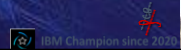




Data Centric – Moving Business Logic into the Database

Birgitta Hauser –
Modernization – Education – Consulting on IBM i
Diplom-Betriebswirt (BA)
Database and Software Architect
Hauser@ModEdCon.com
<https://modedcon.com/>



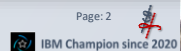
Landsberg am Lech – Stadtmauer / Townwall



03.11.2022

QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page: 2



Agenda

Application Centric versus Data Centric Thinking

Externalizing Data Access

Views

Warranty Data Consistency

- Check Constraints
- Referential Integrities
- Trigger
- Commitment Control

Row and Column Access Control (RCAC)

03.11.2022

QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page 3



IBM Champion since 2020



Application Centric versus Data Centric

03.11.2022

QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page 4



IBM Champion since 2020

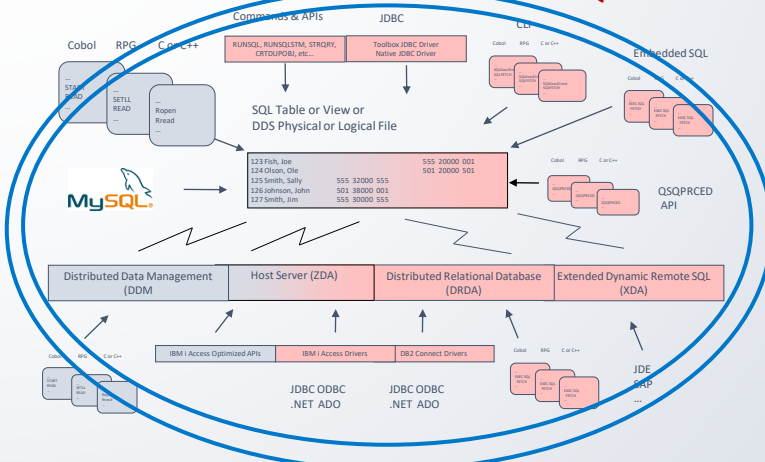


Application-Centric

Native Interfaces

SQL Interfaces

- Expensive in Maintenance and Test
 - Logic is located in the program code
 - Procedures with the same Functionality coded in multiple different programming languages
- Data Security not guaranteed!



Quelle: Db2 Ping Pong - POW3R 2018
Scott Forstie (IBM) / Birgitta Hauser

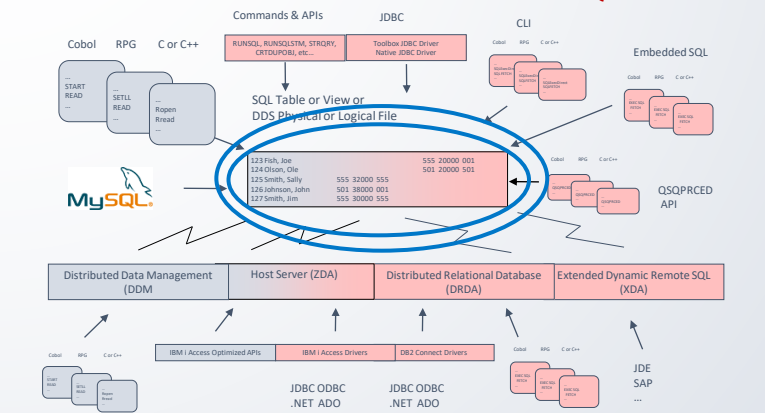


Data Centric

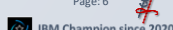
Native Interfaces

SQL Interfaces

- Rules directly linked with the database objects
 - Activated and forced by the Database Manager
 - No extra coding in any Program Language → Reducing the Program Source Code to a minimum
- Data Security guaranteed!



