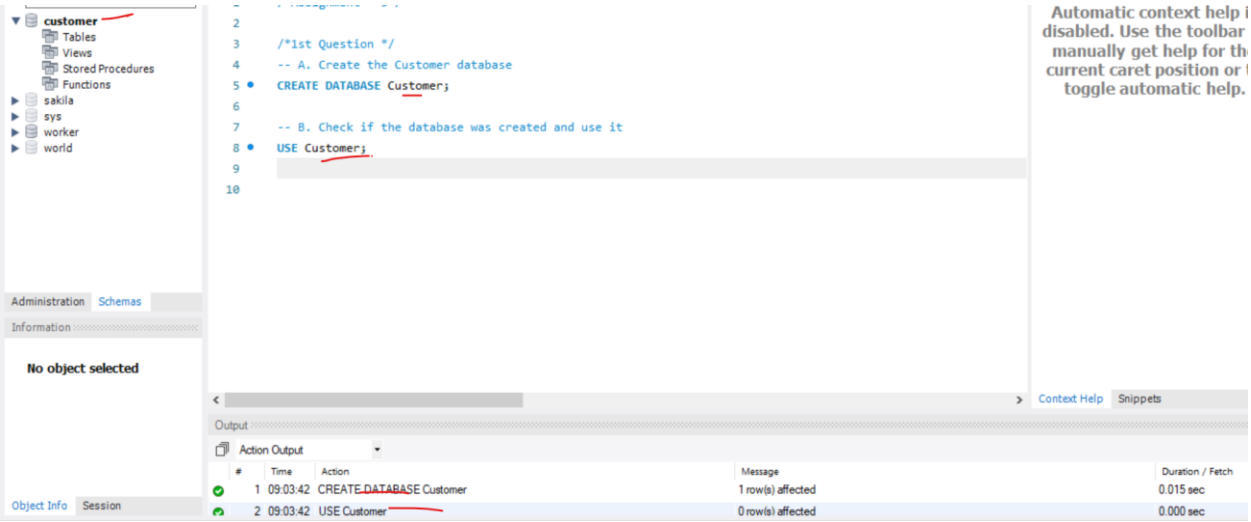
INFO 531: DATA WAREHOUSING IN THE CLOUD

ASSIGNMENT– 3

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.

```sql
-- A. Create staging table CustomerChurn_Stage
CREATE TABLE Customer.CustomerChurn_Stage (
    CustomerId INT,
    Surname VARCHAR(255),
    CreditScore INT,
    Geography VARCHAR(255),
    Gender VARCHAR(50),
    Age INT,
    Balance DECIMAL(10, 2),
    Exited INT
);
```

```sql
-- B. Create persistent table CustomerChurn with additional columns
CREATE TABLE Customer.CustomerChurn (
    CustomerId INT PRIMARY KEY,
    Surname VARCHAR(255),
    CreditScore INT,
    Geography VARCHAR(255),
    Gender VARCHAR(50),
    Age INT,
    Balance DECIMAL(10, 2),
    Exited INT,
    SourceSystemNm NVARCHAR(20) NOT NULL,
    CreateAgentId NVARCHAR(20) NOT NULL,
    CreateDtm DATETIME NOT NULL,
    ChangeAgentId NVARCHAR(20) NOT NULL,
    ChangeDtm DATETIME NOT NULL
);
```

Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.

Context Help | Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
5	10:37:53	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Custo...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0	0.015 sec
6	10:37:53	SELECT * FROM Customer.CustomerChum_Stage ORDER BY CustomerId LIMIT 0...	100 row(s) returned	0.000 sec / 0.000 sec
7	11:17:14	DROP TABLE `customer`.`customerchurn`	0 row(s) affected	0.047 sec
8	11:22:49	CREATE TABLE Customer.CustomerChum (CustomerId INT PRIMARY KEY, ...	0 row(s) affected, 3 warning(s): 3720 NATIONAL/NCHAR/NVARCHAR implies the ...	0.031 sec

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 '.

```
57      -- A. Load data into CustomerChurn_Stage from CustomerChurn1.csv
58  •   LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/CustomerChurn1.csv'
59      INTO TABLE Customer.CustomerChurn_Stage
60      FIELDS TERMINATED BY ','
61      LINES TERMINATED BY '\n'
62      IGNORE 1 ROWS
63      (CustomerId, Surname, CreditScore, Geography, Gender, Age, Balance, Exited);
```

```
65      -- B. Verify data
66  •   SELECT * FROM Customer.CustomerChurn_Stage ORDER BY CustomerId;
67
68
```

Result Grid | Filter Rows: | Export: | 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

Result Grid | Form Editor | Read Only | Context Help | Snippets

Churn_Stage 1

Output

Action Output

#	Time	Action	Message	Duration / Fe
3	10:36:10	CREATE TABLE Customer.CustomerChurn_Stage (CustomerId INT, Surname ...	0 row(s) affected	0.047 sec
4	10:36:20	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Custo...	Error Code: 1054. Unknown column 'SeniorCitizen' in 'field list'	0.000 sec
5	10:37:53	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Custo...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0	0.015 sec
6	10:37:53	SELECT * FROM Customer.CustomerChurn_Stage ORDER BY CustomerId LIMIT 0...	100 row(s) returned	0.000 sec / 0

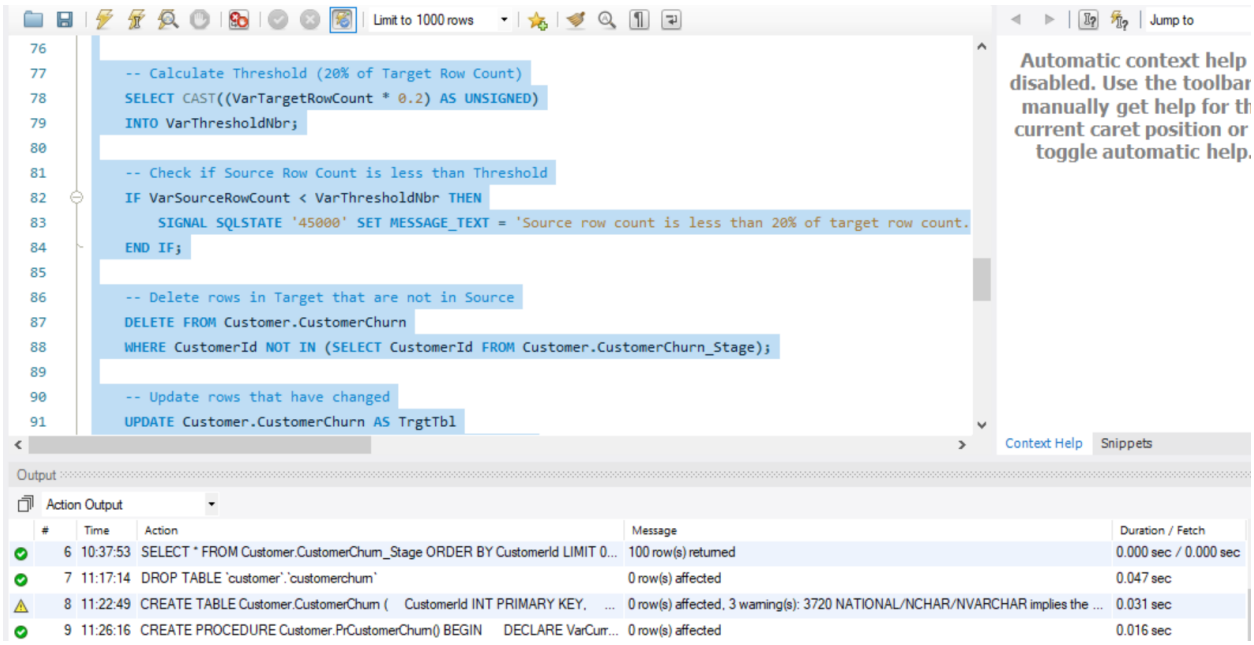
Q4. Create a database stored procedure based on the template provided along with this assignment << StoredProc_Template.txt >>. Name the stored procedure name this: ** Customer.PrCustomerChurn ** . [[NOTE : This stored procedure will use the table, ** Customer.CustomerChurn_Stage ** , as the source (aka,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

SQL qUERY:

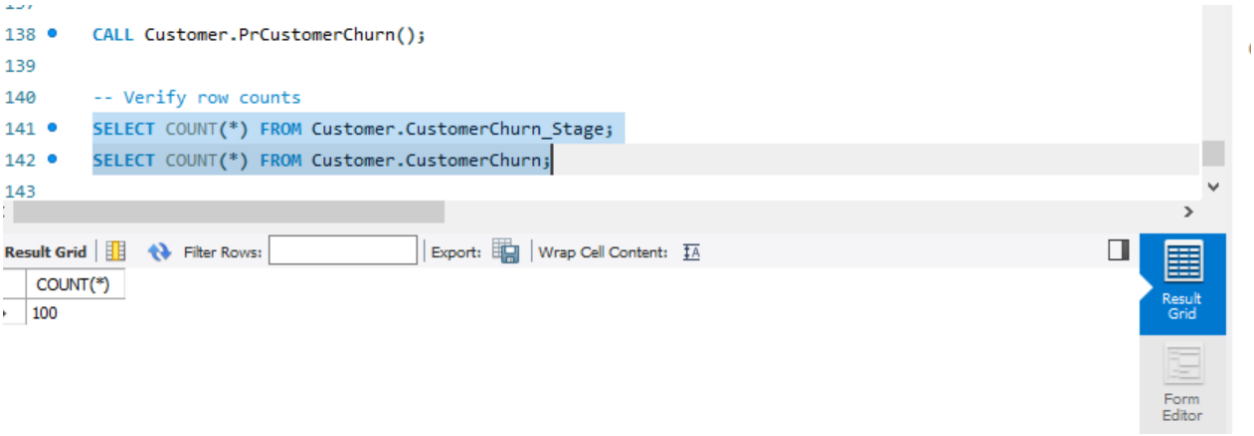
```
DELIMITER //

CREATE PROCEDURE Customer.PrCustomerChurn()
BEGIN

  DECLARE VarCurrentTimestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP;
  DECLARE VarSourceRowCount, VarTargetRowCount, VarThresholdNbr INTEGER DEFAULT 0;
  DECLARE VarTinyIntVal TINYINT;

  -- Get Source and Target Row Counts
  SELECT COUNT(*)
  INTO VarSourceRowCount
  FROM Customer.CustomerChurn_Stage;

  SELECT COUNT(*)
  INTO VarTargetRowCount
  FROM Customer.CustomerChurn;

  -- Calculate Threshold (20% of Target Row Count)
  SELECT CAST((VarTargetRowCount * 0.2) AS UNSIGNED)
  INTO VarThresholdNbr;
```

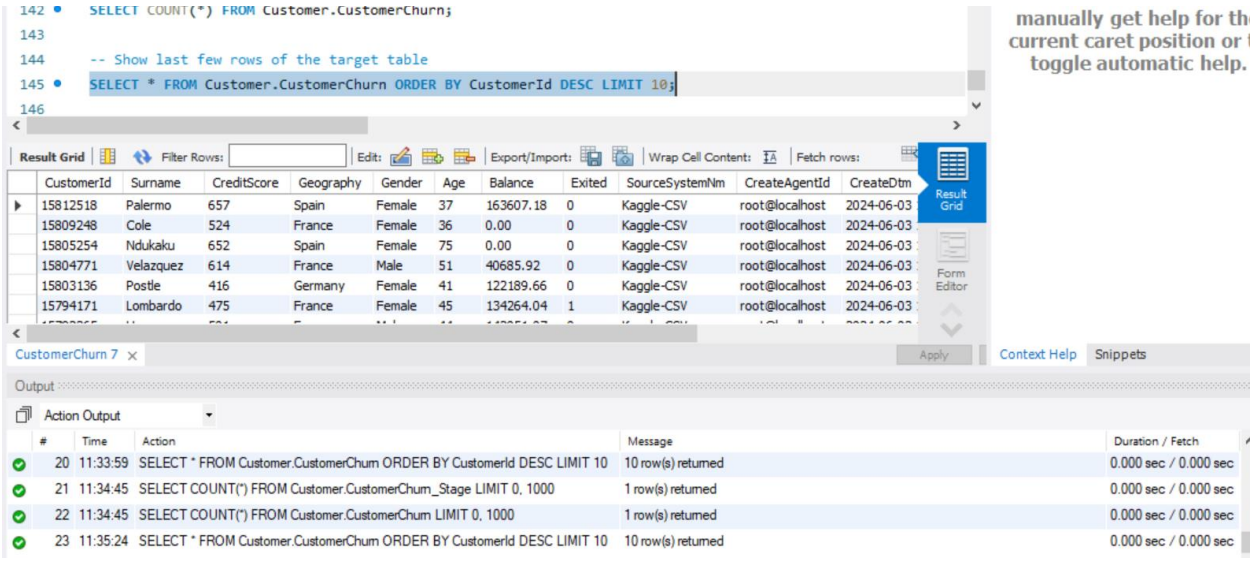
```
    -- Check if Source Row Count is less than Threshold
    IF VarSourceRowCount < VarThresholdNbr THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Source row count is less than 20% of target
row count.';
    END IF;

    -- Delete rows in Target that are not in Source
    DELETE FROM Customer.CustomerChurn
    WHERE CustomerId NOT IN (SELECT CustomerId FROM Customer.CustomerChurn_Stage);

    -- Update rows that have changed
    UPDATE Customer.CustomerChurn AS TrgtTbl
    INNER JOIN Customer.CustomerChurn_Stage AS SrcTbl
    ON TrgtTbl.CustomerId = SrcTbl.CustomerId
    SET TrgtTbl.Surname = SrcTbl.Surname,
        TrgtTbl.CreditScore = SrcTbl.CreditScore,
        TrgtTbl.Geography = SrcTbl.Geography,
        TrgtTbl.Gender = SrcTbl.Gender,
        TrgtTbl.Age = SrcTbl.Age,
        TrgtTbl.Balance = SrcTbl.Balance,
        TrgtTbl.Exited = SrcTbl.Exited,
        TrgtTbl.ChangeAgentId = CURRENT_USER(),
        TrgtTbl.ChangeDtm = VarCurrentTimestamp
    WHERE (
        COALESCE(TrgtTbl.Surname, '*') <> COALESCE(SrcTbl.Surname, '*') OR
```

```
        COALESCE(TrgtTbl.CreditScore, '*') <> COALESCE(SrcTbl.CreditScore, '*') OR
        COALESCE(TrgtTbl.Geography, '*') <> COALESCE(SrcTbl.Geography, '*') OR
        COALESCE(TrgtTbl.Gender, '*') <> COALESCE(SrcTbl.Gender, '*') OR
        COALESCE(TrgtTbl.Age, '*') <> COALESCE(SrcTbl.Age, '*') OR
        COALESCE(TrgtTbl.Balance, '*') <> COALESCE(SrcTbl.Balance, '*') OR
        COALESCE(TrgtTbl.Exited, '*') <> COALESCE(SrcTbl.Exited, '*')
    );

    -- Insert new rows
    INSERT INTO Customer.CustomerChurn (
        CustomerId, Surname, CreditScore, Geography, Gender, Age, Balance, Exited,
        SourceSystemNm, CreateAgentId, CreateDtm, ChangeAgentId, ChangeDtm
    )
    SELECT
        SrcTbl.CustomerId, SrcTbl.Surname, SrcTbl.CreditScore, SrcTbl.Geography,
        SrcTbl.Gender, SrcTbl.Age, SrcTbl.Balance, SrcTbl.Exited,
        'Kaggle-CSV', CURRENT_USER(), VarCurrentTimestamp, CURRENT_USER(),
VarCurrentTimestamp
    FROM Customer.CustomerChurn_Stage AS SrcTbl
    LEFT JOIN Customer.CustomerChurn AS TrgtTbl
    ON SrcTbl.CustomerId = TrgtTbl.CustomerId
    WHERE TrgtTbl.CustomerId IS NULL;

END //

DELIMITER ;
```



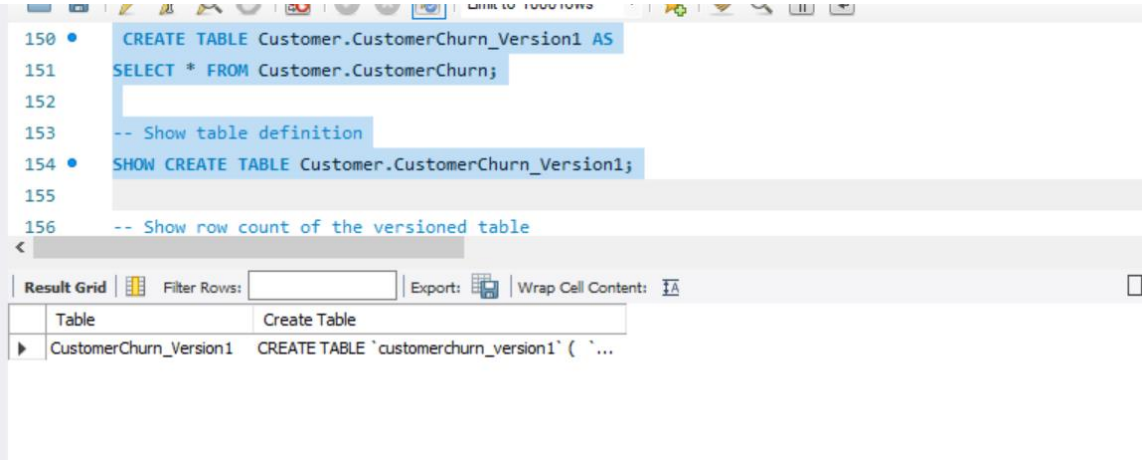
Q5. Execute the stored procedure, ** Customer.PrCustomerChurn **, that was created in Q4. After execution, the stored procedure should load data from the stage to the persistent table: ** Customer.CustomerChurn **. {A} Verify data by comparing the row counts between the staging table, ** Customer.CustomerChurn_Stage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

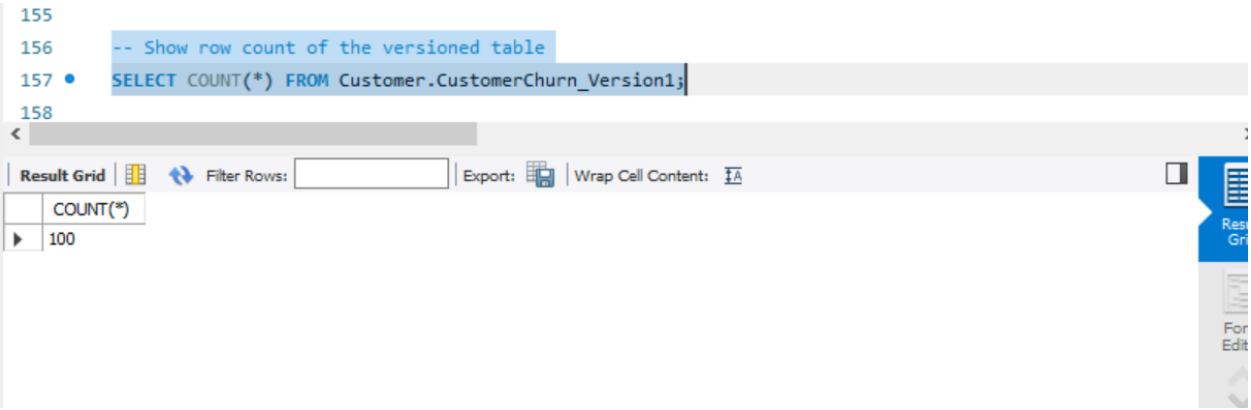




Q6. After data verification is completed, in Q5 , { A } create table, ** Customer.CustomerChurn_Version1 **, with data from ** Customer.CustomerChurn ** (that was already loaded from Customer.CustomerChurn_Stage via the stored procedure). { B } Show table definition of Customer.CustomerChurn_Version1 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.





```sql
-- Show last few rows of the versioned table
SELECT * FROM Customer.CustomerChurn_Version1 ORDER BY CustomerId DESC LIMIT 10;
```

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

CustomerId	Surname	CreditScore	Geography	Gender	Age	Balance	Exited	SourceSystemNm	CreateAgentId	CreateD
15812518	Palermo	657	Spain	Female	37	163607.18	0	Kaggle-CSV	root@localhost	2024-06-
15809248	Cole	524	France	Female	36	0.00	0	Kaggle-CSV	root@localhost	2024-06-
15805254	Ndukaku	652	Spain	Female	75	0.00	0	Kaggle-CSV	root@localhost	2024-06-
15804771	Velazquez	614	France	Male	51	40685.92	0	Kaggle-CSV	root@localhost	2024-06-
15803136	Postle	416	Germany	Female	41	122189.66	0	Kaggle-CSV	root@localhost	2024-06-
15794171	Lombardo	475	France	Female	45	134264.04	1	Kaggle-CSV	root@localhost	2024-06-

```sql
-- Empty the staging table
TRUNCATE TABLE Customer.CustomerChurn_Stage;
```

```sql
-- Load new data from CustomerChurn2.csv
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/CustomerChurn2.csv'
INTO TABLE Customer.CustomerChurn_Stage
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(CustomerId, Surname, CreditScore, Geography, Gender, Age, Balance, Exited);
```

current caret position o
toggle automatic hel

Context Help | Snippets

on Output

	Time	Action	Message	Duration / Fetch
7	11:52:39	CREATE DATABASE Customer	Error Code: 1007. Can't create database 'customer'; database exists	0.000 sec
8	11:52:54	SELECT * FROM Customer.CustomerChurn_Version1 ORDER BY CustomerId DES...	10 row(s) returned	0.000 sec / 0.000 sec
9	11:54:08	TRUNCATE TABLE Customer.CustomerChurn_Stage	0 row(s) affected	0.047 sec
0	11:56:16	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Custo...	101 row(s) affected Records: 101 Deleted: 0 Skipped: 0 Warnings: 0	0.015 sec

```
173    -- Verify new data row count
174 •  SELECT COUNT(*) FROM Customer.CustomerChurn_Stage;
175
176    -- Show last few rows of the new data in the staging table
177 •  SELECT * FROM Customer.CustomerChurn_Stage ORDER BY CustomerId DESC LIMIT 10;
```

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

COUNT(*)
101

Result Grid | Form Editor

Result 11 ×

Read Only

```
175
176    -- Show last few rows of the new data in the staging table
177 •  SELECT * FROM Customer.CustomerChurn_Stage ORDER BY CustomerId DESC LIMIT 10;
178
```

current caret position or t
toggle automatic help.

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

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
15794171	Lombardo	475	France	Female	45	134264.04	1
15792365	He	501	France	Male	44	142051.07	0

Result Grid | Form Editor

CustomerChurn_Stage 12 ×

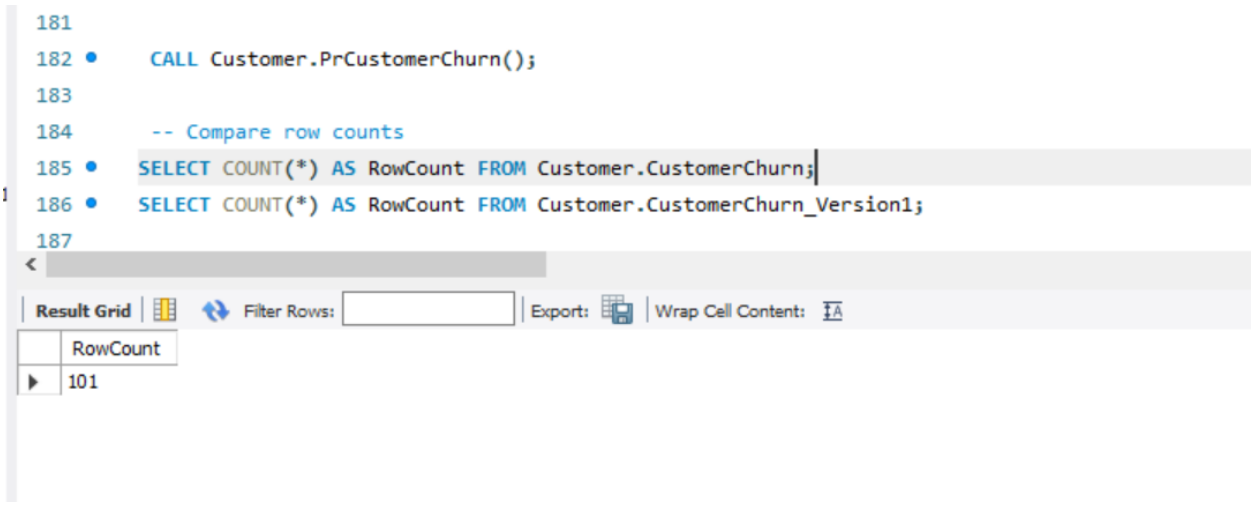
Read Only | Context Help | Snippets

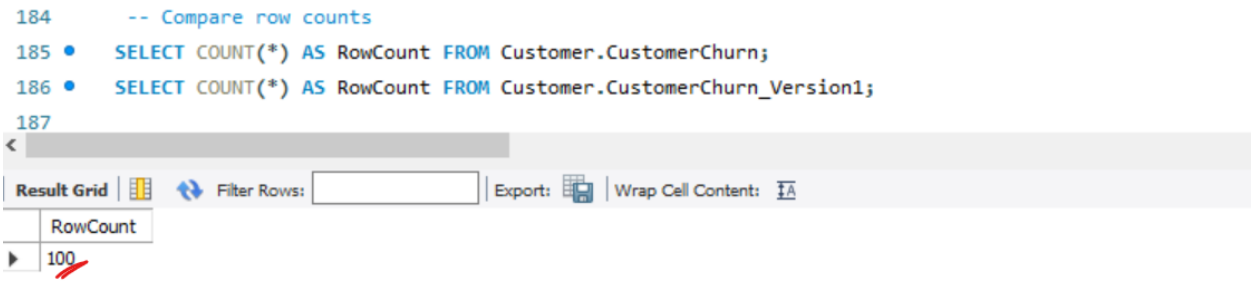
Output

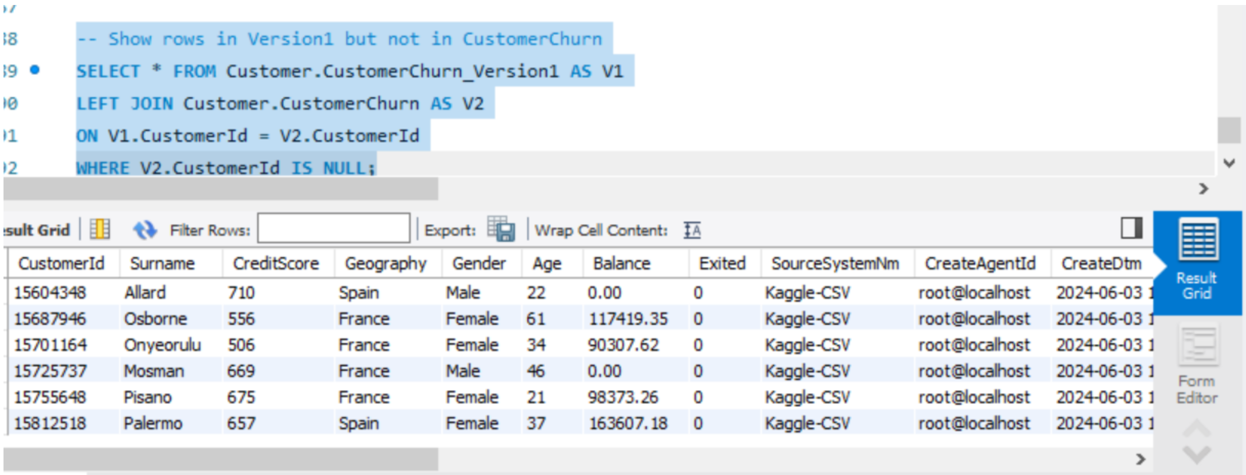
Action Output

#	Time	Action	Message	Duration / Fetch
29	11:54:08	TRUNCATE TABLE Customer.CustomerChurn_Stage	0 row(s) affected	0.047 sec
30	11:56:16	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Custo...	101 row(s) affected Records: 101 Deleted: 0 Skipped: 0 Warnings: 0	0.015 sec
31	11:57:16	SELECT COUNT(*) FROM Customer.CustomerChurn_Stage LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
32	11:57:45	SELECT * FROM Customer.CustomerChurn_Stage ORDER BY CustomerId DESC L...	10 row(s) returned	0.000 sec / 0.000 sec

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).

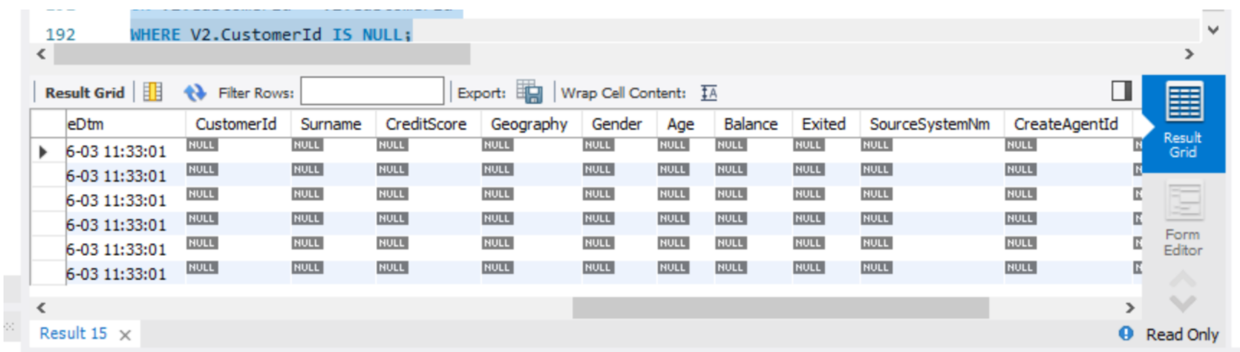






```sql
-- Show rows in Version1 but not in CustomerChurn
SELECT * FROM Customer.CustomerChurn_Version1 AS V1
LEFT JOIN Customer.CustomerChurn AS V2
ON V1.CustomerId = V2.CustomerId
WHERE V2.CustomerId IS NULL;
```

CustomerId	Surname	CreditScore	Geography	Gender	Age	Balance	Exited	SourceSystemNm	CreateAgentId	CreateDtm
15604348	Allard	710	Spain	Male	22	0.00	0	Kaggle-CSV	root@localhost	2024-06-03 1
15687946	Osborne	556	France	Female	61	117419.35	0	Kaggle-CSV	root@localhost	2024-06-03 1
15701164	Onyeorulu	506	France	Female	34	90307.62	0	Kaggle-CSV	root@localhost	2024-06-03 1
15725737	Mosman	669	France	Male	46	0.00	0	Kaggle-CSV	root@localhost	2024-06-03 1
15755648	Pisano	675	France	Female	21	98373.26	0	Kaggle-CSV	root@localhost	2024-06-03 1
15812518	Palermo	657	Spain	Female	37	163607.18	0	Kaggle-CSV	root@localhost	2024-06-03 1



```sql
WHERE V2.CustomerId IS NULL;
```

eDtm	CustomerId	Surname	CreditScore	Geography	Gender	Age	Balance	Exited	SourceSystemNm	CreateAgentId
6-03 11:33:01	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
6-03 11:33:01	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
6-03 11:33:01	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
6-03 11:33:01	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
6-03 11:33:01	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
6-03 11:33:01	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result 15 · Read Only

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.

```
197 •  SELECT *
198    FROM Customer.CustomerChurn AS V2
199    INNER JOIN Customer.CustomerChurn_Version1 AS V1
200    ON V2.CustomerId = V1.CustomerId
201    WHERE V2.Surname <> V1.Surname
202       OR V2.CreditScore <> V1.CreditScore
203       OR V2.Geography <> V1.Geography
204       OR V2.Gender <> V1.Gender
```

Automatic context help disabled. Use the toolbar manually get help for th current caret position or toggle automatic help.

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

CustomerId	Surname	CreditScore	Geography	Gender	Age	Balance	Exited	SourceSystemNm	CreateAgentId	CreateDtm
15592389	Vivek	684	India	Male	27	134603.88	0	Kaggle-CSV	root@localhost	2024-06-03
15597945	Dellucci	700	Spain	Female	32	654.00	0	Kaggle-CSV	root@localhost	2024-06-03
15683553	Osman	788	USA	Male	22	75888.30	0	Kaggle-CSV	root@localhost	2024-06-03
15691483	Chin	549	France	Female	25	12345.50	0	Kaggle-CSV	root@localhost	2024-06-03
15737173	Andrews	497	Spain	Male	30	0.00	1	Kaggle-CSV	root@localhost	2024-06-03
15738751	Beit	493	France	Female	46	321.00	0	Kaggle-CSV	root@localhost	2024-06-03
15767821	Sharon	528	Hong Kong	Male	31	102016.72	0	Kaggle-CSV	root@localhost	2024-06-03

Result Grid | Form Editor

Result 16 ×

Read Only | Context Help | Snippets

Output

Action Output

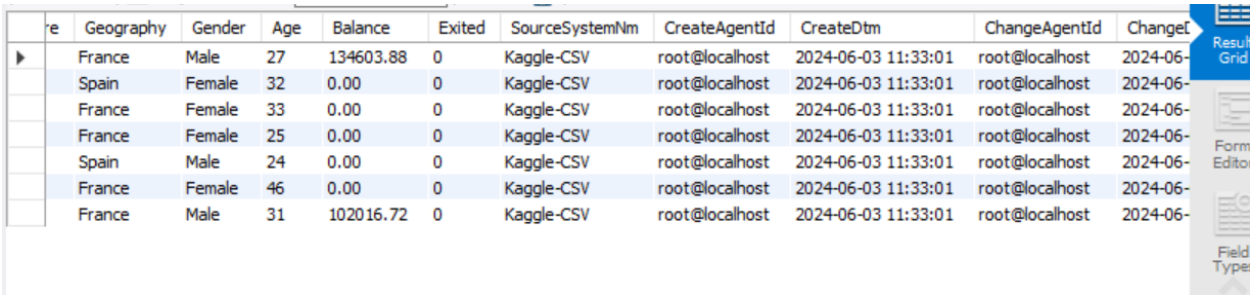
#	Time	Action	Message	Duration / Fetch
34	12:02:36	SELECT COUNT(*) AS RowCount FROM Customer.CustomerChurn LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
35	12:02:36	SELECT COUNT(*) AS RowCount FROM Customer.CustomerChurn_Version1 LIMI...	1 row(s) returned	0.000 sec / 0.000 sec
36	12:04:53	SELECT * FROM Customer.CustomerChurn_Version1 AS V1 LEFT JOIN Customer....	6 row(s) returned	0.000 sec / 0.000 sec

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

CustomerId	Surname	CreditScore	Geography	Gender	Age	Balance	Exited	SourceSystemNm	CreateAgentId	CreateDtm
15592389	Vivek	684	India	Male	27	134603.88	0	Kaggle-CSV	root@localhost	2024-06-03 11
15597945	Dellucci	700	Spain	Female	32	654.00	0	Kaggle-CSV	root@localhost	2024-06-03 11
15683553	Osman	788	USA	Male	22	75888.30	0	Kaggle-CSV	root@localhost	2024-06-03 11
15691483	Chin	549	France	Female	25	12345.50	0	Kaggle-CSV	root@localhost	2024-06-03 11
15737173	Andrews	497	Spain	Male	30	0.00	1	Kaggle-CSV	root@localhost	2024-06-03 11
15738751	Beit	493	France	Female	46	321.00	0	Kaggle-CSV	root@localhost	2024-06-03 11
15767821	Sharon	528	Hong Kong	Male	31	102016.72	0	Kaggle-CSV	root@localhost	2024-06-03 11

Result Grid | Form Editor | Field Types

SourceSystemNm	CreateAgentId	CreateDtm	ChangeAgentId	ChangeDtm	CustomerId	Surname	CreditScore
Kaggle-CSV	root@localhost	2024-06-03 11:33:01	root@localhost	2024-06-03 12:02:07	15592389	H?	684
Kaggle-CSV	root@localhost	2024-06-03 11:33:01	root@localhost	2024-06-03 12:02:07	15597945	Dellucci	636
Kaggle-CSV	root@localhost	2024-06-03 11:33:01	root@localhost	2024-06-03 12:02:07	15683553	O'Brien	788
Kaggle-CSV	root@localhost	2024-06-03 11:33:01	root@localhost	2024-06-03 12:02:07	15691483	Chin	549
Kaggle-CSV	root@localhost	2024-06-03 11:33:01	root@localhost	2024-06-03 12:02:07	15737173	Andrews	497
Kaggle-CSV	root@localhost	2024-06-03 11:33:01	root@localhost	2024-06-03 12:02:07	15738751	Beit	493
Kaggle-CSV	root@localhost	2024-06-03 11:33:01	root@localhost	2024-06-03 12:02:07	15767821	Bearce	528



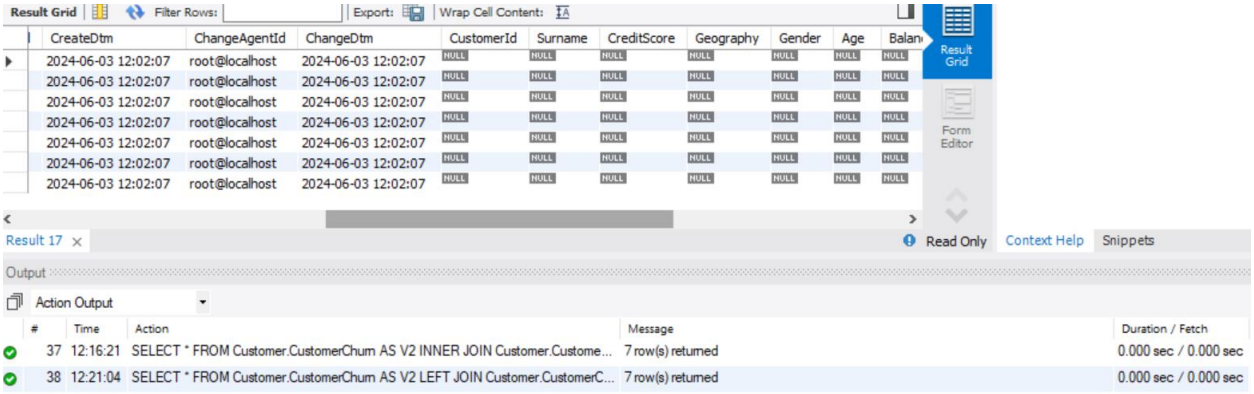
Geography	Gender	Age	Balance	Exited	SourceSystemNm	CreateAgentId	CreateDtm	ChangeAgentId	ChangeD
France	Male	27	134603.88	0	Kaggle-CSV	root@localhost	2024-06-03 11:33:01	root@localhost	2024-06-
Spain	Female	32	0.00	0	Kaggle-CSV	root@localhost	2024-06-03 11:33:01	root@localhost	2024-06-
France	Female	33	0.00	0	Kaggle-CSV	root@localhost	2024-06-03 11:33:01	root@localhost	2024-06-
France	Female	25	0.00	0	Kaggle-CSV	root@localhost	2024-06-03 11:33:01	root@localhost	2024-06-
Spain	Male	24	0.00	0	Kaggle-CSV	root@localhost	2024-06-03 11:33:01	root@localhost	2024-06-
France	Female	46	0.00	0	Kaggle-CSV	root@localhost	2024-06-03 11:33:01	root@localhost	2024-06-
France	Male	31	102016.72	0	Kaggle-CSV	root@localhost	2024-06-03 11:33:01	root@localhost	2024-06-

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.

```
212
213 •  SELECT *
214    FROM Customer.CustomerChurn AS V2
215    LEFT JOIN Customer.CustomerChurn_Version1 AS V1
216    ON V2.CustomerId = V1.CustomerId
217    WHERE V1.CustomerId IS NULL
218    ORDER BY V2.CustomerId;
219
```

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

CustomerId	Surname	CreditScore	Geography	Gender	Age	Balance	Exited	SourceSystemNm	CreateAgentId	CreateDtm
15589975	Maclean	646	France	Female	73	97259.25	0	Kaggle-CSV	root@localhost	2024-06-03 12
15657566	Wieck	634	Germany	Male	24	103097.85	0	Kaggle-CSV	root@localhost	2024-06-03 12
15698932	Groves	756	Germany	Male	44	137452.09	0	Kaggle-CSV	root@localhost	2024-06-03 12
15726676	Marshall	616	Spain	Male	30	0.00	0	Kaggle-CSV	root@localhost	2024-06-03 12
15727556	O'Donnell	744	Spain	Female	26	166297.89	0	Kaggle-CSV	root@localhost	2024-06-03 12
15771977	T'ao	730	France	Female	39	99010.67	0	Kaggle-CSV	root@localhost	2024-06-03 12
16812518	Palermo	657	Spain	Female	37	163607.18	0	Kaggle-CSV	root@localhost	2024-06-03 12


Result Grid
Filter Rows:
Export:
Wrap Cell Content:
CreateDtm
ChangeAgentId
ChangeDtm
CustomerId
Surname
CreditScore
Geography
Gender
Age
Balan
2024-06-03 12:02:07
root@localhost
NULL
Result Grid
Form Editor
Result 17
Read Only
Context Help
Snippets
Output
Action Output
Time Action Message Duration / Fetch
37 12:16:21 SELECT * FROM Customer.CustomerChum AS V2 INNER JOIN Customer.Custome... 7 row(s) returned 0.000 sec / 0.000 sec
38 12:21:04 SELECT * FROM Customer.CustomerChum AS V2 LEFT JOIN Customer.CustomerC... 7 row(s) returned 0.000 sec / 0.000 sec