

Name: S.K Udara

Student Reference Number: 10749130

Module Code: PUSL2019

Module Name: Information Management & Retrieval

Coursework Title:  
Student Management System

Deadline Date: 24<sup>nd</sup> Dec 2021

Member of staff responsible for coursework:  
Kevin Samarasinghe

Programme: Bsc (Hons) Software Engineering

Please note that University Academic Regulations are available under Rules and Regulations on the University website [www.plymouth.ac.uk/studenthandbook](http://www.plymouth.ac.uk/studenthandbook).

Group work: please list all names of all participants formally associated with this work and state whether the work was undertaken alone or as part of a team. Please note you may be required to identify individual responsibility for component parts.

- Project and Group Leader – S K Udara (10749130)
- Planning Leader – K K D B Gimhana (10749170)
- Technical Leader – M M S P Marasinghe (10750062)
- Programming Leader – R S Rashmika (10749135)
- Quality Leader – W M M Fernando (10749140)
- Testing and Maintenance Leader – W P Y U Perera (10749131)

***We confirm that we have read and understood the Plymouth University regulations relating to Assessment Offences and that we are aware of the possible penalties for any breach of these regulations. We confirm that this is the independent work of the group.***

Signed on behalf of the group:



Individual assignment: ***I confirm that I have read and understood the Plymouth University regulations relating to Assessment Offences and that I am aware of the possible penalties for any breach of these regulations. I confirm that this is my own independent work.***

Signed :



Use of translation software: failure to declare that translation software or a similar writing aid has been used will be treated as an assessment offence.

I \*have used/not used translation software.

If used, please state name of software.....

Overall mark \_\_\_\_\_ % Assessors Initials \_\_\_\_\_ Date \_\_\_\_\_

# Table of Contents

Table of Contents.....	2
SECTION 01 .....	3
Introduction .....	3
Extended Entity Relationship Diagram.....	4
Diagram .....	4
Assumptions .....	5
Relational Mapping .....	6
Mapping of Strong Entities.....	6
Mapping of Weak Entities .....	6
Mapping of Binary 1:1 Relationships .....	7
Mapping of Binary 1:M Relationships .....	7
Mapping of Binary M:N Relationships.....	8
Mapping of Multi-Valued Attributes.....	9
Mapping of N-ary Relationships.....	10
Mapping of Recursive Relationships.....	10
Mapping of Extended-Entity Relationships .....	11
Normalization.....	12
Dependencies.....	12
1st Normal Form (1NF).....	13
2nd Normal Form (2NF).....	13
3rd Normal Form (3NF).....	13
Data Dictionary .....	14
SECTION 02 .....	17
Create Table Statements .....	17
Database Diagram .....	21
Database Sample Records .....	22
SECTION 03.....	28
Create Trigger Statements.....	28
Create Function Statements.....	29
Create View Statements.....	29
Create Procedure Statements.....	30
SECTION 04.....	31
Critical Evaluation.....	31
Future Implementations.....	32
Workload Matrix.....	33

## Section 1

### Introduction

- We are going to plan a solution for the university to implement an effective student's management system.

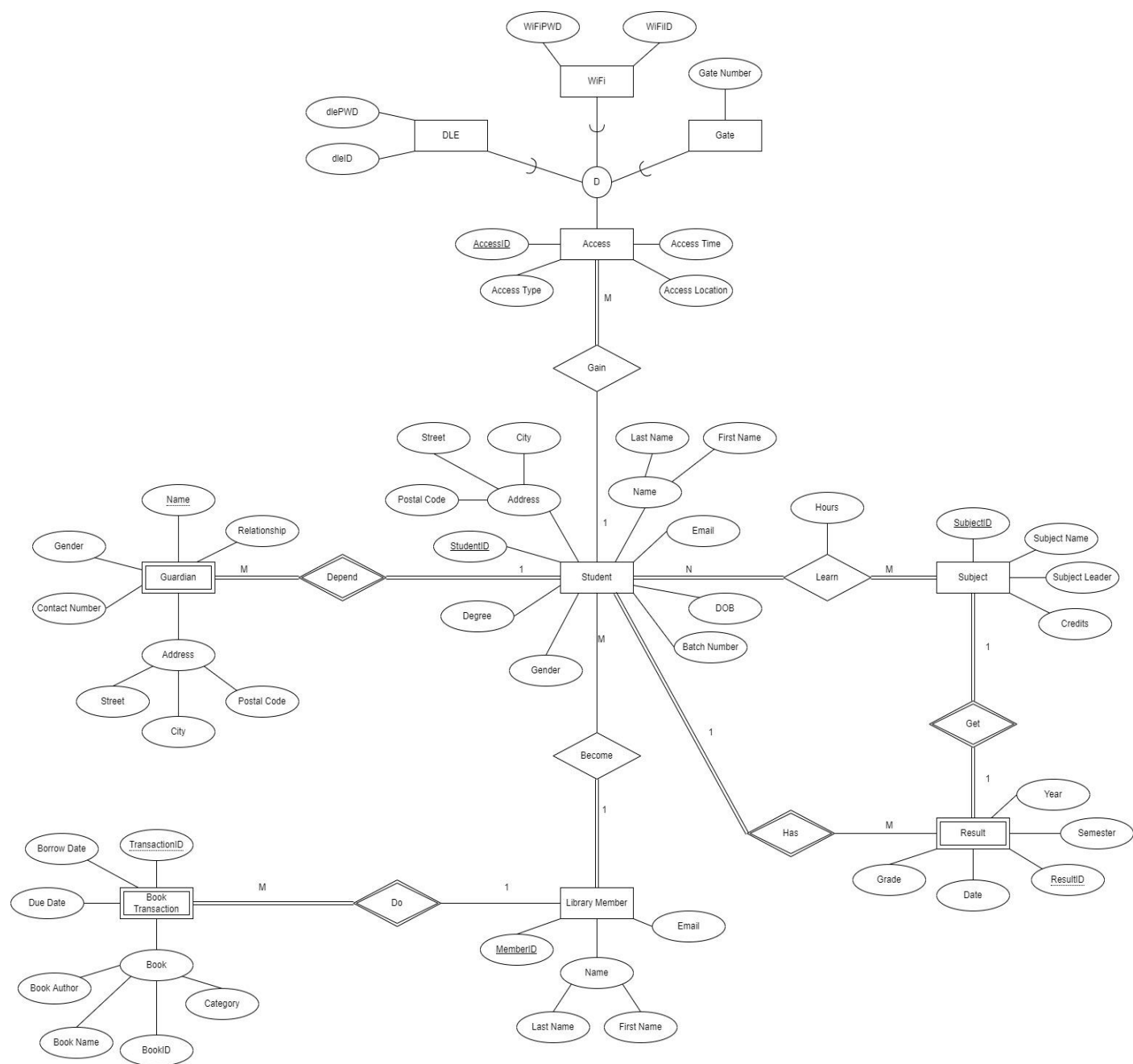
Currently, the university is operating as a file-based system which is time consuming and will require more space when managing departments in the university.

Considering about the mentioned problems, we hope to categorise the departments such as.

- ✓ Examination department: This will manage all the subjects and student details separately.
  - ✓ Registration department: This will maintain all the student registration details and personal details with their educational.
  - ✓ Library access department: This will maintain and manage all the details of students accessing the library facility.
  - ✓ IT department: This will maintain and check details of students accessing Wi-Fi and DLE and record their entries.
- In order for the university to operate an efficient student management system, a database structure needs to be implemented as a solution to categorise these departments and maintain records in an advanced manner.
  - As we have introduced the solution, we are going to further discuss about the appropriate database design for the given scenario.

