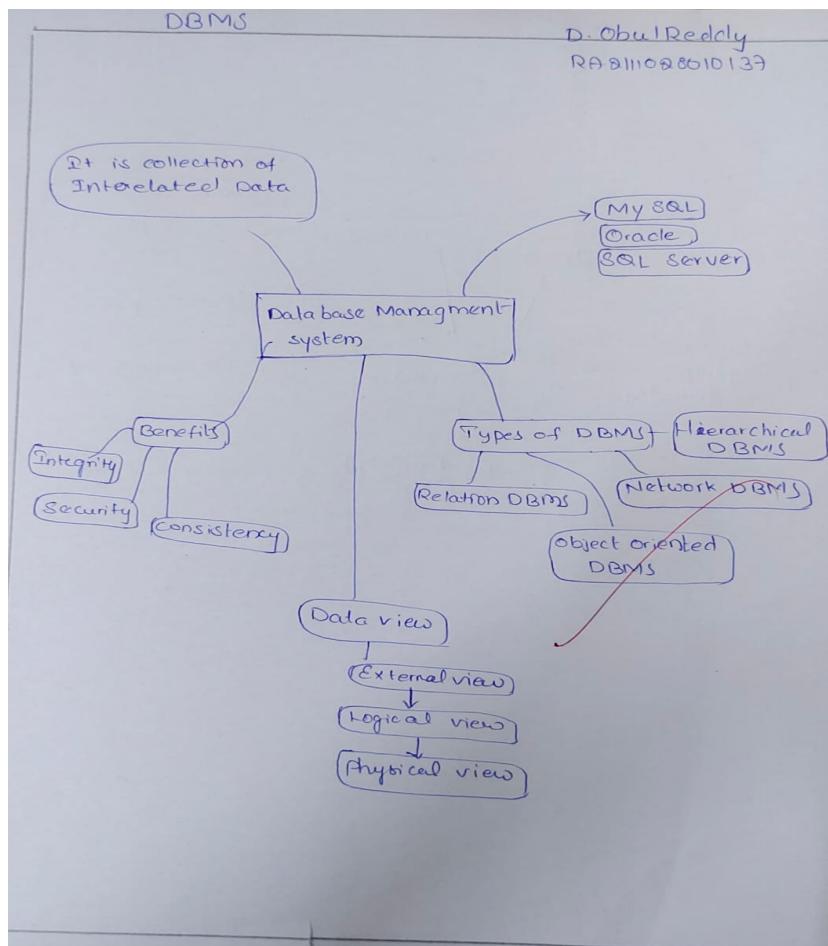
Course Teacher : Dr. S. Angayarkanni

#### **List of Assignments:**

- Mind Map of DBMS
- ER Diagram
- SQL Query formulation
- DBMS Quiz

Signature of the student

- Mind Map of DBMS



- DBMS Quiz

**DBMS-CIS\_Assignment I\_Q1** Total points 0/0 ?

The respondent's email (dv3490@srmist.edu.in) was recorded on submission of this form.

REGISTER NUMBER \*

RA2111028010137

1. Example for Network DBMS (2M) \*

Integrated DBMS

2. State the advantages of DBMS (2M) \*

Data retrieval is fast, security is high,data integrity and redundancy will be there..used to store create modify alter the data

3. Need of SQL (2M) \*

It is an interface between user and data base system

4. Example for Object oriented DBMS (ODBMS) (2M) \*

Network DBMS , hierarchical DBMS

5. Advantages of DBMS over File processing (2) \*

Data retrieval,data redunancy

6. Example for Hierarchical DBMS (2M) \*

Information management system and RDM mode

7. Syntax and Example for CREATE TABLE Command (3M) \*

CREATE TABLE table\_name(colname datatype....)

