



SQL Server Advanced – Database Development Lab Book

Document Revision History

Date	Revision No.	Author	Summary of Changes
25 th July 2011	2.0	Latha S.	Changes in Material made based on integration process
3 rd April, 2012	3.0	Shilpa Bhosle	Changes in Lab Book are made as an upgrade to SQL Server 2008
21 st Aug 2013	4.0	Shashank Saudagar	Changes in Lab Book are made as an upgrade to SQL Server 2012
14 th May 2015	5.0	Vaishali Kasture	Changes in Lab Book as per new curriculum of SQL Server 2012
9 th May 2016	6.0	Shital A Patil	Changes in Material made based on integration process as per Capgemini Course Structure
29 th May 2017	7.0	Shital A Patil	Changes in Lab Book are made based 10 Weeks Course Structure



Table of Contents

Lab 1.	Getting connected to the SQL Server 2012 Server	5
1.1	Steps to connect to the SQL Server 2012 Server	5
1.2	Getting Familiar with SQL Server	6
1.3	Indexes and Views.....	7
1.4	Procedures and Exception Handling in SQL server	9
Lab 2.	SQL Server 2012 Advanced Stretched Assignment	13
2.1	Indexes and Views.....	13
	Appendix A: Table Structure.....	14
	Appendix B: Table of Figures	17



Getting Started

Overview

This Lab book is a guided tour for Learning SQL server 2012. Each section contains some examples and assignments. Follow the steps provided in the solved examples and then work out the Assignments given.

Setup Checklist for SQL Server 2012

Here is what is expected on your machine in order for the lab to work.

Minimum System Requirements

Processor, HDD & RAM

- Processor - Minimum: AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support
- Processor speed: Minimum: 1.4 GHz
- Recommended: 2.0 GHz or faster
- RAM - Minimum: 512 MB, Recommended: 2.048 GB or more
- HDD – 150 GB

Operating System

- Windows XP Professional x64
- Windows 7 Professional 64 bit

SQL Server 2012 Developer Edition

- SQL server 2012 client and a SQL server 2012 Server instance running on the Server.

A database called Training will be available. All objects for the lab session would be stored in that database alone.

Lab 1. Getting connected to the SQL Server 2012 Server

1.1 Steps to connect to the SQL Server 2012 Server

Step 1: Click Start, Programs, Microsoft SQL Server 2012, SQL Server Management Studio.



Figure 1: Connecting to SQL Server 2012

Step 2:

Enter the Login, Password and the Server name provided to you.

Login: <loginid> Passwd: <password>

Step 3: Click on New Query.

1.2 Getting Familiar with SQL Server

1. Identify all the system and user defined database in your system.
2. Make master database as your current database , by using the command

```
Use <databasename>  
go
```

3. Find out if your active database is master ,by giving the command

```
Select DB_NAME()  
go
```

4. Now make Training database as your active database
5. Find out the content of the database by giving the following command. Observe the output

```
sp_help  
go
```

6. Repeat the above steps for master database and Northwind database
7. Find out the version of your SQL Server by giving the following command

```
Select @@version  
go
```

8. Find out the server date by giving the following commands

```
Select getdate()  
go
```

9. Make Northwind as your current database , find out information about tables using the command - Categories ,Products , Orders, Order Details , Employees

```
sp_help <tablename>  
go
```

10. Make a note of all related tables and foreign key columns
11. Repeat the above operation of Training database tables as well

1.3 Indexes and Views

1. Create a Unique index on Department Name for Department master Table.
2. Try inserting the following values and observe the output

Dept Code	Dept Name
100	Home Science
200	Home Science
300	NULL
400	NULL

3. Create a non-clustered index for Book_Trans table on the following columns
Boo_code, Staff_name, student name, date of issue. Try adding some values. Do you experience any difficulties?
4. List the indexes created in the previous questions, from the sysindexes table.
5. Create a View with the name StaffDetails_view with the following column name
Staff Code, Staff Name, Department Name, Desig Name salary
6. Try inserting some records in the view; Are you able to add records? Why not? Write your answers here.

