

## Assignment - Week-4 Data Warehousing and Analytics

*Student's Full Name: Anand Ramaswamy Jayshree*

*Course Title: INFO531*

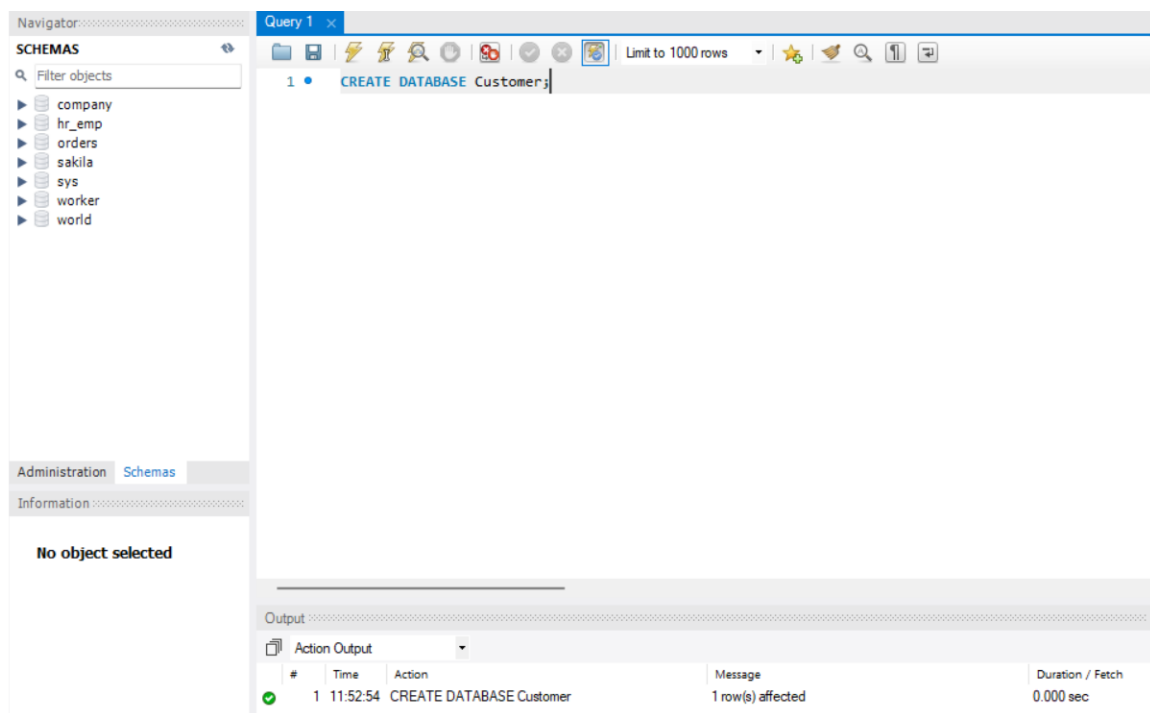
*Term name and year: Summer 2024*

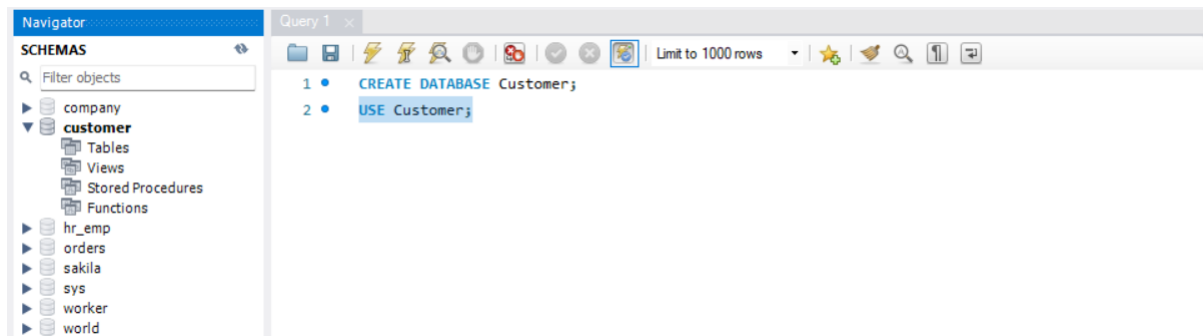
*Submission Week: Week 4 - Assignment 3*

