**University Scholarship Management System**

**Normalization and Tables creation**

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

**Entities and Attributes:**The existing entities in the system along with their attributes are listed below.

It's important to note that while there can be multiple candidate keys in a table, only one of them is selected as the primary key.

* **STUDENT:**

|  |  |
| --- | --- |
| **Attributes** | **Keys** |
| student\_id | **Primary Key** |
| first\_name |  |
| last\_name |  |
| level |  |
| date\_of\_birth |  |
| age |  |
| address |  |
| country |  |
| gpa |  |
| major\_id | Foreign Key, referencing MAJOR |
| scholarship\_id | Foreign Key, referencing SCHOLARSHIP |

* **MAJOR:**

|  |  |
| --- | --- |
| **Attributes** | **Keys** |
| major\_id | **Primary Key** |
| name | Candidate Key |
| duration |  |
| accreditation |  |
| department\_id | Foreign Key, referencing DEPARTMENT |

* **DEPARTMENT:**

|  |  |
| --- | --- |
| **Attributes** | **Keys** |
| department\_id | **Primary Key** |
| name | Candidate Key |
| phone |  |
| chairman |  |
| email | Candidate Key |
| website |  |

* **SCHOLARSHIP:**

|  |  |
| --- | --- |
| **Attributes** | **Keys** |
| scholarship\_id | **Primary Key** |
| name |  |
| criteria |  |
| amount |  |
| department\_id | Foreign Key, referencing DEPARTMENT |

* **SPONSOR:**

|  |  |
| --- | --- |
| **Attributes** | **Keys** |
| sponsor\_id | **Primary Key** |
| name | Candidate Key |
| phone |  |
| email | Candidate Key |
| type |  |
| scholarship\_id | Foreign Key, referencing SCHOLARSHIP |

* **SCHOLARSHIP\_SPONSOR:** The **bridge table** which is used to resolve the many-to-many M: N relationship between SCHOLARSHIP and SPONSOR tables.

|  |  |
| --- | --- |
| **Attributes** | **Keys** |
| scholarship\_id, sponsor\_id | **Composite Primary Key** |

**Normalizing the Tables:**

**First Normal Form (1NF):** All tables are already in 1NF since all attributes have atomic values and each table has a primary key.

**Second Normal Form (2NF):** For a table to be in 2NF:

* It should be in 1NF.
* It should not have any partial dependency, i.e., non-prime attributes (attributes that are not part of any candidate key) should be fully functionally dependent on the primary key.

In the above tables, there exists no partial dependencies between attributes and are in 2NF.

**Third Normal Form (3NF):** For a table to be in 3NF:

* It should be in 2NF.
* It should not have transitive dependencies, i.e., non-prime attributes should not depend on other non-prime attributes.

In the **STUDENT** table, **age** and **date\_of\_birth** are transitive as date\_of\_birth determines age.

**date\_of\_birth**

**level**

**name**

**address**

**gpa**

**age**

**student\_id**

So we break the **STUDENT** table as follows:

**address**

**gpa**

**date\_of\_birth**

**level**

**name**

**student\_id**

The attribute date\_of\_birth becomes the primary\_key in the second table **STUDENT\_AGE.**  
Now, date\_of\_birth 🡪 age.

**age**

**date\_of\_birth**

**Boyce-Codd Normal Form (BCNF):** It is a higher version of the Third Normal Form (3NF). For a table to be in BCNF:

* It should be in 3NF.
* For every non-trivial functional dependency (X 🡪 Y), X is a super key. A non-trivial functional dependency means that Y is not a subset of X.

Examining the tables if they are in BCNF.

**STUDENT Table**:

* All attributes are functionally dependent on **student\_id**. There doesn't seem to be any non-trivial functional dependency where the left-hand side isn't a super key.   
  This table appears to be in BCNF.

**MAJOR Table**:

* Candidate Keys: **major\_id, name**
* All attributes are functionally dependent on **major\_id**. No apparent violations here, so this table is in BCNF.

**DEPARTMENT Table**:

* Candidate Keys: **department\_id, name, email**
* All the attributes depend on **department\_id**, and none of them depends on name or email in such a way that would violate BCNF. Therefore, this table is in BCNF.

**SCHOLARSHIP Table**:

* Candidate Key: **scholarship\_id**
* All attributes are functionally dependent on **scholarship\_id**. This table is in BCNF.

**SPONSOR Table**:

* Candidate Keys: **sponsor\_id, name, email**
* Each attribute depends on **sponsor\_id**. The table is in BCNF.

**SCHOLARSHIP\_SPONSOR Table**:

* Candidate Key: **scholarship\_id, sponsor\_id** (composite key)
* his is a bridge table to represent a many-to-many relationship. Usually, these tables are in BCNF since the composite key determines every other attribute (if there are any other attributes). In this case, there aren't any other attributes, so it's in BCNF.

**STUDENT\_AGE** **Table:**

* Candidate Key: **date\_of\_birth**
* The attribute age is functionally dependent on date\_of\_birth. So this table is in BCNF.

**Tables creation:**

We are using MySQL relational database to build the tables.

*Note: While building the tables, create the child tables first before creating the parent tables as there are foreign key relationships between the tables.*

Database systems enforce referential integrity, which means that a foreign key value in a table must match a primary key value in another table. Therefore, the referenced table (the one holding the primary key) must exist before you can define a foreign key in another table.

Below are the MySQL queries used for creating the database and tables in MySQL Workbench:

**-- Create project DATABASE**   
create database scholarship\_management;  
use scholarship\_management;

**-- DEPARTMENT Table**

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)

);

**-- MAJOR Table**

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)

);

**-- SCHOLARSHIP Table**

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)

);

**-- SPONSOR Table**

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)

);

**-- SCHOLARSHIP\_SPONSOR Table**

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)

);

**-- STUDENT\_AGE**

CREATE TABLE STUDENT\_AGE (

date\_of\_birth DATE NOT NULL,

age INT NOT NULL,

PRIMARY KEY (date\_of\_birth)

);

**-- STUDENT**

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)

);

A screenshot of a computer

Description automatically generated  
 *Fig 1: Snapshot of the tables in MySQL*

**Reverse Engineering:**

MySQL Workbench provides a visual way to design and view the database schema.

* Connect to your database.
* Go to the "Database" menu.
* Choose "Reverse Engineer".
* Follow the wizard. Once complete, you'll be presented with an ERD of your database.

**A screenshot of a computer

Description automatically generated** *Fig 2: ERD generated by MySQL Workbench from the database schema.*

This visual representation can be very handy, especially when working on complex databases, as it provides a clear overview of how tables are structured and related.