



**IN
PARTNERSHIP
WITH
PLYMOUTH
UNIVERSITY**

Information Management & Retrieval

PUSL2019

Coursework

Group Members	
Lanka Pathmakumara	10899186
Senadheerage Dinethmin	10899275
Thuseya Rankelum	10899423
Liyanage Perera	10898869
Kenath Kapilarathna	10899412
Ranuja T Liyanaarachchi	10899415
Hondamuni A Kumari	10900325
Nawagamuwage Perera	10898873
Thuduwa W Silva	10899212

Table of Contents

INTRODUCTION.....	3
EXTENDED ENTITY RELATIONSHIP(EER)	4
ASSUMPTIONS	5
RELATIONAL MAPPING	6
DATA NORMALIZATION	7
DATA DICTIONARY	11
MICROSOFT SQL SERVER CREATES TABLE STATEMENTS.	18
DATABASE DIAGRAM.....	24
TABLES.....	25
CREATE TRIGGER	32
CREATE VIEW.....	33
CREATE PROCEDURE	34
CREATE FUNCTION	35
CRITICAL APPRAISAL OF THE SOLUTION	36
FUTURE IMPLEMENTATION.....	38

INTRODUCTION

Several crucial factors must be considered for the database upgrade at NSBM Green University to guarantee a thorough and effective solution. A strong database is necessary for the university's management system to manage different student-related tasks like registrations, exams, and access-related information safely and effectively.

Apart from the aforementioned factors, it is essential to augment the database solution by integrating a function to block student cards for enhanced security and control over access. The goal of this improvement is allowing the university to take disciplinary actions effectively. This enhancement adds an extra layer of security and control over access permissions.

Crucial information and things to keep in mind for this solution:

Student Information: It is essential to compile thorough information about students, such as contact information, academic background, and personal details. This data might include the student's name, ID, address, phone number, and program enrollment, among other things.

Enrollment and Registration: It's critical to handle the registration process with efficiency. This entails keeping track of academic schedules, details about the semester, and enrolled courses. Prerequisites, credit information, and related lecturers may apply to each course.

Exams and Results: Maintaining a database of exam results linked to students, modules, and degrees may be necessary.

Payment Details: Financial transparency and fee management are ensured by integrating payment data, such as tuition costs, payment dates, and transaction records.

Access Validity and Permissions: Managing access rights to university.

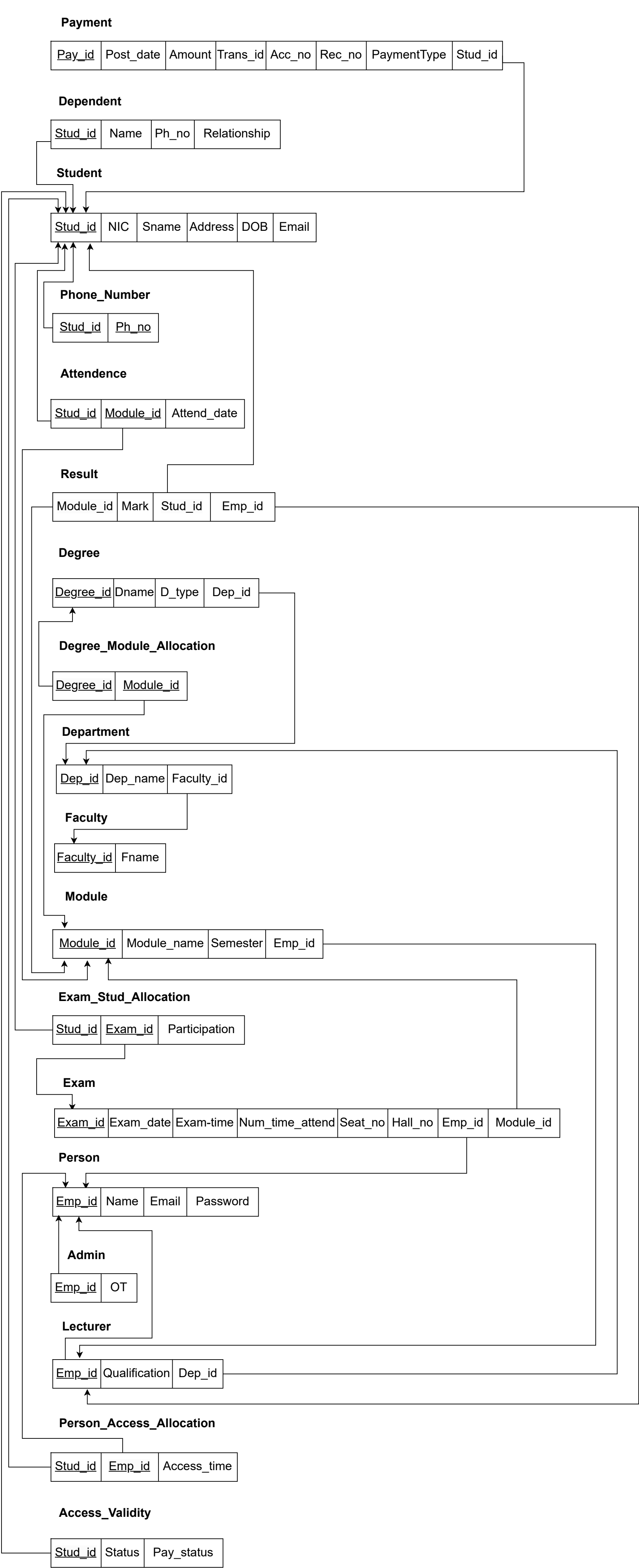
Faculty Information: It is crucial for academic operations to keep a database of lecturers, their departments, contact details, and the modules they teach.

Dependency Data: Maintaining a record of dependencies, such as family members' emergency contact information.

Department and Degree Information: Academic planning and student guidance depend on the definition of departments, their programs, available degrees, courses/modules, and their relationships.

ASSUMPTIONS

- A student has a one dependent only.
- A student has number of phone numbers.
- A student can attempt an online examination several times.
- Online and Physical exams are available.
- Exam can schedule one person only.
- There are many staff in the university, but we considered only admins and lecturers.
- A student can enroll for one degree only.
- A module taught by one lecturer.
- Payment methods are bank deposit and cash only.



DATA NORMALIZATION

1. PAYMENT

1NF

<u>Pay_id</u>	Post_date	Amount	Trans_id	Acc_no	Rec_no	PaymentType	Stud_id
---------------	-----------	--------	----------	--------	--------	-------------	---------

- This is already in 2NF

3NF

PAYMENT

<u>Pay_id</u>	<u>Stud_id</u>	Post_date	Amount	Trans_id	Rec_no	PaymentType
---------------	----------------	-----------	--------	----------	--------	-------------

ACCOUNT DETAILS

<u>Trans_id</u>	<u>Stu_id</u>	Acc_no
-----------------	---------------	--------

2. DEPENDENT

1NF

<u>Stu_id</u>	Name	Ph_no	Relationship
---------------	------	-------	--------------

- This is already in 2NF and 3NF.

3. STUDENT

1NF

<u>Stu_id</u>	NIC	Sname	Address	DOB	Email	Password
---------------	-----	-------	---------	-----	-------	----------

- This is already in 2NF and 3NF.

4. PHONE NUMBER

1NF

<u>Stud_id</u>	<u>Ph_no</u>
----------------	--------------

- This is already in 2NF and 3NF.

5. ATTENDANCE

1NF

<u>Stud_id</u>	<u>Module_id</u>	Attend_date
----------------	------------------	-------------

- This is already in 2NF and 3NF.

6. RESULT

1NF

<u>Stud_id</u>	<u>Module_id</u>	Mark	Emp_id
----------------	------------------	------	--------

- This is already in 2NF and 3NF.

7. DEGREE

1NF

<u>Degree_id</u>	Dname	D_type	Dep_id
------------------	-------	--------	--------

- This is already in 2NF and 3NF.

8. DEGREE_MODULE_ALLOCATION

1NF

<u>Degree_id</u>	<u>Module_id</u>
------------------	------------------

- This is already in 2NF and 3NF.

9. DEPARTMENT

1NF

<u>Dep_id</u>	Dep_name	Faculty_id
---------------	----------	------------

- This is already in 2NF and 3NF.

10. FACULTY

1NF

<u>Faculty_id</u>	Fname
-------------------	-------

- This is already in 2NF and 3NF.

11. MODULE

1NF

<u>Module_id</u>	Module_name	Semester	Emp_id
------------------	-------------	----------	--------

- This is already in 2NF and 3NF.

12. EXAM_STUD_ALLOCATION

1NF

<u>Stud_id</u>	<u>Exam_id</u>	Participation
----------------	----------------	---------------

- This is already in 2NF and 3NF.

13. EXAM

1NF

<u>Stud_id</u>	<u>Module_id</u>	Exam_date	Exam_time	Num_time_attend	Seat_no	Hall_no
----------------	------------------	-----------	-----------	-----------------	---------	---------

- This is already in 2NF and 3NF.

14. EXAM_ALLOCATION

1NF

<u>Exam_id</u>	Module_id	Emp_id
----------------	-----------	--------

- This is already in 2NF and 3NF.

15. PERSON

1NF

<u>Emp_id</u>	Name	Email	Password
---------------	------	-------	----------

- This is already in 2NF and 3NF.

16. ADMIN

1NF

<u>Emp_id</u>	OT
---------------	----

- This is already in 2NF and 3NF.

17. LECTURER

1NF

<u>Emp_id</u>	Qualification	Dep_id
---------------	---------------	--------

- This is already in 2NF and 3NF.

18. PERSON_ACCESS-ALLOCATION

1NF

<u>Stud_id</u>	<u>Emp_id</u>	Access_time
----------------	---------------	-------------

- This is already in 2NF and 3NF.

19. ACCESS_VALIDITY

1NF

<u>Stud_id</u>	Status	Pay_status
----------------	--------	------------

- This is already in 2NF and 3NF.

DATA DICTIONARY

Students Table

Column Name	Data Type	Size	Nullable	Identity	Constraints	Description
StudentID	int		No	Yes	Primary Key	Unique identifier for each student
NIC	VARCHAR	13	Yes	No		National Identity Card number
Name	VARCHAR	200	Yes	No		Name of the student
Address	VARCHAR	200	Yes	No		Address of the student
DOB	VARCHAR	15	Yes	No		Date of Birth of the student
Email	VARCHAR	100	Yes	No		Email address of the student
Password	VARCHAR	45	Yes	No		Password for student's account

Dependent Table

Column Name	Data Type	Size	Nullable	Constraints	Description
StudentID	int		No	REFERENCES Student(StudentID)	Unique identifier for each student
Name	VARCHAR	200	Yes		Name of the dependent
PhoneNo	VARCHAR	10	Yes		Phone number of the dependent
Relationship	VARCHAR	100	Yes		Relationship of the dependent to the student

PhoneNo Table

Column Name	Data Type	Size	Nullable	Constraints	Description
StudentID	int		No	Composite Key	Unique identifier for each student
PhoneNo	VARCHAR	10	No	Composite Key	Phone number of the student

Faculty Table

Column Name	Data Type	Size	Nullable	Constraints	Description
FacultyID	VARCHAR	10	No	Primary Key	Unique identifier for each faculty
Fname	VARCHAR	50	Yes		Name of the faculty

Department Table

Column Name	Data Type	Size	Nullable	Constraints	Description
DepartmentID	Int		No	Primary Key	Unique identifier for each department
DepName	VARCHAR	50	Yes		Name of the department
FacultyID	VARCHAR	10	Yes	REFERENCES Faculty(FacultyID)	Unique identifier for each faculty

Degree Table

Column Name	Data Type	Size	Nullable	Constraints	Description
DegreeID	Int		No	Primary Key	Unique identifier for each degree
Dname	VARCHAR	50	Yes		Name of the degree
Dtype	VARCHAR	20	Yes		The type of the degree program
DepartmentID	int		Yes	REFERENCES Department(DepartmentID)	Unique identifier for each department

Person Table

Column Name	Data Type	Size	Nullable	Constraints	Description
EmpID	Int		No	Primary Key	Unique identifier for each person
Name	VARCHAR	50	Yes		Name of the person
Email	VARCHAR	100	Yes		Email of the person
Password	VARCHAR	45	Yes		Password of the person's account

Lecturer Table

Column Name	Data Type	Size	Nullable	Constraints	Description
EmpID	Int		No	Primary Key REFERENCES Person(EmpID)	Unique identifier for each lecturer
Qualification	VARCHAR	200	Yes		Qualification of the lecturer
DepartmentID	int		Yes	REFERENCES Department(DepartmentID)	Unique identifier for each department

