

# **Database Design Document (DDD)**

## **Student Record Keeping Database**

### **Group**

Adam Growden, Hoi Hin Ng, Julia Parente, Sirtaj Khroud

## Table of Contents

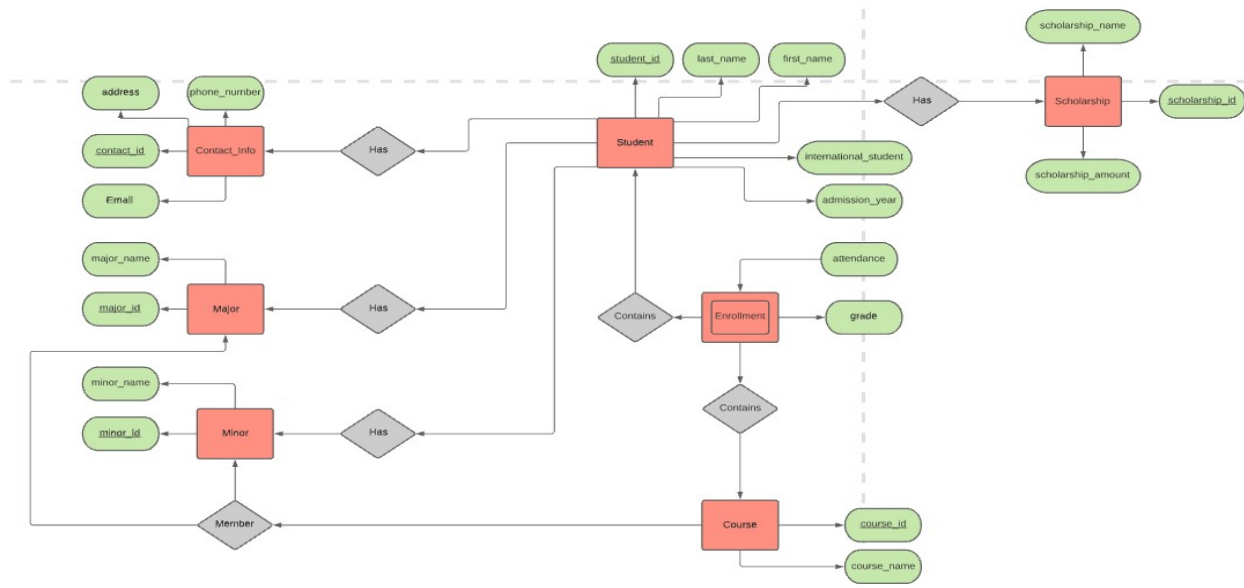
<b>1 Introduction .....</b>	<b>3</b>
<b>2 Detailed Database Design .....</b>	<b>3</b>
<b>2.1 Entity Relationship Diagram .....</b>	<b>3</b>
<b>2.2 Conceptual Diagram .....</b>	<b>4</b>
<b>2.3 Purpose of Tables .....</b>	<b>5</b>
2.3.1 Purpose of Student Table .....	5
2.3.2 Purpose of Contact_Info Table .....	6
2.3.3 Purpose of Course Table .....	7
2.3.4 Purpose of Enrollment Table .....	8
2.3.5 Purpose of Scholarship Table .....	9
2.3.6 Purpose of Major Table .....	9
2.3.7 Purpose of Minor Table .....	10
<b>3 Testing .....</b>	<b>10</b>

