

Project 3

Nidhi Surya Prakash
SUID:215895619

Create 4 *Stored Procedures* which can be executed by database role “Graders” (GRANT EXECUTE ON SCHEMA::[yourSchema] TO Graders;) :

- SP to use cursor(s)
- SP to update data in a table (perform validation)
- SP to delete data from a table
- (5% bonus point) 1 SP of your own choice, performing a business action¹.

Create 1 *Function* which can be executed by “Graders”. Not the function you did for the lab.

You will also *create a view* (named as “Benefits”) which can be viewed by “Graders” (GRANT SELECT ON SCHEMA::[yourSchema] TO Graders;) that shows every employee’s name, ID, benefit’s type, benefit coverage, employee premium and employer premium.

Your deliverables will be:

- scripts used to create all of the DB objects described above; each object needs a **short** explanation as to its purpose or goal (i.e. “This SP does this, that and the other thing..”)
- screenshots demonstrating that the SPs work as expected (including valid/invalid inputs). Refer to our SP lab for more details.
- a text file with SELECTS against your views, EXECUTE against your function & stored procedures

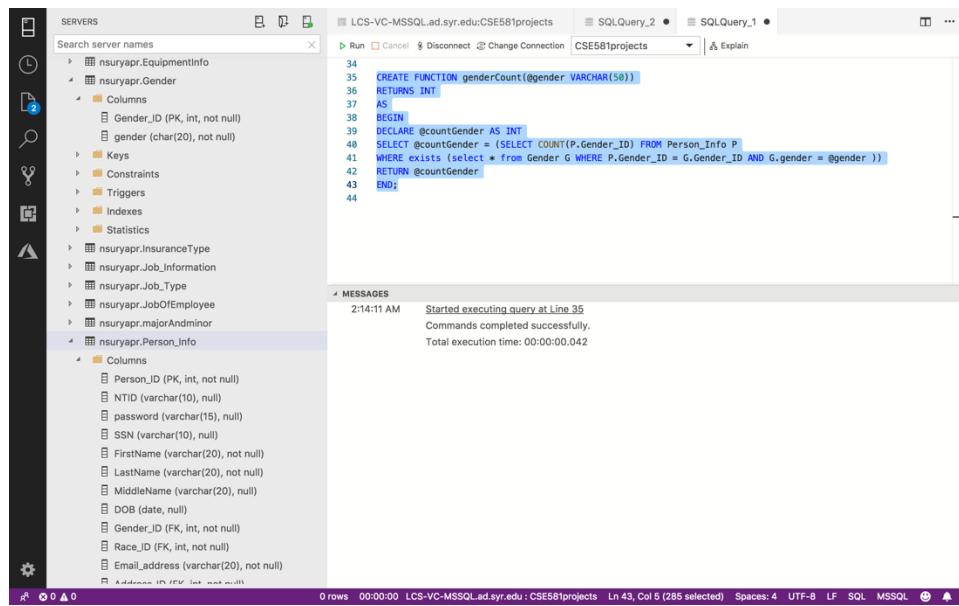
Requirements:

1. You **shall** create 1 view.
2. You **shall** create 4 stored procedures.
3. You **shall** create 1 function.
4. You **shall** submit all of your SQL code, used to create the DB objects (view, SPs and function).
5. You **shall** submit a short (a single sentence) explanation of what the purpose of each of the DB objects is.
6. You **shall** execute all of the SPs/function (provide screenshots of execution the way you did in the previous labs (SP & function labs)).
7. You **shall** submit a text file that will run SELECTs against of your view and execution of your SPs and function.

¹ You will get 5% bonus if this is a valid action within the scope of the business problem, and the SP is reasonably complex. In other words, a 4 line single-table select SP will get you nothing.

FUNCTION:

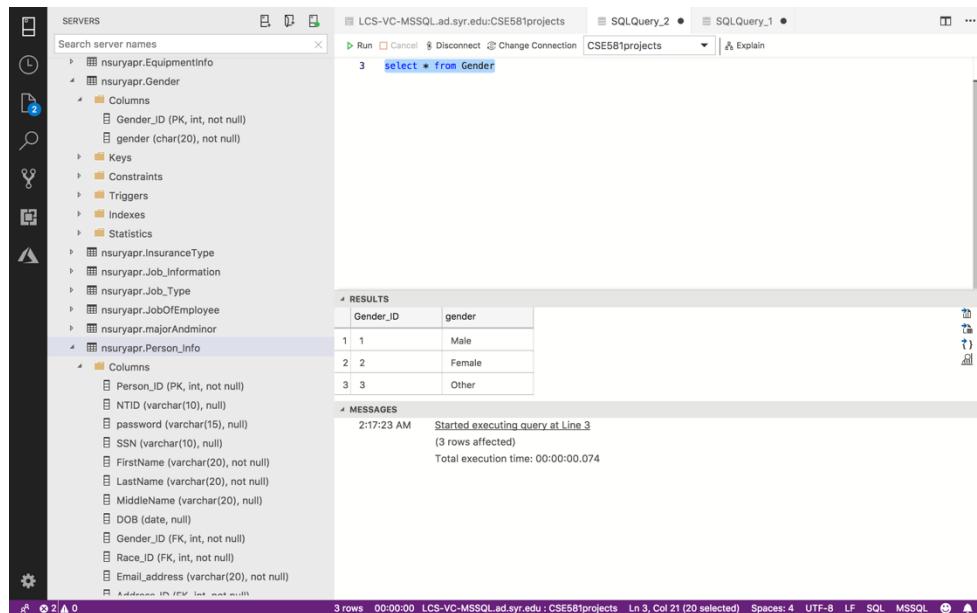
The function created will return the count of a specific gender.



Servers pane shows the database structure for nsuryapr. The Person_Info table is selected. The Object Explorer pane shows the creation of a function named genderCount.

```
CREATE FUNCTION genderCount(@gender VARCHAR(50))
RETURNS INT
AS
BEGIN
    DECLARE @countGender AS INT
    SELECT @countGender = (SELECT COUNT(P.Gender_ID) FROM Person_Info P
    WHERE exists (select * from Gender G WHERE P.Gender_ID = G.Gender_ID AND G.gender = @gender ))
    RETURN @countGender
END
```

Messages pane shows the query was started at 2:14:11 AM and completed successfully with a total execution time of 00:00:00.042.



Servers pane shows the database structure for nsuryapr. The Person_Info table is selected. The Object Explorer pane shows the execution of a query to select all rows from the Gender table.

```
select * from Gender
```

Results pane shows the output of the query, displaying three rows of data:

Gender_ID	gender
1	Male
2	Female
3	Other

Messages pane shows the query was started at 2:17:23 AM and completed successfully with a total execution time of 00:00:00.074.

LCS-VC-MSSQL.ad.syr.edu:CSE581projects SQLQuery_2 ● SQLQuery_1 ●

Run Cancel Disconnect Change Connection CSE581projects Explain

```
3 SELECT * FROM Gender
4
5 SELECT * FROM Person_Info
```

RESULTS

Person_ID	NTID	password	SSN	FirstName	LastName	MiddleName	DOB	Gender_ID	Race_ID
1	nsuryapr	pwd123	12345	Nidhi	Prakash	NULL	1995-05-24	2	7
2	apmaha	pwd234	23456	Anuja	Mahajan	P	1991-10-22	2	7
3	ndange	pwd456	78901	Nikhil	Dange	Nishikanth	1995-12-08	1	10
4	sshaikh	NULL	67891	Sofiya	Shaikh	NULL	1993-07-19	2	9
5	masheth	pwd789	11123	Manushi	Sheth	Manish	1995-09-19	2	10
6	maujawan	NULL	78654	Malhar	Ujawane	NULL	1995-08-27	1	9
7	viagrawa	pwd45	43561	Vikas	Agrawal	NULL	1992-01-26	1	8
8	akjagtap	NULL	67234	Akshay	Jagtap	NULL	1995-03-23	1	10
9	kvasudev	pwd555	55777	Kunal	Vasudev	NULL	1995-05-04	1	8
10	megsanj	NULL	24561	Meghana	Sanjay	NULL	1995-05-15	2	7
11	khebar	pwd890	67240	Kushal	Hebbar	Sriram	1995-04-22	1	9
12	apkemkar	NULL	32410	Apurva	Kemkar	NULL	1995-09-24	2	7
13	lwang	pwd544	777654	Li	Wang	NULL	1985-05-07	2	10
14	edmyu	pwd665	11122	Edmund	Yu	NULL	1975-04-02	2	10
15	ercer	pwd007	11122	Ercanili	Erhart	NULL	1980-02-12	2	10

MESSAGES
2:18:00 AM Started executing query at Line 5
(15 rows affected)

15 rows 00:00:00 LCS-VC-MSSQL.ad.syr.edu : CSE581projects Ln 5, Col 27 (26 selected) Spaces: 4 UTF-8 LF SQL MSSQL

LCS-VC-MSSQL.ad.syr.edu:CSE581projects SQLQuery_2 ● SQLQuery_1 ●

Run Cancel Disconnect Change Connection CSE581projects Explain

```
1
2
3
4 CREATE FUNCTION genderCount(@gender VARCHAR(50))
5 RETURNS INT
6 AS
7 BEGIN
8     DECLARE @countGender AS INT
9     SELECT @countGender = (SELECT COUNT(P.Gender_ID) FROM Person_Info P
10    WHERE exists (select * from Gender G WHERE P.Gender_ID = G.Gender_ID AND G.gender = @gender))
11    RETURN @countGender
12 END;
13
14 SELECT nsuryapr.genderCount('Female') as count_of_females;
15
16
```

RESULTS

count_of_females
1 9

MESSAGES
2:32:50 AM Started executing query at Line 14
(1 row affected)
Total execution time: 00:00:00.052

1 rows 00:00:00 LCS-VC-MSSQL.ad.syr.edu : CSE581projects Ln 15, Col 1 (59 selected) Spaces: 4 UTF-8 LF SQL MSSQL

VIEW:

Shows every employee's name, ID, benefit's type, benefit coverage, employee premium and employer premium.

The screenshot shows the Object Explorer on the left with the 'nsuryapr' database selected. In the center, the 'SQLQuery_2' tab contains the following SQL code:

```
15
16 CREATE VIEW Benefits_view
17 AS
18
19 SELECT a.FirstName + ' ' + a.LastName as Employee_Name, b.Employee_ID,
20 c.benefit_type as benefit_type ,d.CoverType_plan as cover_type ,f.Employee_PA ,f.Employer_PA
21 FROM
22 Person_Info a INNER JOIN Employee_Benefits b ON b.Employee_ID = a.Person_ID
23 INNER JOIN Benefits f ON b.Insurance_ID = f.Insurance_ID
24 INNER JOIN InsuranceType c ON f.Insurance_for_ID = c.Insurance_for_ID
25 INNER JOIN CoverType d ON f.Cover_type_ID = d.Cover_Type_ID
26
27
28
29
```

The 'MESSAGES' pane at the bottom shows the query execution details:

```
3:07:00 AM Started executing query at Line 17
Commands completed successfully.
Total execution time: 00:00:00.048
```

The screenshot shows the Object Explorer on the left with the 'nsuryapr' database selected. In the center, the 'SQLQuery_2' tab contains the following SQL code:

```
3 SELECT * FROM Gender
4
5 SELECT * FROM Person_Info
6
7 SELECT * FROM Benefits
8 SELECT * FROM Employee_Benefits
9 SELECT * FROM InsuranceType
10 SELECT * FROM CoverType
```

The 'RESULTS' pane on the right displays the output of the query:

	RESULTS	Insurance_ID	Insurance_for_ID	Cover_type_ID	Employee_PA	Employer_PA
1	1	1	3	5000	10000	
2	2	2	1	7000	6500	
3	3	3	1	10000	10000	
4	4	3	2	5000	7500	
5	5	1	4	10000	3000	
6	6	2	3	6500	10000	

The 'MESSAGES' pane at the bottom shows the query execution details:

```
3:08:47 AM Started executing query at Line 7
(6 rows affected)
Total execution time: 00:00:00.041
```

Servers

Search server names

```

3  SELECT * FROM Gender
4
5  SELECT * FROM Person_Info
6
7  SELECT * FROM Benefits
8  SELECT * FROM Employee_Benefits
9  SELECT * FROM InsuranceType
10 SELECT * FROM CoverType

```

RESULTS

Insurance_ID	Insurance_for_ID	Cover_type_ID	Employee_PA	Employer_PA
1 1	1	3	5000	10000
2 2	2	1	7000	6500
3 3	3	1	10000	10000
4 4	3	2	5000	7500
5 5	1	4	10000	3000
6 6	2	3	6500	10000

