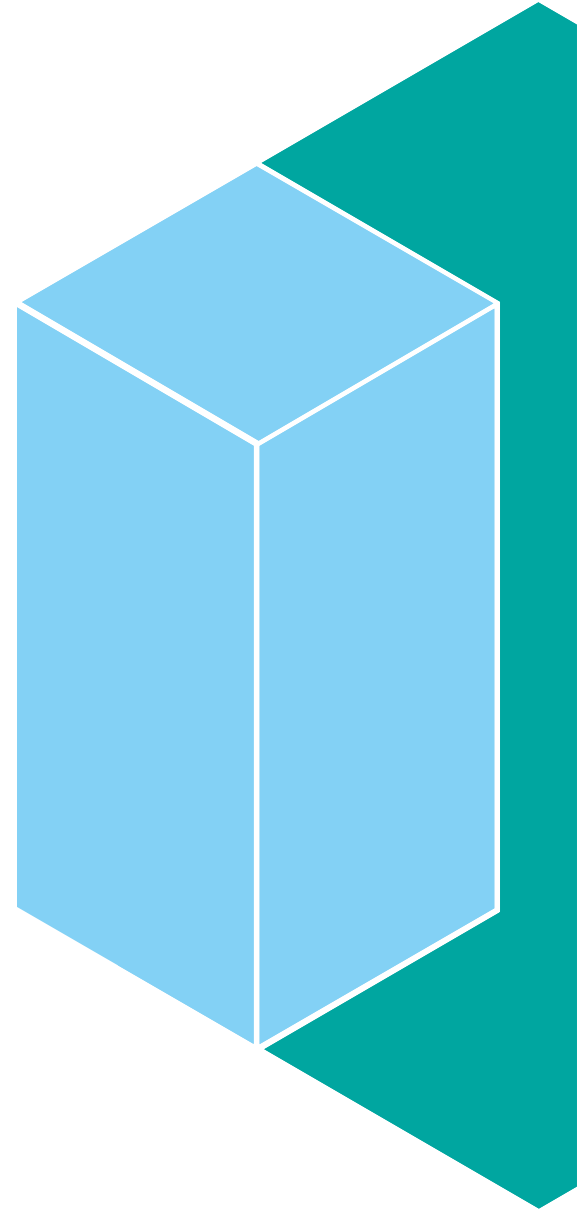


C02 A Data Definition Language for Multiple Data Types

Greg Vance

IMS Development

gvance@us.ibm.com



Sharpen your competitive edge

2016 IMS Technical Symposium

March 7 – 10, 2016

Wiesbaden, Germany

www.ims-symposium.com

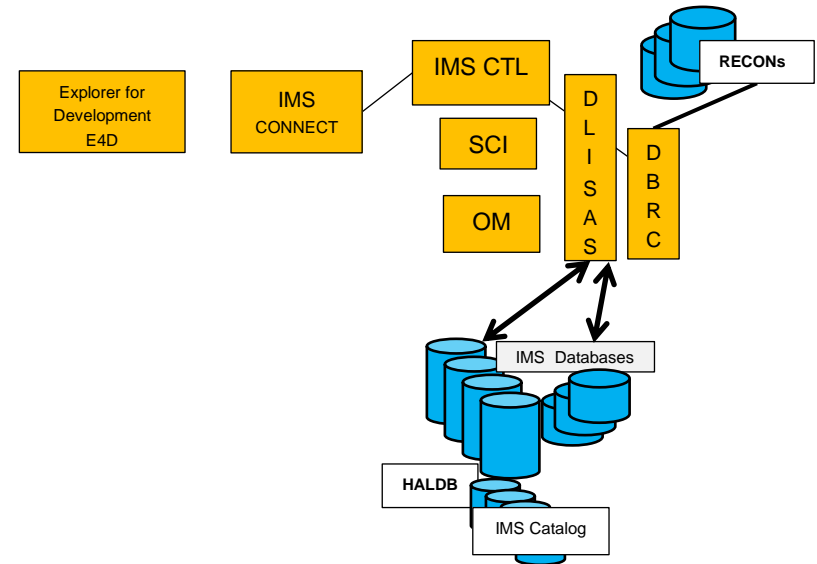
Agenda

- **Background**
- **Implementation and Usage**
- **Using IMS Explorer for Development with DDL**
- **Using the Batch SQL Utility for DDL**
- **Samples**

Dynamic Database Definition

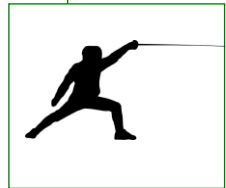
▪ Increasing the availability of IMS environments

- High Availability Large Database HALDB
- IMS Connect
- Dynamic Resource Definition DRD
- IMS Catalog V12
- SQL with COBOL V13
- **IMS Managed ACBs V14**
- **Data Definition Language V14**



Why this is important

- DDL is the industry standard
- DDL-authoring tools are prevalent in the market



Data Definition Language – DDL

- Generically, Data Definition Language refers to any formalized language that describes data, information structures of a data construct or access to those structures.
- The language uses a collection of command verbs to manipulate control schemas. Schemas can be manipulated: they may be added, changed or deleted during the life of the formalized data construct.
- DDL is often considered a subset of SQL.

▪ Data Definition Language – DDL

All database systems have a uniqueness to the data structures required in their specific environment.

- Standard DDL + DBMS specific Extensions
- IMS is no different in this respect.
- IMS 14 provides for using the standard DDL.
- IMS 14 also includes extensions specific to IMS structures to allow more detailed database definitions. These extensions closely match the current non DDL definitions used in IMS.

Dynamic Database Definition

- **Leverage the Industry standard Data Definition Language (DDL) to affect database and schema changes.**
 - Exploit DDL authoring tools such as the IMS Enterprise Suite Explorer for Development (E4D) to model database changes and create DDL.
 - DDL – authoring tools are prevalent in the market.
 - DDL – authoring tools use IMS Universal drivers or supporting tooling.
 - Align with industry practices and expectations.
- **Uses the IMS 14 Catalog as the trusted Repository**
 - IMS 14 provides for the IMS Catalog to be the central focal point for IMS database and program changes.
- **Dynamic implementation of IMS control blocks.**
 - Reduced time and complexity of creating IMS databases and program definitions.
 - An alternative to proprietary PSBGEN, DBDGEN, and ACBGEN processes.
 - Changes may be activated when committed
- **Provide an audit trail capturing changes made:**
 - Log record x'2A' for DDL Information.
 - Log record x'2B' for IMS Catalog changes.
 - SMF type 29 sub-type 3 for Catalog activity.