# 1 Introduction

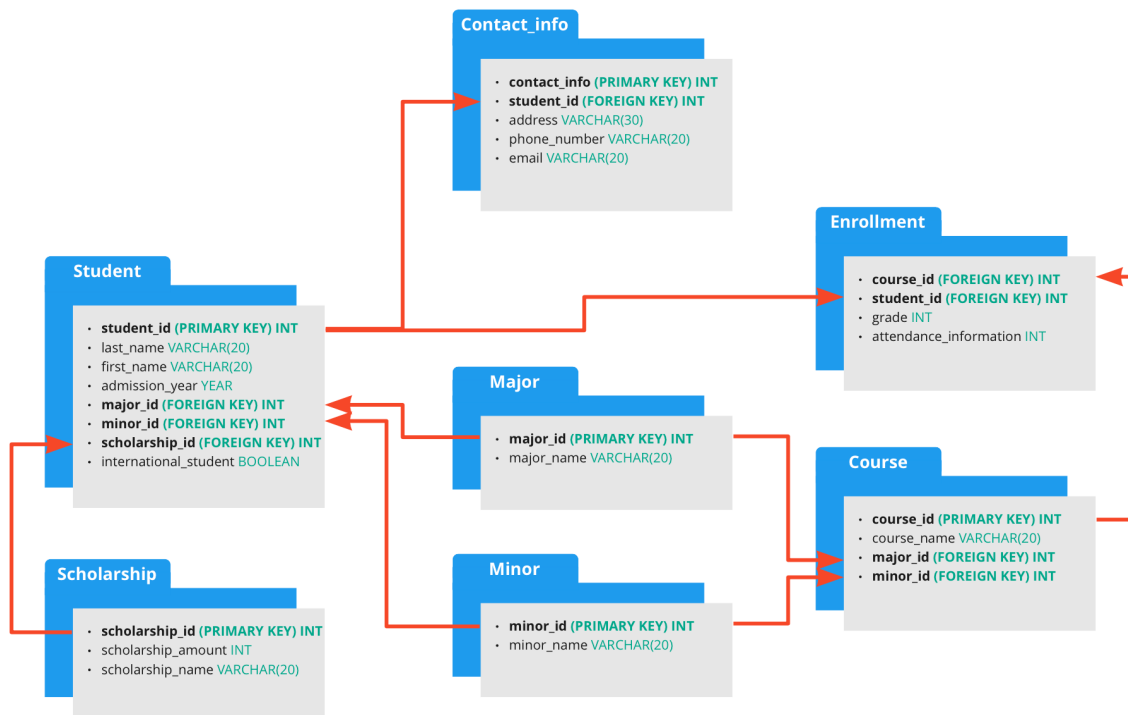
This Database Design Document (DDD) illustrates the report of the database design for the student record-keeping database. It contains the entity relationship and conceptual diagram of the database, description of each table, and SQL statements for implementing and testing the database. The 4 states of our database design process are requirements analysis, architecture design, implementation, and testing. For the project requirements, the database contains information for students, contact information, student enrollment (courses, grades, and attendance), scholarship information, course information, major and minor information. Based on this requirement, we designed the database with the entity relationship and conceptual diagram and constructed the database using SQL language statements. Finally, we stored the training data into the database and tested the database using SQL language statement testing tasks.

## 2 Detailed Database Designs

### 2.1 Entity Relationship Diagram



## 2.2 Conceptual Diagram



The diagram illustrates the conceptual model of the student record-keeping database. In this diagram, it contains the 7 tables which are the Student, Scholarship, Contact\_info, Enrollment, Course, Major, and Minor tables, and each table represents different roles for the database. The red arrows show how the tables connect and relate to each other by the primary and foreign keys.

## 2.3 Purpose of Tables

### 2.3.1 Purpose of Student Table

Student	
•	<b>student_id (PRIMARY KEY) INT</b>
•	last_name VARCHAR(20)
•	first_name VARCHAR(20)
•	admission_year YEAR
•	<b>major_id (FOREIGN KEY) INT</b>
•	<b>minor_id (FOREIGN KEY) INT</b>
•	<b>scholarship_id (FOREIGN KEY) INT</b>
•	international_student BOOLEAN

```
mysql> CREATE TABLE student (
    student_id int,
    last_name varchar (20),
    first_name varchar (20),
    major_id int,
    minor_id int,
    scholarship_id int,
    international_student boolean,
    admission_year year,
    PRIMARY KEY (student_id),
    FOREIGN KEY (major_id) REFERENCES major (major_id),
    FOREIGN KEY (minor_id) REFERENCES minor (minor_id),
    FOREIGN KEY (scholarship_id) REFERENCES scholarship
(scholarship_id),
);
```