8. Create a Student database table with 5 columns (8M) \*

```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,
    ...
);
CREATE TABLE Persons (
    PersonID int,
    Linga varchar(255),
    Raju varchar(255),
    Dhoni varchar(255),
    Rahul varchar(255),
    Sayeed varchar(678)
);
```

9. Databases for Mobile Application development \*

MY SQL , SQL SERVER

---

10. Write the syntax of command used to modify the base table structure \*

ALTER TABLE <table\_name>  
ADD/DROP <column\_name> [datatype];  
ALTER TABLE <table>  
MODIFY <column> <new\_definition>;If we want to add a column named Mark in the  
student table . MySql> ALTER TABLE Student ADD Mark int(3);

---

11. Write the difference between DROP and TRUNCATE. \*

DROP command is used to remove the whole database  
TRUNCATE command is used to remove all the rows from the table.

---

Explain the differences between physical level, Logical level and view level of data abstraction. \*

Physical Level: Defines how the data is actually stored employing various data structures.  
Logical level: Describes the relationship which exists among the stored data.  
View level: Provides a high level view of a section of data

---

- DBMS Quiz

## DBMSCIS Assignment - 3

dv3490@srmist.edu.in [Switch account](#)  Draft saved

\* Indicates required question

Email \*

Record dv3490@srmist.edu.in as the email to be included with my response

The query given below will not give an error. Which one of the following has to be \* replaced to get the desired output?

**SELECT** ID, name, dept name, salary \* 1.1  
**WHERE** instructor;

Salary\*1.1  
 ID  
 Where  
 Instructor

In the given query which of the keyword has to be inserted? \*

**INSERT INTO** employee \_\_\_\_ (1002,Joey,2000);

- Table
- Values
- Relation
- Field

Which of the following is not a valid type of user of a dbms? \*

- Database Administrators
- Sophisticated Users
- Naïve Users
- System Administrators

Which is not considered as mobile database? \*

- SQL Anywhere
- SQLite
- SQLBase
- MySQL

**SELECT \* FROM dept** \*

What type of statement is this?

- TCL
- DCL
- DML
- DDL

Database \_\_\_\_\_ which is the logical design of the database, and the database \_\_\_\_\_ which is a snapshot of the data in the database at a given instant in time. \*

- Instance, Schema
- Relation, Schema
- Relation, Domain
- Schema, Instance

**SELECT \* FROM employee WHERE salary>10000 AND dept\_id=101;** \*  
Which of the following fields are displayed as output?

- Salary
- All the field of employee relation
- Employee
- Salary, dept\_id

Which of the following statements contains an error? \*

- Select \* from emp where empid = 10003;
- Select empid from emp;
- Select empid where empid = 1009 and lastname = 'GELLER';
- Select empid from emp where empid = 10006;

**CREATE TABLE** student (name **VARCHAR**, id **INTEGER**) \*

What type of statement is this?

- DML
- DDL
- DCL
- TCL

Which of the following is an example for network database ? \*

- IDS
- Oracle
- MySQL
- Ingress

A \_\_\_\_\_ in a table represents a relationship among a set of values. \*

- Column
- Row
- Key

What is the highest level of abstraction in the DBMS? \*

- Physical level
- Logical level
- View level
- File level

\_\_\_\_\_ is used to give / get back / control the privileges of an object by the owner \*

- GRANT
- REVOKE
- COMMIT
- SAVEPOINT

NAME \*

obul reddy

**SELECT** name \_\_\_\_\_ instructor.name, course.id  
**FROM** instructor, teaches  
**WHERE** instructor.ID= teaches.ID;

\*

Which keyword must be used here to rename the field name?

- From
- As
- Join
- Rename

In SQL the spaces at the end of the string are removed by \_\_\_\_\_ function.\*

- Upper
- String
- Trim
- Lower

The \_\_\_\_\_ clause allows us to select only those rows in the result relation of the \_\_\_\_\_ clause that satisfy a specified predicate.\*

- Where, from
- From, select
- Select, from
- From, where

A domain is atomic if elements of the domain are considered to be \_\_\_\_\_ \*  
units.

- Different
- Indivisible
- Constant
- Divisible

\_\_\_\_\_ require a user to specify what data are needed and how to get those data. \*

- Procedural DMLs
- Non Procedural DMLs
- DDL
- DCL

REGISTER NUMBER \*

RA2111028010137

What are the components of a disk storage? \*

- Data
- Indices
- DML Compiler
- Statistical data
- Query Evaluation Engine

\_\_\_\_\_ acts as a visualization tool to represent a database? \*

- Object model
- ER Model
- Hierarchical model
- View

```
SELECT name  
FROM student  
WHERE dept name = 'NWC'  
ORDER BY name;
```

\*

By default, the order by clause lists items in \_\_\_\_\_ order.

- Same
- Ascending
- Descending
- Any

Updates that violate \_\_\_\_\_ are disallowed. \*

- Integrity constraints
- Transaction control
- Authorization
- DDL constraints

DBA stands for : \*

- Data Bank Account
- Database Access
- Data Bank Administration
- Data Base Administrator

1) Create a table called movie with the attributes movie id, Name, director, Actor, Language, Musician, Year.

Create another table called actor with attributes actor id, Name, age, date of birth, home town. Find the movies casted by Shahrukh Khan. List the movies 2020. Count the no. of movies composed by AR Rahman. List the full details of any one of favorite actor.

Ans CREATE TABLE movie (

```
MOVIE_ID int,
NAME varchar(10),
DIRECTOR varchar(10),
ACTOR varchar(10),
LANGUAGE varchar(10),
MUSICIAN varchar(10),
YEAR int
);
```

CREATE TABLE actor (

```
ACTOR_ID int,
NAME varchar(10),
AGE int,
DATE_OF_BIRTH date,
HOME_TOWN varchar(10)
);
```

Insert into movie ('MIB4', Name of the movie, Director, Actor, Language, musician, year);

values ('100', 'Pushpa', 'Sukumar', 'Allu Arjun', 'Telugu',  
"Devi Sri Prasad", 2020);

Insert into movie ('MovieId', Name of the Movie, Director, Actor, Language, Musician, year)

values ('101', 'Baahubali', 'Rajamouli', 'Prabhas', 'Telugu',  
'Devi Sri Prasad', 2020)

values ('102', "Master", "Lokesh", "Vijay", "Tamil", "Anirudh", '2019');

values ('103', "RRR", "Rajamouli", "Ramcharan", "Telugu", "Keravani", '2022')

Select from \* movie .

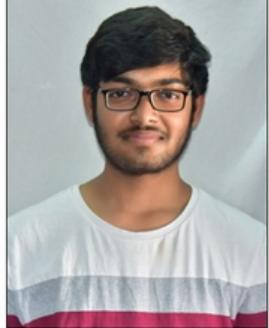
Movie Id	Name	Director	Actor	Language	Music	Year
100	Pushpa	Sukumar	Allu Arjun	Telugu	DSP	2020
101	Bahubali	Rajamouli	Prabhas	Telugu	Kervani	2018
102	Master	Lokesh	Vijay	Tamil	Anirudh	2019
103	RRR	Rajamouli	Ram charan	Telugu	Kervani	2021

## **APPENDIX-1**

### **DATABASE MANAGEMENT SYSTEMS AND CLOUD INTEGRATION SERVICES - 18CSC109J**

**B. Tech CSE (Cloud Computing)**

#### **Student Portfolio**



Name : SYED SAYEED

Reg No : RA2111028010135

Year : 2<sup>ND</sup> YEAR

Section : CSE CC (Q1)

Course Teacher : Dr. S. A. Angayarkanni

#### **List of Activities:**

- Mind Map of DBMS
- ER Diagram
- SQL Query formulation
- DBMS Quiz

SYED SAYEED

Signature of the student

- DBMS QUIZ

**DBMS-CIS\_Assignment I\_Q1** Total points **0/0**

The respondent's email ([ss5238@srmist.edu.in](mailto:ss5238@srmist.edu.in)) was recorded on submission of this form.

