WHO, WHAT AND WHEN

TEMPORAL TABLES IN SQL SERVER

KEVIN BELL
VIRTUAL DEVCON 2020 APRIL 24, 2020

SQL SERVER AS A BACKEND?

- Peter has set up a quick survey… link in the chat window

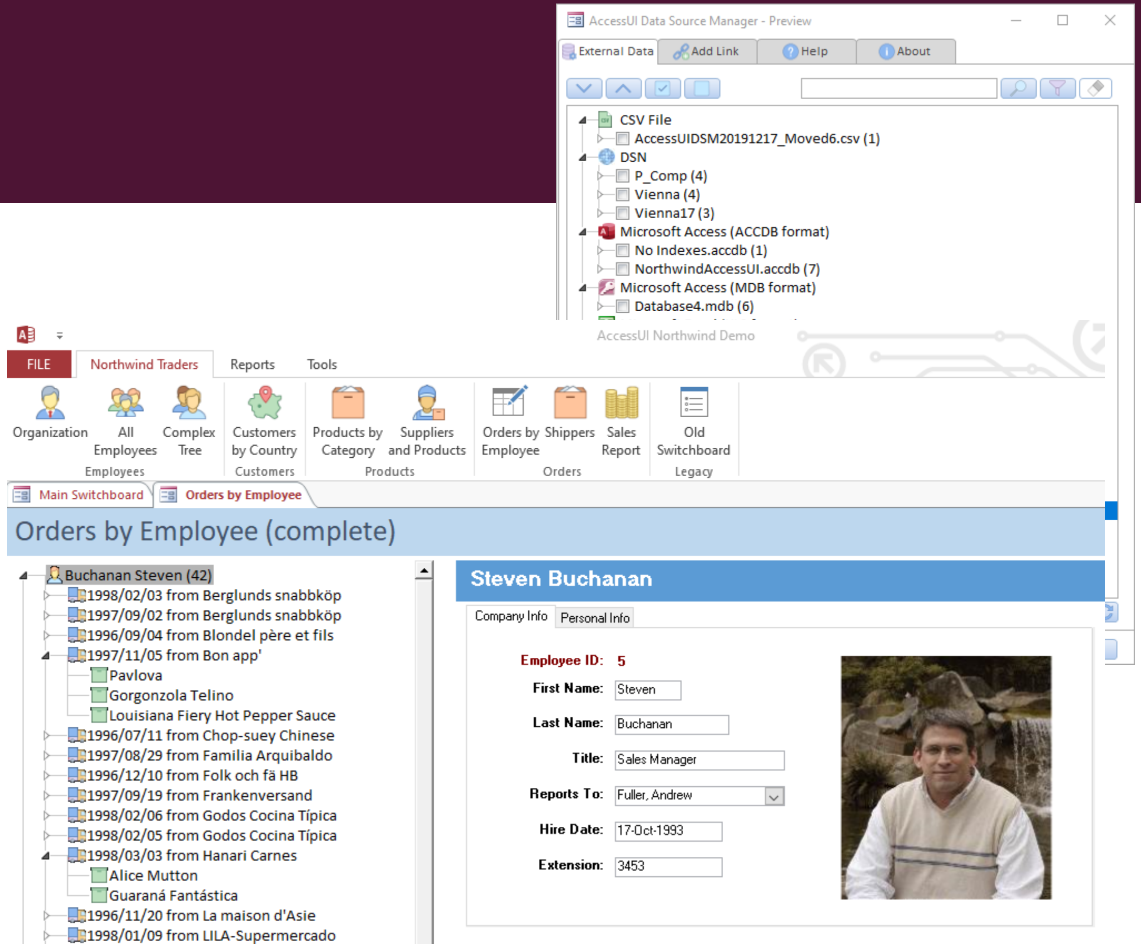
- How often do you use SQL Server as a back-end? (or other ANSI SQL implementation)
  - Almost all the time (>80%)
  - Some of the time (>40%)
  - Rarely (<40%)
  - Never

A LITTLE ABOUT ME

- “Developing” with Microsoft Access since version 1.0
- Started using SQL Server with a beta version of Access 2.0
- Ran a small consulting firm in Colorado for 15+ years
- Joined the Microsoft Access team as a test engineer in 2008 and worked on three release of Access (2010, 2013 and 2016)
- Been fortunate to present at Access events all over the world
- Currently back consulting, specializing in migrating Access apps to the cloud (Access/SQL Azure/.Net Core MVC)
- Enjoy playing baseball and traveling the world looking for a place to semi-retire (with good beer)

ACCESS ADD-INS

- AccessUI Data Source Manager
  - Free
- AccessUI Ribbon & Tree Builder
  - Demo in the last session 18:00 UTC (3 hours from now)
  - Discount code DevCon2020 will lower the price to $79 until May 31 2020
  - https://accessui.com/eventcode



CSV File
AccessUIDSM20191217_Moved6.csv (1)
DSN
P_Comp (4)
Vienna (4)
Vienna17 (3)
Microsoft Access (ACCDB format)
No Indexes.accdb (1)
NorthwindAccessUI.accdb (7)
Microsoft Access (MDB format)
Database4.mdb (6)

Orders by Employee (complete)

Buchanan Steven (42)
1998/02/03 from Berglunds snabbköp
1997/09/02 from Berglunds snabbköp
1996/09/04 from Blondel père et fils
1997/11/05 from Bon app'
Pavlova
Gorgonzola Telino
Louisiana Fiery Hot Pepper Sauce
1996/07/11 from Chop-suey Chinese
1997/08/29 from Familia Arquibaldo
1996/12/10 from Folk och fä HB
1997/09/19 from Frankenversand
1998/02/06 from Godos Cocina Típica
1998/02/05 from Godos Cocina Típica
1998/03/03 from Hanari Carnes
Alice Mutton
Guaraná Fantástica
1996/11/20 from La maison d'Asie
1998/01/09 from LILA-Supermercado

Steven Buchanan

Employee ID: 5
First Name: Steven
Last Name: Buchanan
Title: Sales Manager
Reports To: Fuller, Andrew
Hire Date: 17-Oct-1993
Extension: 3453

CORONA… NOT THE BEER

- Seattle was the first hot spot in the US
- We had a trip planned to Portugal last month
- About half way to Amsterdam I learned of the pending travel ban
- Once we landed, got our tickets change and took the same plane back to Seattle
- We had 30 minutes until boarding, so we had a very expensive Heineken, then flew back

SURVEY SAYS…

WHO, WHAT & WHEN

- Tracking who changed what records, and when
- Who – what user or process changed the data
  - Help identify user training or malicious users
- What – the data that has changed
  - Usually log the entire row,
  - I have seen systems log only the fields that have changed
    - Seems like a lot of extra work to get meaningful results, but can save storage space
- When – the time a record was edited or deleted
  - Server time in UTC is the most flexible

WHY TRACK A RECORDS HISTORY?

- Auditing, regulatory compliance and data forensics
- “Undo” – historical records can be re-applied
- Analyzing data trends – what table and records get edited the most
- Repairing low level data corruption
- **Point in time reports**

WHO HAS HAD TO DO THIS?

- Rhetorical question… I'm sure most everyone has been ask to implement this at some level
- Largely a pain in the a$$

ACCESS MDB FILE

- No way to enforce at the table level
- Must add logic on every form
- Must keep table, history table and code in sync
- Who
  - User Level "Security"
  - Windows Login
  - Roll your own

ACCESS ACCDB FILE

- Starting with Access 2010 you can use data macros to enforce at the table level
- ACCDB is not as scalable as SQL Server
- Must keep table, history table and data macro in sync
- Who
  - No longer ULS
  - Window Login
  - Roll your own
- Sample files has VBA code that will build data macros for all tables in a database

SQL SERVER TRIGGERS

- Enforced at the table level
- Way more scalable that ACCDB
- Triggers are "hidden" and easily forgotten about
  - Seems Trigger have fallen out of favor?
- Must keep table, history table and trigger in sync
- Who
  - Must come from SQL Server
    - Problematic with a generic login

CDC AND CT - SQL SERVER 2008

- Change Data Capture
  - Records INSERTs, UPDATEs and DELETEs in a SQL table
  - dbo.MyTable >> cdc.dbo_MyTable_CT
  - https://www.red-gate.com/simple-talk/sql/learn-sql-server/introduction-to-change-data-capture-cdc-in-sql-server-2008/
- Change Tracking
  - Identifies the rows that have changed, but doesn’t provide information about the values that were changed
  - Uses an in-memory rowstore, flushed on every checkpoint
  - https://solutioncenter.apexsql.com/what-is-change-tracking-and-how-to-set-it-up/

TEMPORAL TABLES - SQL SERVER 2016

- Temporal Tables were introduced in the ANSI SQL 2011 standard
- Implemented in SQL Server 2016 as System-Versioned tables
- Enforced at the table level
- “Nothing” to maintain
  - You can implement this in an afternoon!
- Allows for time travel

TEMPORAL TABLES REQUIREMENTS

- Tables must have
  - Primary key
  - Two time period columns defined as datetime2 data type
  - Have their history table in the same database
    - Can be in a different schema
- Tables can not have
  - DELETE and UPDATE CASCADE if the table is in a FK relation
  - INSTEAD_OF triggers
- History table can not have any constrains (PK, FK, DEFAULTS, etc.)

TEMPORAL TABLE CAUTIONS

- Temporal tables support large data types (varbinary, varchar, nvarchar)
  - That could lead to increased storage costs and potential performance issues
  - Might need to redesign table structure to minimize bloat
- Possible bug with Access updating or deleting records - reported by Access MVP Tom van Stiphout
  - I've not seen it with Azure
  - Doesn't appear to be on prem vs. Azure
  - Doesn't appear to be a driver issue
  - Might be the implementation in the sample Word Wide Importers database

THE WHAT AND THE WHEN

- What – Turning on System Versioning and SQL will take care of the rest
- When – You do need to add a FROM date and a TO date to each table
  - Need to be datetime2
- Demo!

THE WHAT AND WHERE SCRIPT

-
  ```
  ALTER TABLE tableName ADD
  ValidFromUTC datetime2 (2) GENERATED ALWAYS AS ROW START HIDDEN
    constraint DF_tableName_ValidFromUTC DEFAULT SYSUTCDATETIME() ,
  ValidToUTC datetime2 (2)  GENERATED ALWAYS AS ROW END HIDDEN
  constraint  DF_tableName_ValidToUTC DEFAULT '9999.12.31 23:59:59.99‘,
  PERIOD FOR SYSTEM_TIME (ValidFromUTC,ValidToUTC)
  ```

-
  ```
  ALTER TABLE tableName
  SET (SYSTEM_VERSIONING = ON (HISTORY_TABLE = dbo.tableNameHistory));
  ```

LOGICAL DELETES

- Logical delete is using a bit or datetime field to show the record is deleted, without physically deleting the record
- With temporal tables you **may not** need logical deletes anymore
  - I've been bitten by not filtering out a "deleted" record once… it was messy

THE WHO… NOT THE BAND

- You have to be clever and add your own fields to track the Who
- Add a created by field that defaults to SUSER_SNAME()
- Add a modified by calculated field that equals SUSER_SNAME()
- Demo!

THE WHO - SCRIPT

- ALTER TABLE *tableName* ADD
  CreatedBy NVARCHAR(128) NOT NULL DEFAULT (SUSER_SNAME()),
  ModifiedBy AS (SUSER_SNAME());

- You want to add both the When and the Who fields in the same script
  - My spAddHistory Stored Procedure add the field and enables System Versioning
- Former Access MVP Klaus Oberdallhoff has some much more advanced spocs
  - https://1drv.ms/u/s!Am2T4hGcNbbMiZMoOV7w-OoZUX-oOw?e=IeluLl

SQL SERVER LOGIN FROM ACCESS

- Should not store user SQL passwords in Access (DSN or DSN-Less)
- Use DSN – Less connection for all your linked tables and views
- When creating the connection, don't include the user name or password
- When Access first opens a connection to a table or view
  - Access/SQL will prompt you for a user name and password
  - That connection will be cached for the rest of the Access session, and other connection to the same server/database will use the cached connection
  - Optionally use a form to get the user name and password, create a temp QueryDef and append user info to the Connect of a linked table, OpenRecordset then close recordset
- Blog post from former MVP Ben Clothier https://www.microsoft.com/en-us/microsoft-365/blog/2011/04/08/power-tip-improve-the-security-of-database-connections/

ADDING SQL USERS AND ROLES

- In master database, add a login
  - CREATE LOGIN *newUser* WITH PASSWORD = ‘*strongPassword*';
- In application database
  - CREATE USER *newUser* FROM LOGIN *newUser* ;
  - ALTER ROLE db_datareader ADD MEMBER *newUser* ;
- https://docs.microsoft.com/en-us/azure/sql-database/sql-database-manage-logins

DRAWBACKS

- You must script any changes… there is no design view when system versioned
  - ALTER TABLE {tableName} ADD {fieldName} {type}
  - Not a bad idea to have to script you changes
    - Same updates to test and production databases
    - Scripts are much faster on SQL Azure than using the designer
- SQL on Prem and SQL Azure seem to act differently in places
- History table show up in the SSMS view designer which clutters things up
- Views with temporal query syntax will not render in the Diagram Pane

TURNING OFF SYSTEM VERSIONING

- ALTER TABLE *tableName*
  SET (SYSTEM_VERSIONING = OFF);

- Removes the system versioning and makes the history table a regular table
  - It doesn't delete the history table

HOW TO QUERY DATA

- Standard SELECT query on the main table – current data
- Standard SELECT query on history table – returns all the history records
- Temporal Queries – Historical Time Travel
  - New clause FOR SYSTEM_TIME
    - ALL
    - AS OF <datetime>
    - FROM <startdatetime> TO <enddatetime>
    - BETWEEN <startdatetime> AND <enddatetime>
    - CONTAINED IN (startdatetime, enddatetime)

TEMPORAL QUERY RESULTS

- FOR SYSTEM_TIME {…}
  - ALL – All rows in temporal table and history table
  - AS OF – Rows valid for a point in time
  - FROM TO - All rows in both tables – excludes “current” end date
  - BETWEEN AND - All rows in both tables – includes “current” end date
  - CONTAINED IN – Only rows in history table that are in the date range

TEMPORAL QUERY DEMO

QUERIES WITH MULTIPLE TABLES

- Temporal queries with more that one table must have a FOR SYSTEM TIME on each table
  - SELECT *
    FROM tableA FOR SYSTEM TIME AS OF ‘2020-2-20’
    LEFT JOIN tableB FOR SYSTEM TIME AS OF ‘2020-2-20’
    ON tableA.fk = tableB.pk
- Or create a view and apply it once
  - SELECT *
    FROM myView FOR SYSTEM TIME AS OF ‘2020-2-20’

QUESTIONS?