# Extended Entity Relationship Diagram

Diagram



## Assumptions

- We assumed that address and name attributes are composite attributes. Which address include the city, street, postal code and name include the first name, last name.
- We assumed that every student must depend on at least one guardian and therefore, we made guardian as a weak entity which is related to the student strong entity.
- We assumed that all the students must participate in learning subjects and all the subjects must have a result. Therefore, we made result as a weak entity which related to both student and subject strong entities. A subject can have many students and a student can learn many subjects and only one result is related to a specific subject and a student must have many results for many subjects but a result must be specific for a one student.
- We assumed that student can become a library member and all the book transactions must be related to only a student who is a library member. Therefore, we made a weak entity as book transaction related to the library member strong entity. Library member can have many book transaction but a specific book transaction must related only to a one library member.
- We assumed that the specialization of the access to DLE, WiFi and Gate Access sub-entities is a disjoint relationship. Therefore, dleID and dlePWD are added to the DLE entity, WiFiID and Wi-FiPWD are added to the WiFi entity and Gate Number is added to the Gate entity.
- A student can have many access but access can be specified only for a one student.

# Relational Mapping

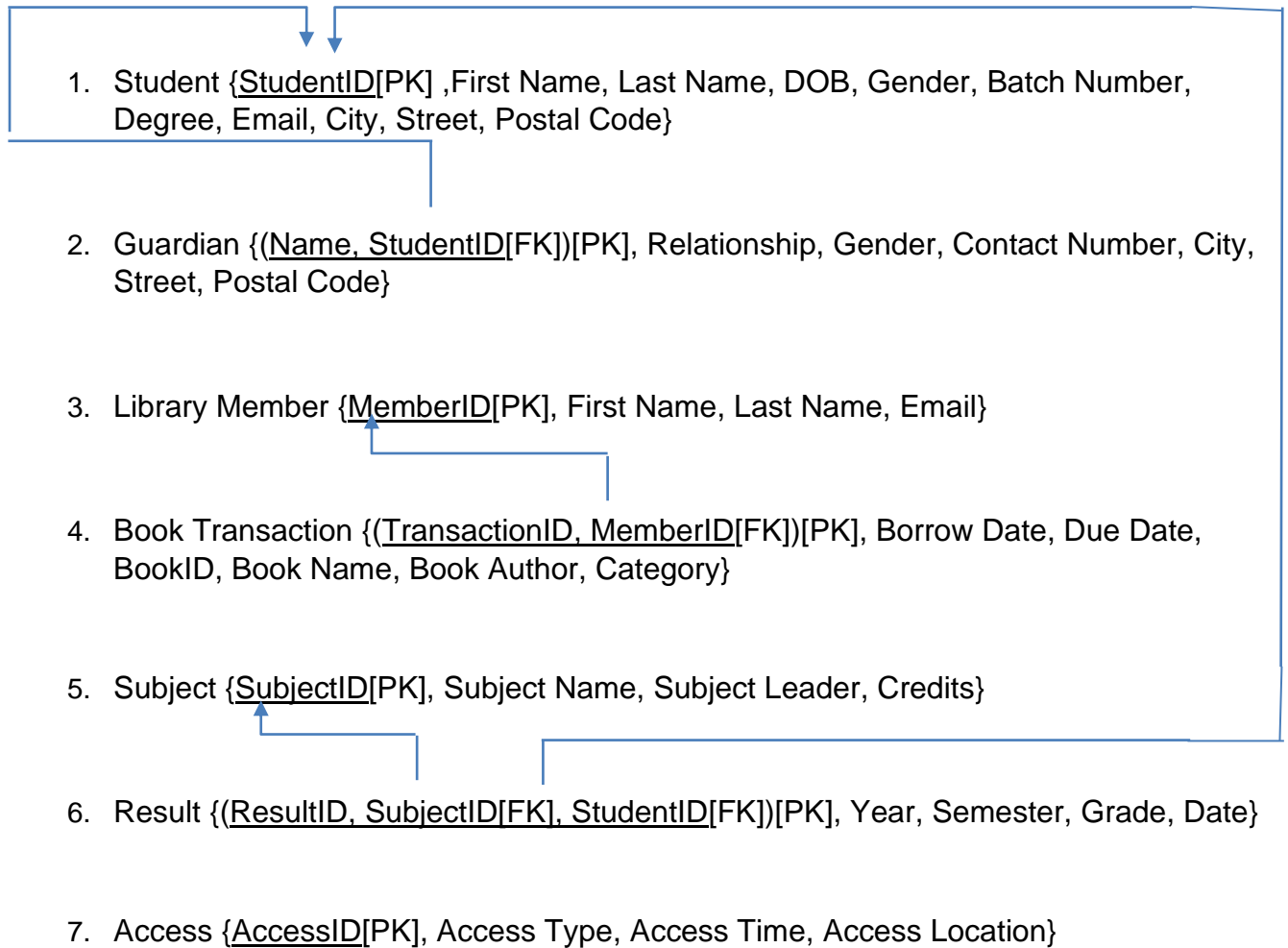
## Mapping of Strong Entities

1. Student {StudentID[PK], First Name, Last Name, DOB, Gender, Batch Number, Degree, Email, City, Street, Postal Code}
2. Library Member {MemberID[PK], First Name, Last Name, Email}
3. Subject {SubjectID[PK], Subject Name, Subject Leader, Credits}
4. Access {AccessID[PK], Access Type, Access Time, Access Location}

## Mapping of Weak Entities

- 
1. Student {StudentID[PK], First Name, Last Name, DOB, Gender, Batch Number, Degree, Email, City, Street, Postal Code}
  2. Guardian {(Name, StudentID[FK])[PK], Relationship, Gender, Contact Number, City, Street, Postal Code}
  3. Library Member {MemberID[PK], First Name, Last Name, Email}
  4. Book Transaction {(TransactionID, MemberID[FK])[PK], Borrow Date, Due Date, BookID, Book Name, Book Author, Category}
  5. Subject {SubjectID[PK], Subject Name, Subject Leader, Credits}
  6. Result {(ResultID, SubjectID[FK], StudentID[FK])[PK], Year, Semester, Grade, Date}
  7. Access {AccessID[PK], Access Type, Access Time, Access Location}