**REGISTER NUMBER \***

RA2111028010135

**1. Example for Network DBMS (2M) \***

Integrated data store,idms

**2. State the advantages of DBMS (2M) \***

Data retrieval is fast, security is high,data integrity and redundancy will be there..used to store create modify alter the data

**3. Need of SQL (2M) \***

It acts as interface between database user and database system

**4. Example for Object oriented DBMS (ODBMS) (2M) \***

Network DBMS, hierarchy DBMS,objectivity DB

**5. Advantages of DBMS over File processing (2) \***

Data retrieval is fast, security is high,data integrity and redundancy will be there

**6. Example for Hierarchical DBMS (2M) \***

Information management system, RDM mobile

**7. Syntax and Example for CREATE TABLE Command (3M) \***

CREATE TABLE table\_name(colname datatype...)

CREATE TABLE employee\_details(name varchar(50),Id int);

8. Create a Student database table with 5 columns (8M) \*

```
CREATE TABLE Persons (
    PersonID int,
    LastName varchar(255),
    FirstName varchar(255),
    Address varchar(255),
    Course varchar(255)
);
```

9. Databases for Mobile Application development \*

My SQL,mongoDB,couchbase

10. Write the syntax of command used to modify the base table structure \*

```
ALTER TABLE <table_name>
ADD/DROP <column_name> [datatype];
ALTER TABLE <table>
MODIFY <column> <new_definition>;If we want to add a column named Mark in the student
table . MySql> ALTER TABLE Student ADD Mark int(3);
```

11. Write the difference between DROP and TRUNCATE. \*

Drop means deleting or removing whole data from database  
Truncate means to remove columns and rows in table

11. Write the difference between DROP and TRUNCATE. \*

Drop means deleting or removing whole data from database  
Truncate means to remove columns and rows in table

Explain the differences between physical level, Logical level and view level of data \* abstraction.

There are three levels of abstraction : Physical Level: Defines how the data is actually stored employing various data structures. Logical level: Describes the relationship which exists among the stored data. View level: Provides a high level view of a section of data

# DBMSCIS Assignment - 3

ss5238@srmist.edu.in [Switch account](#)



Draft saved

\* Indicates required question

Email \*

Record ss5238@srmist.edu.in as the email to be included with my response

What are the components of a disk storage? \*

- Data
- Indices
- DML Compiler
- Statistical data
- Query Evaluation Engine

A domain is atomic if elements of the domain are considered to be \_\_\_\_\_ \*  
units.

- Different
- Indivisible
- Constant
- Divisible

Database \_\_\_\_\_ which is the logical design of the database, and the database \*  
\_\_\_\_\_ which is a snapshot of the data in the database at a given instant in time.

- Instance, Schema
- Relation, Schema
- Relation, Domain
- Schema, Instance

Which is not considered as mobile database? \*

- SQL Anywhere
- SQLite
- SQLBase
- MySQL

\_\_\_\_\_ is used to give / get back / control the privileges of an object by the owner \*

- GRANT
- REVOKE
- COMMIT
- SAVEPOINT

```
SELECT name  
FROM student  
WHERE dept name = 'NWC'  
ORDER BY name;
```

\*

By default, the order by clause lists items in \_\_\_\_\_ order.

- Same
- Ascending
- Descending
- Any

The \_\_\_\_\_ clause allows us to select only those rows in the result relation of the \*  
\_\_\_\_\_ clause that satisfy a specified predicate.

- Where, from
- From, select
- Select, from
- From, where

Updates that violate \_\_\_\_\_ are disallowed. \*

- Integrity constraints
- Transaction control
- Authorization
- DDL constraints

What is the highest level of abstraction in the DBMS? \*

- Physical level
- Logical level
- View level
- File level

Which of the following is an example for network database ? \*

- IDS
- Oracle
- MySQL
- Ingress

**CREATE TABLE** student (name **VARCHAR**, id **INTEGER**) \*

What type of statement is this?

