

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Incremental Schema Upgrade Methodology for Microsoft SQL Database

Yogesh S Patil *	Priti Subramanium
Department of Computer Science & Engineering	Department of Computer Science & Engineering
Student (M.E), S. S. G. B. C. O. E. & Technology,	Asst. Professor, S. S. G. B. C. O. E. & Technology,
Bhusawal, India	Bhusawal, India
yogeshe146@gmail.com	pritikanna559@gmail.com

Abstract: There are many problems associated, when we upgrade our product database schema to next version. Many times it is found that there is data corruption or we find that data integrity and consistency is lost. There are many solutions/methods, but each method has its own drawback to overcome these problems. We have suggested an incremental methodology to upgrade a database schema from one version to next version having no data corruption and maintaining the database consistency and integrity.

Keywords: schema upgrade, schema migration, Schema Evolution, Object database, schema integration.

I. INTRODUCTION

A simple method of performing a schema migration is to execute SQL script files before or after the software upgrade [1]. It is painstaking aspects of to updating a deployed application and to transforming existing data [2].The life cycle of any product consists of many releases. For e.g. at the beginning version 1.0 of a product with limited or targeted functionality is released to the customer. While customer start evaluating/using the product, the developers start working on enhancing, add new functionality and fixing issues of version 1.0 of product to develop next version of the product. This cycle goes on and newer version of product are released to the customer like (version 2.0, 3.0 etc). This creates the need for upgrade of product from lower version to later versions.

With this approach of development there are following challenges to be address:

- a. Upgrade of code files (class files for Java/ dlls for C#.net).
- b. Upgrade of database without the loss or corruption of data.

This methodology provides a complete guidance about how upgrade of database can be achieved.

II. DATABASE SCHEMA

A problem in building and maintaining large, long-lived application systems is to cope with all the changes that inevitably will be imposed on the systems over time. Numbers of large application systems are centred on a database [3]. Schema evolution is one of the most important aspects of the use of a database, because once the database has been constructed, users' requirements tend to change [4].In the simplest definition, a database is a collection of data. Software applications that are designed to help create, maintain and use databases are called database management systems (DBMS). DBMSs provide abstractions by which the logical model of the data can be separated from the way it is physically stored. A metadata evolution technology has been use in which they realize the mapping from the old data schema to new data schema smoothly during schema upgrade. So the schema upgrade can be completed without data migration [5].

A Schema extraction and evolution method has been used when they release any history of an application, they start extracts the database schemas embedded in the application and compare the schemas, and produce a tally of schema evolution results [6]. Any update to be executed we have to check all integrity constraints with efficient integrity methods [7].The dependencies between attributes within tables are measure using an information-theoretic .Then they construct a dependency graph for each table capturing the dependencies among attributes after then they find matching node pairs across the dependency graphs by running a graph-matching algorithm [8].

III. IMPLEMENTATION

Upgrade from source schema version to target schema version should work smoothly. First take out the checksum of target database schema and name it as target checksum. Then manually write upgrade scripts once we complete writing the upgrade script then apply this upgrade scripts sequentially on the old database schema and calculate its checksum and name it as upgrade checksum. If target checksum value is equal to upgrade checksum value then we can say that the database schema is successfully upgraded to target version. Figures 1 show overview of Incremental Schema Upgrade Methodology.



Target checksum= Upgraded Checksum

Figure 1 Incremental Schema Upgrade Methodology

IV. ALGORITHM

- a. Set database name and path of upgrade scripts and checksum=false.
- b. Take the backup of the existing database using the new Product's script for backup and restore.
- c. Sort all the upgrade scripts from the location specified in path of upgrade scripts option and start applying them sequentially on the target database.
- d. Perform steps 5, 6 and 7 only if checksum flag is true
- e. Calculate the checksum of the database schema after applying each script and compare it with the checksum stored into that script file.
- f. If upgrade script does not have a pre-calculated checksum in it, then upgrade should fail and exit.
- g. If checksum matches then it verifies that the script is applied correctly.
- h. Once that is verified, add a new row should be added Product's_DBversion table with the schema version for that script (mapping and sp versions will not change in that case).
- i. Repeat steps 4 and 8
- j. If checksum does not match then upgrade should fail and exits by restoring the database to the original.
- Once all the scripts are applied, all stored procedures, views and mapping files (except UDA mapping files) should be copied to target database.

V. EXPERIMENTAL RESULT

To determine whether our methodology work or not we have perform a simple experiment in which we have develop a simple software of book search system with limited functionality in first version the database name given as bss db in this database we have create following tables Author, Subject, Book. In Author table we have following Field Name, Address, email id, contact and author id .In subject table we have Name, Description and subject id field. In Book table we have Title, Book_id, Description, author_id, subject_id and Price field. After than we have modified the existing software of book search system by adding more functionality to it for example in Author table we have added age field. In Book table we have added publisher id and stream id field and drop the field subject_id. Rename table Subjects to Streams and added one table of Publisher with following field publisher_id, Name, Address, and Contact. We have make publisher_id as foreign key in Book table and also Rename subject_id column to stream_id. Known when we want to move our database bss_db to next version we have written upgrade scripts for every modification that we have done in first version the upgrade scripts are as follows:-

A. Upgrade script to add table Publisher:

CREATE TABLE [dbo].[Publisher]([publisher_id] [int] IDENTITY(1,1) NOT NULL, [Name] [nvarchar](1000) COLLATE Latin1_General_CI_AS NULL, [Address] [nvarchar](1000) COLLATE Latin1_General_CI_AS NULL, [Contact] [nvarchar](1000) COLLATE Latin1_General_CI_AS NULL, CONSTRAINT [PK_Publisher] PRIMARY KEY CLUSTERED (

[publisher_id] ASC)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]) ON [PRIMARY] Go

B. Upgrade script to add age column:

Alter table author add age int; go

C. Upgrade script to add Publisher_id column:

alter table book add publisher_id int go

D. Upgrade script to make publisher_id as foreign key:

alter table book

ADD CONSTRAINT fk_Publisher FOREIGN KEY (publisher_id) REFERENCES dbo.Publisher (publisher_id) Go

E. Upgrade script to Rename table Subjects to Streams:

sp_Rename 'Subject', 'Streams'

F. Upgrade script to Rename subject_id column to stream_id:

sp_rename 'streams.subject_id', 'stream_id' go

sp_rename 'book.subject_id', 'stream_id' go

Now before running this upgrade scripts when have taken backup of existing database that is bss_db in backup file and then we have calculated checksum value of database schema of both version that is bss_db for first version and bss_db2 for second version as shown in Figure 2 We found that both values are different, then we have run upgrade scripts and taken checksum value again and we found that both value are same as shown in Figure 3 We have seen that database as been successfully upgraded to next version. The Figure 4 to Figure 10 shows database table before and after upgrade scripts run.

Administrates (AWindows) autors 23 and ave	_ D X
Administratori C:(windows/systems2)(cmd.exe	
Microsoft Windous [Version 6.0.6001] Copyright (c) 2006 Microsoft Corporation. All rights reserved.	^
C:\Users\hansraj>cd	
C:\Users/cd	
C:\>cd me project 2012	
C:∖ME Project 2012>cd hansraj	
C:\ME Project 2012\Hansraj>DBUtils.exe /ac:checksum/db:bss_db	
C:\ME Project 2012\Hansraj)DBUtils.exe /ac:checksun /db:bss_db Checksun of database bss_db = 6455252	
C:\ME Project 2012\Hansraj\DBUtils.exe /ac:checksun /db:bss_db2 Checksun of database bss_db2 = 7363835	
C:\ME Project 2012\Hansraj>DBUtils.exe /ac:upgrade /db:bss_db /sn /osp:"c:\ME Project 2012\Hansraj\Book Search 1.0\Booksearcnsystem\Upgra s"_	localhost de Script
	-

Figure 2 Checksum value of database schema before upgrade scripts run.

go



Figure 3 Checksum value of database schema after upgrade scripts run.