Column Name	Description
student_id	student_id is the primary key for the student table which allows the identification of a unique table row. With the primary key you are able to query the student information specific to each individual.
last_name	last_name is of type varchar and stores a student's last name.
first_name	first_name is of type varchar and stores a student's first name.
admission_year	admission_year is of type year and stores the student's year of entry to the university.
major_id	major_id is a foreign key that represents the major of the student and references the major table. The major_id is an integer that corresponds to a major_id in the major table. This allows a subquery to the major table to retrieve information specific to that major. **If the major_id is NULL then the student does not have a major**
minor_id	minor_id is a foreign key that represents the minor of the student and references the minor table. The minor_id is an integer that corresponds to a minor_id in the minor table. This allows a subquery to the minor table to retrieve information specific to that minor. **If the minor_id is NULL then the student does not have a minor**
scholarship_id	scholarship_id is a foreign key that represents the scholarship the student has and references the scholarship table. The scholarship_id is an integer that corresponds to a scholarship_id in the scholarship table. This allows a subquery to the scholarship table to retrieve information specific to that scholarship **If the scholarship_id is NULL then the student does not have a scholarship**
international_student	international_student is of type Boolean and represents if the student is an international_student or not. If the value is true the student is an international student and if it is false the student is not.

### 2.3.2 Purpose of Contact\_Info Table

#### Contact\_info

- **contact\_info (PRIMARY KEY) INT**
- **student\_id (FOREIGN KEY) INT**
- address VARCHAR(30)
- phone\_number VARCHAR(20)
- email VARCHAR(20)

```
mysql> CREATE TABLE contact_info (
    contact_id int,
    student_id int,
    address varchar (20),
    phone_number varchar(20),
    email varchar (20)
    PRIMARY KEY(contact_id),
    FOREIGN KEY (student_id) REFERENCES student (student_id)
);
```

Column Name	Description
contact_info	contact_info is the primary key for the contact table which allows the identification of a unique table row. With the primary key you are able to query the contact information specific to an entity.
student_id	student_id is a foreign key that represents the student from the reference to the student table. student_id is an integer that corresponds to an student_id in the student table. This allows a subquery to the student table to retrieve information specific to the student. This use of the foreign key is also used to query the contact_info table to allow the selection of a specific student's contact information.
address	address is of type varchar and stores the address for a specific contact_info record.
phone_number	phone_number is of type varchar and stores the phone number of a specific contact_info record.
email	email is of type varchar and stores the email for a specific contact_info record.

### 2.3.3 Purpose of Course Table

Course
<ul style="list-style-type: none"> <li>• <b>course_id</b> (PRIMARY KEY) INT</li> <li>• course_name VARCHAR(20)</li> <li>• <b>major_id</b> (FOREIGN KEY) INT</li> <li>• <b>minor_id</b> (FOREIGN KEY) INT</li> </ul>

```
mysql> CREATE TABLE course (
    course_id int,
    course_name varchar(20),
    major_id int,
    minor_id int
    PRIMARY KEY (course_id),
    FOREIGN KEY (major_id) REFERENCES major (major_id),
    FOREIGN KEY (minor_id) REFERENCES minor (minor_id)
);
```

Column Name	Description
course_id	course_id is the primary key for the course table which allows the identification of a unique table row. With the primary key you are able to query course information specific to each course.
course_name	Course_name is of type varchar and stores the course name of the specific course record.
major_id	major_id is a foreign key that represents the major the course belongs to and references the major table. The major_id is an integer that corresponds to a major_id in the major table. This allows a subquery to the major table to retrieve information specific to that major. **If the major_id is NULL then the course does not belong to a major**
minor_id	minor_id is a foreign key that represents the minor the course belongs to and references the minor table. The minor_id is an integer that corresponds to a minor_id in the minor table. This allows a subquery to the minor table to retrieve information specific to that minor. **If the minor_id is NULL then the course does not belong to a minor**