MESSAGES

3:08:47 AM Started executing query at Line 7
(6 rows affected)
Total execution time: 00:00:00.041

6 rows 00:00:00 LCS-VC-MSSQL.ad.syr.edu : CSE581projects Ln 9, Col 1 (32 selected) Spaces: 4 UTF-8 LF SQL MSSQL 😊 🔔

Servers

Search server names

```

3  SELECT * FROM Gender
4
5  SELECT * FROM Person_Info
6
7  SELECT * FROM Benefits
8  SELECT * FROM Employee_Benefits
9  SELECT * FROM InsuranceType
10 SELECT * FROM CoverType

```

RESULTS

Insurance_ID	Insurance_for_ID	Cover_type_ID	Employee_PA	Employer_PA
1 1	1	3	5000	10000
2 2	2	1	7000	6500
3 3	3	1	10000	10000
4 4	3	2	5000	7500
5 5	1	4	10000	3000
6 6	2	3	6500	10000

MESSAGES

3:08:47 AM Started executing query at Line 7
(6 rows affected)
Total execution time: 00:00:00.041

6 rows 00:00:00 LCS-VC-MSSQL.ad.syr.edu : CSE581projects Ln 10, Col 1 (28 selected) Spaces: 4 UTF-8 LF SQL MSSQL 😊 🔔

SERVERS Run Cancel Disconnect Change Connection CSE581projects SQLQuery_2 SQLQuery_1 Explain

```

3   SELECT * FROM Gender
4
5   SELECT * FROM Person_Info
6
7   SELECT * FROM Benefits
8   SELECT * FROM Employee_Benefits
9   SELECT * FROM InsuranceType
10  SELECT * FROM CoverType

```

RESULTS

Insurance_ID	Insurance_for_ID	Cover_type_ID	Employee_PA	Employer_PA
1 1	1	3	5000	10000
2 2	2	1	7000	6500
3 3	3	1	10000	10000
4 4	3	2	5000	7500
5 5	1	4	10000	3000
6 6	2	3	6500	10000

MESSAGES

3:08:47 AM Started executing query at Line 7
(6 rows affected)
Total execution time: 00:00:00.041

6 rows 00:00:00 LCS-VC-MSSQL.ad.syr.edu : CSE581projects Ln 10, Col 24 (23 selected) Spaces: 4 UTF-8 LF SQL MSSQL

SERVERS Run Cancel Disconnect Change Connection CSE581projects SQLQuery_2 SQLQuery_1 Explain

```

3   SELECT * FROM Gender
4
5   SELECT * FROM Person_Info
6
7   SELECT * FROM Benefits
8   SELECT * FROM Employee_Benefits
9   SELECT * FROM InsuranceType
10  SELECT * FROM CoverType

```

RESULTS

Insurance_ID	Insurance_for_ID	Cover_type_ID	Employee_PA	Employer_PA
1 1	1	3	5000	10000
2 2	2	1	7000	6500
3 3	3	1	10000	10000
4 4	3	2	5000	7500
5 5	1	4	10000	3000
6 6	2	3	6500	10000

MESSAGES

3:08:47 AM Started executing query at Line 7
(6 rows affected)
Total execution time: 00:00:00.041

6 rows 00:00:00 LCS-VC-MSSQL.ad.syr.edu : CSE581projects Ln 6, Col 1 (27 selected) Spaces: 4 UTF-8 LF SQL MSSQL

```

15 -----View
16
17 CREATE VIEW Benefits_view
18 AS
19 SELECT a.FirstName + ' ' + a.LastName AS Employee_Name, b.Employee_ID,
20 c.benefit_type AS benefit_type, d.CoverType_plan AS cover_type, f.Employee_PA ,f.Employer_PA
21 FROM
22 Person_Info a INNER JOIN Employee_Benefits b ON b.Employee_ID = a.Person_ID
23 INNER JOIN Benefits f ON b.Insurance_ID = f.Insurance_ID
24 INNER JOIN InsuranceType c ON f.Insurance_for_ID = c.Insurance_for_ID
25 INNER JOIN CoverType d ON f.Cover_type_ID = d.Cover_type_ID
26
27 SELECT * FROM Benefits_view
28
29
30
31
32

```

	Employee_Name	Employee_ID	benefit_type	cover_type	Employee_PA	Employer_PA
1	Akshay Jagtap	8	Vision	Employee With ...	5000	7500
2	Apurva Kemahr	12	vision	Employee Only	10000	10000
3	Li Wang	13	Denatal	Employee With ...	6500	10000
4	Edmund Yu	14	Health	Employee With ...	5000	10000
5	Ercanili Erhart	15	Denatal	Employee Only	7000	6500

MESSAGES

3:07:37 AM Started executing query at Line 27
(5 rows affected)
Total execution time: 00:00:00.060

SP to update data in a table (perform validation):

The update function takes the person ID, password provided, phone number and email and updates the appropriate table after verifying the following:

- If password is of atleast length 5 else throws an error message
- If phone number entered is valid
- If email entered already exists in the database.

```

30 -----Stored Procedure to update and perform validation -----
31
32 CREATE PROCEDURE updateProc(@ID INT,@password VARCHAR(50),@cellphone VARCHAR(20), @email VARCHAR(50))
33 AS
34 DECLARE @valid1 INT
35 SELECT @valid1 = (SELECT LEN(@password))
36 IF(@valid1>5)
37 BEGIN
38 UPDATE Person_Info
39 SET password = @password WHERE Person_ID = @ID
40 END
41 ELSE
42 BEGIN
43 PRINT 'Enter a valid password of length greater than 5'
44 END
45
46 IF ISNUMERIC(@cellphone) = 1
47 BEGIN
48 UPDATE Person_Info
49 SET Cell_phone = @cellphone WHERE Person_ID = @ID
50 END
51 ELSE
52 BEGIN
53 PRINT 'Invalid phone number provided'
54 END

```

MESSAGES

4:37:04 AM Started executing query at Line 32
Commands completed successfully.
Total execution time: 00:00:00.063

LCS-VC-MSSQL.ad.syr.edu:CSE581projects SQLQuery_2 SQLQuery_1.sql

```

5  SELECT * FROM Person_Info
6
7  SELECT * FROM Benefits
8  SELECT * FROM Employee_Benefits
9  SELECT * FROM InsuranceType
10 SELECT * FROM CoverType
11
12 SELECT * FROM Person_Info
13 SELECT * FROM Address