Admin Table

Column Name	Data Type	Size	Nullable	Constraints	Description
EmpID	Int		No	REFERENCES Person(EmpID)	Unique identifier for each admin
OT	int		Yes		OT of admin

Module Table

Column Name	Data Type	Size	Nullable	Constraints	Description
ModuleID	Int		No	Primary Key	Unique identifier for each module
ModuleName	VARCHAR	200	Yes		Name of the module
Semester	VARCHAR	20	Yes		Semester of the module
EmpID	int		Yes	REFERENCES Lecturer(EmpID)	Unique identifier for each lecturer

PersonAccess Table

Column Name	Data Type	Size	Nullable	Constraints	Description
StudentID	Int		No	Composite Key REFERENCES Student(StudentID)	Unique identifier for each student
EmpID	int		No	Composite Key REFERENCES Person(EmpID)	Unique identifier for each person
AccessTime	VARCHAR	20	Yes		Unique identifier for each department

AccessValidity Table

Column Name	Data Type	Size	Nullable	Constraints	Description
StudentID	Int		No	Primary Key REFERENCES Student(StudentID)	Unique identifier for each student
Status	VARCHAR	20	Yes		Student's active status
PayStatus	VARCHAR	20	Yes		The payment status of the students

Attendance Table

Column Name	Data Type	Size	Nullable	Constraints	Description
StudentID	Int		No	Composite Key REFERENCES Student(StudentID)	Unique identifier for each student
ModuleID	int		No	Composite Key REFERENCES Module(ModuleID)	Unique identifier for each module
AttendDate	VARCHAR	20	Yes		Attendance of the students

Payment Table

Column Name	Data Type	Size	Nullable	Constraints	Description
PayID	int		No	Composite Key	Unique identifier for each payment
StudentID	int		No	Composite Key REFERENCES Student(StudentID)	Unique identifier for each student
PostDate	VARCHAR	200	Yes		Payment date
Amount	float		Yes		Payment amount
TransID	Int		Yes	REFERENCES AccountDetails(TransID)	Unique identifier for each transcript
RecNo	int		Yes		Receipt no of the payment
PaymentType	VARCHAR	45	Yes		Type of the payment

AccountDetails Table

Column Name	Data Type	Size	Nullable	Constraints	Description
TransID	Int		No	Primary Key	Unique identifier for each transcript
StudentID	int		Yes	REFERENCES Student(StudentID)	Unique identifier for each student
AccountNo	VARCHAR	20	Yes		Account no of the students

Result Table

Column Name	Data Type	Size	Nullable	Constraints	Description
StudentID	Int		No	Composite Key REFERENCES Student(StudentID)	Unique identifier for each student
ModuleID	int		No	Composite Key REFERENCES Module(ModuleID)	Unique identifier for each module
Mark	float		Yes		Marks of the students
EmpID	int		Yes	REFERENCES Lecturer(EmpID)	Unique identifier for each lecturer

DegreeModuleAllocation Table

Column Name	Data Type	Size	Nullable	Constraints	Description
DegreeID	Int		No	Composite Key REFERENCES Degree(DegreeID)	Unique identifier for each degree
ModuleID	int		No	Composite Key REFERENCES Module(ModuleID)	Unique identifier for each module

ExamAllocation Table

Column Name	Data Type	Size	Nullable	Constraints	Description
ExamID	Int		No	Primary Key	Unique identifier for each exam
ModuleID	int		No	REFERENCES Module(ModuleID)	Unique identifier for each module
EmpID	int		Yes	REFERENCES Person(EmpID)	Unique identifier for each person

ExamStudentAllocation Table

Column Name	Data Type	Size	Nullable	Constraints	Description
ExamID	Int		No	Composite Key REFERENCES ExamAllocation(ExamID)	Unique identifier for each exam
StudentID	int		No	Composite Key REFERENCES Student(StudentID)	Unique identifier for each student
Participation	int		Yes		Participation of the students

Exam Table

Column Name	Data Type	Size	Nullable	Constraints	Description
StudentID	int		No	Composite Key REFERENCES Student(StudentID)	Unique identifier for each student
ModuleID	int		No	Composite Key REFERENCES Module(ModuleID)	Unique identifier for each module
ExamDate	VARCHAR	20	Yes		Exam date
ExamTime	VARCHAR	20	Yes		Exam time
NumTimeAttended	Int		Yes		The time the student used during the exam(Online)
SeatNo	int		Yes		Seat number of students
HallNo	VARCHAR	10	Yes		Hall number of students

MICROSOFT SQL SERVER CREATES TABLE STATEMENTS.

- Student

```
create table Student(  
    StudentID int not null Identity(0,1),  
    NIC varchar(13),  
    Name varchar(200),  
    Address varchar(200),  
    DOB varchar(15),  
    Email varchar(100),  
    Password varchar(45),  
    primary key(StudentID)  
);
```

- Dependent

```
create table Dependent(  
    StudentID int,  
    Name varchar(200),  
    PhoneNo varchar(10),  
    Relationship varchar(100),  
    Foreign key(StudentID)REFERENCES Student(StudentID)  
);
```

- PhoneNo

```
create table PhoneNo(  
    StudentID int,  
    PhoneNo varchar(10),  
    Foreign key(StudentID)REFERENCES Student(StudentID),  
    Primary key(StudentID,PhoneNo)  
);
```

- Faculty

```
create table Faculty(  
    FacultyID varchar(10) not null,  
    Fname varchar(50),  
    Primary key(FacultyID)  
);
```

- Department

```
create table Department(
  DepartmentID int not null,
  DepName varchar(50),
  FacultyID varchar(10),
  Primary key(DepartmentID),
  Foreign key(FacultyID)REFERENCES Faculty(FacultyID)
);
```

- Degree

```
create table Degree(
  DegreeID int not null,
  Dname varchar(50),
  Dtype varchar(20),
  DepartmentID int,
  Primary key(DegreeID),
  Foreign key(DepartmentID)REFERENCES Department(DepartmentID)
);
```

- Person

```
create table Person(
  EmpID int not null,
  Name varchar(50),
  Email varchar(100),
  Password varchar(45),
  Primary key(EmpID)
);
```

- Lecturer

```
create table Lecturer(
  EmpID int not null,
  Qualification varchar(200),
  DepartmentID int,
  Primary key(EmpID),
  Foreign key(DepartmentID)REFERENCES Department(DepartmentID),
  Foreign key(EmpID)REFERENCES Person(EmpID)
);
```

- Admin

```
create table Admin(
  EmpID int not null,
  OT int,
  Foreign key(EmpID)REFERENCES Person(EmpID)
);
```

- Module

```
create table Module(
  ModuleID int not null,
  ModuleName varchar(200),
  Semester varchar(20),
  EmpID int,
  Primary key(ModuleID),
  Foreign key(EmpID)REFERENCES Lecturer(EmpID)
);
```

- PersonAccess

```
create table PersonAccess(
  StudentID int not null,
  EmpID int not null,
  AccessTime varchar(20),
  Primary key(StudentID,EmpID),
  Foreign key(EmpID)REFERENCES Person(EmpID),
  Foreign key(StudentID)REFERENCES Student(StudentID)
);
```

- AccessValidity

```
create table AccessValidity(
  StudentID int not null,
  Status varchar(20),
  PayStatus varchar(20),
  Primary key(StudentID),
  Foreign key(StudentID)REFERENCES Student(StudentID)
);
```

- Attendance

```
create table Attendance(
  StudentID int not null,
  ModuleID int not null,
  AttendDate varchar(20),
  Primary key(StudentID,ModuleID),
  Foreign key(ModuleID)REFERENCES Module(ModuleID),
  Foreign key(StudentID)REFERENCES Student(StudentID)
);
```

- Payment

```
create table Payment(
  PayID int not null,
  StudentID int not null,
  PostDate varchar(20),
  Amount float,
  TransID int,
  RecNo int,
  PaymentType varchar(40),
  Primary key(PayID,StudentID),
  Foreign key(StudentID)REFERENCES Student(StudentID),
  Foreign key(TransID)REFERENCES AccountDetails(TransID)
);
```

- AccountDetails

```
create table AccountDetails(
    TransID int not null,
    StudentID int,
    AccountNo varchar(20),
    Primary key(TransID),
    Foreign key(StudentID)REFERENCES Student(StudentID)
);
```

- Result

```
create table Result(
    StudentID int not null,
    ModuleID int not null,
    Mark float,
    EmpID int,
    Primary key(StudentID,ModuleID),
    Foreign key(ModuleID)REFERENCES Module(ModuleID),
    Foreign key(StudentID)REFERENCES Student(StudentID),
    Foreign key(EmpID)REFERENCES Lecturer(EmpID)
);
```

- DegreeModuleAllocation

```
create table DegreeModuleAllocation(
    DegreeID int not null,
    ModuleID int not null,
    Primary key(DegreeID,ModuleID),
    Foreign key(ModuleID)REFERENCES Module(ModuleID),
    Foreign key(DegreeID)REFERENCES Degree(DegreeID),
);
```

- ExamAllocation

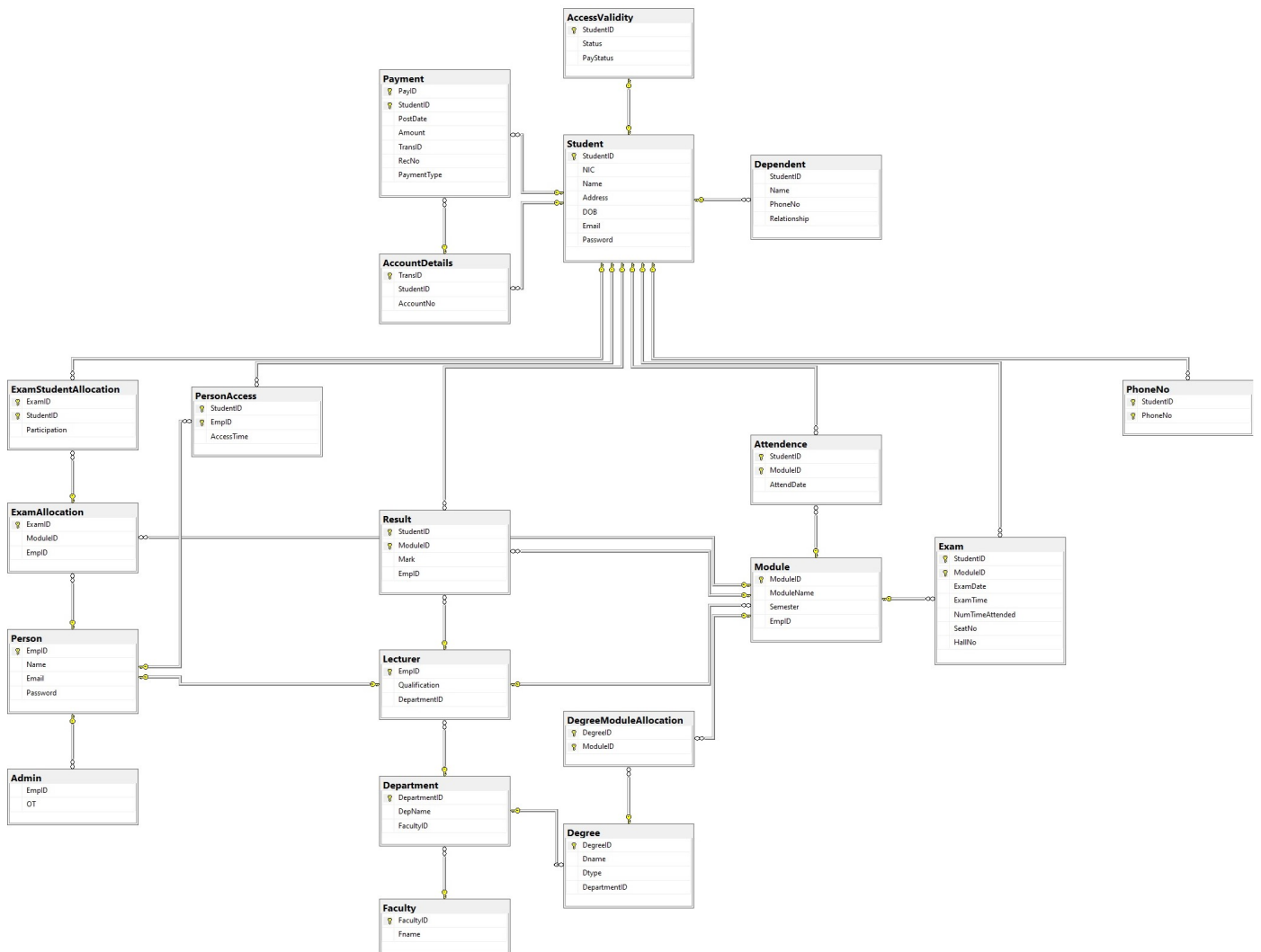
```
create table ExamAllocation(
    ExamID int not null,
    ModuleID int not null,
    EmpID int,
    Primary key(ExamID),
    Foreign key(ModuleID)REFERENCES Module(ModuleID),
    Foreign key(EmpID)REFERENCES Person(EmpID)
);
```

- ExamStudentAllocation

```
create table ExamStudentAllocation(  
  ExamID int not null,  
  StudentID int not null,  
  Participation int,  
  Primary key(ExamID,StudentID),  
  Foreign key(StudentID)REFERENCES Student(StudentID),  
  Foreign key(ExamID)REFERENCES ExamAllocation(ExamID)  
);
```

- Exam

```
create table Exam(  
  StudentID int not null,  
  ModuleID int not null,  
  ExamDate varchar(20),  
  ExamTime varchar(20),  
  NumTimeAttended int,  
  SeatNo int,  
  HallNo varchar(10),  
  Primary key(StudentID,ModuleID),  
  Foreign key(ModuleID)REFERENCES Module(ModuleID),  
  Foreign key(StudentID)REFERENCES Student(StudentID)  
);
```



TABLES

- Students Table

Results		Messages					
	StudentID	NIC	Name	Address	DOB	Email	Password
1	1	200218500212	Chanuda	564/1,Piliyandala Rd,Maharagama	2002/02/03	chanuda@gmail.com	1234
2	2	200218546212	Dinethmin	564/1,Piliyandala Rd,Maharagama	2002/08/12	Dinethmin@gmail.com	1234
3	4	200218545134	John Doe	123 Main St	2002/12/12	john.doe@example.com	securepassword
4	7	200018500132	Kamal	Colombo 07	2000/12/03	kamal@gmail.com	224466
5	8	200224350132	Chaminda	07/2 Maharagama, Pannipitiya	2002/05/20	Chaminda@gmail.com	467823
6	9	200468210132	Sunil	Colombo 05	2004/02/04	Sunil@gmail.com	33465
7	10	200114625125	Supun	542/D Kaluthara	2001/10/13	Supun@gmail.com	2468
8	11	200246782546	Manel	12/A Mathara	2002/06/07	Manel@gmail.com	2244
9	12	200046821654	Tharushi	Colombo 07	2000/04/14	Tharushi@gmail.com	33166
10	13	200176558246	Sadun	18/2 Kasbswa	2001/08/19	Sadun@gmail.com	1234567
11	14	200349853454	Oshara	Colombo 07	2003/11/01	Oshara@gmail.com	298246

- Dependent Table

Results		Messages		
	StudentID	Name	PhoneNo	Relationship
1	2	John	1234567890	Father
2	7	Doe	9876543210	Father
3	1	Manjula	5551234567	Mother
4	9	Alice	7778889999	Mother
5	12	Alex	4445556666	Father
6	4	Johan	1234567890	Father
7	11	Chaminda	9876543210	Father
8	10	Manoja	5551234567	Mother
9	8	Nayana	7778889999	Mother
10	13	Dhanidu	4445556666	Father

- PhoneNo Table

Results		Messages	
	StudentID	PhoneNo	
1	1	0762035507	
2	1	1234567890	
3	1	5551234567	
4	2	0762235507	
5	2	1112233445	
6	2	9876543210	
7	4	4445556666	
8	4	7778889999	
9	7	9998887776	
10	8	3332221110	
11	9	6667778888	
12	10	2223334444	

- Faculty Table

Results Messages		
	FacultyID	Fname
1	FOB	Faculty of Business
2	FOC	Faculty of Computing
3	FOE	Faculty of Engineering
4	FOP	Faculty of Postgraduate Studies
5	FOS	Faculty of Science

- Department Table

Results Messages			
	DepartmentID	DepName	FacultyID
1	1	Department of Computer and Data Science	FOC
2	2	Department of Network and Security	FOC
3	3	Department of Software Engineering	FOC
4	4	Department of Information and Systems Sciences	FOC
5	5	Department of Mechatronic and Industry Engineering	FOE
6	6	Department of Design Studies	FOE
7	7	Department of Biomedical Science	FOS
8	8	Department of Management	FOB
9	9	Department of Accounting and Finance	FOB
10	10	Department of Marketing and Tourism	FOB
11	11	Department of Operations and Logistics	FOB
12	12	Department of English and Modern Languages	FOB

- Degree Table

Results Messages				
	DegreeID	Dname	Dtype	DepartmentID
1	1	BSC (HONS) COMPUTER SCIENCE	PLYMOUTH	1
2	2	BSC (HONS) COMPUTER SECURITY	PLYMOUTH	2
3	3	BSC (HONS) IN SOFTWARE ENGINEERING	PLYMOUTH	3
4	4	BSC (HONS) TECHNOLOGY MANAGEMENT	PLYMOUTH	4
5	5	BENG (HONS) CIVIL AND STRUCTURAL ENGINEERING	PLYMOUTH	5
6	6	BACHELOR OF INTERIOR DESIGN	UGC	6
7	7	BSC (HONS) PSYCHOLOGY	PLYMOUTH	7
8	8	BSC (HONS) IN SOFTWARE ENGINEERING	UGC	3
9	9	BM (HONS) IN ACCOUNTING AND FINANCE	UGC	9
10	10	BSC (HONS) COMPUTER NETWORKS	UGC	2
11	11	BSC (HONOURS) IN DATA SCIENCE	PLYMOUTH	1
12	12	BACHELOR OF INFORMATION TECHNOLOGY	VICTORIA	2

- Person Table

	Results	Messages		
	EmpID	Name	Email	Password
1	1	John Doe	john.doe@example.com	password1
2	2	Jane Smith	jane.smith@example.com	password2
3	3	Bob Johnson	bob.johnson@example.com	password3
4	4	Alice Brown	alice.brown@example.com	password4
5	5	Charlie Davis	charlie.davis@example.com	password5
6	6	Eva White	eva.white@example.com	password6
7	7	David Lee	david.lee@example.com	password7
8	8	Sophie Miller	sophie.miller@example.com	password8
9	9	Michael Wilson	michael.wilson@example.com	password9
10	10	Olivia Harris	olivia.harris@example.com	password10
11	11	Sophia Davis	sophia.davis@example.com	password11
12	12	Liam Wilson	liam.wilson@example.com	password12
13	13	Emma Brown	emma.brown@example.com	password13
14	14	Mia Johnson	mia.johnson@example.com	password14
15	15	Aiden Taylor	aiden.taylor@example.com	password15
16	16	Ella White	ella.white@example.com	password16
17	17	Carter Anders...	carter.anderson@example.c...	password17
18	18	Aria Martinez	aria.martinez@example.com	password18
19	19	Lucas Smith	lucas.smith@example.com	password19
20	20	Isabella Davis	isabella.davis@example.com	password20

- Lecturer Table

	Results	Messages		
	EmpID	Qualification	DepartmentID	
1	1	Ph.D. in Computer Science	1	
2	2	M.Sc. in Electrical Engineering	2	
3	3	Ph.D. in Psychology	3	
4	4	M.A. in English Literature	4	
5	5	Ph.D. in Business Administration	5	
6	6	M.Sc. in Physics	6	
7	7	Ph.D. in Mechanical Engineering	7	
8	8	M.A. in Sociology	8	
9	9	Ph.D. in Economics	9	
10	10	M.Sc. in Computer Networks	10	
11	11	Ph.D. in Data Science	1	
12	12	M.Sc. in Information Technology	2	

- Admin Table

	EmpID	OT
1	10	1
2	11	0
3	12	1
4	13	1
5	14	1
6	15	0
7	16	3
8	17	0
9	18	2
10	19	0
11	20	2

- Module Table

	ModuleID	ModuleName	Semester	EmpID
1	1	Introduction to Programming	1	1
2	2	Database Management	2	2
3	3	Web Development	2	3
4	4	Data Structures and Algorithms	1	4
5	5	Artificial Intelligence	2	5
6	6	Software Engineering	1	6
7	7	Network Security	1	7
8	8	Mobile App Development	2	8
9	9	Human-Computer Interaction	1	9
10	10	Machine Learning	2	10

- PersonAccess Table

	StudentID	EmpID	AccessTime
1	1	2	2024-01-14 09:15:00
2	2	1	2024-01-14 08:00:00
3	4	3	2024-01-14 10:30:00
4	7	7	2024-01-14 15:30:00
5	8	6	2024-01-14 14:15:00
6	9	8	2024-01-14 16:45:00
7	10	5	2024-01-14 13:00:00
8	12	4	2024-01-14 11:45:00

- AccessValidity Table

	StudentID	Status	PayStatus
1	1	Active	Paid
2	2	Active	Not Paid
3	4	Active	Paid
4	7	Deactive	Not Paid
5	8	Active	Paid
6	9	Deactive	Not Paid
7	10	Active	Paid
8	12	Deactive	Not Paid

- Attendance Table

	StudentID	ModuleID	AttendDate
1	1	1	80
2	1	2	90
3	2	1	79
4	2	3	75
5	4	2	60
6	7	3	83
7	7	4	100
8	8	4	95
9	9	5	100
10	10	5	66

- Payment Table

	PayID	StudentID	PostDate	Amount	TransID	RecNo	Payment Type
1	1	1	2024-01-25	500	1	NULL	Bank
2	2	2	2024-01-26	750	2	NULL	Bank
3	3	7	2024-01-27	600	3	NULL	Bank
4	4	4	2024-01-28	800	4	NULL	Bank
5	5	8	2024-01-29	700	5	NULL	Bank
6	6	11	2024-01-30	550	NULL	1	Cash
7	7	12	2024-01-31	900	NULL	3	Cash
8	8	13	2024-02-01	650	NULL	2	Cash
9	9	9	2024-02-02	700	NULL	4	Cash
10	10	10	2024-02-03	850	NULL	5	Cash

- AccountDetails Table

	TransID	StudentID	AccountNo
1	1	1	12345678
2	2	2	23456789
3	3	7	34567890
4	4	4	45678901
5	5	8	56789012

- Result Table

	StudentID	ModuleID	Mark	EmpID
1	1	1	95	1
2	1	2	75	2
3	2	1	88	1
4	2	3	80	3
5	4	2	92	2
6	4	3	87	3
7	7	4	93	4
8	8	4	78	4
9	9	5	85	5
10	10	5	90	5

- DegreeModuleAllocation Table

	DegreeID	ModuleID
1	1	1
2	1	2
3	1	3
4	1	4
5	1	5
6	2	6
7	2	7
8	3	6
9	3	8
10	4	9
11	5	10

- ExamAllocation Table

	Results	Messages	
	ExamID	ModuleID	EmpID
1	1	1	1
2	2	2	20
3	3	3	9
4	4	4	4
5	5	5	14
6	6	6	16
7	7	7	11
8	8	8	4
9	9	9	6
10	10	10	13

- ExamStudentAllocation Table

	Results	Messages	
	ExamID	StudentID	Participation
1	1	1	1
2	1	2	1
3	1	4	0
4	2	4	1
5	2	7	1
6	2	8	0
7	3	7	1
8	3	8	1
9	3	9	0
10	4	10	1

- Exam Table

Results

Messages

	StudentID	ModuleID	ExamDate	ExamTime	NumTimeAttended	SeatNo	HallNo
1	1	1	2024-01-15	09:00 AM	3	NULL	NULL
2	2	2	2024-01-16	02:00 PM	3	NULL	NULL
3	4	3	2024-01-17	11:00 AM	3	NULL	NULL
4	7	4	2024-01-18	10:00 AM	3	NULL	NULL
5	8	5	2024-01-19	01:00 PM	NULL	13	Hall5
6	9	6	2024-01-20	03:00 PM	NULL	20	Hall6
7	10	7	2024-01-21	12:00 PM	NULL	60	Hall7
8	11	9	2024-01-23	02:30 PM	NULL	63	Hall9
9	12	8	2024-01-22	04:00 PM	NULL	40	Hall8
10	13	10	2024-01-24	09:30 AM	NULL	24	Hall10

CREATE TRIGGER

1. Insert trigger.

```
create trigger student_insert
on Student after insert
as
begin
    select Name,NIC,Address,DOB,Email from Student
end;
```

```
--trigger1
insert into Student
values('200111220137','Mendis','Colombo 07','2001/04/22','mendis@gmail.com','hi123');
```

	Name	NIC	Address	DOB	Email
1	Chanuda	200218500212	564/1,Piliyandala Rd,Maharagama	2002/02/03	chanuda@gmail.com
2	Dinethmin	200218546212	564/1,Piliyandala Rd,Maharagama	2002/08/12	Dinethmin@gmail.com
3	John Doe	200218545134	123 Main St	2002/12/12	john.doe@example.com
4	Kamal	200018500132	Colombo 07	2000/12/03	kamal@gmail.com
5	Chaminda	200224350132	07/2 Maharagama, Pannipitiya	2002/05/20	Chaminda@gmail.com
6	Sunil	200468210132	Colombo 05	2004/02/04	Sunil@gmail.com
7	Supun	200114625125	542/D Kaluthara	2001/10/13	Supun@gmail.com
8	Manel	200246782546	12/A Mathara	2002/06/07	Manel@gmail.com
9	Tharushi	200046821654	Colombo 07	2000/04/14	Tharushi@gmail.com
10	Sadun	200176558246	18/2 Kasbswa	2001/08/19	Sadun@gmail.com
11	Oshara	200349853454	Colombo 07	2003/11/01	Oshara@gmail.com
12	Mendis	200111220137	Colombo 07	2001/04/22	mendis@gmail.com

2. Update trigger

```
create trigger student_update
on Student after update
as
begin
    select StudentID,Name,NIC,Address,DOB,Email from Student
end;
```

```
--trigger2
update Student set Address = 'Maharagama' where StudentID =15;
```

	StudentID	Name	NIC	Address	DOB	Email
1	1	Chanuda	200218500212	564/1,Piliyandala Rd,Maharagama	2002/02/03	chanuda@gmail.com
2	2	Dinethmin	200218546212	564/1,Piliyandala Rd,Maharagama	2002/08/12	Dinethmin@gmail.com
3	4	John Doe	200218545134	123 Main St	2002/12/12	john.doe@example.com
4	7	Kamal	200018500132	Colombo 07	2000/12/03	kamal@gmail.com
5	8	Chaminda	200224350132	07/2 Maharagama, Pannipitiya	2002/05/20	Chaminda@gmail.com
6	9	Sunil	200468210132	Colombo 05	2004/02/04	Sunil@gmail.com
7	10	Supun	200114625125	542/D Kaluthara	2001/10/13	Supun@gmail.com
8	11	Manel	200246782546	12/A Mathara	2002/06/07	Manel@gmail.com
9	12	Tharushi	200046821654	Maharagama	2000/04/14	Tharushi@gmail.com
10	13	Sadun	200176558246	18/2 Kasbswa	2001/08/19	Sadun@gmail.com
11	14	Oshara	200349853454	Colombo 07	2003/11/01	Oshara@gmail.com
12	15	Mendis	200111220137	Maharagama	2001/04/22	mendis@gmail.com

CREATE VIEW

1. Module and Degree

```
create view ModuleandDegree
as
select ModuleName,DegreeID from Module
join DegreeModuleAllocation on Module.ModuleID = DegreeModuleAllocation.ModuleID;
```

	ModuleName	DegreeID
1	Introduction to Programming	1
2	Database Management	1
3	Web Development	1
4	Data Structures and Algorithms	1
5	Artificial Intelligence	1
6	Software Engineering	2
7	Network Security	2
8	Software Engineering	3
9	Mobile App Development	3
10	Human-Computer Interaction	4
11	Machine Learning	5

2. Student Pay

```
create view studentPay
as
select StudentID,PostDate as Payment_Date,Amount,PaymentType
from Payment;
```

	StudentID	Payment_Date	Amount	Payment Type
1	1	2024-01-25	500	Bank
2	2	2024-01-26	750	Bank
3	7	2024-01-27	600	Bank
4	4	2024-01-28	800	Bank
5	8	2024-01-29	700	Bank
6	11	2024-01-30	550	Cash
7	12	2024-01-31	900	Cash
8	13	2024-02-01	650	Cash
9	9	2024-02-02	700	Cash
10	10	2024-02-03	850	Cash

CREATE PROCEDURE

1. Add Phone No

```
CREATE PROCEDURE AddPhoneNo(  
    @StudentID int,  
    @PhoneNo varchar(255)  
) AS  
begin  
    insert into PhoneNo(StudentID,PhoneNo)  
    values (@StudentID,@PhoneNo);  
end;  
EXEC addPhoneNo  
    @StudentID = 2,  
    @PhoneNO = '0762235123';
```

2. Add New Student

```
create procedure AddNewStudent1(  
    @nic varchar(255),  
    @name varchar(255),  
    @address varchar(255),  
    @dob varchar(255),  
    @email varchar(255),  
    @password varchar(255)  
) AS  
begin  
    insert into Student(NIC,name,Address,DOB,Email>Password)  
    values(@nic,@name,@address,@dob,@email,@password);  
end;  
EXEC AddNewStudent1  
    @nic = '200218500212',  
    @name = 'Jhohan Doe',  
    @address = '456 Main St',  
    @dob = '2002/10/09',  
    @email = 'jhohandoe@gmail.com',  
    @password = 'password';
```

133 %

Results					
	Name	NIC	Address	DOB	Email
1	Chanuda	200218500212	564/1,Piliyandala Rd,Maharagama	2002/02/03	chanuda@gmail.com
2	Dinethmin	200218546212	564/1,Piliyandala Rd,Maharagama	2002/08/12	Dinethmin@gmail.com
3	John Doe	200218545134	123 Main St	2002/12/12	john.doe@example.com
4	Kamal	200018500132	Colombo 07	2000/12/03	kamal@gmail.com
5	Chaminda	200224350132	07/2 Maharagama, Pannipitiya	2002/05/20	Chaminda@gmail.com
6	Sunil	200468210132	Colombo 05	2004/02/04	Sunil@gmail.com
7	Supun	200114625125	542/D Kaluthara	2001/10/13	Supun@gmail.com
8	Manel	200246782546	12/A Mathara	2002/06/07	Manel@gmail.com
9	Tharushi	200046821654	Maharagama	2000/04/14	Tharushi@gmail.com
10	Sadun	200176558246	18/2 Kasbswa	2001/08/19	Sadun@gmail.com
11	Oshara	200349853454	Colombo 07	2003/11/01	Oshara@gmail.com
12	Mendis	200111220137	Maharagama	2001/04/22	mendis@gmail.com
13	Jhohan Doe	200218500212	456 Main St	2002/10/09	jhohandoe@gmail.com

CREATE FUNCTION

1.

```
CREATE FUNCTION Find_Name (@id int)
RETURNS varchar(50)
AS
BEGIN
    DECLARE @name varchar(50)

    SELECT @name = Name
    FROM student
    WHERE StudentID = @id

    RETURN @name
END
```

2.

```
CREATE FUNCTION dbo.GetTotalAmountPaid(@Stud_id INT)
RETURNS MONEY
AS
BEGIN
    DECLARE @TotalAmount MONEY
    SELECT @TotalAmount = SUM(Amount)
    FROM Payment
    WHERE StudentID = @Stud_id
    RETURN @TotalAmount
END;
```

Call,

```
--User DF1
DECLARE @StudentName VARCHAR(50);
SELECT @StudentName = dbo.Find_Name(1);

PRINT 'Student Name: ' + @StudentName;

--User DF2
DECLARE @Total VARCHAR(50);
SELECT @Total = dbo.GetTotalAmountPaid(2);

PRINT 'Payment Total: ' + @Total;
```

121 %

Messages

Student Name: Chanuda
Payment Total: 750.00

Completion time: 2024-01-14T19:54:22.0566588+05:30

CRITICAL APPRAISAL OF THE SOLUTION

We have made this database for the university administrative management system. There are several procedures that we are using to perform this. I will explain it for you.

- Firstly, As an Entity, we have used "PAYMENT". It has several attributes called as, 'pay_id', 'post_date', 'amount', 'Trans_id', 'Acc_no', 'Rec_no', 'PaymentType' and 'Stud_id'. As the foreign key (Stud_id). It depends on the 'Stud_id' in the 'STUDENT' table.
- Then, as the entity "Dependent". It has several attributes called as, 'stud_id', 'Name', 'ph_no', 'Relationship'. And, As the primary key (Stud_id) it depends on the 'Stud_id' in the 'STUDENT' table.
- As the entity "STUDENT". It also has several attributes called as, 'Stud_id', 'NIC', 'Sname', 'Address', 'DOB' and 'Email'.
- As the entity "Phone_Number". It also has several attributes called as, 'Stud_id' and 'ph_no'. They are working as a composite key.
- As the entity "ATTENDANCE". It also has several attributes called as, 'Stud_id', 'Module_id' and 'Attend_date'. And 'Stud_id', 'Module_id' work as the Composite keys. 'Module_id' depending on the 'Module_id' in the "MODULE" table.
- As the entity "RESULT". It also has several attributes called as, 'Module_id', 'Mark', 'Stud_id' and 'Emp_id'. And 'Module_id' depending on the 'Module_id' "MODULE" table, 'Stud_id' depends on the 'Stud_id' in the 'STUDENT' table.
- As the entity "DEGREE". It also has several attributes called as, 'Degree_id', 'Dname', 'D_type' and 'Dep_id'. And 'Dep_id' depending on the 'Dep_id' in the "DEPARTMENT" table.
- As the entity "DEGREE_MODULE_ALLOCATION". It also has several attributes called as, 'Degree_id' and 'Module_id'. They are working as the composite key. 'Degree_id' is dependent on 'Degree_id' in the "DEGREE" table and 'Module_id' is depending on the 'Module_id' in the "MODULE" table.
- As the entity "DEPARTMENT". It also has several attributes called as, 'Dep_id', 'Dep_name' and 'Faculty_id'. And 'Faculty_id' depends on the 'Faculty_id' in the "FACULTY" table.

- As the entity "FACULTY". It also has several attributes called as, 'Faculty_id' and 'Fname'.
- As the entity "MODULE". It also has several attributes called as, 'Module_id', 'Module_name', 'Semester' and 'Emp_id'. And 'Emp_id' is dependent on 'Emp_id' in the "LECTURER" table.
- As the entity "EXAM_STUD_ALLOCATION". It also has several attributes called as, 'Stud_id', 'Exam_id' and 'Participation'. 'Stud_id', 'Exam_id' both working as the composite key. 'Stud_id' depending on the 'Stud_id' in the "STUDENT" table. 'Exam_id' is also depending on the 'Exam_id' in the "EXAM" table.
- As the entity "EXAM". It also has several attributes called as, 'Exam_id', 'Exam_date', 'Exam-time', 'Num_time_attend', 'Seat_no', 'Hall_no', 'Emp_id' and 'Module_id'. And also 'Emp_id' is depending on the 'Emp_id' in the "PERSON" table and 'Module_id' is depending on the 'Module_id' is depending on the 'Module_id' in the "MODULE" table.
- As the entity "PERSON". It also has several attributes called as, 'Emp_id', 'Name', 'Email', and 'Password'.
- As the entity "ADMIN". It also has several attributes called, 'Emp_id' and 'OT'. And 'Emp_id' depends on the 'Emp_id' in the "PERSON" table.
- As the entity "LECTURER". It also has several attributes called, 'Emp_id', 'Qualification', and 'Dep_id'. And 'Emp_id' is depending on the, 'Emp_id' in the "PERSON" table and 'Dep_id' is depending on the 'Dep_id' in the "DEPARTMENT" table.
- As the entity "PERSON_ACCESS-ALLOCATION". It also has several attributes called as, 'Stud_id', 'Emp_id' and 'Access_time'. And 'Stud_id' is dependent on 'Stud_id' in the "STUDENT" table and 'Emp_id' is dependent on 'Emp_id' in the "PERSON" table.
- As the entity "ACCESS_VALIDITY". It also has several attributes called as, 'Stud_id', 'Status' and 'Pay_status'. And 'Stud_id' depends on the 'Stud_id' in the "STUDENT" table.

FUTURE IMPLEMENTATION

In future, we hope to add some new implementation for this University Database Management System. As a result of it, we have got an idea for adding new operation using the University Library.

In here we mainly focus on Library Resources, User Registration, Book Availability, Ordering Books, Borrowing Books, Returning Books, Late Return and Payments and Integration with University DBMS. Those are the key components and considerations for implementing the library operation in DBMS.

- The first users must register with the library.
- After registration, the user can see the physical books and e-resources in the database.
- Users got permission to place orders for physical books and access to e-resources.
- Users can see the availability status in the physical books.
- Late return users must be fined for late returns based on predefined rules. (daily fine rates).
- Share relevant information with the University DBMS.

By implementing these features in University DBMS, it can effectively handle the above key consideration and providing a comprehensive solution for managing both physical and e-resource libraries.