



SQL Programming Project





Submitted by

Mohammad Monjur-E-Elahi

Course: SQL Programming [DSA03]

Program: Data Science and Application - Advanced Diploma [6060]

Metro College of Technology

Date: 01 September, 2021





Phase 01

- 1. Create a database for a banking application called “Bank”. [Basic]**
- 2. Create all the tables mentioned in the database diagram. [Moderate]**
- 3. Create all the constraints based on the database diagram. [Advanced]**
- 4. Insert at least 5 rows in each table. [Basic]**



Codes for Database, Tables, Constraints and Data Insertion

```
1  create database dbprojectbank;
2  use dbprojectbank;
3  --Tables in the project are:
4  --01. UserLogins
5  --02. UserSecurityQuestions
6  --03. SavingsInterestRates
7  --04. TransactionType
8  --05. AccountType
9  --06. AccountStatusType
10 --07. LoginErrorLog
11 --08. FailedTransactionErrorType
12 --09. Employee
13 --10. Customer
14 --11. Account
15 --12. OverDraftLog
16 --13. LoginAccount
17 --14. UserSecurityAnswers
18 --15. CustomerAccount
19 --16. TransactionLog
20 --17. FailedTransactionLog
21
22
23
24
25 -- Table count 1
26
27 create table UserLogins
28 (
29     UserLoginID smallint primary key not null ,
30     UserName char(15) not null,
31     UserPassword varchar(20) not null
32 );
33
34
35 --Table count 2
36
37 create table UserSecurityQuestions
38 (
39     UserSecurityQuestionID tinyint primary key not null,
40     UserSecurityQuestion1 varchar(50) not null,
41     UserSecurityQuestion2 varchar(50) not null,
42     UserSecurityQuestion3 varchar(50) not null
43 );
44
45
46 --Table count 3
47
48 create table SavingsInterestRates
49 (
50     SavingsInterestRateID tinyint primary key not null,
51     SavingsInterestRateValue numeric(9,9) not null,
52     SavingsInterestRateDescription varchar(20) not null
```

```
53    );
54
55 --Table count 4
56
57 create table TransactionType
58 (
59     TransactionTypeID tinyint primary key not null,
60     TransactionTypeName char(10) not null,
61     TransactionTypeDescription varchar(50),
62     TransactionFeeAmount smallmoney
63 );
64
65 --Table count 5
66
67 create table AccountType
68 (
69     AccountTypeID tinyint primary key not null,
70     AccountTypeDescription varchar(30) not null
71 );
72
73
74
75
76
77 --Table count 6
78
79 create table AccountStatusType
80 (
81     AccountStatusTypeID tinyint primary key not null,
82     AccountStatusDescription varchar(30) not null,
83
84 );
85
86
87
88 --Table count 7
89
90 create table LoginErrorLog
91 (
92     ErrorLogID int primary key not null,
93     ErrorTime datetime not null,
94     FailedTransactionXML xml not null,
95
96 );
97
98 --Table count 8
99
100 create table FailedTransactionErrorType
101 (
102     FailedTransactionErrorTypeID tinyint primary key not null,
103     FailedTransactionDescription varchar(50)
104
```

```
105 );
106
107 --Table count 9
108
109 create table Employee
110 (
111     EmployeeID int primary key not null,
112     EmployeeFirstName varchar(25) not null,
113     EmployeeMiddleInitial char(1),
114     EmployeeLastName varchar(25) not null,
115     EmployeeIsManager bit not null
116
117 );
118
119 --Table count 10
120
121
122 create table Customer
123 (
124     CustomerID int primary key not null,
125     CustomerAddress1 varchar(30) not null,
126     CustomerAddress2 varchar(30),
127     CustomerFirstName varchar(30) not null,
128     CustomerMiddleInitial char(1),
129     CustomerLastName varchar(30) not null,
130     City varchar(30) not null,
131     State char(2) not null,
132     Zipcode char(10) not null,
133     Email varchar(40) not null,
134     HomePhone char(10),
135     CellPhone char(10) not null,
136     WorkPhone char(10),
137     SSN char(9) not null,
138     UserLoginID smallint,
139     Constraint FK2 foreign key(UserLoginID) references UserLogins(UserLoginID)
140 );
141
142 --Table count 11
143
144
145 create table Account
146 (
147     AccountID int primary key not null,
148     CurrentBalance int not null,
149     AccountTypeID tinyint not null,
150     AccountStatusTypeID tinyint not null,
151     InterestSavingsRateID tinyint,
152     Constraint FK4 foreign key(AccountTypeID) references AccountType(AccountTypeID),
153     Constraint FK5 foreign key(AccountStatusTypeID) references AccountStatusType(AccountStatusTypeID),
154     Constraint FK6 foreign key(InterestSavingsRateID) references SavingsInterestRates(SavingsInterestRateID)
155
156
```

```
157 );
158
159 --ALTER TABLE Account ALTER COLUMN InterestSavingsRateID tinyint NULL;
160
161 --Table count 12
162
163 create table OverDraftLog
164 (
165     AccountID int primary key not null,
166     OverDraftDate datetime not null,
167     OverDraftAmount money not null,
168     OverDraftTransactionXml xml not null,
169     Constraint FK1 foreign key(AccountID) references Account(AccountID)
170 );
171
172 --Table count 13
173
174 create table LoginAccount
175 (
176     UserLoginID smallint not null,
177     AccountID int not null,
178     Constraint FK11 foreign key(AccountID) references Account(AccountID)
179 );
180
181 --Table count 14
182
183 create table UserSecurityAnswers
184 (
185     UserLoginID smallint primary key not null,
186     UserSecurityAnswer1 varchar(25) not null,
187     UserSecurityAnswer2 varchar(25) not null,
188     UserSecurityAnswer3 varchar(25) not null,
189     UserSecurityQuestionID tinyint not null,
190     Constraint FK12 foreign key(UserLoginID) references UserLogins(UserLoginID),
191     Constraint FK13 foreign key(UserSecurityQuestionID) references UserSecurityQuestions(UserSecurityQuestionID)
192 );
193
194 --Table count 15
195
196 create table CustomerAccount
197 (
198     AccountID int not null,
199     CustomerID int not null,
200     Constraint FK14 foreign key(AccountID) references Account(AccountID),
201     Constraint FK15 foreign key(CustomerID) references Customer(CustomerID)
202 );
203
204 --Table count 16
205
206 create table TransactionLog
207 (
208     TransactionID int IDENTITY(56770001, 1) primary key not null,
```

```
209     TransactionDate datetime not null,
210     TransactionTypeID tinyint not null,
211     TransactionAmount money not null,
212     NewBalance money not null,
213     AccountID int not null,
214     CustomerID int not null,
215     EmployeeID int not null,
216     UserLoginID smallint not null,
217     Constraint FK7 foreign key(TransactionTypeID) references TransactionType(TransactionTypeID),
218     Constraint FK16 foreign key(AccountID) references Account(AccountID),
219     Constraint FK3 foreign key(CustomerID) references Customer(CustomerID),
220     Constraint FK17 foreign key(EmployeeID) references Employee(EmployeeID),
221     Constraint FK18 foreign key(UserLoginID) references UserLogins(UserLoginID)
222   );
223
224 --Table count 17
225
226
227 create table FailedTransactionLog
228 (
229     FailedTransactionID int primary key not null,
230     FailedTransactionErrorTypeID tinyint not null,
231     FailedTransactionErrorTime datetime not null,
232     FailedTransactionXML xml not null,
233     Constraint FK19 foreign key(FailedTransactionErrorTypeID) references FailedTransactionErrorType(FailedTransactionErrorTypeID)
234 );
235
236
237 --Bulk date insertion 1
238
239 BULK INSERT UserLogins
240   FROM 'C:\Users\ruzdomain\Desktop\SQL\PROJECT\UserLogins.csv'
241   WITH (
242     FORMAT = 'CSV',
243     FIRSTROW = 2,
244     FIELDTERMINATOR = ',',
245     ROWTERMINATOR = '\n'
246   )
247
248 select * from UserLogins;
249
250 --Bulk date insertion 2
251
252 BULK INSERT UserSecurityQuestions
253   FROM 'C:\Users\ruzdomain\Desktop\SQL\PROJECT\UserSecurityQuestions.csv'
254   WITH (
255     FORMAT = 'CSV',
256     FIRSTROW = 2,
257     FIELDTERMINATOR = ',',
258     ROWTERMINATOR = '\n'
259   )
```

```
261 select * from UserSecurityQuestions;
262
263 --Bulk date insertion 3
264
265 BULK INSERT SavingsInterestRates
266   FROM 'C:\Users\ruzdomain\Desktop\SQL\PROJECT\SavingsInterestRates.csv'
267   WITH (
268     FORMAT = 'CSV',
269     FIRSTROW = 2,
270     FIELDTERMINATOR = ',',
271     ROWTERMINATOR = '\n'
272   )
273
274 select * from SavingsInterestRates;
275
276 --Bulk date insertion 4
277
278 BULK INSERT TransactionType
279   FROM 'C:\Users\ruzdomain\Desktop\SQL\PROJECT\TransactionType.csv'
280   WITH (
281     FORMAT = 'CSV',
282     FIRSTROW = 2,
283     FIELDTERMINATOR = ',',
284     ROWTERMINATOR = '\n'
285   )
286
287 select * from TransactionType;
288
289 --Bulk date insertion 5
290
291 BULK INSERT AccountType
292   FROM 'C:\Users\ruzdomain\Desktop\SQL\PROJECT\AccountType.csv'
293   WITH (
294     FORMAT = 'CSV',
295     FIRSTROW = 2,
296     FIELDTERMINATOR = ',',
297     ROWTERMINATOR = '\n'
298   )
299
300 select * from AccountType;
301
302 --Bulk date insertion 6
303
304 BULK INSERT AccountStatusType
305   FROM 'C:\Users\ruzdomain\Desktop\SQL\PROJECT\AccountStatusType.csv'
306   WITH (
307     FORMAT = 'CSV',
308     FIRSTROW = 2,
309     FIELDTERMINATOR = ',',
310     ROWTERMINATOR = '\n'
311   )
312
```

```
313 select * from AccountStatusType;
314
315 --Bulk date insertion 7
316
317 BULK INSERT FailedTransactionErrorType
318     FROM 'C:\Users\ruzdomain\Desktop\SQL\PROJECT\FailedTransactionErrorType.csv'
319     WITH (
320         FORMAT = 'CSV',
321         FIRSTROW = 2,
322         FIELDTERMINATOR = ',',
323         ROWTERMINATOR = '\n'
324     )
325
326
327 select * from FailedTransactionErrorType;
328
329 --Bulk date insertion 8
330
331 BULK INSERT Employee
332     FROM 'C:\Users\ruzdomain\Desktop\SQL\PROJECT\Employee.csv'
333     WITH (
334         FORMAT = 'CSV',
335         FIRSTROW = 2,
336         FIELDTERMINATOR = ',',
337         ROWTERMINATOR = '\n'
338     )
339
340
341 select * from Employee;
342
343 --Bulk date insertion 9
344
345 BULK INSERT Customer
346     FROM 'C:\Users\ruzdomain\Desktop\SQL\PROJECT\Customer.csv'
347     WITH (
348         FORMAT = 'CSV',
349         FIRSTROW = 2,
350         FIELDTERMINATOR = ',',
351         ROWTERMINATOR = '\n'
352     )
353
354 select * from Customer;
355
356
357 --Bulk date insertion 10
358
359
360
361 BULK INSERT Account
362     FROM 'C:\Users\ruzdomain\Desktop\SQL\PROJECT\Account.csv'
363     WITH (
364         FORMAT = 'CSV',
```

```
365     FIRSTROW = 2,
366     FIELDTERMINATOR = ',',
367     ROWTERMINATOR = '\n'
368 )
369
370 select * from Account;
371
372
373 --Bulk date insertion 11
374
375 BULK INSERT LoginAccount
376   FROM 'C:\Users\ruzdomain\Desktop\SQL\PROJECT>LoginAccount.csv'
377   WITH (
378     FORMAT = 'CSV',
379     FIRSTROW = 2,
380     FIELDTERMINATOR = ',',
381     ROWTERMINATOR = '\n'
382 )
383
384
385
386 select * from LoginAccount;
387
388 --Bulk date insertion 12
389
390 BULK INSERT UserSecurityAnswers
391   FROM 'C:\Users\ruzdomain\Desktop\SQL\PROJECT\UserSecurityAnswers.csv'
392   WITH (
393     FORMAT = 'CSV',
394     FIRSTROW = 2,
395     FIELDTERMINATOR = ',',
396     ROWTERMINATOR = '\n'
397 )
398
399
400
401 select * from UserSecurityAnswers;
402
403 --Bulk date insertion 3
404
405 BULK INSERT CustomerAccount
406   FROM 'C:\Users\ruzdomain\Desktop\SQL\PROJECT\CustomerAccount.csv'
407   WITH (
408     FORMAT = 'CSV',
409     FIRSTROW = 2,
410     FIELDTERMINATOR = ',',
411     ROWTERMINATOR = '\n'
412 )
413
414
415
416 select * from CustomerAccount;
```

```
417  
418 --Bulk date insertion 14  
419  
420 BULK INSERT TransactionLog  
421     FROM 'C:\Users\ruzdomain\Desktop\SQL\PROJECT\TransactionLog.csv'  
422     WITH (  
423         FORMAT = 'CSV',  
424         FIRSTROW = 2,  
425         FIELDTERMINATOR = ',',  
426         ROWTERMINATOR = '\n'  
427     )  
428  
429  
430 select * from TransactionLog;  
431  
432  
433  
434  
435 Insert into LoginErrorLog Values  
436     ('68666001','2021-05-08 10:10:10','<?xml version="1.0" encoding="utf-8"?>  
437         <Error>  
438             <UserLoginID>711</UserLoginID>  
439             <Code>68666001</Code>  
440             <Description>Insufficient Funds </Description>  
441             <Message> You do not have sufficient funds in your account. </Message>  
442         </Error>'),  
443     ('68666002','2021-05-11 11:10:10','<?xml version="1.0" encoding="utf-8"?>  
444         <Error>  
445             <UserLoginID>733</UserLoginID>  
446             <Code>68666002</Code>  
447             <Description>Account Frozen.</Description>  
448             <Message> Account Freezed Due to suspicious transaction. </Message>  
449         </Error>'),  
450  
451     ('68666003','2021-05-19 13:10:10','<?xml version="1.0" encoding="utf-8"?>  
452         <Error>  
453             <UserLoginID>719</UserLoginID>  
454             <Code>68666003</Code>  
455             <Description>Account Inactive.</Description>  
456             <Message>Account Inactive due to lack of any transaction for log time. </Message>  
457         </Error>'),  
458     ('68666004','2021-06-03 14:10:10','<?xml version="1.0" encoding="utf-8"?>  
459         <Error>  
460             <UserLoginID>725</UserLoginID>  
461             <Code>68666004</Code>  
462             <Description>Account Reactivation In Progress.</Description>  
463             <Message>Transaction cannot be done unless the account is fully reactivated. </Message>  
464         </Error>'),  
465     ('68666005','2021-06-04 15:10:10','<?xml version="1.0" encoding="utf-8"?>  
466         <Error>  
467             <UserLoginID>749</UserLoginID>  
468             <Code>68666005</Code>
```

```
469 <Description>Account Temporary Halted.</Description>
470 <Message>Account halted being instructed by the customer. </Message>
471 </Error>');
472
473 select * from LoginErrorLog;
474
475
476
477
478 Insert into OverDraftLog Values
479     (110072,'2021-06-11 10:10:10', 300, '<?xml version="1.0" encoding="utf-8"?>
480         <Overdraft>
481             <CustomerID>2340047</CustomerID>
482             <Amount>300</Amount>
483             <Description>Overdraft Took Place</Description>
484             <Message> Overdraft has a fee associated with it. Please be notified. </Message>
485         </Overdraft>'),
486     (110073,'2021-06-11 11:10:10',200,'<?xml version="1.0" encoding="utf-8"?>
487         <Overdraft>
488             <CustomerID>2340048</CustomerID>
489             <Amount>200</Amount>
490             <Description>Overdraft Took Place</Description>
491             <Message> Overdraft has a fee associated with it. Please be notified. </Message>
492         </Overdraft>'),
493
494     (110074,'2021-06-11 13:10:10',700,'<?xml version="1.0" encoding="utf-8"?>
495         <Overdraft>
496             <CustomerID>2340049</CustomerID>
497             <Amount>700</Amount>
498             <Description>Overdraft Took Place</Description>
499             <Message>Overdraft has a fee associated with it. Please be notified. </Message>
500         </Overdraft>'),
501     (110075,'2021-06-11 14:10:10',1500,'<?xml version="1.0" encoding="utf-8"?>
502         <Overdraft>
503             <CustomerID>2340050</CustomerID>
504             <Amount>1500</Amount>
505             <Description>Overdraft Took Place</Description>
506             <Message>Overdraft has a fee associated with it. Please be notified. </Message>
507         </Overdraft>');
508
509 select * from OverDraftLog;
510
511
512
513
514 Insert into FailedTransactionLog Values
515     (57770076,101, '2021-06-03 10:10:10', '<?xml version="1.0" encoding="utf-8"?>
516         <FailedTransaction>
517             <CustomerID>2340011</CustomerID>
518             <Description>Insufficient Balance</Description>
519             <Message>Insufficient Balance Causes the transaction to fail.</Message>
520         </FailedTransaction>'),
```

```
521 (57770077,102,'2021-06-07 11:10:10','<?xml version="1.0" encoding="utf-8"?>
522 <FailedTransaction>
523   <CustomerID>2340017</CustomerID>
524   <Description>Limit Exceeded</Description>
525   <Message>Transaction fails as the amount exceeds the limit set by customer instruction.</Message>
526 </FailedTransaction>');
527
528 (57770078,102,'2021-06-09 13:10:10','<?xml version="1.0" encoding="utf-8"?>
529 <FailedTransaction>
530   <CustomerID>2340029</CustomerID>
531   <Description>Limit Exceeded</Description>
532   <Message>Transaction fails as the amount exceeds the limit set by customer instruction.</Message>
533 </FailedTransaction>'),
534 (57770079,101,'2021-06-13 14:10:10','<?xml version="1.0" encoding="utf-8"?>
535 <FailedTransaction>
536   <CustomerID>2340039</CustomerID>
537   <Description>Insufficient Balance</Description>
538   <Message>Insufficient Balance Causes the transaction to fail.</Message>
539 </FailedTransaction>');
540
```