IMS and DDL value additions

▪ Simplifies the process of **adding a new database resource**

– Without DDL:

- Define your database characteristics (DBD).
- Compile/link database definitions (DBDGEN).
- Define your program specifications (PSB).
- Compile/link your program specifications (PSBGEN).
- Perform ACBGEN.
- Allocate database data sets.
- Define DBRC definitions.
- Define IMS database for dynamic allocation (MDA).
- Initialize and Load IMS databases.
- Define database and program online resource definitions:
 - CREATE DB / PGM command to create the database and program (DRD).*
 - Specify system definition macros and Online Change.*
- Perform online change to load IMS application-related definitions (MOLC).
- Establish database recovery point by taking an image copy (IC).
- Start IMS resources: - Databases, Programs, Transactions, Route Codes

IMS and DDL value additions

▪ Simplifying the process of adding a database resource.

– With DDL:

- Generate the DDL statements for the database and program views.
- Submit DDL statements

DDL changes will be held in the IMS Catalog in a pending state.

- Allocate database data sets.
- Define DBRC definitions.
- Define IMS database for dynamic allocation (MDA).
- Initialize and Load IMS databases.
- Define database and program online resource definitions:
CREATE DB / PGM command to create the database and program (DRD).
Specify system definition macros and Online Change.

- Activate the database and program definitions.

IMPORT DEFN SOURCE(CATALOG)

- Establish database recovery point by taking an image copy (IC).
- Start IMS resources: - Databases, Programs, Transactions, Route Codes

IMS and DDL value additions

- **Simplifies the process of adding new application metadata to the Catalog**

- Without DDL:

- The DBD source would have to be updated with the COBOL copybook or PL/I include information for each segment overlay
- DBDGEN / ACBGEN needs to be performed
- Online Change needs to be completed

- With DDL:

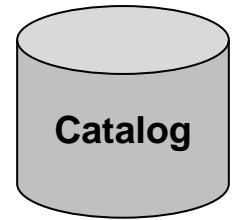
- DDL ALTER TABLE to add the information to the catalog and make it available with the IMPORT command

Agenda

- **Background**
- **Implementation and Usage**
- **Using IMS Explorer for Development with DDL**
- **Using the Batch SQL Utility for DDL**
- **Samples**

Dynamic Database Definition

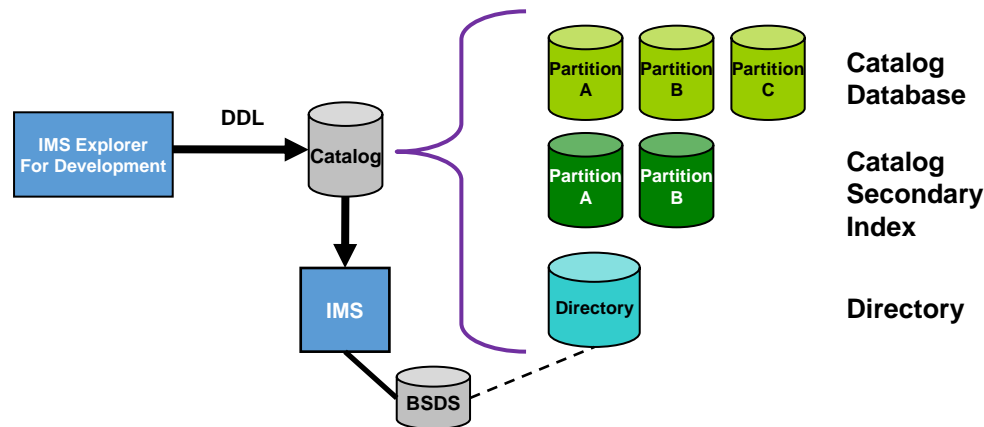
- IMS Catalog must be implemented.
 - IMS Management of ACBs must be enabled.
 - DDL is supported only through:
 - Java™ client using the IMS Universal drivers.
 - IMS Enterprise Suite Explorer for Development.
 - Additional tools that support the IMS Universal drivers.
 - Manual steps are required for database changes.
 - User makes DDL change to the catalog.
 - Perform manual steps such as data base load/reload, image copy.
 - IMPORT the changes from the catalog into IMS.
- IMPORT DEFN SOURCE(CATALOG)*
- The user must have security authority to use the IMS Catalog PSB DFSCP001.



Dynamic Database Definition

■ IMS 14 Catalog

- An IMS Catalog is made up of several components
 - A catalog HALDB database (many partitions, 4 Data Set Groups)
 - A Secondary Index
 - Directory dataset(s)

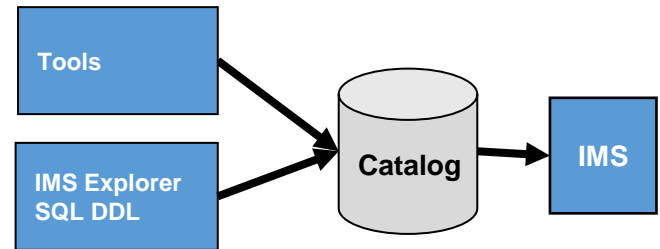


Implementing DDL – IMS Managed ACBs Setup

- **IMS Managed ACBs must be enabled before DDL processing is allowed**

- **New IMS Catalog user:**

- Set up the IMS catalog database
 - Generate the supplied DBDs & PSBs
 - Generate the ACBs
 - DFSDFnnn proclib member update or DFS3CDX0 exit
 - DBRC definition
- Populate the IMS catalog from running ACBLIB(s)
 - DFS3PU00 utility with MANAGEDACBS=SETUP
- Enable IMS to use the IMS catalog and directory by specifying IMS Managed ACBs is enabled in the DFSDFxxx PROCLIB member or the equivalent user exit (DFS3CDX0)
- Restart IMS region



- **Existing IMS Catalog user:**

- Augment the catalog with information to allow IMS to use this as its ACB information source.
 - DFS3PU00 utility with MANAGEDACBS=SETUP
- Enable IMS to use the IMS catalog and directory by specifying IMS Managed ACBs is enabled in the DFSDFxxx PROCLIB member or the equivalent user exit (DFS3CDX0) for DBCTL users
- Restart IMS region

Implementing DDL - – IMS Managed ACBs Setup (cont.)

DFSDFxxx syntax

```

                                | NO- |
>>-----+CATALOG-----+|=+YES+-----ALIAS=xxxx----->>
        | CATALOGXXX- |

                                +-ACCESS=UPDATE- |
        | -ACBMGMT=ACBLIB--+-----+-----+
        |                               | -ACCESS=READ--- |
>>-----+-----+-----+-----+----->>
        |                               | -ACCESS=READ--- |
        | -ACBMGMT=CATALOG-+-----+-----+
                                | -ACCESS=UPDATE- |
```

CATALOG default is **NO**

ACBMGMT default is **ACBLIB**

ACCESS default is **READ** when ACBMGMT is ACBLIB

ACCESS default is **UPDATE** when ACBMGMT is CATALOG

Resource Name Comparison

- IMS will support the standard DDL syntax for CREATE, ALTER and DROP of Databases and Tables
 - Consume the standard DDL generated without IMS affinity
- The equivalent IMS to DDL statements are shown in the Table here.

IMS	GEN statement	DDL
Database	DBD	DATABASE
Segment	SEGM	TABLE
Field	FIELD	COLUMN
Dataset	DATASET	TABLESPACE
Area	AREA	TABLESPACE
Program	PSBGEN	PROGRAMVIEW
PCB	PCB	SCHEMA
Senseg	SENSEG	SENSEGVIEW

IMS specific parameters in DDL

- **The DDL standard does not contain all of the options for IMS**
 - It is not IMS specific
 - Other DBMSs have similar specific requirements
- **Many options existing in the PSBGEN and DBDGEN macros are unique to IMS.**
 - e.g., DB Access Types: PHIDAM, HIDAM, PHDAM, etc.
- **IMS provides defaults based on access types**
 - Modifiable to user requirements
 - The recommended defaults can be seen in the syntax train tracks in the IMS SQL programming reference

DDL defaults and Enhanced IMS syntax

- Overriding the IMS system defaults
 - Enhanced DDL syntax
- All parameters that can be specified in the DBDGEN or PSBGEN macros are optional parameters in the IMS Enhanced DDL syntax
- The IMS Enhanced DDL syntax can be used with existing defaults
 - Defaults values may be overridden when specified
- DDL syntax has also been Enhanced to fully specify their PSB definitions via PROGRAMVIEW

Standard DDL syntax - continued

Multiple DDL statements are needed in order to describe an IMS database, such as: database organization type, access method, record mapping and record relationships.

- Database
- Table spaces (dataset/area)
- Tables (segments) and columns (field) mappings in a database record.
- Relationships with other databases Primary keys and foreign keys.
- Programview (PSB) describing a program's characteristics.
 - Tables and columns to which the program is sensitive

Creating resources

CREATE *resource_name*

- A resource can be created based on IMS defined defaults
- A resource can be created from scratch specifying all the needed attributes for the resource being defined.
- A created resource may be imported to active status in the IMS.
- A created resource is always a new resource.
 - A DRD or Sysgen is needed for IMS to be aware of the resource.

Altering resources

ALTER resource_name

- Resource must currently exist in the catalog.
- Resource defaults do not apply.
- The alter is applied only to the value(s) being altered.
- Values that are not specifically altered remain the same.
- An altered resource must be manually imported to Active status in the IMS in most cases.
- PRGRAMVIEWS cannot be altered

Dropping resources

DROP resource_name

- Resource must currently exist in the catalog.
- Dropping some resources may cause an ALTER to occur to an existing resource.
 - e.g. the Drop of a TABLESPACE within a DATABASE
- A Dropped resource is manually imported to remove its active status in the IMS.
 - System definitions are not removed

DDL Processing

▪ Upon receiving the DDL statements

- Updates in catalog are locked until a COMMIT DDL

This can impact other catalog access.

- Once committed, changes are held pending activation

Some changes may be activated immediately

- Changes are activated via the IMPORT command

IMPORT DEFN SOURCE(CATALOG)

▪ No need to use Online Change or recycle IMS system to activate an ACB

DDL Update Activation

- **During the IMPORT from the IMS catalog IMS will do the following:**
 - Automatically quiesce and stop all activity on resources impacted by any catalog changes.
 - Remove all old copies of the resources from memory.
 - This will cast out DMBs and PSBs from their respective pools.
 - Coordinate the change across the IMSplex for sharing IMS catalogs.
 - Apply the pending changes from the IMS catalog into the directory.
 - Make all of the previously quiesced resources available after the change to the catalog has been activated.
- **IMS Resources are now available**

DDL Activation – Impact

- Some DBRC commands are enhanced to either specify a catalog name or display the current default IMS catalog
- New messages may need to be monitored
- IMS Utility changes to access the catalog
- Exit DFS3CXD0 may need modification

DDL Creating an IMS database

- The following scenario focuses on the elements of the database creation that apply to most database types.
 - User uses Explorer to model the database graphically
 - User uses Explorer to create the DDL
 - Create DDL for DBD(s)
 - CREATE DATABASE, TABLESPACE, TABLE, FIELD(s)
 - Create DDL for PSB (user or dynamic)
 - CREATE PROGRAMVIEW, SCHEMA
 - Explorer notifies user the impact of the DDL
 - User submits the DDL to IMS (this could be repeated)

DDL Altering an IMS database

- **The following scenario focuses on the elements of the database alter that apply to most database types.**
 - User uses Explorer to model the database graphically
 - User uses Explorer to create the DDL
 - Alter DBD(s)
 - ALTER DATABASE
 - Change PROGRAMVIEW
 - DROP PROGRAMVIEW
 - CREATE PROGRAMVIEW (new)
 - Explorer notifies user the impact of the DDL
 - User submits the DDL to IMS
 - Any manually invoked changes e.g. Database unload/reload
 - IMPORT change to activate.

DDL Dropping an IMS database

- **The following scenario focuses on the elements of the drop database that apply to all database types.**
 - User uses Explorer to determine which database to remove
 - User uses Explorer to build the DDL
 - Drop DBD
 - DROP DATABASE
 - Explorer notifies user the impact of the DDL
 - User submits the DDL to IMS
 - IMPORT to activate to IMS

Some DDL usage scenarios

- **Tools can generate DDL based on Database metadata retrieved through standard discovery mechanisms like JDBC Database Metadata.**
 - e.g., IMS Explorer for Development and Optim Data Studio can generate DDL for an IMS database using JDBC discovery.
- **Tools can generate DDL for storing its own metadata repository**
 - e.g., Cognos generates DDL for its own content store that holds information on created reports.
- **Business Analytics can take existing data and enhance that data through analytics. The enhanced data can then be written back into the database for later use.**

Dynamic Database Definition - Considerations

▪ Utilities

- UDR and ULU type regions require DBDLIB and PSBLIB datasets
 - IMS Utilities and tools running as these region types

▪ Scalability

- Storage for DDL use is limited
 - Limit resources changes to 40 per commit points (recommended)

▪ Coexistence

- IMS must be Version 14
- Shared ACB systems must all be Version 14
- ACBMGMT=CATALOG

Dynamic Database Definition - Benefits

- **Dynamic implementation of IMS control blocks.**
 - An alternative to PSBGEN, DBDGEN, and ACBGEN processes.
- **Single trusted repository for metadata**
 - IMS will load from the catalog where changes are made through DDL
- **Use the industry standard DDL** to define and modify database and schema creation and changes.
- **Exploit DDL authoring tools** such as the IMS Enterprise Suite Explorer for Development (E4D) to model database changes and create DDL statements.
- **Provide an audit trail** capturing changes made.
- **Align with industry practices and expectations.**

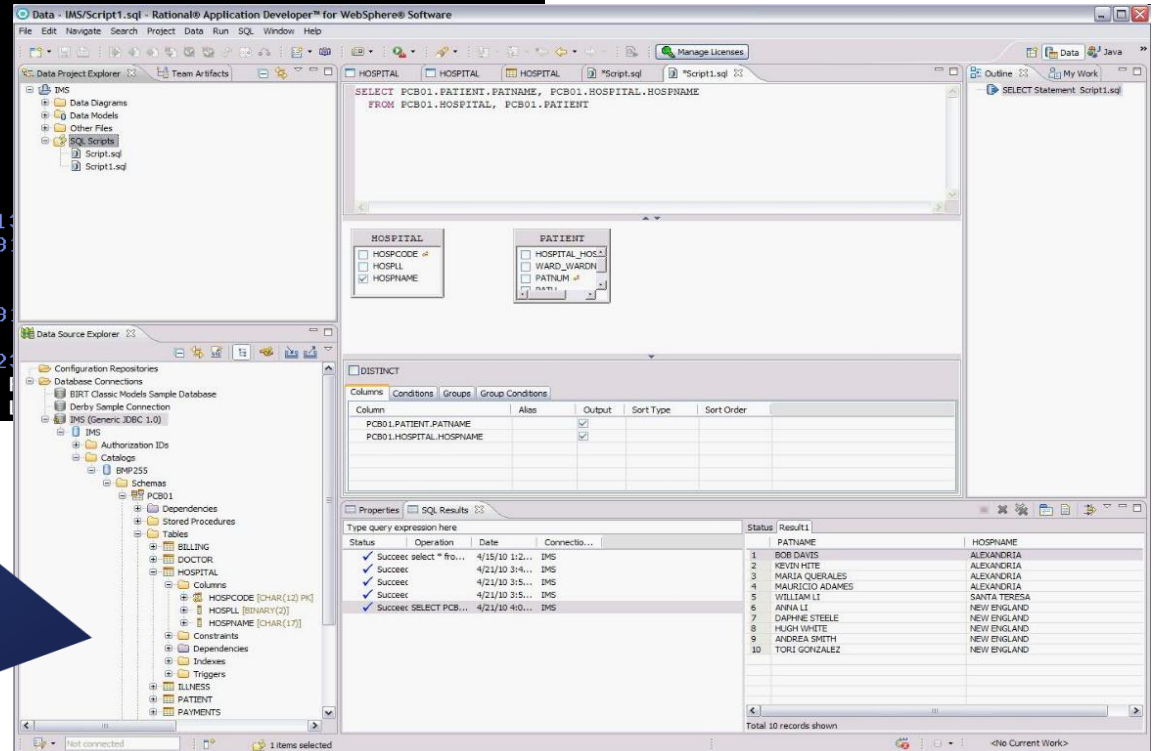
Agenda

- **Background**
- **Implementation and Processing**
- **Using IMS Explorer for Development with DDL**
- **Using the Batch SQL Utility for DDL**
- **Samples**

IMS Explorer for Development

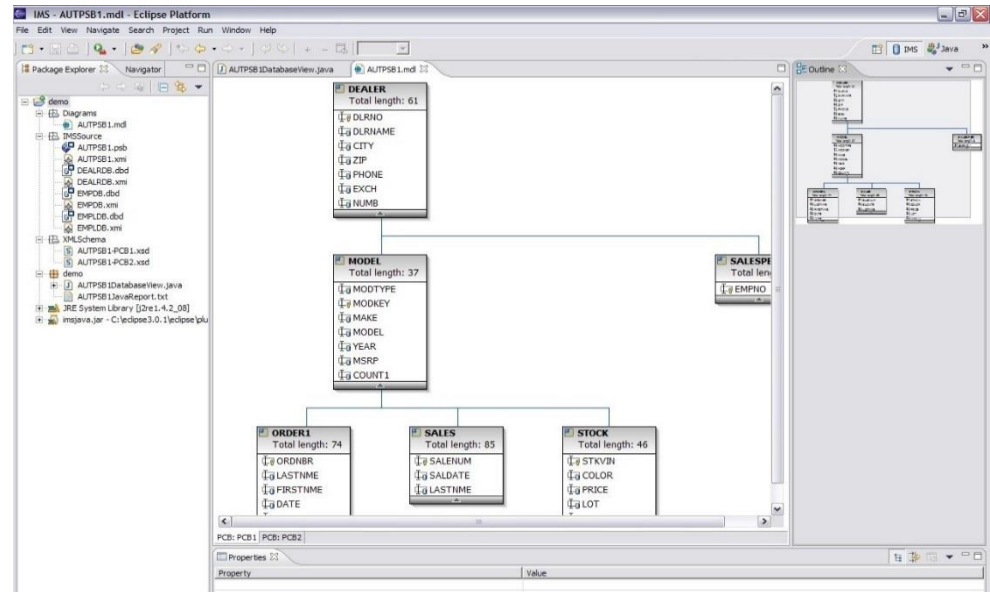
The IMS Explorer for Development is a tool to help with database visualization and querying.

```
$DDLTO NEWJCL F1 V 80 Trunc=80 Size=96 Line=25 Col=1 Alt=0
====>
00022 U *****
00023 WTO Start of the DDLTO stream
00024 U status card has all 1's so all tracing is ON.
00025 U status card has 00002 so we use the second PCB in the PSB
00026 S 1 1 1 1 1 00002
00027 WTO Now doing GN through the database
00028 L GN
00029 E DATA KAA11**K1*
00030 E 01 K1 0005KAA11
00031 L GN
00032 E DATA KBBB11**K2
00033 E 02 K2 0011KAA11KBBB11
00034 L GN
00035 E DATA KAA31KEE31K313111312131
00036 E 03 K3K5 0021KAA11KBBB11KAA3
00037 L GN
00038 E DATA KAA31**K1*
00039 E 04 K1X 0026KAA11KBBB11KAA3
00040 L GN
00041 E DATA KAA31KEE32K313211322132
PF 1 FIG 2 SCREEN 2 3 QUIT 4
PF 7 BACKWARD 8 FORWARD 9 XFILE 10
```



IMS Explorer for Development – DDL creation

- The IMS Explorer for Development has Enhanced DDL editing and generation features
 - A full text DDL editor
 - users can manually write their own DDL scripts
 - A graphical interface for Creating, Altering & Dropping DDL resources
- The generated DDL uses the enhanced IMS DDL syntax



IMS Explorer for Development – DDL creation

The screenshot displays the IMS Explorer for Development interface. The top toolbar includes the 'IMS Explorer Task Launcher' and tabs for 'DFSCD000.dbd', 'DI21PART.dbd', and 'Script1.sql'. The main window shows a database diagram for 'DI21PART' with the access method '(HISAM,VSAM)'. The diagram is a hierarchical structure:

- PARTROOT** (Length: 50 bytes) is the root node, containing a highlighted **PART_NO_EDIT** field.
- STANINFO** (Length: 85 bytes) is a child of PARTROOT, containing a **STANKEY** field.
- STOKSTAT** (Length: 160 bytes) is another child of PARTROOT, containing a **STOCKEY** field.
- CYCCOUNT** (Length: 25 bytes) is a child of STOKSTAT, containing a **FILL_0** field.
- BACKORDR** (Length: 75 bytes) is another child of STOKSTAT, containing a **BACKKEY** field.

A context menu is open over the 'IMPOT99' database connection in the left pane. The menu options are:

- Add to Overview Diagram
- Generate DDL...
- Analyze Impact...
- Analyze Model
- Compare With
- New SQL Script
- Manage Privileges
- Visualize Topology Diagram
- Copy
- Paste... (Ctrl+V)
- Refresh (F5)
- Properties

The bottom status bar shows the current database as '<Database> IMPOT99'.

IMS Explorer for Development – DDL creation

The screenshot displays the IMS Explorer for Development interface. The main window shows a database structure diagram for 'DI21PART'. The diagram is hierarchical, starting with 'PARTROOT' (Length: 50 bytes) at the top, which contains 'PART_NO_EDIT'. Below 'PARTROOT' are two branches: 'STANINFO' (Length: 85 bytes) and 'STOKSTAT' (Length: 160 bytes). 'STANINFO' contains 'STANKEY'. 'STOKSTAT' contains 'STOCKEY', which further branches into 'CYCCOUNT' (Length: 25 bytes) and 'BACKORDR' (Length: 75 bytes). 'CYCCOUNT' contains 'FILL_0', and 'BACKORDR' contains 'BACKKEY'.

Below the diagram, the 'DBD Source' tab is active, showing properties for 'IMPOT99'. The properties are:

Property	Value
Name	IMPOT99
Database type	IMS
Database version	V13

Overlaid on the right is the 'Generate DDL' dialog box. The 'Save and Run DDL' section is active, showing a path to save the script: 'Folder: .sqlxeditor_project' and 'File name: Script3.sql'. The 'Preview DDL' section shows the generated SQL code:

```
--<ScriptOptions statementTerminator=";" />

CREATE TABLE PARTROOT (
  PART_NO_EDIT CHAR(17) NOT NULL,
  REJECT_CODE CHAR(1) NOT NULL,
  PART_NO_EDIT_NEW CHAR(17) NOT NULL,
  FILL_0 CHAR(4) NOT NULL
);

CREATE TABLE BACKORDR (
  PARTROOT_PART_NO_EDIT CHAR(17) NOT NULL,
  STOKSTAT_STOCKEY CHAR(16) NOT NULL,
  BACKKEY CHAR(10) NOT NULL,
  WORK_ORDER CHAR(8) NOT NULL,
  WO_QTY DECIMAL(8, 1) NOT NULL,
  FILLER1 CHAR(2) NOT NULL,
  FILLER2 CHAR(53) NOT NULL
);
```

The 'Statement terminator' is set to ';'. The 'Run DDL on server' checkbox is unchecked, and the 'Open DDL file for editing' checkbox is checked. The 'Restore Defaults' button is visible at the bottom.

IMS Explorer for Development – DDL creation

The screenshot displays the IMS Explorer Task Launcher interface. The main window shows a SQL script for creating a database and tables. A green speech bubble points to a red 'X' icon next to the 'FOREIGN KEY REFERENCES' line, indicating a syntax validation error. The right-hand pane shows an 'Outline' view with a list of database objects.

Connection: IMS DB []

```
CREATE DATABASE DI21PART ACCESS HISAM;

CREATE TABLESPACE DI21PART IN DI21PART
  RECORD(678,678)
  OVERFLOW(DI21PAR0);

CREATE TABLE PARTROOT (
  PARTKEY CHAR(17) PRIMARY KEY,
  PARTNAME CHAR(15),
  PART CHAR(33),
  PART_DESCRIPTION CHAR(20)
) IN DATABASE DI21PART;

CREATE TABLE STANINFO (
  STANKEY CHAR(2) PRIMARY KEY,
  LONGFIELD CHAR(82),
  PROCCODE CHAR(2),
  INVCODE CHAR(1),
  PLANNUM CHAR(2),
  MAKE_DEPARTMENT CHAR(2),
  MAKE_COST CHAR(2),
  COMMCODE CHAR(4),
  FOREIGN KEY REFERENCES PARTROOT
) QIN DATABASE DI21PART;

CREATE TABLE STOKSTAT (
  STOCKEY CHAR(16) PRIMARY KEY,
  AREA CHAR(3),
```

Outline

- + CREATE DATABASE
- + CREATE TABLESPACE
- + CREATE TABLE
- + CREATE TABLE
- + CREATE TABLE
- + CREATE TABLE
- + CREATE TABLE

Syntax
Validation

Generated DDL for Alter

The screenshot displays the IMS Explorer Task Launcher interface. The main window shows the database structure for 'DI21PART'. The 'PARTROOT' table (Length: 50 bytes) is highlighted, with its fields 'PARTKEY', 'PARTNAME', 'PART', and 'PART_DESCRIPTION' listed. A green callout bubble labeled 'Change length' points to the 'PART_DESCRIPTION' field. The 'STANINFO' table (Length: 85 bytes) is also visible, with its fields 'STANKEY', 'LONGFIELD', 'PROCCODE', 'INVCODE', 'PLANNUM', 'MAKE_DEPARTMENT', 'MAKE_COST', 'COMMCODE', and 'NEWDATA' listed. A green callout bubble labeled 'New field' points to the 'NEWDATA' field. The 'STOKSTAT' table (Length: 160 bytes) and 'CYCCOUNT' table (Length: 25 bytes) are also shown. A 'Submit DBD Changes to IMS' dialog box is open in the foreground. It contains the following text: 'Select the connection profile for your target IMS. A series of DDL statements, embodying the changes made to the DBD, will be generated and displayed for your review. Click "Proceed" to submit the changes to IMS.' The 'Profile' dropdown is set to 'IMS DB'. The dialog box displays the following DDL statements:

```
ALTER TABLE PARTROOT IN DATABASE DI21PART  
ALTER COLUMN PART_DESCRIPTION  
CHAR(23) START 27 TYPE C  
INTERNAL TYPECONVERTER CHAR  
CCSID 'Cp1047';  
  
ALTER TABLE STANINFO IN DATABASE DI21PART  
ADD COLUMN NEWDATA  
CHAR(15) START 60 INTERNALNAME NEWDATA;
```

 The dialog box has 'Proceed' and 'Cancel' buttons at the bottom.

Change length

New field

Submit DBD Changes to IMS

Select the connection profile for your target IMS. A series of DDL statements, embodying the changes made to the DBD, will be generated and displayed for your review. Click "Proceed" to submit the changes to IMS.

Profile: IMS DB

```
ALTER TABLE PARTROOT IN DATABASE DI21PART  
ALTER COLUMN PART_DESCRIPTION  
CHAR(23) START 27 TYPE C  
INTERNAL TYPECONVERTER CHAR  
CCSID 'Cp1047';  
  
ALTER TABLE STANINFO IN DATABASE DI21PART  
ADD COLUMN NEWDATA  
CHAR(15) START 60 INTERNALNAME NEWDATA;
```

Proceed Cancel

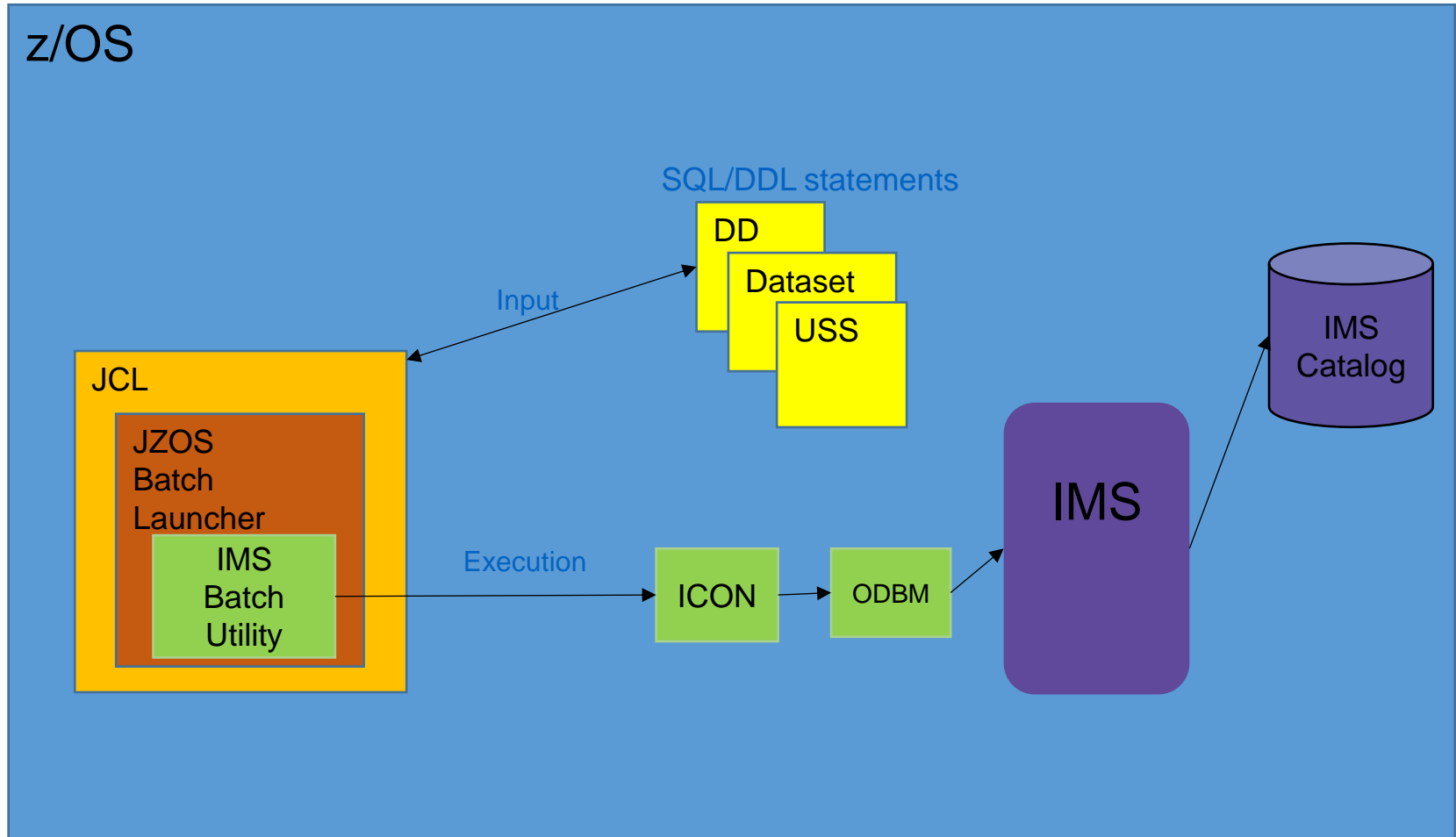
Agenda

- **Background**
- **Implementation and Processing**
- **Using Explorer for Development with DDL**
- **Using the Batch SQL Utility for DDL**
- **Samples**

Batch SQL Utility

- The Batch SQL Utility is made as a way to invoke DDL statements via a JCL on the z platform
- **Requirements:**
 - The utility is bundled in the IMS JDBC driver (imsudb.jar)
 - APAR PI30848
 - Uses IMS JDBC Type-4 connections to invoke SQL statements
 - IMS Connect, ODBM, SCI
 - IBM Java for z/OS (JZOS) Batch Launcher

Batch DDL Utility architecture



Batch DDL Utility Sample JCL

```
//IMSSAMPL JOB (999,XXX), 'JAVA BPXBATCH', CLASS=A, MSGLEVEL=(1,1),
//  MSGCLASS=E, REGION=0M, NOTIFY=&SYSUID
// SET P1='com.ibm.ims.jdbc.batch.BatchUtil'
//JAVAJVM EXEC PGM=JVMLDMxx, REGION=0M,
//  PARM='/ &P1'
//STEPLIB DD DISP=SHR,
//  DSN=USER.CUSTOM.JZOS.LOADLIB
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//STDOUT DD SYSOUT=*
//STDERR DD SYSOUT=*
//CEEDUMP DD SYSOUT=*
//ABNLIGNR DD DUMMY
//MAINARGS DD *
//IMSSQL DD DISP=SHR,
//  DSN=MYPDS (MYSCRIPT)
//STDENV DD *
export JAVA_HOME=myJavaHomePath
export PATH=/bin:${JAVA_HOME}/bin
LIBPATH=/lib:/usr/lib:${JAVA_HOME}/bin
export LIBPATH="$LIBPATH":
APP_HOME=$JAVA_HOME
CLASSPATH=$APP_HOME:${JAVA_HOME}/lib:${JAVA_HOME}/lib/ext
CLASSPATH="$CLASSPATH":myLibPath/imsudb.jar
```

Batch SQL Utility syntax

- The input statements supported by the Batch DDL Utility include all supported SQL and DDL statements by the IMS JDBC driver
- The following additional statements are supported:
 - CONNECT [JDBC URL];
 - This will create a JDBC connection to the IMS system using the specified JDBC URL
 - COMMIT;
 - This will commit work on the open connection.
 - ROLLBACK;
 - This will rollback work on the open connection.
 - DISCONNECT;
 - This will disconnect the current connection.
- Statements must be delimited by a semi-colon.

Batch DDL Utility Sample Input

```
CONNECT jdbc:ims://myConnectServer:5555/DFSCP001;

CREATE DATABASE ACCUNTDA ACCESS HDAM OSAM
      RMNAME(DFSHDC40 RMANCH 5 RMRBN 280);

CREATE TABLESPACE ACCUNTDA IN ACCUNTDA
      SIZE PRIMARY 4096;

CREATE TABLE ACCT0001 (
      ACCT0KEY CHAR(10)
            START 3 TYPE C
            PRIMARY KEY
) IN ACCUNTDA.ACCUNTDA
      MAXBYTES 34 ;

COMMIT;
DISCONNECT;
```

Batch DDL Utility error codes

The following error codes will be thrown by the JZOS Batch Launcher

Return Code	Description	Action
11	There was a connection error to IMS Connect	Verify that the connection parameters are correct
12	There was an error with the SQL statement execution. All work will be rollback to the prior commit point.	Verify that the SQL statement is valid
13	There was an issue with the commit	Check the JDBC error messages
14	There was an issue with the rollback	Check the JDBC error messages
15	There was an issue cleaning up the connection	Check the JDBC error messages
16	An invalid command was specified	Verify that valid commands were provided in the input file

Thank You!

Agenda

- **Background**
- **Implementation and Processing**
- **Using Explorer for Development with DDL**
- **Using the Batch SQL Utility for DDL**
- **Samples**

CREATE DATABASE

- The CREATE DATABASE statement defines a database. This creates a definition of the database only. Segments and Datasets are created separately.
 - Options will vary depending on database type.
- The creation of database is always a new resource.
- A created database is manually imported to active status in the IMS.
- Other definitions usually entered on the DBDGEN are created with separate CREATE resource_name statements.
 - Dataset
 - Area
 - Segment
 - Field

```
>>-CREATE--DATABASE--database_name---| Organization Specific Options |--->
```

IMS DDL - CREATE DATABASE (HDAM)

```
DBD      NAME=COGDBD,                                C
          ENCODING=Cp1047,                             C
          ACCESS= (HDAM,OSAM) ,                         C
          RMNAME= (DFSHDC40,3,3,25) ,                   C
          PASSWD=NO,                                    C
          VERSION='Latest version of COGDBD'
```

```
CREATE DATABASE COGDBD
  ACCESS HDAM OSAM
  RMNAME (DFSHDC40 RMANCH 3 RMRBN 3 RMBYTES 25)
  VERSION 'Latest version of COGDBD'
  PASSWDNO
  CCSID 'Cp1047';
```


IMS DDL syntax – ALTER DATABASE

- The ALTER DATABASE statement changes attributes of the database.
- The ALTER DATABASE keywords are the same as the keywords of CREATE DATABASE.
- There are no defaults for an ALTER DATABASE.
 - Any keywords entered are will override the existing values.
 - Keyword values not entered will remain the same value.
- An Altered database must be manually Imported in order Active status in the IMS.

```
>>-ALTER--DATABASE-database_name--| Options |---->
```

IMS DDL syntax – DROP DATABASE

DROP DATABASE database_name

- Identifies the database to drop. The name must identify a database that exists to IMS. When a database is dropped, all of its tables and indexes are also dropped.
- A dropped database is Imported to remove it from IMS.
- Resources may need to be removed via DRD or Sysgen source

```
>>-DROP--DATABASE-----database_name-----+-----+-----><  
                                     '-CASCADE-'
```

DDL syntax: CREATE TABLESPACE

- CREATE TABLESPACE
- The CREATE TABLESPACE statement defines a group dataset within the database or an area for a DEDB. This is equivalent to the DATASET or AREA statements used in the IMS DBDGEN source.
- The ddnames used on the CREATE TABLESPACE statement must be unique within an IMS system. Non-unique ddnames in two or more DBDs may result in destruction of the database.

Options vary depending on dataset type.

- Note:
 - Tablespace created with a Database may be automatically Imported with the Database to Active status in the IMS.
 - A Tablespace added to an existing Database must be manually Imported.

```
>>-CREATE--TABLESPACE--ddname---IN--database_name---| Options |---->
```

IMS DDL syntax – CREATE TABLESPACE Example

```
DBD    NAME=COGDBD,                                C
        ENCODING=Cp1047,                            C
        ACCESS=(HDAM,OSAM),                          C
        RMNAME=(DFSHDC40,3,3,25),                    C
        PASSWD=NO
DATASET DD1=COGDATA,                                C
        DEVICE=3390,                                  C
        SIZE=(8192),                                  C
        REMARKS='Dataset Group 1'
SEGM   NAME=ROOT,                                    C
        PARENT=0,                                      C
        BYTES=(20),                                    C
        RULES=(LLL,HERE)
FIELD  NAME=(ROOTKEY,SEQ,U),                          C
        BYTES=12,                                      C
        START=1,                                       C
        TYPE=C,                                       C
        DATATYPE=CHAR
FIELD  NAME=TABTYPE,
        BYTES=8,
        START=13,
        TYPE=C,
        DATATYPE=CHAR
```

```
CREATE DATABASE cogdbd
  ACCESS HDAM OSAM
  RMNAME(DFSHDC40 3 3 25);

CREATE TABLESPACE cogdata
  IN cogdbd
  SIZE PRIMARY 8192;
COMMENT ON TABLESPACE cogdata IN cogdbd IS 'Dataset Group 1';
```

DDL syntax – ALTER TABLESPACE

- The ALTER TABLESPACE statement changes attributes of the data set group within the database or an area for a DEDB. This is equivalent to the DATASET or AREA statements as defined in IMS DBDGEN source.
- The ALTER TABLESPACE keywords are the same as the keywords of CREATE TABLESPACE.
- There are no defaults for an ALTER TABLESPACE.
 - Any keywords entered are overrides to the existing values.
 - Keyword values not entered will remain the same value.
- An Altered Tablespace must be manually Imported.

```
>>-ALTER--TABLESPACE--ddname---IN--database_name---| Options |---->
```

