

Table of Contents

<i>Introduction</i>	2
<i>Mission</i>	2
<i>Objectives</i>	2
<i>Database Design and Tables</i>	2
Table 1: Student	2
Table 2: Courses	3
Table 3: Department	4
Table 4: Faculty	4
Table 5: Classroom	5
Table 6: Enrollment	5
Table 7: Grades	6
Table 8: Process	6
<i>ER Diagram and Relationships</i>	7
<i>Joins</i>	8
<i>Conclusion</i>	8
<i>Views</i>	9
<i>Appendices</i>	9
Appendix A: Database Tables and SQL Queries	9
Appendix B: Test Queries	12

Introduction

In today's educational landscape, efficient data management is crucial. SAIT (Southern Alberta Institute of Technology, Calgary) has implemented a comprehensive Database Management System (DBMS) that streamlines the handling of academic and administrative data. By integrating student, course, and faculty management, SAIT's DBMS ensures the effective organization of essential information while supporting the institution's mission to prepare students for successful careers.

Mission

SAIT is dedicated to empowering students with hands-on skills and training that prepare them for dynamic careers. Its DBMS aligns with this mission by ensuring the efficient organization and retrieval of vital academic and administrative information.

Objectives

The main objectives of SAIT's DBMS are:

- Centralizing and organizing institutional data to improve efficiency.
 - Reducing redundancy and enhancing data accuracy.
 - Facilitating quick access to student records, courses, faculty details, and grades for administrators and staff.
-

Database Design and Tables

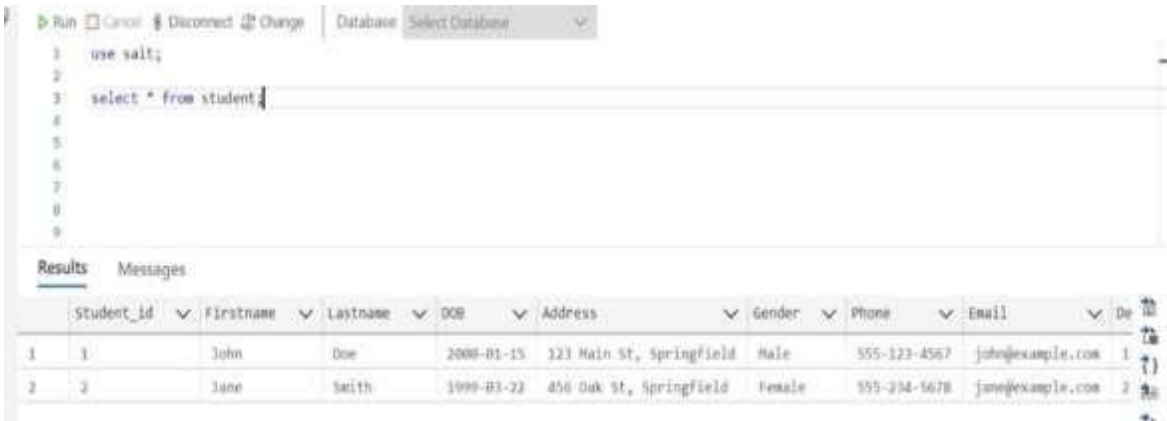
The SAIT DBMS is structured around several core tables. Below is an outline of the key tables, their fields, and data types.

Table 1: Student

- Fields:

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)
Department_id	INT FK



The screenshot shows a SQL query execution window with the following SQL code:

```
1 use salt;
2
3 select * from student;
```

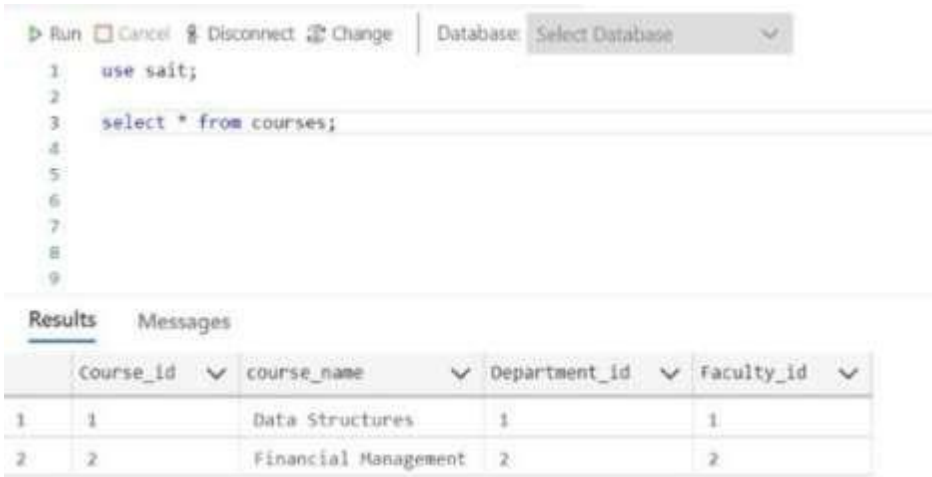
The results are displayed in a table with the following columns: Student_id, Firstname, Lastname, DOB, Address, Gender, Phone, Email, and Department_id. The data is as follows:

	Student_id	Firstname	Lastname	DOB	Address	Gender	Phone	Email	Department_id
1	1	John	Doe	2000-01-15	123 Main St, Springfield	Male	555-123-4567	john@example.com	1
2	2	Jane	Smith	1999-03-22	456 Oak St, Springfield	Female	555-234-5678	jane@example.com	2

Table 2: Courses

- Fields:

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



The screenshot shows a SQL query execution window with the following SQL code:

```
1 use salt;
2
3 select * from courses;
```

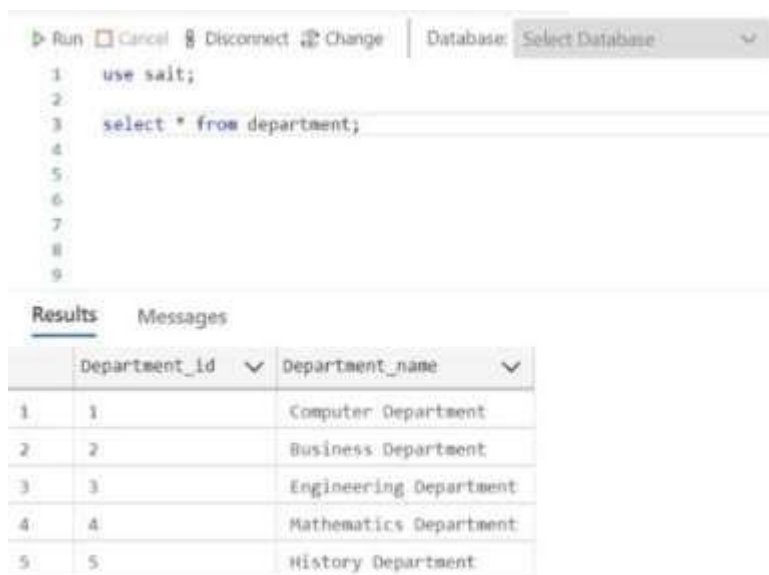
The results are displayed in a table with the following columns: Course_id, course_name, Department_id, and Faculty_id. The data is as follows:

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

Table 3: Department

- Fields:

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



The screenshot shows a SQL IDE interface. At the top, there are buttons for 'Run', 'Cancel', 'Disconnect', and 'Change', along with a 'Database:' dropdown menu set to 'Select Database'. Below this is a query editor with the following SQL code:

```
1 use salt;
2
3 select * from department;
4
5
6
7
8
9
```

Below the query editor, there are two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with the following data:

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

Table 4: Faculty

- Fields:

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

Run

Cancel

Disconnect

Change

Database:

Select Database

1

use sa1t;

2

3

select * from faculty;

4

5

6

7

8

9

Results

Messages

	Faculty_id	Firstname	Lastname	Phone	Email	Department_id
1	1	Michael	Jordan	555-987-6543	michael@example.com	1
2	2	Sarah	Connor	555-876-5432	sarah@example.com	3

Table 5: Classroom

- Fields:

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

Run	Cancel	Disconnect	Change	Database: Select Database
1	use sait;			
2				
3	select * from classroom;			
4				
5				
6				
7				
8				
9				

Results		Messages	
	Classroom_id	Classroom_no	Building_name
1	1	C-101	Science Building
2	2	B-201	Business Building

Table 6: Enrollment

- Fields:

Field Name	Data Type
Enrollment_id	INT PK

Student_id	INT FK
Course_id	INT FK
Enrollment_date	DATE

Run

Cancel

Disconnect

Change

Database: Select Database

```

1 use sait;
2
3 select * from enrollment;

```

Run

Cancel

Disconnect

Change

Database: Select Database

```

1 use sait;
2
3 select * from grades;
4
5
6
7

```

Run

Cancel

Disconnect

Change

Database: Select Database

```

1 use sait;
2
3 select * from grades;
4
5
6
7
8
9

```

Results

Messages

	Grades_id	Student_id	Course_id	Grade
1	1	1	1	A
2	2	2	2	B

Grades_id	INT PK
Student_id	INT FK
Course_id	INT FK
Grade	VARCHAR (5)

Table 8: Process

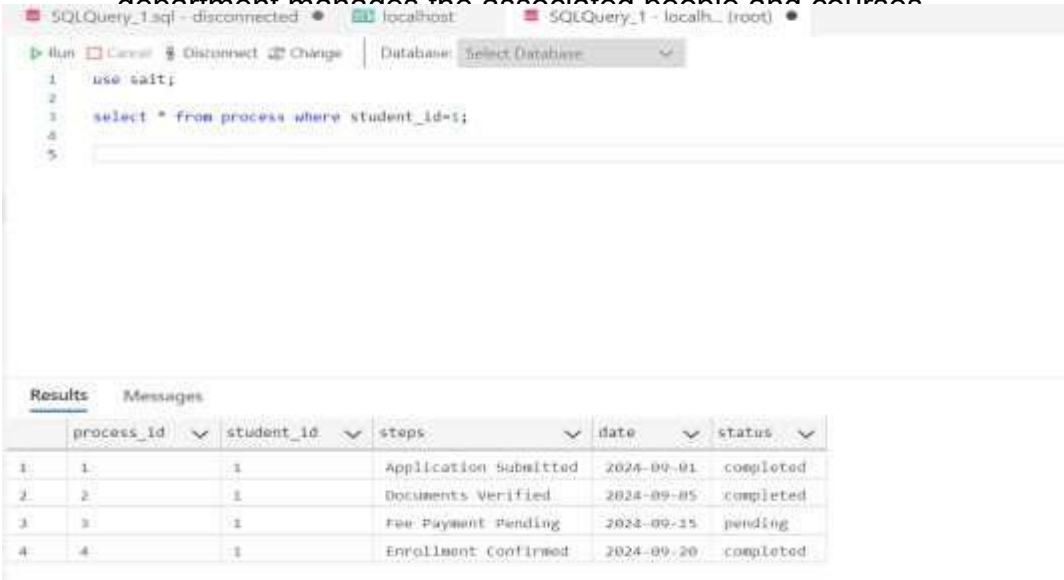
- Fields:

Field Name	Data Type
Process_id	INT PK
Status (Completed, Pending)	VARCHAR(50)

ER Diagram and Relationships

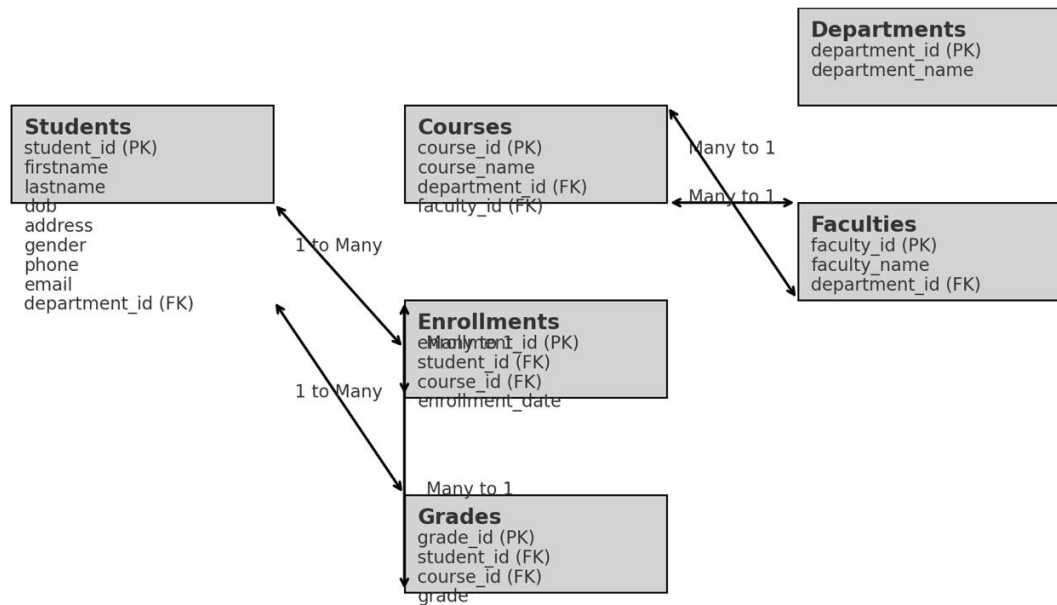
SAIT database is structured using an Entity-Relationship Diagram (ERD), where each table (entity) is connected through relationships. These relationships are established via primary and foreign keys:

- Enrolment acts as the central table that links the Student, Courses, and Faculty tables. This allows administrators to track which students are enrolled in which courses, who is teaching those courses, and the grades students receive.
- The Department table is related to both Students and Faculty, ensuring each department manages the associated people and courses.



```
1 use sait;
2
3 select * from process where student_id=1;
4
5
```

	process_id	student_id	steps	date	status
1.	1	1	Application Submitted	2024-09-01	completed
2.	2	1	Documents Verified	2024-09-05	completed
3.	3	1	Fee Payment Pending	2024-09-15	pending
4.	4	1	Enrollment Confirmed	2024-09-20	completed



Joins

- Enrollment is the central table in this query.
- Student is joined on Student_id to get student details.
- Courses is joined on Course_id to get course details
- Grades is joined using both Student_id and Course_id to get the students' grades for each course.
- The ORDER BY clause orders the results by student and course.

Inner Join Queries:

SQLQuery_1.sql - disconnected • SQLQuery_1 - localh... (root) •

Run Cancel Disconnect Change Database: sql

```

1  SELECT
2      Student.Student_id,
3      Student.Firstname AS student_firstname,
4      Student.Lastname AS student_lastname,
5      Student.Email AS student_email,
6      Courses.Course_id,
7      Courses.Course_name,
8      Grades.Grade,
9      Enrollment.Enrollment_date
10 FROM
11     Enrollment
12 INNER JOIN Student ON Enrollment.Student_id = Student.Student_id
13 INNER JOIN Courses ON Enrollment.Course_id = Courses.Course_id
14 INNER JOIN Grades ON Student.Student_id = Grades.Student_id AND Courses.Course_id = Grades.Course_id
15 ORDER BY Student.Student_id, Courses.Course_id;
16
  
