# **CHAPTER 1**

**INTRODUCTION**

Database is a collection of data and Management System is a set of programs to store and retrieve those data. Based on this one can define DBMS as a collection of inter-related data and set of programs to store & access those data in an easy and effective manner.

1. **What is the need of DBMS?**

Database systems are basically developed for large amount of data. When dealing with huge amount of data, there are two things that require optimization: Storage of data and retrieval of data**.**

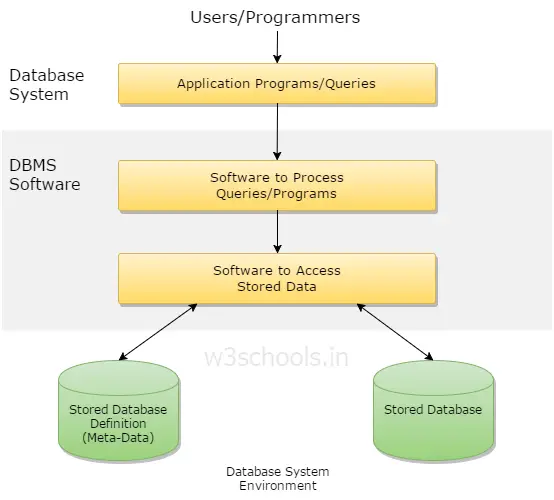
**Storage**: According to the principles of database systems, the data is stored in such a way that it acquires lot less space as the redundant data (duplicate data) has been removed before storage. Let’s take a layman example to understand this. In a banking system, suppose a customer is having two accounts, one is saving account and another is salary account. Let’s say bank stores saving account data at one place (these places are called tables we will learn them later) and salary account data at another place, in that case if the customer information such as customer name, address etc. are stored at both places then this is just a wastage of storage (redundancy/ duplication of data), to organize the data in a better way the information should be stored at one place and both the accounts should be linked to that information somehow. The same thing we achieve in DBMS.

**Fast Retrieval of data:** Along with storing the data in an optimized and systematic manner, it is also important that we retrieve the data quickly when needed. Database systems ensure that the data is retrieved as quickly as possible.

The choice of a database product is often influenced by factors such as:

1. the computing platform (i.e., hardware, operating system)
2. the volume of data to be managed
3. the number of transactions required per second
4. existing applications or interfaces that an organization may have
5. support for heterogeneous and/or distributed computing
6. cost
7. vendor support
8. **Design and Modeling:**

The first task of a database designer is to produce a conceptual datamodel that reflects the structure of the information to be held in the database. A common approach to this is to develop an entity-relationship model, often with the aid of drawing tools. Another popular approach is the Unified Modeling Language. A successful data model will accurately reflect the possible state of the external world being modeled: for example, if people can have more than one phone number, it will allow this information to be captured.



**Figure 1.1: Simplified database system environment**

1. **Objective**

The main objective of this project is to determine how an interactive inventory management system helps in the smooth functioning of a warehouse compared to traditional inventory management, by digitizing all the records and transacting everything on a computer rather than on paper. This project is a two-tier architecture application.

1. **Problem Statement**

Existing systems for Warehouse Inventory Management are very inefficient and mostly involve a lot of manual work to be done by the manager of the system. With this project, we want to automate as many tasks as possible using the available technology and the internet.

1. **Scope of the report**

The essential framework of this report would be to elaborate the design of E.R-diagram, Schema Diagram and to display how the modules of the program work in order to achieve the automation.

**CHAPTER 2**

**SYSTEM AND SOFTWARE REQUIREMENTS AND SPECIFICATIONS**

The program works on Desktop PC and is executed using a PHP 5 interface which interacts with a MySQL database running on localhost.

# **1 FUNCTIONAL REQUIREMENTS**

A description of the facility or feature required. Functional requirements deal with what the system should do or provide for users. They include description of the required functions, outlines of associated reports or online queries, and details of data to be held in the system.

## **Interface Requirements:**

* The system shall provide an option to add/delete quizzes with questions.
* The system shall provide an option to see and attend the quizzes.
* The system should give option for login for staffs and students.
* The system shall provide option to see scores.