CREATE TABLE

- The CREATE TABLE statement defines a segment. This creates a definition of the database only.
 - Options will vary depending on database type.
- The creation of table is always a new resource.
- A created table is manually Imported to Active status in the IMS.
- Other definitions usually entered on the DBDGEN are created with separate CREATE resource_name statements.
 - Dataset
 - Area
 - Segment
 - Field

IMS DDL syntax – CREATE TABLE Example

```
SEGM      NAME=TDEC,                                C
          PARENT=ROOT,                              C
          BYTES=(10,6),                              C
          REMARKS='This describes table TDEC.',      C
          RULES=(LLL,HERE)
FIELD     NAME=RNUM,                                C
          BYTES=4,                                    C
          START=3,                                    C
          DATATYPE=INT
FIELD     NAME=LL,                                  C
          BYTES=2,                                    C
          START=1,
          DATATYPE=SHORT
FIELD     NAME=CDEC,                                C
          EXTERNALNAME=CDECIN,                       C
          BYTES=4,                                    C
          START=7,
          DATATYPE=DECIMAL(7,2)
```

```
CREATE TABLE table_decimal (
  r_number INT INTERNALNAME RNUM,
  ll SHORT INTERNALNAME LL,
  c_decimal DECIMAL(7,2) INTERNALNAME CDEC,
  FOREIGN KEY REFERENCES table_root
) IN COGDBD.COGDATA
INTERNALNAME TDEC
MAXBYTES 10
MINBYTES 6
INSERT LOGICAL
DELETE LOGICAL
REPLACE LOGICAL
AMBIGUOUS INSERT HERE;
```

```
COMMENT ON TABLE table_decimal IN COGDBD IS 'This describes table TDEC.';
```

IMS DDL syntax – ALTER TABLE

- The ALTER TABLE statement changes attributes of the table within the database or an area for a DEDB. This is equivalent of a SEGMENT as defined in IMS DBDGEN source.
- The ALTER TABLE keywords are the same as the keywords of CREATE TABLE.
 - There are no defaults for an ALTER TABLE.
 - Any keywords entered are will override the existing values.
 - Keyword values not entered will remain the same value
- An Altered Table is manually Imported.

IMS DDL – Alter Table

Original DATABASE DBD (generated version 0):

```
DBD  NAME=COGDBD, C
      ENCODING=Cp1047, C
      ACCESS=(HDAM,OSAM), C
      RMNAME=(DFSHDC40,3,3,25), C
      PASSWD=NO
DATASET DD1=COGDATA, C
        DEVICE=3390, C
        SIZE=(8192)
SEGM  NAME=ROOT, C
      PARENT=0, C
      BYTES=(28), C
      RULES=(LLL,HERE)
FIELD NAME=(ROOTKEY,SEQ,U), C
      BYTES=12, C
      START=1, C
      TYPE=C, C
      DATATYPE=CHAR
FIELD NAME=TABTYPE, C
      BYTES=8, C
      START=13, C
      TYPE=C, C
      DATATYPE=CHAR
```

```
ALTER DATABASE COGDBD;
ALTER TABLE root
  IN COGDBD
  MAXBYTES 28
  ADD COLUMN New_Field_01 INTERNALNAME newfld01 DOUBLE START 21 ;
```

IMS DDL syntax – DROP TABLE

- All the tables (segments) in an hierarchy can be dropped by a single DROP TABLE statement.
- The DROP TABLE statement will drop child segments.
- A Dropped Table is Imported to activate the removal of the segments from the active IMS database.
- IMS resources may be removed from DRD or SYSGEN source.

```
>>-DROP--TABLE--table_name-IN-database-----><
```

CREATE PROGRAMVIEW Syntax

- CREATE PROGRAMVIEW is the equivalent of a PSB statement.
- SCHEMA is equivalent to a PCB.
- SENSEGVIEW is the equivalent of a SENSEG.
- DDL syntax has been Enhanced for users to write their own PSBs instead of relying on system defaulted ones.

ProgramView Generation

- Nested in the CREATE PROGRAMVIEW must be **one or more** CREATE SCHEMA statements. SCHEMA statements describe the PCBs.
- Nested in each CREATE SCHEMA for a DBs must be one or more CREATE SENSEGVIEW statements to describe the SENSEGs.
- Nested in each SENSEGVIEW may be sensitive fields.
e.g. CREATE SENSEGVIEW segment (flda WITH START (1),
WITH START (m)) ...
- **A Program view is Imported to be active in an IMS.**

CREATE PROGRAMVIEW - (PSB) Example

```
DFSIVP2  PSBSOR:
        PCB      TYPE=TP,MODIFY=YES
        PCB      TYPE=DB,DBDNAME=IVPDB2,PROCOPT=A,KEYLEN=10
            SENSEG NAME=A1111111,PARENT=0,PROCOPT=A
            PSBGEN LANG=ASSEM,PSBNAME=DFSIVP2
        END
```

```
CREATE PROGRAMVIEW DFSIVP2 (
    CREATE SCHEMA TP pcb01 MODIFYYES,
    CREATE SCHEMA pcb02 USING IVPDB2 (
        CREATE SENSEGVIEW A1111111 WITH
        PROCOPT 'A',
        ) PROCOPT 'A',
    ) LANGASSEM;
```

IMS DDL syntax – ALTER PROGRAMVIEW

- The ALTER statement does not apply to a PROGRAMVIEW.

- Altering a PROGRAMVIEW is performed by:

DROP PROGRAMVIEW

Must identify an existing defined program.

CREATE PROGRAMVIEW

With the new information for the Application Program.

Drop PROGRAMVIEW

- The DROP statement removes a resource from IMS. Any resources that are directly or indirectly dependent on that resource are deleted. Whenever a resource is deleted, its description is deleted from the catalog of the current IMS.
- The CASCADE keyword will drop any associated transactions and routing codes for that PSB.

```
>>-DROP--PROGRAMVIEW--psb_name---+-----+-----><  
                                '-CASCADE-'
```

- Resources may need to be removed via DRD or Sysgen source

COMMENT ON Syntax

- This statement provides optional user comments.
- Equivalent to the REMARKs keyword on the DBD / PSB source.
 - Optional user comments. A 1- to 256-character string enclosed in single quotation marks. The value specified cannot contain the following characters:
 - Less than (<) and Greater than (>) symbols.
 - Ampersands (&).
 - Double quotation marks.
 - Single quotation marks, except when they are used to enclose the full comment string. The following examples show correct and incorrect usages of single quotation marks:
 - CORRECT
... IS 'These remarks apply to the XYZ application'
 - INCORRECT
...IS 'These remarks apply to the 'XYZ' application'