Case Study Article on SAIT

Written By: Jigar Patel

Guided By: DR. Junaid Qazi

LinkedIn: Jigar Patel

Table of Contents

1. Introduction

1.1. Purpose of the database design

The purpose of SAIT database design is to efficiently manage and organize large volumes of data related to students, faculty, courses, and administrative functions. It ensures data accuracy, consistency, and security while streamlining operations like student enrollment, academic record tracking, class scheduling, and reporting. A well-structured design supports seamless access to information, enabling informed decision-making and improving the overall efficiency of academic and administrative processes.

1.2. Overview of SAIT

The Southern Alberta Institute of Technology (SAIT) offers hands-on degree, diploma, certificate, continuing education and corporate training programs to develop the skills and knowledge employer's demand.

2. Mission

SAIT is dedicated to preparing students for successful careers and fulfilling lives by providing hands-on skills and training that empower them to thrive in a dynamic world.

2.1. Objectives

- **Skills for the future**: Ensuring students gain relevant skills for emerging industries.
- **Learning for life**: Promoting continuous learning and adaptability.
- **Global perspective**: Instilling a broad worldview for international competitiveness.
- **Industry - driven**: Aligning with industry standards and demands.
- **Commitment to excellence**: Upholding high standards of academic and operational excellence.

3. Database Design Overview

Database design is the process of structuring data to create an efficient, logical system for storing, retrieving, and managing information. It involves defining tables, fields, relationships, and constraints to organize data according to user needs. The design process typically includes conceptual (high-level), logical (detailed structure), and physical (storage method) phases. Key goals are ensuring data integrity, reducing redundancy, optimizing performance, and supporting scalability for future growth.

3.1. Key Entities

- Student
- Faculty
- Courses
- Departments
- Classrooms
- Grades
- Enrollment
- Process (Enrollment Process Tracking)

4. Tables and Their Attribute

4.1. Student Table

Description: This table maintains detailed profiles of students. It helps track student demographics, contact details, and academic standing.

Fields: Student_id, Firstname, Lastname, DOB (Date of Birth), Address, Gender, Phone, Email

4.2. Faculty Table

Description: This table records faculty members' information, which helps in course assignment and communication.

Fields: Faculty_id, Firstname, Lastname, Phone, Email

4.3. Course Table

Description: This table catalogs all courses offered by the university, facilitating course selection and enrollment.

Fields: Course_id, Course_name

4.4. Department Table

Description: This table organizes academic departments and helps in managing courses and faculty assignments.

Fields: Department_id, Department_name

4.5. Classroom Table

Description: This table tracks physical classroom resources, helping in scheduling courses and managing space.

Fields: Classroom_id, Classroom_no, Building_name

4.6. Grades Table

Description: This table records academic performance, which is crucial for assessing student progress and eligibility for graduation.

Fields: Grade_id, Grade

4.7. Enrollment Table

Description: This table tracks which students are enrolled in which courses, managing course registrations and updates.

Fields: Enrollment_id, Enrollment_date

4.8. Process Table

Description: This table monitors the steps in the enrollment process, ensuring that all necessary actions are tracked for each student.

Fields: Process_id, Steps, Date, Status

5. Relationship Between Tables

5.1. Student → Department (Many – to - One)
Many students can belong to one department, but a student belongs to only one department.

5.2. Student → Enrollment (One - to - Many)
A student can enroll in many courses, but each enrollment record is specific to a single student.

5.3. Student → Courses (Many – to – Many) (via Enrollment table)
A single course can have many students enrolled, and a student can enroll in many courses.

5.4. Department → Courses (One- to - Many)
A department offers many courses, but each course belongs to a specific department.

5.5. Courses → Faculty (Many - to - One)
A faculty member can teach many courses, but each course typically has one assigned faculty member.

5.6. Faculty → Department (Many - to - One)
A department can have many faculty members, but each faculty member belongs to a single department.

5.7. Classroom → Courses (One – to – Many)
Each classroom can host multiple courses over time, but a course is usually assigned to one classroom.

5.8. Student → Grades (One - to - Many)
A student receives many grades, but each grade entry is associated with a single student and a specific course.