# **NON-FUNCTIONAL REQUIREMENTS:**

Non-functional requirements define the overall qualities or attributes of the resulting system.

## **Usability**

Usability is the ease with which a user can learn to operate the online examination system and get results.

## **Security**

Security requirements are included in a system to ensure:

* All questions and users are well secured
* SQL injection is prevented

## **Reliability**

Reliability is the ability of a system to perform its required functions under stated conditions for a specific period of time. Constraints on the run-time behavior of the system can be considered under two separate headings:

* Availability: is the system available for service when requested by end-users.
* Failure rate: how often does the system fail to deliver the service as expected by end- users.

# **SOFTWARE REQUIREMENTS**

Programming language : MYSQL

Operating system : ANY OS (Recommended: Windows 8 , XP)

Application required : MYSQL Test bench

# **HARDWARE REQUIREMENTS**

CPU : Pentium IV 2.4 GHz or above

Memory (Primary) : 512 MB, 1 GB or above

Hard Disk : 40 GB, 80GB, 160GB or above

Monitor : 15 VGA color

**CHAPTER 3**

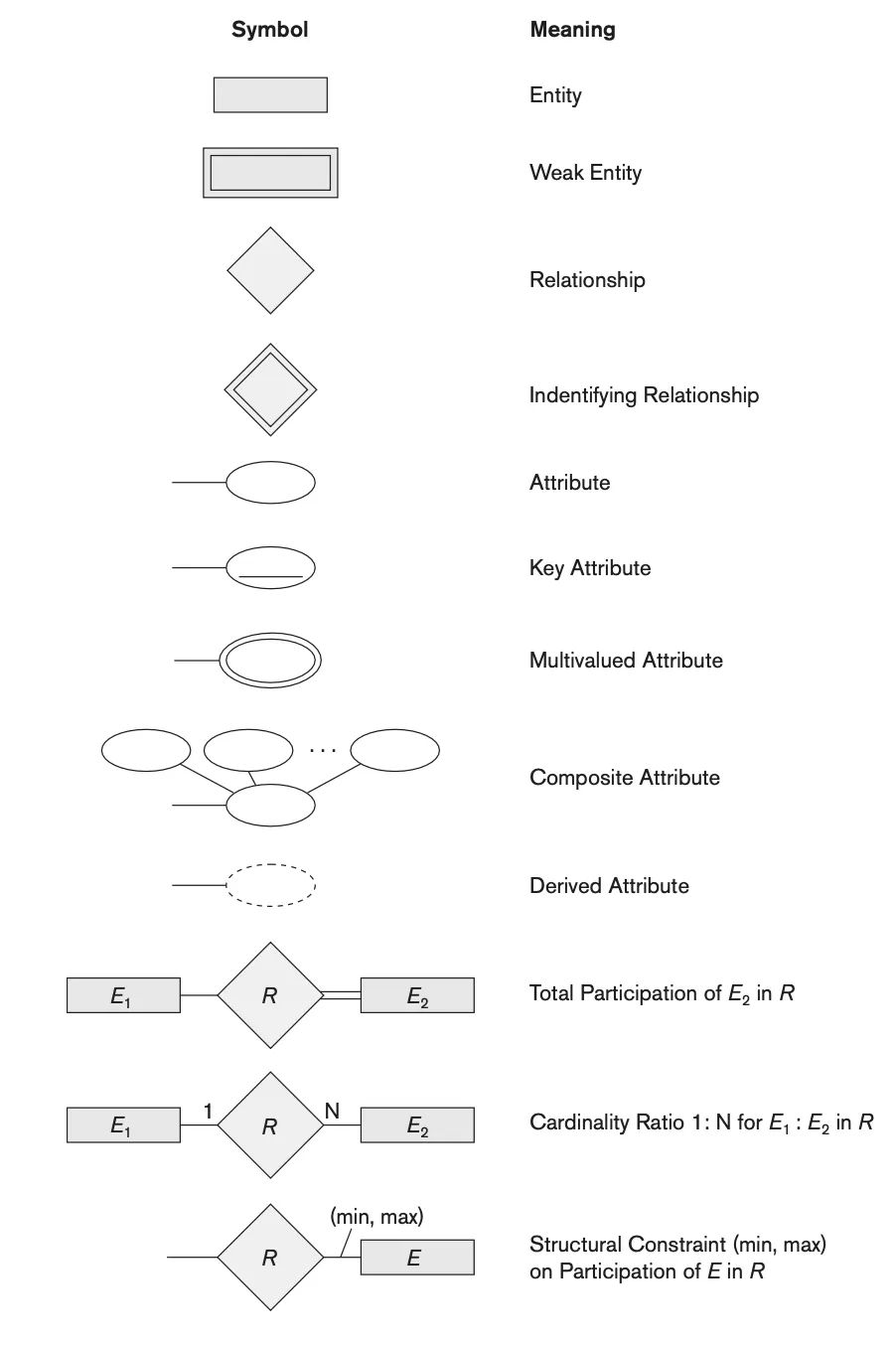
**ER DIAGRAM, SCHEMA DIAGRAM**

This chapter of the report describes the structure of the project, followed by Entity Relationship Diagram, Schema Diagram and the table structures.

1. **ER Diagram with relationships and cardinality ratio**

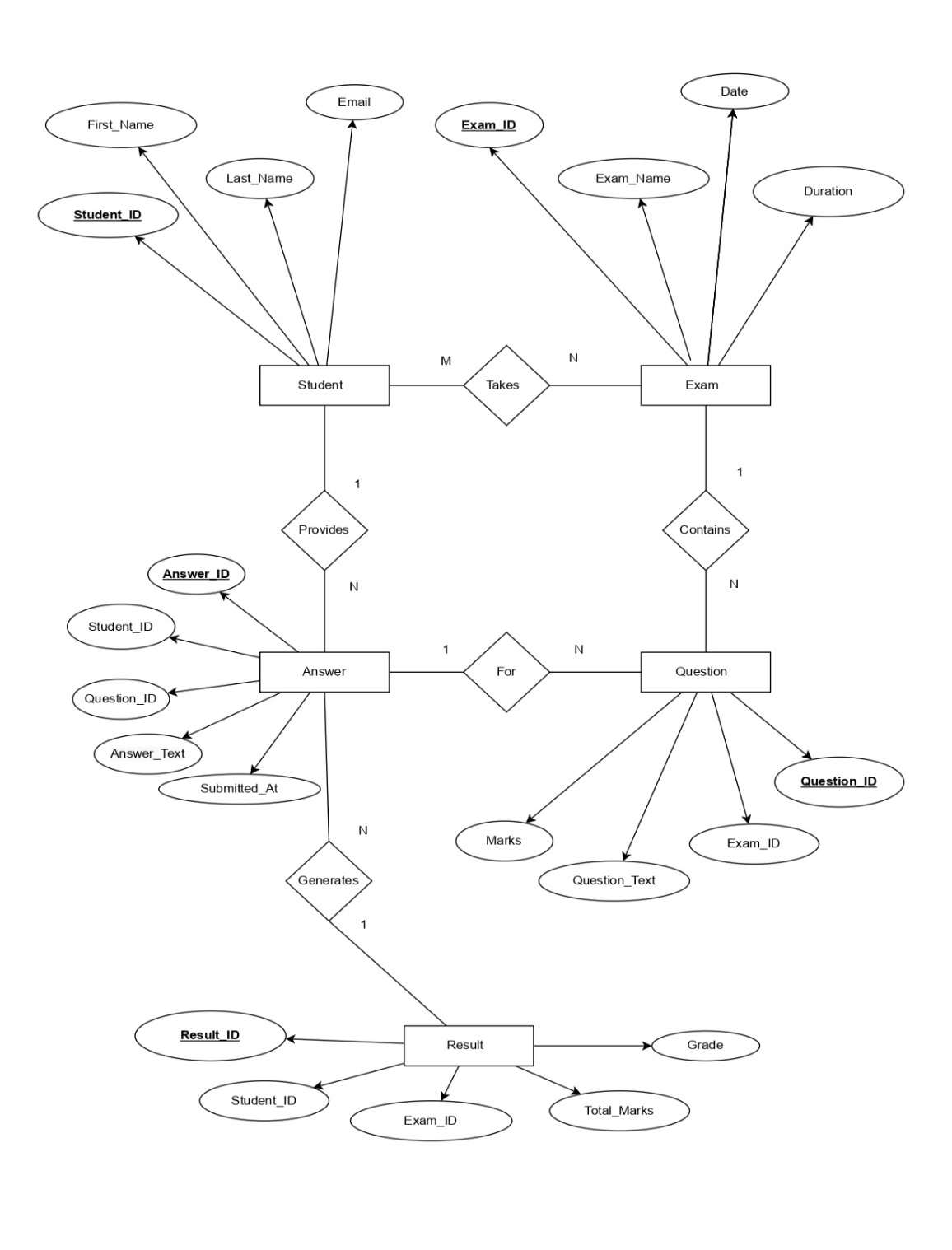
An entity relationship model, also called an entity-relationship (ER) diagram, is a graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of data within databases or information systems. An entity is a piece of data-an object or concept about which data is stored.

