SHAHJALAL UNIVERSITY OF SCIENCE AND TECHNOLOGY, SYLHET

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COURSE CODE: CSE333 COURSE TITLE: DATABASE MANAGEMENT SYSTEM

Assignment

On

Understanding databases other than MySQL (IBM DB2)

SUBMITTED TO:

Ishtiaque Zahid

Lecturer

Department of Computer Science and Engineering

Shahjalal University of Science and Technology, Sylhet

SUBMITTED BY:

Name : Nehal Hasnain Alve

Registration: **2021331063**

3rd Year 1st Semester

Department of Computer Science and Engineering

Shahjalal University of Science and Technology, Sylhet

Date of Submission: 24.11.2025

IBM DB2

Introduction:

IBM DB2 is a family of data management products developed by IBM, primarily known as a powerful Relational Database Management System **(RDBMS)**. Initially created for IBM mainframes, DB2 has evolved into a multi-platform system that runs on Linux, UNIX, Windows, IBM iSeries, and z/OS systems. Over the years, it has added support for both SQL and NoSQL-style features, including XML storage, JSON integration, and support for big data platforms.

The "2" in the name stands for the second family of database software from IBM, which moved from the hierarchical data model to the relational model described by Edgar F. Codd in the 1970s. DB2 is designed to efficiently store, analyze, and retrieve large volumes of data and is primarily used in enterprise environments for mission-critical tasks like transaction processing (OLTP**)**, data warehousing (OLAP), and sophisticated analytics such as banking, government, healthcare, retail, telecommunications, and enterprise-level applications due to its performance, reliability, and scalability. It provides strong ACID compliance, high concurrency, and advanced data security—making it suitable for mission-critical systems.

Usage:

IBM DB2 can be used to create databases, manage data, run SQL queries, and build applications that interact with data using programming languages like Python, Java, C#, PHP, and more.

Below is an example demonstrating how to create and use a database in DB2.

1. Creating a Database:

```
CREATE DATABASE studentDB;
```

2. Connecting to the Database:

```
CONNECT TO studentDB USER db2user USING password;
```

3. Creating a Table:

```
CREATE TABLE Students (
    student_id INT PRIMARY KEY,
    name VARCHAR(100),
    department VARCHAR(50),
    cgpa DECIMAL(3,2)
);
```

4. Inserting Data:

```
INSERT INTO Students (student_id, name, department, cgpa)
VALUES (101, 'Nehal Hasnain Alve', 'CSE', 3.72);
```

5. Querying Data:

```
SELECT name, cgpa
FROM Students
WHERE department = 'CSE';
```

Use Case: Financial Services (Transaction Processing)

A common use case for IBM DB2, particularly the DB2 for z/OS version running on IBM mainframes, is in the banking and financial services industry.

- Scenario: A large bank requires a robust, highly available system to process millions of customer transactions (deposits, withdrawals, transfers) daily.
- Role of DB2: DB2 provides the necessary high performance, reliability, and data integrity to handle these massive, concurrent, and mission-critical OLTP workloads. Its features like ACID compliance, advanced locking, and High Availability Disaster Recovery (HADR) ensure that transactions are processed quickly and accurately, and the system experiences near-zero downtime, which is crucial for a financial institution.

Advantages and Disadvantages

Advantages

1. **High Performance and Scalability**
   DB2 is optimized for large-scale enterprise applications and supports parallel processing, in-memory computing, and high throughput.
2. **Robust Security**
   It offers comprehensive security features, including column-level encryption, fine-grained access controls, and auditing, which is essential for regulated industries like finance and healthcare.
3. **Cross-Platform Availability**
   Works on Linux, Windows, UNIX, IBM i, and IBM mainframes, making it suitable for organizations with multiple systems.
4. **Advanced Data Types Support**
   Supports XML, JSON, graph data, and spatial data, making it versatile for modern applications.
5. **Automatic Maintenance Tools**
   Features like self-tuning memory management reduce the need for manual database tuning.

Disadvantages

1. **Not Completely Free**
   Enterprise versions require licensing fees, which can be expensive for students or small organizations.
2. **Complex Setup for Beginners**
   Installation, configuration, and performance tuning require expertise.
3. **Smaller Community Support**
   Compared to MySQL or PostgreSQL, DB2 has fewer community forums and open-source examples.
4. **Heavily Enterprise-Focused**
   Best suited for large organizations; may be unnecessary for small-scale projects.