Quelle: Db2 Ping Pong - POW3R 2018
Scott Forstie (IBM) / Birgitta Hauser

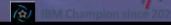


Record Level Access (RLA) versus SQL Access

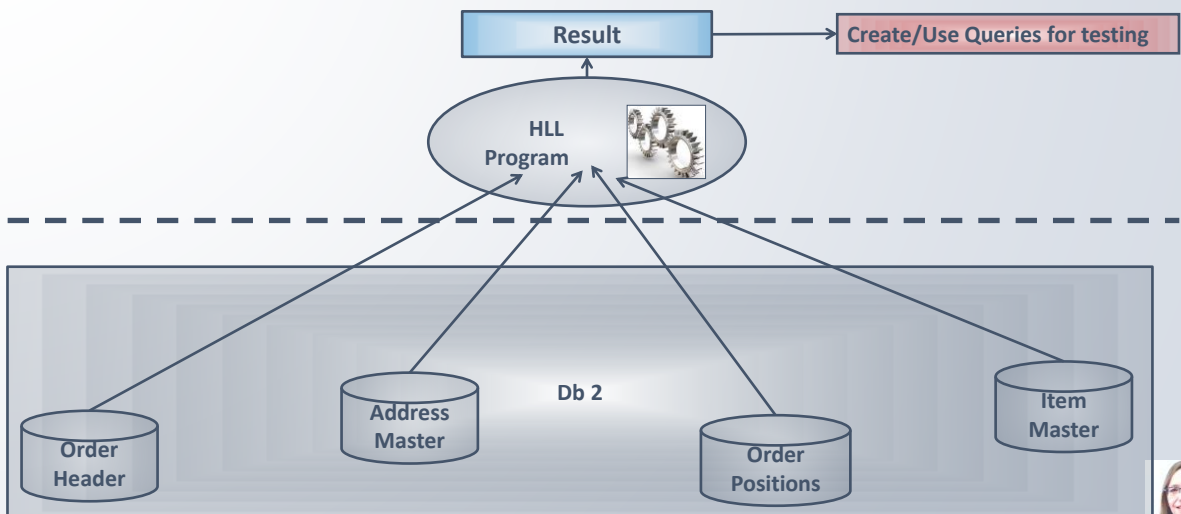
03.11.2022

QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page 7



Traditional Access Method: Record Level Access



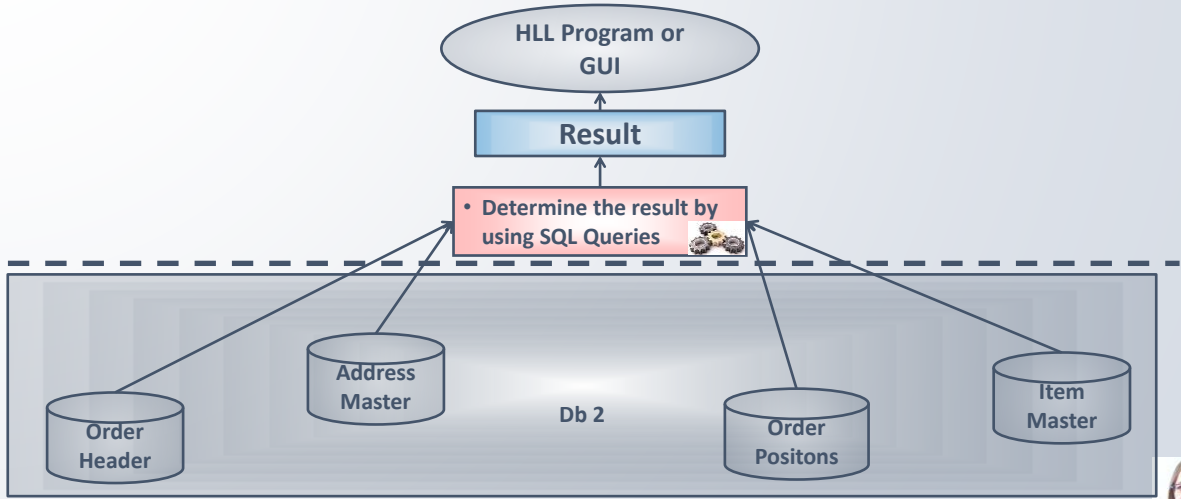
03.11.2022

QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page 8



SQL – Data Centric Programming



03.11.2022

QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page 9

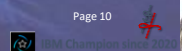


Externalizing Data Access

03.11.2022

QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page 10



Externalizing Data Access – Functions for Insert / Update / Delete

Standard Procedures in Service Programs for Insert/Update/Delete and Single Row Access