The cardinality or fundamental principle of one data aspect with respect to another is a critical feature. The relationship of one to the other must be precise and exact between each other in order to explain how each aspect links together. In simple words Cardinality is a way to define the relationship between two entities.

The following are the notations of the ER diagram:

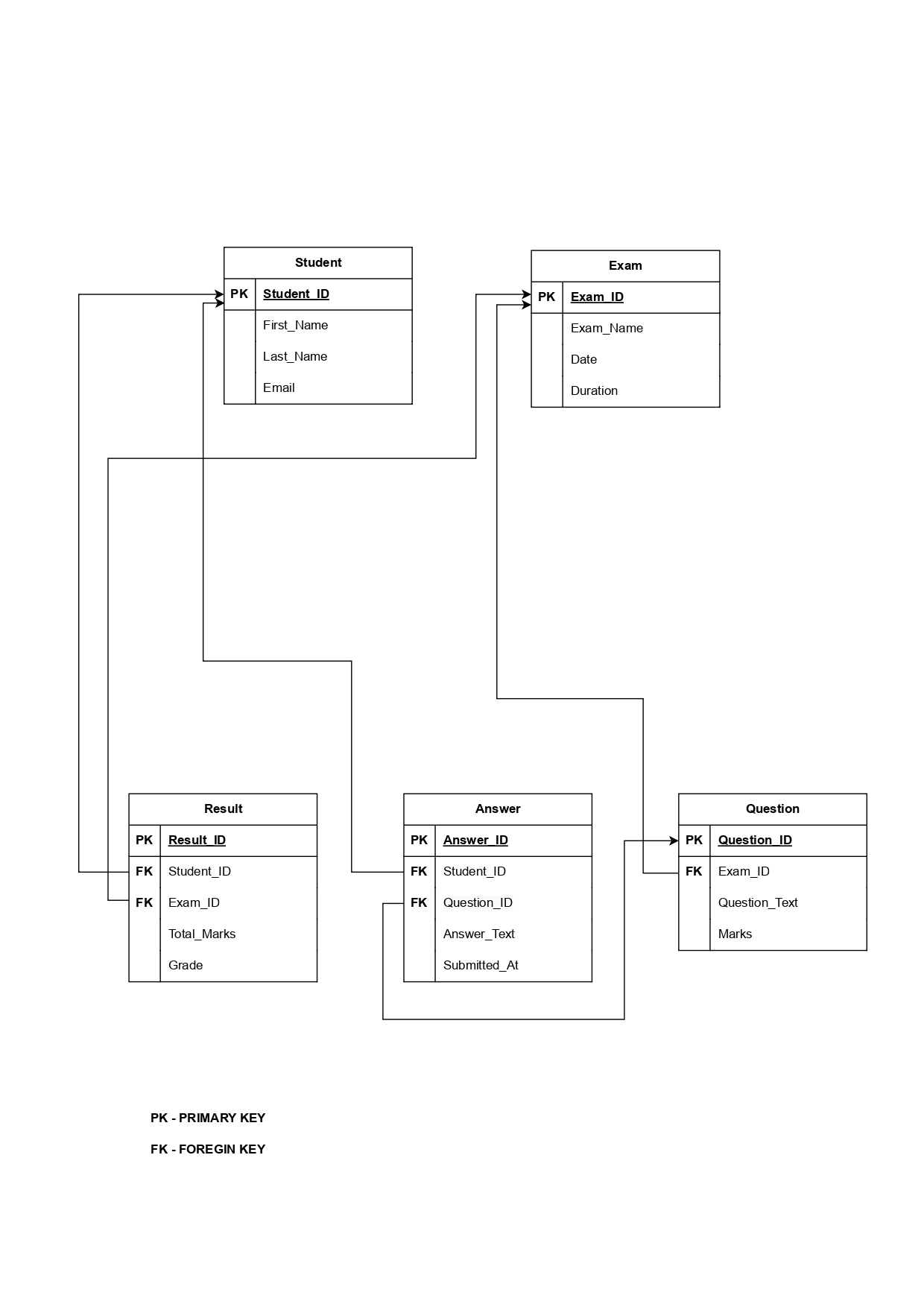
**Figure 3.1: Notations for ER Diagrams**