File Query Project Tools Window Help

New Query Execute

master

Object Explorer

Connect

- AdiyatPC (SQL Server 15.0.2000 - ADIVATPC\ruzdomain)
- LenovoWin10 (SQL Server 15.0.2000 - LENOVOWIN10\ruzdomain)
- DINARTRIOSUEDUCA\SQLEXPRESS (SQL Server 15.0.2000)
 - Databases
 - System Databases
 - Database Snapshots
 - dbbank
 - dbFDAS
 - dbmetrocollege
 - dbprojectbank
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.Account
 - dbo.AccountStatusType
 - dbo.AccountType
 - dbo.Customer
 - dbo.CustomerAccount
 - dbo.Employee
 - dbo.FailedTransactionErrorType
 - dbo.FailedTransactionLog
 - dbo.LoginAccount
 - dbo.LoginErrorLog
 - dbo.OverDraftLog
 - dbo.SavingsInterestRates
 - dbo.TransactionLog
 - dbo.TransactionType
 - dbo.UserLogins
 - dbo.UserSecurityAnswers
 - dbo.UserSecurityQuestions
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Service Broker
 - Storage
 - Security
 - dbstor
 - dbtest1
 - dbtest2
 - dbtest22
 - dbtest3
 - dvdrental

```
1 create database dbprojectbank;
2 use dbprojectbank;
--Tables in the project are:
3 --01. UserLogins
4 --02. UserSecurityQuestions
5 --03. SavingsInterestRates
6 --04. TransactionType
7 --05. AccountType
8 --06. AccountStatusType
9 --07. LoginErrorLog
10 --08. FailedTransactionErrorType
11 --09. Employee
12 --10. Customer
13 --11. Account
14 --12. OverDraftLog
15 --13. LoginAccount
16 --14. UserSecurityAnswers
17 --15. CustomerAccount
18 --16. TransactionLog
19 --17. FailedTransactionLog
20
21
22
23
24
25 -- Table count 1
26
27 create table UserLogins
28 (
29     UserLoginID smallint primary key not null ,
30     UserName char(15) not null,
31     UserPassword varchar(20) not null
32 );
```

150 %

Messages

Query executed successfully.

Parents Tables are taken care of before the child table





Object Explorer

Connect ▾

- DB DINARTRIOSEDUCA\SQLEXPRESS (SQL Server 15.0.2000 - DINARTRIOSEDUCA\ruzdomain (58))
 - Databases
 - + System Databases
 - + Database Snapshots
 - + dbbank
 - + dbFDAS
 - + dbmetrocollege
 - + **dbprojectbank**
 - + Database Diagrams
 - Tables
 - + System Tables
 - + FileTables
 - + External Tables
 - + Graph Tables
 - + dbo.Account
 - + dbo.AccountStatusType
 - + dbo.AccountType
 - + dbo.Customer
 - + dbo.CustomerAccount
 - + dbo.Employee
 - + dbo.FailedTransactionErrorType
 - + dbo.FailedTransactionLog
 - + dbo.LoginAccount
 - + dbo.LoginErrorLog
 - + dbo.OverDraftLog
 - + dbo.SavingsInterestRates
 - + dbo.TransactionLog
 - + dbo.TransactionType
 - + dbo.UserLogins
 - + dbo.UserSecurityAnswers
 - + dbo.UserSecurityQuestions
 - + Views
 - + External Resources
 - + Synonyms
 - + Programmability
 - + Service Broker
 - + Storage
 - + Security
 - + dbstor
 - + dbtest1
 - + dbtest2
 - + dbtest22
 - + dbtest3
 - + dvdrental
 - + metro_uni
 - + mydb

SQLQuery4.sql - DL...CA\ruzdomain (58) ▾ X L5-join andsql...UCA\ruzdomain (56)

MohammadMonjurEE...\ruzdomain (55)

MohammadMonjurEE...\ruzdomain (53)*

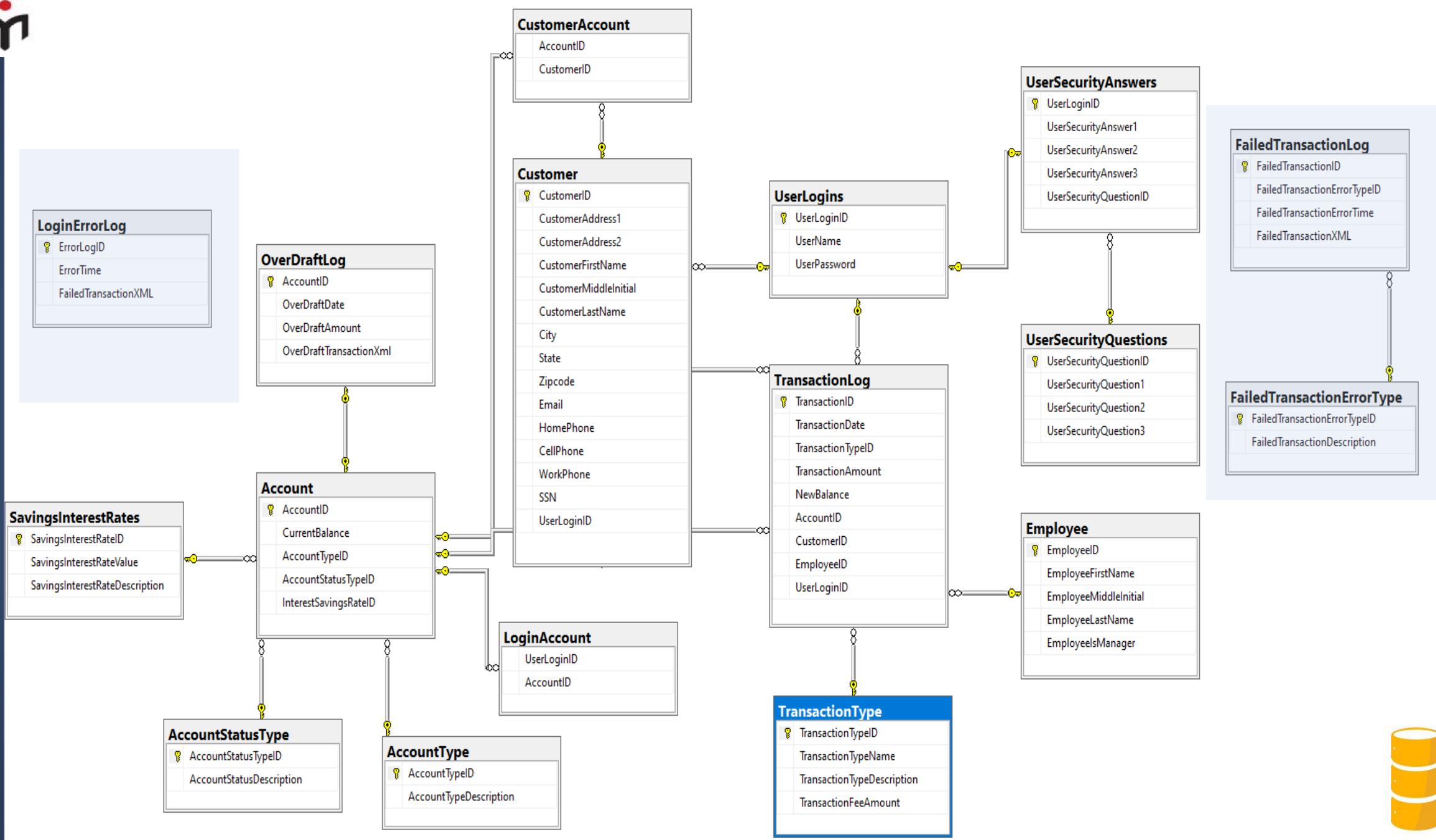
L6-Stored procedu...CA\ruzdomain (54)

1

165 %
Connected. (1/1)

DINARTRIOSEDUCA\SQLEXPRESS ... | DINARTRIOSEDUCA\ruzdom... | dbprc

Database Diagrams



Data Insertion Methods

SOLQuerv4.sal - DINARTRIOSUEDUCA\SQLEXPRESS.master (DINARTRIOSUEDUCA\ruzdomain (58))* - Microsoft SQL Server Management Studio

File Query Project Tools Window Help

Object Explorer

Connect New Query Execute

LENOVOWIN10\SQLE..bank - Diagram_0* SQLQuery4.sql - DI...CA\ruzdomain (58)* L5-join andsql...UCA\ruzdomain (56) MohammadMonjurEE..\ruzdomain (55) MohammadMonjurEE..\ruzdomain (53)*

343 --Bulk data insertion
344
345 BULK INSERT Customer
346 FROM 'C:\Users\ruzdomain\Desktop\SQL\PROJECT\Customer.csv'
347 WITH (
348 FORMAT = 'CSV',
349 FIRSTROW = 2,
350 FIELDTERMINATOR = ',',
351 ROWTERMINATOR = '\n'
352)
353
354 select * from Customer;
355
356
357 --Bulk data insertion
358
359
360
361 BULK INSERT Account
362 FROM 'C:\Users\ruzdomain\Desktop\SQL\PROJECT\Account.csv'
363 WITH (
364 FORMAT = 'CSV',
365 FIRSTROW = 2,
366 FIELDTERMINATOR = ',',
367 ROWTERMINATOR = '\n'
368)
369
370 select * from Account;
371

14 out of 17 tables
are created by bulk
insert using .csv files

165 %

Messages

Query executed successfully.

DINARTRIOSUEDUCA\SQLEXPRESS ... | DINARTRIOSUEDUCA\ruzdom...

Ln 8 Col 18 Ch 18 INS

Ready





Object Explorer

Connect ▾

- AdiyatPC (SQL Server 15.0.2000 - ADIYATPC\ruzdomain)
- LenovoWin10 (SQL Server 15.0.2000 - LENOVOWIN10\ruzdomain)
- DINARTRIOSEDUCA\SQLEXPRESS (SQL Server 15.0.2000)
 - Databases
 - System Databases
 - Database Snapshots
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.Account
 - dbo.AccountStatusType
 - dbo.AccountType
 - dbo.Customer
 - dbo.CustomerAccount
 - dbo.Employee
 - dbo.FailedTransactionErrorType
 - dbo.FailedTransactionLog
 - dbo.LoginAccount
 - dbo.LoginErrorLog
 - dbo.OverDraftLog
 - dbo.SavingsInterestRates
 - dbo.TransactionLog
 - dbo.TransactionType
 - dbo.UserLogins
 - dbo.UserSecurityAnswers
 - dbo.UserSecurityQuestions
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Service Broker
 - Storage
 - Security
- dbstor
- dbtest1
- dbtest2
- dbtest22
- dbtest3
- dvdrental

478 Insert into OverDraftLog Values
479 (110072, '2021-06-11 10:10:10', 300, '<?xml version="1.0" encoding="utf-8"?>
480 <Overdraft>
481 <CustomerID>2340047</CustomerID>
482 <Amount>300</Amount>
483 <Description>Overdraft Took Place</Description>
484 <Message> Overdraft has a fee associated with it. Please be notified. </Message>
485 </Overdraft>'),
486 (110073, '2021-06-11 11:10:10', 200, '<?xml version="1.0" encoding="utf-8"?>
487 <Overdraft>
488 <CustomerID>2340048</CustomerID>
489 <Amount>200</Amount>
490 <Description>Overdraft Took Place</Description>
491 <Message> Overdraft has a fee associated with it. Please be notified. </Message>
492 </Overdraft>'),
493 (110074, '2021-06-11 13:10:10', 700, '<?xml version="1.0" encoding="utf-8"?>
494 <Overdraft>
495 <CustomerID>2340049</CustomerID>
496 <Amount>700</Amount>
497 <Description>Overdraft Took Place</Description>
498 <Message>Overdraft has a fee associated with it. Please be notified. </Message>
499 </Overdraft>'),
500 (110075, '2021-06-11 14:10:10', 1500, '<?xml version="1.0" encoding="utf-8"?>
501 <Overdraft>
502 <CustomerID>2340050</CustomerID>
503 <Amount>1500</Amount>
504 <Description>Overdraft Took Place</Description>
505 <Message>Overdraft has a fee associated with it. Please be notified. </Message>
506 </Overdraft>');
507
508
509 select * from OverDraftLog;

3 out of 17 tables
are created by
Insert Into
command

Messages
Query executed successfully.