- Parameter for Insert/Update: Data Structure **Base View**
- Return Value for Insert: **Unique Key** (e.g. Identity Colum or compound key) or Relative Record No
- Parameter for Delete/Single Row: **Unique Key** and/or **Relative Record No**
- Return Value for Single Row: **Indicator** i.e. Found or Not
- Output Parameter for Single Row: Initialized Data Structure **Base View**
- Additional Parameters: Handling Record Locks, Execution with/without Commitment Control, Send Escape Message if an Error occurs, Update Always

→ **Base Source Code can be automatically generated based on Templates**

Additional Procedures (must be manually completed)

- Delete: Check Procedure, whether a record/row can be deleted or not
- Insert/Update: Check Procedure for Input Values / Initialization with Default Values

These Standard Functions must be used for ALL Insert/Update/Delete Operations

03.11.2022

QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page 11



Externalizing Data Access – Data Centric

Hiding Query Complexity in SQL Views

- Creating a **Base View** (for each table with all columns and may be additional columns)
- Creating a View for **each task** (based on the **base views**)

Generated Service Programs for each View

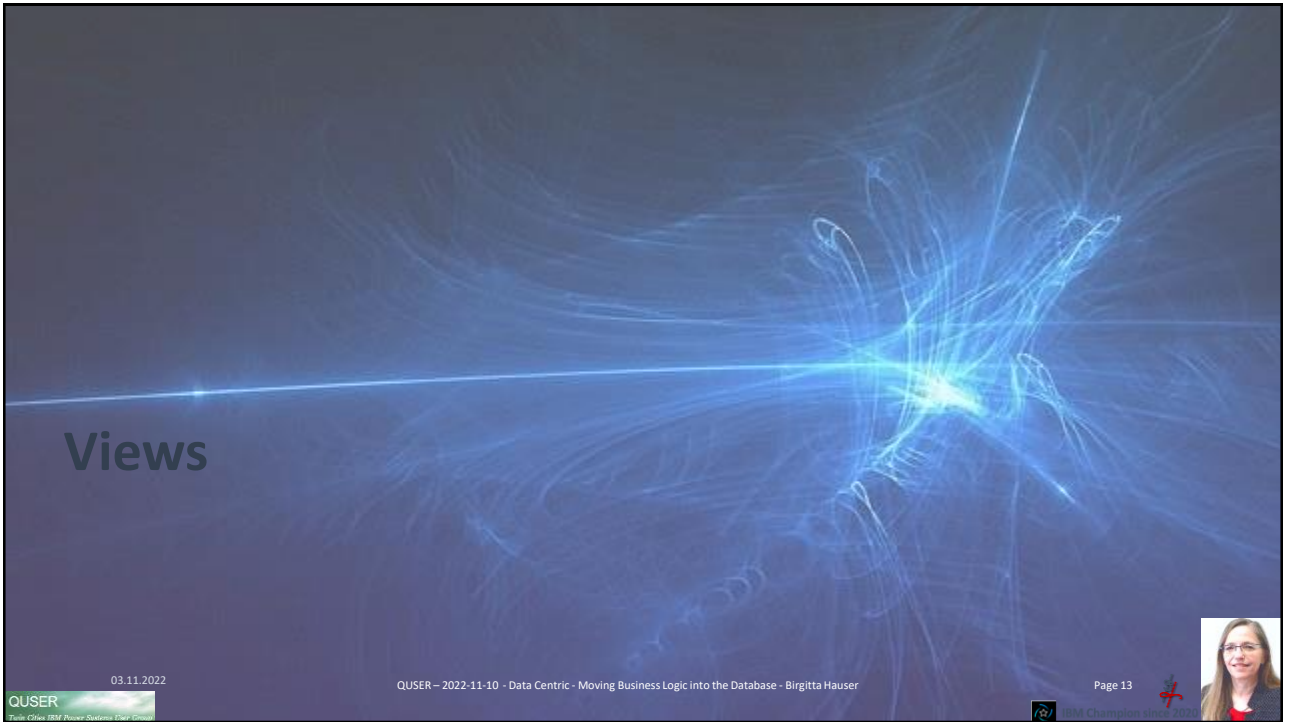
- With **all procedures** that access this specific view
- Use **Call Back Processing** when reading the view in the same loop, but execute different procedures (Procedures can be passed as Procedure Pointers)

