DAMG6210 - Data Management and Database Design

Homework 10

```
1.
CREATE FUNCTION CalculateDiscountedBudget (
@project_no CHAR(4),
@discount_percentage DECIMAL(5, 2)
)
RETURNS FLOAT
AS
BEGIN
DECLARE @budget FLOAT;
SELECT @budget = budget FROM project WHERE project_no = @project_no;
RETURN @budget - (@budget * @discount_percentage / 100);
END;
GO

Execute stored procedure:
SELECT dbo.CalculateDiscountedBudget('p2', 10) AS DiscountedBudget;
SELECT dbo.CalculateDiscountedBudget('p2', 40) AS DiscountedBudget;
```

```
Users > sushmaka > Documents > Dmdd > ■ SQLQuery_Oct23.sql
                                                                         器 Estimated Plan 📅 Enabl
▶ Run ☐ Cancel & Disconnect ® Change
                                      Database: sample2
 סבט שבו שבער בווט = שם
 527
       WHERE emp_no = 15000;
       SELECT * FROM EmpDeptChangeHistory;
 528
 529
 530
 531
 532
 533
       --CalculateDiscountedBudget
 534
       CREATE FUNCTION CalculateDiscountedBudget (
 535
            @project_no CHAR(4),
 536
           @discount_percentage DECIMAL(5, 2)
 537
       RETURNS FLOAT
 538
 539
       AS
 540
       BEGIN
 541
            DECLARE @budget FLOAT;
 542
            SELECT @budget = budget FROM project WHERE project_no = @project_no;
 543
           RETURN @budget - (@budget * @discount_percentage / 100);
 544
 545
       END;
 546
       G0
 547
       SELECT dbo.CalculateDiscountedBudget('p2', 10) AS DiscountedBudget;
 548
 549
        SELECT dbo.CalculateDiscountedBudget('p2', 40) AS DiscountedBudget;
 550
 551
 552
 553
Results
          Messages
     DiscountedBudget
      81000
1
     DiscountedBudget
1
      54000
```

```
CREATE TABLE project_log (
  log_id INT IDENTITY(1,1) PRIMARY KEY,
  emp_no INT,
  project_no CHAR(4),
  job VARCHAR(50),
  enter_date DATE,
  log_date DATETIME DEFAULT GETDATE()
);
\mathbf{GO}
CREATE TRIGGER trg_LogNewProjectAssignment
ON works on
AFTER INSERT
AS
BEGIN
  INSERT INTO project_log (emp_no, project_no, job, enter_date, log_date)
  SELECT emp_no, project_no, job, enter_date, GETDATE()
  FROM inserted;
END;
\mathbf{GO}
Execute stored procedure:
INSERT INTO employee (emp_no, emp_fname, emp_lname, dept_no)
VALUES (15009, 'Jenny', 'Kim', 'D3');
INSERT INTO works_on (emp_no, project_no, job, enter_date)
VALUES (15009, 'p3', 'Quality Analyst', '2024-11-12');
SELECT * FROM project_log WHERE emp_no = 15009 AND project_no = 'p3';
```

```
Users > sushmaka > Documents > Dmdd > ■ SQLQuery_Oct23.sql

ightharpoonup Run \square Cancel \mbox{\constant} Disconnect \mbox{\constant} Change
                                                                                                            Database: sample2
                                                                                                                                                                                                             음 Estimated Plan : Enable Actual Plan 🗸 Parse
  555
                     --project_log
   556
                      CREATE TABLE project_log (
   557
                                 log_id INT IDENTITY(1,1) PRIMARY KEY,
   558
                                 emp_no INT,
   559
                                 project_no CHAR(4),
   560
                                 job VARCHAR(50),
   561
                                 enter_date DATE,
                                 log_date DATETIME DEFAULT GETDATE()
   562
   563
                      );
  564
                     G0
   565
                      --AFTER INSERT
   566
                     \label{local_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continu
   567
                     ON works_on
                     AFTER INSERT
   568
   569
                     AS
   570
                      BEGIN
   571
                                 INSERT INTO project_log (emp_no, project_no, job, enter_date, log_date)
   572
                                  SELECT emp_no, project_no, job, enter_date, GETDATE()
   573
                                 FROM inserted;
  574
                      END;
  575
                      G0
  576
   577
                      INSERT INTO employee (emp_no, emp_fname, emp_lname, dept_no)
  578
                      VALUES (15009, 'Jenny', 'Kim', 'D3');
  579
   580
                      INSERT INTO works_on (emp_no, project_no, job, enter_date)
                      VALUES (15009, 'p3', 'Quality Analyst', '2024-11-12');
   581
   582
   583
                      SELECT ** FROM project_log WHERE emp_no = 15009 AND project_no = 'p3';
  584
   FOF
 Results
                          Messages

√ project_no √ job

                                                                                                                                                                                  enter_date

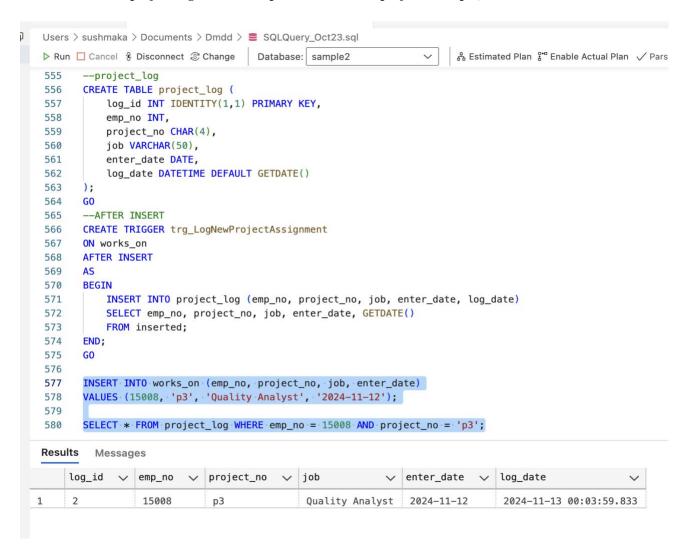
√ log_date

               log_id \vee emp_no
                                                   15009
                                                                                                                                    Quality Analyst
                                                                                                                                                                                                                                     2024-11-12 23:59:03.423
1
                 1
                                                                                      p3
                                                                                                                                                                                      2024-11-12
```

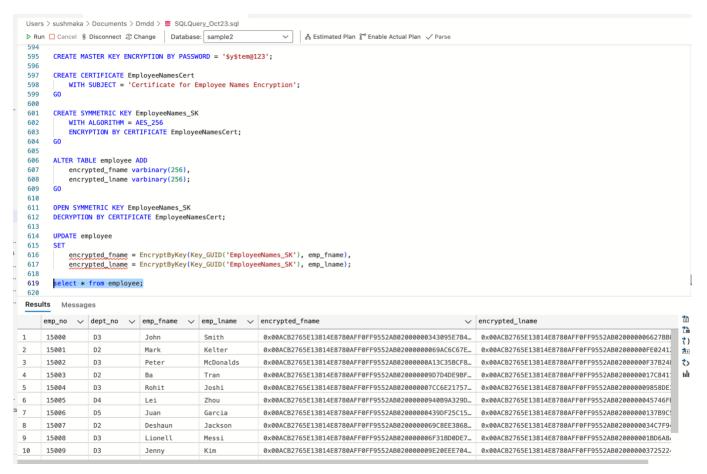
Execute stored procedure:

INSERT INTO works_on (emp_no, project_no, job, enter_date) VALUES (15008, 'p3', 'Quality Analyst', '2024-11-12');

SELECT * FROM project_log WHERE emp_no = 15008 AND project_no = 'p3';



```
3.
CREATE MASTER KEY ENCRYPTION BY PASSWORD = '$y$tem@123';
CREATE CERTIFICATE EmployeeNamesCert
 WITH SUBJECT = 'Certificate for Employee Names Encryption';
GO
CREATE SYMMETRIC KEY EmployeeNames SK
 WITH ALGORITHM = AES 256
 ENCRYPTION BY CERTIFICATE EmployeeNamesCert;
GO
ALTER TABLE employee ADD
 encrypted fname varbinary(256),
 encrypted_lname varbinary(256);
GO
OPEN SYMMETRIC KEY EmployeeNames SK
DECRYPTION BY CERTIFICATE EmployeeNamesCert;
UPDATE employee
SET
 encrypted_fname = EncryptByKey(Key_GUID('EmployeeNames_SK'), emp_fname),
 encrypted_lname = EncryptByKey(Key_GUID('EmployeeNames_SK'), emp_lname);
select * from employee;
Go
```



SELECT

emp_no,

CONVERT(VARCHAR(20), DecryptByKey(encrypted_fname)) AS emp_fname_decrypted, CONVERT(VARCHAR(20), DecryptByKey(encrypted_lname)) AS emp_lname_decrypted FROM employee;

CLOSE SYMMETRIC KEY EmployeeNames_SK;