Number of observations of the tables

ORDER	TABLE NAME	ROWS INSERTED
1	UserLogins	50
2	UserSecurityQuestions	05
3	SavingsInterestRates	04
4	TransactionType	06
5	AccountType	02
6	AccountStatusType	06
7	LoginErrorLog	05
8	FailedTransactionErrorType	02
9	Employee	05
10	Customer	50
11	Account	75
12	OverDraftLog	04
13	LoginAccount	75
14	UserSecurityAnswers	50
15	CustomerAccount	75
16	TransactionLog	75
17	FailedTransactionLog	04





Phase 02

01. Create a view to get all customers with checking account from ON province. [Moderate]
02. Create a view to get all customers with total account balance (including interest rate) greater than 5000. [Advanced]
03. Create a view to get counts of checking and savings accounts by customer. [Moderate]
04. Create a view to get any particular user's login and password using AccountId. [Moderate]
05. Create a view to get all customers' overdraft amount. [Moderate]
06. Create a stored procedure to add "User_" as a prefix to everyone's login (username). [Moderate]
07. Create a stored procedure that accepts AccountId as a parameter and returns customer's full name. [Advanced]
08. Create a stored procedure that returns error logs inserted in the last 24 hours. [Advanced]
09. Create a stored procedure that takes a deposit as a parameter and updates CurrentBalance value for that particular account. [Advanced]
10. Create a stored procedure that takes a withdrawal amount as a parameter and updates





Phase 02: Q01

**Create a view to get all customers with
checking account from ON province.
[Moderate]**





Object Explorer

Connect ▾

- Databases
 - System Databases
 - Database Snapshots
 - dbbank
 - dbFDAS
 - dbmetrocollege
 - dbprojectbank
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.Account
 - dbo.AccountStatusType
 - dbo.AccountType
 - dbo.Customer
 - dbo.CustomerAccount
 - dbo.Employee
 - dbo.FailedTransactionErrorType
 - dbo.FailedTransactionLog
 - dbo.LoginAccount
 - dbo.LoginErrorLog
 - dbo.OverDraftLog
 - dbo.SavingsInterestRates
 - dbo.TransactionLog
 - dbo.TransactionType
 - dbo.UserLogins
 - dbo.UserSecurityAnswers
 - dbo.UserSecurityQuestions
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Service Broker
 - Storage
 - Security
 - dbstor
 - dbtest1
 - dbtest2
 - dbtest22
 - dbtest3
 - dvdrental
 - metro_uni
 - mydb
 - Security

SQLQuery4.sql - DI...CA\ruzdomain (58) MohammadMonjurEE...\ruzdomain (55)* X LENOVOWIN10\SQLEx...bank - Diagram_0* MohammadMonjurEE...\ruzdomain (53)*

```
1 use dbprojectbank;
2 GO
3 -- Q1: Create a view to get all customers with checking account from ON province. [Moderate]
4
5 IF EXISTS (SELECT * FROM sys.all_views WHERE NAME='VW_ON_ChekkingAccount')
6 DROP VIEW VW_ON_ChekkingAccount;
7 GO
8 create view VW_ON_ChekkingAccount
9     as
10    select C.CustomerID,C.CustomerFirstName+' '+C.CustomerLastName [Customer Name],C.State, AT.AccountTypeDescription
11    from Customer C
12    join CustomerAccount CA
13        on C.CustomerID=CA.CustomerID
14    join Account A
15        on CA.AccountID=A.AccountID
16    join AccountType AT
17        on A.AccountTypeID=AT.AccountTypeID
18    where AT.AccountTypeDescription='Checking Account' and C.State='ON';
19 GO
20
21 select * from VW_ON_ChekkingAccount;
22 GO
23 GO
```

150 % Messages

Commands completed successfully.

Completion time: 2021-09-02T00:09:00.3043853-04:00

150 % Query executed successfully.

DINARTRIOSEDUCA\SQLEXPRESS ... | DINARTRIOSEDUCA\ruzdom... | dbpr...

Ln 23

Col 3

Ch 3

INS





dbprojectbank

Execute

Object Explorer

Databases

- System Databases
- Database Snapshots
- dbbank
- dbFDAS
- dbmetrocollege
- dbprojectbank
- Database Diagrams
- Tables
- System Tables
- FileTables
- External Tables
- Graph Tables
- dbo.Account
- dbo.AccountStatusType
- dbo.AccountType
- dbo.Customer
- dbo.CustomerAccount
- dbo.Employee
- dbo.FailedTransactionErrorType
- dbo.FailedTransactionLog
- dbo.LoginAccount
- dbo.LoginErrorLog
- dbo.OverDraftLog
- dbo.SavingsInterestRates
- dbo.TransactionLog
- dbo.TransactionType
- dbo.UserLogins
- dbo.UserSecurityAnswers
- dbo.UserSecurityQuestions

Views

- External Resources
- Synonyms
- Programmability
- Service Broker
- Storage
- Security

dbstor

dbtest1

dbtest2

dbtest22

dbtest3

dvdrental

metro_uni

mydb

Security

SQLQuery4.sql - DI...CA\ruzdomain (58) MohammadMonjurEE...\\ruzdomain (55)* X LENOVOWIN10\SQL...bank - Diagram_0* MohammadMonjurEE...\\ruzdomain (53)*

```
1 use dbprojectbank;
2 GO
3 -- Q1: Create a view to get all customers with checking account from ON province. [Moderate]
4
5 IF EXISTS (SELECT * FROM sys.all_views WHERE NAME='VW_ON_ChekkingAccount')
6 DROP VIEW VW_ON_ChekkingAccount;
7 GO
8 create view VW_ON_ChekkingAccount
9 as
10 select C.CustomerID,C.CustomerFirstName+' '+C.CustomerLastName [Customer Name],C.State, AT.AccountTypeDescription
11 from Customer C
12 join CustomerAccount CA
13 on C.CustomerID=CA.CustomerID
14 join Account A
15 on CA.AccountID=A.AccountID
16 join AccountType AT
17 on A.AccountTypeID=AT.AccountTypeID
18 where AT.AccountTypeDescription='Checking Account' and C.State='ON';
19 GO
20
21 select * from VW_ON_ChekkingAccount;
22 GO
23
```

150 %

Messages

Commands completed successfully.

Completion time: 2021-09-02T00:10:48.9967120-04:00

Query executed successfully.

DINARTRIOSEDUCA\SQLEXPRESS ... | DINARTRIOSEDUCA\ruzdom... | dbpr...





Object Explorer

Connect ▾

- Databases
 - System Databases
 - Database Snapshots
 - dbbank
 - dbFDAS
 - dbmetrocollege
 - dbprojectbank
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.Account
 - dbo.AccountStatusType
 - dbo.AccountType
 - dbo.Customer
 - dbo.CustomerAccount
 - dbo.Employee
 - dbo.FailedTransactionErrorType
 - dbo.FailedTransactionLog
 - dbo.LoginAccount
 - dbo.LoginErrorLog
 - dbo.OverDraftLog
 - dbo.SavingsInterestRates
 - dbo.TransactionLog
 - dbo.TransactionType
 - dbo.UserLogins
 - dbo.UserSecurityAnswers
 - dbo.UserSecurityQuestions
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Service Broker
 - Storage
 - Security
 - dbstor
 - dbtest1
 - dbtest2
 - dbtest22
 - dbtest3
 - dvdrental
 - metro_uni
 - mydb
 - Security

SQLQuery4.sql - DI...CA\ruzdomain (58)

MohammadMonjurEE...\\ruzdomain (55)*

LENOVOWIN10\SQLEXPRESS...bank - Diagram_0*

MohammadMonjurEE...\\ruzdomain (53)*

```
7 GO
8 create view VW_ON_ChekkingAccount
9   as
10  select C.CustomerID,C.CustomerFirstName+' '+C.CustomerLastName [Customer Name],C.State, AT.AccountTypeDescription
11  from Customer C
12    join CustomerAccount CA
13      on C.CustomerID=CA.CustomerID
14    join Account A
15      on CA.AccountID=A.AccountID
16    join AccountType AT
17      on A.AccountTypeID=AT.AccountTypeID
18  where AT.AccountTypeDescription='Checking Account' and C.State='ON';
19 GO
20
21 select * from VW_ON_ChekkingAccount;
```

150 %

CustomerID	Customer Name	State	AccountTypeDescription
1	Anderson Simard	ON	Checking Account
2	Beaulieu Thomas	ON	Checking Account
3	Belanger Thompson	ON	Checking Account
4	Campbell Williams	ON	Checking Account
5	Chen Zhang	ON	Checking Account
6	Fortin Chan	ON	Checking Account
7	Gagnon-á Lee	ON	Checking Account
8	Gauthier Jones	ON	Checking Account
9	Gill Kelly	ON	Checking Account
10	Girard Reid	ON	Checking Account
11	Grant-á Roy	ON	Checking Account
12	Hall Scott	ON	Checking Account
13	Morrison Caron-á	ON	Checking Account
14	Murphy Clark-á	ON	Checking Account
15	Murray Clarke	ON	Checking Account
16	Nadeau Cloutier	ON	Checking Account
17	Richard Taylor	ON	Checking Account

Query executed successfully.

DINARTRIOSEDUCA\SQLEXPRESS ... | DINARTRIOSEDUCA\ruzdom... | dbproj

Ln 21

Col 1

Ch 1

INS





Phase 02: Q02

Create a view to get all customers with total account balance (including interest rate) greater than 5000. [Advanced]





Object Explorer Execute New Query Save Print Copy Paste Find Replace Refresh

dbprojectbank SQLQuery4.sql - DI...CA\ruzdomain (58) MohammadMonjurEE...\ruzdomain (55)* LENOVOWIN10\SQLExpress - Diagram_0* MohammadMonjurEE...\ruzdomain (53)*

```
23 GO
24 --Q2: Create a view to get all customers with total account balance (including interest rate) greater than 5000. [Advanced]
25 --Here we just focus on the Savings account Balance.
26 GO
27
28 IF EXISTS (SELECT * FROM sys.all_views WHERE NAME='VW_Customer_Balance_TTLQ')
29 DROP VIEW VW_Customer_Balance_TTLQ;
30 GO
31 CREATE VIEW VW_Customer_Balance_TTLQ AS
32
33 SELECT C.CustomerFirstName + ' ' + C.CustomerLastName AS [Customer Name], A.AccountID, A.CurrentBalance AS Account_Balance,
34 S.SavingsInterestRateValue,
35 cast((A.CurrentBalance + A.CurrentBalance*(S.SavingsInterestRateValue/12)*Datediff(MM,TL.TransactionDate, GETDATE())) as decimal(10,2))
36 AS Total_Account_Balance
37
38 FROM
39 SavingsInterestRates S
40 join Account A
41 on A.InterestSavingsRateID = S.SavingsInterestRateID
42 join TransactionLog TL
43 on TL.AccountID = A.AccountID
44 join Customer C
45 on C.CustomerID = TL.CustomerID
46 where (A.CurrentBalance + A.CurrentBalance*(S.SavingsInterestRateValue/12)*Datediff(MM,TL.TransactionDate, GETDATE())) > 5000
47
48 GO
```

150 %

Messages

Commands completed successfully.

Completion time: 2021-09-02T02:38:35.8744545-04:00

150 %

Query executed successfully.





Object Explorer

Connect ▾

- Databases
 - System Datal
 - Database Sna
 - dbbank
 - dbFDAS
 - dbmetrocoll
 - dbprojectbar
 - Database
 - Tables
 - System
 - FileTa
 - Exterr
 - Graph
 - dbo.A
 - dbo.A
 - dbo.A
 - dbo.C
 - dbo.C
 - dbo.E
 - dbo.F
 - dbo.F
 - dbo.L
 - dbo.L
 - dbo.C
 - dbo.S
 - dbo.T
 - dbo.T
 - dbo.U
 - dbo.U
 - Views
 - External F
 - Synonym
 - Program
 - Service Bi
 - Storage
 - Security
 - dbstor
 - dbtest1
 - dbtest2
 - dbtest22
 - dbtest3
 - dvdrental
 - metro_uni
 - mydb
 - Security

```
40      join Account A
41          on A.InterestSavingsRateID = S.SavingsInterestRateID
42          join TransactionLog TL
43              on TL.AccountID = A.AccountID
44          join Customer C
45              on C.CustomerID = TL.CustomerID
46          where (A.CurrentBalance + A.CurrentBalance*(S.SavingsInterestRateValue/12)*Datediff(MM,TL.TransactionDate, GETDATE())) > 5000
47      ;
48      GO
49
50      select * from VW_Customer_Balance_TTLQ;
51      GO
52
```

150 %

Results Messages

	Customer Name	AccountID	Account_Balance	SavingsInterestRateValue	Total_Account_Balance
1	Cote Allen	100008	5000	0.040000000	5066.67
2	Davis Bell	100009	5100	0.060000000	5202.00
3	Evans Brown	100010	7500	0.060000000	7650.00
4	Fortin Chan	100011	9000	0.060000000	9180.00
5	Gagnon- Lee	100012	10000	0.060000000	10200.00
6	Gill Kelly	100014	11000	0.100000000	11366.67
7	Graham Ross	100016	19000	0.100000000	19633.33
8	Grant- Roy	100017	6600	0.060000000	6732.00
9	Hall Scott	100018	9910	0.060000000	10108.20
10	Haris Singh	100019	10001	0.100000000	10334.37
11	Johnson Wong	100021	45000	0.100000000	46500.00
12	Johnston Young	100022	100000	0.100000000	103333.33
13	Kennedy Desjardins	100023	108000	0.100000000	116000.00
14	Khan Fraser	100024	17250	0.100000000	17825.00
15	King Gagne	100025	28000	0.100000000	28933.33

Query executed successfully.