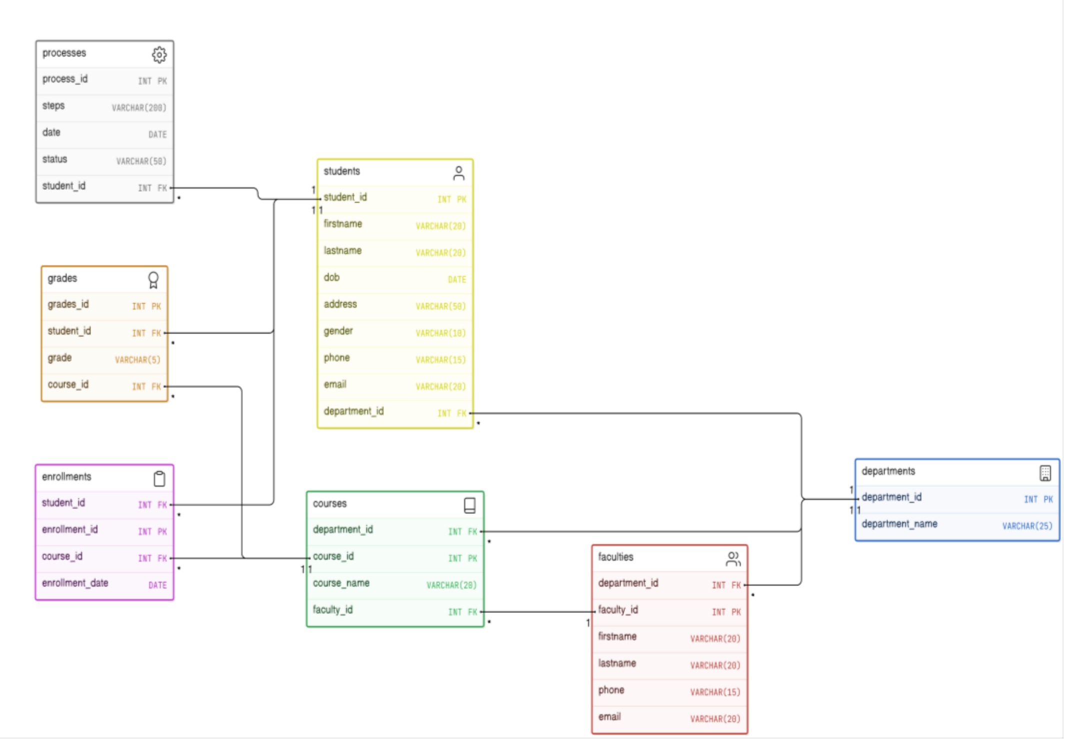
5.9. Courses → Grades (One - to - Many)
A course can have many grades assigned to different students, but each grade belongs to a specific course.

6. Entity Relationship Diagram (ERD)

6.1. Overview of the ERD

An Entity-Relationship Diagram (ERD) visually represents the relationships between data entities in a database system. In the context of SAIT enrollment system, an ERD outlines key entities like Students, Courses, Faculty, Departments, Classrooms, Grades, and Enrollment, along with their attributes and how they relate to one another.

6.2. ER Diagram



7. Conclusion

In conclusion, the SAIT enrollment database management system streamlines and enhances the efficiency of managing academic data, including student information, course offerings, faculty details, and enrollment records. It ensures data integrity, reduces redundancy, and facilitates easy access to critical information, enabling smooth enrollment processes and academic progress tracking. By centralizing and automating administrative functions, the system improves decision-making, supports real-time reporting, and enhances the overall operational effectiveness of the university. It is an essential tool for modern educational institutions to manage and scale their academic operations efficiently.

8. Appendix

8.1. Table and Description

- **Student Table**

Field Name	Data Type
Student_id	INT PK
Firstname	VARCHAR (20)
Lastname	VARCHAR (20)
DOB(Date of Birth)	DATE
Address	VARCHAR (50)
Gender	VARCHAR (10)
Phone	VARCHAR (15)
Email	VARCHAR (20)

- **Faculty Table**

Field Name	Data Type
Faculty_id	INT PK
First name	VARCHAR (20)
Lastname	VARCHAR (20)
Phone	VARCHAR (15)
Email	VARCHAR (20)

- **Course Table**

Field Name	Data Type
Course_id	INT PK
Course_name	VARCHAR (20)

- **Department Table**

Field Name	Data Type
Department_id	INT PK
Department_name	VARCHAR (25)