- DML
- DDL
- DCL
- TCL

\_\_\_\_\_ acts as a visualization tool to represent a database? \*

- Object model
- ER Model
- Hierarchical model
- View

Which of the following statements contains an error? \*

- Select \* from emp where empid = 10003;
- Select empid from emp;
- Select empid where empid = 1009 and lastname = 'GELLER';
- Select empid from emp where empid = 10006;

Which of the following is not a valid type of user of a dbms? \*

- Database Administrators
- Sophisticated Users
- Naïve Users
- System Administrators

**SELECT \* FROM dept** \*

What type of statement is this?

- TCL
- DCL
- DML
- DDL

In the given query which of the keyword has to be inserted? \*

**INSERT INTO** employee \_\_\_\_ (1002,Joey,2000);

- Table
- Values
- Relation
- Field

The query given below will not give an error. Which one of the following has to be \* replaced to get the desired output?

**SELECT** ID, name, dept name, salary \* 1.1

**WHERE** instructor;

- Salary\*1.1
- ID
- Where
- Instructor

A \_\_\_\_\_ in a table represents a relationship among a set of values. \*

- Column
- Row
- Key
- Entry

DBA stands for : \*

- Data Bank Account
- Database Access
- Data Bank Administration
- Data Base Administrator

\_\_\_\_\_ require a user to specify what data are needed and how to get those data. \*

- Procedural DMLs
- Non Procedural DMLs
- DDL
- DCL

**SELECT \* FROM employee WHERE salary>10000 AND dept\_id=101; \***

Which of the following fields are displayed as output?

- Salary
- All the field of employee relation
- Employee
- Salary, dept\_id

REGISTER NUMBER \*

RA2111028010135

NAME \*

SYED SAYEED

In SQL the spaces at the end of the string are removed by \_\_\_\_\_ function. \*

- Upper
- String
- Trim
- Lower

**SELECT** name \_\_\_\_ instructor.name, course.id  
**FROM** instructor, teaches  
**WHERE** instructor.ID= teaches.ID;

\*

Which keyword must be used here to rename the field name?

- From
- As
- Join
- Rename

Which one of the following is used to define the structure of the relation, deleting \* relations and relating schemas?

- DML(Data Manipulation Language)
- DDL(Data Definition Language)
- Query
- Relational Schema

1) Create a table called movie with the attributes movie id, Name, director, language, musician, year.

Create another table called actors with attributes actor id, name, age, date of birth, home town. Find movies casted by Shahrukh Khan list movies 2020 . count the no. of movies composed by AR Rahman. List the full details of any one of favorite actors.

Ans CREATE TABLE movie(

```
movie ID int,
NAME varchar(10),
DIRECTOR varchar(10),
ACTOR varchar(10),
LANGUAGE varchar(10),
MUSICIAN varchar(10),
YEAR int
);
```

CREATE TABLE actors(

```
ACTOR ID int,
NAME varchar(10),
AGE int,
DATE OF BIRTH date,
Home town varchar(10));
```

Insert into movie ("MID", Name of the movie, Director  
Actor , Language, musician, year);

```
values ('100', "pushpa", "Sukumar", "Allu Arjun",
"Telugu", "DSP", 2020);
```

Insert into movie ("MovieID", "Name of the movie", Director, Actor, language, Musician, Year)

values ('101', "Pathaan", "Ranbir", SRK, "Hindi", "Telugu", "Keeravani", '2022');

