Student's Full Name: Surajit Pal
Course Title: Data Warehousing and Analytics in the Cloud
Term name and year: Spring 2023
Submission Week: Week 4 - Assignment 3
Instructor's Name: Dr.Nayem Rahman
Date of Submission: 22nd Nov 2023

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.

Query 1

Limit to 1000 rows

```
1 •  CREATE DATABASE Customer;
2
3 •  SHOW DATABASES
```

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

Database
customer
information_schema
mysql
performance_schema
sakila
sys
worker
world

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.

Query 1

Limit to 1000 rows

```sql
Use Customer;

drop table if exists CustomerChurn_Stage;

CREATE TABLE CustomerChurn_Stage (
    CustomerID integer not Null,
    Surname char(30) not null,
    CreditScore integer not null,
    Geography char(50) not null,
    Gender char(6) not null,
    Age integer not null,
    Balance decimal(15,2) not null,
    Exited boolean,

    PRIMARY KEY (CustomerID)
);

Desc CustomerChurn_Stage;
```

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

Field	Type	Null	Key	Default	Extra
CustomerID	int	NO	PRI	NULL	
Surname	char(30)	NO		NULL	
CreditScore	int	NO		NULL	
Geography	char(50)	NO		NULL	
Gender	char(6)	NO		NULL	
Age	int	NO		NULL	
Balance	decimal(15,2)	NO		NULL	
Exited	tinyint(1)	YES		NULL	

Query 1

Limit to 1000 rows

```sql
drop table if exists CustomerChurn;
CREATE TABLE CustomerChurn (
    CustomerID integer not Null,
    Surname char(30) not null,
    CreditScore integer not null,
    Geography char(50) not null,
    Gender char(6) not null,
    Age integer not null,
    Balance decimal(15,2) not null,
    Exited boolean,
    SourceSystemNm nvarchar(20) not null,
    CreateAgentId nvarchar(20) not null,
    CreateDtm DateTime  not null,
    ChangeAgentId nvarchar(20) not null,
    ChangeDtm Datetime not null,

    PRIMARY KEY (CustomerID)
);

Desc CustomerChurn;
```

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

Field	Type	Null	Key	Default	Extra
CustomerID	int	NO	PRI	NULL	
Surname	char(30)	NO		NULL	
CreditScore	int	NO		NULL	
Geography	char(50)	NO		NULL	
Gender	char(6)	NO		NULL	
Age	int	NO		NULL	
Balance	decimal(15,2)	NO		NULL	

Result 1

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 '.