*Instructor's Name: Dr.Nayem Rahman*

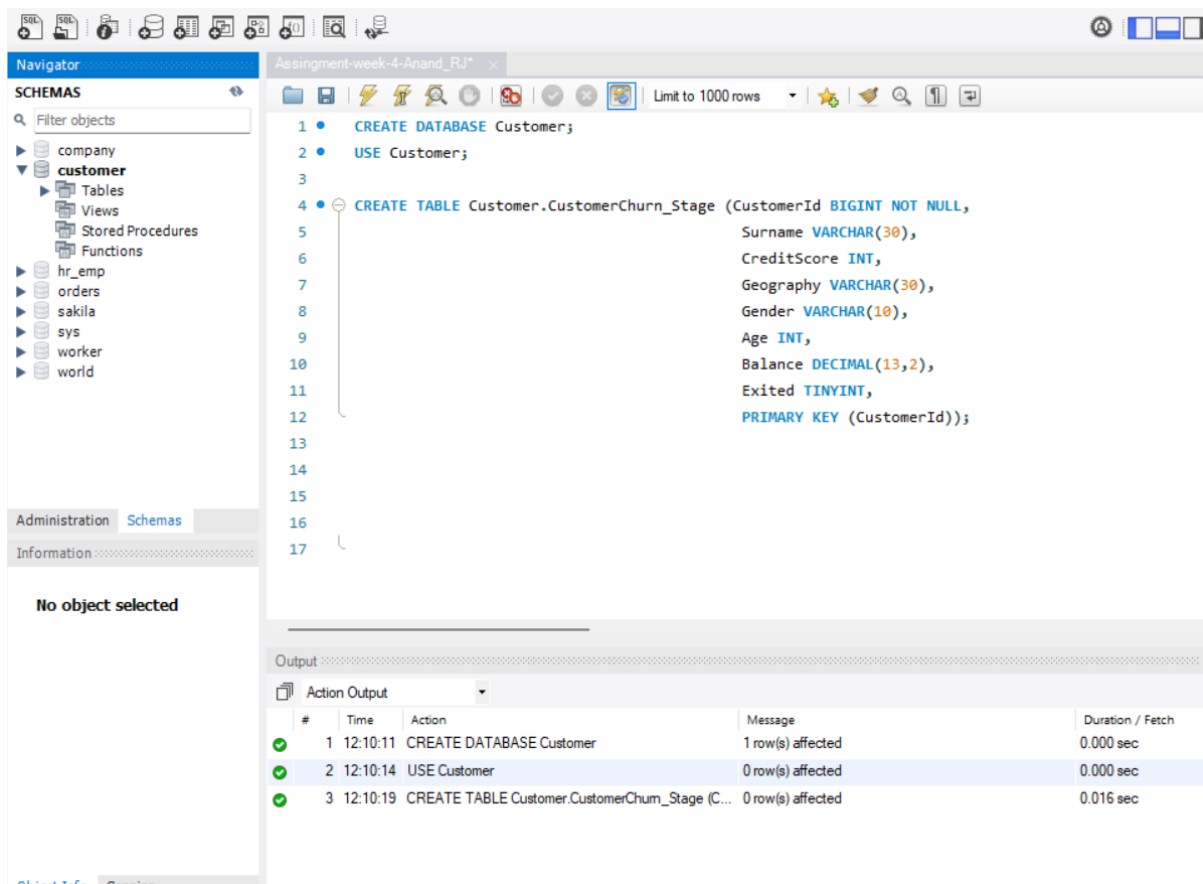
*Date of Submission: August 5, 2024*

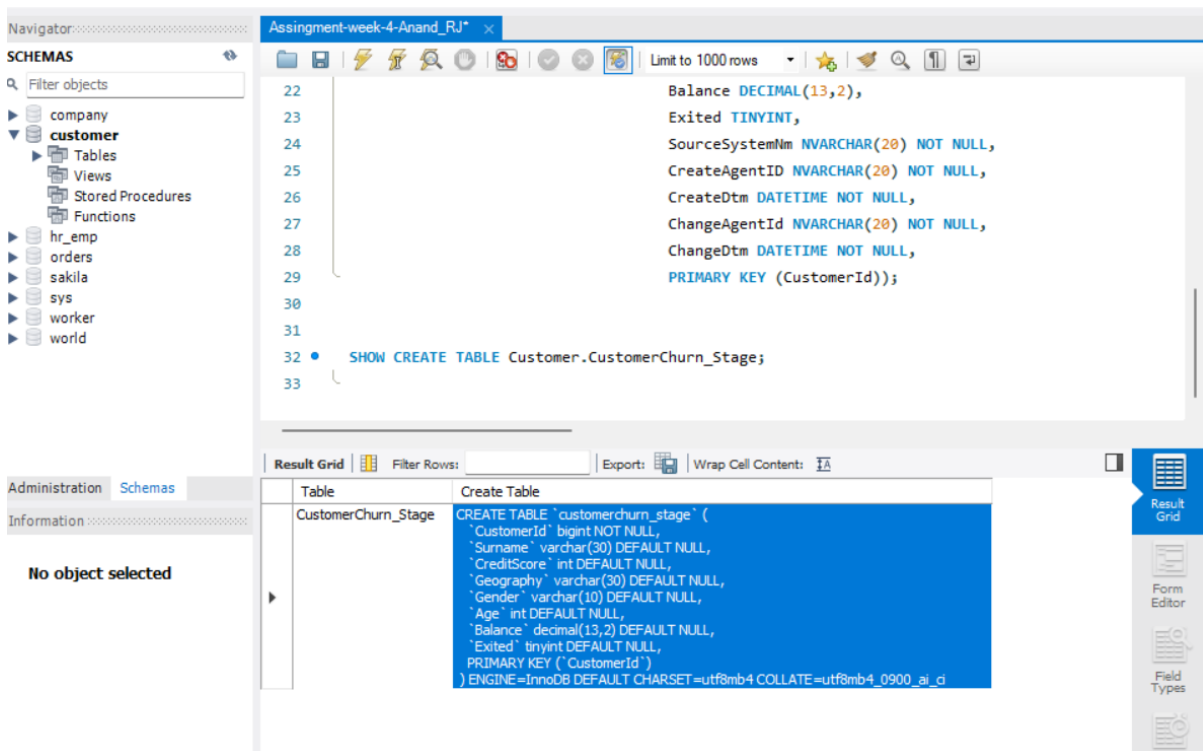
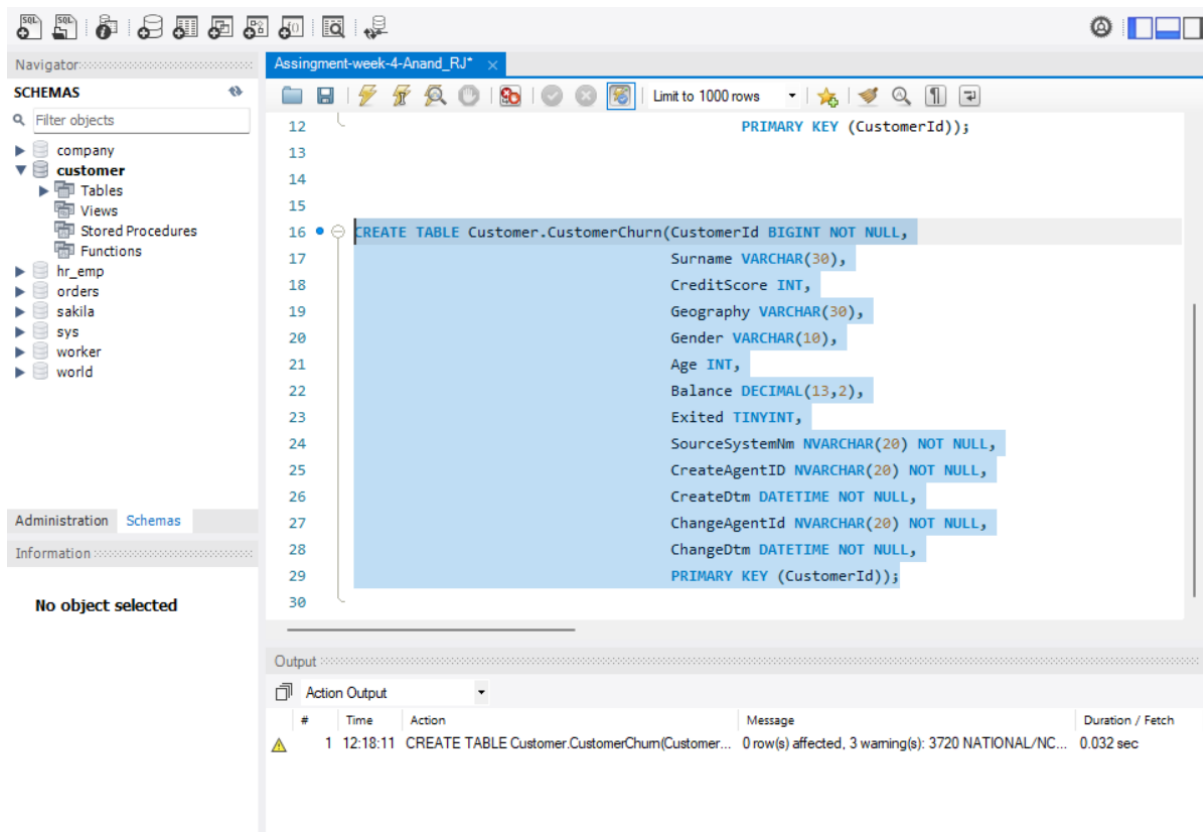
**Q1. { A } Using the MySQL Workbench, create a database called Customer. The database must be named "Customer". { B } Check if the database was created and use the same for further questions**





**Q2. { A } Create a staging table, *\*\* Customer.CustomerChurn\_Stage \*\**, in a database system, with the column list provided in the CSV file. Define the 'CustomerId' as the Primary Key (PK). Get the table definition (DDL) from the database system and capture it in a Word document for submission. { B } Create a persistent table, *\*\* Customer.CustomerChurn \*\**, with the column list provided in the CSV file + following 5 columns : << SourceSystemNm NVARCHAR(20) NOT NULL , CreateAgentId NVARCHAR(20) NOT NULL , CreateDtm DATETIME NOT NULL, ChangeAgentId NVARCHAR(20) NOT NULL , ChangeDtm DATETIME NOT NULL >> Define the 'CustomerId' as the Primary Key (PK). Get the table definition (DDL) from the database system and capture it in a Word document for submission**





**Q3. { A } Load the staging table, \*\* Customer.CustomerChurn\_Stage \*\*, with data from the CSV file, CustomerChurn1.csv . { B } Verify data by comparing the row counts between the CSV file and the staging table, \*\* Customer.CustomerChurn\_Stage [Data Source: CustomerChurn1.CSV] \*\*. Provide the screenshot of last few rows using the ' SELECT \* ' .**

***Make sure the output shows all column values. The SELECT statement must use the ORDER BY 'CustomerId'***

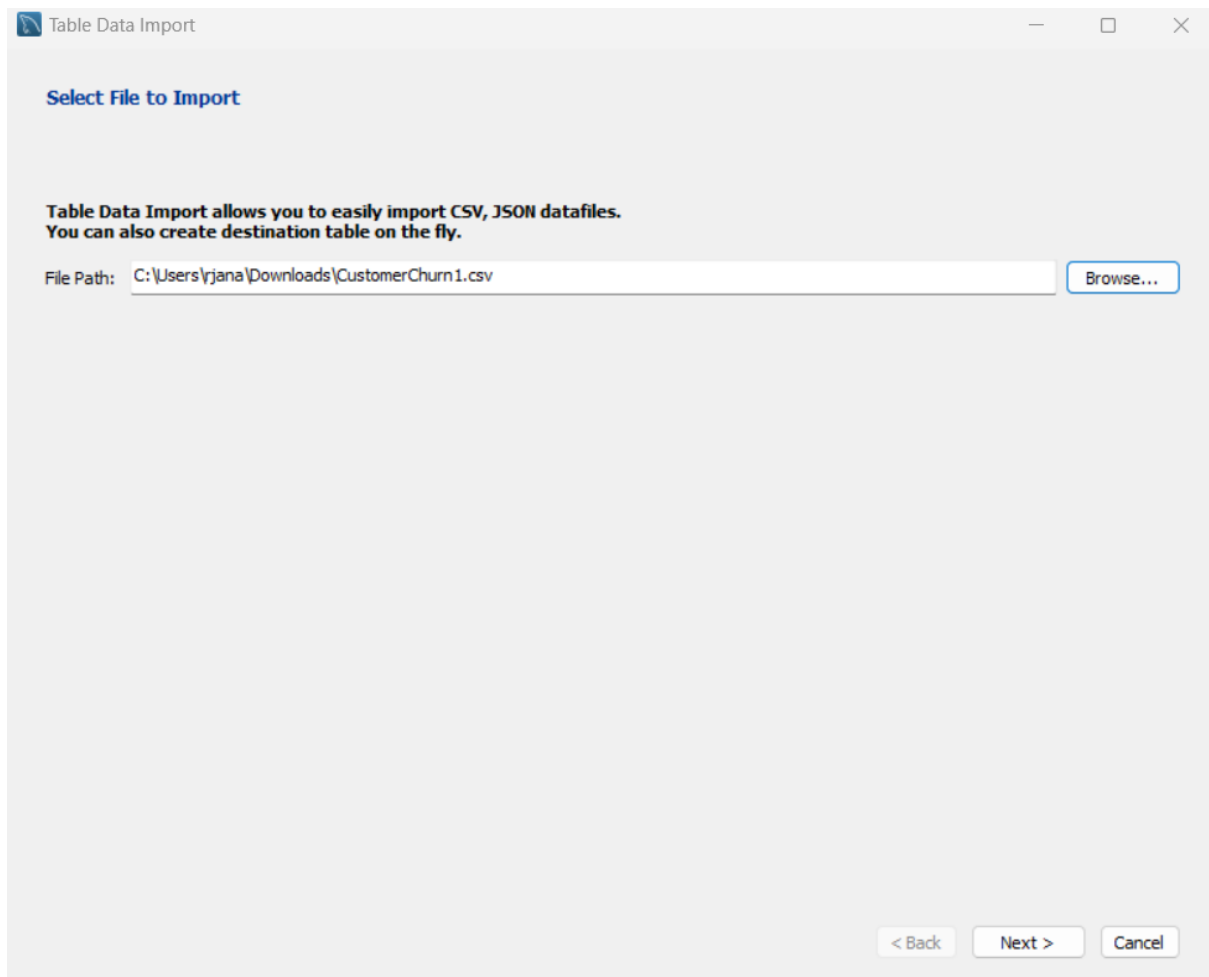


Table Data Import

Select File to Import

Table Data Import allows you to easily import CSV, JSON datafiles.  
You can also create destination table on the fly.

File Path: C:\Users\vjana\Downloads\CustomerChurn1.csv

Browse...

< Back Next > Cancel



**SCHEMAS**

- company
- customer**
  - Tables
    - customerchurn\_stage
      - Columns
      - Indexes
      - Foreign Keys
      - Triggers
    - Views
    - Stored Procedures
    - Functions
  - hr\_emp
  - orders
  - sakila
  - sys
  - worker
  - world

Administration Schemas

Information

Schema: **customer**

**SQL Query Window:**

```

20 Gender VARCHAR(10),
21 Age INT,
22 Balance DECIMAL(13,2),
23 Exited TINYINT,
24 SourceSystemNm NVARCHAR(20) NOT NULL,
25 CreateAgentID NVARCHAR(20) NOT NULL,
26 CreateDtm DATETIME NOT NULL,
27 ChangeAgentId NVARCHAR(20) NOT NULL,
28 ChangeDtm DATETIME NOT NULL,
29 PRIMARY KEY (CustomerId));
30
31
32 • SHOW CREATE TABLE Customer.CustomerChurn_Stage;
33
34 • SHOW CREATE TABLE Customer.CustomerChurn;
35
36 • SELECT COUNT(*) FROM Customer.CustomerChurn_Stage;
37 • SELECT * FROM Customer.CustomerChurn_Stage ORDER BY CustomerId DESC LIMIT 5;
38

```

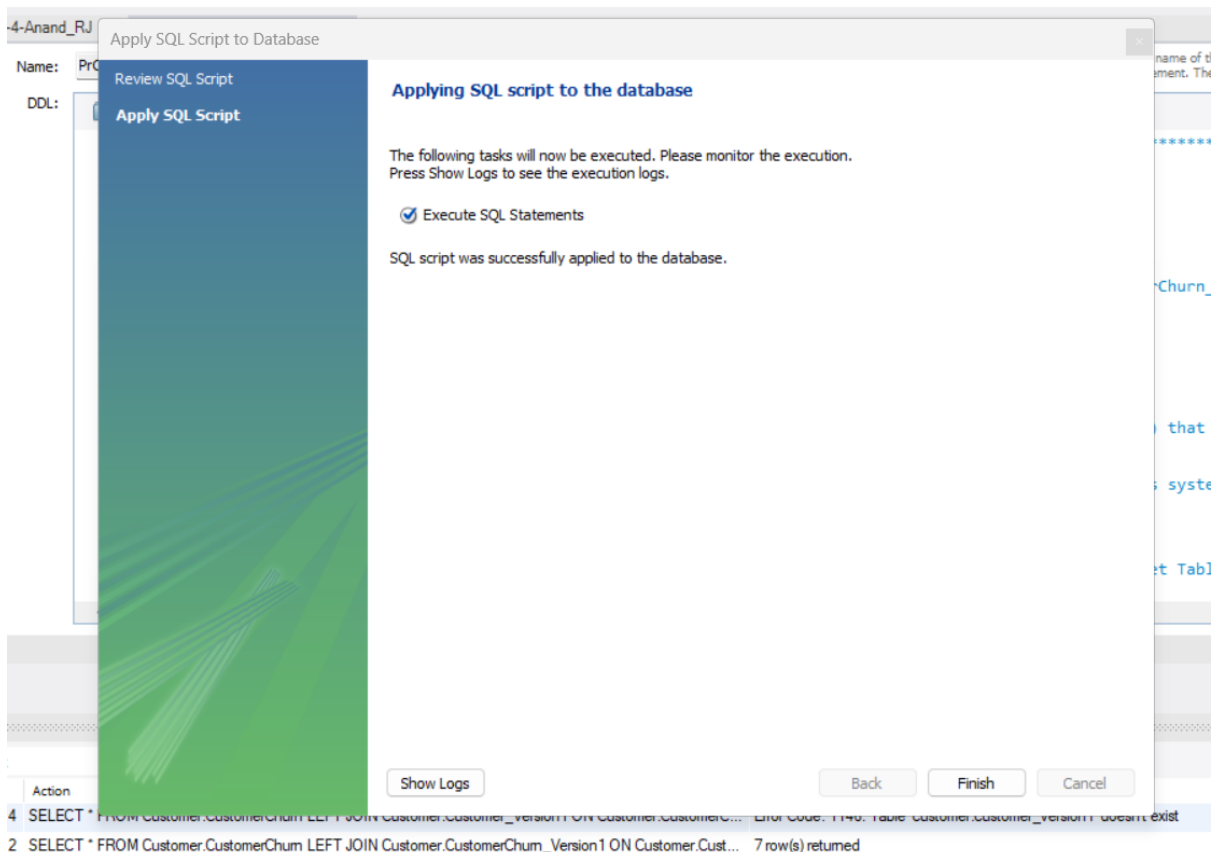
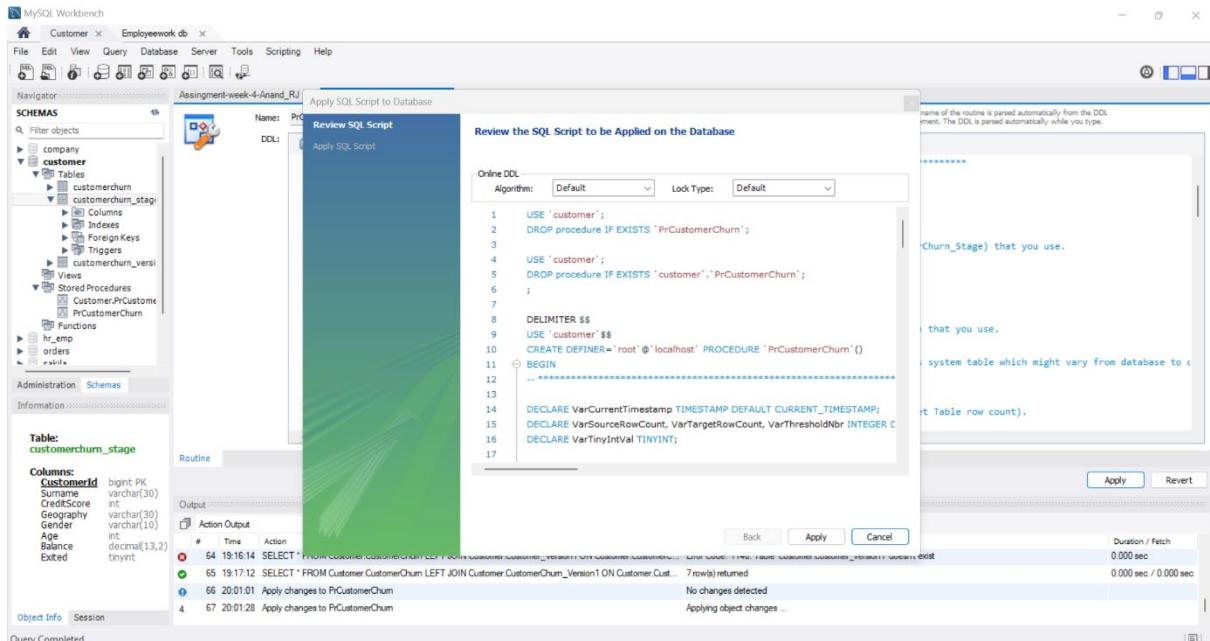
**Result Grid:**