DINARTRIOSEDUCA\SQLEXPRESS ... DINARTRIOSEDUCA\ruzdom... dbproj







Phase 02: Q03

Create a view to get counts of checking and savings accounts by customer. [Moderate]





dbprojectbank

SQLQuery4.sql - DI...CA\ruzdomain (58)

MohammadMonjurEE...\ruzdomain (55) X

LENOVOWIN10\SQLE...bank - Diagram_0*

X

MohammadMonjurEE...\ruzdomain (53))*

Object Explorer

- Connect
- Databases
 - System Datal
 - Database Sna
 - dbbank
 - dbFDAS
 - dbmetrocoll
 - dbprojectbar
 - Database
 - Tables
 - Customer
 - FileTa
 - Extern
 - Graph
 - dbo.A
 - dbo.A
 - dbo.A
 - dbo.C
 - dbo.C
 - dbo.E
 - dbo.F
 - dbo.F
 - dbo.L
 - dbo.L
 - dbo.C
 - dbo.S
 - dbo.T
 - dbo.T
 - dbo.U
 - dbo.U
 - dbo.U
 - Views
 - External F
 - Synonym
 - Program
 - Service Bi
 - Storage
 - Security
 - dbstor
 - dbtest1
 - dbtest2
 - dbtest22
 - dbtest3
 - dvdrental
 - metro_uni
 - mydb
 - Security

```
54 --Q3: Create a view to get counts of checking and savings accounts by customer. [Moderate]
55
56 IF EXISTS (SELECT * FROM sys.all_views WHERE NAME='VW_Counts_Account')
57 DROP VIEW VW_Counts_Account;
58 GO
59
60 create view VW_Counts_Account
61   as
62     select C.CustomerFirstName+' '+C.CustomerLastName [Customer Name], AT.AccountTypeDescription, count(*) [Number of Account]
63     from Customer C
64       join CustomerAccount CA
65         on C.CustomerID=CA.CustomerID
66         join Account A
67           on CA.AccountID=A.AccountID
68           join AccountType AT
69             on A.AccountTypeID=AT.AccountTypeID
70               group by C.CustomerFirstName+' '+C.CustomerLastName, AT.AccountTypeDescription;
71
72 GO
73 select * from VW_Counts_Account
74   order by [Customer Name];
75
76 GO
```

150 %

Messages

Commands completed successfully.

Completion time: 2021-09-02T02:41:42.0087639-04:00

150 %

Query executed successfully.

DINARTRIOSEDUCA\SQLEXPRESS ... | DINARTRIOSEDUCA\ruzdom... | dbprc

Ln 70

Col 84

Ch 81

INS





Object Explorer

- Connect
- Databases
 - System Datal
 - Database Sna
 - dbbank
 - dbFDAS
 - dbmetrocoll
 - dbprojectbar
 - Database
 - Tables
 - System
 - FileTa
 - Extern
 - Graph
 - dbo.A
 - dbo.A
 - dbo.A
 - dbo.C
 - dbo.C
 - dbo.E
 - dbo.F
 - dbo.F
 - dbo.L
 - dbo.L
 - dbo.C
 - dbo.S
 - dbo.T
 - dbo.T
 - dbo.U
 - dbo.U
 - Views
 - External F
 - Synonym
 - Program
 - Service Bi
 - Storage
 - Security
 - dbostor
 - dbtest1
 - dbtest2
 - dbtest22
 - dbtest3
 - dvdrental
 - metro_uni
 - mydb
 - Security

```
60  create view VW_Counts_Account
61  as
62  select C.CustomerFirstName+' '+C.CustomerLastName [Customer Name], AT.AccountTypeDescription, count(*) [Number of Account]
63  from Customer C
64  join CustomerAccount CA
65  on C.CustomerID=CA.CustomerID
66  join Account A
67  on CA.AccountID=A.AccountID
68  join AccountType AT
69  on A.AccountTypeID=AT.AccountTypeID
70  group by C.CustomerFirstName+' '+C.CustomerLastName, AT.AccountTypeDescription;
71
72 GO
73 select * from VW_Counts_Account
74 order by [Customer Name];
75 GO
```

150 %

Results Messages

Customer Name	AccountTypeDescription	Number of Account
1 Anderson Simard	Checking Account	1
2 Anderson Simard	Savings Account	1
3 Beaulieu Thomas	Checking Account	1
4 Beaulieu Thomas	Savings Account	1
5 Belanger Thompson	Checking Account	1
6 Belanger Thompson	Savings Account	1
7 Bouchard Tremblay	Checking Account	1
8 Bouchard Tremblay	Savings Account	1
9 Cameron Wilson-á	Checking Account	1
10 Cameron Wilson-á	Savings Account	1
11 Campbell Williams	Checking Account	1
12 Campbell Williams	Savings Account	1
13 Chen Zhang	Checking Account	1
14 Chen Zhang	Savings Account	1
15 Cote Allen	Checking Account	1
16 Cote Allen	Savings Account	1
17 Davis Bell	Checking Account	1
18 Davis Bell	Savings Account	1
19 Evans Brown	Checking Account	1
20 Evans Brown	Savings Account	1

Query executed successfully.

DINARTRIOSEDUCA\SQLEXPRESS ... DINARTRIOSEDUCA\ruzdom... dbproj





dbprojectbank

MohammadMonjurEE..\\ruzdomain (55)*

MohammadMonjurEE..\\ruzdomain (53)*

L6-Stored procedu...CA\\ruzdomain (54)

	Customer Name	Account Type Description	Number of Account
1	Anderson Simard	Checking Account	1
2	Anderson Simard	Savings Account	1
3	Beaulieu Thomas	Checking Account	1
4	Beaulieu Thomas	Savings Account	1
5	Belanger Thompson	Checking Account	1
6	Belanger Thompson	Savings Account	1
7	Bouchard Tremblay	Checking Account	1
8	Bouchard Tremblay	Savings Account	1
9	Cameron Wilson-á	Checking Account	1
10	Cameron Wilson-á	Savings Account	1
11	Campbell Williams	Checking Account	1
12	Campbell Williams	Savings Account	1
13	Chen Zhang	Checking Account	1
14	Chen Zhang	Savings Account	1
15	Cote Allen	Checking Account	1
16	Cote Allen	Savings Account	1
17	Davis Bell	Checking Account	1
18	Davis Bell	Savings Account	1
19	Evans Brown	Checking Account	1
20	Evans Brown	Savings Account	1
21	Fortin Chan	Checking Account	1
22	Fortin Chan	Savings Account	1
23	Gagnon-á Lee	Checking Account	1
24	Gagnon-á Lee	Savings Account	1
25	Gauthier Jones	Checking Account	1
26	Gauthier Jones	Savings Account	1
27	Gill Kelly	Checking Account	1
28	Gill Kelly	Savings Account	1
29	Girard Reid	Checking Account	1
30	Girard Reid	Savings Account	1
31	Graham Ross	Checking Account	1
32	Graham Ross	Savings Account	1
33	Grant-á Roy	Checking Account	1
34	Grant-á Roy	Savings Account	1
35	Hall Scott	Checking Account	1
36	Hall Scott	Savings Account	1
37	Hamis Singh	Checking Account	1
38	Hamis Singh	Savings Account	1
39	Jackson White	Checking Account	1
40	Jackson White	Savings Account	1
41	Johnson Wong	Checking Account	1

Query executed successfully.

DINARTRIOSEDUCA\SQLEXPRESS ... DINARTRIOSEDUCA\ruzdom... dbproj





	Customer Name	Account Type Description	Number of Account
35	Hall Scott	Checking Account	1
36	Hall Scott	Savings Account	1
37	Haris Singh	Checking Account	1
38	Haris Singh	Savings Account	1
39	Jackson White	Checking Account	1
40	Jackson White	Savings Account	1
41	Johnson Wong	Checking Account	1
42	Johnson Wong	Savings Account	1
43	Johnston Young	Checking Account	1
44	Johnston Young	Savings Account	1
45	Kennedy Desjardins	Checking Account	1
46	Kennedy Desjardins	Savings Account	1
47	Khan Fraser	Checking Account	1
48	Khan Fraser	Savings Account	1
49	King Gagne	Checking Account	1
50	King Gagne	Savings Account	1
51	Landy Green	Checking Account	1
52	Lapointe Hamilton	Checking Account	1
53	Lavoie Hill	Checking Account	1
54	Leblanc Moore	Checking Account	1
55	Levesque Poirier	Checking Account	1
56	Lewis Wang	Checking Account	1
57	Lin Watson	Checking Account	1
58	MacDonald Wood	Checking Account	1
59	Martin Adams-á	Checking Account	1
60	Michaud Bergeron	Checking Account	1
61	Mitchell Baker	Checking Account	1
62	Morin Boucher	Checking Account	1
63	Morrison Caron-á	Checking Account	1
64	Murphy Clark-á	Checking Account	1
65	Murray Clarke	Checking Account	1
66	Nadeau Cloutier	Checking Account	1
67	Nguyen Dube	Checking Account	1
68	Ouellet Fournier	Checking Account	1
69	Patel Lefebvre	Checking Account	1
70	Pelletier Miller	Checking Account	1
71	Peters Smith	Checking Account	1
72	Reynolds Stewart	Checking Account	1
73	Richard Taylor	Checking Account	1
74	Robinson Walker-á	Checking Account	1
75	Rogen Wright-á	Checking Account	1

 Query executed successfully.

| PINARTRIOSEDUCA\SOLEXPRESS ... | PINARTRIOSEDUCA\ruzdom... | dbproj



Phase 02: Q04

Create a view to get any particular user's login and password using AccountId. [Moderate]





Object Explorer

- Databases
 - System Datal
 - Database Sna
 - dbbank
 - dbFDAS
 - dbmetrocoll
 - dbprojectbar
 - Database
 - Tables
 - System
 - FileTa
 - Extern
 - Graph
 - dbo.A
 - dbo.A
 - dbo.A
 - dbo.C
 - dbo.C
 - dbo.E
 - dbo.F
 - dbo.F
 - dbo.L
 - dbo.L
 - dbo.C
 - dbo.S
 - dbo.T
 - dbo.T
 - dbo.U
 - dbo.U
 - Views
 - External F
 - Synonym
 - Program
 - Service Bi
 - Storage
 - Security
 - dbstor
 - dbtest1
 - dbtest2
 - dbtest22
 - dbtest3
 - dvdrental
 - metro_uni
 - mydb
 - Security

```
79  --Q4: Create a view to get any particular user's login and password using AccountId. [Moderate]
80
81  IF EXISTS (SELECT * FROM sys.all_views WHERE NAME='VW_Users_Login_password')
82  DROP VIEW VW_Users_Login_password;
83  GO
84  create view VW_Users_Login_password
85  as
86  select C.CustomerFirstName+' '+C.CustomerLastName [Customer Name], UL.UserName UserLoginName, UL.UserPassword UserLoginPassword
87  from Customer C
88  join CustomerAccount CA
89  on C.CustomerID=CA.CustomerID
90  join LoginAccount LA
91  on CA.AccountID=LA.AccountID
92  join UserLogins UL
93  on LA.UserLoginID=UL.UserLoginID
94  where CA.AccountID = 100025;
95
96  GO
97  select * from VW_Users_Login_password;
98
99  GO
```

150 %

Messages

Commands completed successfully.

Completion time: 2021-09-02T02:44:34.4523265-04:00

150 %

Query executed successfully.

DINARTRIOSEDUCA\SQLEXPRESS ... | DINARTRIOSEDUCA\ruzdom... | dbpr...





Object Explorer

- Databases
 - System Datal
 - Database Sna
 - dbbank
 - dbFDAS
 - dbmetrocoll
 - dbprojectbar
 - Database
 - Tables
 - System
 - FileTa
 - Extern
 - Graph
 - dbo.A
 - dbo.A
 - dbo.A
 - dbo.C
 - dbo.C
 - dbo.E
 - dbo.F
 - dbo.F
 - dbo.L
 - dbo.L
 - dbo.C
 - dbo.S
 - dbo.T
 - dbo.T
 - dbo.U
 - dbo.U
 - Views
 - External F
 - Synonym
 - Program
 - Service Bi
 - Storage
 - Security
 - dbstor
 - dbtest1
 - dbtest2
 - dbtest22
 - dbtest3
 - dvdrental
 - metro_uni
 - mydb
 - Security

```
79  --Q4: Create a view to get any particular user's login and password using AccountId. [Moderate]
80
81 IF EXISTS (SELECT * FROM sys.all_views WHERE NAME='VW_Users_Login_password')
82 DROP VIEW VW_Users_Login_password;
83 GO
84 create view VW_Users_Login_password
85   as
86     select C.CustomerFirstName+' '+C.CustomerLastName [Customer Name], UL.UserName UserLoginName, UL.UserPassword UserLoginPassword
87     from Customer C
88       join CustomerAccount CA
89         on C.CustomerID=CA.CustomerID
90       join LoginAccount LA
91         on CA.AccountID=LA.AccountID
92       join UserLogins UL
93         on LA.UserLoginID=UL.UserLoginID
94     where CA.AccountID = 100025;
95
96 GO
97 select * from VW_Users_Login_password;
98
99 GO
```

100 %

Results

	Customer Name	UserLoginName	UserLoginPassword
1	King Gagne	User_King	TopSecret147

Query executed successfully.

DINARTRIOSEDUCA\SQLEXPRESS ... | DINARTRIOSEDUCA\ruzdom... | dbpr...





Phase 02: Q05

Create a view to get all customers' overdraft amount. [Moderate]





Object Explorer

- Databases
 - System Datal
 - Database Sna
 - dbbank
 - dbFDAS
 - dbmetrocoll
 - dbprojectbar
 - Database
 - Tables
 - Customer
 - CustomerAccount
 - OverDraftLog
 - Views
 - External F
 - Synonym
 - Program
 - Service Bi
 - Storage
 - Security
 - dbo
 - dbstor
 - dbtest1
 - dbtest2
 - dbtest22
 - dbtest3
 - dvdrental
 - metro_uni
 - mydb
 - Security

```
101  
102    --Q5: Create a view to get all customers' overdraft amount. [Moderate]  
103  
104    IF EXISTS (SELECT * FROM sys.all_views WHERE NAME='VW_All_Customers_Overdraft_Amount')  
105        DROP VIEW VW_All_Customers_Overdraft_Amount;  
106        GO  
107    create view VW_All_Customers_Overdraft_Amount  
108        as  
109            select C.CustomerFirstName+' '+C.CustomerLastName [Customer Name], ODL.OverDraftAmount, ODL.OverDraftDate,  
110                ODL.OverDraftTransactionXml  
111            from Customer C  
112                join CustomerAccount CA  
113                    on C.CustomerID=CA.CustomerID  
114                join OverDraftLog ODL  
115                    on CA.AccountID=ODL.AccountID;  
116  
117  
118        GO  
119    select * from VW_All_Customers_Overdraft_Amount;  
120  
121        GO
```

150 %

Messages

Commands completed successfully.

Completion time: 2021-09-02T02:46:55.3555079-04:00

150 %

Query executed successfully.

DINARTRIOSEDUCA\SQLEXPRESS ... | DINARTRIOSEDUCA\ruzdom... | dbprc





Object Explorer

- Databases
 - System Datal
 - Database Sna
 - dbbank
 - dbFDAS
 - dbmetrocoll
 - dbprojectbar
 - Database
 - Tables
 - Customer
 - CustomerAccount
 - OverDraftLog
 - Views
 - External F
 - Synonym
 - Programm
 - Service Bi
 - Storage
 - Security
 - dbo
 - dbstor
 - dbtest1
 - dbtest2
 - dbtest22
 - dbtest3
 - dvdrental
 - metro_uni
 - mydb
 - Security

```
101  
102    --Q5: Create a view to get all customers' overdraft amount. [Moderate]  
103  
104    IF EXISTS (SELECT * FROM sys.all_views WHERE NAME='VW_All_Customers_Overdraft_Amount')  
105        DROP VIEW VW_All_Customers_Overdraft_Amount;  
106        GO  
107    create view VW_All_Customers_Overdraft_Amount  
108        as  
109            select C.CustomerFirstName+' '+C.CustomerLastName [Customer Name], ODL.OverDraftAmount, ODL.OverDraftDate,  
110                ODL.OverDraftTransactionXml  
111            from Customer C  
112            join CustomerAccount CA  
113                on C.CustomerID=CA.CustomerID  
114            join OverDraftLog ODL  
115                on CA.AccountID=ODL.AccountID;  
116  
117        GO  
118    select * from VW_All_Customers_Overdraft_Amount;  
119        GO  
120        GO  
121        GO
```

Results

	Customer Name	OverDraftAmount	OverDraftDate	OverDraftTransactionXml
1	Reynolds Stewart	300.00	2021-06-11 10:10:10.000	<Overdraft><CustomerID>2340047</CustomerID><Amou...
2	Richard Taylor	200.00	2021-06-11 11:10:10.000	<Overdraft><CustomerID>2340048</CustomerID><Amou...
3	Rogen Wright-á	700.00	2021-06-11 13:10:10.000	<Overdraft><CustomerID>2340049</CustomerID><Amou...
4	Robinson Walker-á	1500.00	2021-06-11 14:10:10.000	<Overdraft><CustomerID>2340050</CustomerID><Amou...

Query executed successfully.

DINARTRIOSEDUCA\SQLEXPRESS ... | DINARTRIOSEDUCA\ruzdom... | dbpr...





Phase 02: Q06

Create a stored procedure to add “User_” as a prefix to everyone’s login (username).

[Moderate]





File Query Project Tools Window Help



Object Explorer

Connect ▾

- AdiyatPC (SQL Server 15.0.2000 - ADIYATPC)\n
 - Databases
 - Security
 - Server Objects
 - Replication
 - PolyBase
 - Management
 - XEvent Profiler

- LenovoWin10 (SQL Server 15.0.2000 - LENOVC\n

- DINARTRIOSEDUCA\SQLEXPRESS (SQL Server

- Databases

- System Databases

- Database Snapshots

- dbbank

- dbFDAS

- dbmetrocollege

- dbprojectbank

- Database Diagrams

- Tables

- System Tables

- FileTables

- External Tables

- Graph Tables

- dbo.Account

- dbo.AccountStatusType

- dbo.AccountType

- dbo.Customer

- dbo.CustomerAccount

- dbo.Employee

- dbo.FailedTransactionErrorType

- dbo.FailedTransactionLog

- dbo.LoginAccount

- dbo.LoginErrorLog

- dbo.OverDraftLog

- dbo.SavingsInterestRates

- dbo.TransactionLog

- dbo.TransactionType

- dbo.UserLogins

- dbo.UserSecurityAnswers

- dbo.UserSecurityQuestions

- Views

- External Resources

- Synonyms

- Programmability

- Service Broker

- Storage

L5-join andsql...UCA\ruzdomain (56) MohammadMonjurEE...\ruzdomain (55)* MohammadMonjurEE...\ruzdomain (53)* L6-Stored procedu...CA\ruzdomain (54))

```
97 --Q6
98 --Before the Store Procedure is created the table
99
100 select * from UserLogins;
```

165 %

Messages

Commands completed successfully.

Completion time: 2021-09-01T00:15:52.3069203-04:00

Query executed successfully.

DINARTRIOSEDUCA\SQLEXPRESS ... DINARTRIOSEDUCA\ruzdom... dbpr...

Ln 1

Col 1

Ch 1

INS





dbprojectbank

Object Explorer

Connect ▾

- AdiyatPC (SQL Server 15.0.2000 - ADIYATPC)\n
 - Databases
 - Security
 - Server Objects
 - Replication
 - PolyBase
 - Management
 - XEvent Profiler
- LenovoWin10 (SQL Server 15.0.2000 - LENOVC\n
 - Databases
 - System Databases
 - Database Snapshots
 - dbbank
 - dbFDAS
 - dbmetrocollege
 - dbprojectbank
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.Account
 - dbo.AccountStatusType
 - dbo.AccountType
 - dbo.Customer
 - dbo.CustomerAccount
 - dbo.Employee
 - dbo.FailedTransactionErrorType
 - dbo.FailedTransactionLog
 - dbo.LoginAccount
 - dbo.LoginErrorLog
 - dbo.OverDraftLog
 - dbo.SavingsInterestRates
 - dbo.TransactionLog
 - dbo.TransactionType
 - dbo.UserLogins
 - dbo.UserSecurityAnswers
 - dbo.UserSecurityQuestions
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Service Broker
 - Storage
- DINARTRIOSEDUCA\SQLEXPRESS (SQL Server\n
 - Databases

L5-join andsql...UCA\ruzdomain (56) MohammadMonjurEE...\ruzdomain (55)* MohammadMonjurEE...\ruzdomain (53)* L6-Stored procedu...CA\ruzdomain (54))

```
97 --Q6
98 --Before the Store Procedure is created the table
99
100 select * from UserLogins;
```

165 %

Results Messages

UserLoginID	UserName	UserPassword
1	701	Anderson
2	702	Beaulieu
3	703	Belanger
4	704	Bouchard
5	705	Cameron
6	706	Campbell
7	707	Chen
8	708	Cote
9	709	Davis
10	710	Evans
11	711	Fortin
12	712	Gagnon-á
13	713	Gauthier
14	714	Gill
15	715	Girard
16	716	Graham
17	717	Grant-á
18	718	Hall
19	719	Hams
20	720	Jackson
21	721	Johnson
22	722	Johnston
23	723	Kennedy
24	724	Khan
25	725	King
26	726	Landry
27	727	Lapointe
28	728	Lavoie
29	729	Leblanc
30	730	Levesque
31	731	Lewis
32	732	Lin
33	733	MacDon...
34	734	Martin
35	735	Mitchell

Before Executing
the Store Procedure

Query executed successfully.

DINARTRIOSEDUCA\SQLEXPRESS ... DINARTRIOSEDUCA\ruzdom... dbproj

Ln 100

Col 1

Ch 1

INS





Object Explorer

Connect ▾

- Databases
 - System Datal
 - Database Sna
 - dbbank
 - dbFDAS
 - dbmetrocoll
 - dbprojectbar
 - Database
 - Tables
 - System
 - FileTa
 - Extern
 - Graph
 - dbo.A
 - dbo.A
 - dbo.A
 - dbo.C
 - dbo.C
 - dbo.E
 - dbo.F
 - dbo.F
 - dbo.L
 - dbo.L
 - dbo.C
 - dbo.S
 - dbo.T
 - dbo.T
 - dbo.U
 - dbo.U
 - Views
 - External F
 - Synonym
 - Program
 - Service Bi
 - Storage
 - Security
 - dbostor
 - dbtest1
 - dbtest2
 - dbtest22
 - dbtest3
 - dvdrental
 - metro_uni
 - mydb
 - Security

```
121 GO
122
123 --Q6: Create a stored procedure to add "User_" as a prefix to everyone's login (username). [Moderate]
124 --Before the Store Procedure is created the table
125
126 select * from UserLogins;
127
128 --Now let us create the Store Procedure
129 GO
130 IF EXISTS (SELECT * FROM sys.procedures WHERE NAME='SP_ModifyUserLogin')
131 DROP PROC SP_ModifyUserLogin
132 GO
133 CREATE PROCEDURE SP_ModifyUserLogin
134 AS
135 UPDATE UserLogins
136 SET UserName = Concat('User_', UserName);
137 GO
138
139
140 EXEC SP_ModifyUserLogin;
141 GO
```

150 %

Messages

Commands completed successfully.

Completion time: 2021-09-02T02:54:39.9101094-04:00

150 %

Query executed successfully.

DINARTRIOSEDUCA\SQLEXPRESS ... | DINARTRIOSEDUCA\ruzdom... | dbpr...





Object Explorer

Connect ▾

- Databases
 - System Datal
 - Database Sna
 - dbbank
 - dbFDAS
 - dbmetrocoll
 - dbprojectbar
 - Database
 - Tables
 - System
 - FileTa
 - Extern
 - Graph
 - dbo.A
 - dbo.A
 - dbo.A
 - dbo.C
 - dbo.C
 - dbo.E
 - dbo.F
 - dbo.F
 - dbo.L
 - dbo.L
 - dbo.C
 - dbo.S
 - dbo.T
 - dbo.T
 - dbo.U
 - dbo.U
 - Views
 - External F
 - Synonym
 - Program
 - Service Bi
 - Storage
 - Security
 - dbostor
 - dbtest1
 - dbtest2
 - dbtest22
 - dbtest3
 - dvdrental
 - metro_uni
 - mydb
 - Security

```
127 --Now let us create the Store Procedure
128 GO
129 IF EXISTS (SELECT * FROM sys.procedures WHERE NAME='SP_ModifyUserLogin')
130 DROP PROC SP_ModifyUserLogin
131 GO
132 CREATE PROCEDURE SP_ModifyUserLogin
133 AS
134 UPDATE UserLogins
135 SET UserName = Concat('User_', UserName);
136 GO
137
138 EXEC SP_ModifyUserLogin;
139 GO
140
141 --After the Store Procedure is created the table
142
143 select * from UserLogins;
144 GO
145
146 GO
147 GO
```

150 %

Messages

Commands completed successfully.

Completion time: 2021-09-02T02:54:39.9101094-04:00

150 %

Query executed successfully.

DINARTRIOSEDUCA\SQLEXPRESS ... | DINARTRIOSEDUCA\ruzdom... | dbpr...





Object Explorer

- Connect ▾
- AdiyatPC (SQL Server 15.0.2000 - ADIYATPC)\n
 - Databases
 - Security
 - Server Objects
 - Replication
 - PolyBase
 - Management
 - XEvent Profiler
- LenovoWin10 (SQL Server 15.0.2000 - LENOVC\n
- DINARTRIOSEDUCA\SQLEXPRESS (SQL Server\n
 - Databases
 - System Databases
 - Database Snapshots
 - dbbank
 - dbFDAS
 - dbmetrocollege
 - dbprojectbank
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.Account
 - dbo.AccountStatusType
 - dbo.AccountType
 - dbo.Customer
 - dbo.CustomerAccount
 - dbo.Employee
 - dbo.FailedTransactionErrorType
 - dbo.FailedTransactionLog
 - dbo.LoginAccount
 - dbo.LoginErrorLog
 - dbo.OverDraftLog
 - dbo.SavingsInterestRates
 - dbo.TransactionLog
 - dbo.TransactionType
 - dbo.UserLogins
 - dbo.UserSecurityAnswers
 - dbo.UserSecurityQuestions
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Service Broker
 - Storage

```
118 GO
119
120 --After the Store Procedure is created the table
121
122 select * from UserLogins;
123
124 GO
125
```

	UserLoginID	UserName	UserPassword
1	701	User_Anderson	TopSecret123
2	702	User_Beaupre	TopSecret124
3	703	User_Belanger	TopSecret125
4	704	User_Bouchard	TopSecret126
5	705	User_Cameron	TopSecret127
6	706	User_Campbell	TopSecret128
7	707	User_Chen	TopSecret129
8	708	User_Cote	TopSecret130
9	709	User_Davis	TopSecret131
10	710	User_Evans	TopSecret132
11	711	User_Fortin	TopSecret133
12	712	User_Gagnon	TopSecret134
13	713	User_Gauthier	TopSecret135
14	714	User_Gill	TopSecret136
15	715	User_Girard	TopSecret137
16	716	User_Graham	TopSecret138
17	717	User_Grant	TopSecret139
18	718	User_Hall	TopSecret140
19	719	User_Harris	TopSecret141
20	720	User_Jackson	TopSecret142
21	721	User_Johnson	TopSecret143
22	722	User_Johnston	TopSecret144
23	723	User_Kennedy	TopSecret145
24	724	User_Khan	TopSecret146
25	725	User_King	TopSecret147
26	726	User_Landy	TopSecret148
27	727	User_Lapointe	TopSecret149
28	728	User_Lavoie	TopSecret150
29	729	User_Leblanc	TopSecret151
30	730	User_Levesque	TopSecret152

After Executing the
Store Procedure



Query executed successfully.

DINARTRIOSEDUCA\SQLEXPRESS ... DINARTRIOSEDUCA\ruzdom... dbproj



Phase 02: Q07

**Create a stored procedure that accepts
AccountId as a parameter and returns
customer's full name. [Advanced]**





Object Explorer

Connect ▾

- AdiyatPC (SQL Server 15.0.2000 - ADIYATPC)\n
 - Databases
 - Security
 - Server Objects
 - Replication
 - PolyBase
 - Management
 - XEvent Profiler
- LenovoWin10 (SQL Server 15.0.2000 - LENOVC)\n
 - Databases
 - System Databases
 - Database Snapshots
 - dbbank
 - dbFDAS
 - dbmetrocollege
 - dbprojectbank
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.Account
 - dbo.AccountStatusType
 - dbo.AccountType
 - dbo.Customer
 - dbo.CustomerAccount
 - dbo.Employee
 - dbo.FailedTransactionErrorType
 - dbo.FailedTransactionLog
 - dbo.LoginAccount
 - dbo.LoginErrorLog
 - dbo.OverDraftLog
 - dbo.SavingsInterestRates
 - dbo.TransactionLog
 - dbo.TransactionType
 - dbo.UserLogins
 - dbo.UserSecurityAnswers
 - dbo.UserSecurityQuestions
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Service Broker
 - Storage
- DINARTRIOSEDUCA\SQLEXPRESS (SQL Server)\n
 - Databases

```
96
97    --Q7
98
99    GO
100   IF EXISTS (SELECT * FROM sys.procedures WHERE NAME='SP_FullNameFromAccountId')
101   DROP PROC SP_FullNameFromAccountId
102   GO
103   create proc SP_FullNameFromAccountId
104       @AccountId int,
105       @Fullname nvarchar(100) output
106   as
107   begin
108       if (@AccountId in (select AccountID from CustomerAccount))
109           begin
110               Select @FullName= C.CustomerFirstName+' '+C.CustomerMiddleInitial+' '+C.CustomerLastName
111               from Customer C
112               join CustomerAccount CA
113               on CA.CustomerID=C.CustomerID
114               where CA.AccountID=@AccountId;
115               set @Fullname=replace (@FullName, ' ', ', ')
116           end
117       else
118           begin
119               print 'There is no Customer with Account Id= '+CONVERT(varchar(12), @AccountId )
120           end
121       end
122   GO
```

165 % Messages

Commands completed successfully.

165 % DINARTRIOSEDUCA\SQLEXPRESS ... | DINARTRIOSEDUCA\ruzdom... | dbpr...

Query executed successfully.





dbprojectbank

MohammadMonjurEE...\\ruzdomain (55)* X MohammadMonjurEE...\\ruzdomain (53)* L6-Stored procedu...CA\\ruzdomain (54)

Object Explorer

Connect ▾

- AdiyatPC (SQL Server 15.0.2000 - ADIYATPC)\n
 - Databases
 - Security
 - Server Objects
 - Replication
 - PolyBase
 - Management
 - XEvent Profiler
- LenovoWin10 (SQL Server 15.0.2000 - LENOVC)\n
 - Databases
- DINARTRIOSEDUCA\SQLEXPRESS (SQL Server)\n
 - Databases
 - System Databases
 - Database Snapshots
 - dbbank
 - dbFDAS
 - dbmetrocollege
 - dbprojectbank
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.Account
 - dbo.AccountStatusType
 - dbo.AccountType
 - dbo.Customer
 - dbo.CustomerAccount
 - dbo.Employee
 - dbo.FailedTransactionErrorType
 - dbo.FailedTransactionLog
 - dbo.LoginAccount
 - dbo.LoginErrorLog
 - dbo.OverDraftLog
 - dbo.SavingsInterestRates
 - dbo.TransactionLog
 - dbo.TransactionType
 - dbo.UserLogins
 - dbo.UserSecurityAnswers
 - dbo.UserSecurityQuestions
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Service Broker
 - Storage

```
126  
127  
128    --Executing for valid account id  
129    Declare @FullName nvarchar(100)  
130    exec SP_FullNameFromAccountId 100022, @FullName out  
131    Print ' Full name is '+@FullName  
132  
133    GO  
134  
135  
136
```

Using a valid
Account ID

Full name is Johnston S Young
Completion time: 2021-08-31T23:05:23.5955069-04:00

Query executed successfully.

DINARTRIOSEDUCA\SQLEXPRESS ... | DINARTRIOSEDUCA\\ruzdom... | dbpr...

Ln 129

Col 1

Ch 1

INS





Object Explorer

- Connect
- AdiyatPC (SQL Server 15.0.2000 - ADIYATPC)\n
 - Databases
 - Security
 - Server Objects
 - Replication
 - PolyBase
 - Management
 - XEvent Profiler
- LenovoWin10 (SQL Server 15.0.2000 - LENOVC)\n
 - Databases
- DINARTRIOSEDUCA\SQLEXPRESS (SQL Server)\n
 - Databases
 - System Databases
 - Database Snapshots
 - dbbank
 - dbFDAS
 - dbmetrocollege
 - dbprojectbank
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.Account
 - dbo.AccountStatusType
 - dbo.AccountType
 - dbo.Customer
 - dbo.CustomerAccount
 - dbo.Employee
 - dbo.FailedTransactionErrorType
 - dbo.FailedTransactionLog
 - dbo.LoginAccount
 - dbo.LoginErrorLog
 - dbo.OverDraftLog
 - dbo.SavingsInterestRates
 - dbo.TransactionLog
 - dbo.TransactionType
 - dbo.UserLogins
 - dbo.UserSecurityAnswers
 - dbo.UserSecurityQuestions
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Service Broker
 - Storage

MohammadMonjurEE...\\ruzdomain (55)* | MohammadMonjurEE...\\ruzdomain (53)* | L6-Stored procedu...CA\\ruzdomain (54)

```
126  
127  
128    --Executing for invalid account id  
129    Declare @FullName nvarchar(100)  
130    exec SP_FullNameFromAccountId 100029, @FullName out  
131    Print ' Full name is '+@FullName  
132  
133  
134    GO  
135  
136
```

Using an invalid
Account ID

Messages
There is no Customer with Account Id= 100029

Completion time: 2021-08-31T23:06:18.0718746-04:00

Query executed successfully.

DINARTRIOSEDUCA\SQLEXPRESS ... | DINARTRIOSEDUCA\\ruzdom... | dbpr...

Ln 129

Col 1

Ch 1

INS





Phase 02: Q08

Create a stored procedure that returns error logs inserted in the last 24 hours. [Advanced]



dMonjurElahi_Project_QuestionsAndAnswers.sql - DINARTRIOSEDUCA\SQLEXPRESS.dbprojectbank (DINARTRIOSEDUCA\ruzdomain (55))* - Microsoft SQL Server Management Studio

Quick Launch (Ctrl+Q)

File Query Project Tools Window Help

New Query

update

Object Explorer

Connect

dbprojectbank Execute

L5-join andsql...UCA\ruzdomain (56) MohammadMonjurEE...\ruzdomain (55)* MohammadMonjurEE...\ruzdomain (53)* L6-Stored procedu...CA\ruzdomain (54)

190
191 --Q8
192
193 IF EXISTS (SELECT * FROM sys.procedures WHERE NAME='SP_ERRORS_24H')
194 DROP PROCEDURE SP_ERRORS_24H;
195 GO
196
197 CREATE PROCEDURE SP_ERRORS_24H
198 AS
199 SELECT * FROM LoginErrorLog
200 WHERE ErrorTime BETWEEN DATEADD(hh, -24, GETDATE()) AND GETDATE();
201 GO
202
203 EXEC SP_ERRORS_24H;
204 GO

165 %

Messages

Commands completed successfully.

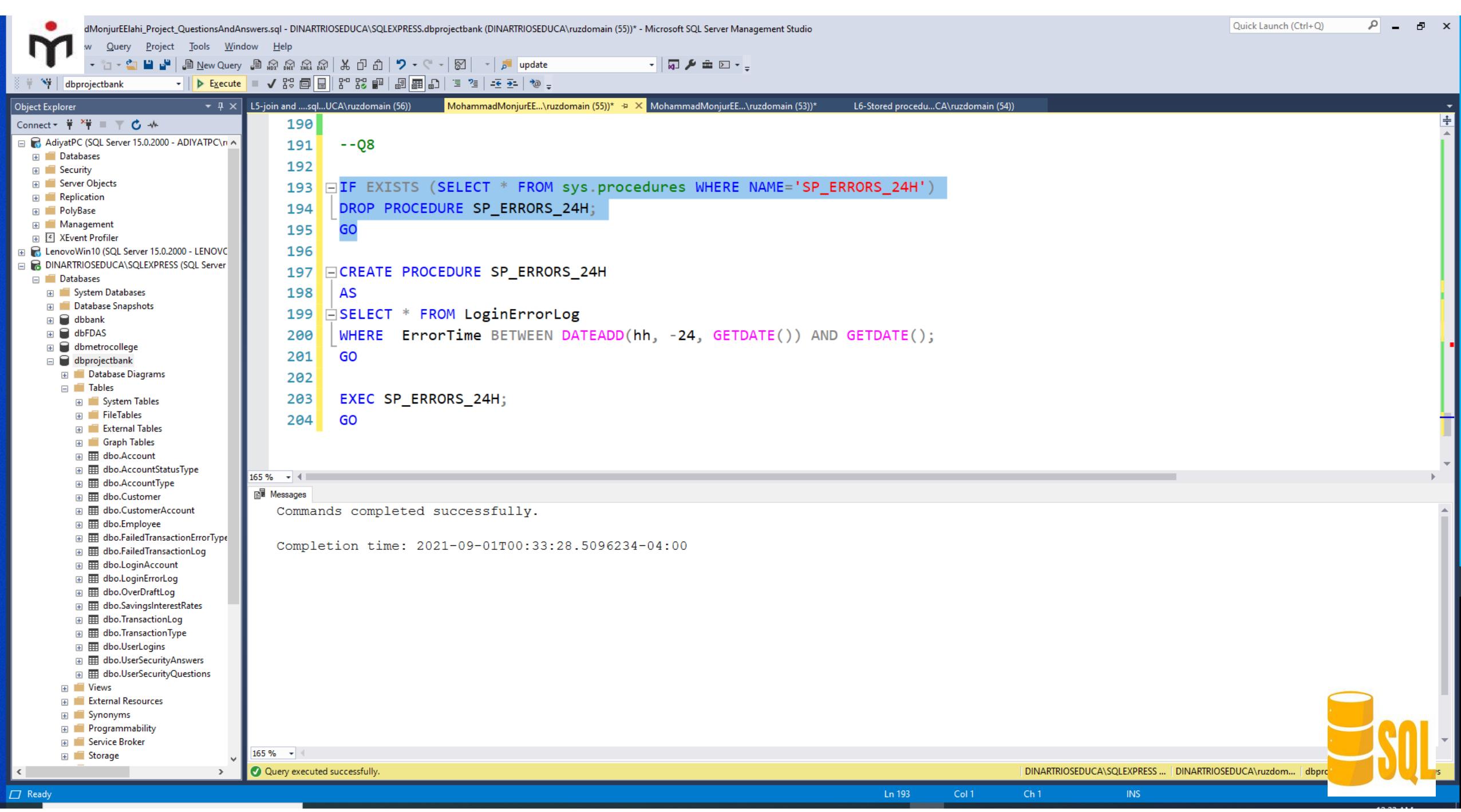
Completion time: 2021-09-01T00:33:28.5096234-04:00

Query executed successfully.

DINARTRIOSEDUCA\SQLEXPRESS ... | DINARTRIOSEDUCA\ruzdom... | dbpr...

Ln 193 Col 1 Ch 1 INS

Ready



dMonjurElahi_Project_QuestionsAndAnswers.sql - DINARTRIOSEDUCA\SQLEXPRESS.dbprojectbank (DINARTRIOSEDUCA\ruzdomain (55))* - Microsoft SQL Server Management Studio

Quick Launch (Ctrl+Q)

File Query Project Tools Window Help

New Query Execute

Object Explorer

Connect

AdiyatPC (SQL Server 15.0.2000 - ADIYATPC)\n Databases Security Server Objects Replication PolyBase Management XEvent Profiler

LenovoWin10 (SQL Server 15.0.2000 - LENOVC) Databases System Databases Database Snapshots dbbank dbFDAS dbmetrocollege dbprojectbank Database Diagrams Tables System Tables FileTables External Tables Graph Tables dbo.Account dbo.AccountStatusType dbo.AccountType dbo.Customer dbo.CustomerAccount dbo.Employee dbo.FailedTransactionErrorType dbo.FailedTransactionLog dbo.LoginAccount dbo.LoginErrorLog dbo.OverDraftLog dbo.SavingsInterestRates dbo.TransactionLog dbo.TransactionType dbo.UserLogins dbo.UserSecurityAnswers dbo.UserSecurityQuestions Views External Resources Synonyms Programmability Service Broker Storage

L5-join andsql...UCA\ruzdomain (56) MohammadMonjurEE...\ruzdomain (55)* MohammadMonjurEE...\ruzdomain (53)* L6-Stored procedu...CA\ruzdomain (54)

190
191 --Q8
192
193 IF EXISTS (SELECT * FROM sys.procedures WHERE NAME='SP_ERRORS_24H')
194 DROP PROCEDURE SP_ERRORS_24H;
195 GO
196
197 CREATE PROCEDURE SP_ERRORS_24H
198 AS
199 SELECT * FROM LoginErrorLog
200 WHERE ErrorTime BETWEEN DATEADD(hh, -24, GETDATE()) AND GETDATE();
201 GO
202
203 EXEC SP_ERRORS_24H;
204 GO

165 % Messages
Commands completed successfully.

Completion time: 2021-09-01T00:34:30.8622553-04:00

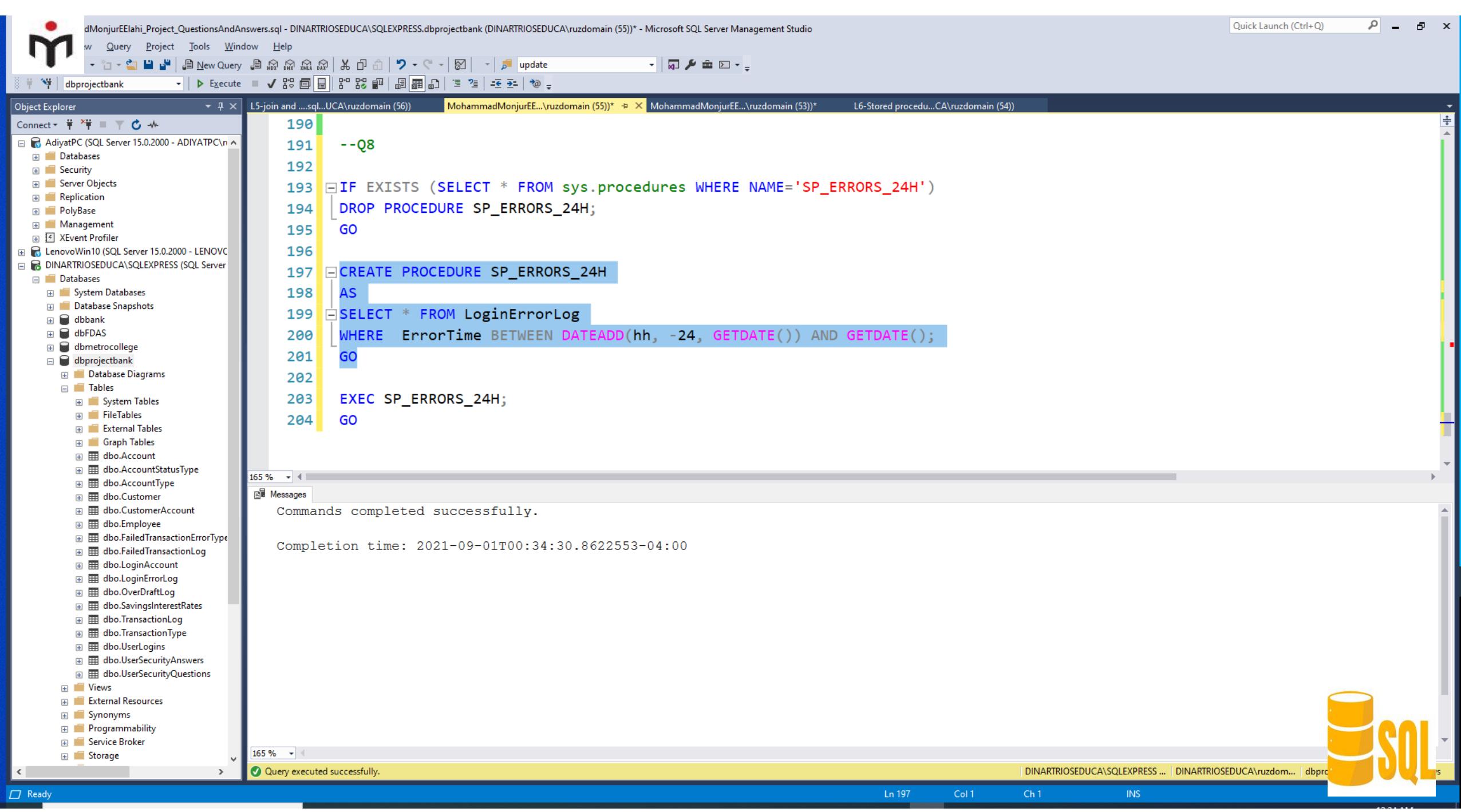
165 %

Query executed successfully.

DINARTRIOSEDUCA\SQLEXPRESS ... | DINARTRIOSEDUCA\ruzdom... | dbprc

Ln 197 Col 1 Ch 1 INS

Ready





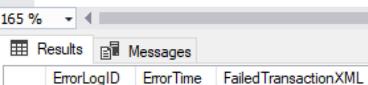
Object Explorer

Connect ▾

- AdiyatPC (SQL Server 15.0.2000 - ADIYATPC)\n
 - Databases
 - Security
 - Server Objects
 - Replication
 - PolyBase
 - Management
 - XEvent Profiler
- LenovoWin10 (SQL Server 15.0.2000 - LENOVC)\n
 - Databases
 - System Databases
 - Database Snapshots
 - dbbank
 - dbFDAS
 - dbmetrocollege
 - dbprojectbank\n
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.Account
 - dbo.AccountStatusType
 - dbo.AccountType
 - dbo.Customer
 - dbo.CustomerAccount
 - dbo.Employee
 - dbo.FailedTransactionErrorType
 - dbo.FailedTransactionLog
 - dbo.LoginAccount
 - dbo.LoginErrorLog
 - dbo.OverDraftLog
 - dbo.SavingsInterestRates
 - dbo.TransactionLog
 - dbo.TransactionType
 - dbo.UserLogins
 - dbo.UserSecurityAnswers
 - dbo.UserSecurityQuestions
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Service Broker
 - Storage

```
190  
191    --Q8  
192  
193    IF EXISTS (SELECT * FROM sys.procedures WHERE NAME='SP_ERRORS_24H')  
194        DROP PROCEDURE SP_ERRORS_24H;  
195        GO  
196  
197    CREATE PROCEDURE SP_ERRORS_24H  
198        AS  
199        SELECT * FROM LoginErrorLog  
200        WHERE ErrorTime BETWEEN DATEADD(hh, -24, GETDATE()) AND GETDATE();  
201        GO  
202  
203    EXEC SP_ERRORS_24H;  
204    GO
```

Errors in my table
are not that recent!!



Query executed successfully.

DINARTRIOSEDUCA\SQLEXPRESS ... | DINARTRIOSEDUCA\ruzdom... | dbpr...





Object Explorer

Connect ▾

- AdiyatPC (SQL Server 15.0.2000 - ADIYATPC)\n
 - Databases
 - Security
 - Server Objects
 - Replication
 - PolyBase
 - Management
 - XEvent Profiler
- LenovoWin10 (SQL Server 15.0.2000 - LENOVC)\n
 - Databases
 - System Databases
 - Database Snapshots
 - dbbank
 - dbFDAS
 - dbmetrocollege
 - dbprojectbank\n
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.Account
 - dbo.AccountStatusType
 - dbo.AccountType
 - dbo.Customer
 - dbo.CustomerAccount
 - dbo.Employee
 - dbo.FailedTransactionErrorType
 - dbo.FailedTransactionLog
 - dbo.LoginAccount
 - dbo.LoginErrorLog
 - dbo.OverDraftLog
 - dbo.SavingsInterestRates
 - dbo.TransactionLog
 - dbo.TransactionType
 - dbo.UserLogins
 - dbo.UserSecurityAnswers
 - dbo.UserSecurityQuestions
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Service Broker
 - Storage

```
205  
206    --Let us create another Store Procedure so that we can capture our data which are 4 months old  
207  
208    IF EXISTS (SELECT * FROM sys.procedures WHERE NAME='SP_ERRORS_4M')  
209        DROP PROCEDURE SP_ERRORS_4M;  
210        GO  
211  
212    CREATE PROCEDURE SP_ERRORS_4M  
213        AS  
214        SELECT * FROM LoginErrorLog  
215        WHERE ErrorTime BETWEEN DATEADD(mm, -4, GETDATE()) AND GETDATE();  
216        GO  
217  
218    EXEC SP_ERRORS_4M;  
219    GO  
220
```

165 %

Results

ErrorLogID | ErrorTime | FailedTransactionXML

ErrorLogID	ErrorTime	FailedTransactionXML

Query executed successfully.

DINARTRIOSEDUCA\SQLEXPRESS ... | DINARTRIOSEDUCA\ruzdom... | dbpr...

Ln 218

Col 19

Ch 19

INS





Object Explorer

Connect ▾

- AdiyatPC (SQL Server 15.0.2000 - ADIYATPC)\n
 - Databases
 - Security
 - Server Objects
 - Replication
 - PolyBase
 - Management
 - XEvent Profiler
- LenovoWin10 (SQL Server 15.0.2000 - LENOVC)\n
 - Databases
 - System Databases
 - Database Snapshots
 - dbbank
 - dbFDAS
 - dbmetrocollege
 - dbprojectbank\n
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.Account
 - dbo.AccountStatusType
 - dbo.AccountType
 - dbo.Customer
 - dbo.CustomerAccount
 - dbo.Employee
 - dbo.FailedTransactionErrorType
 - dbo.FailedTransactionLog
 - dbo.LoginAccount
 - dbo.LoginErrorLog
 - dbo.OverDraftLog
 - dbo.SavingsInterestRates
 - dbo.TransactionLog
 - dbo.TransactionType
 - dbo.UserLogins
 - dbo.UserSecurityAnswers
 - dbo.UserSecurityQuestions
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Service Broker
 - Storage

```
205  
206 --Let us create another Store Procedure so that we can capture our data which are 4 months old  
207  
208 IF EXISTS (SELECT * FROM sys.procedures WHERE NAME='SP_ERRORS_4M')  
209 DROP PROCEDURE SP_ERRORS_4M;  
210 GO  
211  
212 CREATE PROCEDURE SP_ERRORS_4M  
213 AS  
214 SELECT * FROM LoginErrorLog  
215 WHERE ErrorTime BETWEEN DATEADD(mm, -4, GETDATE()) AND GETDATE();  
216 GO  
217  
218 EXEC SP_ERRORS_4M;  
219 GO  
220
```

Changed the timeline to 4 months to get an output

Results

ErrorLogID	ErrorTime	FailedTransactionXML
1	68666001	<Error><UserLoginID>711</UserLoginID><Code>68666...
2	68666002	<Error><UserLoginID>733</UserLoginID><Code>68666...
3	68666003	<Error><UserLoginID>719</UserLoginID><Code>68666...
4	68666004	<Error><UserLoginID>725</UserLoginID><Code>68666...
5	68666005	<Error><UserLoginID>749</UserLoginID><Code>68666...

Query executed successfully.

DINARTRIOSEDUCA\SQLEXPRESS ... | DINARTRIOSEDUCA\ruzdom... | dbpr...





Phase 02: Q09

Create a stored procedure that takes a deposit as a parameter and updates CurrentBalance value for that particular account. [Advanced]





```
223
224 --Q9: Create a stored procedure that takes a deposit as a parameter and updates CurrentBalance value for that particular account. [Advanced]
225 --Just let us have a look at the Current balance before the Store Procedure
226 GO
227 select * from Account
228 where AccountID = 100023;
229 GO
230 --Now, let us create the Store Procedure and check the change
231 IF EXISTS (SELECT * FROM sys.procedures WHERE NAME='SP_Update_cBalance_After_Deposit')
232 DROP PROCEDURE SP_Update_cBalance_After_Deposit;
233 GO
234 CREATE PROCEDURE SP_Update_cBalance_After_Deposit @AccountID INT, @Deposit INT
235 AS
236 begin
237 if (@AccountID in (select AccountID from Account))
238 begin
239 UPDATE Account
240 SET CurrentBalance = CurrentBalance + @Deposit
241 where AccountID = @AccountID;
242 end
243 else
244 begin
245 print 'There is no Customer with Account Id= '+CONVERT(varchar(12), @AccountID )
```

Before the update in
the balance

	AccountID	CurrentBalance	AccountTypeID	AccountStatusTypeID	InterestSavingsRateID
1	100023	117000	131	142	214

Query executed successfully.

DINARTRIOSEDUCA\SQLEXPRESS ... | DINARTRIOSEDUCA\ruzdom... | dbpr...





Object Explorer

- Databases
 - System Datal
 - Database Sna
 - dbbank
 - dbFDAS
 - dbmetrocoll
 - dbprojectbar
 - Database
 - Tables
 - System
 - FileTa
 - Extern
 - Graph
 - dbo.A
 - dbo.A
 - dbo.A
 - dbo.C
 - dbo.C
 - dbo.E
 - dbo.F
 - dbo.F
 - dbo.L
 - dbo.L
 - dbo.C
 - dbo.S
 - dbo.T
 - dbo.T
 - dbo.U
 - dbo.U
 - Views
 - External F
 - Synonym
 - Program
 - Service Bi
 - Storage
 - Security
 - dbstor
 - dbtest1
 - dbtest2
 - dbtest22
 - dbtest3
 - dvdrental
 - metro_uni
 - mydb
 - Security

```
226 GO
227 select * from Account
228     where AccountID = 100023;
229 GO
230 --Now, let us create the Store Procedure and check the change
231 IF EXISTS (SELECT * FROM sys.procedures WHERE NAME='SP_Update_cBalance_After_Deposit')
232 DROP PROCEDURE SP_Update_cBalance_After_Deposit;
233 GO
234 CREATE PROCEDURE SP_Update_cBalance_After_Deposit @AccountID INT, @Deposit INT
235 AS
236 begin
237 if (@AccountID in (select AccountID from Account))
238 begin
239 --    if (@Deposit <= (select CurrentBalance from Account where AccountID =@AccountID ))
240     UPDATE Account
241     SET CurrentBalance = CurrentBalance + @Deposit
242     where AccountID = @AccountID;
243 end
244 else
245 begin
246 print 'There is no Customer with Account Id= '+CONVERT(varchar(12), @AccountID )
247 end
248 end
249 GO
250 EXEC SP_Update_cBalance_After_Deposit 100023, 3000;
251 GO
252 select * from Account
253     where AccountID = 100023;
254 GO
255
```

150 %

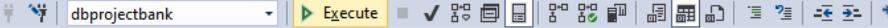
Messages

Commands completed successfully.

150 %

Query executed successfully.





Object Explorer Connect SQLQuery4.sql - DI...CA\ruzdomain (58) MohammadMonjurEE...\ruzdomain (55)* LENOVOWIN10\SQL...bank - Diagram_0* MohammadMonjurEE...\ruzdomain (53)*

```
232 DROP PROCEDURE SP_Update_cBalance_After_Deposit;
233 GO
234 CREATE PROCEDURE SP_Update_cBalance_After_Deposit @AccountID INT, @Deposit INT
235 AS
236 begin
237 if (@AccountID in (select AccountID from Account))
238 begin
239 UPDATE Account
240 SET CurrentBalance = CurrentBalance + @Deposit
241 where AccountID = @AccountID;
242 end
243 else
244 begin
245 print 'There is no Customer with Account Id= '+CONVERT(varchar(12), @AccountID )
246 end
247 end
248 GO
249 EXEC SP_Update_cBalance_After_Deposit 100023, 3000;
250 GO
251 select * from Account
252 where AccountID = 100023;
253 GO
```

After the update in
the balance

	AccountID	CurrentBalance	AccountTypeID	AccountStatusTypeID	InterestSavingsRateID
1	100023	120000	131	142	214

Query executed successfully.

DINARTRIOSEDUCA\SQLEXPRESS ... | DINARTRIOSEDUCA\ruzdom... | dbpr...





Phase 02: Q10

Create a stored procedure that takes a withdrawal amount as a parameter and updates





Object Explorer

- Databases
 - System Datal
 - Database Sna
 - dbbank
 - dbFDAS
 - dbmetrocoll
 - dbprojectbar
 - Database
 - Tables
 - System
 - FileTa
 - Extern
 - Graph
 - dbo.A
 - dbo.A
 - dbo.A
 - dbo.C
 - dbo.C
 - dbo.E
 - dbo.F
 - dbo.F
 - dbo.L
 - dbo.L
 - dbo.C
 - dbo.S
 - dbo.T
 - dbo.T
 - dbo.U
 - dbo.U
 - Views
 - External F
 - Synonym
 - Program
 - Service Bi
 - Storage
 - Security
 - dbstor
 - dbtest1
 - dbtest2
 - dbtest22
 - dbtest3
 - dvdrental
 - metro_uni
 - mydb
 - Security

```
256 --Q10: Create a stored procedure that takes a withdrawal amount as a parameter and updates
257
258 --Just let us have alook at the Current balance before the Store Procedure
259
260 GO
261 select * from Account
262 where AccountID = 100015;
263 GO
264
265 --Now let us create the store procedure
266
267 IF EXISTS (SELECT * FROM sys.procedures WHERE NAME='SP_Update_cBalance_After_Withdrawal')
268 DROP PROCEDURE SP_Update_cBalance_After_Withdrawal;
269 GO
270
271 CREATE PROCEDURE SP_Update_cBalance_After_withdrawal @AccountID INT, @Withdraw INT
272 AS
273 begin
274 if (@AccountID in (select AccountID from Account))
275 begin
276 if (@Withdraw <= (select CurrentBalance from Account where AccountID =@AccountID ))
277 begin
278 UPDATE Account
```

150 %

Results Messages

	AccountID	CurrentBalance	AccountTypeID	AccountStatusTypeID	InterestSavingsRateID
1	100015	3300	131	142	212

Before the
withdrawal

Query executed successfully.

DINARTRIOSEDUCA\SQLEXPRESS ... | DINARTRIOSEDUCA\ruzdom... | dbpr...





Object Explorer

Connect ▾

- Databases
 - System Datal
 - Database Sna
 - dbbank
 - dbFDAS
 - dbmetrocoll
 - dbprojectbar
 - Database
 - Tables
 - Syste
 - FileTa
 - Extern
 - Graph
 - dbo.A
 - dbo.A
 - dbo.A
 - dbo.C
 - dbo.C
 - dbo.E
 - dbo.F
 - dbo.F
 - dbo.L
 - dbo.L
 - dbo.C
 - dbo.S
 - dbo.T
 - dbo.T
 - dbo.U
 - dbo.U
 - Views
 - External F
 - Synonym
 - Program
 - Service Bi
 - Storage
 - Security
 - dbstor
 - dbtest1
 - dbtest2
 - dbtest22
 - dbtest3
 - dvdrental
 - metro_uni
 - mydb
 - Security

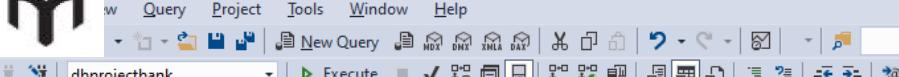
```
265      --Now let us create the store procedure
266
267  IF EXISTS (SELECT * FROM sys.procedures WHERE NAME='SP_Update_cBalance_After_Withdrawal')
268  DROP PROCEDURE SP_Update_cBalance_After_Withdrawal;
269  GO
270
271  CREATE PROCEDURE SP_Update_cBalance_After_withdrawal @AccountID INT, @Withdraw INT
272  AS
273  begin
274    if (@AccountID in (select AccountID from Account))
275    begin
276      if (@Withdraw <= (select CurrentBalance from Account where AccountID =@AccountID ))
277      begin
278        UPDATE Account
279        SET CurrentBalance = CurrentBalance - @Withdraw
280        where AccountID = @AccountID;
281      end
282    else
283      begin
284        print 'Current Balance is less than the withdrawal amount '+CONVERT(varchar(12),@Withdraw )
285      end
286    end
287    else
288      begin
289        print 'There is no Customer with Account Id= '+CONVERT(varchar(12), @AccountID )
290      end
291    end
292  GO
293
294  EXEC SP_Update_cBalance_After_Withdrawal 100005, 10;
295  GO
296
```

150 %

Results Messages

Query executed successfully.





dbprojectbank

SQLQuery4.sql - DI...CA\ruzdomain (58)

MohammadMonjurEE...\ruzdomain (55)*

LENOWIN10\SQL...bank - Diagram_0*

MohammadMonjurEE...\ruzdomain (53)*

Object Explorer

Connect
Databases
System Datal
Database Sna
dbbank
dbFDAS
dbmetrocoll
dbprojectbar
Database
Tables
System
FileTa
Extern
Graph
dbo.A
dbo.A
dbo.A
dbo.C
dbo.C
dbo.E
dbo.F
dbo.F
dbo.L
dbo.L
dbo.C
dbo.S
dbo.T
dbo.T
dbo.U
dbo.U
Views
External F
Synonym
Programm
Service Bi
Storage
Security
dbstor
dbtest1
dbtest2
dbtest22
dbtest3
dvrental
metro_uni
mydb
Security

```
280      where AccountID = @AccountID;
281      end
282      else
283      begin
284          print 'Current Balance is less than the withdrawal amount '+CONVERT(varchar(12),@Withdraw )
285          end
286      end
287      else
288      begin
289          print 'There is no Customer with Account Id= '+CONVERT(varchar(12), @AccountID )
290          end
291      end
292      GO
293
294 EXEC SP_Update_cBalance_After_Withdrawal 100015, 300;
295 GO
296
297 --Just let us have alook at the Current balance after the Store Procedure execution
298
299 GO
300 select * from Account
301 where AccountID = 100015;
302 GO
```

After the
withdrawal

Results	Messages										
<table border="1"><thead><tr><th>AccountID</th><th>CurrentBalance</th><th>AccountTypeID</th><th>AccountStatusTypeID</th><th>InterestSavingsRateID</th></tr></thead><tbody><tr><td>100015</td><td>3000</td><td>131</td><td>142</td><td>212</td></tr></tbody></table>	AccountID	CurrentBalance	AccountTypeID	AccountStatusTypeID	InterestSavingsRateID	100015	3000	131	142	212	
AccountID	CurrentBalance	AccountTypeID	AccountStatusTypeID	InterestSavingsRateID							
100015	3000	131	142	212							

Query executed successfully.

DINARTRIOSEDUCA\SQLEXPRESS ... | DINARTRIOSEDUCA\ruzdom... | dbpr...



Code for the Project Problems

```

1 use dbprojectbank;
2 GO
3 -- Q1: Create a view to get all customers with checking account from ON province. [Moderate]
4
5 IF EXISTS (SELECT * FROM sys.all_views WHERE NAME='VW_ON_ChekkingAccount')
6 DROP VIEW VW_ON_ChekkingAccount;
7 GO
8 create view VW_ON_ChekkingAccount
9 as
10 select C.CustomerID,C.CustomerFirstName+' '+C.CustomerLastName [Customer Name],C.State, AT.AccountTypeDescription
11 from Customer C
12 join CustomerAccount CA
13 on C.CustomerID=CA.CustomerID
14 join Account A
15 on CA.AccountID=A.AccountID
16 join AccountType AT
17 on A.AccountTypeID=AT.AccountTypeID
18 where AT.AccountTypeDescription='Checking Account' and C.State='ON';
19 GO
20
21 select * from VW_ON_ChekkingAccount;
22
23 GO
24 --Q2: Create a view to get all customers with total account balance (including interest rate) greater than 5000. [Advanced]
25 --Here we just focus on the Savings account Balance.
26 GO
27
28 IF EXISTS (SELECT * FROM sys.all_views WHERE NAME='VW_Customer_Balance_TTLQ')
29 DROP VIEW VW_Customer_Balance_TTLQ;
30 GO
31 CREATE VIEW VW_Customer_Balance_TTLQ AS
32
33 SELECT C.CustomerFirstName+' '+C.CustomerLastName AS [Customer Name], A.AccountID, A.CurrentBalance AS Account_Balance,
34 S.SavingsInterestRateValue,
35 cast((A.CurrentBalance + A.CurrentBalance*(S.SavingsInterestRateValue/12)*Datediff(MM,TL.TransactionDate, GETDATE())) as decimal(10,2))
36 AS Total_Account_Balance
37
38 FROM
39 SavingsInterestRates S
40 join Account A
41 on A.InterestSavingsRateID = S.SavingsInterestRateID
42 join TransactionLog TL
43 on TL.AccountID = A.AccountID
44 join Customer C
45 on C.CustomerID = TL.CustomerID
46 where (A.CurrentBalance + A.CurrentBalance*(S.SavingsInterestRateValue/12)*Datediff(MM,TL.TransactionDate, GETDATE())) > 5000
47 ;
48 GO
49
50 select * from VW_Customer_Balance_TTLQ;
51 GO
52

```

```

53
54 --Q3: Create a view to get counts of checking and savings accounts by customer. [Moderate]
55
56 IF EXISTS (SELECT * FROM sys.all_views WHERE NAME='VW_Counts_Account')
57 DROP VIEW VW_Counts_Account;
58 GO
59
60 create view VW_Counts_Account
61   as
62     select C.CustomerFirstName+' '+C.CustomerLastName [Customer Name], AT.AccountTypeDescription, count(*) [Number of Account]
63     from Customer C
64       join CustomerAccount CA
65         on C.CustomerID=CA.CustomerID
66       join Account A
67         on CA.AccountID=A.AccountID
68       join AccountType AT
69         on A.AccountTypeID=AT.AccountTypeID
70     group by C.CustomerFirstName+' '+C.CustomerLastName, AT.AccountTypeDescription;
71
72 GO
73 select * from VW_Counts_Account
74   order by [Customer Name];
75
76 GO
77
78
79 --Q4: Create a view to get any particular user's login and password using AccountId. [Moderate]
80
81 IF EXISTS (SELECT * FROM sys.all_views WHERE NAME='VW_Users_Login_password')
82 DROP VIEW VW_Users_Login_password;
83 GO
84 create view VW_Users_Login_password
85   as
86     select C.CustomerFirstName+' '+C.CustomerLastName [Customer Name], UL.UserName UserLoginName, UL.UserPassword UserLoginPassword
87     from Customer C
88       join CustomerAccount CA
89         on C.CustomerID=CA.CustomerID
90       join LoginAccount LA
91         on CA.AccountID=LA.AccountID
92       join UserLogins UL
93         on LA.UserLoginID=UL.UserLoginID
94     where CA.AccountID = 100025;
95
96 GO
97 select * from VW_Users_Login_password;
98
99 GO
100
101
102 --Q5: Create a view to get all customers' overdraft amount. [Moderate]
103
104 IF EXISTS (SELECT * FROM sys.all_views WHERE NAME='VW_All_Customers_Overdraft_Amount')

```

```
105 DROP VIEW VW_All_Customers_Overdraft_Amount;
106 GO
107 create view VW_All_Customers_Overdraft_Amount
108   as
109   select C.CustomerFirstName+' '+C.CustomerLastName [Customer Name], ODL.OverDraftAmount, ODL.OverDraftDate,
110     ODL.OverDraftTransactionXml
111   from Customer C
112     join CustomerAccount CA
113       on C.CustomerID=CA.CustomerID
114     join OverDraftLog ODL
115       on CA.AccountID=ODL.AccountID;
116
117
118 GO
119 select * from VW_All_Customers_Overdraft_Amount;
120
121 GO
122
123 --Q6: Create a stored procedure to add "User_" as a prefix to everyone's login (username). [Moderate]
124 --Before the Store Procedure is created the table
125
126 select * from UserLogins;
127
128 --Now let us create the Store Procedure
129 GO
130 IF EXISTS (SELECT * FROM sys.procedures WHERE NAME='SP_ModifyUserLogin')
131 DROP PROC SP_ModifyUserLogin
132 GO
133 CREATE PROCEDURE SP_ModifyUserLogin
134   AS
135   UPDATE UserLogins
136     SET UserName = Concat('User_', UserName);
137
138 GO
139
140 EXEC SP_ModifyUserLogin;
141 GO
142
143 --After the Store Procedure is created the table
144
145 select * from UserLogins;
146
147 GO
148
149
150
151
152
153
154 --Q7: Create a stored procedure that accepts AccountId as a parameter and returns customer's full name. [Advanced]
155
156 GO
```

```

157 IF EXISTS (SELECT * FROM sys.procedures WHERE NAME='SP_FullNameFromAccountId')
158 DROP PROC SP_FullNameFromAccountId
159 GO
160 create proc SP_FullNameFromAccountId
161     @AccountID int,
162     @Fullname nvarchar(100) output
163 as
164 begin
165     if (@AccountID in (select AccountID from CustomerAccount))
166         begin
167             Select @FullName= C.CustomerFirstName+' '+C.CustomerMiddleInitial+' '+C.CustomerLastName
168             from Customer C
169             join CustomerAccount CA
170             on CA.CustomerID=C.CustomerID
171             where CA.AccountID=@AccountID;
172             set @Fullname=replace (@FullName, ' ', ' ')
173         end
174     else
175         begin
176             print 'There is no Customer with Account Id= '+CONVERT(varchar(12), @AccountID )
177         end
178     end
179 GO
180
181
182
183
184
185 --Executing for invalid account id
186 Declare @FullName nvarchar(100)
187 exec SP_FullNameFromAccountId 100029, @FullName out
188 Print ' Full name is '+@FullName
189
190
191 GO
192
193 --Q8: Create a stored procedure that returns error logs inserted in the last 24 hours. [Advanced]
194
195 IF EXISTS (SELECT * FROM sys.procedures WHERE NAME='SP_ERRORS_24H')
196 DROP PROCEDURE SP_ERRORS_24H;
197 GO
198
199 CREATE PROCEDURE SP_ERRORS_24H
200 AS
201 SELECT * FROM LoginErrorLog
202 WHERE ErrorTime BETWEEN DATEADD(hh, -24, GETDATE()) AND GETDATE();
203 GO
204
205 EXEC SP_ERRORS_24H;
206 GO
207
208 --Let us create another Store Procedure so that we can capture our data which are 4 months old

```

```
209
210 IF EXISTS (SELECT * FROM sys.procedures WHERE NAME='SP_ERRORS_4M')
211 DROP PROCEDURE SP_ERRORS_4M;
212 GO
213
214 CREATE PROCEDURE SP_ERRORS_4M
215 AS
216 SELECT * FROM LoginErrorLog
217 WHERE ErrorTime BETWEEN DATEADD(mm, -4, GETDATE()) AND GETDATE();
218 GO
219
220 EXEC SP_ERRORS_4M;
221 GO
222
223
224 --Q9: Create a stored procedure that takes a deposit as a parameter and updates CurrentBalance value for that particular account. [Advanced]
225 --Just let us have alook at the Current balance before the Store Procedure
226 GO
227 select * from Account
228 where AccountID = 100023;
229 GO
230 --Now, let us create the Store Procedure and check the change
231 IF EXISTS (SELECT * FROM sys.procedures WHERE NAME='SP_Update_cBalance_After_Deposit')
232 DROP PROCEDURE SP_Update_cBalance_After_Deposit;
233 GO
234 CREATE PROCEDURE SP_Update_cBalance_After_Deposit @AccountID INT, @Deposit INT
235 AS
236 begin
237     if (@AccountID in (select AccountID from Account))
238     begin
239         UPDATE Account
240             SET CurrentBalance = CurrentBalance + @Deposit
241             where AccountID = @AccountID;
242     end
243     else
244     begin
245         print 'There is no Customer with Account Id= '+CONVERT(varchar(12), @AccountID )
246     end
247 end
248 GO
249 EXEC SP_Update_cBalance_After_Deposit 100023, 3000;
250 GO
251 select * from Account
252 where AccountID = 100023;
253 GO
254
255
256 --Q10: Create a stored procedure that takes a withdrawal amount as a parameter and updates
257 --Just let us have alook at the Current balance before the Store Procedure
258 GO
```

```
261 select * from Account
262   where AccountID = 100015;
263 GO
264
265 --Now let us create the store procedure
266
267 IF EXISTS (SELECT * FROM sys.procedures WHERE NAME='SP_Update_cBalance_After_Withdrawal')
268 DROP PROCEDURE SP_Update_cBalance_After_Withdrawal;
269 GO
270
271 CREATE PROCEDURE SP_Update_cBalance_After_withdrawal @AccountID INT, @Withdraw INT
272 AS
273 begin
274   if (@AccountID in (select AccountID from Account))
275     begin
276       if (@Withdraw <= (select CurrentBalance from Account where AccountID =@AccountID ))
277         begin
278           UPDATE Account
279             SET CurrentBalance = CurrentBalance - @Withdraw
280             where AccountID = @AccountID;
281         end
282       else
283         begin
284           print 'Current Balance is less than the withdrawal amount '+CONVERT(varchar(12),@Withdraw )
285         end
286       end
287     else
288       begin
289         print 'There is no Customer with Account Id= '+CONVERT(varchar(12), @AccountID )
290       end
291     end
292 GO
293
294 EXEC SP_Update_cBalance_After_Withdrawal 100015, 300;
295 GO
296
297 --Just let us have alook at the Current balance after the Store Procedure execution
298
299 GO
300 select * from Account
301   where AccountID = 100015;
302 GO
303
304
```

Conclusion

- Worked end to end on the project to have valuable hands-on experience
- Faced and surpassed many hurdles during the journey
- Understood the importance of more and more practice to master SQL
- We are really thankful to Mr. Hamid Rajaee for his thorough coverage of topics including advanced features