```

RESULTS

	Person_ID	NTID	password	SSN	FirstName	LastName	MiddleName	DOB	Gender_ID	Race_ID
1	1	nsuryapr	pwd123	12345	Nidhi	Prakash	NULL	1995-05-24	2	7
2	2	apmahaja	pwd234	23456	Anuja	Mahajan	P	1991-10-22	2	7
3	3	ndange	pwd456	78901	Nikhil	Dange	Nishikanth	1995-12-08	1	10
4	4	sshaikh	NULL	67891	Sofiya	Shaikh	NULL	1993-07-19	2	9
5	5	masheth	pwd789	11123	Manushi	Sheth	Manish	1995-09-19	2	10
6	6	maujawan	NULL	78654	Malhar	Ujawane	NULL	1995-08-27	1	9
7	7	viagrawa	pwd455	43561	Vikas	Agrawal	NULL	1992-01-26	1	8
8	8	akjagtap	NULL	67234	Akshay	Jagtap	NULL	1995-03-23	1	10
9	9	kvasudev	pwd555	55777	Kunal	Vasudev	NULL	1995-05-04	1	8
10	10	megsanj	NULL	24561	Meghana	Sanjay	NULL	1995-05-15	2	7
11	11	khebbar	pwd890	67240	Kushal	Hebbar	Sriram	1995-04-22	1	9
12	12	apkemkar	NULL	32410	Apurva	Kemakr	NULL	1995-09-24	2	7
13	13	lwang	pwd544	777654	Li	Wang	NULL	1985-05-07	2	10
14	14	edmyu	pwd865	11122	Edmund	Yu	NULL	1975-04-02	2	10
15	15	ercer	pwd007	11122	Ercanilli	Erhart	NULL	1980-02-12	2	10

MESSAGES

(15 rows affected)

LCS-VC-MSSQL.ad.syr.edu:CSE581projects SQLQuery_2 SQLQuery_1.sql

```

5  SELECT * FROM Person_Info
6
7  SELECT * FROM Benefits
8  SELECT * FROM Employee_Benefits
9  SELECT * FROM InsuranceType
10 SELECT * FROM CoverType
11
12 SELECT * FROM Person_Info
13 SELECT * FROM Address

```

RESULTS

	FirstName	LastName	MiddleName	DOB	Gender_ID	Race_ID	Email_address	Address_ID	Cell_phone
345	Nidhi	Prakash	NULL	1995-05-24	2	7	nsuryapr@syr....	7	3156641844
3456	Anuja	Mahajan	P	1991-10-22	2	7	apmahaja@syr....	1	3156641841
78901	Nikhil	Dange	Nishikanth	1995-12-08	1	10	ndange@syr.edu	10	3156641845
7891	Sofiya	Shaikh	NULL	1993-07-19	2	9	sshaikh@syr.edu	9	3156694789
123	Manushi	Sheth	Manish	1995-09-19	2	10	maasheth@syr....	2	2158956198
78654	Malhar	Ujawane	NULL	1995-08-27	1	9	maujawan@syr....	3	3167739265
3561	Vikas	Agrawal	NULL	1992-01-26	1	8	viagrawa@syr....	1	2637186054
7234	Akshay	Jagtap	NULL	1995-03-23	1	10	akjagtap123@s....	4	3159259935
5777	Kunal	Vasudev	NULL	1995-05-04	1	8	kvasudev@usc....	2	2138924630
4561	Meghana	Sanjay	NULL	1995-05-15	2	7	megsanj@uic.e...	7	2124416678
7240	Kushal	Hebbar	Sriram	1995-04-22	1	9	khebbar@clem....	4	8900022661
2410	Apurva	Kemakr	NULL	1995-09-24	2	7	apkemkar@syr....	8	4028976541
77654	Li	Wang	NULL	1985-05-07	2	10	liwang@syr.edu	11	3156641844
122	Edmund	Yu	NULL	1975-04-02	2	10	edmyu@syr.edu	12	3156641325
122	Ercanilli	Erhart	NULL	1980-02-12	2	10	ercer@syr.edu	13	3156641325

MESSAGES

(15 rows affected)

LCS-VC-MSSQL.ad.syr.edu:CSE581projects SQLQuery_2 SQLQuery_1.sql

```
65 EXEC updateProc 5,'syracuse','3156678942,'msheeth@syr.edu'
66 EXEC updateProc 8,'aksh','3xbxhhxh','akjagtap123@syr.edu'
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
```

MESSAGES

4:38:24 AM Started executing query at Line 66
(1 row affected)
(1 row affected)
(1 row affected)

Total execution time: 00:00:00.047

0 rows 00:00:00 LCS-VC-MSSQL.ad.syr.edu : CSE581projects Ln 67, Col 1 (57 selected) Spaces: 4 UTF-8 LF SQL MSSQL

LCS-VC-MSSQL.ad.syr.edu:CSE581projects SQLQuery_2 SQLQuery_1.sql

```
5 SELECT * FROM Person_Info
6
7 SELECT * FROM Benefits
8 SELECT * FROM Employee_Benefits
9 SELECT * FROM InsuranceType
10 SELECT * FROM CoverType
11
12 SELECT * FROM Person_Info
13 SELECT * FROM Address
```

RESULTS

Person_ID	NTID	password	SSN	FirstName	LastName	MiddleName	DOB	Gender_ID	Race_ID
1	nsuryapr	pwd123	12345	Nidhi	Prakash	NULL	1995-05-24	2	7
2	apmahaha	pwd234	23456	Anuja	Mahajan	P	1991-10-22	2	7
3	ndange	pwd456	78901	Nikhil	Dange	Nishikanth	1995-12-08	1	10
4	sshalkh	NULL	67891	Sofiya	Shaikh	NULL	1993-07-19	2	9
5	masheth	syracuse	11123	Manushi	Sheth	Manish	1995-09-19	2	10
6	maujawan	NULL	78654	Malhar	Ujawane	NULL	1995-08-27	1	9
7	viagrava	pwd455	43561	Vikas	Agrawal	NULL	1992-01-26	1	8
8	akjagtap	NULL	67234	Akshay	Jagtap	NULL	1995-03-23	1	10
9	kvasudev	pwd555	55777	Kunal	Vasudev	NULL	1995-05-04	1	8
10	meganj	NULL	24561	Meghana	Sanjay	NULL	1995-05-15	2	7
11	khebar	pwd690	67240	Kushal	Hebar	Sriram	1995-04-22	1	9
12	apkemkar	NULL	32410	Apurva	Kemkar	NULL	1995-09-24	2	7
13	lwang	pwd644	777654	Li	Wang	NULL	1985-05-07	2	10
14	edmyu	pwd665	11122	Edmund	Yu	NULL	1975-04-02	2	10

MESSAGES

Total execution time: 00:00:00.042

15 rows 00:00:00 LCS-VC-MSSQL.ad.syr.edu : CSE581projects Ln 13, Col 1 (26 selected) Spaces: 4 UTF-8 LF SQL MSSQL

LCS-VC-MSSQL.ad.syr.edu:CSE581projects SQLQuery_2 ● SQLQuery_1.sql ●

```

5  SELECT * FROM Person_Info
6
7  SELECT * FROM Benefits
8  SELECT * FROM Employee_Benefits
9  SELECT * FROM InsuranceType
10 SELECT * FROM CoverType
11
12 SELECT * FROM Person_Info
13 SELECT * FROM Address

```

RESULTS

N	FirstName	LastName	MiddleName	DOB	Gender_ID	Race_ID	Email_address	Address_ID	Cell_phone
2345	Nidhi	Prakash	NULL	1995-05-24	2	7	nsuryapr@syr....	7	3156641944
3456	Anuja	Mahajan	P	1991-10-22	2	7	apmaha@sy....	1	3156641841
3901	Nikhil	Dange	Nishikanth	1995-12-08	1	10	ndange@syr.edu	10	3156641845
7891	Sofiya	Shaikh	NULL	1993-07-19	2	9	sshaikh@syr.edu	9	3156694789
123	Manushl	Sheth	Manish	1995-09-19	2	10	msheth@syr.edu	2	3156678942
3654	Malhar	Ujawane	NULL	1995-08-27	1	9	maujawan@syr....	3	3167739265
3561	Vikas	Agrawal	NULL	1992-01-26	1	8	vigrawa@syr....	1	2637186054
7234	Akshay	Jagtap	NULL	1995-03-23	1	10	akjagtap123@s....	4	3159259935
5777	Kunal	Vasudev	NULL	1995-05-04	1	8	kvasudev@usc....	2	2135924630
4561	Meghana	Sanjay	NULL	1995-05-15	2	7	megsan@uic.e...	7	2124416678
7240	Kushal	Hebbar	Sriram	1995-04-22	1	9	khebbar@clem....	4	8900022661
2410	Apurva	Kemakr	NULL	1995-09-24	2	7	apkemkar@syr....	8	4028976541
77654	Li	Wang	NULL	1985-05-07	2	10	liwang@syr.edu	11	3156641844
122	Edmund	Yu	NULL	1975-04-02	2	10	edmvu@svr.edu	12	3156641325

MESSAGES

```

Total execution time: 00:00:00.042

```

LCS-VC-MSSQL.ad.syr.edu:CSE581projects SQLQuery_2 ● SQLQuery_1.sql ●

```

65
66  EXEC updateProc 5,'syracuse', '3156678942', 'msheth@syr.edu'
67  EXEC updateProc 8, 'aksh', '3xbxhhxhh', 'akjagtap123@syr.edu'
68
69
70
71
72
73
74
75
76
77

```

MESSAGES

```

4:39:00 AM Started executing query at Line 67
Enter a valid password of length greater than 5
Invalid phone number provided
EMAIL ID already exists
Total execution time: 00:00:00.039

```

SP to use cursor(s):

The cursor stored procedure calculates the average grade of student by taking in the student ID as a parameter.

The screenshot shows the SQL Server Management Studio interface. In the Object Explorer, under SERVERS, there are four connections listed: LCS-VC-MSSQL.ad.syr.edu, CSE581pr..., LCS-VC-MSSQL.ad.syr.edu, <default>..., LCS-VC-MSSQL.ad.syr.edu, CSE581a..., and LCS-VC-MSSQL.ad.syr.edu, Northwin... The current database is CSE581projects. In the center pane, a new stored procedure is being created:

```
36 ----- Stored procedure to use cursor -----
37 CREATE PROCEDURE AvgGrades_cursor2(@studentID int, @average as DECIMAL(5,2) OUT)
38 AS
39 BEGIN
40     DECLARE @aG DECIMAL(10,2),@finalGrade int = 0, @c float=0.0, @grades int
41     DECLARE avgCursor CURSOR FOR
42         SELECT final_marks FROM nsuryapr.CourseEnrollment WHERE Student_ID = @studentID AND
43             final_marks is not NULL
44     OPEN avgCursor
45     FETCH NEXT FROM avgCursor INTO @grades
46     WHILE @@FETCH_STATUS = 0
47     BEGIN
48         SELECT @c = @c + 1
49         SELECT @finalGrade = @finalGrade + @grades
50         FETCH NEXT FROM avgCursor INTO @grades
51     END
52     SELECT @aG = ROUND(@finalGrade / @c,2)
53     CLOSE avgCursor
54     DEALLOCATE avgCursor
55     SELECT @average = @aG
56 END
57
58 DECLARE @avg as DECIMAL(5,2)
59 EXEC AvgGrades_cursor2 2, @avg out
60 SELECT @avg [Average of the student]
```

In the Messages pane, the output is:

```
10:31:14 PM Started executing query at Line 37
Commands completed successfully.
Total execution time: 00:00:00.048
```

The screenshot shows the SQL Server Management Studio interface. In the Object Explorer, under SERVERS, there are four connections listed: LCS-VC-MSSQL.ad.syr.edu, CSE581pr..., LCS-VC-MSSQL.ad.syr.edu, <default>..., LCS-VC-MSSQL.ad.syr.edu, CSE581a..., and LCS-VC-MSSQL.ad.syr.edu, Northwin... The current database is CSE581projects. In the center pane, a simple query is being run:

```
1 SELECT * from CourseEnrollment
2
3
4
5
6
7
8
9
```

The results pane displays the data from the CourseEnrollment table:

	Student_ID	CRN	EnrollmentStatus...	midterm_grade	final_grade	final_marks
1	1	1	1	A-	A+	99
2	2	1	1	A+	A+	95
3	2	2	1	A-	A-	89
4	3	3	1	A+	A-	90
5	3	4	1	A-	A-	92
6	4	5	1	B-	B+	99
7	6	3	1	C-	B-	90
8	6	4	1	B-	B+	85
9	9	4	2	A+	A+	91

In the Messages pane, the output is:

```
10:00:14 PM Started executing query at Line 1
(9 rows affected)
Total execution time: 00:00:00.043
```

SQLQuery1.sql • SQLQuery2 • proj2_Nidhi (1).sql • SQLQuery1 •

Run Cancel Disconnect Change Connection CSE581projects Explain

```
36 ━━━━━━━━━━━━━━━━━━━━━━━━ Stored procedure to use cursor
37 CREATE PROCEDURE AvgGrades_cursor2(@studentID as int, @average as DECIMAL(5,2) OUT)
38 AS
39 BEGIN
40     DECLARE @G DECIMAL(10,2),@finalGrade int = 0, @c float=0.0, @grades int
41     DECLARE avgCursor CURSOR FOR
42     SELECT final_marks FROM nsuryspr.CourseEnrollment WHERE Student_ID = @studentID AND
43     final_marks is not NULL
44     OPEN avgCursor
45     FETCH NEXT FROM avgCursor INTO @grades
46     WHILE @@FETCH_STATUS = 0
47     BEGIN
48         SELECT @c = @c + 1
49         SELECT @finalGrade = @finalGrade + @grades
50         FETCH NEXT FROM avgCursor INTO @grades
51     END
52     SELECT @avg = ROUND(@finalGrade / @c,2)
53     CLOSE avgCursor
54     DEALLOCATE avgCursor
55     SELECT @average = @avg
56 END
57
58 DECLARE @avg as DECIMAL(5,2)
59 EXEC AvgGrades_cursor2 2, @avg out
60 SELECT @avg [Average of the student]
61
```

RESULTS

Average of the student
1 92.00

MESSAGES

10:31:38 PM Started executing query at Line 58
(1 row affected)
Total execution time: 00:00:00.052

1 rows 00:00:00 LCS-VC-MSSQL.ad.syr.edu : CSE581projects Ln 60, Col 37 (101 selected) Spaces: 4 UTF-8 LF SQL MSSQL

SP to delete data from a table:

If an employee decides to leave the university, the employee's records and all classes handled by him/her will be removed from the database.

Servers

Search server names

LCS-VC-MSSQL.ad.syr.edu, CSE581pr...
LCS-VC-MSSQL.ad.syr.edu, <default>...
LCS-VC-MSSQL.ad.syr.edu, CSE581la...
LCS-VC-MSSQL.ad.syr.edu, Northwind...

SQLQuery_1.sql • SQLQuery_2 • SQLQuery_1 •

Run Cancel Disconnect Change Connection CSE581/projects Explain

99
100 CREATE PROCEDURE deleteCourse(@emp INT)
101 AS
102 DECLARE @Course taught INT
103 SELECT @Course taught = (SELECT CRN FROM EmployeeAndCourse WHERE Employee_ID = @emp)
104 DELETE FROM EmployeeAndCourse WHERE CRN = @Course taught
105 DELETE FROM CourseEnrollment WHERE CRN = @Course taught
106 DELETE FROM Course_schedule WHERE CRN = @Course taught
107 DELETE FROM JobOfferEmployee WHERE Employee_ID = @emp
108 DELETE FROM Employee_Benefits WHERE Employee_ID = @emp
109 DELETE FROM Employee WHERE Employee_ID = @emp
110
111 EXEC deleteCourse 16
112
113 SELECT * FROM EmployeeAndCourse
114 SELECT * FROM CourseEnrollment
115 SELECT * FROM Course_schedule

MESSAGES

9:27:15 PM Started executing query at Line 100
Commands completed successfully.
Total execution time: 00:00:00.046

0 rows 00:00:00 LCS-VC-MSSQL.ad.syr.edu : CSE581projects Ln 109, Col 46 (469 selected) Spaces: 4 UTF-8 LF SQL MSSQL

Servers

SQLQuery_1.sql • SQLQuery_2 • SQLQuery_1 • CSE581projects

```

99
100 CREATE PROCEDURE deleteCourse(@emp INT)
101 AS
102 DECLARE @Course taught INT
103 SELECT @Course taught = (SELECT CRN FROM EmployeeAndCourse WHERE Employee_ID = @emp)
104 DELETE FROM EmployeeAndCourse WHERE CRN = @Course taught
105 DELETE FROM CourseEnrollment WHERE CRN = @Course taught
106 DELETE FROM Course_schedule WHERE CRN = @Course taught
107 DELETE FROM JobOfferEmployee WHERE Employee_ID = @emp
108 DELETE FROM Employee_Benefits WHERE Employee_ID = @emp
109 DELETE FROM Employee WHERE Employee_ID = @emp
110
111 EXEC deleteCourse 16
112
113 SELECT * FROM EmployeeAndCourse
114 SELECT * FROM CourseEnrollment
115 SELECT * FROM Course_schedule

```

MESSAGES

9:27:25 PM Started executing query at Line 111
 (1 row affected)
 (1 row affected)

Total execution time: 00:00:00.056

0 rows 00:00:00 LCS-VC-MSSQL.ad.syr.edu : CSE581projects Ln 112, Col 1 (21 selected) Spaces: 4 UTF-8 LF SQL MSSQL

Employee and course table before and after execution of delete stored procedure

Servers

SQLQuery_1.sql • SQLQuery_2 • SQLQuery_1 • CSE581projects

```

113
114 SELECT * FROM EmployeeAndCourse
115 SELECT * FROM CourseEnrollment
116 SELECT * FROM Course_schedule
117 SELECT * FROM JobOfferEmployee
118 SELECT * FROM Employee_Benefits
119 SELECT * FROM Employee
120
121
122
123
124
125
126
127
128
129

```

RESULTS

CRN	Employee_ID
1 1	13
2 3	12
3 4	14
4 5	15
5 11	16

MESSAGES

9:24:32 PM Started executing query at Line 114
 (5 rows affected)

Total execution time: 00:00:00.128

5 rows 00:00:00 LCS-VC-MSSQL.ad.syr.edu : CSE581projects Ln 115, Col 1 (32 selected) Spaces: 4 UTF-8 LF SQL MSSQL