```

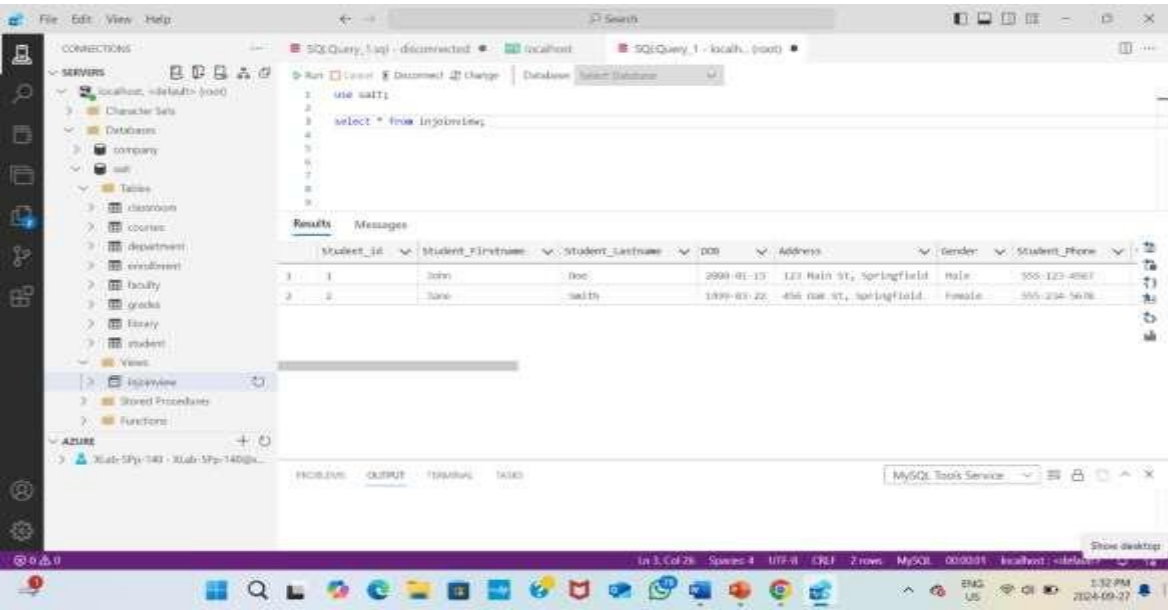
Results Messages

	Student_id	Student_firstname	Student_lastname	Student_email	Course_id	Course_name	Grade	Enrollment_date
1	1	John	Doe	john@example.com	1	Data Structures	A	
2	2	Jane	Smith	jane@example.com	2	Financial Management	B	

students. The use of views, queries, and a well-structured ERD further streamlines operations, benefiting both administrators and students alike.

Views

This view is created using an inner join query, which helps make data retrieval easier, saves time and effort, and ensures that the same logic and relationships between tables are maintained every time the view is applied.



Appendices

Appendix A: Database Tables and SQL Queries

This appendix provides detailed information about the tables used in the database system for student and course management at SAIT. Additionally, it includes sample SQL queries for testing and validating the relationships between the entities.

1. Tables

The database consists of the following tables:

Table Name	Description
Student	Stores information about students, including personal details and unique student IDs.
Course	Contains details about the courses offered, including course ID, course name, and department.

Department	Stores department details, such as department ID and department name.
Faculty	Contains faculty information, including faculty ID, name, and department affiliation.
Classroom	Includes details about classroom locations, such as room numbers and seating capacity.
Enrolment	Represents the enrollment of students in courses, linking the student ID and course ID.
Grades	Stores the grades students receive for their enrolled courses.
Process	Contains additional details regarding student processes or administrative actions.

2. Field Definitions

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)
Department_id	INT FK

Course Table:

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

Department Table:

Field Name	Data Type
------------	-----------

Department_id	INT PK
Department_name	VARCHAR (25)

Faculty Table:

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

Classroom Table:

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

Enrollment Table:

Field Name	Data Type
Enrollment_id	INT PK
Student_id	INT FK
Course_id	INT FK
Enrollment_date	DATE

Grades Table:

Field Name	Data Type
Grades_id	INT PK
Student_id	INT FK
Course_id	INT FK

Grade	VARCHAR (5)
-------	-------------

Process Table:

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

Appendix B: Test Queries

Test SQL Queries The provided test queries demonstrate practical applications of SAIT's DBMS, showcasing how to retrieve, manipulate, and display academic data for various purposes.

Test Queries

To demonstrate how SAIT's DBMS works, here are some sample SQL queries for data retrieval for student course grades.

SQL Query to Create the View

```
CREATE VIEW StudentCourseGrades AS
```

```
SELECT
```

```
    s.Student_id,
```

```
    s.Student_Name,
```

```
    d.Department_Name,
```

```
    c.Course_Name,
```

```
    f.Faculty_Name,
```

```
    g.Grade
```

```
FROM
```

```

    Enrollment e
INNER JOIN
    Student s ON e.Student_id = s.Student_id
INNER JOIN
    Courses c ON e.Course_id = c.Course_id
INNER JOIN
    Grades g ON e.Enrollment_id = g.Enrollment_id
INNER JOIN
    Department d ON s.Department_id = d.Department_id
INNER JOIN
    Faculty f ON c.Faculty_id = f.Faculty_id
ORDER BY
    s.Student_id, c.Course_Name;

```

Explanation of the Query

- **CREATE VIEW:** This statement creates a new view called StudentCourseGrades.
- **SELECT:** Specifies the columns to include in the view.
- **INNER JOINS:** Combines the tables based on their relationships:
 - Enrollment is the central table, linking students to their courses and grades.
 - Joins are made on the relevant fields to retrieve student details, course names, department names, faculty names, and grades.
- **ORDER BY:** The results are ordered first by Student_id and then by Course_Name.

Usage of the View

After creating the view, you can retrieve the combined information using a simple SELECT statement:

```
SELECT * FROM StudentCourseGrades;
```

It will show the course grade of the student in that particular course.

