



Student Department Management System

A comprehensive DBMS mini-project for academic workflow management

Introduction & Purpose

System Overview

A menu-driven database application designed to streamline student-department data management in educational institutions.

Core functionality: efficient record handling, departmental organization, and real-time data operations.



Project Objectives



Centralized Management

Single interface for all student-department operations



Efficient Retrieval

Quick access to departmental student records



Dynamic Updates

Real-time add, modify, and delete capabilities



User-Friendly Design

Intuitive menu-driven navigation system

System Workflow Architecture



View Module

Department-wise student listing



Manage Module

Add • Delete • Update operations



Exit Option

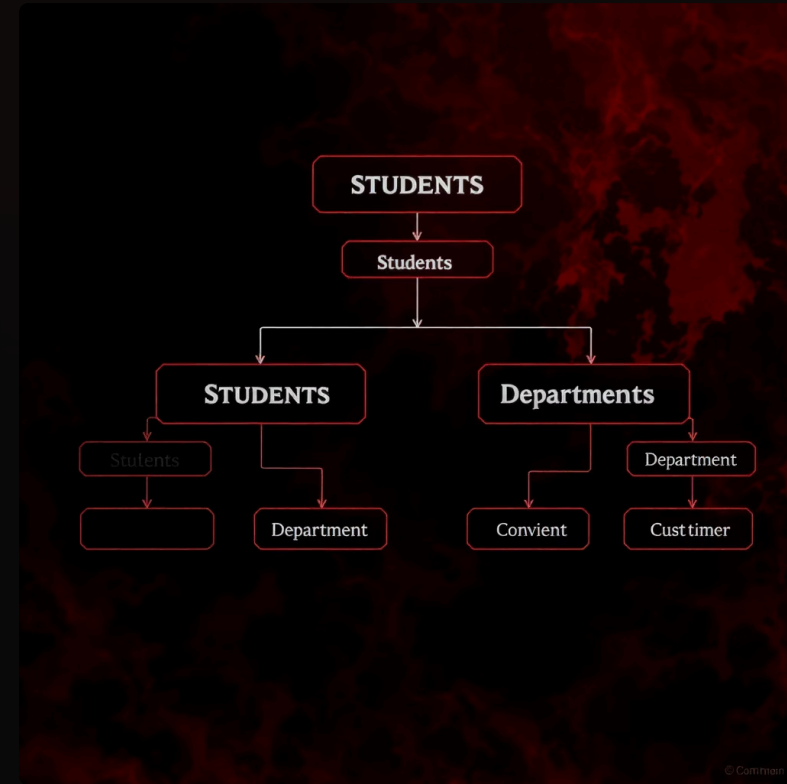
Safe program termination

Database Design Structure

Entity Relationships

Students table linked to **Departments** table via **dept_id** foreign key

- One-to-many relationship
- Referential integrity maintained
- Normalized schema design



❏ **Key Constraint:** Each student belongs to exactly one department; departments can have multiple students

Table Structure Specifications

1

Students Table

- **roll_no** (Primary Key) – Unique identifier
- **name** (VARCHAR) – Student full name
- **dept_id** (Foreign Key) – Department reference
- **email** (VARCHAR) – Contact information

2

Departments Table

- **dept_id** (Primary Key) – Department code
- **dept_name** (VARCHAR) – Department name
- **hod_name** (VARCHAR) – Head of department
- **location** (VARCHAR) – Building/floor info

Implementation Technology Stack



Java Application Layer

Menu-driven console interface with object-oriented design patterns for robust functionality



SQL Database Engine

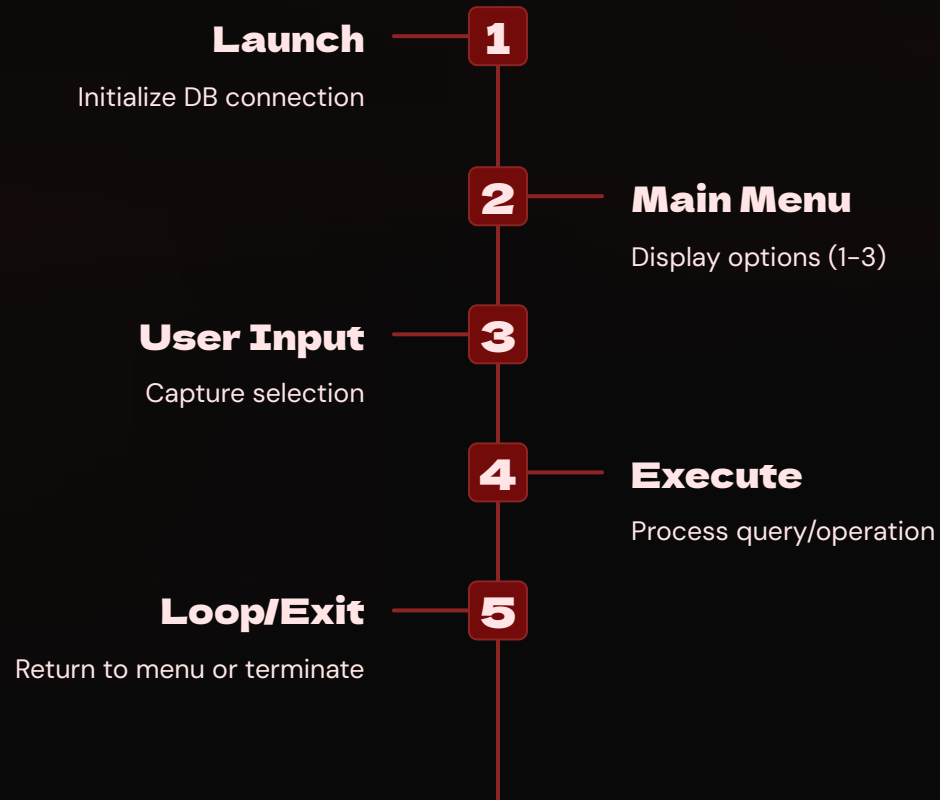
Oracle/MySQL RDBMS for structured data storage with ACID transaction support



JDBC Connectivity

Java Database Connectivity API bridges application and database seamlessly

Program Flow Mechanism



"Manage sub-menu offers Add, Delete, Update operations with validation and confirmation prompts before database commits"

Key Features & Advantages



Department-Centric Viewing

Filter and display students by academic department instantly



Complete CRUD Operations

Create, Read, Update, Delete — full data lifecycle control



Data Integrity

Foreign key constraints prevent orphaned records



Scalability Ready

Architecture supports expansion to additional modules



Future Enhancements



Advanced Search

Multi-criteria filtering: by name, roll number, email, or department



Analytics Dashboard

Department-wise statistics, enrollment trends, visual reports



Web Interface

Browser-based GUI replacing console for better accessibility



Authentication System

Role-based access control for admin, faculty, and student users

Thank You

Questions & Discussion Welcome