7. Working with Filtered Index – The following Filtered Index created on
Production.BillOfMaterials table, cover queries that return the columns defined in
The index and that select only rows with a non-NULL value for EndDate.

```
USE Adventure Works;

GO
CREATE NONCLUSTERED INDEX FIBillOfMaterialsWithEndDate
ON Production.BillOfMaterials (ComponentID, StartDate)
WHERE EndDate IS NOT NULL;

GO
```

8. View the definition of the view using the following syntax.
Sp_helptext <viewname>
9. Using the view , List out all the staffs who have joined in the month of June



10. Create a non-clustered column store index on EmployeeID of Employees table

1.4 Procedures and Exception Handling in SQL server

1. Write a procedure that accept Staff_Code and updates the salary and store the old salary details in Staff_Master_Back (Staff_Master_Back has the same structure without any constraint) table. The procedure should return the updated salary as the return value

Exp < 2 then no Update

Exp >= 2 and <= 5 then 20% of salary

Exp > 5 then 25% of salary

2. Write a procedure to insert details into Book_Transaction table. Procedure should accept the book code and staff/student code. Date of issue is current date and the expected return date should be 10 days from the current date. If the expected return date falls on Saturday or Sunday, then it should be the next working day. Suitable exceptions should be handled.
3. Modify question 1 and display the results by specifying With result sets
4. Create a procedure that accepts the book code as parameter from the user. Display the details of the students/staff that have borrowed that book and has not returned the same. The following details should be displayed

Student/StaffCode Student/StaffName IssueDate Designation ExpectedRet_Date

5. Write a procedure to update the marks details in the Student_marks table. The following is the logic.
 - The procedure should accept student code , and marks as input parameter
 - Year should be the current year.
 - Student code cannot be null, but marks can be null.
 - Student code should exist in the student master.
 - The entering record should be unique ,i.e. no previous record should exist
 - Suitable exceptions should be raised and procedure should return -1.
 - IF the data is correct, it should be added in the Student marks table and a success value of 0 should be returned.

Working with THROW Statement

Task 1 – Raising and Catching an Exception

Now, we can use the **TestRethrow** table to force an exception. As you will see, the query runs successfully, but catches the error when attempting to insert the same primary key twice in the table, and shows an error message.

1. Copy and paste the following code segment in query editor.

SQL

```
USE Training;
BEGIN TRY
INSERT dbo.TestRethrow(ID) VALUES(1);
-- Force error 2627, Violation of PRIMARY KEY constraint to be raised.
INSERT dbo.TestRethrow(ID) VALUES(1);
END TRY
BEGIN CATCH
    PRINT 'In catch block.';
END CATCH;
```

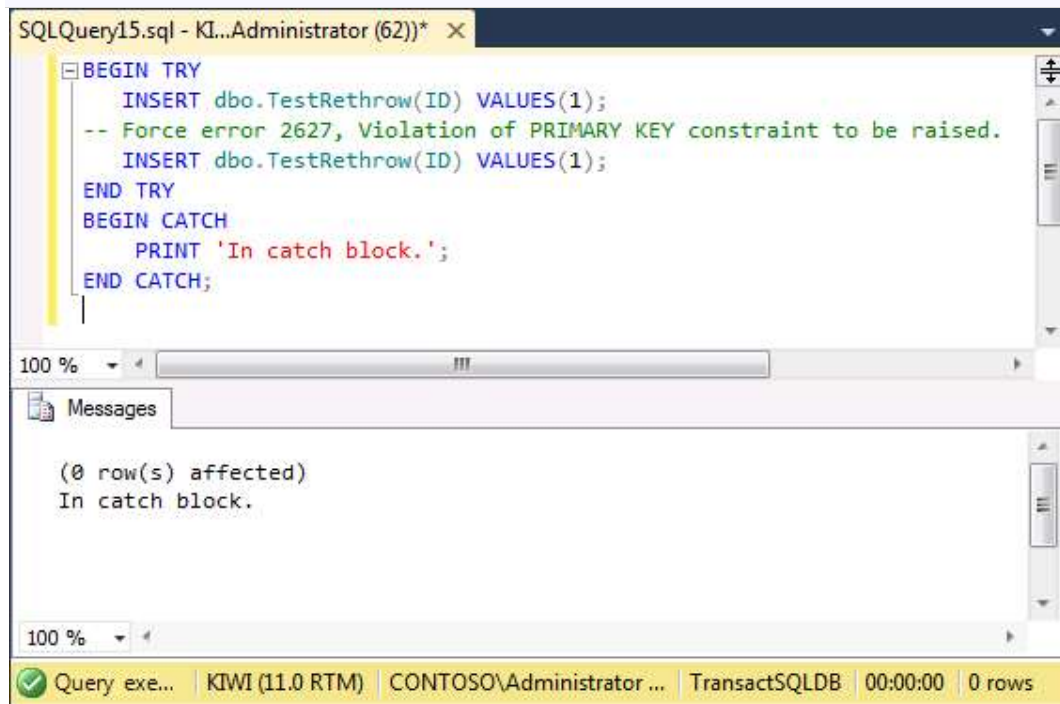


Figure 2: Attempting to insert the same row twice raises an exception



SQL SERVER ADVANCED LAB BOOK

Task 2 – Using Throw to Raise an Exception Again in a Catch Block

Finally, we can add a **Throw** statement in the **Catch** block. This can be useful when there is a chain of procedures executed, so exceptions are bubbled up.

1. Copy and paste the following code segment in query editor.

SQL

```
USE Training;
```

```
BEGIN TRY
```

```
INSERT dbo.TestRethrow(ID) VALUES(1);
```

```
-- Force error 2627, Violation of PRIMARY KEY constraint to be raised.
```

```
INSERT dbo.TestRethrow(ID) VALUES(1);
```

```
END TRY
```

```
BEGIN CATCH
```

```
    PRINT 'In catch block.';
```

```
    THROW;
```

```
END CATCH;
```

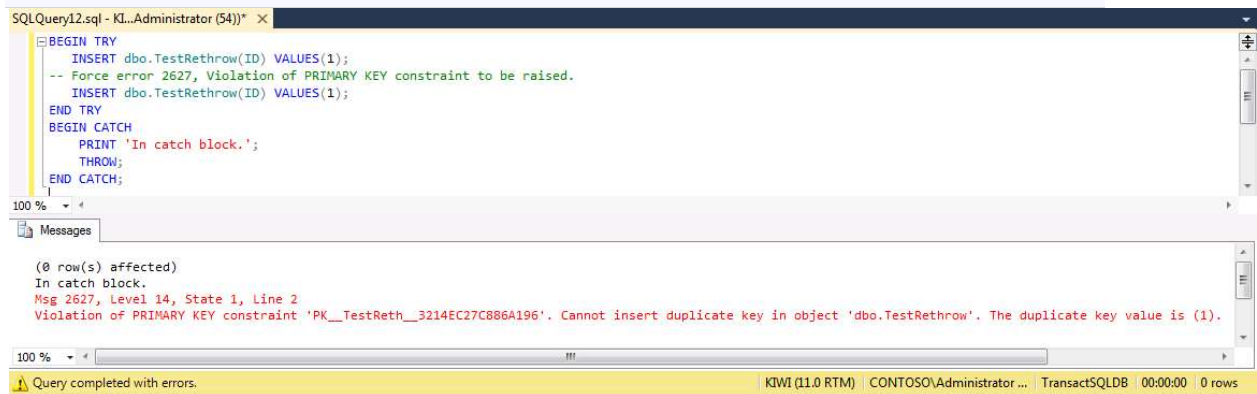


Figure 3: Re-throwing the exception shows actual error, & the query completes with errors



Lab 2. SQL Server 2012 Advanced Stretched Assignment

2.1 Indexes and Views

1. Create a Filtered Index HumanResources.Employee table present in the AdventureWorks database for the column EmployeeID. The index should cover all the queries that uses EmployeeID for its search & that select only rows with "Marketing Manager" for Title column.

Appendices

Appendix A: Table Structure

Desig_Master

Name	Null?	Type
<u>Design_code</u>	Not Null	int
Design_name		Varchar(50)

Department_Master

Name	Null?	Type
<u>Dept_Code</u>	Not Null	int
Dept_name		Varchar(50)

Student_Master Table

Name	Null?	Type	
<u>Student_Code</u>	Not Null	int	
Student_name	Not Null	Varchar2(50)	
Dept_Code		int	FK ->Dept_Master
Student_dob		Datetime	
Student_Address		Varchar(240)	

Student_Marks

Name	Null?	Type	
<u>Student_Code</u>		int	FK->Student_master
<u>Student_Year</u>	Not Null	int	
Subject1		int	
Subject2		int	
Subject3		int	

Staff_Master

Name	Null?	Type	
<u>Staff_code</u>	Not Null	int	
Staff_Name	Not Null	Varchar(50)	
Design_code		int	FK->Design_master
Dept_code		int	FK->Dept_Master
HireDate		Datetime	
Staff_dob		Datetime	
Staff_address		Varchar(240)	
Mgr_code		int	
Staff_sal		decimal (10,2)	

Book_Master

Name	Null?	Type
<u>Book_Code</u>	Not Null	int
Book_Name	Not Null	Varchar(50)
Book_pub_year		int
Book_pub_author	Not Null	Varchar(50)
Book_category	Not null	Varchar(10)

Book_Transaction

Name	Null?	Type	
<u>Book_Code</u>		int	Fk ->Book_master
<u>Student_code</u>	Null	int	FK->Student_master
<u>Staff_code</u>	Null	int	FK->Staff_master
<u>Book_Issue_date</u>	Not Null	Datetime	
Book_expected_return_date	Not Null	Datetime	
Book_actual_return_date	Null	Datetime	



SQL SERVER ADVANCED LAB BOOK



Appendix B: Table of Figures

Figure 1: Connecting to SQL Server 2012	5
Figure 3: Attempting to insert the same row twice raises an exception	10
Figure 4: Re-throwing the exception shows actual error, & the query completes with errors	12