**IST 278 Unit 7 Lab**

Name: Michael Bacu Date: 7/6/2023

# **Lab Instructions**

1. Type your name and the date in the spaces provided.
2. Use the SQL Server Management Studio and the AP database as specified below.
3. Complete all the 6 exercises in this lab
4. Upload and submit a completed copy of this lab sheet before the due date.

**Database Security Lab**

# **1. Unit 7 Lab Exercise 1**

Write a script that creates a user-defined database role in the AP database. Name this new role PaymentEntryXX where the XX are your initials. Give UPDATE permission to the new role for the Invoices table, UPDATE and INSERT permission for the InvoiceLineItems table, and SELECT permission for all user tables. Make sure to include a comment at the top of the script with your name and the date you coded the script. Format this comment like the following sample:

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* PaymentEntryXX Role Creation Script \*/

/\* Coded by: Your Name \*/

/\* Coded on: MM/DD/YY \*/

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**---** **Paste below this line the Script you wrote for this exercise --**

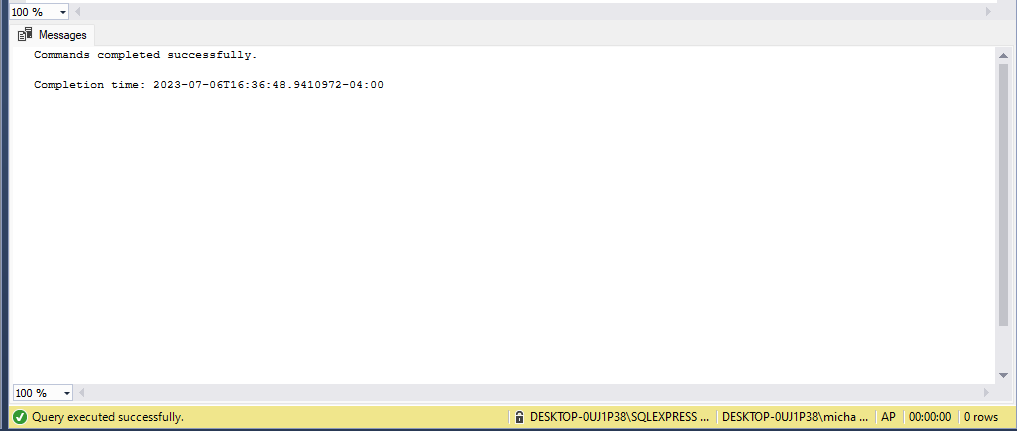
CREATE ROLE PaymentEntryMB;

GRANT UPDATE ON Invoices TO PaymentEntryMB;

GRANT UPDATE, INSERT ON InvoiceLineItems TO PaymentEntryMB;

GRANT SELECT ON DATABASE :: [AP] TO PaymentEntryMB;

**---** **Paste below this line the run results from executing the script you wrote--**



# **2. Unit 7 Lab Exercise 2**

Use the AP database and write a script that:

1. Creates a login ID named “GTCS01XX” where the XX are your initials
2. Assigns this login an initial password of “GATr99999”
3. Sets the default database for this login to the AP database
4. Creates a user named “GTCS01XX” where the XX are your initials
5. Assigns the “GTCS01XX” user to the PaymentEntryXX role you created in exercise 1
6. Includes a comment at the top of the script with your name and the date you coded the script. Format this comment like the following sample:

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* GTCS01XX Login and User Creation Script \*/

/\* Coded by: Your Name \*/

/\* Coded on: MM/DD/YY \*/

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**--- Paste below this line the Script you wrote for this exercise --**

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* GTCS01MB Login and User Creation Script \*/

-- Coded by: Michael Bacu /

-- Coded on: 7/6/23 /

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

CREATE LOGIN GTCS01MB WITH PASSWORD = 'GATr99999';

ALTER LOGIN GTCS01MB WITH PASSWORD = 'GATr99999';

ALTER LOGIN GTCS01MB WITH DEFAULT\_DATABASE = AP;

CREATE USER GTCS01MB;

ALTER ROLE PaymentEntryMB ADD MEMBER GTCS01MB;

**---** **Paste below this line the run results from executing the script you wrote--**

A screenshot of a computer

Description automatically generated

# **3. Unit 7 Lab Exercise 3**

Write a script that uses a table, dynamic SQL, and a cursor to create login IDs in the AP database based on the contents of the table.

1. Get started by first creating the table. Below is a create statement that you can modify and execute to create this table. Make sure to change the XX in the table name to your initials:

CREATE TABLE NewLoginsXX (LoginName varchar(128));

1. After the table is created. Insert four rows in it. Below is an insert statement that you can modify and execute to put the rows in your NewLoginsXX table. In addition to modifying the table name, make sure to change the XX at the start of each login name to your initials before running this insert.

INSERT NewLoginsXX VALUES ('XXBBrown'), ('XXCChaplin'), ('XXDDyer'), ('XXEEbbers');

1. Start coding the script with a comment block. Format this comment like the following sample (make sure to revise the sample to reflect your name and the date you coded it):

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Login Generator Script \*/

/\* This Script uses dynamic SQL and a cursor to process the \*/

/\* NEWLoginsXX table one row at a time. For each row on this \*/

/\* table the script will perform four actions. \*/

/\* \*/

/\* Coded by: Your Name \*/

/\* Coded on: MM/DD/YY \*/

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

1. Code the rest of your script using dynamic SQL and a cursor to perform four actions for each row in this table:

(1) create a login with a temporary password that's based on the first four letters of the login name followed by “99999”

(2) set the default database to the AP database

(3) create a user for the login with the same name as the login

(4) assign the user to the PaymentEntryXX role you created in exercise 1.

**--- Paste below this line the Script you wrote for this exercise --**

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Login Generator Script \*/

/\* This Script uses dynamic SQL and a cursor to process the \*/

/\* NEWLoginsXX table one row at a time. For each row on this \*/

/\* table the script will perform four actions. \*/

/\* \*/

-- Coded by: Michael Bacu /

-- Coded on: 7/6/23 /

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

USE AP;

CREATE TABLE NewLoginsMB (LoginName varchar(128));

DECLARE @LoginName VARCHAR(128);

DECLARE loginCursor CURSOR FOR

SELECT LoginName FROM NewLoginsMB;

OPEN loginCursor;

FETCH NEXT FROM loginCursor INTO @LoginName;

WHILE @@FETCH\_STATUS = 0

BEGIN

DECLARE @Password VARCHAR(128) = LEFT(@LoginName, 4) + '99999';

DECLARE @CreateLoginSQL NVARCHAR(MAX) = 'CREATE LOGIN ' + QUOTENAME(@LoginName) + ' WITH PASSWORD = ' + QUOTENAME(@Password) + ', CHECK\_POLICY = OFF;';

EXEC sp\_executesql @CreateLoginSQL;

DECLARE @SetDefaultDBSQL NVARCHAR(MAX) = 'ALTER LOGIN ' + QUOTENAME(@LoginName) + ' WITH DEFAULT\_DATABASE = AP;';

EXEC sp\_executesql @SetDefaultDBSQL;

DECLARE @CreateUserSQL NVARCHAR(MAX) = 'CREATE USER ' + QUOTENAME(@LoginName) + ' FOR LOGIN ' + QUOTENAME(@LoginName) + ';';

EXEC sp\_executesql @CreateUserSQL;

DECLARE @AssignRoleSQL NVARCHAR(MAX) = 'EXEC sp\_addrolemember ''PaymentEntryMB'', ' + QUOTENAME(@LoginName) + ';';

EXEC sp\_executesql @AssignRoleSQL;

FETCH NEXT FROM loginCursor INTO @LoginName;

END

CLOSE loginCursor;

DEALLOCATE loginCursor;

**---** **Paste below this line the run results from executing the script you wrote—**

A screenshot of a computer

Description automatically generated

# **4. Unit 7 Lab Exercise 4**

Using the Management Studio, create a login ID named “FFalkXX” where the XX are your initials. Assign this login ID a password “FFal99999,” and set the default database to the AP database. Then, grant the login ID access to the AP database, create a user for the login ID named “FFalkXX”, and assign the user to the PaymentEntry role you created in exercise 1.

Note: If you get an error that says, “The MUST\_CHANGE option is not supported”, you can deselect the “Enforce password policy” option for the login ID.

**---** **Paste below this line a screen shot that shows the new login creation window with the FFalkXX information showing—**

A screenshot of a computer

Description automatically generated

**---** **Paste below a screen shot that shows the login properties window with the FFalkXX’s mapping to the AP database and membership in the PaymentEntryXX role you created in exercise 1.--**

A screenshot of a computer

Description automatically generated

# **5. Unit 7 Lab Exercise 5**

Write a script that removes the user-defined database role named PaymentEntryXX. Make sure to include a comment at the top of the script with your name and the date it was coded. Format this comment like the following sample:

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Script to remove the users and the role created during \*/

/\* the unit 7 lab. \*/

/\* \*/

/\* Coded by: Your Name \*/

/\* Coded on: MM/DD/YY \*/

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Code the script to do the following:

1. drop all the users from the PaymentEntryXX role.
2. drop the PaymentEntryXX role.

**--- Paste below this line the Script you wrote for this exercise --**

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Script to remove the users and the role created during \*/

/\* the unit 7 lab. \*/

/\* \*/

-- Coded by: Michael Bacu /

-- Coded on: 7/6/23 /

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

-- Drop the users from the PaymentEntryXX role

EXEC sp\_droprolemember 'PaymentEntryMB', 'XXBBrown';

EXEC sp\_droprolemember 'PaymentEntryMB', 'XXEEbbers1';

EXEC sp\_droprolemember 'PaymentEntryMB', 'XXDDyer';

EXEC sp\_droprolemember 'PaymentEntryMB', 'FFalkMB';

EXEC sp\_droprolemember 'PaymentEntryMB', 'MBBBrown';

EXEC sp\_droprolemember 'PaymentEntryMB', 'XXCChaplin1';

EXEC sp\_droprolemember 'PaymentEntryMB', 'XXEEbbers';

EXEC sp\_droprolemember 'PaymentEntryMB', 'MBEEbbers';

EXEC sp\_droprolemember 'PaymentEntryMB', 'XXCChaplin';

EXEC sp\_droprolemember 'PaymentEntryMB', 'XXBBrown1';

EXEC sp\_droprolemember 'PaymentEntryMB', 'XXDDyer1';

EXEC sp\_droprolemember 'PaymentEntryMB', 'MBCChaplin';

EXEC sp\_droprolemember 'PaymentEntryMB', 'GTCS01MB';

EXEC sp\_droprolemember 'PaymentEntryMB', 'MBDDyer';

-- Drop the PaymentEntryXX role

DROP ROLE PaymentEntryMB;

**---** **Paste below this line the run results from executing the script you wrote--**

A screenshot of a computer

Description automatically generated

# **6. Unit 7 Lab Exercise 6**

Write a script for the AP database as follows:

1. Start coding the script with a comment block. Format this comment like the following sample (make sure to revise the sample to reflect your name and the date you coded it):

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Schema Practice Script – Unit 7 Exercise 6 \*/

/\* This Script creates a schema, transfers a table into the \*/

/\* new schema, assigns the new schema as a user’s default \*/

/\* schema, and grants that user privileges for the new schema.\*/

/\* \*/

/\* Coded by: Your Name \*/

/\* Coded on: MM/DD/YY \*/

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

1. After the comment block, code the script to do the following:

(1) create a schema named AdminXX (where the XX are your initials)

(2) transfer the table named ContactUpdates from the dbo schema to the AdminXX schema

(3) assign the AdminXX schema as the default schema for the user named GTCS01XX (where the XX are your initials - this is a user you created in exercise 2 of this lab)

(4) grant all standard privileges except for REFERENCES and ALTER to GTCS01XX for the AdminXX schema

**--- Paste below this line the Script you wrote for this exercise --**

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Schema Practice Script – Unit 7 Exercise 6 \*/

/\* This Script creates a schema, transfers a table into the \*/

/\* new schema, assigns the new schema as a user’s default \*/

/\* schema, and grants that user privileges for the new schema.\*/

/\* \*/

/\* Coded by: Michael Bacu \*/

/\* Coded on: 7/6/2023 \*/

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

--CREATE SCHEMA AdminMB;

ALTER SCHEMA AdminMB TRANSFER [dbo].[ContactUpdates]; -- [AdminMB].[ContactUpdates];

ALTER USER GTCS01MB WITH DEFAULT\_SCHEMA = AdminMB;

GRANT SELECT, INSERT, UPDATE, DELETE, EXECUTE ON SCHEMA::AdminMB TO GTCS01MB;

**---** **Paste below this line the run results from executing the script you wrote--**

A screenshot of a computer

Description automatically generated