Servers

SQLQuery_1.sql • SQLQuery_2 • SQLQuery_1 • CSE581projects

```

114
115 SELECT * FROM EmployeeAndCourse
116 SELECT * FROM CourseEnrollment
117 SELECT * FROM Course_schedule
118 SELECT * FROM JobOfferEmployee
119 SELECT * FROM Employee_Benefits
120
121
122
123
124
125
126
127
128

```

RESULTS

CRN	Employee_ID
1 1	13
2 3	12
3 4	14
4 5	15

MESSAGES

9:27:43 PM Started executing query at Line 115
 (4 rows affected)

Total execution time: 00:00:00.114

4 rows 00:00:00 LCS-VC-MSSQL.ad.syr.edu : CSE581projects Ln 116, Col 1 (32 selected) Spaces: 4 UTF-8 LF SQL MSSQL

Course enrollment table before and after execution of delete stored procedure

Servers: LCS-VC-MSSQL.ad.syr.edu, CSE581pr...

```

113
114    SELECT * FROM EmployeeAndCourse
115    SELECT * FROM CourseEnrollment
116    SELECT * FROM Course_schedule
117    SELECT * FROM JobOfferEmployee
118    SELECT * FROM Employee_Benefits
119    SELECT * FROM Employee
120
121
122
123
124
125
126
127
128
129

```

RESULTS

Student_ID	CRN	EnrollmentStatus...	midterm_grade	final_grade
1 1	1	1	A-	A+
2 2	1	1	A+	A+
3 3	4	1	A-	A-
4 4	5	1	B-	B+
5 6	3	1	C-	B-
6 9	4	2	A+	A+
7 10	11	1	NULL	NULL

MESSAGES

```

9:25:09 PM Started executing query at Line 115
(7 rows affected)
Total execution time: 00:00:00.047

```

Servers: LCS-VC-MSSQL.ad.syr.edu, CSE581pr...

```

112
113    SELECT * FROM EmployeeAndCourse
114    SELECT * FROM CourseEnrollment
115    SELECT * FROM Course_schedule
116    SELECT * FROM JobOfferEmployee
117    SELECT * FROM Employee_Benefits
118    SELECT * FROM Employee
119
120
121
122
123
124
125
126
127
128

```

RESULTS

Student_ID	CRN	EnrollmentStatus...	midterm_grade	final_grade
1 1	1	1	A-	A+
2 2	1	1	A+	A+
3 3	4	1	A-	A-
4 4	5	1	B-	B+
5 6	3	1	C-	B-
6 9	4	2	A+	A+

MESSAGES

```

9:28:09 PM Started executing query at Line 114
(8 rows affected)
Total execution time: 00:00:00.052

```

Course schedule table before and after execution of delete stored procedure

Servers: LCS-VC-MSSQL.ad.syr.edu, CSE581pr...

```

113
114    SELECT * FROM EmployeeAndCourse
115    SELECT * FROM CourseEnrollment
116    SELECT * FROM Course_schedule
117    SELECT * FROM JobOfferEmployee
118    SELECT * FROM Employee_Benefits
119    SELECT * FROM Employee
120
121
122
123
124
125
126
127
128
129

```

RESULTS

CRN	Course_code	Course_number	Semester_ID	Timings_ID	Classroom_ID
1 1	581	1	3	4	2
2 2	314	3	4	1	2
3 3	628	6	3	2	3
4 4	701	8	5	3	4
5 5	622	12	1	5	5
6 11	798	13	3	4	3

MESSAGES

```

9:28:37 PM Started executing query at Line 116
(6 rows affected)
Total execution time: 00:00:00.047

```

Servers: LCS-VC-MSSQL.ad.syr.edu, CSE581proj...

SQLQuery_1.sql • SQLQuery_2 • SQLQuery_1

```

112
113 SELECT * FROM EmployeeAndCourse
114 SELECT * FROM CourseEnrollment
115 SELECT * FROM Course_Schedule
116 SELECT * FROM Job0fEmployee
117 SELECT * FROM Employee_Benefits
118 SELECT * FROM Employee
119
120
121
122
123
124
125
126
127
128

```

RESULTS

CRN	Course_code	Course_number	Semester_ID	Timings_ID	Classroom_ID
1 1	581	1	3	4	2
2 2	314	3	4	1	2
3 3	628	6	3	2	3
4 4	701	8	5	3	4
5 5	522	12	1	5	5

MESSAGES

9:28:34 PM Started executing query at Line 115
(5 rows affected)
Total execution time: 0:00:00.048

5 rows 00:00:00 LCS-VC-MSSQL.ad.syr.edu : CSE581proj... Ln 116, Col 1 (30 selected) Spaces: 4 UTF-8 LF SQL MSSQL

Job of employee table before and after execution of delete stored procedure

Servers: LCS-VC-MSSQL.ad.syr.edu, CSE581proj...

SQLQuery_1.sql • SQLQuery_2 • SQLQuery_1

```

113
114 SELECT * FROM EmployeeAndCourse
115 SELECT * FROM CourseEnrollment
116 SELECT * FROM Course_Schedule
117 SELECT * FROM Job0fEmployee
118 SELECT * FROM Employee_Benefits
119 SELECT * FROM Employee
120
121
122
123
124
125
126
127
128
129

```

RESULTS

Employee_ID	Job_code
1 8	5
2 12	1
3 13	5
4 14	1
5 15	1
6 16	5

MESSAGES

9:25:59 PM Started executing query at Line 117
(6 rows affected)
Total execution time: 0:00:00.058

6 rows 00:00:00 LCS-VC-MSSQL.ad.syr.edu : CSE581proj... Ln 118, Col 1 (28 selected) Spaces: 4 UTF-8 LF SQL MSSQL

Servers: LCS-VC-MSSQL.ad.syr.edu, CSE581proj...

SQLQuery_1.sql • SQLQuery_2 • SQLQuery_1

```

112
113 SELECT * FROM EmployeeAndCourse
114 SELECT * FROM CourseEnrollment
115 SELECT * FROM Course_Schedule
116 SELECT * FROM Job0fEmployee
117 SELECT * FROM Employee_Benefits
118 SELECT * FROM Employee
119
120
121
122
123
124
125
126
127
128

```

RESULTS

Employee_ID	Job_code
1 8	5
2 12	1
3 13	5
4 14	1
5 15	1

MESSAGES

9:28:52 PM Started executing query at Line 116
(5 rows affected)
Total execution time: 0:00:00.078

6 rows 00:00:00 LCS-VC-MSSQL.ad.syr.edu : CSE581proj... Ln 117, Col 1 (28 selected) Spaces: 4 UTF-8 LF SQL MSSQL

Employee benefits table before and after execution of delete stored procedure

The screenshot shows the SQL Server Management Studio interface. In the top pane, there are three tabs: SQLQuery_1.sql, SQLQuery_2, and SQLQuery_1. The SQLQuery_1 tab contains the following T-SQL code:

```
113 SELECT * FROM EmployeeAndCourse
114 SELECT * FROM CourseEnrollment
115 SELECT * FROM Course_schedule
116 SELECT * FROM JobOfferEmployee
117 SELECT * FROM Employee_Benefits
118 SELECT * FROM Employee
119
120
121
122
123
124
125
126
127
128
129
```

In the middle pane, the RESULTS tab displays the contents of the Employee_Benefits table:

Employee_ID	Insurance_ID
1	8
2	12
3	13
4	14
5	15
6	16

In the bottom pane, the MESSAGES tab shows the execution details:

```
9:26:13 PM Started executing query at Line 118
(6 rows affected)
Total execution time: 00:00:00.048
```

At the bottom of the interface, status information is displayed: 6 rows, 00:00:00, LCS-VC-MSSQL.ad.syr.edu : CSE581\projects, Ln 119, Col 1 (32 selected), Spaces: 4, UTF-8, LF, SQL, MSSQL.

The screenshot shows the SQL Server Management Studio interface. The layout is identical to the first screenshot, but the results have changed. The Employee_Benefits table now contains only five rows:

Employee_ID	Insurance_ID
1	8
2	12
3	13
4	14
5	15

The MESSAGES tab shows the execution details again:

```
9:29:10 PM Started executing query at Line 117
(5 rows affected)
Total execution time: 00:00:00.046
```

At the bottom, the status information remains the same: 6 rows, 00:00:00, LCS-VC-MSSQL.ad.syr.edu : CSE581\projects, Ln 119, Col 1 (32 selected), Spaces: 4, UTF-8, LF, SQL, MSSQL.

Employee table before and after execution of delete stored procedure

The screenshot shows the SQL Server Management Studio interface with three query panes: SQLQuery_1.sql, SQLQuery_2, and SQLQuery_1. The SQLQuery_1.sql pane contains a series of SELECT statements from various tables. The results pane shows the Employee table with 6 rows. The messages pane indicates the query started at 9:26:34 PM, affected 6 rows, and took 0:00:00.140.

Employee_ID	Annual_Salary	
1	8	35000
2	12	32000
3	13	40000
4	14	38000
5	15	30000
6	16	47000

MESSAGES
9:26:34 PM Started executing query at Line 119
(6 rows affected)
Total execution time: 0:00:00.140

The screenshot shows the SQL Server Management Studio interface with three query panes: SQLQuery_1.sql, SQLQuery_2, and SQLQuery_1. The SQLQuery_1.sql pane contains the same series of SELECT statements. The results pane shows the Employee table with 5 rows. The messages pane indicates the query started at 9:29:30 PM, affected 5 rows, and took 0:00:00.048.

Employee_ID	Annual_Salary	
1	8	35000
2	12	32000
3	13	40000
4	14	38000
5	15	30000

MESSAGES
9:29:30 PM Started executing query at Line 118
(5 rows affected)
Total execution time: 0:00:00.048

BONUS QUESTION – STORED PROCEDURE

If some student/faculty would like to know immediately the details of a class he/she has to attend or know the details of – such as start time and the professor handling the lecture. The stored procedure takes in the class registration number (crn) and gives the details.

The screenshot shows the SQL Server Management Studio interface. In the Object Explorer on the left, under the 'nsuryapr.Course_schedule' node, there is a 'Columns' folder containing columns: CRN, Course_code, Course_number, Semester_ID, Timings_ID, and Classroom_ID. In the center pane, a T-SQL script is being run:

```
1 CREATE PROCEDURE CourseInfoToAttend (@crn INT)
2 AS
3     SELECT p.FirstName + ' ' + p.LastName AS Professor, b.Building_name, c.Class_number, ct.StartTime, ct.EndTime
4     FROM EmployeeAndCourse es INNER JOIN Course_schedule cs ON es.CRN = cs.CRN
5     INNER JOIN Person_Info p ON es.Employee_ID = p.Person_ID
6     INNER JOIN Classroom c ON cs.Classroom_ID = c.Classroom_ID
7     INNER JOIN Buildings b ON b.Building_ID = c.Building_ID
8     INNER JOIN CourseTimings ct ON ct.Timings_ID = cs.Timings_ID
9 
```

In the 'MESSAGES' pane at the bottom, it says "Started executing query at Line 1" and "Commands completed successfully. Total execution time: 00:00:00.042".

The screenshot shows the SQL Server Management Studio interface. The same T-SQL script is run again, but now includes the EXEC command:

```
1 CREATE PROCEDURE CourseInfoToAttend (@crn INT)
2 AS
3     SELECT p.FirstName + ' ' + p.LastName AS Professor, b.Building_name, c.Class_number, ct.StartTime, ct.EndTime
4     FROM EmployeeAndCourse es INNER JOIN Course_schedule cs ON es.CRN = cs.CRN
5     INNER JOIN Person_Info p ON es.Employee_ID = p.Person_ID
6     INNER JOIN Classroom c ON cs.Classroom_ID = c.Classroom_ID
7     INNER JOIN Buildings b ON b.Building_ID = c.Building_ID
8     INNER JOIN CourseTimings ct ON ct.Timings_ID = cs.Timings_ID
9
10 EXEC CourseInfoToAttend 1
11
12 
```

The results pane shows the output of the stored procedure:

	Professor	Building_name	Class_number	StartTime	EndTime
1	Li Wang	Slocum Hall	401	7:00 pm	8:00 pm
2	Apurva Kemakr	Steel Hall	234	11:00 am	12:00 pm
3	Edmund Yu	Watson Theater	500	5:00 pm	6:35 pm
4	Ercanli Erhart	Life Sciences	311	3:00 pm	4:15 pm

In the 'MESSAGES' pane, it says "Started executing query at Line 10", "(4 rows affected)", and "Total execution time: 00:00:00.057".