The ER diagram below shows the relationship between the many tables that exist in the database for the functioning of Warehouse Inventory Management System.



**Figure 3.2: ER Diagram of Online Examination System**

1. **Schema Diagram**

In any data model it is important to distinguish between the description of the database and the database itself. The description of a database is called the database schema, which is specified during database design and is not expected to frequently.

**Figure 3.3: Schema Diagram**

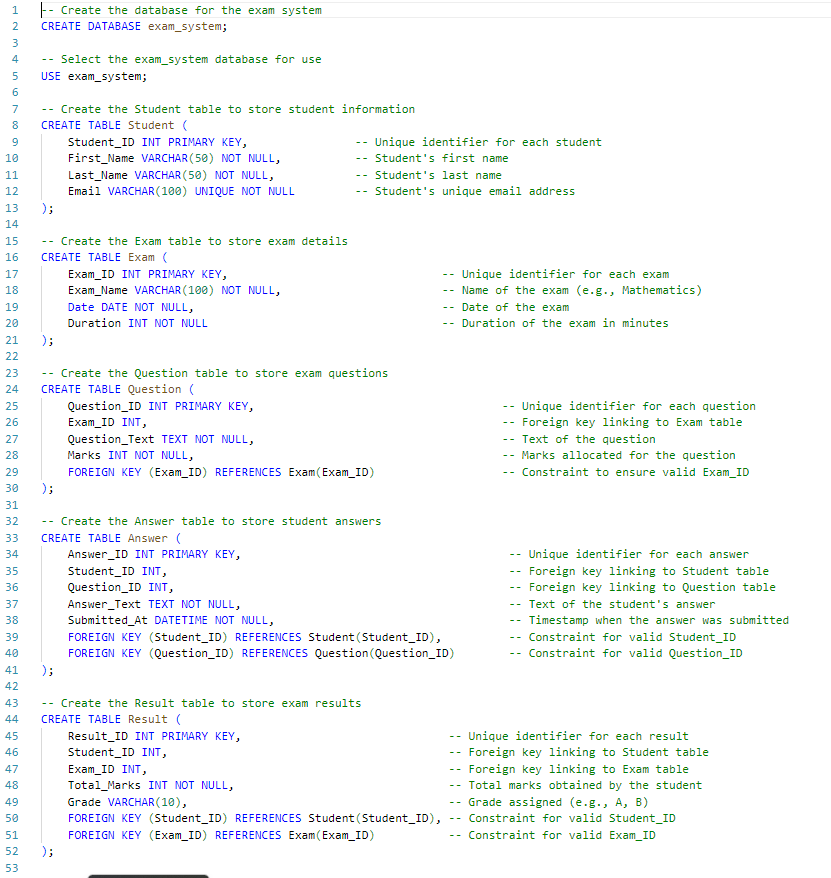
A displayed schema is called a schema diagram. A schema diagram displays only some aspects of a schema, such as the names of record types and data items, and some types of constraints.

**CHAPTER 4**

**SQL code snippet**

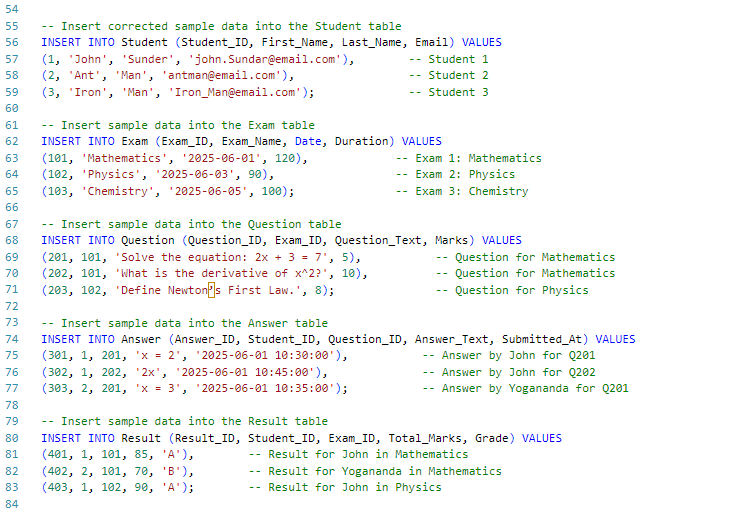
1. **SQL code**

SQL stands for **Structured Query Language**. It is a standardized programming language used to **manage**and manipulate relational databases. It enables users to perform a variety of tasks such as querying data, creating and modifying database structures, and managing access permissions.



**Figure 4.1: SQL code**

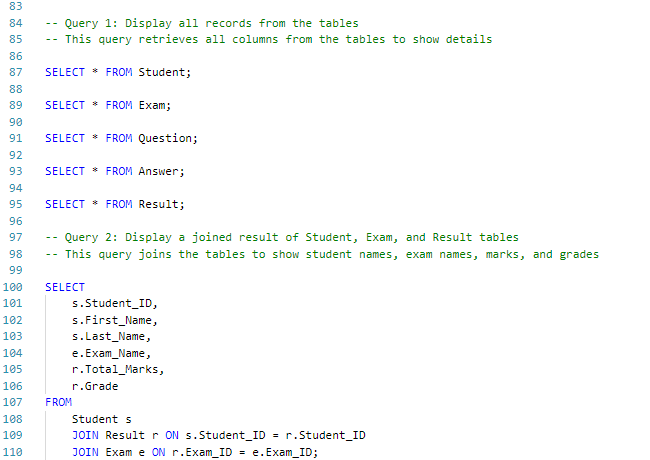
1. **Sampel inputs**

****

**Figure 4.2: Sampel inputs**

1. **Quearys**

A query in a database is a request made by the user or application to retrive or manipulate data stored in database

****

**Figure 4.3: Schema Diagram**

**CHAPTER 5**

**MODULES IN ONLINE EXAMINATION SYSTEM**

**1. User Management Module**

* Purpose: Manages user registration, authentication, and role-based access.
* Features:
  + Registration and login for students, administrators, and examiners.
  + Profile management (e.g., updating personal details like First\_Name, Last\_Name, Email from the Student table).
  + Role assignment (e.g., student, admin, instructor).
  + Password recovery and security features.
* Database Connection: Relies on the Student table for storing user details.

**2. Exam Management Module**

* Purpose: Handles creation, scheduling, and management of exams.
* Features:
  + Create and edit exams (e.g., Exam\_Name, Date, Duration in the Exam table).
  + Schedule exams with specific dates and times.
  + Assign exams to specific students or groups.
  + Set exam rules (e.g., duration, number of questions).
* Database Connection: Uses the Exam table to store exam details.

**3. Question Bank Module**

* Purpose: Manages the creation, storage, and organization of questions.
* Features:
  + Add, edit, or delete questions (e.g., Question\_Text, Marks in the Question table).
  + Categorize questions by exam or subject (linked via Exam\_ID).
  + Support multiple question types (e.g., MCQs, descriptive, true/false).
  + Randomize questions for different students to prevent cheating.
* Database Connection: Relies on the Question table for storing questions.

**4. Answer Submission Module**

* Purpose: Allows students to submit answers during the exam.
* Features:
  + Provide a user-friendly interface for students to answer questions.
  + Record answers with timestamps (e.g., Answer\_Text, Submitted\_At in the Answer table).
  + Support file uploads for descriptive answers (if applicable).
  + Auto-save answers to prevent data loss.
  + Time tracking to enforce exam duration.
* Database Connection: Uses the Answer table to store student responses.

**5. Result and Grading Module**

* Purpose: Evaluates answers, calculates marks, and generates results.
* Features:
  + Auto-evaluate objective questions (e.g., MCQs).
  + Manual evaluation for descriptive answers by examiners.
  + Calculate total marks and assign grades (e.g., Total\_Marks, Grade in the Result table).
  + Generate result reports and rank lists.
  + Notify students of results via email or dashboard.
* Database Connection: Relies on the Result table for storing results.

**CHAPTER 6**

**CONCLUSION**

The development and implementation of an Online Examination System within a Database Management System (DBMS) framework represent a significant advancement in modernizing educational assessment processes. This system leverages the power of DBMS to ensure efficient data organization, secure storage, and seamless management of examination-related information, including user profiles, question banks, test schedules, and results. Key points highlighting the system's impact and benefits include:

1. **Efficiency and Scalability**:

The system automates critical processes such as question retrieval, test conduction, and result generation, significantly reducing manual effort and time. Its scalable database architecture supports a growing number of users and exams without compromising performance.

1. **Data Integrity and Security**:

By utilizing DBMS features like data normalization, constraints, and encryption, the system ensures the accuracy, consistency, and confidentiality of sensitive information, such as student records and exam results, fostering trust among stakeholders.

1. **Accessibility and Flexibility**:

The online platform enables remote access, allowing students to take exams from any location with an internet connection. The system’s ability to support diverse question types and adaptive testing enhances its flexibility to cater to various educational needs.

1. **Real-Time Analytics and Reporting**:

The integration of DBMS enables real-time tracking of performance metrics and generation of detailed reports, empowering educators to make data-driven decisions for improving teaching and assessment strategies.

1. **Challenges and Future Scope**:

While the system offers numerous advantages, challenges such as ensuring robust cybersecurity, handling large-scale concurrent users, and maintaining user-friendly interfaces must be addressed. Future enhancements could include AI-driven question generation, advanced proctoring mechanisms, and integration with learning management systems.

1. **SCOPE OF ENHANCEMENT:**

There are also few features which can be integrated with the system to make it more flexible.

* + - Implementing the timer for the quiz.
    - Sending mails on sign up and when student takes the quiz.
    - Supporting all type of questions including MCQ’s**.**

**BIBLIOGRAPHY**

**BOOK REFERENCE**

1. Database System Models, Languages, Ramez Elmasri and Sham Kant B. Navathe, 7th Edition, 2017 Pearson.

2. Fundamentals of Web Development, Randy Connolly and Ricardo Hoar ,First Impression, 2016 Pearson

**WEBSITES**

**[1]**

[www.stackoverflow.com](http://www.stackoverflow.com)

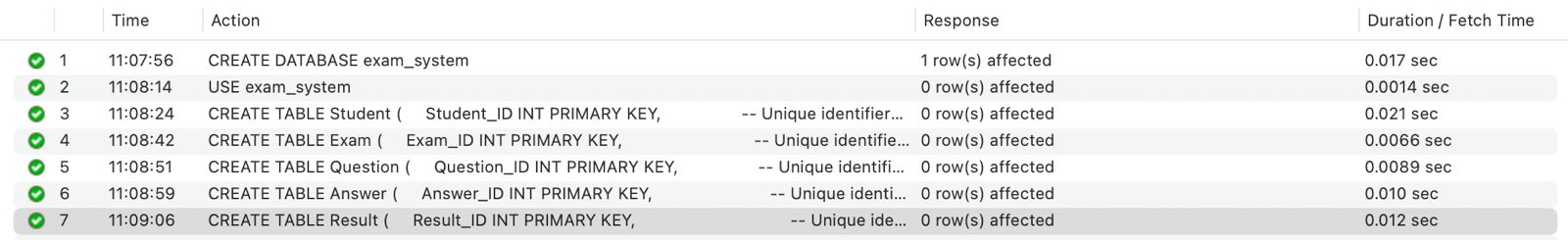
**[2]** [www.youtube.com/](https://www.youtube.com/" \o "www.youtube.com)

**[3]** [www.google.com](https://www.google.co.in/)

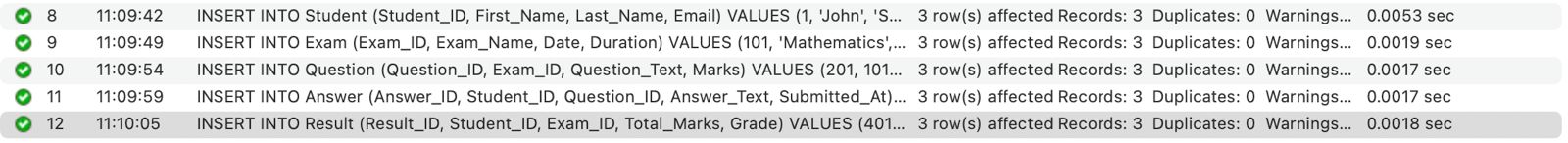
**[4]** [www.w3schools.com](http://www.w3schools.com)

**APPENDIX A**

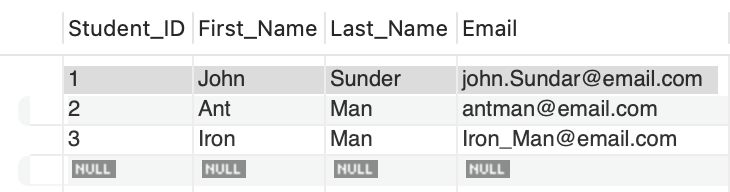
**SNAPSHOTS**



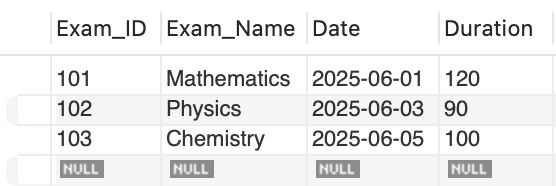
**Snapshot 1: Success creation of Database**



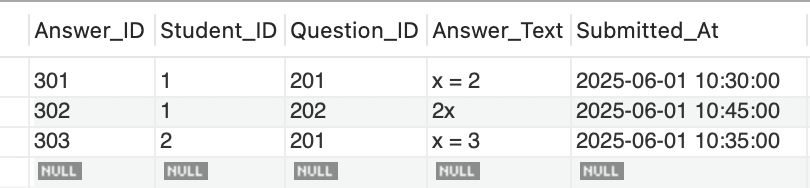
**Snapshot 2: Insert of data into the Tabl**



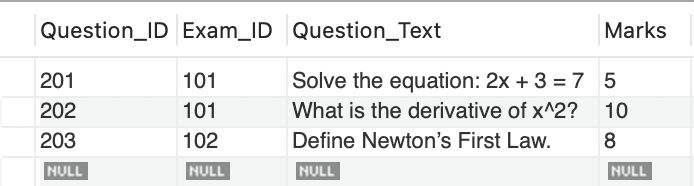
**Snapshot 3: Content of Student table**



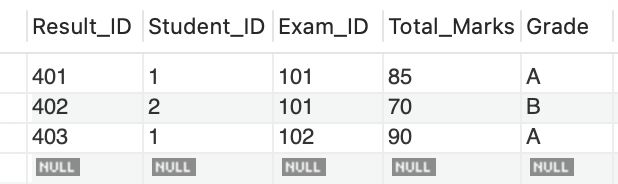
**Snapshot 4: Content of Exam table**



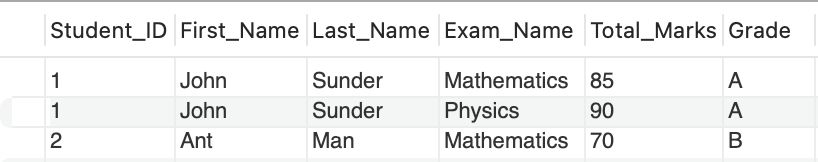
**Snapshot 5: Content of Answer table**



**Snapshot 6: Content of Question table**



**Snapshot 7: Content of Result table**



**Snapshot 8: Joined result of Student, Exam, and Result tables and show student id , student names, exam names, marks, and grades**