03.11.2022

QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page 12





Views


03.11.2022

QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page 13

QUSER
Think. Create. IBM. Power. Systems. Clear. Choice.

IBM Champion since 2020



View – Non-Keyed logical File

Creation with the SQL command **CREATE VIEW**

- Describes the data to be accessed based on a **SELECT** statement
- Does **not** contain any data
- Equivalent to a **non-keyed logical file** object

03.11.2022

QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page 14

QUSER
Think. Create. IBM. Power. Systems. Clear. Choice.

IBM Champion since 2020



View – Non-Keyed logical File

Everything that is allowed in a SELECT statement can be used in a view with the **exception of ORDER BY**

- Column selection and generating new columns
- All types of JOIN expressions (Inner Join, Left/Right/Full Outer Join, Exception Join, Cross Join)
- WHERE conditions
- GROUP BY (including multidimensional grouping) and HAVING clauses
- Scalar Functions / User Defined Functions / User Defined Table Functions
- CASE Expressions
- UNION / EXCEPT / INTERSECT
- Common Table Expressions (CTE) / nested Sub-Selects
- Recursive CTEs and Hierarchical Query Clauses
- Use of Global Variables

A view can be built over an other view → **Nested Views**

- Break complex queries into multiple logical steps

Non-keyed → Access Path Maintenance *REBLD

- Thousands of views can exist **without** any performance decrease



03.11.2022

QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page 15

View – Ungeschlüsselte logische Datei - Beispiele

```

Create View HSCOMMON10.SalesQuart as
Select
  CustNo,
  Year(SalesDate) as SalesYear,
  Cast(sum(case when Quarter(SalesDate)= 1 then Amount else 0 end) as Dec(11, 2)) as Q1,
  Cast(sum(case when Quarter(SalesDate)= 2 then Amount else 0 end) as Dec(11, 2)) as Q2,
  Cast(sum(case when Quarter(SalesDate)= 3 then Amount else 0 end) as Dec(11, 2)) as Q3,
  Cast(sum(case when Quarter(SalesDate)= 4 then Amount else 0 end) as Dec(11, 2)) as Q4,
  Cast(sum(Amount) as Dec(13, 2)) as Total
from Sales
group by CustNo, Year(SalesDate);

```

View: Quartals-Umsätze

```

Create View HSCOMMON10.SalesQCust
as Select
  a.CustNo as ACustNo, SalesYear, Q1, Q2, Q3, Q4, Total,
  b.*
from SalesQuart a Left outer join AddressX b
on a.CustNo = b.CustNo;;

```

View: Verknüpfung der Quartals-Umsätze mit Adress-Stamm

```

Select SalesYear, ACustNo, CustName1, City, Q1, Q2, Q3, Q4, Total
from SalesQCust
where SalesYear = 2009 and ACustNo in ('10003', '10005')

```

Zugriff auf View SalesQCust

| SALESYEAR | ACUSTNO | CUSTNAME1 | CITY | Q1 | Q2 | Q3 | Q4 | TOTAL |
|-----------|---------|--------------------------|----------------|---------|---------|---------|--------|---------|
| 2009 | 10003 | ... Goldbach GmbH | ... Alzenau... | 733,00 | 1057,91 | 2227,45 | 571,50 | 4589,86 |
| 2009 | 10005 | ... Alzenauer Dönertreff | ... Alzenau... | 1475,00 | 285,75 | 1587,50 | 393,70 | 3741,95 |