## Mapping of Binary 1:1 Relationships

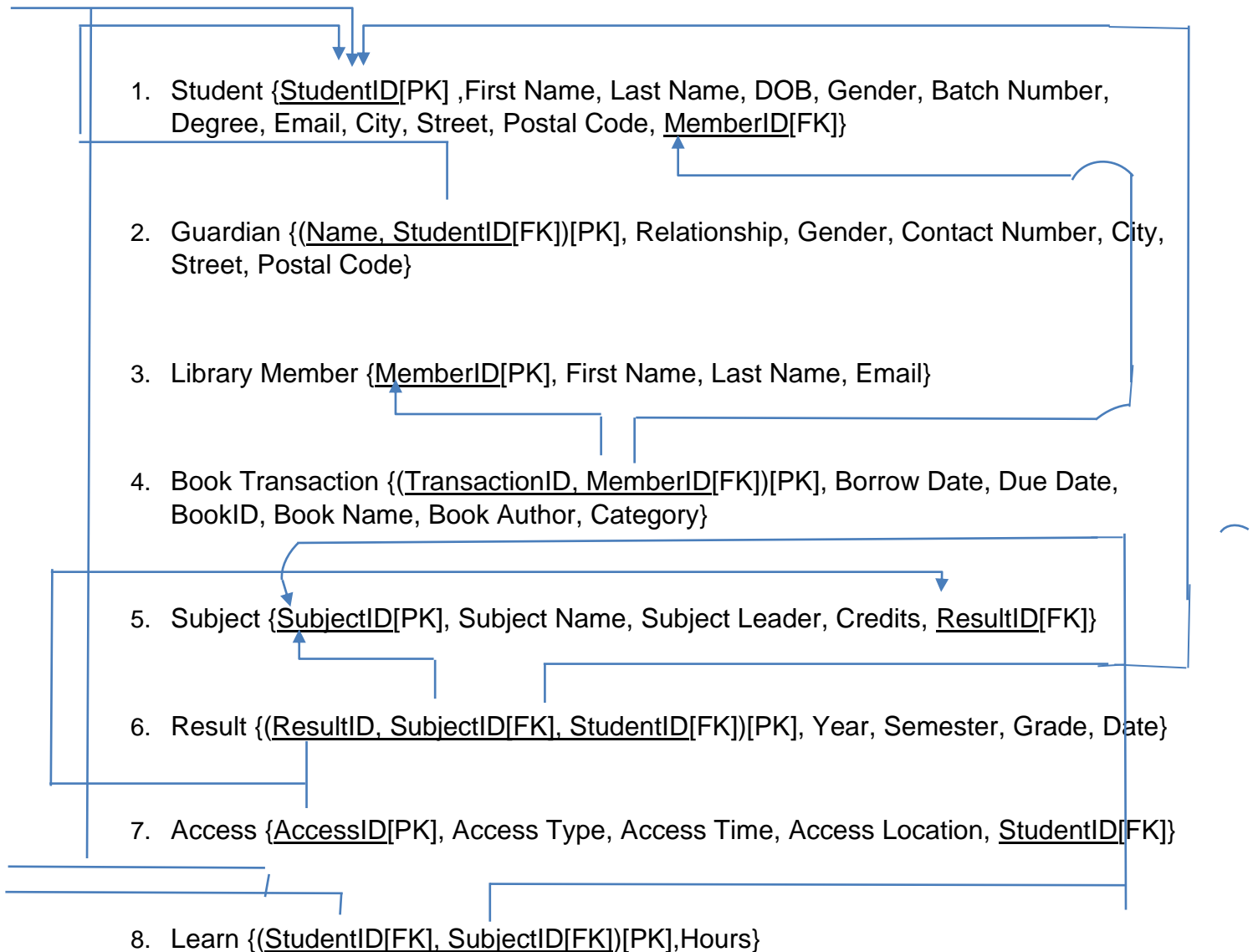




## Mapping of Binary 1:M Relationships

- 
1. Student {StudentID[PK], First Name, Last Name, DOB, Gender, Batch Number, Degree, Email, City, Street, Postal Code, MemberID[FK]}
  2. Guardian {(Name, StudentID[FK])[PK], Relationship, Gender, Contact Number, City, Street, Postal Code}
  3. Library Member {MemberID[PK], First Name, Last Name, Email}
  4. Book Transaction {(TransactionID, MemberID[FK])[PK], Borrow Date, Due Date, BookID, Book Name, Book Author, Category}
  5. Subject {SubjectID[PK], Subject Name, Subject Leader, Credits, ResultID[FK]}
  6. Result {(ResultID, SubjectID[FK], StudentID[FK])[PK], Year, Semester, Grade, Date}
  7. Access {AccessID[PK], Access Type, Access Time, Access Location, StudentID[FK]}

## Mapping of Binary M:N Relationships



## Mapping of Multi-Valued Attributes

\*\* No Multi-Valued Attributes Found.

## Mapping of N-ary Relationships

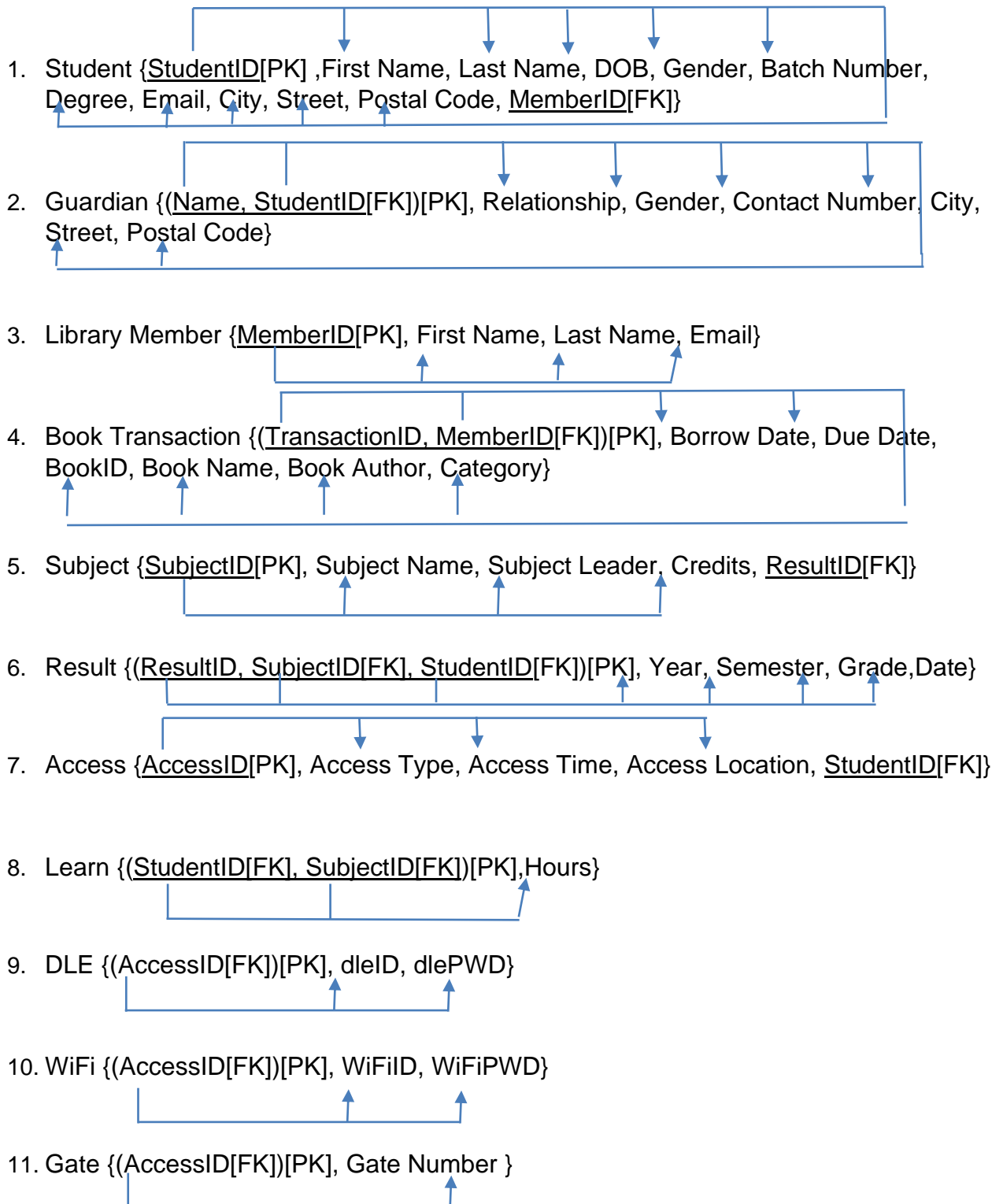
\*\* No N-ary Relationships Found.

## Mapping of Extended-Entity Relationships

- 
1. Student {StudentID[PK], First Name, Last Name, DOB, Gender, Batch Number, Degree, Email, City, Street, Postal Code, MemberID[FK]}
  2. Guardian {(Name, StudentID[FK])[PK], Relationship, Gender, Contact Number, City, Street, Postal Code}
  3. Library Member {MemberID[PK], First Name, Last Name, Email}
  4. Book Transaction {(TransactionID, MemberID[FK])[PK], Borrow Date, Due Date, BookID, Book Name, Book Author, Category}
  5. Subject {SubjectID[PK], Subject Name, Subject Leader, Credits, ResultID[FK]}
  6. Result {(ResultID, SubjectID[FK], StudentID[FK])[PK], Year, Semester, Grade, Date}
  7. Access {AccessID[PK], Access Type, Access Time, Access Location, StudentID[FK]}
  8. Learn {(StudentID[FK], SubjectID[FK])[PK], Hours}
  9. DLE {(AccessID[FK])[PK], dleID, dlePWD}
  10. WiFi {(AccessID[FK])[PK], WiFiID, WiFiPWD}
  11. Gate {(AccessID[FK])[PK], Gate Number}

