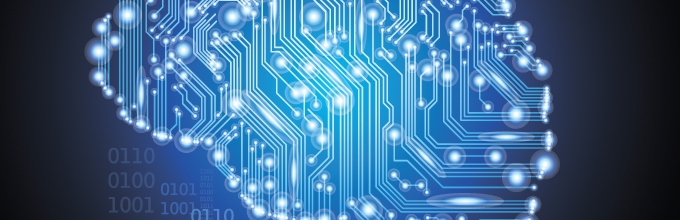


­­

IS PROJECT SPECIFICATION

­

**Module: Database Management**

**Code: MLDM183-01**

**Project Title: playing game (MMORPG)**

**Project number: 1A**

**Student name: Michel Chungu Nyembo**

**Student number: R4S6NSBG1**

2018

**TABLE OF CONTENTS**

Purpose……………………………………………………………………………………………….page1

Description of the program…………………………………………………………………..page1

System requirements……………………………………………………………………………page1

Installation guide line…………………………………………………………………………...page1

Method to run program………………………………………………………………………..page1

ER diagram……………………………………………………………………………………………page3

Database and tables………………………………………………………………………………page4

Inserting data on table…………………………………………………………………………..page7

views ……………………………………………………………………………..........................page11

Store procedures…………………………………………………………………………………..page13

Triggers………………………………………………………………………………………………….page17

Indexes………………………………………………………………………………………………….page18

Bibliography…………………………………………………………………………………………..page19

**Purpose**

This project is for study purpose only. The CTI makes no claims, promises, or grantees about the accuracy, completeness, or adequacy of the project and no legal liability or other responsibility is accepted by CTI for any errors, omissions, or documentation on this project. Project based on Microsoft SQL2017.

**Description**

The program used is SQL Server, which was developed by Microsoft. The SQL Server Management Studio is an integrated application that allows access to most graphical tools. It has components than be implemented separately or as a group.

**System requirements**

**Windows**

* GHz Intel dual core processer.
* Microsoft windows 10 service pack 1 (windows8.1 can also be used).
* 2GB RAM.
* DVD-ROM.
* Internet connection.
* 1024\*768, 16bit display.
* 20GB of available disk space.

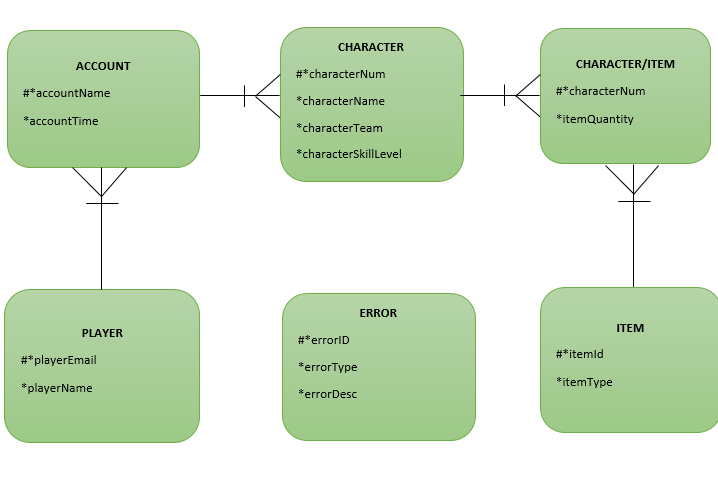
**Installation guide**

* First of all you must have internet connection.
* Install NET Framework SP1, or download run manually.
* Download SQL Server 2016 and install run set up and you need to accept all license agreement and click install.
* Set up will be complete successfully.

**A method to run the program**

* Click the start button and search for Microsoft SQL Server tools 2017.
* Right click on the Microsoft SQL Server tools 2017.

ERD



TABLES

USE master

GO

--CREATION OF DATABASE START--

IF EXISTS (SELECT name FROM master.dbo.sysdatabases

WHERE name = 'online\_playing\_game')

BEGIN

DROP DATABASE online\_playing\_game

PRINT 'Existing database online\_playing\_game deleted.'

END

GO

CREATE DATABASE online\_playing\_game

ON PRIMARY

(

NAME = 'online\_playing\_game\_data',

FILENAME = 'C:\sqlmyProject\online\_playing\_game\_data.mdf', -- primary data

SIZE = 5MB,

FILEGROWTH = 10%

)

LOG ON

(

NAME = 'online\_playing\_game\_log',

FILENAME = 'C:\sqlmyProject\online\_playing\_game\_log.ldf',-- transaction log data

SIZE = 5MB,

FILEGROWTH = 10%

)

GO

ALTER DATABASE online\_playing\_game

ADD FILEGROUP onlineplayinggame\_filegroup

GO

ALTER DATABASE online\_playing\_game

ADD FILE

(

NAME = onlineplayinggame\_info,

FILENAME = 'C:\sqlmyProject\online\_playing\_game\_info.ndf', -- Secondary data

SIZE = 5MB

)

GO

--END--

--CREATE TABLES START--

CREATE TABLE player --creating player's table

(

playerEmail VARCHAR (30) NOT NULL PRIMARY KEY, -- primary key

playerName VARCHAR (30) NOT NULL,

CHECK(playerEmail LIKE '%\_@%')

)

GO

CREATE TABLE account --creating account's table

(

accountName VARCHAR (30) NOT NULL PRIMARY KEY, -- primary key

accountTime INT NOT NULL CHECK (accountTime > -1), -- checking constraint

player\_Email VARCHAR (30) NOT NULL REFERENCES player (playerEmail) -- foreign key

ON DELETE CASCADE

)

GO

CREATE TABLE errors --creating errors' table

(

errorID INT NOT NULL PRIMARY KEY, -- primary key

errorDescr VARCHAR (200) NOT NULL,

errorType VARCHAR (40) NOT NULL

)

GO

CREATE TABLE characters --creating crharacters' table

(

characterNum INT NOT NULL PRIMARY KEY, -- primary key

characterTeam VARCHAR (30) NOT NULL,

characterName VARCHAR (30) NOT NULL,

characterSkillLevel VARCHAR (30) NOT NULL,

account\_Name VARCHAR (30) NOT NULL REFERENCES account (accountName)

ON DELETE CASCADE

)

GO

CREATE TABLE item --creating item's table

(

itemID INT NOT NULL PRIMARY KEY, -- primary key

itemType VARCHAR (50) NOT NULL,

itemDesc VARCHAR (300) NOT NULL

)

GO

CREATE TABLE characters\_inventory --creating characters\_item's table

(

characterNum INT NOT NULL PRIMARY KEY, -- primary key

itemQuantity INT NOT NULL DEFAULT 1,

item\_ID INT NOT NULL REFERENCES item (itemID) -- primary key

)

GO

--END--

--INSERT DATA INTO TABLES START--

INSERT INTO player --inserting data into player's table

VALUES

('marcnyembo13@gmail.com', 'marc'),

('michelnyembo03@gmail.com', 'michel'),

('jeannyembo@yahoo.com', 'jean'),

('tabithakabulo@hotmail.com', 'tabitha'),

('patientmbayo11@gmail.com', 'patient'),

('laurekidi@gmail.com', 'laure'),

('davidyembo1@yahoo.com', 'david'),

('sylviemwange@gmail.com', 'sylvie'),

('djomushesha@gmail.com', 'djo'),

('leonmwilambwe@hotmail.com', 'leon'),

('dimaelkidi@gmail.com', 'dimael'),

('michaelnyembo@gmail.com', 'michael'),

('eldjonaymuperzili@yahoo.com', 'eldjonay'),

('lysetteluvungu@hotmail.com', 'lysette'),

('robertmuyumba@gmail.com', 'robert'),

('linardnkulu12@yahoo.com', 'linard'),

('jeanmarc@yahoo.com', 'jeanmarc'),

('jeanmichel@gmail.com', 'jeanmichel'),

('jeanclaude@hotmail.com', 'jeanclaude'),

('julienlwamba@hotmail.com', 'julien'),

('enochlwamba@hotmail.com', 'enoch'),

('marechal12@gmail.com', 'marechal12'),

('Benjamin14@hotmail.com', 'Benjamin14')

GO

INSERT INTO account (player\_Email, accountName, accountTime)

--inserting data into account's table

VALUES

('marcnyembo13@gmail.com', 'marc11', 0),

('michelnyembo03@gmail.com', 'michel32', 3),

('jeannyembo@yahoo.com', 'jean43', 4),

('tabithakabulo@hotmail.com', 'tabitha54', 5),

('patientmbayo11@gmail.com', 'patient5', 3),

('laurekidi@gmail.com', 'laure65', 0),

('davidyembo1@yahoo.com', 'david87', 0),

('sylviemwange@gmail.com', 'sylvie676', 0),

('djomushesha@gmail.com', 'djo78', 2),

('leonmwilambwe@hotmail.com', 'leon87', 5),

('dimaelkidi@gmail.com', 'dimael98', 4),

('michaelnyembo@gmail.com', 'michael867', 6),

('eldjonaymuperzili@yahoo.com', 'eldjonay907', 5),

('lysetteluvungu@hotmail.com', 'lysette6', 4),

('robertmuyumba@gmail.com', 'robert98', 3),

('linardnkulu12@yahoo.com', 'linard09', 5),

('jeanmarc@yahoo.com', 'jeanmarc55', 4),

('jeanmichel@gmail.com', 'jeanmichel23', 5),

('jeanclaude@hotmail.com', 'jeanclaude54', 7),

('julienlwamba@hotmail.com', 'julien12', 6),

('enochlwamba@hotmail.com', 'enoch14', 6),

('Benjamin14@hotmail.com', 'Benjamin14', 6)

GO

INSERT INTO errors --inserting data into error's table

VALUES

(001, 'invalid information', 'Try'),

(002, 'Bad email address. Your recipients

emails may contain typos.', 'Unavailable service'),

(003, 'Invalid mailbox address.', ' syntax'),

(004, 'invalid address.', 'Information reply')

GO

INSERT INTO characters(characterNum, characterTeam, characterName,

characterSkillLevel, account\_Name) --inserting data into characters' table

VALUES

(11, 'Soldier', 'Headhunter', 'Berseker', 'marc11'),

(16, 'Soldier', 'Renegade', 'Berseker','michel32'),

(15, 'Soldier', 'Wukong', 'Berseker','jean43'),

(12, 'Soldier', 'Jonesy', 'Commando','tabitha54'),

(14, 'Soldier', 'Ramirez', 'Commando','patient5'),

(13, 'Soldier', 'Spitfire', 'Commando','laure65'),

(19, 'Ninja', 'Ken', 'Assassin', 'david87'),

(18, 'Ninja', 'Sarah', 'Assassin','sylvie676'),

(17, 'Ninja', 'Luna', 'Piercing Lotus', 'djo78'),

(20, 'Ninja', 'Edge', 'Piercing Lotus', 'leon87'),

(24, 'Ninja', 'Mari', 'Energy Thief', 'dimael98'),

(22, 'Constructor', 'Bull', 'Guardian', 'michael867'),

(25, 'Constructor', 'Knox', 'Guardian', 'eldjonay907'),

(21, 'Constructor', 'Penny', 'Guardian', 'lysette6'),

(23, 'Constructor', 'Thora', 'Machinist', 'robert98'),

(29, 'Constructor', 'Harper', 'Machinist', 'linard09'),

(27, 'Outlander', 'Grizzly', 'Enforcer Grizzly','jeanmarc55'),

(31, 'Outlander', 'A.C', 'Striker', 'jeanmichel23'),

(26, 'Outlander', 'Specter', 'TrailBlazer','jeanclaude54'),

(30, 'Outlander', 'Jess', 'TrailBlazer', 'julien12'),

(28, 'Outlander', 'Quinn', 'TrailBlazer', 'enoch14')

GO

INSERT INTO item (itemID, itemType, itemDesc) --inserting data into characters' table

VALUES

(101, 'Cluster Bomb','Frag Grenade releases 6

cluster explosions around the impact site that will

each do 25% of the original damage.'),

(111, 'Hearty Blade ', 'Every 4th hit with a melee

weapon restores 24 base health'),

(202, 'Hammer Critical Chance', 'Increases blunt melee critical rating by 18.'),

(222, 'Capacitor', 'Increases duration of Shock Tower by 1 second.')

GO

INSERT INTO characters\_inventory(characterNum, item\_ID,itemQuantity)

--inserting data into characters\_item's table

VALUES

(11, 101, 3),

(16, 101, 5),

(15, 101, 6),

(12, 101, 4),

(14, 101, 8),

(13, 101, 5),

(19, 111, 7),

(18, 111, 3),

(17, 111, 8),

(20, 111, 4),

(24, 111, 6),

(22, 202, 2),

(25, 202, 5),

(21, 202, 8),

(23, 202, 3),

(29, 202, 7),

(27, 222, 9),

(31, 222, 6),

(26, 222, 5),

(30, 222, 6)

GO

-- CREATE VIEWS

CREATE VIEW vwBlockedAccounts -- creating all currently blocked accounts

AS

SELECT account.accountName, accountTime, player\_Email

FROM account

WHERE accountTime = 0

GO

SELECT \* FROM vwBlockedAccounts

GO

CREATE VIEW vwTopSkill

AS

SELECT TOP 20 characters.characterSkillLevel, account.accountName, account.accountTime

FROM characters

JOIN account

ON characters.account\_Name = account.accountName

ORDER BY characters.characterSkillLevel

GO

SELECT \* FROM vwTopSkill

GO

CREATE VIEW vwTopStackedItems

AS

SELECT TOP 20 characters\_inventory.item\_ID, characters\_inventory.itemQuantity, item.itemtype, characters.characterName,

item.itemDesc

FROM characters\_inventory

JOIN characters

ON characters.characterNum = characters\_inventory.characterNum

JOIN item

ON characters\_inventory.item\_ID = item.itemID

ORDER BY itemQuantity

GO

SELECT \* FROM vwTopStackedItems

GO

CREATE VIEW vwPopItems

AS

SELECT TOP 5 characters\_inventory.itemQuantity, characters\_inventory.item\_ID,characterNum, item.itemType

FROM characters\_inventory

INNER JOIN item

ON characters\_inventory.item\_ID = item.itemID

ORDER BY itemQuantity

GO

SELECT \* FROM vwPopItems

-- CREATE STORED PROCEDURES

CREATE PROCEDURE spRegister

@accountName VARCHAR (30),

@accountTime INT,

@player\_Email VARCHAR (30)

AS

IF NOT EXISTS ( SELECT playerEmail FROM player WHERE playerEmail = @player\_Email)

BEGIN

PRINT'Invalid player email, choose an existing one or register a new player before continuing!'

END

INSERT INTO account ( account.accountName, account.accountTime, account.player\_Email)

VALUES (@accountName, @accountTime, @player\_Email)

GO

EXEC spRegister 'marechal12', 4, 'marechal12@gmail.com'

GO

CREATE PROCEDURE spAddTime

@accountName VARCHAR (30),

@accountTime INT

AS

IF NOT EXISTS ( SELECT accountName FROM account WHERE accountName = @accountName)

BEGIN

PRINT'The account does not exist, try another one'

END

UPDATE account

SET accountTime = accountTime + @accountTime

WHERE accountName = @accountName

GO

EXEC spAddTime 'robert98', 1

GO

CREATE PROCEDURE spAddItem

@characterNum INT,

@itemQuantity INT,

@item\_ID INT

AS

IF NOT EXISTS( SELECT characterNum FROM characters\_inventory WHERE characterNum = @characterNum)

BEGIN

PRINT'the item does not exist'

END

INSERT INTO characters\_inventory(characterNum, itemQuantity, item\_ID)

VALUES (@characterNum, @itemQuantity, @item\_ID)

GO

EXEC spAddItem 28, 6, 222

GO

CREATE PROCEDURE spAddChar

@characterNum INT,

@characterTeam VARCHAR (30),

@characterName VARCHAR (30),

@characterSkillLevel VARCHAR (30),

@account\_Name VARCHAR (30)

AS

IF EXISTS (SELECT characterNum FROM characters WHERE characterNum = @characterNum )

BEGIN

PRINT'Exists'

END

IF EXISTS (SELECT account\_Name FROM characters WHERE account\_Name = @account\_Name)

BEGIN

PRINT'Exists'

END

UPDATE characters

SET characterNum = @characterNum,

account\_Name = @account\_Name

WHERE characterNum = @characterNum

GO

EXEC spAddChar 31, 'Outlander', 'Wiu', 'TrailBlazer', 'Benjamin14'

GO

CREATE PROCEDURE spSenderLetter

AS

SELECT \*

-- the accName column could have been selected by itself

INTO #tempAccount

FROM account

--Add a column called Printed

ALTER TABLE #tempAccount

ADD

PRINTED SMALLINT

--Set all the values in that column equal to 0

UPDATE #tempAccount

SET Printed = 0

-- SQL does not print column directly, requires varialbles.

DECLARE @accountName VARCHAR (30)

DECLARE @accountTime INT

DECLARE @player\_Email VARCHAR (30)

--Creates a while loop to iterate through the rows

WHILE EXISTS (SELECT \*

FROM #tempAccount

WHERE Printed = 0)

BEGIN

SELECT @accountName = MIN(accountName)

FROM #tempAccount

WHERE Printed = 0

SELECT @accountName = #tempAccount.accountName,

@player\_Email = player.playerEmail,

@accountTime = #tempAccount.accountTime

FROM #tempAccount

JOIN player

ON #tempAccount.player\_email = player.playerEmail

WHERE @accountName = #tempAccount.accountName

--Prints out the report fro each account

PRINT 'Hi '+@accountName

PRINT ' '

PRINT 'We have emailed this letter to ' + (@player\_Email) + 'informing you that you have ' + CAST(@accountTime AS VARCHAR) + 'left'

PRINT 'Let us know if this change'

PRINT 'Regards, Head Office'

UPDATE #tempAccount

SET Printed = 1

WHERE @accountName = accountName

END

GO

EXEC spSenderLetter

GO

CREATE TRIGGER tr\_notify

ON player

AFTER INSERT

AS

PRINT 'Player inserted successfully'

GO

--trying triggers

INSERT INTO player

VALUES ('Carlos12@gmail.com', 'Carlos12')

GO

--creating triggers on account

CREATE TRIGGER tr\_edit\_account

ON account

AFTER INSERT

AS

PRINT ' You cannot edit this information in the table'

GO

--trying triggers

INSERT INTO account

VALUES ('Carlos12', 6, 'Carlos12@gmail.com')

GO

-- Creating a Nonclustered index of the player tables

CREATE NONCLUSTERED INDEX IX\_player

ON player

( playerEmail ) ;

GO

-- Creating a Nonclustered index of the account table

CREATE NONCLUSTERED INDEX IX\_account

ON account

( accountName ) ;

GO

-- Creating a Nonclustered index of the playerChar table

CREATE NONCLUSTERED INDEX IX\_playerChar

ON characters

( characterNum ) ;

GO

CREATE NONCLUSTERED INDEX IX\_item

ON item

( itemID ) ;

GO

-- Creating a Nonclustered index of the itemQuantity table

CREATE NONCLUSTERED INDEX IX\_itemQuantity

ON characters\_inventory

( characterNum ) ;

GO

-- Creating a Nonclustered index of the Error table

CREATE NONCLUSTERED INDEX IX\_Error

ON errors

( errorID ) ;

GO

BIBLIOGRAPHY

Author: Ben Fortas, published 2017 Microsoft SQL Server S-SQL in 10 minutes