File Edit View Project Query Designer Tools V	Window Comm	unity	Help					
👤 New Query 🐚 📸 😚 🚱 🔓 🖉 🖉 🖉	IBBS							
Change Typer 🖗 (E 🕇 🗃		D.						
Dbject Explorer	• ‡ X	Ta	ole - dbo.Author	Table - dbo.Author	HANSRAJ-PC.ma	stestore1_008.sql		;)
Connect 🛛 🛃 🔳 🗿 🍸			Name	Address	enaitD	contact	author_id	
🗄 🔒 HANSRAJ-PC (SQL Server 9.0.1399 - sa)	A)	Yogesh	Bhusaval	ys@gmail.com	962336374	1	
🗄 📄 Databases			Praful	shirpur	ps@gmail.com	9898765423	2	
🗄 📋 System Databases			sharad	latur	ps@gmail.com	9898722223	3	
🗄 📄 Database Snapshots			vijay	bhusaval	vijay@hotnail.com	9850652855	4	
⊞ 📙 BSS_DB			newAuthor	asdasd	asdasd	987898	6	
⊞ U BSS_U62	-		hans	hth	hth	hhh	7	
H OSSLU	2		sdfs	sdfsfd	sfsdf	sdfsdf	8	
Database Diagrams			rahul	bhusawal	rahul@omail.com	1234354232	10	
🛛 🛄 Tables			iadison	america	ißonal.com	3214	12	
🗑 📄 System Tables			Gries D	Pune	G@onal.com	aune	16	
🗊 🔳 dbo.Author			Welsh 1	dehi	w@mai.com	4356543267	18	
🗄 🔲 dbo.Book			Welch 1	husaval	w@mail.com	6543256432	19	
🛞 🛄 dbo.Subject			Ornanick ET	D DP	odemai.com	nine	20	
i i i i i i i i i i i i i i i i i i i			Heian D K	nachik	h@mail.com	242855	21	
a 📴 Synonyms 🛙 🗇 Programmability			Darant M	0.09	offernal com	547765	21	-
🗑 🛄 Service Broker		((ran u	ur	1	.nonu)
🗑 📋 Storage		1	(1 of!	503))))= (i				

Figure: 4 Author table before upgrade scripts run.

Fle Edit View Project QueryDesigner Tools Window	v Commu 🖪 隆 🖷	nity Help						
	pr (yr 🗉	8						
hiert Funkter	, 1 X	Table - dbo Author	Gummani					
ionnect 🔹 🔛 👔 🏹	4	Nane	Address	enaiD	contact	author id	80e	-
R HANSRAI-PC (SQL Server Q.0.1399 - ca)		Yogesh	Bhusaval	ys@gnail.con	962336374	1	NUL	
B Databases		Praful	shirpur	ps@gmail.com	9898765423	2	NUL	
🗑 📄 System Databases		sharad	latur	ps@gmail.com	9898722223	3	NUL	
🗑 📄 Database Snapshots		vijay	bhusaval	vijay@hotnail.com	9850652855	4	NUL	
₩ U BSS_DB		newAuthor	asdasd	asdasd	987898	6	NUL	
B 0 800_062	:	hans	hth	łłh	hhh	1	NUL	
H bss1 0 Data		stfs	sdfsfd	sfsdf	sdfsdf	8	NUL	
🛛 📔 BSS1_Obakup		rahul	bhusawal	rahul@gmail.com	1234354232	10	NUL	
🗑 📄 Database Diagrams		jadison	anerica	j@gmail.com	3214	12	NUL	
🛛 📋 Tables		Gries D	Pune	G@gmail.com	pune	16	NUL	
🗑 🛄 System Tables		Welsh J	dehi	w@gnail.com	4356543267	18	NUL	
i i i i i i i i i i i i i i i i i i i		Welsh J	bhusawal	w@gnail.com	6543256432	19	NUL	
🗑 🧧 do.Publisher		Organick E I	pune	o@gmail.com	pune	20	NUL	
🗉 🔳 dbo.Streams		Hsiao D K	nashik	h@gmail.com	243855	21	NUL	
🗄 🧰 Views		Parent M	pune	p@gmail.com	543265	22	NUL	
🗄 🛄 Synonyms 🗟 👕 Programmability	Ţ.	((1 of	503))		rimti	m		

Figure: 5 Author table after upgrade scripts run.

🕽 New Quey 🔓 🖓 🖓 🖓 🚱 🔓 🕼 🖉 🖗	1994	÷							
🖥 🖬 🖉 Change Type 🕴 👘 🕼 🛔	-11	1.1	I. JL. D. I. (TJ		T.U. J. R.K.	L.) T.U. J.	. D I. T.LI. JL	. A. 41-12	
Connect 🛛 🛃 🔳 🕅 🍸	• • •	140	ne • obolbook Tai	Book id	Description	author id	o.boox 1801e - op subject id	Price	_
		•	Networks System	1	big book on net	2	3	405	_
R Databases	1		software	2	engineering	1	1	259	
🗄 🛅 System Databases			Compiler Constr	16	complier	385	6	345	
🖁 📜 Database Snapshots			Sequential Progr	17	program	371	9	435	
⊞ 📙 BSS_DB			Sequential Progr	18	program	371	76	342	
⊞ <mark> </mark> BSS_DB2	:		Programmers Vie	19	electronic	369	102	547	
H OSSLU			Advanced Datab	20	computer	370	68	450	
R Database Diagrams			Logic and Progra	21	computer	421	74	453	
🖯 🛄 Tables			Introduction to	22	computer	421	85	234	
🗄 📋 System Tables			Multiprocessor O	23	computer	471	91	760	
🗑 📃 dba.Author	_		Distributed Data	25	computer	110	123	345	
🗑 🗐 doo Book			Word Processina	26	computer	42	9	342	
H 🗐 add.sudjett D 🗇 Views			Fundamentals of	27	computer	142	74	670	
I Synonyms			Introduction to	28	computer	273	73	125	
Programmability			A Structured Dis.	29	corrouter	53	78	456	
🗄 🛅 Senvice Broker			n	11		w	103	/m	

Figure: 6 Book table before upgrade scripts run.

Yogesh S Patilet al, International Journal of Advanced Research in Computer Science, 4 (4), March - April, 2013,114-118

Re Edit View Project QueryDesigner Tools Window Wen Query Di B B B D D B D D D D D D D D D D D D	Comm Comm	unity E	Help							
a a a l'orande lite. I a le la a a pet£nque	ļχ	1	hle•dbo.Book Ta	ble - dbo.Author	r Summary					•)
Connect 🚦 🛛 🖞			Title	Bookjid	Description	authorjid	stean_it	Price	publisher_id	
HAN&RAI-PC (SQL Server 9.0.1399 - sa)	٨)	Networks System	1	big book on n <mark>et</mark>	2	3	405	MUL	-
🛛 🗋 Databases	٦		software	1	engineering	1	1	259	NUL	
🖲 词 System Databæes			Compiler Constr	ъ	compler	385	6	345	NUL	
🖲 🛅 Database Snapshots			Sequential Progr	1	program	371	9	435	NUL	
∎ <mark> </mark> BSS_DB = ⊖ BSS_DB			Sequential Progr	8	program	371	76	342	NUL	
B B Pri D			Programmers Vie	9	electronic	369	102	547	NUL	
a i bst 0 Data			Advanced Datab	20	computer	370	68	450	NUL	
e 🛛 BSSI Obakup			Logic and Progra	21	computer	Q1	74	453	NUL	
🗑 📋 Database Diagrams			Introduction to	12	computer	Q1	85	234	MUL	
🖯 🛄 Tables			Nultiprocessor 0	3	computer	971	91	760	NUL	
🖲 📜 System Tables			Distributed Data	35	computer	110	123	345	NUL	
🗑 📃 doo Auchor			Word Processing	36	computer	Q	9	342	NUL	
🛛 🖬 doc Publisher			Fundamentals of	7	computer	142	74	670	MIL	
🗑 📃 dbo.Streems			Introduction to	38	computer	273	73	125	MUL	
🗑 📋 Views			A Structured Dis	3	computer	53	78	456	NUL	
e 📄 Synonyms e 🗎 Precrammakility		1	n	, ,))):		w	m	<i>m</i>	un	

Figure: 7 Book table after upgrade scripts run.

Microsoft SQL Server Management Studio	L						
File Edit View Project Query Designer Tools Window (lomn	nunity	Help				
🖞 New Query 🔥 🖓 🖓 🏠 🍃 🎜 🖉 🖗 🦉		2					
回目回目 (hanne Tune y 上風) (日午初)	-						
Chiert Evolution	v	173	L. H. D. P.L.	(T.L. U. N.L.	T.I.I. JL. A.A.		
Vujeti cipilie 🖣 🖉	^	<u> </u>	ie - abo.rubisner	19016 - 0001900K	Table - obo.Autno	r Summary	• X
Connect* 🛃 🛛 🖉 Y			publisher_id	Name	Address	Contact	
🗄 🚺 HANSRAI-PC (SQL Server 9.0.1399 - sa)	٨	*	NUL	MUL	MUL	MUL	
🗄 📄 Databases							
🗄 🛅 System Databases							
🗄 🛄 Database Snapshots							
H U BSS_DB							
⊞ <mark> </mark> BSS_DB2							
Berline Contraction	1						
B Dost Q Data							
B Databas Diaman							
🗄 🛄 Uatabase Ulagrams							
🗃 🛄 Tables							
🗑 🛄 System Tables							
🗉 🖬 doo Rook							
E doo Publisher							
I dbo.Streams							
I Tiews							
🗄 📋 Synonyms							
🗄 🧰 Programmability	•	10	0 of O)))))))))))))))))))))))))))))))))))))))			
Ready							
nooj	_		_	_	_		

Figure: 8 Publisher table created after upgrade scripts run.

😓 Microsoft SQL Server Management Studio						DX
File Edit View Project QueryDesigner Tools Window Ci	omm	unity	Help			
🖞 New Query 🔥 😚 😚 🚱 🕞 💕 🗐 🖉 🧎 🦉	1					
回用回用 (hanne Tuney) 同作 告約	-					
	v	<i>(</i> 1)				
	*	labl	e - dbo.Subject	able - dbo.Book		₹X
Connect 🛛 🛃 🔳 📓 🍸			Name	Description	subject_id	_ ^
🗄 🐻 HANSRAJ-PC (SQL Server 9.0.1399 - sa)	A)	Science	science and tech	1	
🛛 📄 Databases			Georaphy	geography	2	5
🗄 📜 System Databases			Mechanical	books related to	3	
🗄 📄 Database Snapshots			Aeronautics	air crafts	4	
🗑 📙 BSS_DB			biology	biology	5	
₩ 📙 BSS_DB2			computer	entineering	6	
₩ <mark> </mark> bss1_0			- laborate	engineering alaasta	-	
□ U bss1_0_Data			ectronic	electric	1	
Database Diagrams			computer	security	9	
Tables			Complier contruc	computer	67	
B j System Tables			ACA	computer	68	
🗑 🛄 dbo.Author	-		Software engg	computer	69	
🔛 💷 000.500K			Math-I	computer	70	
I I I I I I I I I I I I I I I I I I I			Math-II	computer	71	
Synonyms			Math-III	computer	72	
🗄 🛅 Programmability			Proramming Lab-I	computer	73	
🗄 🛅 Service Broker			n		74	
🗄 🛄 Storage	Ŧ		1 of 69))))		
Danda						
Incouy	_			_		đ

Figure: 9 Subject table before upgrade scripts run.

Microsoft SQL Server Management Studio			N. 1		
File Edit View Project Query Designer Tools Window Cr	ommur	ity Help			
일 New Query 👔 🔒 🚱 🍒 🔓 🔐 🖉 🕼 🗿 🦉	1				
🛛 🗐 🖉 🖉 Change Type 🔹 🐂 🎼 🙀					
Object Explorer 🔹 🖡	X	Table - dbo.Streams	Table - dbo.Publisł	ner Table - dbo.Book	₹x
Connect 🛛 🛃 🔳 📓 🍸		Name	Description	stream_id	٨
🗑 🚹 HANSRAJ-PC (SQL Server 9.0.1399 - sa)	.)	Science	science and tech	1	
🗄 🛄 Databases		Georaphy	geography	2	E
🗑 🛄 System Databases		Mechanical	books related to	3	
🗑 🛄 Database Snapshots		Aeronautics	air crafts	4	
₩ U BSS_DB		biology	biology	5	
	:	computer	engineering	6	
H OSSLV		elctronic	electric	7	
BSS1 Obakup		computer	security	9	
🖩 🧊 Database Diagrams		Complier contruc	computer	67	
🗄 🛄 Tables		404	computer	68	
😠 📄 System Tables	4	Software enon	computer	69	
🗄 📃 dbo.Author		Math.I	computer	20	
🗑 🗐 dbo.Book		Math.IT	computer	71	
🗄 🔲 dbo.Publisher		Math.III	computer	71	
e 🖬 vouvarcente a 🛅 Views		MdUI-111	computer	72	
🖩 🧰 Svnorvms		Proramming Lab-1	computer	10	,
n 🛄 Programmability		(1 of 69			
	<u> </u>	-			
neady	_				

Figure: 10 Subject table became streams table after upgrade scripts run.

VI. ADVANTAGES

- Maintain relationships among attributes, and the conditions for maintaining consistent views of programs.
- b. Each schema version is expressed as a (database) view which constitutes actual interface to the application programs.
- c. Schema changes are made transparent to the application programs.
- d. No need of extra data cleanup.
- e. Easy to find out if any schema written wrong.

VII. CONCLUSION

Finally we conclude that our incremental methodology to upgrade database schema solve the problem associated with relational database upgrade to enterprise software. In future we can work on analyzing different methodology including my methodology that are available to upgrade data base schema and try to make such software that will directly write SQL script for us.

VIII. REFERENCES

 Robert M. Marks," A Metadata Driven Approach to Performing Multi-Vendor Database Schema Upgrades",' 19th IEEE International Conference and Workshops on Engineering of Computer-Based Systems', 2012, PP 108-116.

- [2]. Praveena Mandapati , U. Tulasi ," Extended Schema Extraction and Evolution",' IEEE International Journal of Emerging Technology and Advanced Engineering', 2012, Vol. No. 2, PP 220-224.
- [3]. Dag.Sjøberg, "Quantifying schema evolution," in Information and Software Technology, vol. 35, no. 1, January 1993, PP 35–44.
- [4]. Junichi Takahashi, "Hybrid Relations for Database Schema Evolution",' Fourteenth Annual International IEEE International Conference on Computer Software and Applications', 1990, PP 465-470.
- [5]. Hui Liu, Shengqi Wu ," Data Storage Schema Upgrade via Metadata Evolution in SaaS ",' IEEE International Conference on Consumer Electronics, Communications & Networks (CECNet)', 2012, PP 3148-3151.
- [6]. Shengfeng Wu, Iulian Neamtiu," Schema Evolution Analysis for Embedded Databases ",' IEEE International Conference on Data Engineering Workshops (ICDEW)', 2011, PP 151-156.
- [7]. Hendrik Decker, Davide Martinenghi, "Inconsistency-Tolerant Integrity Checking ",' IEEE Transactions on Knowledge and Data Engineering ', Vol. 23, No. 2, February 2011.PP 218-234.
- [8]. Jaewoo Kang ,Jeffrey F. Naughton," Schema Matching Using Interattribute Dependencies ",' IEEE Transactions on Knowledge and Data Engineering ', Vol. 20, No. 10, October 2008. PP 1393-1407.