# Normalization

## Dependencies



## 1<sup>st</sup> Normal Form (1NF)

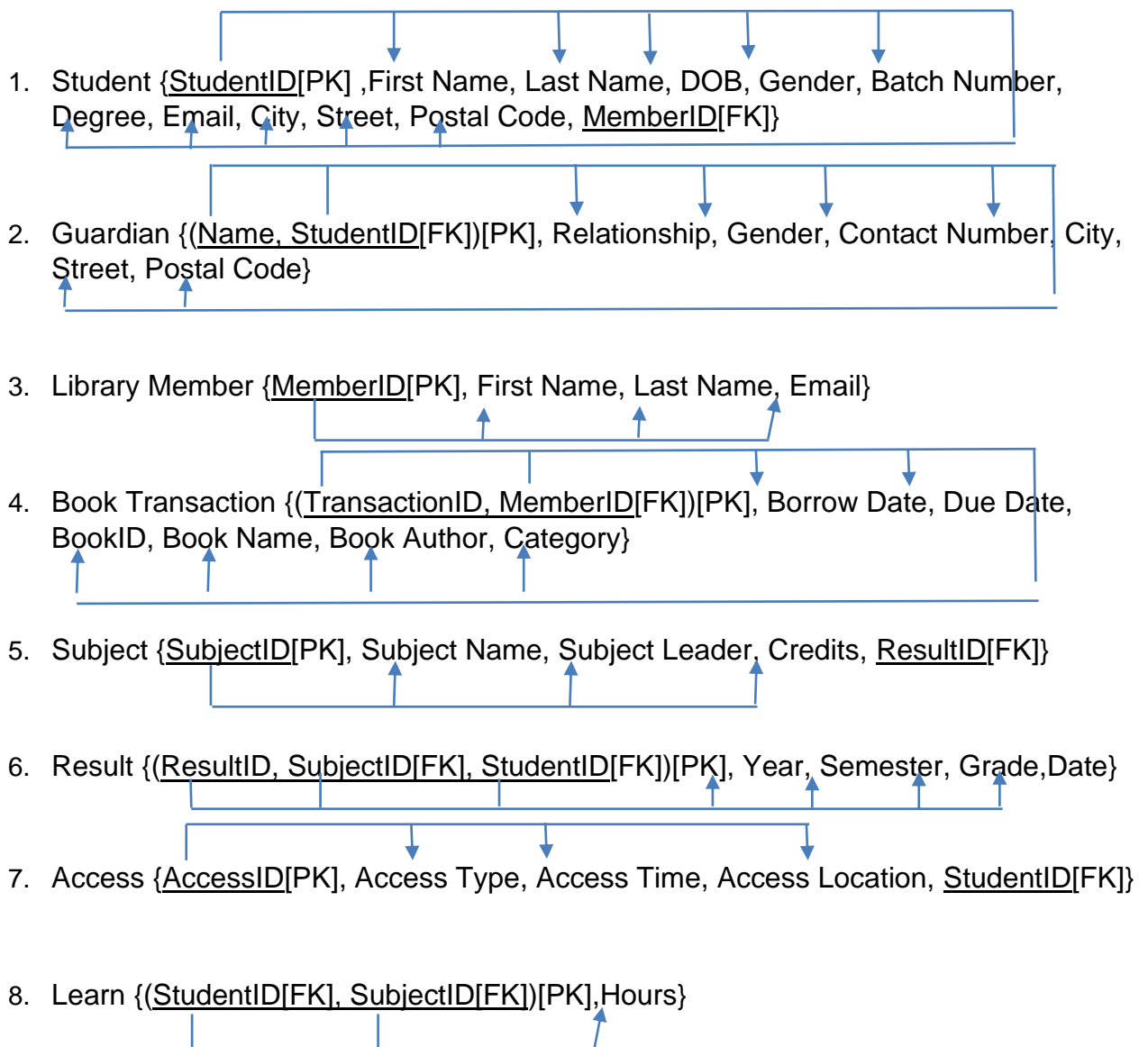
As there are no multi-valued attributes and has only atomic valued attributes, the above relational schema is already in First Normal Form.

## 2<sup>nd</sup> Normal Form (2NF)

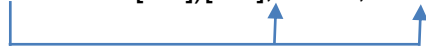
As there are no partial dependencies, the above relational schema is already in Second Normal Form.

## 3<sup>rd</sup> Normal Form (3NF)

As there are no transitive dependencies, the following relational schema can be considered as in the Third Normal Form.



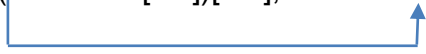
9. DLE {(AccessID[FK])[PK], dleID, dlePWD}



10. WiFi {(AccessID[FK])[PK], WiFiID, WiFiPWD}



11. Gate {(AccessID[FK])[PK], Gate Number }



## Data Dictionary

### 1. Student Table

Field Name	Data Type	Constraints	Constraint Name	Reference Table
StudentID	INTEGER	PRIMARY	pk_StudentID	
First Name	VARCHAR(30)	NOT NULL		
Last Name	VARCHAR(30)	NOT NULL		
DOB	DATE	NOT NULL		
Gender	CHAR(1)	NOT NULL		
Batch Number	VARCHAR(30)	NOT NULL		
Degree	VARCHAR(60)	NOT NULL		
Email	VARCHAR(60)	NOT NULL		
City	VARCHAR(20)	NOT NULL		
Street	VARCHAR(20)	NOT NULL		
Postal Code	VARCHAR(10)	NOT NULL		
MemberID	INTEGER	FOREIGN	fk_MemberID	Library Member

### 2. Guardian Table

Field Name	Data Type	Constraints	Constraint Name	Reference Table
Name	VARCHAR(60)	PRIMARY	pk_Name	
StudentID	INTEGER	PRIMARY	pk_StudentID	
		FOREIGN	fk_StudentID	Student
Relationship	VARCHAR(20)	NOT NULL		
Gender	CHAR(1)	NOT NULL		
Contact Number	VARCHAR(30)	NOT NULL		
City	VARCHAR(20)	NOT NULL		
Street	VARCHAR(20)	NOT NULL		
Postal Code	VARCHAR(10)	NOT NULL		

### 3. Library Member Table

Field Name	Data Type	Constraints	Constraint Name	Reference Table
MemberID	INTEGER	PRIMARY		
First Name	VARCHAR(30)			
Last Name	VARCHAR(30)			
Email	VARCHAR(60)			

#### 4. Book Transaction Table

Field Name	Data Type	Constraints	Constraint Name	Reference Table
TransactionID	INTEGER	PRIMARY	pk_TransactionID	
MemberID	INTEGER	PRIMARY	pk_MemberID	
		FOREIGN	fk_MemberID	Library Member
Borrow Date	DATE	NOT NULL		
Due Date	DATE	NOT NULL		
BookID	VARCHAR(30)	NOT NULL		
Book Name	VARCHAR(30)	NOT NULL		
Book Author	VARCHAR(30)	NOT NULL		
Category	VARCHAR(20)	NOT NULL		

#### 5. Subject Table

Field Name	Data Type	Constraints	Constraint Name	Reference Table
SubjectID	INTEGER	PRIMARY	pk_SubjectID	
Subject Name	VARCHAR(30)	NOT NULL		
Subject Leader	VARCHAR(30)	NOT NULL		
Credits	INTEGER	NOT NULL		
ResultID	INTEGER	FOREIGN	fk_ResultID	Result

#### 6. Result Table

Field Name	Data Type	Constraints	Constraint Name	Reference Table
ResultID	INTEGER	PRIMARY	pk_ResultID	
SubjectID	INTEGER	PRIMARY	pk_SubjectID	
		FOREIGN	fk_SubjectID	Subject
StudentID	INTEGER	PRIMARY	pk_StudentID	
		FOREIGN	fk_StudentID	Student
Year	INTEGER			
Semester	INTEGER			
Grade	CHAR(1)			
Date	DATE			

#### 7. Access Table

Field Name	Data Type	Constraints	Constraint Name	Reference Table
AccessID	INTEGER	PRIMARY	pk_AccessID	
Access Type	VARCHAR(10)	NOT NULL		
Access Time	TIME	NOT NULL		
Access Location	VARCHAR(10)	NOT NULL		
StudentID	INTEGER	FOREIGN	fk_StudentID	Student



## 8. Learn Table

Field Name	Data Type	Constraints	Constraint Name	Reference Table
StudentID	INTEGER	PRIMARY	pk_StudentID	
		FOREIGN	fk_StudentID	Student
SubjectID	INTEGER	PRIMARY	pk_SubjectID	
		FOREIGN	fk_SubjectID	Subject
Hours	INTEGER	NOT NULL		

## 9. DLE Table

Field Name	Data Type	Constraints	Constraint Name	Reference Table
AccessID	INTEGER	PRIMARY	pk_AccessID	
		FOREIGN	fk_AccessID	Access
dleID	VARCHAR(30)	NOT NULL		
dlePWD	VARCHAR(30)	NOT NULL		

## 10. WiFi Table

Field Name	Data Type	Constraints	Constraint Name	Reference Table
AccessID	INTEGER	PRIMARY	pk_AccessID	
		FOREIGN	fk_AccessID	Access
WiFiID	VARCHAR(30)	NOT NULL		
WiFiPWD	VARCHAR(30)	NOT NULL		

## 11. Gate Table

Field Name	Data Type	Constraints	Constraint Name	Reference Table
AccessID	INTEGER	PRIMARY	pk_AccessID	
		FOREIGN	fk_AccessID	Access
Gate Number	INTEGER	NOT NULL		

Following Data Formats are considered at the Data Entry.

1. Date: YY-MM-DD

2. Gender: 'M' or 'F'

## Section 02

### Create Table Statements

#### 1. Student Table

```
Create TABLE Student(
StudentID INTEGER NOT NULL,
First_Name VARCHAR(30) NOT NULL,
Last_Name VARCHAR(30) NOT NULL,
DOB DATE NOT NULL,
Gender CHAR(1) NOT NULL,
Batch_Number VARCHAR(30) NOT NULL,
Degree VARCHAR(60) NOT NULL,
Email VARCHAR(60) NOT NULL,
City VARCHAR(20) NOT NULL,
Street VARCHAR(20) NOT NULL,
Postal_Code VARCHAR(10) NOT NULL,
MemberID INTEGER NOT NULL,
CONSTRAINT pk_StudentID PRIMARY KEY (StudentID),
CONSTRAINT fk_MemberID FOREIGN KEY (MemberID) REFERENCES LibraryMember(MemberID)
);
```

#### 2. Guardian Table

```
CREATE TABLE Guardian(
Name VARCHAR(60),
StudentID INTEGER NOT NULL,
Relationship VARCHAR(20) NOT NULL,
Gender CHAR(1) NOT NULL,
Contact_Number VARCHAR(30) NOT NULL,
City VARCHAR(20) NOT NULL,
Street VARCHAR(20) NOT NULL,
PostalCode VARCHAR(10) NOT NULL,
CONSTRAINT pk_Name PRIMARY KEY (Name),
CONSTRAINT fk_StudentID FOREIGN KEY (StudentID) REFERENCES Student (StudentID)
);
```

#### 3. Library Member Table

```
CREATE TABLE [dbo].[LibraryMember](
[MemberID] [int] NOT NULL,
[First_Name] [varchar](30) NOT NULL,
[Last_Name] [varchar](30) NOT NULL,
[Email] [varchar](60) NOT NULL,
CONSTRAINT [pk_MemberID] PRIMARY KEY CLUSTERED
(
```

## 4. Book Transaction Table

```
CREATE TABLE [dbo].[BookTransaction](
    [TransactionID] [int] NOT NULL,
    [MemberID] [int] NOT NULL,
    [Borrow_Date] [date] NOT NULL,
    [Due_Date] [date] NOT NULL,
    [BookID] [varchar](30) NOT NULL,
    [BookName] [varchar](30) NOT NULL,
    [BookAuthor] [varchar](30) NOT NULL,
    [Category] [varchar](20) NOT NULL,
    CONSTRAINT [pk_TransactionID] PRIMARY KEY CLUSTERED
(
    [TransactionID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[BookTransaction] WITH CHECK ADD CONSTRAINT [fk_TMemberID] FOREIGN KEY([MemberID])
REFERENCES [dbo].[LibraryMember] ([MemberID])
GO

ALTER TABLE [dbo].[BookTransaction] CHECK CONSTRAINT [fk_TMemberID]
GO
```

## 5. Subject Table

```
CREATE TABLE [dbo].[Subject](
    [SubjectID] [int] NOT NULL,
    [Subject_Name] [varchar](30) NOT NULL,
    [Subject_Leader] [varchar](30) NOT NULL,
    [Credits] [int] NOT NULL,
    [ResultID] [int] NOT NULL,
    CONSTRAINT [pk_SubjectID] PRIMARY KEY CLUSTERED
(
    [SubjectID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Subject] WITH CHECK ADD FOREIGN KEY([ResultID])
REFERENCES [dbo].[Result] ([ResultID])
GO
```

## 6. Result Table

```
CREATE TABLE Result(
    ResultID INTEGER NOT NULL,
    SubjectID INTEGER NOT NULL,
    StudentID INTEGER NOT NULL,
    Year INTEGER NOT NULL,
    Semester INTEGER NOT NULL,
    Grade CHAR(1) NOT NULL,
    Date DATE NOT NULL,
    CONSTRAINT pk_ResultID PRIMARY KEY (ResultID, SubjectID, StudentID),
    CONSTRAINT fk_SubjectID FOREIGN KEY (SubjectID) REFERENCES Subject (SubjectID),
    CONSTRAINT fk_RStudentID FOREIGN KEY(StudentID) REFERENCES Student (StudentID)
);
```

## 7. Access Table

```
CREATE TABLE Access(
    AccessID INT NOT NULL,
    AccessType VARCHAR(10) NOT NULL,
    AccessTime TIME NOT NULL,
    AccessLocation VARCHAR(10) NOT NULL,
    StudentID INT NOT NULL,
    CONSTRAINT pk_AccessID PRIMARY KEY (AccessID),
    CONSTRAINT fk_AccStudentID FOREIGN KEY (AccessID) REFERENCES Student (StudentID)
);
```

## 8. Learn Table

---

```
CREATE TABLE Learn(  
  StudentID INT NOT NULL,  
  SubjectID INT NOT NULL,  
  Hours INT NOT NULL,  
  CONSTRAINT pk_LStudentID PRIMARY KEY (StudentID),  
  CONSTRAINT fk_LStudentID FOREIGN KEY (StudentID) REFERENCES Student (StudentID),  
  CONSTRAINT fk_LSubjectID FOREIGN KEY (SubjectID) REFERENCES Subject (SubjectID)  
);
```

## 9. DLE Table

```
CREATE TABLE DLE (  
  AccessID INT NOT NULL,  
  dleID VARCHAR(30) NOT NULL,  
  dlePWD VARCHAR(30) NOT NULL,  
  CONSTRAINT pk_DLEAccessID PRIMARY KEY (AccessID),  
  CONSTRAINT fk_DLEAccessID FOREIGN KEY (AccessID) REFERENCES Access (AccessID)  
);
```

