



Using SQL in CA 2E – Latest best practices





Introduction/Welcome Message

Modern CA:2E 8.7 allows us to transform our database from traditional DDS specifications, to more modern and accepted SQL. This session will show the process of moving from DDS to SQL, and how this can be done in a practical setting.

Speaker



Jason Olson

CM First Group

Agenda

- What Exactly is SQL?
- Benefits of Moving to SQL
 - Developer Benefits
 - SQL Performance Benefits
- SQL Objects vs DDS Objects
- SQL Source vs DDS Source
- SQL Specific Data Types
- How to Configure SQL Support



What Exactly is SQL?

What Exactly is SQL?

- SQL Stands for Structured Query Language.
- Developed at IBM in the Early 1970s.
- Consists of a DDL (Data Definition Language)
 - Examples are, CREATE TABLE, CREATE INDEX
- Consists of a DML (Data Manipulation Language)
 - Examples are, INSERT INTO, UPDATE, DELETE,
- To a Large Extent Consists of Simple English Statements.

Benefits of Moving to SQL

Benefits of Moving to SQL

- Whole World Uses SQL. New resources are more familiar with SQL then DDS.
- Long Field Names are Available to Outside World. YAY! ☺
- Data is Validated when Written to the Database.
- DDL Defined Indexes use a 64K page size vs 8K used by Logical Files.
- RPG Programs that Use SQL Access Don't Need a Re-compile.
- Benefits outside of 2E.
 - Null Capable Fields.
 - New Data Types (Blob, Clob, Lob)
 - Automatic Identity Columns.

Benefits of Moving to SQL

- No Enhancements to DDS Language.
- Constraints are Built into Database.
- Built in Encryption.
- Easier Modification of Table Layouts.
- RPG Programs that Use SQL Access Don't Need a Re-compile.
- Automatic Identity Columns.

SQL Objects vs DDS Objects

SQL Objects vs DDS Objects

- DDS Objects Create Physical Files & Logical Files
 - The Object types are *FILE PF and *FILE LF
 - Most common source is DDS and stored in QDDSSRC.
 - Source Member type is PF for Physical files, and LF for Logical Files.
 - Compiled like a program using option 14 from PDM, or using the CRTPF or CRTLF commands.
- SQL Objects create Physical Files & Logical Files as well.
 - The Object types are *FILE PF and *FILE LF
 - Most common source is SQL and stored in QSQLSRC.
 - Source Member type can be named anything.
 - In 2E the member types are YSQL.

SQL Objects vs DDS Objects

```
Library . . . . . : QGPL
Type of file . . . . . : Physical
File type . . . . . : FILETYPE *DATA
Auxiliary storage pool ID . . . . . : 00001
Data Base File Attributes
Externally described file . . . . . : Yes
SQL file type . . . . . : TABLE
```

F3=Exit F12=Cancel F19=Left F20=Right F24=More keys

SQL Objects vs DDS Objects

```
Library . . . . . : QGPL
Type of file . . . . . : Physical
File type . . . . . : FILETYPE *DATA
Auxiliary storage pool ID . . . . . : 00001
Data Base File Attributes
Externally described file . . . . . : Yes
SQL file type . . . . . : TABLE
```

F3=Exit F12=Cancel F19=Left F20=Right F24=More keys

SQL Objects vs DDS Objects

```
File type . . . . . : FILETYPE    *DATA
Auxiliary storage pool ID . . . . . :          00001
Data Base File Attributes
Externally described file . . . . . :          Yes
SQL file type . . . . . :          VIEW
```

F3=Exit F12=Cancel F19=Left F20=Right F24=More keys

```
Auxiliary storage pool ID . . . . . :          00001
Data Base File Attributes
Externally described file . . . . . :          Yes
SQL file type . . . . . :          INDEX
```

Mo

F3=Exit F12=Cancel F19=Left F20=Right F24=More keys

SQL Source vs DDS Source

SQL Source vs DDS Source

- SQL Source is stored in source file QSQLSRC.
 - Most common source file, and 2E uses this as well.
- Source member type can be named anything.
 - In 2E they are YSQL.
- SQL Objects are created using the RUNSQLSTM command.
 - Normal GEN process in 2E performs this under the covers.

SQL Source vs DDS Source

```
EXEC SQL
CREATE TABLE AAAQCPP (
    SPOOLFILE FOR COLUMN AQAQVN          CHAR(10)
    NOT NULL WITH DEFAULT
, SPOOLFILE_NUMBER FOR COLUMN AQA3NB          DECIMAL(10,0)
    NOT NULL WITH DEFAULT
, DATE_CREATED FOR COLUMN AQADDZ          DATE
    NOT NULL WITH DEFAULT
, TIME_CREATED FOR COLUMN AQADTZ          TIME
    NOT NULL WITH DEFAULT
, USER_DATA FOR COLUMN AQATTX          CHAR(25)
    NOT NULL WITH DEFAULT
```

SQL Source vs DDS Source

```
RCDFMT FAQCPB2  
END-EXEC
```

```
EXEC SQL  
  LABEL ON TABLE AAAQCPP IS  
    'Spool File Details      Physical file'  
END-EXEC
```

```
LABEL ON AAANCPP (  
  ID TEXT IS 'ID'  
  ,STATUS TEXT IS 'Status'  
  ,FIRST_NAME TEXT IS 'First Name'  
  ,LAST_NAME TEXT IS 'Last Name'
```

SQL Source vs DDS Source

○ Items to Note!

- SQL objects are created using the RUNSQLSTM Command.
- SQL View And Indexes produce separate IBM i Objects. (LFs) 2E does this as well.
- No problem combining SQL Created tables and DDS Created LFs. (Common)

```
Run SQL Statements (RUNSQLSTM)

Type choices, press Enter.

Source file . . . . . _____ Name
Library . . . . . *LIBL _____ Name, *LIBL, *CURLIB
Source member . . . . . _____ Name
Source stream file . . . . . _____

_____
Commitment control . . . . . *CHG *CHG, *UR, *CS, *ALL, *RS...
Naming . . . . . *SYS *SYS, *SQL
```

SQL Source vs DDS Source

- DML is used to manipulate the data in the database.
 - SELECT (to read data)
 - INSERT (to insert data)
 - UPDATE (to update data)
 - DELETE (to delete data)
- Big advantage is the use of the DB2 Database Engine for automatic index selection.
 - If an index exists that matches the search criteria then that index will be used automatically. In standard record level access from RPG this is not the case.
- SQL DML Statements can be incorporated directly into an RPG program.
 - There is no longer a need to use the traditional record level access provided by RPG.

SQL Source vs DDS Source

```
C+      DECLARE AAAOCPL4XCSR      CURSOR FOR
C+      SELECT * FROM AAAOCPL4
C+          WHERE (AOAIVN      = :AOAIVN
C+          AND      AOAJVN      = :AOAJVN
C+          )
C+      ORDER BY  AOAIVN      ASC,
C+          AOAJVN      ASC
```

SQL Source vs DDS Source

```
C/EXEC SQL  
C+   OPEN AAAOCPL4XCSR  
C/END-EXEC
```

```
C/EXEC SQL  
C+   FETCH AAAOCPL4XCSR  
C+       INTO :QAAA01  
C/END-EXEC
```

```
D QAAA01          E DS          EXTNAME(AAAOCPL4)
```


SQL Source vs DDS Source

```
C/EXEC SQL
C+      UPDATE AAAOCPL0
C+          SET AOAGST          = :AOAGST
C+          ,   AOAHST          = :AOAHST
C+          ,   AOAIIST         = :AOAIIST
C+          ,   AOAJST          = :AOAJST
C+          ,   AOAFST          = :AOAFST
C+          ,   AOAGVN          = :AOAGVN
C+          ,   AOABDZ          = :AOABDZ
C+          ,   AOABTZ          = :AOABTZ
C+          ,   AOAHVN          = :AOAHVN
C+          ,   AOACDZ          = :AOACDZ
C+          ,   AOACTZ          = :AOACTZ
C+      WHERE ( AOAIIVN          = :AOAIIVN      AND
C+          AOAJVN              = :AOAJVN      )
C/END-EXEC
```

SQL Specific Data Types

SQL Specific Data Types

- Items that can be performed outside of 2E.
 - BLOB data type to store binary data.
 - Examples are text files, audio files, spreadsheets
 - SQL Only Data Type! (No DDS Support)
 - BLOBs can be up to 2GB in size! YAY!
 - Created As Column In Table Definition.
 - Cannot Be Modified Outside of Program. (NOT in IFS)
 - Takes Advantage of IBM i Security.
 - Automatic Replication to Target HA System.
 - Others are LOB, and CLOB.
 - NULLs
 - SQL only. (No DDS Support)

How to Configure SQL Support

How to Configure SQL Support

- Multiple approaches can be taken to change a model to support SQL.
 - The model can be changed to use the 2E generated table name, but use longer field name support. This allows you to expose the longer field names to the outside world. No change to RPG.
 - The model can be changed to use both the longer table name, and also the longer field names. Again, easier for the outside world. No change to RPG.
 - The model can be changed so generated function to use SQL record level access regardless of the database type.
 - The model can be changed to use SQL for the database, and also SQL record access for all the functions. DDS would not be generated at all in this case.

How to Configure SQL Support

- To change the model at the model level, multiple model values need to be configured.

Name	Description	Values
YSQLVNM	SQL naming	*DDS
YDDLDBA	Database Access Method	*RLA / *SQL
YSQLEN	SQL naming length	10 to 25
YSQLOPT	Generate SQL OPTIMIZE clause	*NO
YSQLFMT	Generate SQL RCD_FMT clause	*NO
YSQLSTM	SQL statement generation type	*EXC
YSQLCOL	Generate SQL Col/Library Name	*YES
YLVCHK	Generate IDX with LVLCHK(*YES)	*NO
YDBFGEN	Method for database file generation	DDS / SQL / DDL
YSQLWHR	specifies whether to use OR or NOT logic when generating SQL WHERE clauses. The default is *OR.	
YSQLLCK		

Model Values in Detail



How to Configure SQL Support

- YSQLVNM - relates to the naming scheme used for fields and files.
 - *DDS – Uses the DDS name
 - *SQL – Use the names of the objects in the model.
 - *LNG – Use long names for fields and files
 - *LNF – Use long field names
 - *LNT – Use long table names

How to Configure SQL Support

- YDDLDBA - specifies the method of accessing the database
 - *RLA - External function generates with RLA access.
 - *SQL - External function generates with SQL access.

How to Configure SQL Support

- YDDLDBA - specifies the method of accessing the database
 - *RLA - External function generates with RLA access.
 - *SQL - External function generates with SQL access.

How to Configure SQL Support

- YSQLLEN – The length of field names up to 25 characters.
 - Any number between 10 and 25.
 - Outside of 2E this number can be much larger.

How to Configure SQL Support

- YSQLFMT - Specifies whether the RCDFMT keyword must be generated for SQL tables, views, and indexes.
 - *YES - the RCDFMT value is calculated using the same rules as are used when DDS files are generated.
 - *NO - the record format is the same as the table, index, or view name.

How to Configure SQL Support

- YDBFGEN – Specifies the language used in creating the logical/view over a physical file/table.
 - *DDS
 - *SQL
 - *DDL

How to Configure SQL Support

- Individual tables and access paths can also be changed without changing model values.

```
Generation mode . . . : M (M-MDLVAL, D-DDS, S-SQL, L-DDL)
Source member name . . : AAACCPP
Source member text . . : Test File Physical file
```

```
      Format      GEN  Format text
? Seq name      pfx  (Based on file)
  1 FACCPAE      AC   Test File
```


How to Configure SQL Support

- Individual tables and access paths can also be changed without changing model values.

```
Generation mode . . . : M (M-MDLVAL, D-DDS, S-SQL, L-DDL)
Source member name . . : AAACCPL0
Source member text . . : Test File Update index

Data access method . . : _ (M-MDLVAL, G-DBFGEN, T-TABLE)

      Format      GEN  Format text
? Seq name      pfx  (Based on file)
_   1 FACCPAF    AC   Test File
```

How to Configure SQL Support

- The same can also be changed for functions.

```
Generation mode . . . . . : A  ( M-MDLVAL, D-DDS, S-SQL, A-ACPVAL )
Generate help   . . . . . : M  ( M-MDLVAL, Y-Yes, N-No, O-Only )
```

Contact

Email: jason.olson@cmfirstgroup.com

Website: www.cmfirstgroup.com