03.11.2022

QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page 16



View – Modernization - What to do and how to start

Create a basic view containing all fields/columns in the same sequence as in the based physical file/table

- **No longer access** the based physical file/table **directly**
→ Using **Instead Of Triggers** will allow a future redesign of your database

Create additional views based on this basic view

- Joins with other (basic) views: e.g. Join OrderHeader, Addresses, OrderPositions, ItemMaster

Create a view for each problem to solve

- Move business logic into database: e.g. Outstanding Orders, accumulated stocks / item no
- If circumstances change, only one or several views must be updated

Force your programmers and query users to **only access these views**

- Use them with embedded SQL, ODBC/JDBC, Query/400, Db2 WebQuery
- When accessing those views, do not use SELECT *
 - If a view is changed, recompilation is only necessary if new columns are added and used
 - Listing Columns may result in performance gains (Less logical DB access, IOA access)

03.11.2022

QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page 17


 IBM Champion since 2020


Data Integrity

03.11.2022

QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page 18

 IBM Champion since 2020


Data Integrity

What is data integrity?

- Data integrity is the principle of **ensuring data values** between tables are **kept in a state** that makes sense to the business.

Methods to accomplish data integrity:

- Constraints** are **rules enforced** by the **database manager**
 - Check Constraints** Check values within a column
 - Key Constraints** Prevent from duplicate rows/records
 - Referential Integrities** Defines dependencies between tables - Primary and Foreign Key Constraints
- Trigger** are **programs activated** by the **database manager**
 - depending on the **trigger time** (before / after)
 - the **trigger event** (insert / update / delete / instead of)

Advantages

- The definition of **business roules** can be **centralized**
- Reduced Source Code** Business roules are moved into the database
- SQE optimizer: **Constraint awarness.**



Methods to accomplish Data Integrity

Unique Key Access Path:

- UNIQUE** Keyword in physical/logical files or SQL Indexes

Unique/Primary Key Constraints

- Adding Primary and Unique Key Constraints to SQL Tables
- Adding Unique Key Constraints: **ALTER TABLE SQL** Command / **ADDPFCST** CL Command

Check Constraints

- Checks on **Column Level**
 - Compare the inserted/modified value with a **predefined List of Values** or a **Data Range**
 - Compare the inserted/modified value with values in **other columns** in the **same row**
- Adding Check Constraints: **ALTER TABLE SQL** Command / **ADDPFCST** CL Command
- Queries with SQE: **Constraint Awarness**



Check Constraints - Examples

```

Alter Table HSCOMMON10.OrderHdr
Add Constraint HSCOMMON10.OHChkOrderType
Check ( OrderType in ('DO', 'UO', 'EX') );

Alter Table HSCOMMON10.OrderHdr
Add Constraint HSCOMMON10.OHChkDelTerms
Check ( DelTerms in ('CPT', 'EXW', 'FOB') );

Alter Table HSCOMMON10.OrderHdr
Add Constraint HSCOMMON10.OHChkStatus
Check ( Status in ('CD', 'CL', 'GT', 'PD') );

-- 2. Order Detail
Alter Table HSCOMMON10.OrderDet
Add Constraint HSCOMMON10.ODChkQuantity
Check( OrderQty >= DelQty
and OrderQty > 0
and DelQty >= 0 );

Alter Table HSCOMMON10.OrderDet
Add Constraint HSCOMMON10.ODChkStatus
Check( Status in ('CD', 'CL', 'GT', 'PD') );

-- 3. Item Master
Alter Table HSCOMMON10.ItemMast
Add Constraint HSCOMMON10.IMChkItem
Check ( Substr(ItemNo, 1, 2) Between 'AA' and 'ZZ'
and Substr(ItemNo, 3, 3) Between '000' and '999');

Alter Table HSCOMMON10.ItemMast
Add Constraint HSCOMMON10.IMChkPrice
Chk(Preise > 0,50);