## 10. WiFi Table

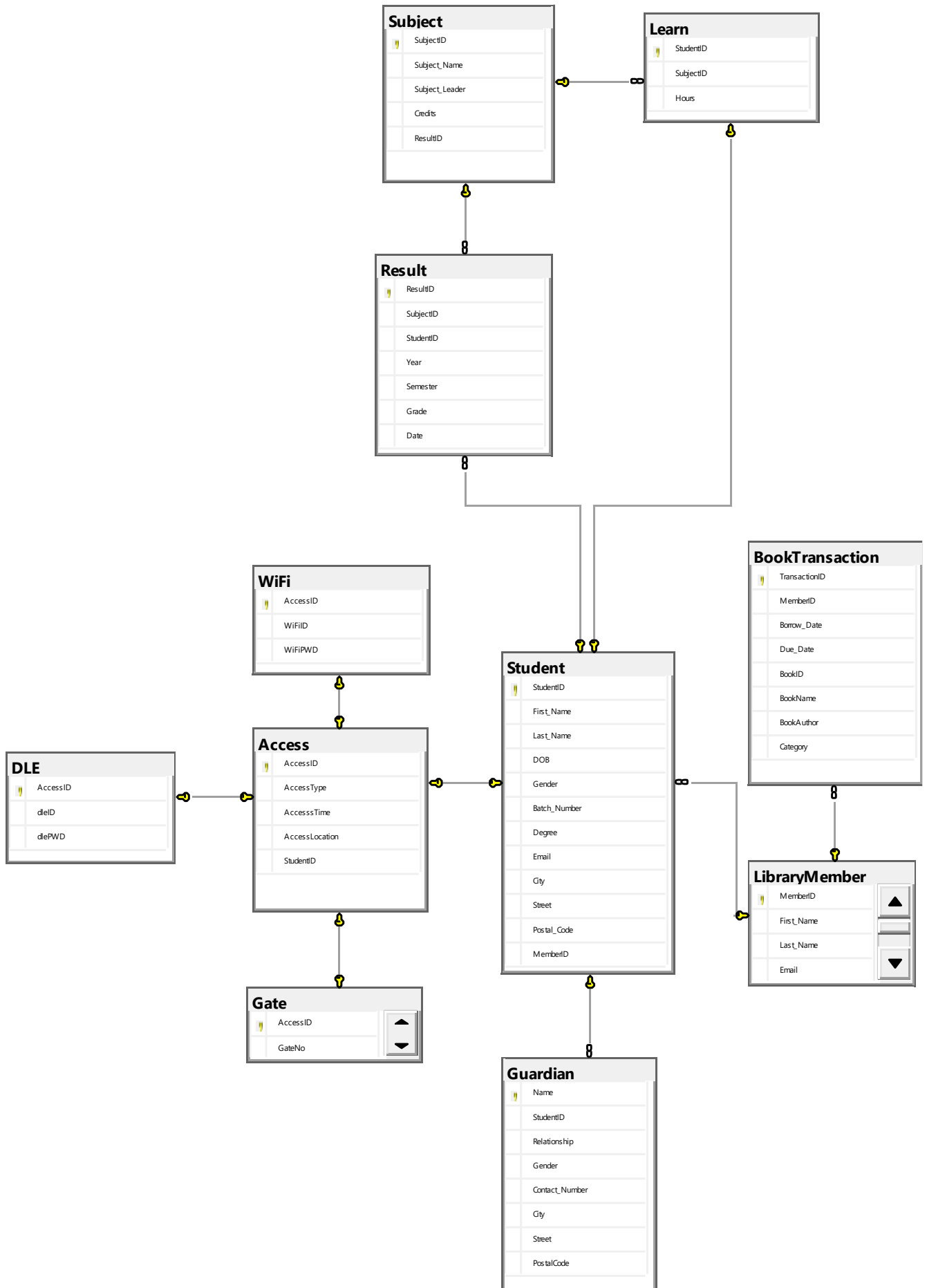
```
CREATE TABLE WiFi(  
  AccessID INT NOT NULL,  
  WiFiID VARCHAR(30) NOT NULL,  
  WiFiPWD VARCHAR(30) NOT NULL,  
  CONSTRAINT pk_WAccessID PRIMARY KEY (AccessID),  
  CONSTRAINT fk_WAccessID FOREIGN KEY (AccessID) REFERENCES Access(AccessID)  
);
```

## 11. Gate Access ID

---

```
CREATE TABLE Gate(  
  AccessID INT NOT NULL,  
  GateNo INT NOT NULL,  
  CONSTRAINT pk_Gate_AccessID PRIMARY KEY (AccessID),  
  CONSTRAINT fk_Gate_AccessID FOREIGN KEY (AccessID) REFERENCES Access (AccessID)  
);
```

## Database Diagram



## Database Sample Records

### 1. Student Table

StudentID	First_Name	Last_Name	DOB	Gender	Batch_Number	Degree	Email	City	Street	Postal_Code	MemberID
20000	John	Wallace	1999-07-23	M	01	Business Management	johnwallace@gmail.com	Nugegoda	23 Main Road	NA 32132	1
20001	Max	Franklin	1999-02-13	M	02	Business Management	maxfranklin@gmail.com	Kandy	13 Main Road	NA 32132	2
20002	Alex	Jones	2000-05-29	M	03	Software Engineering	alexjones@gmail.com	Aluthgama	23 Palm Street	PS 82981	3
20003	Steve	Harden	1999-07-23	M	02	Computer Science	steveharden@gmail.com	Negombo	03 Main Road	MR 32131	4
20004	Ned	Leeds	1998-10-01	M	01	Accounting and Finance	nedleeds@gmail.com	Gampaha	09 Main Road	NL 89898	5
20005	George	Walton	2000-01-28	M	03	Computer Networks	georgewalton@gmail.com	Kadawatha	11 Main Road	PF 43478	6
20006	Kevin	Bates	1998-11-27	M	01	Computer Security	kevinbates@gmail.com	Ja-Ela	07 Main Road	BA 43434	7
20007	Peter	Rodriguez	1999-12-25	M	02	Computer Science	peterrodriguez@gmail.com	Wennap...	01 Main Road	KU 65653	8
20008	Sally	Cleveland	1999-06-03	F	02	Business Management	sallycleveland@gmail.com	Kiribathg...	07 Main Road	VW 77676	9
20009	Mary	Russell	2000-08-04	F	03	Software Engineering	maryrussell@gmail.com	Yakkala	25 Main Road	TS 54545	10
20010	Jane	Watson	1999-02-14	F	02	Logistic Management	janewatson@gmail.com	Nugegoda	15 Main Road	FR 87742	11
20011	Vanessa	Mccloy	1999-05-29	F	02	Computer Networks	vanessamccloy@gmail.c...	Gampaha	09 Main Road	GS 67345	12

### 2. Guardian Table

Name	StudentID	Relationship	Gender	Contact_Number	City	Street	PostalCode
Ben Jones	20002	Father	M	0331257702	Aluthgama	23 Palm Street	PS 82981
Chris Leeds	20004	Father	M	0333239281	Gampaha	09 Main Road	NL 89898
Fiona Walton	20005	Mother	F	0332333883	Kadawatha	11 Main Road	PF 43478
Frank Cleveland	20008	Father	M	0332437439	Kiribathgoda	07 Main Road	VW 77676
Fred Wallace	20000	Father	M	0332266749	Nugegoda	23 Main Road	NA 32132
Hank Watson	20010	Father	M	0334343784	Nugegoda	15 Main Road	FR 87742
James Harden	20003	Father	M	0333954749	Negombo	03 Main Road	MR 32132
Kate Russell	20009	Mother	F	0330098213	Yakkala	25 Main Road	TS 54545
May Franklin	20001	Mother	F	0333443946	Kandy	13 Main Road	NA 32132
Mia Rodriguez	20007	Mother	F	0332121275	Wennappuwa	01 Main Road	KU 65653
Patrick Bates	20006	Father	M	0332298989	Ja-Ela	07 Main Road	BA 43434
Walt Mccloy	20011	Father	M	0330084621	Gampaha	09 Main Road	GS 67345

### 3. Library Member

MemberID	First_Name	Last_Name	Email
1	John	Wallace	johnwallace@gmail.com
2	Max	Franklin	maxfranklin@gmail.com
3	Alex	Jones	alexjones@gmail.com
4	Steve	Harden	steveharden@gmail.com
5	Ned	Leeds	nedleeds@gmail.com
6	George	Walton	georgewalton@gmail.com
7	Kevin	Bates	kevinbates@gmail.com
8	Peter	Rodriguez	peterrodriguez@gmail.com
9	Sally	Cleveland	sallycleveland@gmail.com
10	Mary	Russell	maryrussell@gmail.com
11	Jane	Watson	janewatson@gmail.com
12	Vanessa	Mccloy	vanessamccloy@gmail.com