### 2.3.4 Purpose of Enrollment Table

#### Enrollment

- **course\_id** (FOREIGN KEY) INT
- **student\_id** (FOREIGN KEY) INT
- grade INT
- attendance\_information INT

```
mysql> CREATE TABLE enrollment (
    course_id int,
    student_id int,
    grade int,
    attendance_information int
    PRIMARY KEY (course_id, student_id),
    FOREIGN KEY (course_id) REFERENCES course (course_id),
    FOREIGN KEY (student_id) REFERENCES student (student_id)
);
```

Column Name	Description
course_id	course_id is a foreign key that references the course table which allows a subquery of the course table to get course specific information. The primary key for the enrollment table is comprised of the two foreign keys, course_id and student_id which allows the query of a specific enrollment record.
student_id	student_id is a foreign key that references the student table which allows a subquery of the student table to get student specific information. The primary key for the enrollment table is comprised of the two foreign keys, course_id and student_id which allows the query of a specific enrollment record.
grade	grade is of type int and stores the grade information for an enrollment record.
attendance_information	attendance_information is of type int and stores the attendance information for an enrollment record.



### 2.3.5 Purpose of Scholarship Table

#### Scholarship

- scholarship\_id (PRIMARY KEY) INT
- scholarship\_amount INT
- scholarship\_name VARCHAR(20)

```
mysql> CREATE TABLE scholarship (
    scholarship_id int,
    scholarship_amount int,
    scholarship_name varchar(20),
    PRIMARY KEY (scholarship_id)
);
```

Column Name	Description
scholarship_id	scholarship_id is the primary key for the scholarship table which allows the identification of a unique table row. Each scholarship has a unique scholarship_id to define the scholarship amount and name.
scholarship_amount	scholarship_amount is of type int and stores the scholarship amount for a scholarship record.
scholarship_name	scholarship_name is of type varchar and stores the scholarship name for a scholarship record.

### 2.3.6 Purpose of Major Table

#### Major

- major\_id (PRIMARY KEY) INT
- major\_name VARCHAR(20)

```
mysql> CREATE TABLE major (
    major_id int,
    major_name varchar (20),
    PRIMARY KEY (major_id)
);
```

Column Name	Description
major_id	major_id is the primary key for the major table which allows the identification of a unique table row. With the primary key you are able to query major information specific to each major.
major_name	major_name is of type varchar and stores the name of the major for the specific major record. Each major_name corresponds to its own major_id such as the major_id 1 corresponds to the major_name of Computer Science.  E.g., (major_id, major_name) (1, "Computer Science") (2, "Business Administration")

### 2.3.7 Purpose of Minor Table

#### Minor

- **minor\_id** (PRIMARY KEY) INT
- minor\_name VARCHAR(20)

```
mysql> CREATE TABLE minor (
    minor_id int,
    minor_name varchar (20),
    PRIMARY KEY (minor_id)
);
```

Column Name	Description
minor_id	minor_id is the primary key for the minor table which allows the identification of a unique table row. With the primary key you are able to query minor information specific to each minor.
minor_name	<p>minor_name is of type varchar and stores the name of the minor for the specific minor record Each minor_name corresponds to its own minor_id such as the minor_id 3 corresponds to the minor_name of User Experience Design.</p> <p>E.g., (minor_id, minor_name)            (1, "Computer Science")            (2, "Business Administration")            (3, "User Experience Design")</p>

## 3 Testing

**SELECT all students who are in computer science major**

```
mysql> SELECT student_id, last_name, first_name
FROM student
WHERE major_id IN (SELECT major_id
FROM major
WHERE major_name = "Computer Science" );
```

```
mysql> use student_record_db
Database changed
mysql> SELECT student_id, last_name, first_name
-> FROM student
-> WHERE major_id IN (SELECT major_id
-> FROM major
-> WHERE major_name = "Computer Science" );
```

student_id	last_name	first_name
1	James	Smith
3	John	Miller
5	William	Lopez
6	David	Brown
7	Richard	Thomas
8	Joseph	Moore

```
6 rows in set (0.00 sec)
```