```

• List of Values

• Comparison with an other column

• Defining a range in composition with predicates and scalar functions

03.11.2022

QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page 21

 IBM Champion since 2020


Methods to accomplish Data Integrity – Referential Integrities

Referential Integrities

- Dependencies between different Tables
 - Forcing Business Rules
 - Definition Rules when **Changing Key Values**
 - Definition Rules when **Deleting the Parent Rows**
- Activated as soon as an **Insert / Update / Delete** Operation is performed
- Adding Ref. Integrities: **ALTER TABLE** SQL Command / **ADDPFCST** CL Command

03.11.2022

QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page 22

 IBM Champion since 2020


Methods to accomplish Data Integrity – Referential Integrities

Constraint that deals with **relationships** between tables

- No Order Header without Order Position
- Item Master cannot be deleted as long as the item is on stock

Advantages

- **Less coding** Rules are **directly** implemented within the database
- **Better performance** **DBMS handles** these rules **faster** than a user-written application program
- **Better portability** Business rules are **not hidden** in the source code of the program
- **More secure** Business and data integrity rules **cannot be circumvented** neither by a faulty nor manually nor an incompletely written application.

Must be planned and implemented **carefully**

- Dependencies between tables within applications **must be known**
- Inserts/Updates/Delete must be in a predefined sequence



03.11.2022

QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page 23



CREATE / ALTER TABLE SQL Commands for Adding Referential Integrities

```
ALTER TABLE YourTableName
ADD CONSTRAINT YourConstraintName
FOREIGN KEY (Column1, Column2, ... ColumnN)
REFERENCES ParentTable (Column1, Column2, ... ColumnN)
```

```
ON DELETE {
  NO ACTION
  RESTRICT
  CASCADE
  SET NULL
  SET DEFAULT
```

```
ON UPDATE {
  NO ACTION
  RESTRICT
```

- Dependencies between Tabellen (Parent/Child)
- Action when Changing Key Values
- Action when Deleting rows in dependent Tables

```
Alter Table TableSchema.OrderPos
Add Constraint Cst_OrderHdr_OrderPos_Id
Foreign Key (Pos_Hdr_Id)
References TableSchema.OrderHdr (Hdr_Id)
On Delete Restrict
On Update Restrict;
```

03.11.2022

QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page 24



Methods to accomplish Data Integrity - Triggers

Programs that are **directly linked with the database**

- SQL Tables or DDS described physical files or SQL Views

Programs will be **activated by the database manager**

- Depending on **Trigger Event** → Insert, Update, Delete, Read
- Trigger Time** → Before, After

Activation occurs always independent of the used interface

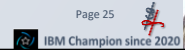
Complement/**enhance** referential integrity / check constraints

- Check an inserted/changed value against values in other tables
- Automatically delete header rows as soon as the last detail is deleted
- Update Summary Tables
- Write History Files

03.11.2022

QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page 25



Methods to accomplish Data Integrity - Triggers

Forcing **Business Rules**

- Rules that cannot be handled by check constraints
- Examples: After having delivered all positions,
copy Order Header and Order Positions into a History Table

Checking **Data Integrities**

- Example: Clerk enters an order → check if he is responsible for the customer

Checking **Data Consistence over multiple Tables**

- Example: Delivery Date is stored in multiple Tables
when changing the delivery date in any table, it is also modified in all other tables

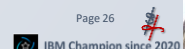
Integrating **new Technologies**

- Example: After a Order is entered, an eMail/SMS will be send to the customer

03.11.2022

QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page 26



Add Trigger Program

```

Create Or Replace Trigger COMDBMOD.ORDER_HEADER_B4IU_DELDATE
Before Insert Or Update Of DELIVERY_DATE_NUM, DELIVERY_DATE
On COMDBMOD.ZZ_ORDER_HEADER
Referencing New as N
Old as O
For Each Row
Mode DB2ROW
Program Name ZZOHBIU01
Not SECURED
Begin Atomic
Declare LocType   VarChar(10) Default '';
Declare Continue Handler for SQLEXCEPTION Set N.Delivery_Date = '8888-12-31';

If   Inserting and N.Delivery_Date > '0001-01-01'
or Updating and O.Delivery_Date <> N.Delivery_Date
Then Set N.Delivery_Date_Num = Dec(N.Delivery_Date, 8, 0);
ElseIf Inserting and N.Delivery_Date_Num > 10101
or Updating and O.Delivery_Date_Num <> N.Delivery_Date_Num
Then Set N.Delivery_Date = Date(Digits(N.Delivery_Date_Num) concat '000000');
ElseIf Inserting
Then Set N.Delivery_Date   = '0001-01-01';
Set N.Delivery_Date_Num = 0;
End If;
End;

```

• Before Insert / Update Trigger for Delivery_Date_Num and Delivery_Date

1. Changed Delivery-Date
Determine/Update Numeric Value
2. Changed Numeric Date
Determine/Update Delivery Date
Invalid Date → Return '8888-12-31'
3. Otherwise
Initialize both dates with the Default



Data Integrity

Add Check Constraints whenever possible

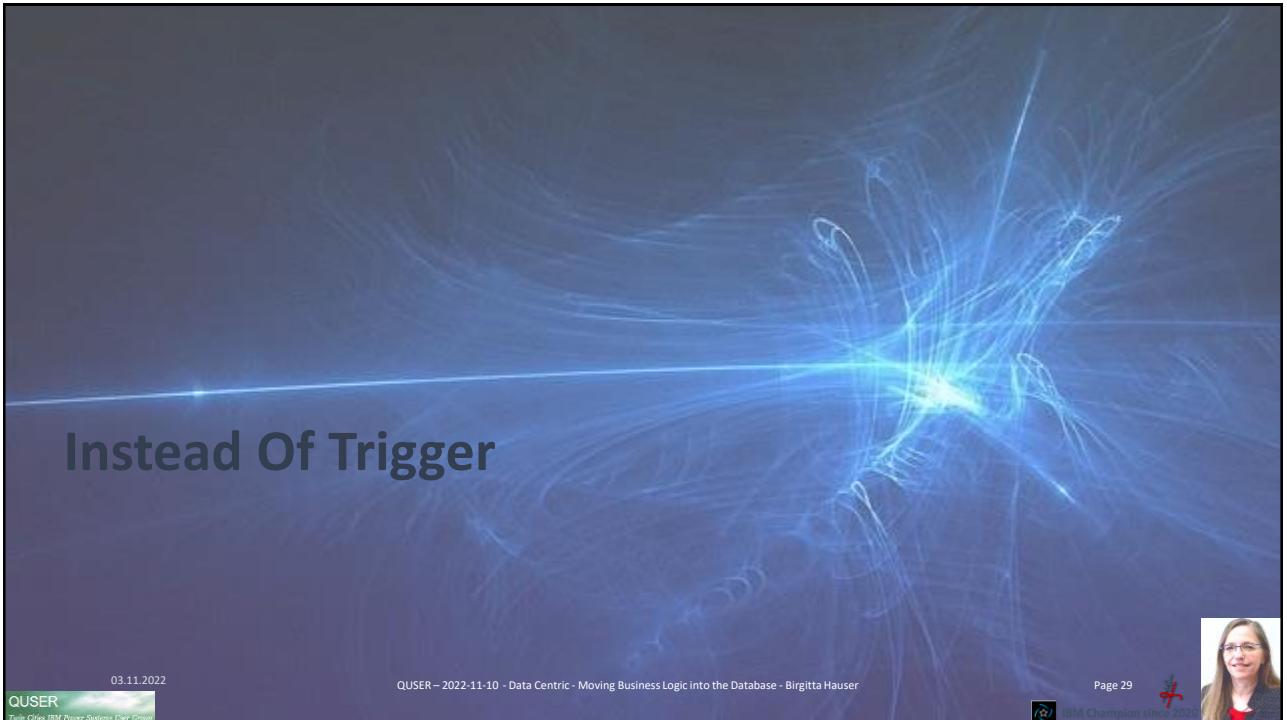
Add Triggers

- **Before** