# **University Scholarship Management System Final Project Report**

Group Members: Chukwuebuka Ernest-Opara, Malavkumar Patel, Sagun Raj Lage, Sai Sravanthi Racharla, Vishal Kumar Yadav

**Introduction**

The purpose of this project is to develop a database for a university scholarship management system. The major users of the database include students, departments of different educational fields and scholarship sponsors.

The database of the system was built using MySQL, and Sai Sravanthi Racharla and Sagun Raj Lage took the initiative of creating the tables in the database. Then, Chukwuebuka Ernest-Opara, Malavkumar Patel and Vishal Kumar Yadav worked jointly to enter data into the tables and populate them while also making sure that the constraints are correctly created between the tables.

**Statements for creating the tables**

The statements for creating the tables are given below:

CREATE TABLE DEPARTMENT (

department\_id INT PRIMARY KEY,

name VARCHAR(255) NOT NULL UNIQUE,

phone VARCHAR(15),

chairman VARCHAR(255),

email VARCHAR(255) UNIQUE,

website VARCHAR(255)

);

CREATE TABLE MAJOR (

major\_id INT PRIMARY KEY,

name VARCHAR(255) NOT NULL UNIQUE,

duration INT,

accreditation VARCHAR(255),

department\_id INT,

FOREIGN KEY (department\_id) REFERENCES DEPARTMENT(department\_id)

);

CREATE TABLE SCHOLARSHIP (

scholarship\_id INT PRIMARY KEY,

name VARCHAR(255),

criteria VARCHAR(255),

amount DECIMAL(10, 2),

department\_id INT,

FOREIGN KEY (department\_id) REFERENCES DEPARTMENT(department\_id)

);

CREATE TABLE SPONSOR (

sponsor\_id INT PRIMARY KEY,

name VARCHAR(255) NOT NULL UNIQUE,

phone VARCHAR(15),

email VARCHAR(255) UNIQUE,

type VARCHAR(255),

address VARCHAR(255)

);

CREATE TABLE STUDENT (

student\_id INT PRIMARY KEY,

first\_name VARCHAR(60) NOT NULL,

last\_name VARCHAR(60) NOT NULL,

level VARCHAR(50),

date\_of\_birth DATE NOT NULL,

address VARCHAR(255),

country VARCHAR(50),

gpa DECIMAL(4,2),

major\_id INT,

scholarship\_id INT,

FOREIGN KEY (date\_of\_birth) REFERENCES STUDENT\_AGE(date\_of\_birth),

FOREIGN KEY (major\_id) REFERENCES MAJOR(major\_id),

FOREIGN KEY (scholarship\_id) REFERENCES SCHOLARSHIP(scholarship\_id)

);

CREATE TABLE SCHOLARSHIP\_SPONSOR (

scholarship\_id INT,

sponsor\_id INT,

PRIMARY KEY (scholarship\_id, sponsor\_id),

FOREIGN KEY (scholarship\_id) REFERENCES SCHOLARSHIP(scholarship\_id),

FOREIGN KEY (sponsor\_id) REFERENCES SPONSOR(sponsor\_id)

);

CREATE TABLE STUDENT\_AGE (

date\_of\_birth DATE NOT NULL,

age INT NOT NULL,

PRIMARY KEY (date\_of\_birth)

);

**Schema diagram**

A screenshot of a computer

Description automatically generatedThe schema diagram of the database after using the statements mentioned above is given below:

Figure 1: Schema diagram of the University Scholarship Management System

**Statements for inserting data into the tables**

The statements for inserting data into the tables are given below:

INSERT INTO `DEPARTMENT` VALUES (1,'Computer Science','(123) 456-7890','Dr. John Smith','cs@university.edu','www.cs.university.edu'),(2,'Engineering','(987) 654-3210','Dr. Jane Doe','engineering@university.edu','www.engineering.university.edu'),(3,'Business','(555) 123-4567','Dr. Peter Jones','business@university.edu','www.business.university.edu'),(4,'Mathematics','(321) 987-6543','Dr. Mary Brown','math@university.edu','www.math.university.edu'),(5,'Art','(789) 456-1230','Dr. David Miller','art@university.edu','www.art.university.edu'),(6,'Music','(654) 321-9870','Dr. Susan Williams','music@university.edu','www.music.university.edu'),(7,'History','(123) 987-6540','Dr. Paul Robinson','history@university.edu','www.history.university.edu'),(8,'Nursing','(888) 456-7890','Dr. Emily Johnson','nursing@university.edu','www.nursing.university.edu');

INSERT INTO `MAJOR` VALUES (1,'Computer Science',4,'ABET',1),(2,'Software Engineering',4,'ABET',1),(3,'Electrical Engineering',4,'ABET',2),(4,'Mechanical Engineering',4,'ABET',2),(5,'Business Administration',4,'AACSB',3),(6,'Accounting',4,'AACSB',3),(7,'Finance',4,'AACSB',3),(8,'Nursing',4,'CCNE',8);

INSERT INTO `SCHOLARSHIP` VALUES (1,'Presidential Scholarship','High GPA and academic achievement',10000.00,1),(2,'Dean\"s Scholarship','High GPA and extracurricular involvement',5000.00,1),(3,'Engineering Scholarship','Strong interest in engineering and math',5000.00,2),(4,'Business Scholarship','Strong interest in business and economics',5000.00,3),(5,'Art Scholarship','Talented artist with a strong portfolio',5000.00,5),(6,'Music Scholarship','Talented musician with a strong musical background',5000.00,6),(7,'History Scholarship','Strong interest in history and a passion for research',5000.00,7),(8,'Nursing Scholarship','Strong interest in nursing and a passion for caring for others',5000.00,8);

INSERT INTO `SPONSOR` VALUES (1,'Microsoft','(425) 555-1212','sponsorship@microsoft.com','Corporate','One Microsoft Way, Redmond, WA 98052'),(2,'Google','(650) 253-0000','sponsorship@google.com','Corporate','1600 Amphitheatre Pkwy, Mountain View, CA 94043'),(3,'Amazon','(206) 266-2992','sponsorship@amazon.com','Corporate','410 Terry Ave N, Seattle, WA 98109'),(4,'Wells Fargo','(415) 353-2500','sponsorship@wellsfargo.com','Corporate','420 Montgomery St, San Francisco, CA 94104'),(5,'Bank of America','(704) 336-6222','sponsorship@bankofamerica.com','Corporate','100 Nigel Morris Ave, Newark, NJ 07123'),(6,'Intuit','(445) 525-1342','sponsorship@intuit.com','Corporate','One Intuit Way, Houston, TX 77009'),(7,'Sabor','(432) 473-8400','sponsorship@sabor.com','Corporate','1 Sabor Pkwy, Richardson, TX 77327'),(8,'Johnson & Johnson','(908) 216-7300','sponsorship@jnj.com','Corporate','1 Johnson & Johnson Plaza New Brunswick, NJ 08933');

INSERT INTO `STUDENT\_AGE` VALUES ('1989-12-31',33),('1990-01-01',33),('1991-02-02',32),('1992-03-03',31),('1993-04-04',30),('1994-05-05',29),('1995-06-06',28),('1996-07-07',27);

INSERT INTO `STUDENT` VALUES (1,'John','Doe','Junior','1990-01-01','123 Main St., Anytown, USA','USA',3.80,1,1),(2,'Jane','Smith','Sophomore','1991-02-02','456 Elm St., Anytown, USA','USA',3.70,2,2),(3,'Peter','Jones','Freshman','1992-03-03','789 Oak St., Anytown, USA','USA',3.60,3,3),(4,'Mary','Brown','Junior','1993-04-04','1011 Maple St., Anytown, USA','USA',3.90,4,4),(5,'David','Miller','Sophomore','1994-05-05','1213 Pine St., Anytown, USA','USA',3.50,5,5),(6,'Susan','Williams','Freshman','1995-06-06','1315 Elm St., Anytown, USA','USA',3.40,6,6),(7,'Paul','Robinson','Junior','1996-07-07','1417 Oak St., Anytown, USA','USA',3.30,7,7),(8,'Sarah','Thompson','Senior','1989-12-31','1519 Maple St., Anytown, USA','USA',4.00,8,8);

INSERT INTO `SCHOLARSHIP\_SPONSOR` VALUES (1,1),(2,1),(3,2),(4,3),(5,4),(6,5),(7,2),(7,6);

**Queries that show all data in the tables**

The queries that show all data in the tables are given below with the respective screenshots of the results:

SELECT \* FROM DEPARTMENT;

A screenshot of a computer

Description automatically generated

SELECT \* FROM MAJOR;

A table with text on it

Description automatically generated

SELECT \* FROM SCHOLARSHIP;

A screenshot of a computer

Description automatically generated

SELECT \* FROM SPONSOR;

A screenshot of a computer

Description automatically generated

SELECT \* FROM STUDENT;

A screenshot of a phone number

Description automatically generated

SELECT \* FROM SCHOLARSHIP\_SPONSOR;

A screenshot of a computer

Description automatically generated

SELECT \* FROM STUDENT\_AGE;

A screenshot of a data

Description automatically generated

**A query using Column Aliases**

A query using column alias is given below with its result:

SELECT student\_id AS ID, first\_name, last\_name FROM STUDENT;

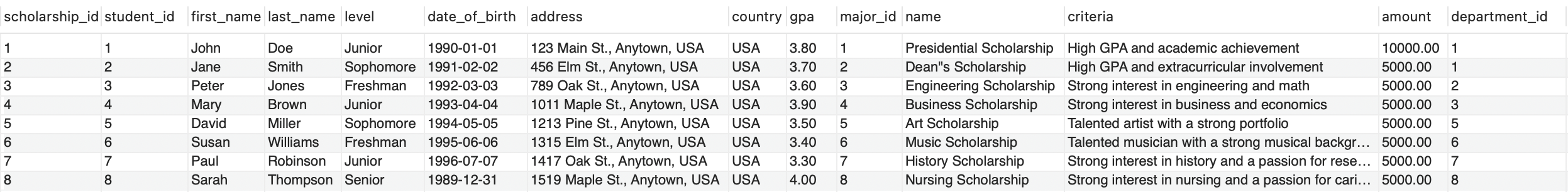
A screenshot of a computer

Description automatically generated

**A query using Natural Join**

A query using natural join is given below with its result:

SELECT \* FROM STUDENT NATURAL JOIN SCHOLARSHIP;



**A query using the JOIN USING syntax**

A query using JOIN USING syntax is given below with its result:

SELECT \* FROM SCHOLARSHIP JOIN DEPARTMENT USING (department\_id);

A screenshot of a computer

Description automatically generated

**A query using the JOIN ON syntax**

A query using JOIN ON syntax is given below with its result:

SELECT student\_id, first\_name, last\_name, STUDENT.major\_id, MAJOR.name FROM STUDENT JOIN MAJOR ON STUDENT.major\_id = MAJOR.major\_id;

A screenshot of a computer

Description automatically generated

**A query using the ORDER BY clause**

A query using ORDER BY clause is given below with its result:

SELECT \* FROM STUDENT ORDER BY date\_of\_birth DESC;

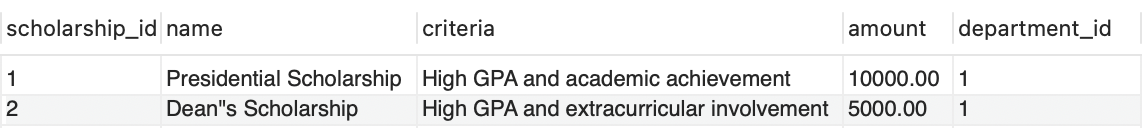
A screenshot of a computer

Description automatically generated

**A query using the WHERE clause to retrieve a specific record**

A query using the WHERE clause to retrieve a specific record is given below with its result:

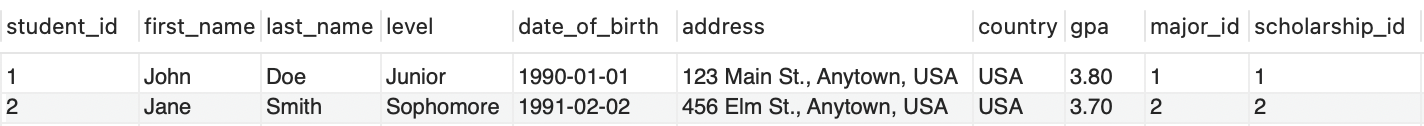
SELECT \* FROM SCHOLARSHIP WHERE department\_id = 1;



**A query using the WHERE clause**

A query using the WHERE clause is given below with its result:

SELECT \* FROM STUDENT WHERE first\_name LIKE "J%";



**Two queries using the logical operators: AND, OR**

A query using the logical operator AND is given below with its result:

SELECT \* FROM MAJOR WHERE accreditation = "AACSB" AND department\_id = 3;

A screenshot of a computer

Description automatically generated

A query using the logical operator OR is given below with its result:

SELECT \* FROM MAJOR WHERE accreditation = "CCNE" OR accreditation = "AACSB";

A screenshot of a computer

Description automatically generated

**A query using the logical operators: AND, OR combined**

A query using the logical operators AND and OR combined is given below with its result:

SELECT \* FROM STUDENT WHERE (gpa >= 3.0 AND gpa <= 3.50) OR level = "Senior";

A screenshot of a phone

Description automatically generated

**A query using the LIKE special operator**

A query using the LIKE special operator is given below with its result:

SELECT \* FROM MAJOR WHERE name LIKE "%Eng%";

A screenshot of a computer

Description automatically generated

**A query using the GROUP BY operator**

A query using the GROUP BY operator is given below with its result:

SELECT accreditation, COUNT(major\_id) AS majors\_count FROM MAJOR GROUP BY accreditation;

A screenshot of a cell phone

Description automatically generated

**A query using the GROUP BY operator and the ORDER BY operator**

A query using the GROUP BY operator and the ORDER BY operator is given below with its result:

SELECT accreditation, COUNT(major\_id) AS majors\_count FROM MAJOR GROUP BY accreditation ORDER BY majors\_count;

A screenshot of a cell phone

Description automatically generated

**A query using the WHERE subqueries**

A query using the WHERE subqueries is given below with its result:

SELECT first\_name, last\_name, gpa FROM STUDENT WHERE gpa <= (SELECT AVG(gpa) FROM STUDENT);

A screenshot of a computer

Description automatically generated

**A query using the DISTINCT and WHERE column\_name IN subqueries**

A query using the DISTINCT and WHERE column\_name IN subqueries is given below with its result:

SELECT DISTINCT first\_name, last\_name, major\_id

FROM STUDENT

WHERE major\_id IN (

SELECT major\_id

FROM MAJOR

WHERE department\_id IN (

SELECT department\_id

FROM DEPARTMENT

WHERE name = 'Computer Science'

)

);

A screenshot of a computer

Description automatically generated