### SELECT all students in the course Interpersonal communication

```
Mysql> SELECT s.student_id, s.last_name, s.first_name, c.course_name
        FROM student s, course c, enrollment e
        WHERE c.course_name = "Interpersonal Communication" AND c.course_id =
              e.course_id AND s.student_id = e.student_id ;
```

```
mysql> SELECT s.student_id, s.last_name, s.first_name, c.course_name
-> FROM student s, course c, enrollment e
-> WHERE c.course_name = "Interpersonal Communication" AND c.course_id = e.course_id AND s.student_id = e.student_id ;
```

student_id	last_name	first_name	course_name
6	David	Brown	Interpersonal Communication
9	Thomas	Taylor	Interpersonal Communication
10	Charles	Lee	Interpersonal Communication

3 rows in set (0.00 sec)

### SELECT all students with a scholarship and the amount

```
Mysql> SELECT s.student_id, s.last_name, s.first_name, sc.scholarship_amount,
        sc.scholarship_name
        FROM student s, scholarship sc
        WHERE s.scholarship_id = sc.scholarship_id;
```

```
mysql> SELECT s.student_id, s.last_name, s.first_name, sc.scholarship_amount, sc.scholarship_name
-> FROM student s, scholarship sc
-> WHERE s.scholarship_id = sc.scholarship_id;
```

student_id	last_name	first_name	scholarship_amount	scholarship_name
5	William	Lopez	500	Steve Adams Scholarship
4	Michael	Davis	1000	Mike Daniles Scholarship
10	Charles	Lee	1000	Mike Daniles Scholarship
6	David	Brown	2000	Mark Zuckerberg Scholarship
1	James	Smith	3000	Wilfrid Laurier Scholarship

5 rows in set (0.00 sec)

## SELECT all students and join their contact info

```
mysql> SELECT student.student_id, student.last_name, student.first_name,
contact_info.address, contact_info.email, contact_info.phone_number
FROM student
INNER JOIN contact_info ON student.student_id = contact_info.student_id ;
```

```
mysql> SELECT student.student_id, student.last_name, student.first_name, contact_info.address, contact_info.email,
-> contact_info.phone_number
-> FROM student
-> INNER JOIN contact_info ON student.student_id = contact_info.student_id ;
```

student_id	last_name	first_name	address	email	phone_number
1	James	Smith	88 jumper st. , Toronto, On. ON N23 1X5	j_smith@mylaurier.ca	196-946-7314
2	Robert	Jones	Line 49, Brunner, ON N0K 1C0	r_jones@mylaurier.ca	236-303-9698
3	John	Miller	4712 Line 67, Milverton, ON N0K 1M0	j_miller@mylaurier.ca	242-999-3952
4	Michael	Davis	16 Mill St W, Milverton, ON N0K 1M0	m_davis@mylaurier.ca	300-890-5713
5	William	Lopez	1206 Snyder's Rd W, New Hamburg, ON N3A 1A4	w_lopez@mylaurier.ca	653-305-2684
6	David	Brown	560 Snyder's Rd E, Baden, ON N3A 3L1	d_brown@mylaurier.ca	548-551-1321
7	Richard	Thomas	751 Pasture Rose St, Waterloo, ON N2V 0C2	r_tho@mylaurier.ca	226-998-0090
8	Joseph	Moore	710 Laurelwood Dr, Waterloo, ON N2V 2V3	j_moo@mylaurier.ca	213-599-1010
9	Thomas	Taylor	200 Ring Rd, Waterloo, ON N2L 3G1	taylor@mylaurier.ca	876-443-1010
10	Charles	Lee	719 Erbsville Rd, Waterloo, ON N2J 3Z4	lee@mylaurier.ca	718-858-9876

10 rows in set (0.00 sec)

## SELECT all students with a major and minor

```
mysql> SELECT s.student_id, s.last_name, s.first_name, m.major_name, mi.minor_name
FROM student s, major m, minor mi
WHERE s.major_id = m.major_id AND s.minor_id = mi.minor_id;
```

```
mysql> SELECT s.student_id, s.last_name, s.first_name, m.major_name, mi.minor_name
-> FROM student s, major m, minor mi
-> WHERE s.major_id = m.major_id AND s.minor_id = mi.minor_id;
```

student_id	last_name	first_name	major_name	minor_name
9	Thomas	Taylor	Business Administration	Computer Science
7	Richard	Thomas	Computer Science	Business Administration
6	David	Brown	Computer Science	Business Administration
3	John	Miller	Computer Science	Business Administration
2	Robert	Jones	Business Administration	User Experience Design
8	Joseph	Moore	Computer Science	User Experience Design