CustomerId	Surname	CreditScore	Geography	Gender	Age	Balance	Exited
15812518	Palermo	657	Spain	Female	37	163607.18	0
15809248	Cole	524	France	Female	36	0.00	0
15805254	Ndukaku	652	Spain	Female	75	0.00	0
15804771	Velazquez	614	France	Male	51	40685.92	0
15803136	Postle	416	Germany	Female	41	122189.66	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

CustomerChurn\_Stage 6 x Apply Revert

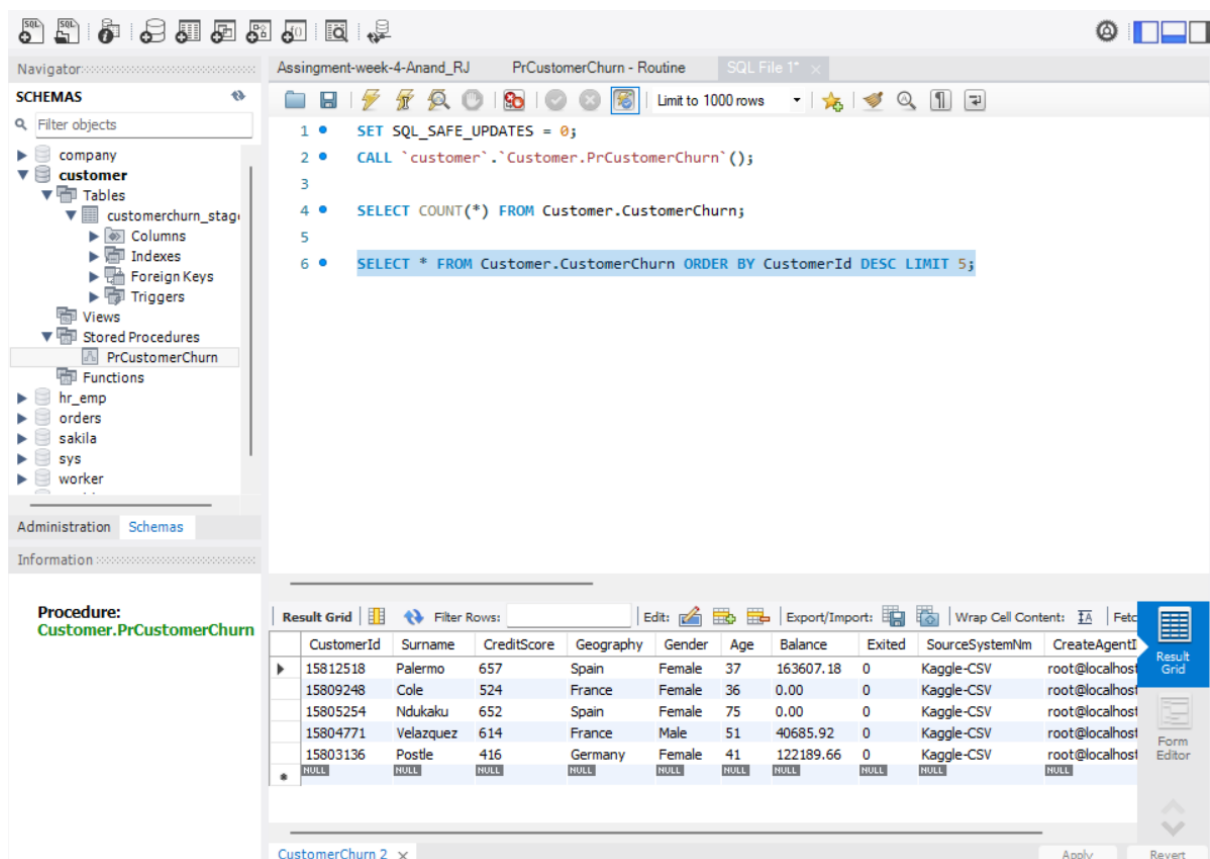
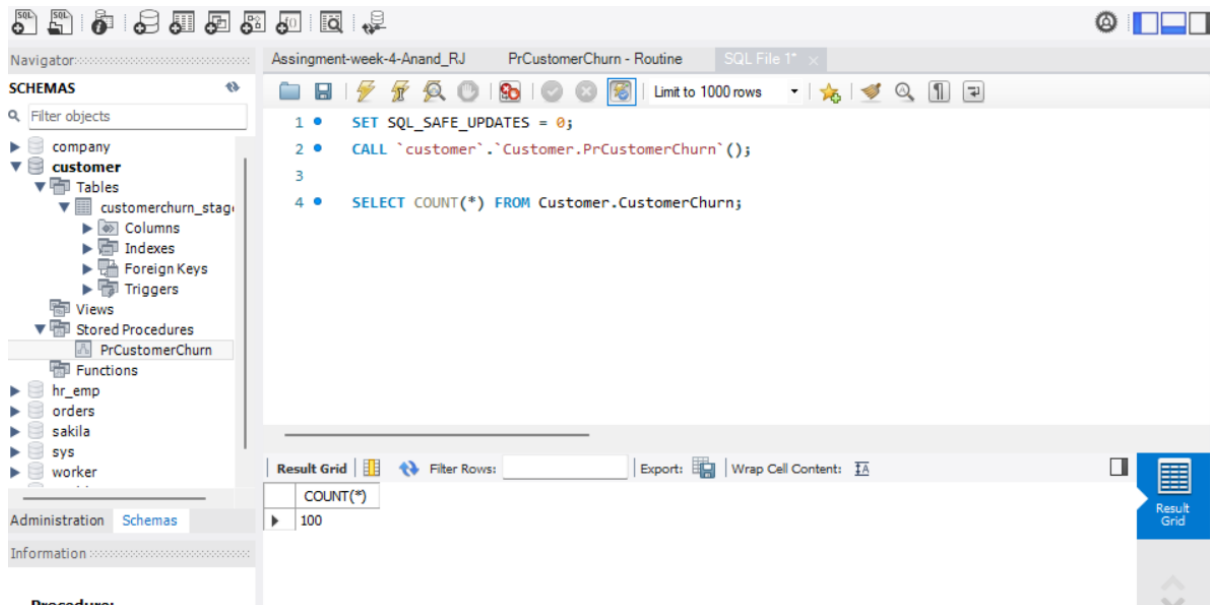
**Q4. Create a database stored procedure based on the template provided along with this assignment << StoredProc\_Template.txt >>. Name the stored procedure name this: **Customer.PrCustomerChurn** \*\*. [[ NOTE : This stored procedure will use the table, **Customer.CustomerChurn\_Stage** \*\*, as the source (aka, staging table). This stored procedure will use the table, **Customer.CustomerChurn** \*\*, as the target (aka, persistent table). ]]**

I have Attached the SQL Document in the Submission



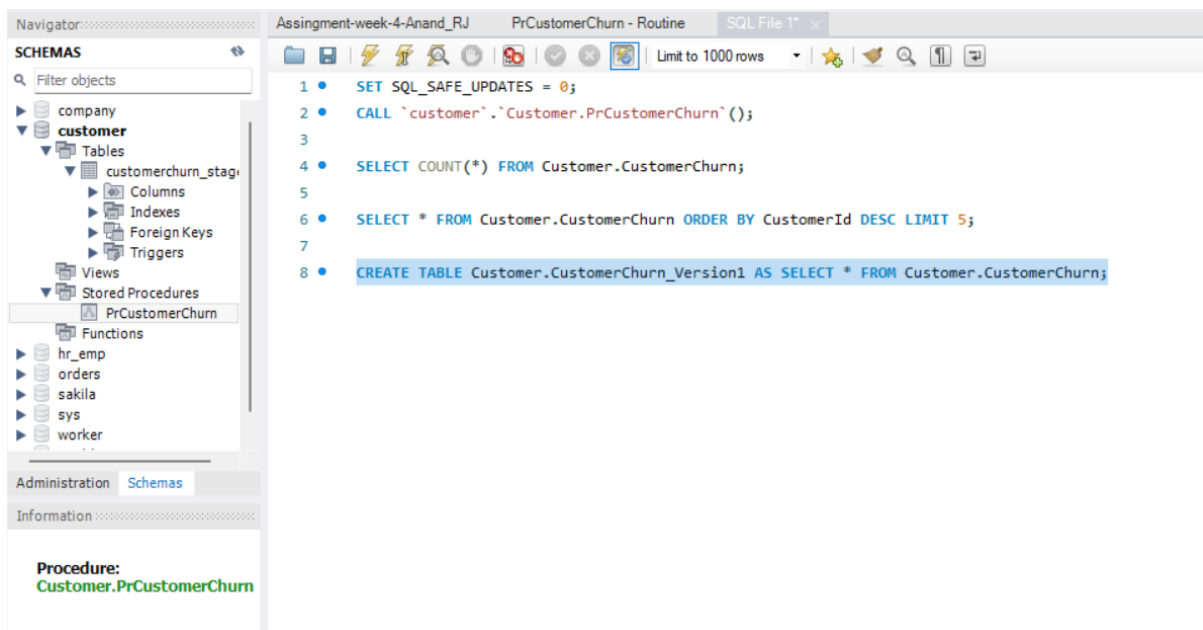
**Q5. Execute the stored procedure, \*\* Customer.PrCustomerChurn \*\*, that was created in Q4. After execution, the stored procedure should load data from the stage to the persistent table: \*\* Customer.CustomerChurn \*\*. {A} Verify data by comparing the row counts between the staging table, \*\* Customer.CustomerChurn\_Stage [Data Source:**

**CustomerChurn1.CSV] \*\* and the persistent table: \*\* Customer.CustomerChurn \*\*. { B }**  
**Provide the screenshot of last few rows using the SELECT \*. Make sure the output shows all column values. The SELECT statement must use the ORDER BY CustomerId.**





Q6. After data verification is completed, in Q5 , { A } create table, \*\* Customer.CustomerChurn\_Version1 \*\*, with data from \*\* Customer.CustomerChurn \*\* (that was already loaded from Customer.CustomerChurn\_Stage via the stored procedure). { B } Show table definition of Customer.CustomerChurn\_Version1 and show the row count of the table, \*\* Customer.CustomerChurn\_Version1 \*\*: { C } Provide the screenshot of last few rows for \*\* Customer.CustomerChurn\_Version1 \*\* [Originally data came from: CustomerChurn1.CSV]. Make sure the output shows all column values. The SELECT statement must use the ORDER BY CustomerId. { D } Empty the staging table, \*\* Customer.CustomerChurn\_Stage \*\*, and load it with data from the CSV file, "CustomerChurn2.csv ". Verify data by comparing the row counts between the CSV file and the staging table, \*\* Customer.CustomerChurn\_Stage \*\* [Data Source: CustomerChurn2.CSV]. Provide the row count of \*\* Customer.CustomerChurn\_Stage \*\* that you loaded from CustomerChurn2.csv file. Provide the screenshot of last few rows using the SELECT \*. Make sure the output shows all column values. The SELECT statement must use the ORDER BY CustomerId.



Navigator: Assingment-week-4-Anand\_RJ PrCustomerChurn - Routine SQL File 1\*

**SCHEMAS**

Filter objects

- company
- customer
  - customerchurn\_stag
    - Columns
    - Indexes
    - Foreign Keys
    - Triggers
  - Views
  - Stored Procedures
    - PrCustomerChurn
  - Functions
- hr\_emp
- orders
- sakila
- sys
- worker

Administration Schemas

Information

**Procedure:**  
Customer.PrCustomerChurn

```

1 • SET SQL_SAFE_UPDATES = 0;
2 • CALL `customer`.`Customer.PrCustomerChurn`();
3
4 • SELECT COUNT(*) FROM Customer.CustomerChurn;
5
6 • SELECT * FROM Customer.CustomerChurn ORDER BY CustomerId DESC LIMIT 5;
7
8 • CREATE TABLE Customer.CustomerChurn_Version1 AS SELECT * FROM Customer.CustomerChurn;
9
10 • SHOW CREATE TABLE Customer.CustomerChurn_Version1;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Table	Create Table
CustomerChurn_Version1	<pre> CREATE TABLE `customerchurn_version1` (   `CustomerId` bigint NOT NULL,   `Surname` varchar(30) DEFAULT NULL,   `CreditScore` int DEFAULT NULL,   `Geography` varchar(30) DEFAULT NULL,   `Gender` varchar(10) DEFAULT NULL,   `Age` int DEFAULT NULL,   `Balance` decimal(13,2) DEFAULT NULL,   `Exited` tinyint DEFAULT NULL,   `SourceSystemNm` varchar(20) CHARACTER SET utf8mb3 COLLATE utf8mb3_general_ci NOT NULL,   `CreateAgentId` varchar(20) CHARACTER SET utf8mb3 COLLATE utf8mb3_general_ci NOT NULL,   `CreateDtm` datetime NOT NULL,   `ChangeAgentId` varchar(20) CHARACTER SET utf8mb3 COLLATE utf8mb3_general_ci NOT NULL,   `ChangeDtm` datetime NOT NULL ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci </pre>

Result 3 x Read Only

Navigator: Assingment-week-4-Anand\_RJ PrCustomerChurn - Routine SQL File 1\*

**SCHEMAS**

Filter objects

- company
- customer
  - customerchurn\_stag
    - Columns
    - Indexes
    - Foreign Keys
    - Triggers
  - Views
  - Stored Procedures
    - PrCustomerChurn
  - Functions
- hr\_emp
- orders
- sakila
- sys
- worker

Administration Schemas

Information

**Procedure:**  
Customer.PrCustomerChurn

```

4 • SELECT COUNT(*) FROM Customer.CustomerChurn;
5
6 • SELECT * FROM Customer.CustomerChurn ORDER BY CustomerId DESC LIMIT 5;
7
8 • CREATE TABLE Customer.CustomerChurn_Version1 AS SELECT * FROM Customer.CustomerChurn;
9
10 • SHOW CREATE TABLE Customer.CustomerChurn_Version1;
11
12 • SELECT COUNT(*) FROM Customer.CustomerChurn_Version1;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

COUNT(*)
100

Result Grid | Form Editor | Field Types

Navigator: Assingment-week-4-Anand\_RJ PrCustomerChurn - Routine SQL File 1\* x

**SCHEMAS**

Filter objects

- company
- customer
  - customerchurn\_stage
    - Columns
    - Indexes
    - Foreign Keys
    - Triggers
  - Views
  - Stored Procedures
    - PrCustomerChurn
  - Functions
- hr\_emp
- orders
- sakila
- sys
- worker

Administration Schemas

Information

**Procedure:**  
Customer.PrCustomerChurn

```

6 • SELECT * FROM Customer.CustomerChurn ORDER BY CustomerId DESC LIMIT 5;
7
8 • CREATE TABLE Customer.CustomerChurn_Version1 AS SELECT * FROM Customer.CustomerChurn;
9
10 • SHOW CREATE TABLE Customer.CustomerChurn_Version1;
11
12 • SELECT COUNT(*) FROM Customer.CustomerChurn_Version1;
13
14 • SELECT * FROM Customer.CustomerChurn_Version1 ORDER BY CustomerId DESC LIMIT 5;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows: |

CustomerId	Surname	CreditScore	Geography	Gender	Age	Balance	Exited	SourceSystemNm	CreateAgentId
15812518	Palermo	657	Spain	Female	37	163607.18	0	Kaggle-CSV	root@localhost
15809248	Cole	524	France	Female	36	0.00	0	Kaggle-CSV	root@localhost
15805254	Ndukaku	652	Spain	Female	75	0.00	0	Kaggle-CSV	root@localhost
15804771	Velazquez	614	France	Male	51	40685.92	0	Kaggle-CSV	root@localhost
15803136	Postle	416	Germany	Female	41	122189.66	0	Kaggle-CSV	root@localhost

Result Grid | Form Editor | Field Types

Navigator: Assingment-week-4-Anand\_RJ PrCustomerChurn - Routine SQL File 1\* x

**SCHEMAS**

Filter objects

- company
- customer
  - customerchurn\_stage
    - Columns
    - Indexes
    - Foreign Keys
    - Triggers
  - Views
  - Stored Procedures
    - PrCustomerChurn
  - Functions
- hr\_emp
- orders
- sakila
- sys
- worker

Administration Schemas

Information

**Procedure:**  
Customer.PrCustomerChurn

```

1 • SET SQL_SAFE_UPDATES = 0;
2 • CALL `customer`.`Customer.PrCustomerChurn`();
3
4 • SELECT COUNT(*) FROM Customer.CustomerChurn;
5
6 • SELECT * FROM Customer.CustomerChurn ORDER BY CustomerId DESC LIMIT 5;
7
8 • CREATE TABLE Customer.CustomerChurn_Version1 AS SELECT * FROM Customer.CustomerChurn;
9
10 • SHOW CREATE TABLE Customer.CustomerChurn_Version1;
11
12 • SELECT COUNT(*) FROM Customer.CustomerChurn_Version1;
13
14 • SELECT * FROM Customer.CustomerChurn_Version1 ORDER BY CustomerId DESC LIMIT 5;
15
16 • TRUNCATE TABLE Customer.CustomerChurn_Stage;

```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
45	18:29:45	CREATE TABLE Customer.CustomerChurn_Versi...	100 row(s) affected Records: 100 Duplicates: 0 ...	0.015 sec
46	18:31:32	SHOW CREATE TABLE Customer.CustomerChur...	1 row(s) returned	0.000 sec / 0.000 sec
47	18:34:07	SELECT COUNT(*) FROM Customer.CustomerCh...	1 row(s) returned	0.000 sec / 0.000 sec
48	18:35:25	SELECT * FROM Customer.CustomerChurn_Versi...	5 row(s) returned	0.000 sec / 0.000 sec
49	18:36:36	TRUNCATE TABLE Customer.CustomerChurn_St...	0 row(s) affected	0.031 sec

Object Info Session

Table Data Import

**Import Results**

File C:\Users\yjana\Downloads\CustomerChurn2.csv was imported in 0.468 s

Table customer.customerchurn\_stage has been used

101 records imported

< Back Finish Cancel

Duration / Fetch  
0.000 sec  
0.000 sec  
0.000 sec  
0.000 sec  
0.000 sec

Assingment-week-4-Anand\_RJ PrCustomerChurn - Routine SQL File 1\*

Limit to 1000 rows

```
12 • SELECT COUNT(*) FROM Customer.CustomerChurn_Version1;
13
14 • SELECT * FROM Customer.CustomerChurn_Version1 ORDER BY CustomerId DESC LIMIT 5;
15
16 • TRUNCATE TABLE Customer.CustomerChurn_Stage;
17
18 • SELECT COUNT(*) FROM Customer.CustomerChurn_Stage;
```

Result Grid

	COUNT(*)
▶	101

Filter Rows: Export: Wrap Cell Content: I

Result Grid  
Form Editor  
Field Types

Table:

Navigator: Schemas

- company
  - customer
    - Tables
      - customerchurn
      - customerchurn\_stagi
      - customerchurn\_versi
    - Columns
    - Indexes
    - Foreign Keys
    - Triggers
    - Views
    - Stored Procedures
      - Customer.PrCustomerChurn
    - Functions
    - hr\_emp
    - orders

Administration Schemas

Information

Table: customerchurn\_stagi

Assingment-week-4-Anand\_RJ PrCustomerChurn - Routine SQL File 1\*

Limit to 1000 rows

```

14 • SELECT * FROM Customer.CustomerChurn_Version1 ORDER BY CustomerId DESC LIMIT 5;
15
16 • TRUNCATE TABLE Customer.CustomerChurn_Stage;
17
18 • SELECT COUNT(*) FROM Customer.CustomerChurn_Stage;
19
20 • SELECT * FROM Customer.CustomerChurn_Stage ORDER BY CustomerId DESC LIMIT 5;

```

Result Grid

CustomerId	Surname	CreditScore	Geography	Gender	Age	Balance	Exited
16812518	Palermo	657	Spain	Female	37	163607.18	0
15809248	Cole	524	France	Female	36	0.00	0
15805254	Ndukaku	652	Spain	Female	75	0.00	0
15804771	Velazquez	614	France	Male	51	40685.92	0
15803136	Postle	416	Germany	Female	41	122189.66	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid

Form Editor

Field Types

**Q7. Execute the stored procedure, Customer.PrCustomerChurn, that was created in Q4. After execution, the stored procedure should load data from the stage to the persistent table: Customer.CustomerChurn. CALL `customer`.`PrCustomerChurn`(); This time, the table will be refreshed via DELETE, UPDATE, and INSERT/SELECT statements in the stored procedure. Show the row count results of both Customer.CustomerChurn\_Version1 table [Data Source: CustomerChurn1.CSV] and the persistent table: Customer.CustomerChurn. Compare the rows between the Customer.CustomerChurn\_Version1 [Data Source: CustomerChurn1.CSV] table and the persistent table: Customer.CustomerChurn [Data Source: CustomerChurn2.CSV]. Show the rows that are available in the Customer.CustomerChurn\_Version1 table but not in the Customer.CustomerChurn table (implementation of brand-new row DELETE statement of the stored procedure).**

Navigator: Schemas

- company
  - customer
    - Tables
      - customerchurn
      - customerchurn\_stagi
      - customerchurn\_versi
    - Columns
    - Indexes
    - Foreign Keys
    - Triggers
    - Views
    - Stored Procedures
      - Customer.PrCustomerChurn
    - Functions
    - hr\_emp
    - orders

Administration Schemas

Information

Table: customerchurn\_stagi

Assingment-week-4-Anand\_RJ PrCustomerChurn - Routine SQL File 1\*

Limit to 1000 rows

```

13
14 • SELECT * FROM Customer.CustomerChurn_Version1 ORDER BY CustomerId DESC LIMIT 5;
15
16 • TRUNCATE TABLE Customer.CustomerChurn_Stage;
17
18 • SELECT COUNT(*) FROM Customer.CustomerChurn_Stage;
19
20 • SELECT * FROM Customer.CustomerChurn_Stage ORDER BY CustomerId DESC LIMIT 5;
21
22 • CALL `customer`.`Customer.PrCustomerChurn`();
23
24 • SELECT COUNT(*) FROM Customer.CustomerChurn_Version1;
25

```

Result Grid

COUNT(*)
100

Result Grid

Navigator: Assingment-week-4-Anand\_RJ PrCustomerChurn - Routine SQL File 1\*

**SCHEMAS**

Filter objects

- company
- customer
  - customerchurn
  - customerchurn\_stage
  - customerchurn\_version1
  - customerchurn\_version2
  - customerchurn\_version3
  - customerchurn\_version4
  - customerchurn\_version5
  - customerchurn\_version6
  - customerchurn\_version7
  - customerchurn\_version8
  - customerchurn\_version9
  - customerchurn\_version10
  - customerchurn\_version11
  - customerchurn\_version12
  - customerchurn\_version13
  - customerchurn\_version14
  - customerchurn\_version15
  - customerchurn\_version16
  - customerchurn\_version17
  - customerchurn\_version18
  - customerchurn\_version19
  - customerchurn\_version20
  - customerchurn\_version21
  - customerchurn\_version22
  - customerchurn\_version23
  - customerchurn\_version24
  - customerchurn\_version25
  - customerchurn\_version26
  - customerchurn\_version27
  - customerchurn\_version28
  - customerchurn\_version29
  - customerchurn\_version30
  - customerchurn\_version31
  - customerchurn\_version32
  - customerchurn\_version33
  - customerchurn\_version34
  - customerchurn\_version35
  - customerchurn\_version36
  - customerchurn\_version37
  - customerchurn\_version38
  - customerchurn\_version39
  - customerchurn\_version40
  - customerchurn\_version41
  - customerchurn\_version42
  - customerchurn\_version43
  - customerchurn\_version44
  - customerchurn\_version45
  - customerchurn\_version46
  - customerchurn\_version47
  - customerchurn\_version48
  - customerchurn\_version49
  - customerchurn\_version50
  - customerchurn\_version51
  - customerchurn\_version52
  - customerchurn\_version53
  - customerchurn\_version54
  - customerchurn\_version55
  - customerchurn\_version56
  - customerchurn\_version57
  - customerchurn\_version58
  - customerchurn\_version59
  - customerchurn\_version60
  - customerchurn\_version61
  - customerchurn\_version62
  - customerchurn\_version63
  - customerchurn\_version64
  - customerchurn\_version65
  - customerchurn\_version66
  - customerchurn\_version67
  - customerchurn\_version68
  - customerchurn\_version69
  - customerchurn\_version70
  - customerchurn\_version71
  - customerchurn\_version72
  - customerchurn\_version73
  - customerchurn\_version74
  - customerchurn\_version75
  - customerchurn\_version76
  - customerchurn\_version77
  - customerchurn\_version78
  - customerchurn\_version79
  - customerchurn\_version80
  - customerchurn\_version81
  - customerchurn\_version82
  - customerchurn\_version83
  - customerchurn\_version84
  - customerchurn\_version85
  - customerchurn\_version86
  - customerchurn\_version87
  - customerchurn\_version88
  - customerchurn\_version89
  - customerchurn\_version90
  - customerchurn\_version91
  - customerchurn\_version92
  - customerchurn\_version93
  - customerchurn\_version94
  - customerchurn\_version95
  - customerchurn\_version96
  - customerchurn\_version97
  - customerchurn\_version98
  - customerchurn\_version99
  - customerchurn\_version100
- hr\_emp
- orders

Administration Schemas

Information

Table: customerchurn\_stage

Columns: CustomerId, CreditScore, Geography, Gender, Age, Balance, Exited, SourceSystemNm, CreateAgentId, CreateDtm, ChangeAgentId, ChangeDtm

Result Grid

COUNT(*)
101

Result 9 x

Read Only

Navigator: Assingment-week-4-Anand\_RJ PrCustomerChurn - Routine SQL File 1\*

**SCHEMAS**

Filter objects

- company
- customer
  - customerchurn
  - customerchurn\_stage
  - customerchurn\_version1
  - customerchurn\_version2
  - customerchurn\_version3
  - customerchurn\_version4
  - customerchurn\_version5
  - customerchurn\_version6
  - customerchurn\_version7
  - customerchurn\_version8
  - customerchurn\_version9
  - customerchurn\_version10
  - customerchurn\_version11
  - customerchurn\_version12
  - customerchurn\_version13
  - customerchurn\_version14
  - customerchurn\_version15
  - customerchurn\_version16
  - customerchurn\_version17
  - customerchurn\_version18
  - customerchurn\_version19
  - customerchurn\_version20
  - customerchurn\_version21
  - customerchurn\_version22
  - customerchurn\_version23
  - customerchurn\_version24
  - customerchurn\_version25
  - customerchurn\_version26
  - customerchurn\_version27
  - customerchurn\_version28
  - customerchurn\_version29
  - customerchurn\_version30
  - customerchurn\_version31
  - customerchurn\_version32
  - customerchurn\_version33
  - customerchurn\_version34
  - customerchurn\_version35
  - customerchurn\_version36
  - customerchurn\_version37
  - customerchurn\_version38
  - customerchurn\_version39
  - customerchurn\_version40
  - customerchurn\_version41
  - customerchurn\_version42
  - customerchurn\_version43
  - customerchurn\_version44
  - customerchurn\_version45
  - customerchurn\_version46
  - customerchurn\_version47
  - customerchurn\_version48
  - customerchurn\_version49
  - customerchurn\_version50
  - customerchurn\_version51
  - customerchurn\_version52
  - customerchurn\_version53
  - customerchurn\_version54
  - customerchurn\_version55
  - customerchurn\_version56
  - customerchurn\_version57
  - customerchurn\_version58
  - customerchurn\_version59
  - customerchurn\_version60
  - customerchurn\_version61
  - customerchurn\_version62
  - customerchurn\_version63
  - customerchurn\_version64
  - customerchurn\_version65
  - customerchurn\_version66
  - customerchurn\_version67
  - customerchurn\_version68
  - customerchurn\_version69
  - customerchurn\_version70
  - customerchurn\_version71
  - customerchurn\_version72
  - customerchurn\_version73
  - customerchurn\_version74
  - customerchurn\_version75
  - customerchurn\_version76
  - customerchurn\_version77
  - customerchurn\_version78
  - customerchurn\_version79
  - customerchurn\_version80
  - customerchurn\_version81
  - customerchurn\_version82
  - customerchurn\_version83
  - customerchurn\_version84
  - customerchurn\_version85
  - customerchurn\_version86
  - customerchurn\_version87
  - customerchurn\_version88
  - customerchurn\_version89
  - customerchurn\_version90
  - customerchurn\_version91
  - customerchurn\_version92
  - customerchurn\_version93
  - customerchurn\_version94
  - customerchurn\_version95
  - customerchurn\_version96
  - customerchurn\_version97
  - customerchurn\_version98
  - customerchurn\_version99
  - customerchurn\_version100
- hr\_emp
- orders

Administration Schemas

Information

Table: customerchurn\_stage

Columns: CustomerId, CreditScore, Geography, Gender, Age, Balance, Exited, SourceSystemNm, CreateAgentId, CreateDtm, ChangeAgentId, ChangeDtm

Result Grid

CustomerId	Surname	CreditScore	Geography	Gender	Age	Balance	Exited	SourceSystemNm	CreateAgentId
15604348	Allard	710	Spain	Male	22	0.00	0	Kaggle-CSV	root@localhos
15687946	Osborne	556	France	Female	61	117419.35	0	Kaggle-CSV	root@localhos
15701164	Onyeorulu	506	France	Female	34	90307.62	0	Kaggle-CSV	root@localhos
15725737	Mosman	669	France	Male	46	0.00	0	Kaggle-CSV	root@localhos
15755648	Pisano	675	France	Female	21	98373.26	0	Kaggle-CSV	root@localhos
15812518	Palermo	657	Spain	Female	37	163607.18	0	Kaggle-CSV	root@localhos

**Q8. Show the rows (SELECT \*) that changed (one or many non-Primary Key columns), in the Customer.CustomerChurn table (implementation of UPDATE statement of the stored procedure). You need to perform a comparison between Customer.CustomerChurn table [Data Source: CustomerChurn2.CSV] and Customer.CustomerChurn\_Version1 table [Data Source: CustomerChurn1.CSV] in terms of non-PK columns (Excludes: SourceSystemNm, CreateAgentId, CreateDtm, ChangeAgentId, ChangeDtm), and with a join condition using the PK column(s). You must do ORDER BY CustomerId. The output of this query should show different values for the CreateDtm and ChangeDtm columns in Customer.CustomerChurn table for the changed rows. Take a screenshot and capture it in the Word document. Make sure all columns including CreateDtm and ChangeDtm of CustomerChurn table are displayed.**



Navigator

Assingment-week-4-Anand\_RJ PrCustomerChurn - Routine SQL-Call\_Function\_Stored\_Proc

Limit to 1000 rows

```

19
20 • SELECT * FROM Customer.CustomerChurn_Stage ORDER BY CustomerId DESC LIMIT 5;
21
22 • CALL 'customer'.PrCustomerChurn();
23
24 • SELECT COUNT(*) FROM Customer.CustomerChurn_Version1;
25
26 • SELECT COUNT(*) FROM Customer.CustomerChurn;
27
28 • SELECT * FROM Customer.CustomerChurn_Version1
29 LEFT JOIN Customer.CustomerChurn
30 ON Customer.CustomerChurn_Version1.CustomerId = Customer.CustomerChurn.CustomerId
31 Where Customer.CustomerChurn.CustomerId IS NULL;
32
33 • SELECT a.CustomerId, a.*, b.CreateDtm AS V1_CreateDtm, b.ChangedDtm AS V1_ChangedDtm
34 FROM Customer.CustomerChurn a
35 JOIN Customer.CustomerChurn_Version1 b

```

Table: customerchurn\_stage

Columns: CustomerId bigint PK, Surname varchar(30), CreditScore int, Geography varchar(30), Age int, Balance decimal(13,2), Exited tinyint

CustomerId	Surname	CreditScore	Geography	Gender	Age	Balance	Exited	SourceSystemIn	CreateAgentID	CreateDtm	ChangeAgentID	ChangeDtm	V1_CreateDtm	V1_ChangedDtm
15592389	Vivek	684	India	Male	27	134603.88	0	Kaggle-CSV	root@localhost	2024-08-10 14:44:32	root@localhost	2024-08-10 18:52:50	2024-08-10 14:44:32	2024-08-10 14:44:32
15597945	Delucci	700	Spain	Female	32	654.00	0	Kaggle-CSV	root@localhost	2024-08-10 14:44:32	root@localhost	2024-08-10 18:52:50	2024-08-10 14:44:32	2024-08-10 14:44:32
15683553	Osman	788	USA	Male	22	75888.30	0	Kaggle-CSV	root@localhost	2024-08-10 14:44:32	root@localhost	2024-08-10 18:52:50	2024-08-10 14:44:32	2024-08-10 14:44:32
15691483	Chen	549	France	Female	25	12345.50	0	Kaggle-CSV	root@localhost	2024-08-10 14:44:32	root@localhost	2024-08-10 18:52:50	2024-08-10 14:44:32	2024-08-10 14:44:32
15732173	Andrews	497	Spain	Male	30	0.00	1	Kaggle-CSV	root@localhost	2024-08-10 14:44:32	root@localhost	2024-08-10 18:52:50	2024-08-10 14:44:32	2024-08-10 14:44:32
15738751	Belt	493	France	Female	46	321.00	0	Kaggle-CSV	root@localhost	2024-08-10 14:44:32	root@localhost	2024-08-10 18:52:50	2024-08-10 14:44:32	2024-08-10 14:44:32
15767821	Sharon	528	Hong Kong	Male	31	102016.72	0	Kaggle-CSV	root@localhost	2024-08-10 14:44:32	root@localhost	2024-08-10 18:52:50	2024-08-10 14:44:32	2024-08-10 14:44:32

Result 11

**Q9. Provide the screenshot of last few rows using the SELECT \* FROM Customer.CustomerChurn. Make sure the output shows all column values. The SELECT statement must use the ORDER BY CustomerId. Show the rows that are available in the Customer.CustomerChurn table [Data Source: CustomerChurn2.CSV] but not in the Customer.CustomerChurn\_Version1 table (implementation of brand-new rows INSERT by the stored procedure). Do a SELECT \* along with ORDER BY CustomerId. Take a screenshot and capture it in the Word document.**

Navigator

Assingment-week-4-Anand\_RJ PrCustomerChurn - Routine SQL-Call\_Function\_Stored\_Proc

Limit to 1000 rows

```

37 WHERE a.Surname <> b.Surname
38 OR a.Creditscore <> b.Creditscore
39 OR a.Geography <> b.Geography
40 OR a.Gender <> b.Gender
41 OR a.Age <> b.Age
42 OR a.Balance <> b.Balance
43 OR a.Exited <> b.Exited
44 ORDER BY a.CustomerId;
45
46 • SELECT * FROM Customer.CustomerChurn
47 LEFT JOIN Customer.CustomerChurn_Version1
48 ON Customer.CustomerChurn.CustomerId = Customer.CustomerChurn_Version1.CustomerId
49 Where Customer.CustomerChurn_Version1.CustomerId IS NULL
50 ORDER BY Customer.CustomerChurn.CustomerId;

```

Table: customerchurn\_stage

Columns: CustomerId bigint PK, Surname varchar(30), CreditScore int, Geography varchar(30), Age int, Balance decimal(13,2), Exited tinyint

CustomerId	Surname	CreditScore	Geography	Gender	Age	Balance	Exited	SourceSystemIn	CreateAgentID	CreateDtm	ChangeAgentID	ChangeDtm	CustomerId	Surname	CreditScore	Geography	Gender	Age	Balance	Exited
15589975	Maclean	646	France	Female	73	97259.25	0	Kaggle-CSV	root@localhost	2024-08-10 18:52:50	root@localhost	2024-08-10 18:52:50	15589975	Maclean	646	France	Female	73	97259.25	0
15657566	Wick	634	Germany	Male	24	103097.85	0	Kaggle-CSV	root@localhost	2024-08-10 18:52:50	root@localhost	2024-08-10 18:52:50	15657566	Wick	634	Germany	Male	24	103097.85	0
15698932	Groves	756	Germany	Male	44	137452.09	0	Kaggle-CSV	root@localhost	2024-08-10 18:52:50	root@localhost	2024-08-10 18:52:50	15698932	Groves	756	Germany	Male	44	137452.09	0
15726676	Marshall	616	Spain	Male	30	0.00	0	Kaggle-CSV	root@localhost	2024-08-10 18:52:50	root@localhost	2024-08-10 18:52:50	15726676	Marshall	616	Spain	Male	30	0.00	0
15727356	O'Donnell	744	Spain	Female	26	166297.89	0	Kaggle-CSV	root@localhost	2024-08-10 18:52:50	root@localhost	2024-08-10 18:52:50	15727356	O'Donnell	744	Spain	Female	26	166297.89	0
15771977	Tao	730	France	Female	39	99010.67	0	Kaggle-CSV	root@localhost	2024-08-10 18:52:50	root@localhost	2024-08-10 18:52:50	15771977	Tao	730	France	Female	39	99010.67	0
16812518	Palermo	657	Spain	Female	37	163607.18	0	Kaggle-CSV	root@localhost	2024-08-10 18:52:50	root@localhost	2024-08-10 18:52:50	16812518	Palermo	657	Spain	Female	37	163607.18	0

Result 12

**Q10. Show the final version of the stored procedure code that was used to load the persistent table, Customer.CustomerChurn. Submit it as a \*.TXT file. Submit your work (QUESTIONS 1-9) in a single Word Document/PDF file.**

Attached the SQL-Call\_Function\_Stored\_Proc.txt file