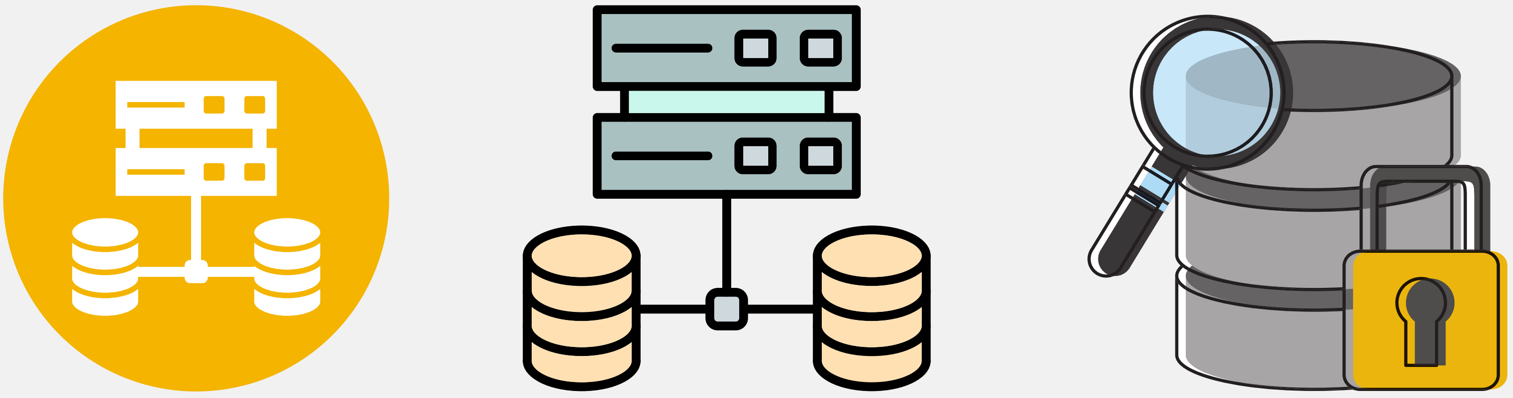
JUNE, 2022

Data Base Project

FYP Management System



Sherwan Qadir
20i-0689

Mutharib Ayub
20i-0476

Abdullah Umar
20i-0444

INTRO:

We were assigned to design and develop a comprehensive web-based system to better facilitates final year projects and the users; like FYP Committee, Project Supervisors, Panel Members, and the Project Group Members (Students). It comes up with different user roles. As there are different users involved so user management is the basic thing. So talking about the workflow of our assignment we started on with designing ER diagram which matches and provides all basic entities and attributes which we have to use in design of our system. Deciding all key entities, attributes and relations between them made it easy for us to keep up with workflow and smoothly manage the assignment. We defined different user roles and different interface for all the users of the system.

Four views created are:

- FYP Committee Interface.
- Panel Members Interface
- Project Supervisors Interface
- Students Interface

WORK DIVISION:

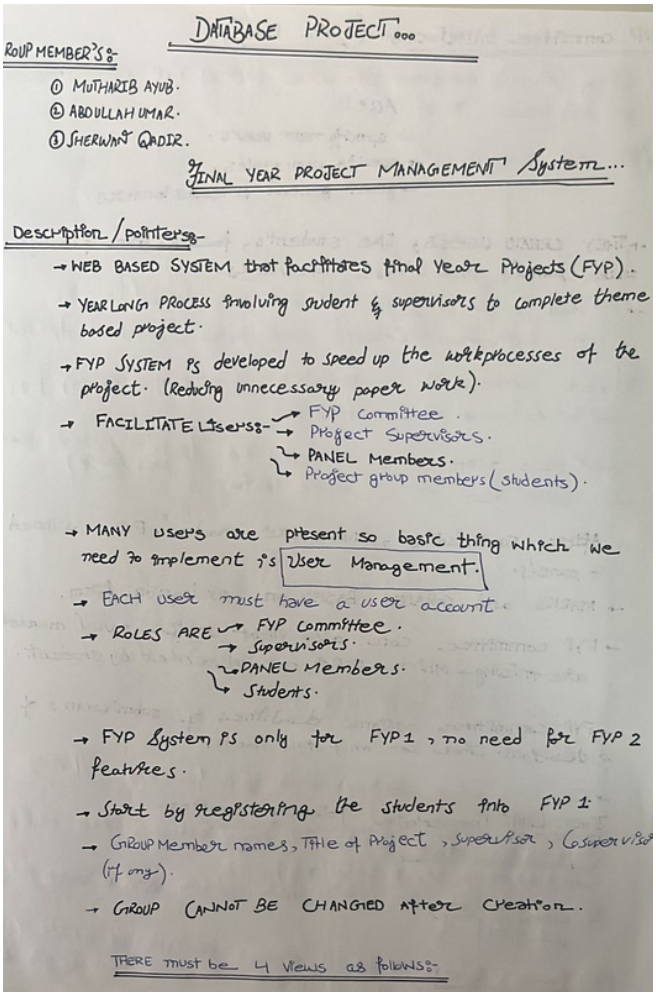
There were three main parts of the project including front end implementation (ASP.NET, HTML), back end (C#) and Data Base (making tables and constraints.

Work was divided as follows:

- Front End: Sherwan Qadir
- Back End: Abdullah Umar
- Data Base: Mutharib Ayub

WORKFLOW AND DESIGN IDEA:

Description:



DATABASE PROJECT...

ROUP MEMBER'S:-

① MUTHARIB AYUB.
② ABDULLAH UMAR.
③ SHERWANI QADIR.

FINAL YEAR PROJECT MANAGEMENT System...

Description / pointers:-

→ WEB BASED SYSTEM that facilitates final Year Projects (FYP).

→ YEAR LONG PROCESS involving student & supervisors to complete theme based project.

→ FYP SYSTEM is developed to speed up the workprocesses of the project. (Reducing unnecessary paper work).

→ FACILITATE Users:-
- → FYP Committee.
- → Project Supervisors.
- → PANEL Members.
- → Project group members (students).

→ MANY users are present so basic thing which we need to implement is $\boxed{\text{User Management.}}$

→ EACH user must have a user account.

→ ROLES ARE
- → FYP Committee.
- → Supervisors.
- → PANEL Members.
- → Students.

→ FYP System is only for FYP 1, no need for FYP 2 features.

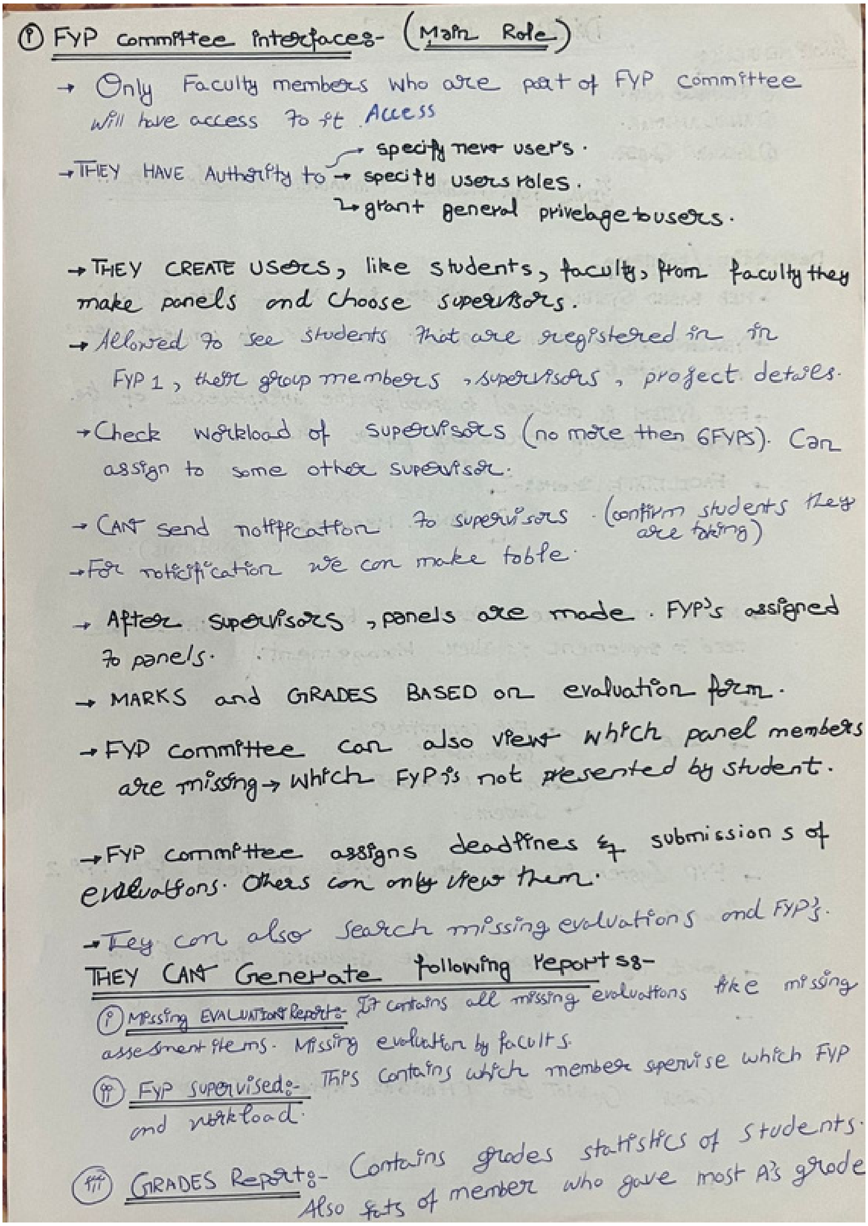
→ Start by registering the students into FYP 1.

→ Group Member names, Title of Project, Supervisor, Cosupervis (if any).

→ GROUP CANNOT BE CHANGED after Creation.

THERE must be 4 Views as follows:-

FYP Committee Interface Requirements :



(i) FYP committee interfaces:- (Main Role)

→ Only Faculty members who are part of FYP committee will have access to it. Access

→ THEY HAVE Authority to
- → specify new user's.
- → specify users roles.
- → grant general privelage to users.

→ THEY CREATE users, like students, faculty, from faculty they make panels and choose supervisors.

→ Allowed to see students that are registered in in FYP 1, their group members, supervisors, project detailes.

→ Check workload of supervisors (no more then 6FYPS). Can assign to some other supervisor.

→ CAN send notification to supervisors. (confirm students they are taking)

→ For notification we can make table.

→ After supervisors, panels are made. FYP's assigned to panels.

→ MARKS and GRADES BASED on evaluation form.

→ FYP committee can also view which panel members are missing → which FYP is not presented by student.

→ FYP committee assigns deadlines & submissions of evaluations. Others can only view them.

→ They can also search missing evaluations and FYP's.

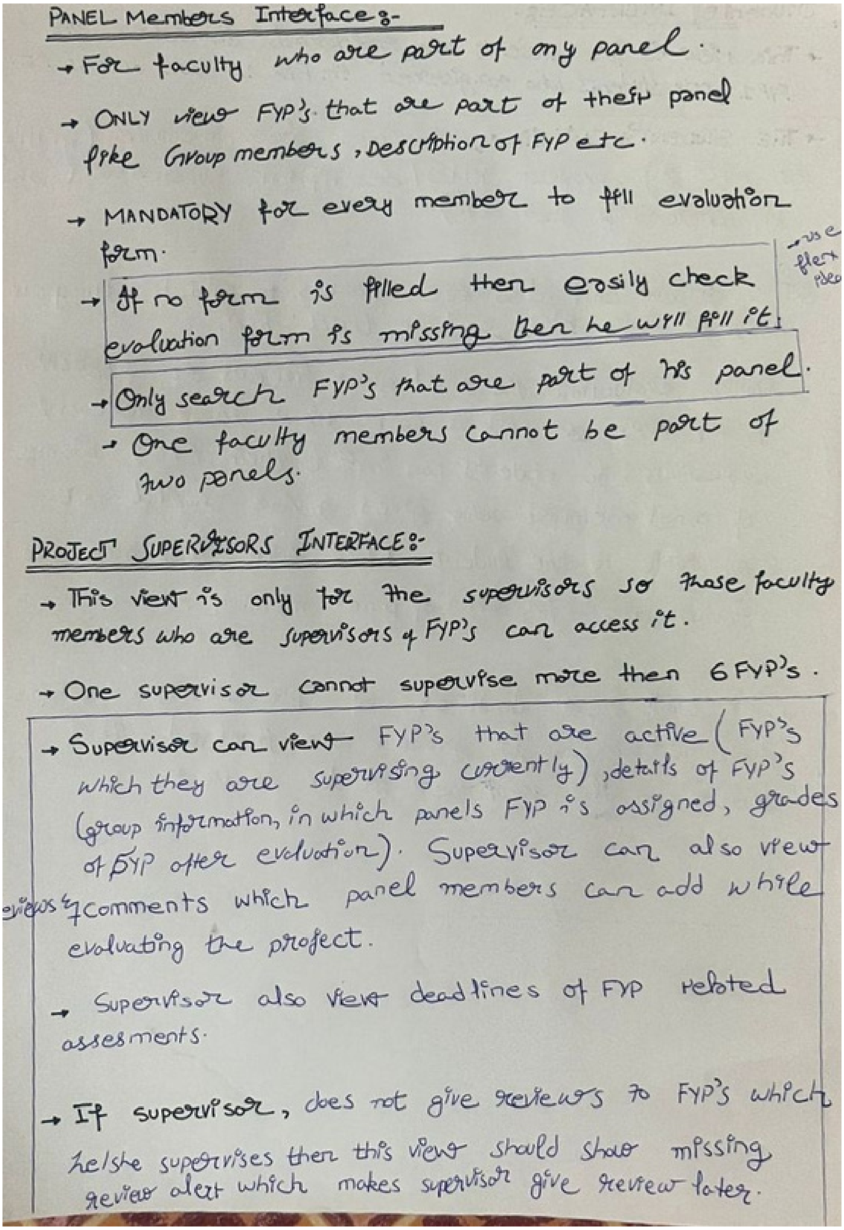
THEY CAN Generate following reports:-

(i) Missing EVALUATION Report:- It contains all missing evaluations like missing assesment items. Missing evaluation by faculty.

(ii) FYP supervised:- This contains which member supervise which FYP and workload.

(iii) GRADES Report:- Contains grades statistics of students. Also stats of member who gave most A's grade

Panel Member and Project Supervisor Interface Requirements:



PANEL Members Interface:-

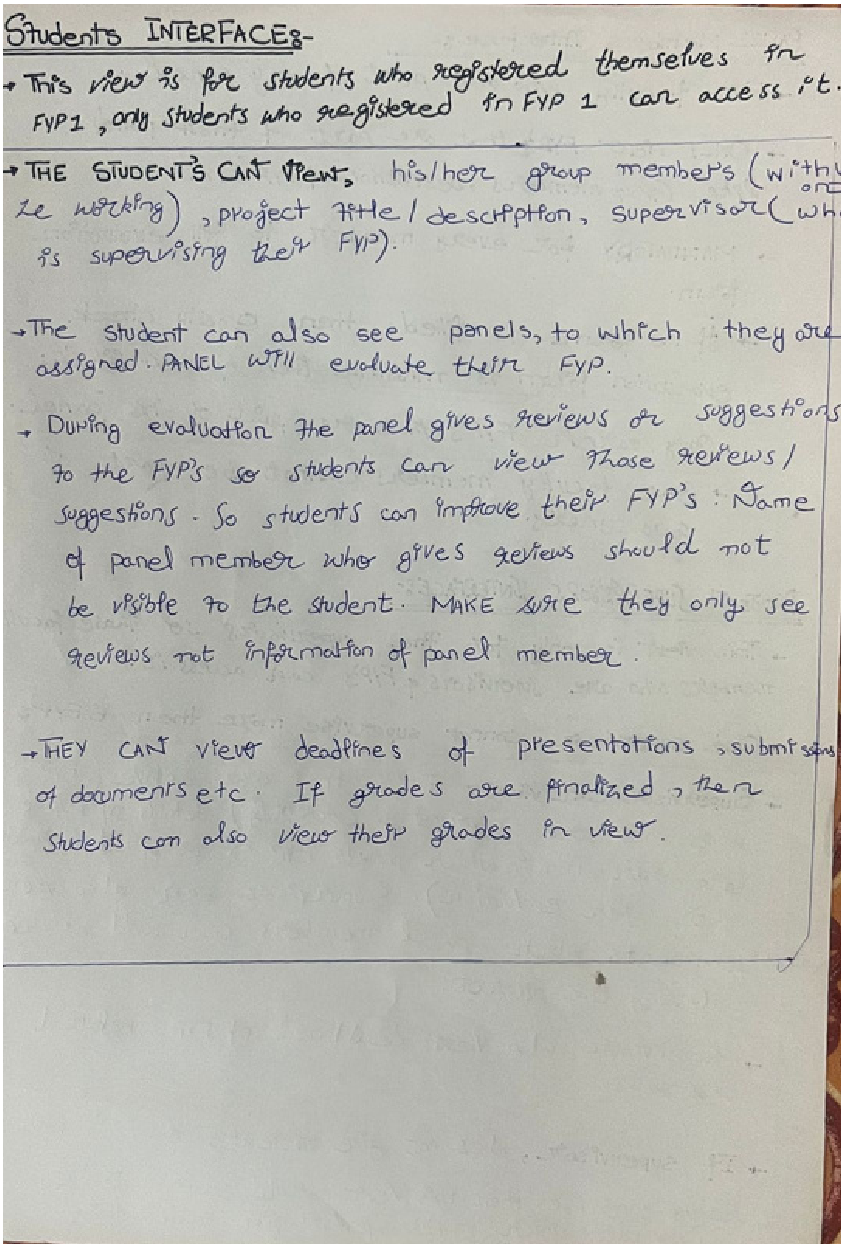
- For faculty who are part of any panel.
- ONLY view FYP's that are part of their panel like Group members, Description of FYP etc.
- MANDATORY for every member to fill evaluation form.
- If no form is filled then easily check evaluation form is missing then he will fill it.
- Only search FYP's that are part of his panel.
- One faculty members cannot be part of two panels.

→ use flex [illegible]

PROJECT SUPERVISORS INTERFACE:-

- This view is only for the supervisors so those faculty members who are supervisors of FYP's can access it.
- One supervisor cannot supervise more then 6 FYP's.
- Supervisor can view FYP's that are active (FYP's which they are supervising currently), details of FYP's (group information, in which panels FYP is assigned, grades of FYP after evaluation). Supervisor can also view reviews & comments which panel members can add while evaluating the project.
- Supervisor also view deadlines of FYP related assesments.
- If supervisor, does not give reviews to FYP's which he/she supervises then this view should show missing review alert which makes supervisor give review later.

Student Interface Requirements:



Students INTERFACE:-

→ This view is for students who registered themselves in FYP1, only students who registered in FYP 1 can access it.

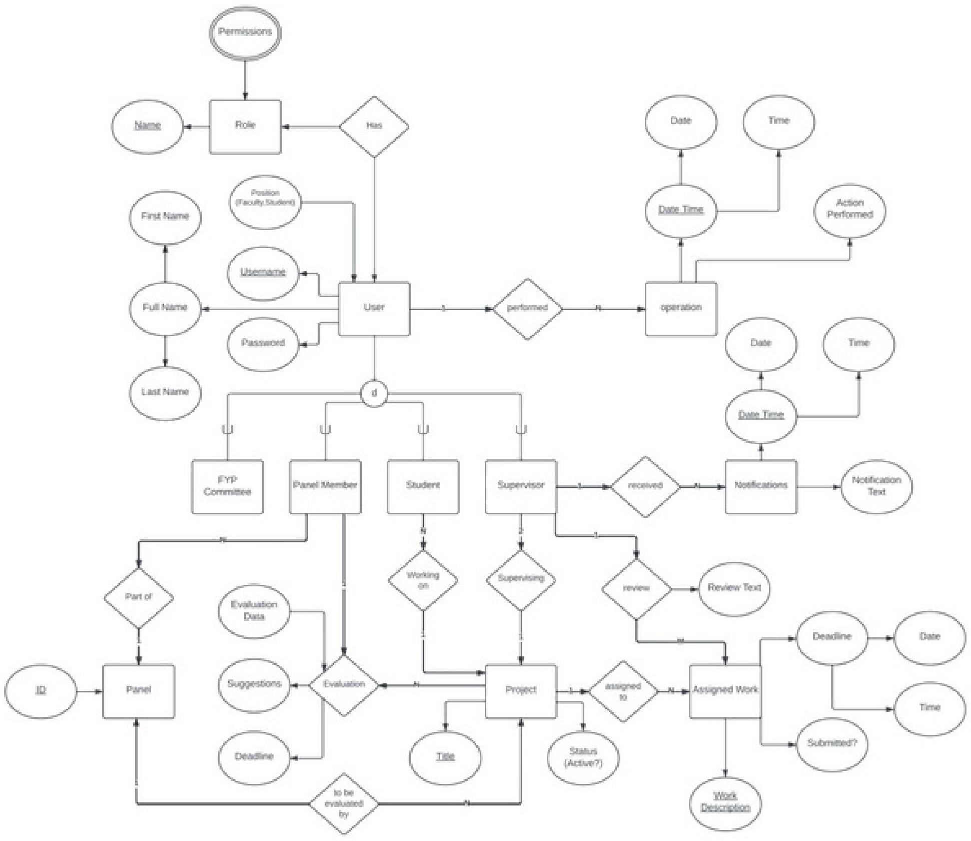
→ THE STUDENT'S CAN VIEW, his/her group members (with
are working), project title / description, supervisor (wh
is supervising their FYP).

→ The student can also see panels, to which they are assigned. PANEL WILL evaluate their FYP.

→ During evaluation the panel gives reviews or suggestions to the FYP's so students can view those reviews / suggestions. So students can improve their FYP's. Name of panel member who gives reviews should not be visible to the student. MAKE sure they only see reviews not information of panel member.

→ THEY CAN view deadlines of presentations, submissions of documents etc. If grades are finalized, then students can also view their grades in view.

ER DIAGRAM:

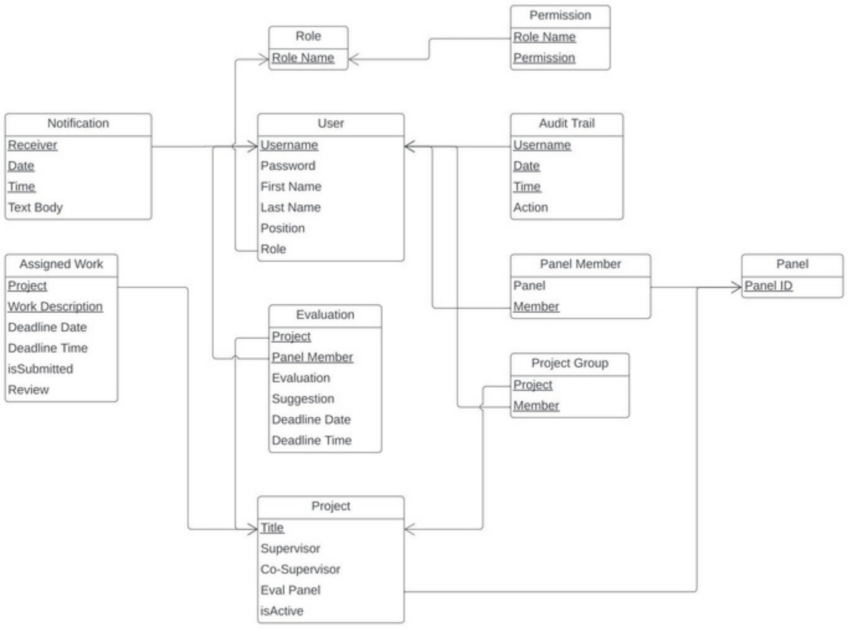

Permissions
Name
Role
Has
Date
Time
Position (Faculty,Student)
First Name
Date Time
Action Performed
Username
Full Name
User
performed
operation
Password
Last Name
Date
Time
d
Date Time
FYP Committee
Panel Member
Student
Supervisor
received
Notifications
Notification Text
Working on
Supervising
review
Review Text
Part of
Evaluation Data
Deadline
Date
ID
Panel
Suggestions
Evaluation
Project
assigned to
Assigned Work
Time
Deadline
Title
Status (Active?)
Submitted?
to be evaluated by
Work Description
1
N
1
N
1
N
1
N
1
N
1
N

DB SCHEMA:

DB Schema:

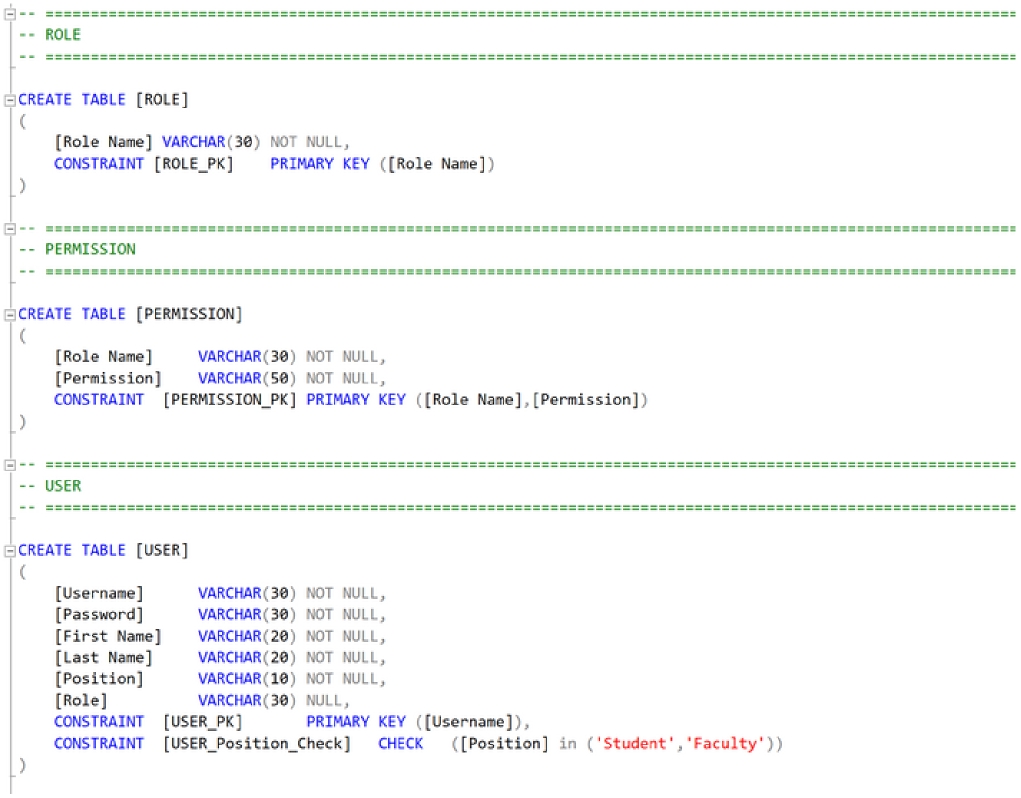
Database Schema was designed through following 9 steps that are used in designing the relational database schema. Foreign keys are used to make relations between different tables. It depends on relationship type (i.e. 1:1 etc.) that how we reference tables. Identifying relations properly for tables was quite a task as working and then creating cross reference tables depend on it.

DB SCHEMA:


Permission
Role Name
Permission
Role
Role Name
Notification
Receiver
Date
Time
Text Body
User
Username
Password
First Name
Last Name
Position
Role
Audit Trail
Username
Date
Time
Action
Assigned Work
Project
Work Description
Deadline Date
Deadline Time
isSubmitted
Review
Panel Member
Panel
Member
Panel
Panel ID
Evaluation
Project
Panel Member
Evaluation
Suggestion
Deadline Date
Deadline Time
Project Group
Project
Member
Project
Title
Supervisor
Co-Supervisor
Eval Panel
isActive

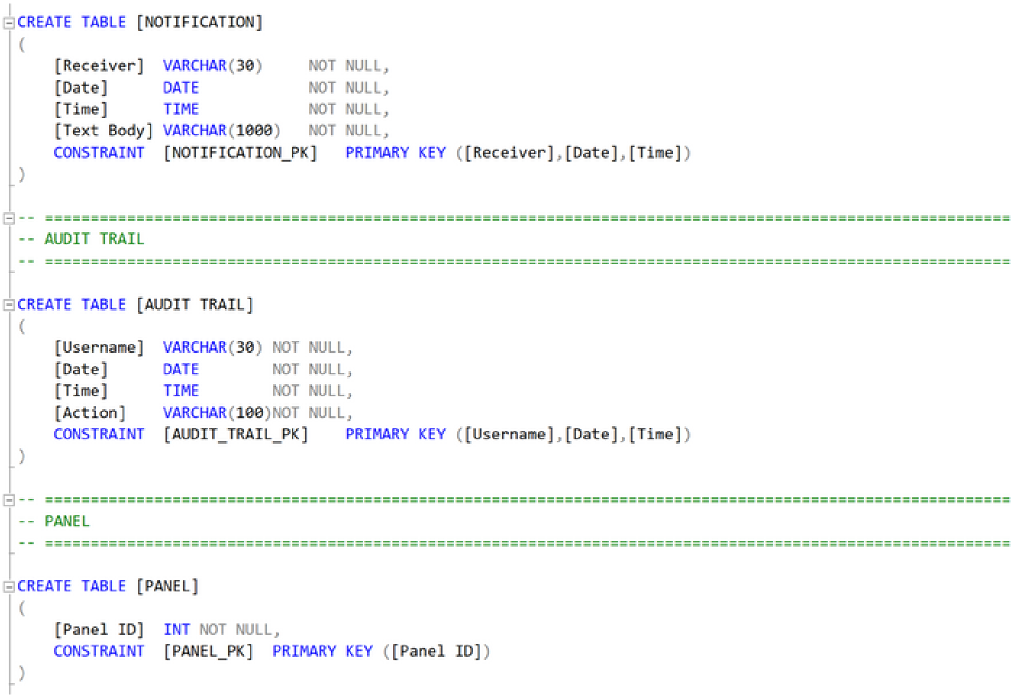
SQL QUERIES:

SQL Queries for Table, Constraints:



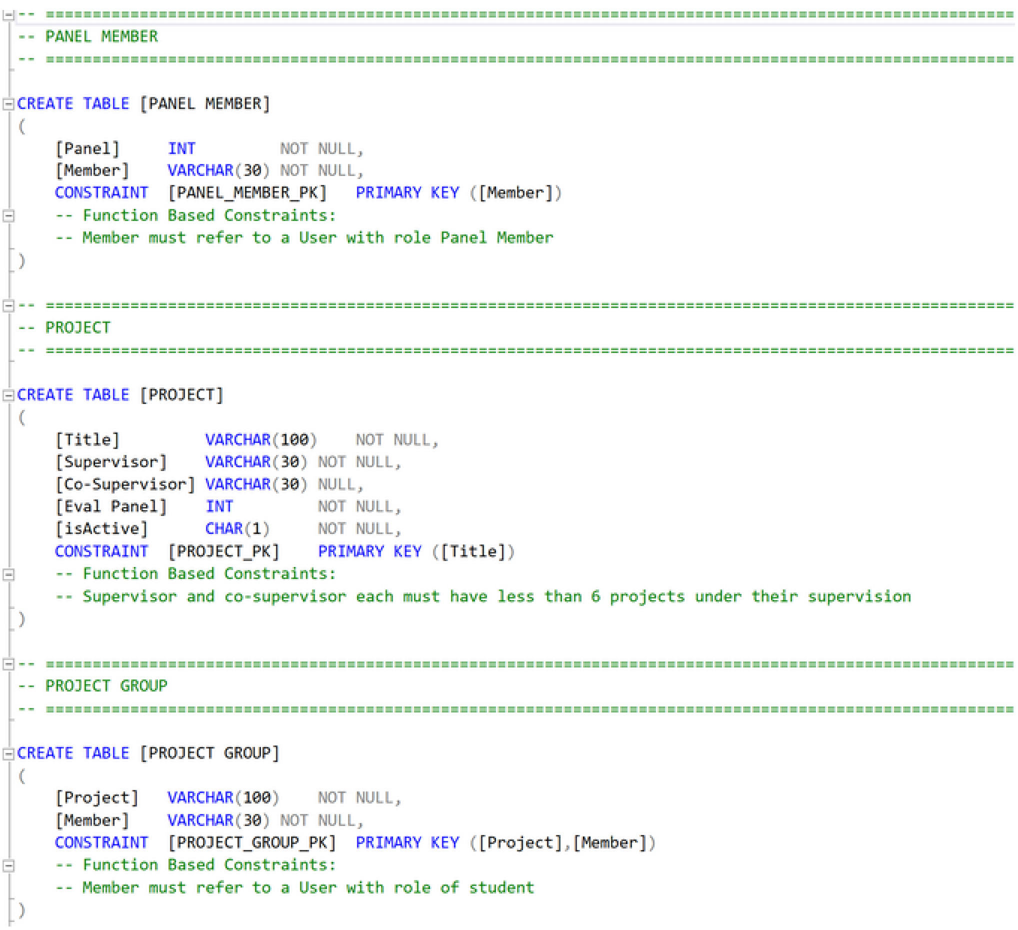
```
-- ===========================================================================================================
-- ROLE
-- ===========================================================================================================

CREATE TABLE [ROLE]
(
    [Role Name] VARCHAR(30) NOT NULL,
    CONSTRAINT [ROLE_PK]    PRIMARY KEY ([Role Name])
)

-- ===========================================================================================================
-- PERMISSION
-- ===========================================================================================================

CREATE TABLE [PERMISSION]
(
    [Role Name]     VARCHAR(30) NOT NULL,
    [Permission]    VARCHAR(50) NOT NULL,
    CONSTRAINT  [PERMISSION_PK] PRIMARY KEY ([Role Name],[Permission])
)

-- ===========================================================================================================
-- USER
-- ===========================================================================================================

CREATE TABLE [USER]
(
    [Username]      VARCHAR(30) NOT NULL,
    [Password]      VARCHAR(30) NOT NULL,
    [First Name]    VARCHAR(20) NOT NULL,
    [Last Name]     VARCHAR(20) NOT NULL,
    [Position]      VARCHAR(10) NOT NULL,
    [Role]          VARCHAR(30) NULL,
    CONSTRAINT  [USER_PK]           PRIMARY KEY ([Username]),
    CONSTRAINT  [USER_Position_Check]   CHECK   ([Position] in ('Student','Faculty'))
)
```

SQL Queries for Table, Constraints:



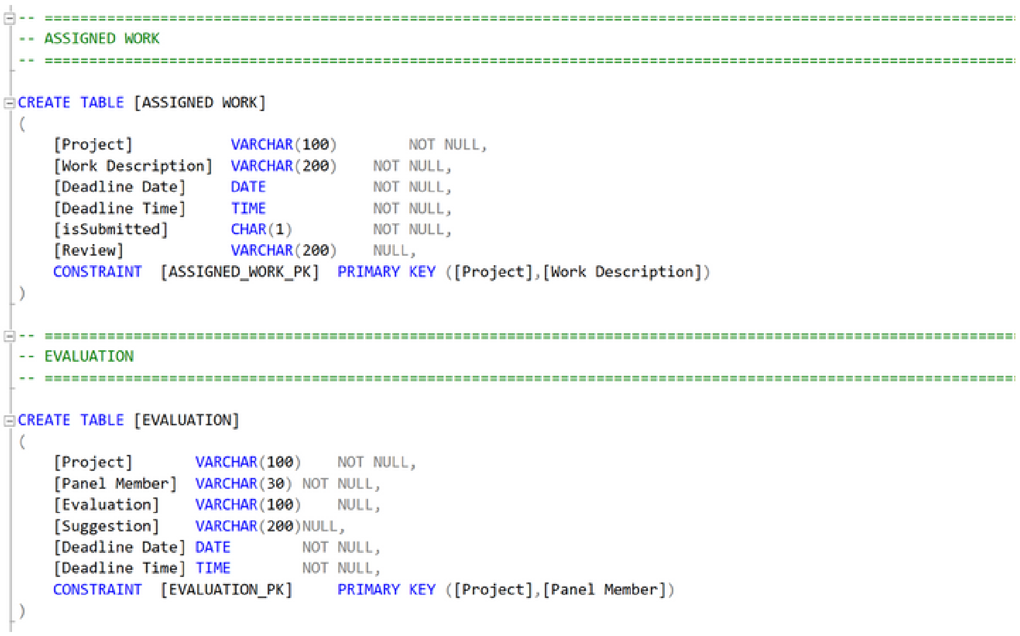
```sql
CREATE TABLE [NOTIFICATION]
(
    [Receiver]  VARCHAR(30)     NOT NULL,
    [Date]      DATE            NOT NULL,
    [Time]      TIME            NOT NULL,
    [Text Body] VARCHAR(1000)   NOT NULL,
    CONSTRAINT  [NOTIFICATION_PK]   PRIMARY KEY ([Receiver],[Date],[Time])
)

-- ===========================================================================================
-- AUDIT TRAIL
-- ===========================================================================================

CREATE TABLE [AUDIT TRAIL]
(
    [Username]  VARCHAR(30) NOT NULL,
    [Date]      DATE        NOT NULL,
    [Time]      TIME        NOT NULL,
    [Action]    VARCHAR(100)NOT NULL,
    CONSTRAINT  [AUDIT_TRAIL_PK]    PRIMARY KEY ([Username],[Date],[Time])
)

-- ===========================================================================================
-- PANEL
-- ===========================================================================================

CREATE TABLE [PANEL]
(
    [Panel ID]  INT NOT NULL,
    CONSTRAINT  [PANEL_PK]  PRIMARY KEY ([Panel ID])
)
```

SQL Queries for Table, Constraints:

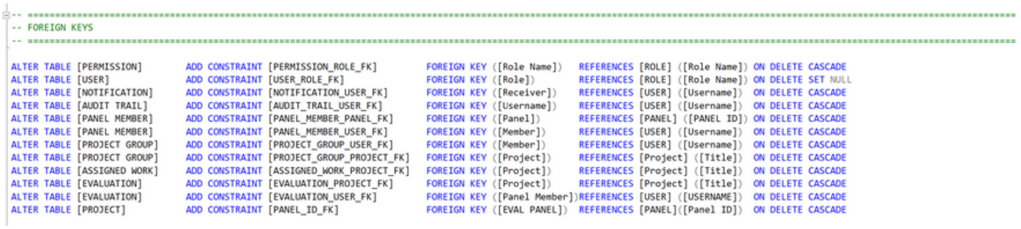


```
-- ===============================================================================================
-- PANEL MEMBER
-- ===============================================================================================

CREATE TABLE [PANEL MEMBER]
(
    [Panel]     INT         NOT NULL,
    [Member]    VARCHAR(30) NOT NULL,
    CONSTRAINT  [PANEL_MEMBER_PK]   PRIMARY KEY ([Member])
    -- Function Based Constraints:
    -- Member must refer to a User with role Panel Member
)

-- ===============================================================================================
-- PROJECT
-- ===============================================================================================

CREATE TABLE [PROJECT]
(
    [Title]         VARCHAR(100)    NOT NULL,
    [Supervisor]    VARCHAR(30) NOT NULL,
    [Co-Supervisor] VARCHAR(30) NULL,
    [Eval Panel]    INT         NOT NULL,
    [isActive]      CHAR(1)     NOT NULL,
    CONSTRAINT  [PROJECT_PK]    PRIMARY KEY ([Title])
    -- Function Based Constraints:
    -- Supervisor and co-supervisor each must have less than 6 projects under their supervision
)

-- ===============================================================================================
-- PROJECT GROUP
-- ===============================================================================================

CREATE TABLE [PROJECT GROUP]
(
    [Project]   VARCHAR(100)    NOT NULL,
    [Member]    VARCHAR(30) NOT NULL,
    CONSTRAINT  [PROJECT_GROUP_PK]  PRIMARY KEY ([Project],[Member])
    -- Function Based Constraints:
    -- Member must refer to a User with role of student
)
```

SQL Queries for Table, Constraints:

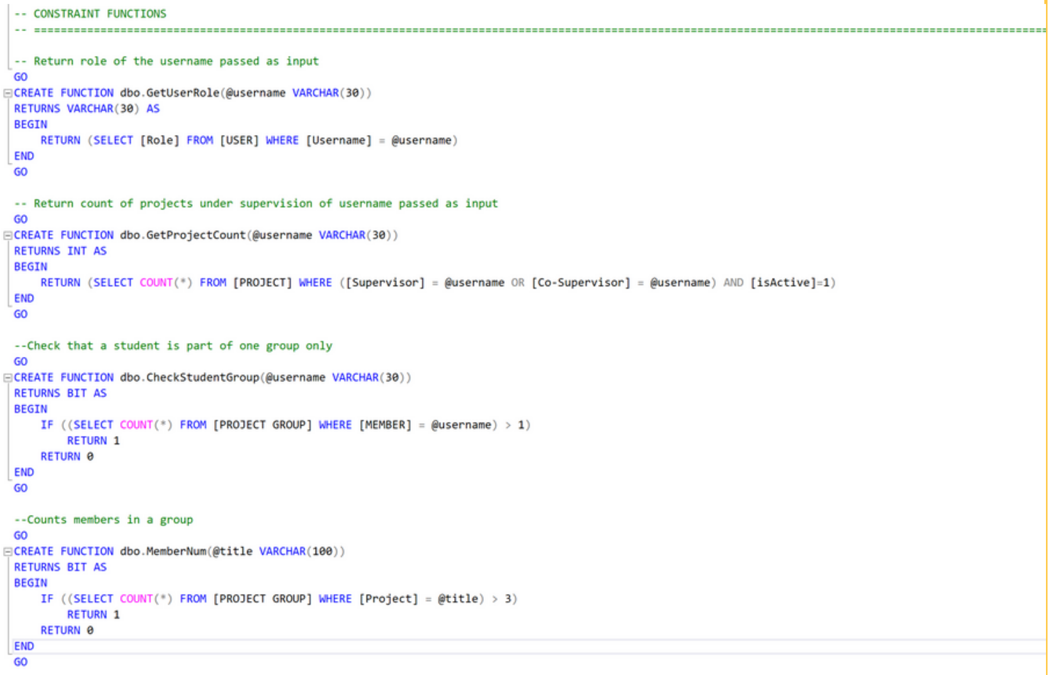


```sql
-- ==========================================================================================
-- ASSIGNED WORK
-- ==========================================================================================

CREATE TABLE [ASSIGNED WORK]
(
    [Project]           VARCHAR(100)        NOT NULL,
    [Work Description]  VARCHAR(200)    NOT NULL,
    [Deadline Date]     DATE            NOT NULL,
    [Deadline Time]     TIME            NOT NULL,
    [isSubmitted]       CHAR(1)         NOT NULL,
    [Review]            VARCHAR(200)    NULL,
    CONSTRAINT  [ASSIGNED_WORK_PK]  PRIMARY KEY ([Project],[Work Description])
)

-- ==========================================================================================
-- EVALUATION
-- ==========================================================================================

CREATE TABLE [EVALUATION]
(
    [Project]       VARCHAR(100)    NOT NULL,
    [Panel Member]  VARCHAR(30) NOT NULL,
    [Evaluation]    VARCHAR(100)    NULL,
    [Suggestion]    VARCHAR(200)NULL,
    [Deadline Date] DATE        NOT NULL,
    [Deadline Time] TIME        NOT NULL,
    CONSTRAINT  [EVALUATION_PK]     PRIMARY KEY ([Project],[Panel Member])
)
```

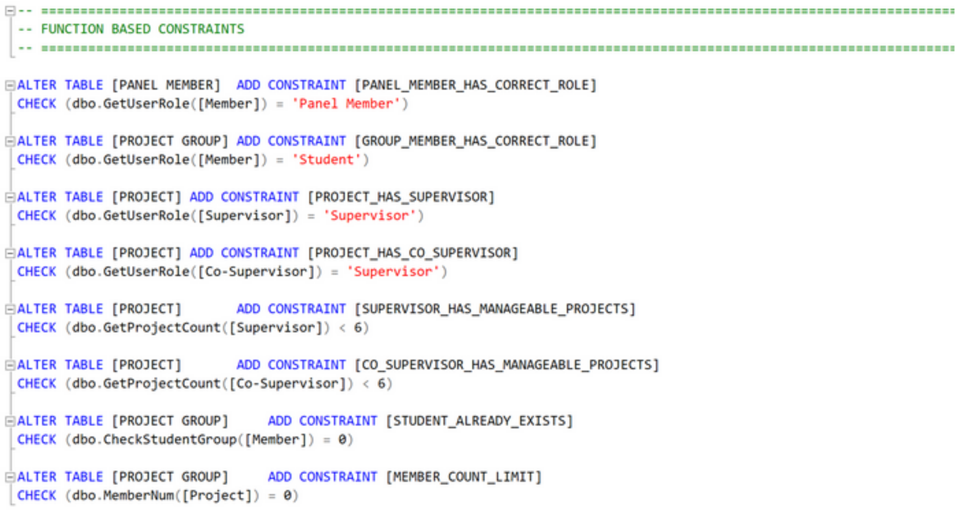


```sql
-- ==========================================================================================
-- FOREIGN KEYS
-- ==========================================================================================

ALTER TABLE [PERMISSION]       ADD CONSTRAINT [PERMISSION_ROLE_FK]         FOREIGN KEY ([Role Name])   REFERENCES [ROLE] ([Role Name]) ON DELETE CASCADE
ALTER TABLE [USER]             ADD CONSTRAINT [USER_ROLE_FK]               FOREIGN KEY ([Role])        REFERENCES [ROLE] ([Role Name]) ON DELETE SET NULL
ALTER TABLE [NOTIFICATION]     ADD CONSTRAINT [NOTIFICATION_USER_FK]       FOREIGN KEY ([Receiver])    REFERENCES [USER] ([Username])  ON DELETE CASCADE
ALTER TABLE [AUDIT TRAIL]      ADD CONSTRAINT [AUDIT_TRAIL_USER_FK]        FOREIGN KEY ([Username])    REFERENCES [USER] ([Username])  ON DELETE CASCADE
ALTER TABLE [PANEL MEMBER]     ADD CONSTRAINT [PANEL_MEMBER_PANEL_FK]      FOREIGN KEY ([Panel])       REFERENCES [PANEL] ([PANEL ID]) ON DELETE CASCADE
ALTER TABLE [PANEL MEMBER]     ADD CONSTRAINT [PANEL_MEMBER_USER_FK]       FOREIGN KEY ([Member])      REFERENCES [USER] ([Username])  ON DELETE CASCADE
ALTER TABLE [PROJECT GROUP]    ADD CONSTRAINT [PROJECT_GROUP_USER_FK]      FOREIGN KEY ([Member])      REFERENCES [USER] ([Username])  ON DELETE CASCADE
ALTER TABLE [PROJECT GROUP]    ADD CONSTRAINT [PROJECT_GROUP_PROJECT_FK]   FOREIGN KEY ([Project])     REFERENCES [Project] ([Title])  ON DELETE CASCADE
ALTER TABLE [ASSIGNED WORK]    ADD CONSTRAINT [ASSIGNED_WORK_PROJECT_FK]   FOREIGN KEY ([Project])     REFERENCES [Project] ([Title])  ON DELETE CASCADE
ALTER TABLE [EVALUATION]       ADD CONSTRAINT [EVALUATION_PROJECT_FK]      FOREIGN KEY ([Project])     REFERENCES [Project] ([Title])  ON DELETE CASCADE
ALTER TABLE [EVALUATION]       ADD CONSTRAINT [EVALUATION_USER_FK]         FOREIGN KEY ([Panel Member])REFERENCES [USER] ([USERNAME])  ON DELETE CASCADE
ALTER TABLE [PROJECT]          ADD CONSTRAINT [PANEL_ID_FK]                FOREIGN KEY ([EVAL PANEL])  REFERENCES [PANEL]([Panel ID])  ON DELETE CASCADE
```

SQL Queries for Table, Constraints:



```
-- CONSTRAINT FUNCTIONS
-- ==================================================================================================================================

-- Return role of the username passed as input
GO
CREATE FUNCTION dbo.GetUserRole(@username VARCHAR(30))
RETURNS VARCHAR(30) AS
BEGIN
    RETURN (SELECT [Role] FROM [USER] WHERE [Username] = @username)
END
GO

-- Return count of projects under supervision of username passed as input
GO
CREATE FUNCTION dbo.GetProjectCount(@username VARCHAR(30))
RETURNS INT AS
BEGIN
    RETURN (SELECT COUNT(*) FROM [PROJECT] WHERE ([Supervisor] = @username OR [Co-Supervisor] = @username) AND [isActive]=1)
END
GO

--Check that a student is part of one group only
GO
CREATE FUNCTION dbo.CheckStudentGroup(@username VARCHAR(30))
RETURNS BIT AS
BEGIN
    IF ((SELECT COUNT(*) FROM [PROJECT GROUP] WHERE [MEMBER] = @username) > 1)
        RETURN 1
    RETURN 0
END
GO

--Counts members in a group
GO
CREATE FUNCTION dbo.MemberNum(@title VARCHAR(100))
RETURNS BIT AS
BEGIN
    IF ((SELECT COUNT(*) FROM [PROJECT GROUP] WHERE [Project] = @title) > 3)
        RETURN 1
    RETURN 0
END
GO
```



```
-- ==============================================================================================================
-- FUNCTION BASED CONSTRAINTS
-- ==============================================================================================================

ALTER TABLE [PANEL MEMBER]  ADD CONSTRAINT [PANEL_MEMBER_HAS_CORRECT_ROLE]
CHECK (dbo.GetUserRole([Member]) = 'Panel Member')

ALTER TABLE [PROJECT GROUP] ADD CONSTRAINT [GROUP_MEMBER_HAS_CORRECT_ROLE]
CHECK (dbo.GetUserRole([Member]) = 'Student')

ALTER TABLE [PROJECT] ADD CONSTRAINT [PROJECT_HAS_SUPERVISOR]
CHECK (dbo.GetUserRole([Supervisor]) = 'Supervisor')

ALTER TABLE [PROJECT] ADD CONSTRAINT [PROJECT_HAS_CO_SUPERVISOR]
CHECK (dbo.GetUserRole([Co-Supervisor]) = 'Supervisor')

ALTER TABLE [PROJECT]       ADD CONSTRAINT [SUPERVISOR_HAS_MANAGEABLE_PROJECTS]
CHECK (dbo.GetProjectCount([Supervisor]) < 6)

ALTER TABLE [PROJECT]       ADD CONSTRAINT [CO_SUPERVISOR_HAS_MANAGEABLE_PROJECTS]
CHECK (dbo.GetProjectCount([Co-Supervisor]) < 6)

ALTER TABLE [PROJECT GROUP]     ADD CONSTRAINT [STUDENT_ALREADY_EXISTS]
CHECK (dbo.CheckStudentGroup([Member]) = 0)

ALTER TABLE [PROJECT GROUP]     ADD CONSTRAINT [MEMBER_COUNT_LIMIT]
CHECK (dbo.MemberNum([Project]) = 0)
```