6 rows in set (0.00 sec)



## SELECT all students, there courses and grades + attendance in each course

```
mysql> SELECT s.student_id, s.last_name, s.first_name, c.course_id, c.course_name, e.grade,
e.attendance_information
```

```
mysql> SELECT s.student_id, s.last_name, s.first_name, c.course_id, c.course_name, e.grade, e.attendance_information
-> FROM student s, course c, enrollment e
-> WHERE c.course_id = e.course_id AND s.student_id = e.student_id ;
```

student_id	last_name	first_name	course_id	course_name	grade	attendance_information
1	James	Smith	5	Windows App. Programming	71	73
1	James	Smith	8	Intro to Microprocessors	47	95
1	James	Smith	9	Digital Electronics	77	92
1	James	Smith	12	Software Engineering	90	100
2	Robert	Jones	11	Algorithm Design/Analysis I	58	58
2	Robert	Jones	18	Fundamentals of Finance	78	76
2	Robert	Jones	20	Business Law	69	81
2	Robert	Jones	24	Interaction Design I	96	64
2	Robert	Jones	25	Design of Immersive Spaces	96	86
3	John	Miller	5	Windows App. Programming	72	67
3	John	Miller	14	Physical Computing	42	90
3	John	Miller	19	Fundamentals of Operations	77	85
4	Michael	Davis	1	Info Processing With Micros	91	46
4	Michael	Davis	15	Understanding Bus. Environment	86	81
4	Michael	Davis	22	Design Thinking I: Foundations	62	42
5	William	Lopez	7	Discrete Struct for Comp Sci	48	97
5	William	Lopez	8	Intro to Microprocessors	77	46
5	William	Lopez	9	Digital Electronics	70	50
5	William	Lopez	10	Data Structures II	83	61
6	David	Brown	13	Digital System Design	98	87
6	David	Brown	14	Physical Computing	80	72
6	David	Brown	16	Intro to Financial Accounting	64	88
6	David	Brown	17	Interpersonal Communication	64	66
7	Richard	Thomas	4	Web Site Design	42	83
7	Richard	Thomas	5	Windows App. Programming	48	46
7	Richard	Thomas	20	Business Law	89	70
8	Joseph	Moore	1	Info Processing With Micros	91	88
8	Joseph	Moore	3	Data Structures I	79	80
8	Joseph	Moore	4	Web Site Design	42	62
8	Joseph	Moore	23	Information Design	88	44
8	Joseph	Moore	25	Design of Immersive Spaces	65	87
9	Thomas	Taylor	17	Interpersonal Communication	82	99
9	Thomas	Taylor	23	Information Design	45	74
9	Thomas	Taylor	24	Interaction Design I	88	49
10	Charles	Lee	15	Understanding Bus. Environment	48	54
10	Charles	Lee	16	Intro to Financial Accounting	92	93
10	Charles	Lee	17	Interpersonal Communication	64	57
10	Charles	Lee	18	Fundamentals of Finance	73	43
10	Charles	Lee	19	Fundamentals of Operations	72	61

```
39 rows in set (0.00 sec)
```

**SELECT all students with an admission year before 2020**

```
mysql> SELECT student_id, last_name, first_name, admission_year  
FROM student  
WHERE admission_year < 2020;
```

```
mysql> SELECT student_id, last_name, first_name, admission_year  
-> FROM student  
-> WHERE admission_year < 2020;
```

student_id	last_name	first_name	admission_year
1	James	Smith	2019
2	Robert	Jones	2017
3	John	Miller	2019
4	Michael	Davis	2018
5	William	Lopez	2016

```
5 rows in set (0.00 sec)
```