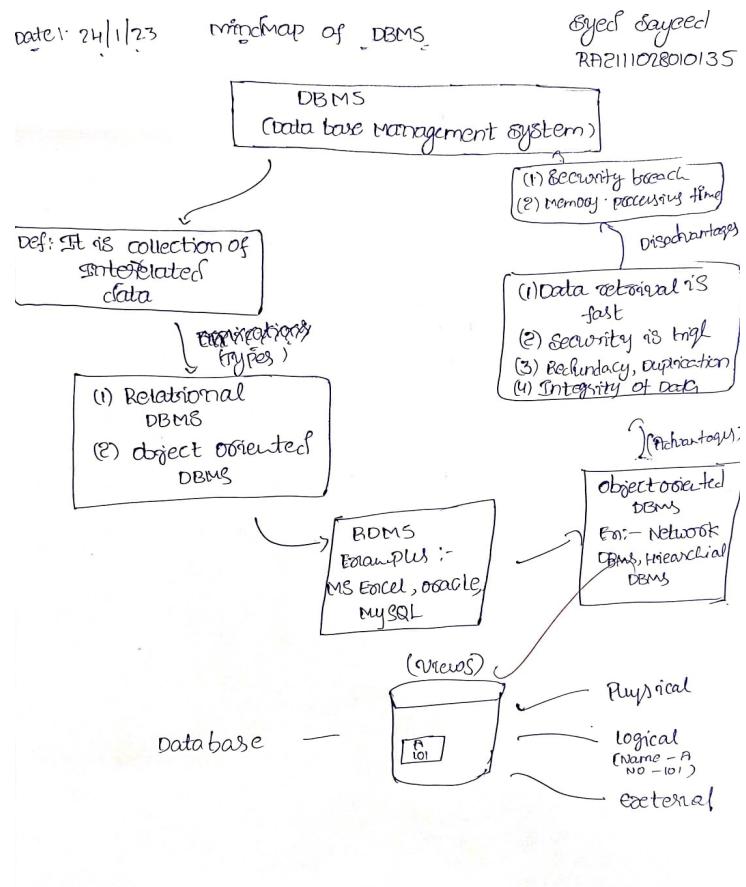
values ('102', "PS-I", "Mani Ratnam", "Lokesh", "Kaithi", "Tamil", "A.R.Rahman", '2022');

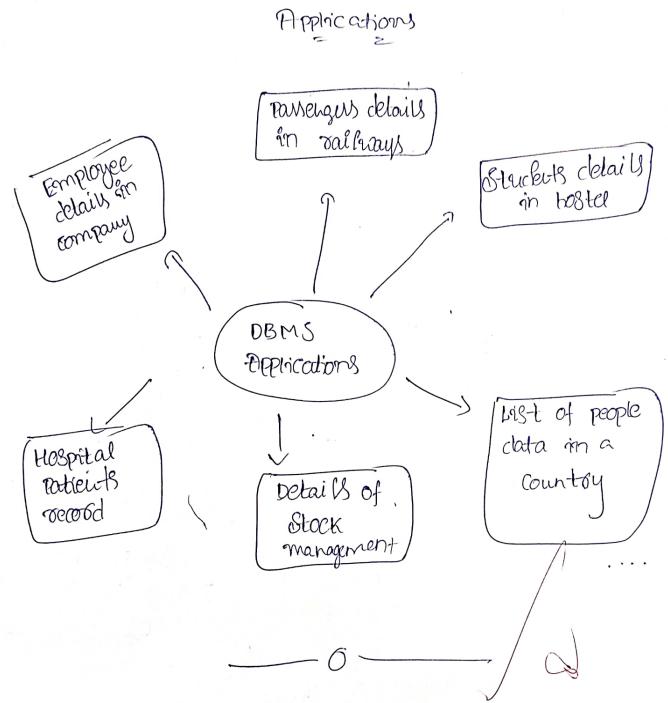
values ('103', "RRR", "Rajamouli", "Ramcharan", "Telugu", "Keeravani", '2022');

Select \* from movie

MovieID	Name	Director	Actor	Language	Music	Year
100	Pushpa	Sukumar	Allu Arjun	Telugu	DSP	2020
101	Pathaan	Ranbir	SRK	Hindi	Keeravani	2022
102	PS-I	Mani Ratnam	Koonthi	Tamil	A.R.Rahman	2022
103	RRR	Rajamouli	Ramcharan	Telugu	Keeravani	2022

## ● MINDMAP OF DBMS





## Appendix II - GitHub Profile

<https://github.com/obulreddy137>

The screenshot shows the GitHub repository page for 'DBMS-assignments'. The repository is public. On the right side, there is a dropdown menu for the user 'obulreddy137' with options like 'Your profile', 'Your repositories', and 'Sign out'. Below the dropdown, there are links for 'Upgrade', 'Try Enterprise', 'Feature preview', 'Help', 'Settings', and 'Sign out'. The main area shows a large text input field with a placeholder 'Drag additional files here to add them to your repository' and a link 'Or choose your files'. Below this, there is a list of four files: 'e1.sql', 'e3.sql', 'e4.sql', and 'e9.sql', each with a delete icon.

The screenshot shows the GitHub user profile page for 'obulreddy137'. The profile picture is a stylized pink cross. The user has 0 followers and is following 1 person. There are tabs for 'Overview', 'Repositories' (6), 'Projects', 'Packages', and 'Stars'. The 'Popular repositories' section lists the following repositories:

- obulreddy137 (Public): Config files for my GitHub profile.
- solid-engine (Public): A repository with a small icon.
- sturdy-train (Public): A repository with a small icon.
- friendly-octo-spoon (Public): A repository with a small icon.
- food-ordering-in-restaurant (Public): A repository with a small icon.
- DBMS-assignments (Public): The repository shown in the first screenshot.

At the bottom, it says '6 contributions in the last year' and 'Contribution settings ▾'.

<https://github.com/sayeed135>

The screenshot shows the GitHub profile of the user 'sayeed135'. At the top, there's a circular profile picture with a green geometric pattern. Below it, the username 'sayeed135' is displayed. A 'Edit profile' button is located just below the profile picture. The main area lists three repositories:

- SAYEED135-EXAM-REGISTRATION** (Public) - Updated 2 minutes ago.
- DBMS-PGRMS** (Public) - Updated 26 minutes ago.
- mini-project-employee-details** (Public) - Updated on Jul 4, 2022.

At the top of the page, there are navigation tabs: Overview, Repositories (8), Projects, Packages, Stars, and a search bar labeled 'Find a repository...'. There are also buttons for Type, Language, Sort, and a 'New' button.

The screenshot shows the repository page for 'DBMS-PGRMS' owned by 'sayeed135'. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The 'Code' tab is selected.

Key statistics shown are: 1 branch, 0 tags, 2 commits, and 0 forks. The commit history is listed as follows:

Author	Message	Time	Commits
sayeed135	Add files via upload	6bc9362 29 minutes ago	2 commits
	README.md	Initial commit	48 minutes ago
	e1.sql	Add files via upload	29 minutes ago
	e10.sql	Add files via upload	29 minutes ago
	e11.sql	Add files via upload	29 minutes ago
	e12.sql	Add files via upload	29 minutes ago
	e2.sql	Add files via upload	29 minutes ago
	e3.sql	Add files via upload	29 minutes ago
	e4.sql	Add files via upload	29 minutes ago
	e5.sql.pdf	Add files via upload	29 minutes ago
	e6.sql.txt	Add files via upload	29 minutes ago
	e7.sql	Add files via upload	29 minutes ago
	e8.sql	Add files via upload	29 minutes ago
	e9.sql	Add files via upload	29 minutes ago

On the right side, there are sections for About (No description, website, or topics provided), Releases (No releases published, Create a new release), and Packages (No packages published, Publish your first package). There are also buttons for Pin, Unwatch (1), Fork (0), and Star (0).

### Appendix III - AWS Course Completion certificate