#### 4. Book Transaction

TransactionID	MemberID	Borrow_Date	Due_Date	BookID	BookName	BookAuthor	Category
1	1	2021-01-03	2021-01-10	93232133	An Untold Story	George Orwell	Fictions, Novels
2	2	2020-05-12	2021-05-19	93434434	Harry Potter	J.K Rowling	Fictions, Novels
3	3	2021-09-23	2021-10-02	93545435	To Kill a Mockingbird	Harper Lee	Southern Gothic
4	4	2021-04-30	2021-05-09	93989677	Mata Kiyanu Mana	Kamala Wijeratne	General Knowledge
5	5	2021-11-03	2021-11-10	93434898	The Lord of the Rings	J.R.R Tolkien	Fantasy, Adventure
6	6	2021-12-03	2021-02-10	93343441	Amma - Mathru Pujanaya	Eric Illayaparachchi	Novels
7	7	2021-02-05	2021-02-12	93232327	1984	Arkady Leokum	Political Fiction
8	8	2021-06-24	2021-07-10	93434929	Animal Farm: A Fair Story	Surath De Mal	Fiction
9	9	2021-08-13	2021-08-20	93983438	Tailorbird	Rick Riordan	Fictions, Novels
10	10	2021-01-03	2021-01-10	93482389	Thee Ha Thaa	Manori Manamperi	Fictions, Novels
12	12	2021-02-17	2021-02-24	932434321	The Adventures of Tom Sawyer	Mark Twain	Fictions, Novels

#### 5. Subject

SubjectID	Subject_Name	Subject_Leader	Credits	ResultID
1	Accounting	Tony Cole	30	11
2	Computing Group Project	Bruce Maxwell	35	12
3	Information Management	Susan Walters	15	13
4	Computer Architecture	Veronica Shubert	25	14
5	Professional Development	Karl Wilmes	30	15
6	Logistics	Lynn Troy	30	16
7	Software Engineering	Gregory Alba	40	17
8	Computer Science	Anthony Matthews	35	18
9	Business Management	Michael Erickson	40	19
10	Computer Networks	Harry Hogan	25	20
11	Web Development	Clint Barton	25	21
12	Database Architecture	Frank Oswalt	25	22

#### 6. Result

ResultID	SubjectID	StudentID	Year	Semester	Grade	Date
11	1	20000	3	2	A	2020-01-13
12	1	20001	2	2	B	2021-03-03
13	3	20002	1	1	A	2021-01-23
14	10	20003	3	1	A	2020-09-24
15	4	20004	3	2	A	2021-01-13
16	5	20005	2	2	C	2019-07-16
17	5	20006	2	2	C	2020-08-12
18	2	20007	2	2	B	2021-08-11
19	6	20008	1	1	B	2021-07-08
20	7	20009	1	1	A	2021-02-03
21	8	20010	3	1	C	2021-08-16
22	9	20011	1	2	B	2021-09-24

## 7. Access

AccessID	AccessType	AccessTime	AccessLocation	StudentID
1	Gate	08:30:00.0000000	Entrance	20000
2	Gate	08:45:00.0000000	Entrance	20001
3	Wi-Fi	09:12:00.0000000	FOB	20002
4	Wi-Fi	08:18:00.0000000	FOC	20003
5	DLE	09:30:00.0000000	FOC	20004
6	DLE	11:25:00.0000000	SOB	20005
7	Wi-Fi	08:23:00.0000000	Entrance	20006
8	Wi-Fi	08:26:00.0000000	Canteen	20007
9	Gate	08:31:00.0000000	Entrance	20008
10	DLE	14:23:00.0000000	FOC	20009
11	Gate	08:18:00.0000000	Entrance	20010
12	Gate	08:28:00.0000000	Entrance	20011

## 8. Learn

StudentID	SubjectID	Hours
20000	1	5
20001	1	3
20002	3	2
20003	10	6
20004	4	4
20005	5	3
20006	5	3
20007	2	6
20008	6	5
20009	7	3
20010	8	4
20011	9	5

## 9. DLE

AccessID	dleID	dlePWD
1	20000	jw20000
2	20001	mf20001
3	20002	aj20002
4	20003	sh20003
5	20004	nl20004
6	20005	gw20005
7	20006	kb20006
8	20007	pr20007
9	20008	sc20008
10	20009	mr20009
11	20010	jw20010
12	20011	vm20011



## 10. Wi-Fi

AccessID	WiFiID	WiFiPWD
1	20000	jw20000
2	20001	mf20001
3	20002	aj20002
4	20003	sh20003
5	20004	nl20004
6	20005	gw20005
7	20006	kb20006
8	20007	pr20007
9	20008	sc20008
10	20009	mr20009
11	20010	jw20010
12	20011	vm20011

## 11. Gate

AccessID	GateNo
1	321123
2	339291
3	338236
4	334537
5	390998
6	303082
7	332139
8	355552
9	320318
10	323913
11	312121
12	308082

## Section 03

### Create Trigger Statements

#### 1. CheckResultAvailability Trigger

```
]CREATE TRIGGER tr_Result_CheckAvailability
ON Result
AFTER INSERT
AS
]BEGIN
    DECLARE @StudentID INTEGER
    DECLARE @AVBL VARCHAR(3)
    SELECT @StudentID = StudentID FROM inserted
    SELECT @AVBL = Grade FROM Result WHERE StudentID = @StudentID
] IF (@AVBL LIKE 'A')
] BEGIN
    PRINT 'The student has passed the subject'
    ROLLBACK
    END
- END
-
```

#### 2. BookInsert Trigger

```
CREATE TRIGGER tr_Book_ForInsert
ON BookTransaction
FOR INSERT
AS
BEGIN
    SELECT * FROM inserted
END
```

#### 3. LearnInsert Trigger

```
]CREATE TRIGGER tr_Learn_ForInsert
ON Learn
FOR INSERT
AS
]BEGIN
    SELECT * FROM inserted
    END
-
```

## Create Function Statements

### 1. StudentDetailsByName Function

```
CREATE FUNCTION Student_Details (@First_Name VARCHAR(30))
RETURNS TABLE
AS
RETURN
(
SELECT *
FROM Access
WHERE StudentID = (SELECT StudentID
                   FROM Student
                   WHERE First_Name = @First_Name)
)
```

```
SELECT * FROM Student_Details('Peter');
```

AccessID	AccessType	AccessTime	AccessLocation	StudentID
8	Wi-Fi	08:26:00.0000000	Canteen	20007

### 2. BookBorrowDetailsByMemberID Function

```
CREATE FUNCTION Book_Borrower_Details (@Borrower_ID INTEGER)
RETURNS TABLE
AS
RETURN
(
SELECT *
FROM BookTransaction
WHERE MemberID = (SELECT MemberID
                  FROM LibraryMember
                  WHERE MemberID = @Borrower_ID)
)
```

```
SELECT * FROM Book_Borrower_Details ('3');
```

TransactionID	MemberID	Borrow_Date	Due_Date	BookID	BookName	BookAuthor	Category
3	3	2021-09-23	2021-10-02	93545435	To Kill a Mockingbird	Harper Lee	Southern Gothic

### 3. GuardianDetailsByStudentName Function

```
CREATE FUNCTION Guardian_Details (@Student_Name VARCHAR(30))
RETURNS TABLE
AS
RETURN
(
SELECT *
FROM Guardian
WHERE StudentID = (SELECT StudentID
                   FROM Student
                   WHERE First_Name = @Student_Name)
)
```

```
SELECT * FROM Guardian_Details ('Alex');
```

Name	StudentID	Relationship	Gender	Contact_Number	City	Street	PostalCode
Ben Jones	20002	Father	M	0331257702	Aluthgama	23 Palm Street	PS 82981

## Create View Statements

### 1. Borrower Details View

```
SELECT * FROM BorrowerDetails;  
AS  
SELECT BookTransaction.MemberID,Student.First_Name,Student.Last_Name,Student.Email,Student.DOB,Student.Gender,Student.City,BookTransaction.Borrow_Date,BookTransaction.Due_Date,BookTransaction.BookID  
FROM (Student  
INNER JOIN BookTransaction ON Student.MemberID = BookTransaction.MemberID)
```

SELECT \* FROM BorrowerDetails:

MemberID	First_Name	Last_Name	Email	DOB	Gender	City	Borrow_Date	Due_Date	BookID
1	John	Wallace	johnwallace@gmail.com	1999-07-23	M	Nugegoda	2021-01-03	2021-01-10	93232133
2	Max	Franklin	maxfranklin@gmail.com	1999-02-13	M	Kandy	2020-05-12	2021-05-19	93434434
3	Alex	Jones	alexjones@gmail.com	2000-05-29	M	Aluthgama	2021-09-23	2021-10-02	93545435
4	Steve	Harden	steveharden@gmail.com	1999-07-23	M	Negombo	2021-04-30	2021-05-09	93989677
5	Ned	Leeds	nedleeds@gmail.com	1998-10-01	M	Gampaha	2021-11-03	2021-11-10	93434898
6	George	Walton	georgewalton@gmail.com	2000-01-28	M	Kadawatha	2021-12-03	2021-02-10	93343441
7	Kevin	Bates	kevinbates@gmail.com	1998-11-27	M	Ja-Ela	2021-02-05	2021-02-12	93232327
8	Peter	Rodriguez	peterrodriguez@gmail.c...	1999-12-25	M	Wennap...	2021-06-24	2021-07-10	93434929
9	Sally	Cleveland	sallycleveland@gmail.c...	1999-06-03	F	Kiribathg...	2021-08-13	2021-08-20	93983438
10	Mary	Russell	maryrussell@gmail.com	2000-08-04	F	Yakkala	2021-01-03	2021-01-10	93482389
12	Vanessa	McCloy	vanessamccloy@gmail....	1999-05-29	F	Gampaha	2021-02-17	2021-02-24	932434...

### 2. Subject Leader Details View

```
CREATE VIEW SubjectLeaderDetails  
AS  
SELECT Learn.SubjectID,Subject.Subject_Name,Subject.Subject_Leader,Learn.Hours  
FROM (Subject  
INNER JOIN Learn ON Subject.SubjectID = Learn.SubjectID)
```

SELECT \* FROM SubjectLeaderDetails;

SubjectID	Subject_Name	Subject_Leader	Hours
1	Accounting	Tony Cole	5
1	Accounting	Tony Cole	3
3	Information Management	Susan Walters	2
10	Computer Networks	Harry Hogan	6
4	Computer Architecture	Veronica Shubert	4
5	Professional Development	Karl Wilmes	3
5	Professional Development	Karl Wilmes	3
2	Computing Group Project	Bruce Maxwell	6
6	Logistics	Lynn Troy	5
7	Software Engineering	Gregory Alba	3
8	Computer Science	Anthony Matthews	4
9	Business Management	Michael Erickson	5

### 3. Student Access Details View

```
CREATE VIEW StudentAccessDetails
AS
SELECT Student.StudentID, Student.First_Name, Student.Last_Name, Access.AccessType, Access.AccessLocation, Access.AccessTime
FROM (Student
INNER JOIN Access ON Student.StudentID = Access.StudentID)
;SELECT * FROM StudentAccessDetails;
```

StudentID	First_Name	Last_Name	AccessType	AccessLocation	AccessTime
20000	John	Wallace	Gate	Entrance	08:30:00.0000000
20001	Max	Franklin	Gate	Entrance	08:45:00.0000000
20002	Alex	Jones	Wi-Fi	FOB	09:12:00.0000000
20003	Steve	Harden	Wi-Fi	FOC	08:18:00.0000000
20004	Ned	Leeds	DLE	FOC	09:30:00.0000000
20005	George	Walton	DLE	SOB	11:25:00.0000000
20006	Kevin	Bates	Wi-Fi	Entrance	08:23:00.0000000
20007	Peter	Rodriguez	Wi-Fi	Canteen	08:26:00.0000000
20008	Sally	Cleveland	Gate	Entrance	08:31:00.0000000
20009	Mary	Russell	DLE	FOC	14:23:00.0000000
20010	Jane	Watson	Gate	Entrance	08:18:00.0000000
20011	Vanessa	Mccloy	Gate	Entrance	08:28:00.0000000

## Create Procedure Statements

### 1. GetGender Procedure

```
CREATE PROCEDURE GetGender
    @Gender char(1)
AS
BEGIN
    SELECT StudentID, First_Name, Gender
    FROM Student
    Where Gender = @Gender
END
```

```
Execute GetGender 'M';
```

StudentID	First_Name	Gender
20000	John	M
20001	Max	M
20002	Alex	M
20003	Steve	M
20004	Ned	M
20005	George	M
20006	Kevin	M
20007	Peter	M

### 2. GetStudentDetailsByBorrowedDateandDueDate Procedure

```
CREATE PROCEDURE GetStudentDetailsByBorrowedDateAndDueDate
    @Borrowed_Date DATE,
    @Due_Date DATE
AS
BEGIN
    SELECT Student.StudentID, Student.MemberID, Student.First_Name, Student.Last_Name, Student.Email, BookTransaction.TransactionID, BookTransaction.BookID, BookTransaction.BookName, BookTransaction.Borrow_Date, BookTransaction.Due_Date
    FROM Student, BookTransaction
    Where Borrow_Date = @Borrowed_Date
    AND Due_Date = @Due_Date
END
```

```
Execute GetStudentDetailsByBorrowedDateAndDueDate '2021-09-23', '2021-10-02';
```

StudentID	MemberID	First_Name	Last_Name	Email	TransactionID	BookID	BookName	Borrow_Date	Due_Date
20000	1	John	Wallace	johnwallace@gmail.com	3	93545435	To Kill a Mockingbird	2021-09-23	2021-10-02

## Section 04

### Critical Evaluation

Our relational database management system is planned, designed, and implemented as a solution to create a Student Management System with a fully functional database application which is located in several areas with Student registration department, examination department, the library and IT department where the student administration is assigned to maintain each and every aspect of the system. The libraries can locate books which are borrowed by students. In

addition, the below functionalities will help the student DLE to easily administrate all the daily activities regarding subject and exam details with ease and reliability.

- Database will validate all the entered data into the system using the Stored Procedures and User-Defined Functions so that the Student administration cannot enter invalid data to the system. It comprises the stored procedures for the Student Table, to get the student details by Gender, to get student details by entering Borrowed Date and Due date accordingly.
- The database also checks the availability of the results when the student has completed his/her exams and is waiting to check. Therefore, the triggered table would roll back the execution process.
- The student management system database has its drawbacks such as.
  - The Result table is connected with the Subject table with an identifying relationship, therefore the Result table is a weak entity and is fully dependent on the Subject table, which is also a weak entity of student table. That could create a conflict to the Result and Subject table and might deliver errors in future implements.
  - The IT department is bound to maintain the student IT Access division, but since there are two types of access as DLE and Wi-Fi and Gate, the system requires a proper mechanism to allocate the department, respectively.
- To overcome the weaknesses of the database, we might have to come up with another solution which would consider the result table only as a weak entity of the subject Table. To allocate the IT staff members accordingly, the system should be able to provide the DLE, Wi-Fi and Gate access details to the staff separately.

## Future Implementations

The current implementation of the Student Management System primarily serves the examination, registration, library and IT related activities and management. As steps and phases of further implementation, we would like to make the following implementations to the current database application.

- I. Implementation of the database to handle students borrowing books from library and book return Management functionalities more accordingly which would notify the book return details with the borrowed date and due date to the student accordingly when a due date is close, warning notifications when the due date is passed, and update the Book Transaction table according with the fee information for each delayed book returns, and book return notification which would update the book availability.
- II. Implementation of the database application to serve Human Resource management functionalities required by Human Resource department of Student Management System, so that the staff can be divided further into academic and non-academic staff which would issue and manage payrolls of the staff and maintain staff allocation with work hours. Hence, payroll reports and all other functionalities of the department will also be handled.
- III. Implementation of the database to have registered and non-registered student library members so that only registered members can borrow books.
- IV. Implementation of the database to with trigger functions to each table along with UPDATE and DELETE statements and using Stored Procedures to improve the efficiency. Using functions with INPUT and OUTPUT parameters to view and improve the database tables in a meaningful way. Usage of views to reduce the complexity of the database schema and implement row and column level security to the tables.
- V. Updating the system into better a software with a good interface, making the current version of the database to keep up with the latest SQL software in the future, and using built-in encryption to protect sensitive database information.



## Workload Matrix

Index Number	EER Diagram	Relational Schema, Data Normalization	Data Dictionary, Table Constraints	Triggers, Functions, Views, Stored Procedures	Evaluation, Future Implementation
10749130	✓			✓	
10749170		✓			✓
10750062	✓			✓	
10749135		✓	✓		
10749140		✓	✓		
10749131			✓		✓