Query 1 | Administration - Server Status

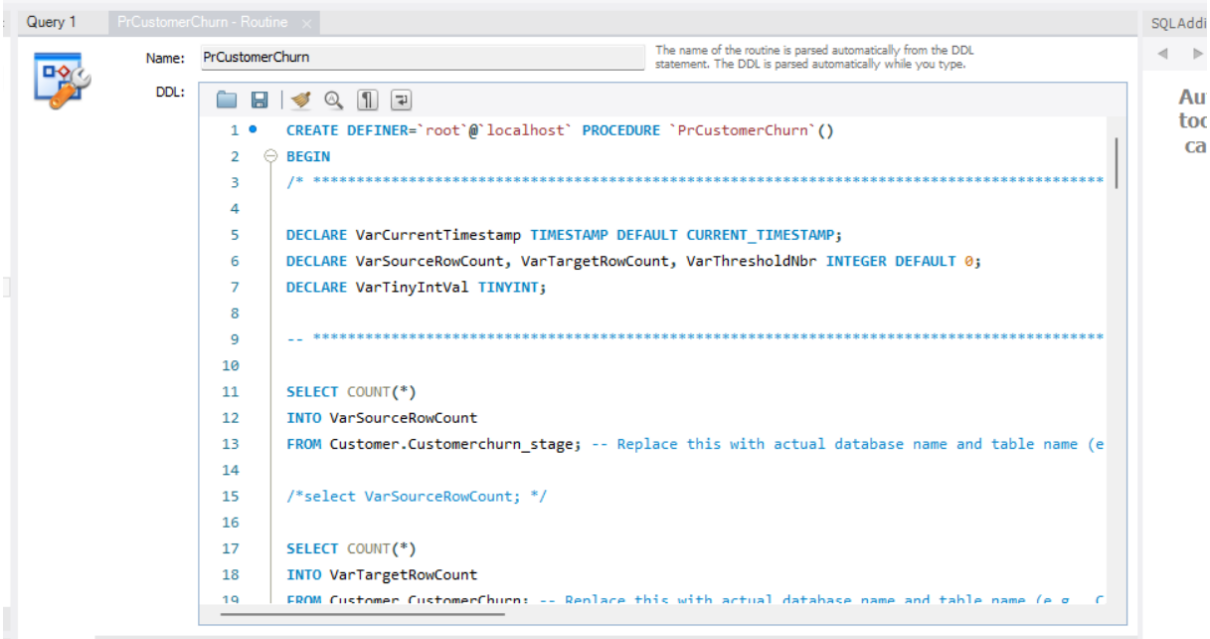
Limit to 1000 rows

```
1    Select * from customerchurn_stage order by CustomerID;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content

CustomerID	Surname	CreditScore	Geography	Gender	Age	Balance	Exited
15568982	Hao	726	France	Female	24	0.00	0
15569590	Yoo	601	Germany	Male	42	98495.72	1
15574012	Chu	645	Spain	Male	44	113755.78	1
15575185	Bushell	757	Spain	Male	33	77253.22	0
15577657	McDonald	732	France	Male	41	0.00	0
15585768	Cameron	582	Germany	Male	41	70349.48	0
15589475	Azikiwe	591	Spain	Female	39	0.00	1
15592389	H?	684	France	Male	27	134603.88	0
15592461	Jackson	603	Germany	Male	26	109166.37	0
15592531	Bartlett	822	France	Male	50	0.00	0
15597945	Dellucci	636	Spain	Female	32	0.00	0
15600882	Scott	635	Spain	Female	35	0.00	0
15602280	Martin	829	Germany	Female	27	112045.67	1
15604348	Allard	710	Spain	Male	22	0.00	0
15614049	Hu	664	France	Male	55	0.00	0
15616550	Chidiebele	698	Germany	Male	44	116363.37	0

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).]]



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.

Q_02_B_Create_Table_Custom... Q_06_A_Create_

```
use customer;
call PrCustomerChurn();
```

Q_02_B_Create_Table_Custom... SQL File 4* Q_06_A_Create_Table_Custom... SQL File 5 Q_06_D_Select_Table_(

Limit to 2000 rows

```
Use Customer;

Select count(*) CustomerChurn_RecordCount from CustomerChurn;

Select count(*) CustomerChurn_Stage_RecordCount from CustomerChurn_Stage;

Select * from CustomerChurn
Order by customerId;

Select * from CustomerChurn_Stage
Order by customerId;
```

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

CustomerChurn_RecordCount
100

Q_02_B_Create_Table_Custom... SQL File 4* Q_06_A_Create_Table_Custom... SQL File 5 Q_06

Limit to 2000 rows

```sql
Use Customer;

Select count(*) CustomerChurn_RecordCount from CustomerChurn;

Select count(*) CustomerChurn_Stage_RecordCount from CustomerChurn_Stage;

Select * from CustomerChurn
Order by customerId;

Select * from CustomerChurn_Stage
Order by customerId;
```

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

CustomerChurn_Stage_RecordCount
100

SQL File 2* Q_05_A_Call_StoredProcedure Q_05_B_Select_Table_Custom...

Limit to 1000 rows

```sql
Use Customer;

Select count(*) CustomerChurn_RecordCount from CustomerChurn;

Select count(*) CustomerChurn_Stage_RecordCount from CustomerChurn_Stage;

Select * from CustomerChurn
Order by customerId;

Select * from CustomerChurn_Stage
Order by customerId;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

CustomerID	Surname	CreditScore	Geography	Gender	Age	Balance	Exited	SourceSystemNm	CreateAgentId	CreateDtm	ChangeAgentId	ChangeDtm
15771573	Okagbue	637	Germany	Female	39	137843.80	1	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:06:31
15771873	Buccho	776	Germany	Female	37	103769.22	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:06:31
15773469	Clark	687	Germany	Female	27	152328.88	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:06:31
15779052	Ballard	604	Germany	Female	25	157780.84	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:06:31
15780961	Cavenagh	735	France	Female	21	178718.19	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:06:31
15788218	Henderson	549	Spain	Female	24	0.00	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:06:31
15788448	Watson	490	Spain	Male	31	145260.23	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:06:31
15789484	Hammond	751	Germany	Female	36	169831.46	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:06:31
15792365	He	501	France	Male	44	142051.07	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:06:31
15794171	Lombardo	475	France	Female	45	134264.04	1	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:06:31
15803136	Postle	416	Germany	Female	41	122189.66	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:06:31
15804771	Velazquez	614	France	Male	51	40685.92	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:06:31
15805254	Ndukaku	652	Spain	Female	75	0.00	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:06:31
15809248	Cole	524	France	Female	36	0.00	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:06:31
15812518	Palermo	657	Spain	Female	37	163607.18	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:06:31
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Q_02_B_Create_Table_Custom... Q_06_A_Create_Table_Custom... SQL File 5 Q_06_D_Select_Table_Custom... Q_07_A_Select_Table_Record...

Limit to 2000 rows

```sql
Use Customer;

Select count(*) CustomerChurn_RecordCount from CustomerChurn;

Select count(*) CustomerChurn_Stage_RecordCount from CustomerChurn_Stage;

Select * from CustomerChurn
Order by customerId;

Select * from CustomerChurn_Stage
Order by customerId;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

CustomerID	Surname	CreditScore	Geography	Gender	Age	Balance	Exited
15736816	Young	756	Germany	Male	36	136815.64	0
15737173	Andrews	497	Spain	Male	24	0.00	0
15737452	Romeo	653	Germany	Male	58	132602.88	1
15737888	Mitchell	850	Spain	Female	43	125510.82	0
15738148	Clarke	465	France	Female	51	122522.32	1
15738191	Maclean	577	France	Male	25	0.00	0
15738721	Graham	773	Spain	Male	41	102827.44	0
15738751	Beit	493	France	Female	46	0.00	0
15750181	Sanderson	553	Germany	Male	41	110112.54	0
15751208	Pirozzi	684	Spain	Male	56	78707.16	0
15754849	Tyler	776	Germany	Female	32	109421.13	0
15755196	Lavine	834	France	Female	49	131394.56	1
15755648	Pisano	675	France	Female	21	98373.26	0
15757535	Heap	647	Spain	Female	44	0.00	1
15760861	Phillipps	619	France	Male	43	125211.92	0
15762418	Gant	750	Spain	Male	22	121681.82	1
15766205	Yin	550	Germany	Male	38	103391.38	0
15767821	Bearce	528	France	Male	31	102016.72	0
15767954	Osborne	635	Germany	Female	28	81623.67	0
15768193	Trevisani	585	Germany	Male	36	146050.97	0
15770811	Wallace	519	France	Male	36	0.00	0
15771573	Okagbue	637	Germany	Female	39	137843.80	1
15771873	Buccho	776	Germany	Female	37	103769.22	0
15773469	Clark	687	Germany	Female	27	152328.88	0
15779052	Ballard	604	Germany	Female	25	157780.84	0
15780961	Cavenagh	735	France	Female	21	178718.19	0
15788218	Henderson	549	Spain	Female	24	0.00	0
15788448	Watson	490	Spain	Male	31	145260.23	0
15789484	Hammond	751	Germany	Female	36	169831.46	0
15792365	He	501	France	Male	44	142051.07	0
15794171	Lombardo	475	France	Female	45	134264.04	1
15803136	Postle	416	Germany	Female	41	122189.66	0
15804771	Velazquez	614	France	Male	51	40685.92	0
15805254	Ndukaku	652	Spain	Female	75	0.00	0
15809248	Cole	524	France	Female	36	0.00	0
15812518	Palermo	657	Spain	Female	37	163607.18	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result 9 Result 10 CustomerChurn 11 CustomerChurn_Stage 12

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.

Query 1 customerchurn_stage

Limit to 1000 rows

```
use customer;

drop table if exists CustomerChurn_Version1;

CREATE TABLE CustomerChurn_Version1 (
    CustomerID integer not Null,
    Surname char(30) not null,
    CreditScore integer not null,
    Geography char(50) not null,
    Gender char(6) not null,
    Age integer not null,
    Balance decimal(15,2) not null,
    Exited boolean,
    SourceSystemNm nvarchar(20) not null,
    CreateAgentId nvarchar(20) not null,
    CreateDtm DateTime  not null,
    ChangeAgentId nvarchar(20) not null,
    ChangeDtm Datetime not null,

    PRIMARY KEY (CustomerID)
);

Insert into CustomerChurn_Version1
Select * from CustomerChurn;

-- Desc CustomerChurn;
```

Query 1

Limit to 1000 rows

```
use customer;

Desc CustomerChurn_Version1;
```

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

Field	Type	Null	Key	Default	Extra
CustomerID	int	NO	PRI	NULL	
Surname	char(30)	NO		NULL	
CreditScore	int	NO		NULL	
Geography	char(50)	NO		NULL	
Gender	char(6)	NO		NULL	
Age	int	NO		NULL	
Balance	decimal(15,2)	NO		NULL	
Exited	tinyint(1)	YES		NULL	
SourceSystemNm	varchar(20)	NO		NULL	
CreateAgentId	varchar(20)	NO		NULL	
CreateDtm	datetime	NO		NULL	
ChangeAgentId	varchar(20)	NO		NULL	
ChangeDtm	datetime	NO		NULL	

Query 1 | customerchurn_stage

Limit to 1000 rows

```sql
SELECT count(*) FROM customer.customerchurn_version1;
```

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

count(*)
100

SQL File 2* | Q_05_A_Call_StoredProcedure | Q_05_B_Select_Table_Custom... | Q_06_A_Create_Table_Custom... | Q_06_C_Select_Table_Custom...

Limit to 1000 rows

```sql
use customer;

Select * from CustomerChurn_Version1
order by CustomerId;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

CustomerID	Surname	CreditScore	Geography	Gender	Age	Balance	Exited	SourceSystemNm	CreateAgentId	CreateDtm	ChangeAgentId	ChangeDtm
15788448	Watson	490	Spain	Male	31	145260.23	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:06:31
15789484	Hammond	751	Germany	Female	36	169831.46	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:06:31
15792365	He	501	France	Male	44	142051.07	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:06:31
15794171	Lombardo	475	France	Female	45	134264.04	1	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:06:31
15803136	Postle	416	Germany	Female	41	122189.66	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:06:31
15804771	Velazquez	614	France	Male	51	40685.92	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:06:31
15805254	Ndukaku	652	Spain	Female	75	0.00	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:06:31
15809248	Cole	524	France	Female	36	0.00	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:06:31
15812518	Palermo	657	Spain	Female	37	163607.18	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:06:31
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Query 1

Limit to 1000 rows

```sql
use customer;

delete from CustomerChurn_stage;
```

Q_06_D_Delete_Table_Custom... | Q_06_D_Select_Table_Custom... | SQL File 2*

Limit to 500 rows

```sql
use customer;
select count(*) RecordCount from Customerchurn_Stage;
Select * from Customerchurn_Stage
Order by CustomerId;
```

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

RecordCount
101

Query 1 customerchurn_stage

Limit to 1000 rows

```
1 • SELECT * FROM customer.customerchurn_stage;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

CustomerID	Surname	CreditScore	Geography	Gender	Age	Balance	Exited
15568982	Hao	726	France	Female	24	0.00	0
15569590	Yoo	601	Germany	Male	42	98495.72	1
15574012	Chu	645	Spain	Male	44	113755.78	1
15575185	Bushell	757	Spain	Male	33	77253.22	0
15577657	McDonald	732	France	Male	41	0.00	0
15585768	Cameron	582	Germany	Male	41	70349.48	0
15589475	Azikiwe	591	Spain	Female	39	0.00	1
15589975	Maclean	646	France	Female	73	97259.25	0
15592389	Vivek	684	India	Male	27	134603.88	0
15592461	Jackson	603	Germany	Male	26	109166.37	0
15592531	Bartlett	822	France	Male	50	0.00	0
15597945	Dellucci	700	Spain	Female	32	654.00	0
15600882	Scott	635	Spain	Female	35	0.00	0
15602280	Martin	829	Germany	Female	27	112045.67	1
15614049	Hu	664	France	Male	55	0.00	0
15616550	Chidiebele	698	Germany	Male	44	116363.37	0
15619304	Onio	502	France	Female	42	159660.80	1
15619360	Hsiao	472	Spain	Male	40	0.00	0
15620344	McKee	813	France	Male	29	0.00	0

customerchurn_stage 2

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).

Q_07_A_Select_Table_Custom... | Q_02_B_Create_Table_Custom... | SQL File 4* | Q_06_A_Create_Table_Custom... | SQL File 5 | Q_06_D_Select_Table_Custom...

Limit to 500 rows

```
USE customer ;
call prCustomerchurn();

USE customer ;
Select count(*) as RecordCount_CustomerChurn_Version1 from Customerchurn_version1;
Select count(*) as RecordCount_CustomerChurn from  CustomerChurn;
```

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

RecordCount_CustomerChurn_Version1
100

Result 6 × Result 7

Output

Action Output

#	Time	Action	Message
9	18:16:43	Select count(*) from CustomerChurn LIMIT 0, 500	1 row(s) returned
10	18:17:48	USE customer	0 row(s) affected
11	18:17:48	call prCustomerchurn()	7 row(s) affected
12	18:17:48	USE customer	0 row(s) affected
13	18:17:48	Select count(*) from Customerchurn_version1 LIMIT 0, 500	1 row(s) returned
14	18:17:48	Select count(*) from CustomerChurn LIMIT 0, 500	1 row(s) returned
15	18:18:33	USE customer	0 row(s) affected
16	18:18:33	Select count(*) as RecordCount_CustomerChurn_Version1 from Customerchurn_version1 LIMIT 0, 500	1 row(s) returned
17	18:19:10	USE customer	0 row(s) affected
18	18:19:10	Select count(*) as RecordCount_CustomerChurn_Version1 from Customerchurn_version1 LIMIT 0, 500	1 row(s) returned
19	18:19:10	Select count(*) as RecordCount_CustomerChurn from CustomerChurn LIMIT 0, 500	1 row(s) returned

ng\22_Sem_02_C5_DWH_Assignment_W4_20231122\02_Working\Q_07_A_Select_Table_CustomerChurn_Version1_RecordCount.sql'

Q_02_B_Create_Table_Custom... | SQL File 4* | Q_06_A_Create_Table_Custom... | SQL File 5 | Q_06_D_Select_Table_Custom... | Q_07_A_Select_Table_Record...

Limit to 500 rows

```sql
USE customer ;
call prCustomerchurn();

USE customer ;
Select count(*) as RecordCount_CustomerChurn_Version1 from Customerchurn_version1;
Select count(*) as RecordCount_CustomerChurn from  CustomerChurn;
```

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

RecordCount_CustomerChurn
101

Result 1 | Result 2

Output

Action Output

#	Time	Action	Message
12	18:17:48	USE customer	0 row(s) affected
13	18:17:48	Select count(*) from Customerchurn_version1 LIMIT 0, 500	1 row(s) returned
14	18:17:48	Select count(*) from CustomerChurn LIMIT 0, 500	1 row(s) returned
15	18:18:33	USE customer	0 row(s) affected
16	18:18:33	Select count(*) as RecordCount_CustomerChurn_Version1 from Customerchurn_version1 LIMIT 0, 500	1 row(s) returned
17	18:19:10	USE customer	0 row(s) affected
18	18:19:10	Select count(*) as RecordCount_CustomerChurn_Version1 from Customerchurn_version1 LIMIT 0, 500	1 row(s) returned
19	18:19:10	Select count(*) as RecordCount_CustomerChurn from CustomerChurn LIMIT 0, 500	1 row(s) returned
20	18:23:45	USE customer	0 row(s) affected
21	18:23:45	Select count(*) as RecordCount_CustomerChurn_Version1 from Customerchurn_version1 LIMIT 0, 500	1 row(s) returned
22	18:23:45	Select count(*) as RecordCount_CustomerChurn from CustomerChurn LIMIT 0, 500	1 row(s) returned

File 2* | Q_05_A_Call_StoredProcedure | Q_05_B_Select_Table_Custom... | Q_06_A_Create_Table_Custom... | Q_06_C_Select_Table_Custom... | Q_06_D_Delete_Table_Custom... | Q_07

Limit to 1000 rows

```sql
use customer;

Select *
from
    customerchurn_version1 as A
where
    A.CustomerId not in (Select CustomerId from CustomerChurn as B where A.CustomerId = B.CustomerId)
Order by
    A.customerId    ;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

CustomerID	Surname	CreditScore	Geography	Gender	Age	Balance	Exited	SourceSystemNm	CreateAgentId	CreateDtm	ChangeAgentId	ChangeDtm
15604348	Allard	710	Spain	Male	22	0.00	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:06:31
15687946	Osborne	556	France	Female	61	117419.35	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:06:31
15701164	Onyeorulu	506	France	Female	34	90307.62	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:06:31
15725737	Mosman	669	France	Male	46	0.00	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:06:31
15755648	Pisano	675	France	Female	21	98373.26	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:06:31
15812518	Palermo	657	Spain	Female	37	163607.18	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:06:31
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

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 (Excluds: 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.

Q_05_A_Call_StoredProcedure | Q_05_B_Select_Table_Custom... | Q_06_A_Create_Table_Custom... | Q_06_C_Select_Table_Custom... | Q_06_D_Delete_Table_Custom... | Q_07_B_S

Limit to 1000 rows

```
use customer;

Select A.* from
    customerchurn as A,
    customerChurn_Version1 as B
where
    A.customerId = B.CustomerId
    and
    ( A.Surname <> B.Surname
    or  A.CreditScore <> B.CreditScore
    or A.Geography <> B.Geography
    or A.Gender <> B.Gender
    or A.Age <> B.Age
    or A.Balance <> B.Balance
    or A.Exited <> B.Exited
    )

Order by
    A.CustomerId
   ;
```

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

CustomerID	Surname	CreditScore	Geography	Gender	Age	Balance	Exited	SourceSystemNm	CreateAgentId	CreateDtm	ChangeAgentId	ChangeDtm
15592389	Vivek	684	India	Male	27	134603.88	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:16:06
15597945	Dellucci	700	Spain	Female	32	654.00	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:16:06
15683553	Osman	788	USA	Male	22	75888.30	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:16:06
15691483	Chin	549	France	Female	25	12345.50	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:16:06
15737173	Andrews	497	Spain	Male	30	0.00	1	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:16:06
15738751	Beit	493	France	Female	46	321.00	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:16:06
15767821	Sharon	528	Hong Kong	Male	31	102016.72	0	Kaggle-CSV	root@localhost	2023-11-22 23:06:31	root@localhost	2023-11-22 23:16:06

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.

05_B_Select_Table_Custom... Q_06_A_Create_Table_Custom... Q_06_C_Select_Table_Custom... Q_06_D_Delete_Table_Custom... Q_07_B_Select_Table_Custom... Q_08_A_Select_Table_Custom...

Limit to 1000 rows

```
use customer;

Select A.*
from
    customerchurn as A
where
    A.customerid not in (select B.customerId from customerchurn_version1 as B where A.Customerid =  B.CustomerId)
Order by
    A.customerId
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

CustomerID	Surname	CreditScore	Geography	Gender	Age	Balance	Exited	SourceSystemNm	CreateAgentId	CreateDtm	ChangeAgentId	ChangeDtm
15589975	Maclean	646	France	Female	73	97259.25	0	Kaggle-CSV	root@localhost	2023-11-22 23:16:06	root@localhost	2023-11-22 23:16:06
15657566	Wieck	634	Germany	Male	24	103097.85	0	Kaggle-CSV	root@localhost	2023-11-22 23:16:06	root@localhost	2023-11-22 23:16:06
15698932	Groves	756	Germany	Male	44	137452.09	0	Kaggle-CSV	root@localhost	2023-11-22 23:16:06	root@localhost	2023-11-22 23:16:06
15726676	Marshall	616	Spain	Male	30	0.00	0	Kaggle-CSV	root@localhost	2023-11-22 23:16:06	root@localhost	2023-11-22 23:16:06
15727556	O'Donnell	744	Spain	Female	26	166297.89	0	Kaggle-CSV	root@localhost	2023-11-22 23:16:06	root@localhost	2023-11-22 23:16:06
15771977	T'ao	730	France	Female	39	99010.67	0	Kaggle-CSV	root@localhost	2023-11-22 23:16:06	root@localhost	2023-11-22 23:16:06
16812518	Palermo	657	Spain	Female	37	163607.18	0	Kaggle-CSV	root@localhost	2023-11-22 23:16:06	root@localhost	2023-11-22 23:16:06
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL