SQL Confessions of an Old-Time DataFlexer

Presenter: Bob Cergol

SYNERGY 2019

CRUISING TO NEW HORIZONS

Blinded by Love

Why I loved embedded DataFlex DBMS

Why I Loved Embedded DataFlex DBMS

- Simple
- Fast
- High Capacities
- Advanced Features
- Multiuser
- Multiplatform

- Easy to learn
- Easy to deploy
- Easy to work with
- FREE
- Helped me compete
- 12 years of comfort
  (1985..1996)

Blind Love Meets Reality

Life grew increasingly complex & costly

Life grew increasingly complex & costly

- **Support Costs**
  - Fragile network infrastructure
  - Index corruption
  - Transactional corruption
  - User-base growth made unmanageable
- **Sales & Marketing**
  - Data-what…?
  - Proprietary DBMS
  - Interoperability & data sharing with other apps
  - No one gets fired for buying IBM/Microsoft

Transitions & Tunnel Vision

Solving problems while avoiding change

Solving problems while avoiding change

- **Btrieve!** (1996..2008)
  - Felt like DataFlex
  - .DAT = .BTR
  - No talk of learning a new language, let alone embedding it!
  - But, some new headaches…
- **MSSQL** forced into in 2009
  - Customers' apps used it
  - Data interoperability
  - Could still pretend it was DF on the backend
  - Could still keep my code 100% DataFlex (but no embedded SQL … please)

The New Love of My Life

How I came to understand the power of SQL

How I came to understand the power of SQL

- **Replacing one-off DF pgms for dev./adm./mnt.**
  With simple SQL command – in a *blink* !
  - **Delete** from table_name where [condition]
  - **Update** table_name … set column_name = [value]
  - **Insert** into [database..table_name] (column_list)
    from select column_list from table_name
    where [condition]
  - **Select** [column list] from table_name ….

How I came to understand the power of SQL

- **My First SQL Project**
  - Consolidating multi-location corp. divisions
  - 1 central, shared database across WAN
  - DF vs. SQL commands
  - Scripting a repeatable migration, merge & update process for 24x7 operation
- **REPORTING**
  - *Dynamic AI* blew my mind & taught me basic SQL (2010)
  - SQL allowed types of reporting I never imagined practical…
  - …Because it was so ungodly fast!

How I came to understand the power of SQL

- **A new focus on using SQL within DataFlex** (2016)
  - Beyond merely a "black-box" back-end….
    - SQL filters
    - Virtual tables: data transformations (views)
    - Embedded SQL
  - **WOW! I'd really been missing the boat!**

How I came to understand the power of SQL

- **SQL can be a superior alternative to DF code for certain tasks**
  - **Data Maintenance**
    - Initialize new columns
    - Move/Copy values from column-to-column
    - Purging / deletion of data
    - Conditional search-and-replace
    - Auditing: find missing or bad data
    - Repair: fixing corrupt totals
    - Migration
  - **Reporting** (stop using DF + temp tables for multi-pass & sorting!)
  - **Populating Grids** (stop adding indices just for a grid!)
  - Encapsulating complex '.pkg' logic in a virtual table
  - Business Processes (depending how complex your DD logic is)

The SQL way of thinking

Top 10 most useful lessons I learned

Top 10 most useful SQL lessons I learned

1. In SQL scripting, it seems physical tables don't really exist
2. Every select statement yields 1 ***derived*** table
3. Visualize that 1 final table required, and work backwards from there, on top of layers of derived tables
4. Need columns? Use JOINs.

   Hint: careful about duplicate rows! Compare row-counts
5. Need more rows with same columns? Use UNIONs.

   Hint: add a column to each *Select* that identifies the source of those rows

Most useful SQL lessons I learned

6. Null values can bite you. (Joins can produce nulls; isNull() works wonders.)
7. Ambiguous column names not allowed. (back to derived tables/alias…)
8. SQL *window functions* solve tough challenges.

   OVER/Partition: Lead, Lag, Row_Number, Row, Sum, Avg, Count, etc.
9. SPs cannot be used with *Select,* – but TVFs can. (same difference…)
10. SQL is a different way of thinking about working with data

    Two different ways of thinking:

    The DF way: Linear: / working with data Row-by-Row

    The SQL way: Set logic: working with data table-by-table

Most useful SQL lessons I learned

- **Whatever you need to do examples exist**
  - My favorite sites
    - www.SqlServerCentral.com
    - www.DatabaseJournal.com
    - https://blog.SqlAuthority.com/

No One Would Believe Me!

Using SQL to Validate Data Processes

Using SQL to Validate Data Processes

- **The *Whale Project***
  - Nightly Legacy DF BP was taking 5+ hours to run, maybe crashed…
    - *Reading data from 5 tables (multi-million rows)*
    - *Writing data to 4 temp tables for export to 3rd party application*
    - *16 subroutines*
    - *15 nested processing loops*
    - *49 conditional blocks with 38 complex selection expressions*
    - *67 formula calculations*
    - *16 separate save routines*
  - All-SQL solution took <4 minutes: no one believed it, afraid to use it
  - I had to PROVE RESULTS – every column, every row in every table

Using SQL to Validate Data Processes

- **DataFlex DBMS testing**
  - Reset DB to reference
  - Run Old Process
  - Check .DAT date/time stamps
  - Reset DB to reference
  - Created limited test case
  - Run New Process
  - *Manually inspect outcome of record updates in affected tables*
- **SQL DBMS Testing**
  - List Tables Updated
  - Create baseline backup
  - Run old process and create 'old' table backup, and restore to baseline
  - Run new process and create 'new' table backup
  - *Compare every column/row in 'old' vs. 'new' and show variance*

EXAMPLES

The simple power of SQL

The simple power of SQL

- **Examples to demonstrate**
  1. Data Process Validation Utility (DF19.1 workspace is available)
  2. DF/SQL Filters Performance (real world example)
  3. Lag() Window Function (find gaps in sequential IDs)
  4. For XML: (concatenate multiple row/col values into single row)
  5. Positional vs. Named Parameters (safe vs. replace-inject)
  6. Stored Procedure vs. Table-Valued Function
  7. Virtual Table (SQL view to replace complex table structure & PKG logic)
  8. Reporting Performance examples (temp DF tables vs. SQL script)
  9. Querying Metadata (useful for discovering structure of DB w/no doc.)

My Final Confession

Is it polygamy if I love both?

Polygamist?...Guilty as charged!

- I love DataFlex for:
  - Handling complex record-oriented logic
  - Building complex user-interfaces
  - Encapsulating complex business rules of data entry applications
- I love SQL for:
  - Handling complex table-oriented logic
  - Building complex derived tables from an app's base tables
  - Delivering the results to an application at lightening speed