- **Classroom Table**

Field Name	Data Type
Classroom_id	INT PK
Classroom_no	VARCHAR (20)
Building_name	VARCHAR (20)

- **Grades Table**

Field Name	Data Type
Grade_id	INT PK
Grade	VARCHAR (5)

- **Enrollment Table**

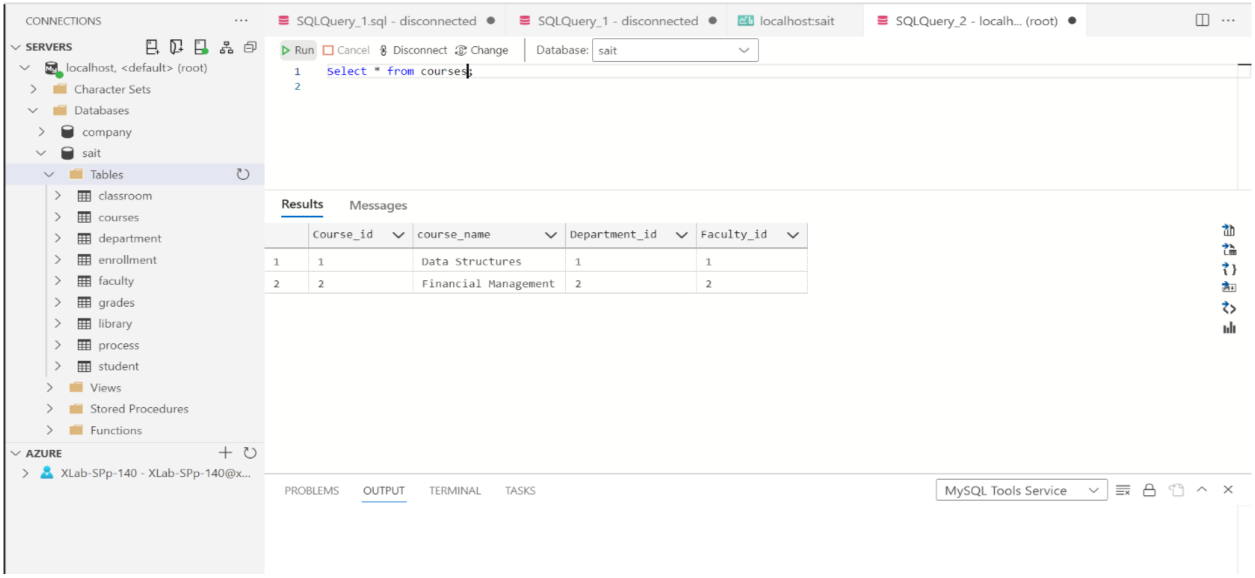
Field Name	Data Type
Enrollment_id	INT PK
Enrollment_date	DATE

- **Process Table**

Field Name	Data Type
Process_id	INT PK
Status (Completed, Pending)	VARCHAR (50)
Date	DATE
Steps	VARCHAR (200)

8.2. Testing Database and Query

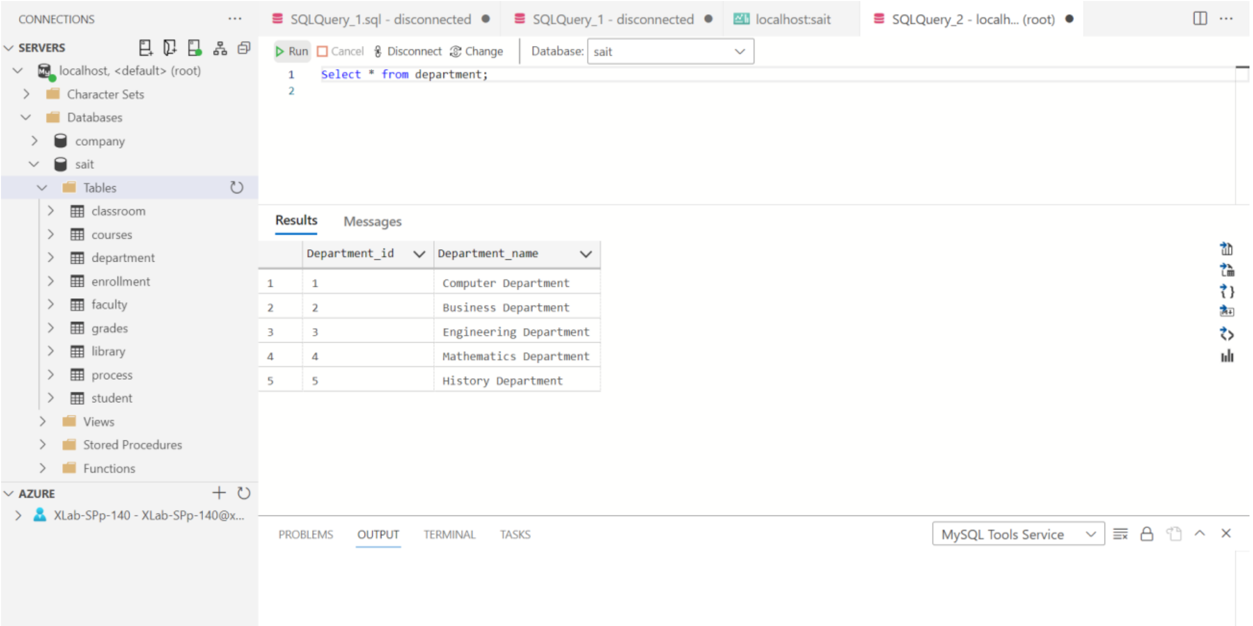
- **Display Course Table Data Using Select Query**



```sql
Select * from courses;
```

	Course_id	course_name	Department_id	Faculty_id
1	1	Data Structures	1	1
2	2	Financial Management	2	2

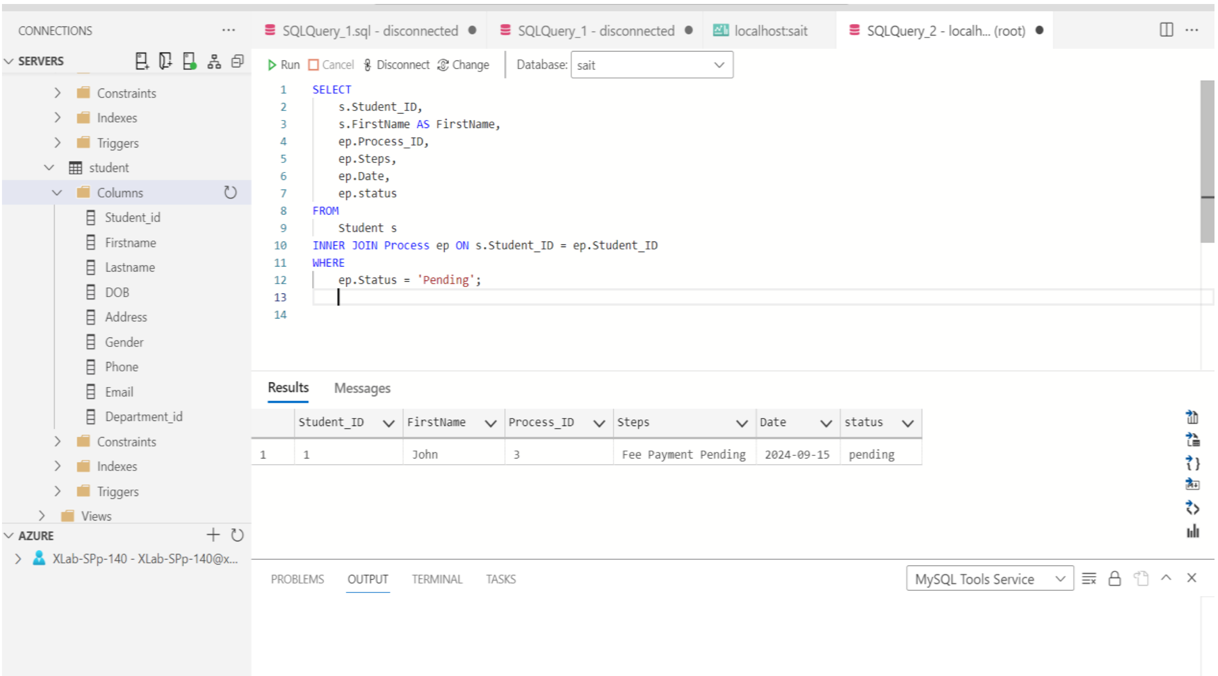
- **Display Department Table Data Using Select Query**



```sql
Select * from department;
```

	Department_id	Department_name
1	1	Computer Department
2	2	Business Department
3	3	Engineering Department
4	4	Mathematics Department
5	5	History Department

- **Finding Pending Enrollment Student Using Inner Join**



```sql
SELECT
    s.Student_ID,
    s.FirstName AS FirstName,
    ep.Process_ID,
    ep.Steps,
    ep.Date,
    ep.status
FROM
    Student s
INNER JOIN Process ep ON s.Student_ID = ep.Student_ID
WHERE
    ep.Status = 'Pending';
```

	Student_ID	FirstName	Process_ID	Steps	Date	status
1	1	John	3	Fee Payment Pending	2024-09-15	pending

8.3. Enrollment Process

Step 1: Application Submission

Relevant Tables: Students, Departments, Process

Description: The student selects a Department during the application submission. The system associates the student with the selected department.

System Activity :

A new record is created in the Student table, linking the student to a department (via department_id).

The Process table logs "Application Submitted."

Example in Tables:

Students Table:

student_id	firstname	Lastname	dob	email	department_id
1	**John**	**Doe**	**2000-01-01**	**john@example.com**	**1**

Departments Table:

department_id	department_name
1	**Computer Science**

Step 2: Document Verification

Relevant Tables: Students, Process

Description: The university verifies the documents provided by the student during the application.

System Activity:

The Process table logs the step as "Documents Verified" when the documents are approved.

Example in Process Table:

Steps	Date	status	student_id
Documents Verified	**2024-08-05**	**completed**	**1**

Step 3: Offer Letter or Admission Confirmation

Relevant Tables: Process, Departments

Description: The department makes the final decision and confirms the student's admission, after verified documents.

System Activity:

The Process table logs "Admission Confirmed."

The Departments table provides the relevant department where the student is being admitted.

Example in Process Table:

Steps	Date	status	student_id
Admission Confirmed	**2024-08-15**	**completed**	**1**

Step 4: Fee Payment

Relevant Tables: Process, Students

Description: The student pays the required fees for their selected courses.

System Activity:

The Process table logs the fee payment step. If the payment is pending, the status remains "pending."

Step 5: Course Enrollment

Relevant Tables: Enrollment, Courses, Process, Faculty

Description: The student selects and enrolls in courses offered by the department. Courses are taught by Faculty members, and they are linked to the student via the Enrollment table.

System Activity

The Enrollment table links the student to the courses they have chosen. The Courses table provides the details of available courses.

The Process table logs the completion of course enrollment.

Example in Enrollment Table:

enrollment_id	student_id	course_id	enrollment_date
1	**1**	**101**	**2024-09-01**

Courses Table:

course_id	course_name	faculty_id	department_id
101	Introduction to Programming	1	1

Faculty Table:

faculty_id	firstname	lastname	email	department_id
1	Dr. Emily	Brown	emily@faculty.com	1

Process Table:

steps	Date	status	student_id
Courses Enrolled	2024-09-01	completed	1

Step 6: Classroom Assign

Relevant Tables: Classrooms, Courses

Description: Each course has a scheduled classroom where the students will attend lectures. Classrooms are assigned based on the course schedule.

System Activity:

The Classrooms table stores information about the room's location, linking it to the course through classroom_id.

Example in Classroom Table:

classroom_id	classroom_no	Classroom_name
1	Room 101	Building A

Courses Table (with classroom):

course_id	course_name	classroom_id
101	Introduction to Programming	1

Step 7: Issuance of Student ID and Access

Relevant Tables: Process, Students

Description: The student receives their student ID and gains access to university services.

System Activity:

The Process table logs when the student ID is issued.

Example in Process Table:

steps	Date	status	student_id
Student ID Issued	**2024-09-05**	**completed**	**1**

Step 8: Grades Awarded

Relevant Tables: Grades, Enrollment, Courses

Description: After completing the course, the student receives a grade. The grades for each course are recorded in the Grades table.

System Activity:

The Grades table stores the grades the student has earned for each course. The Process table logs the completion of the course as a final step.

Example in Grades Table:

grade_id	student_id	course_id	grade
1	**1**	**101**	**A**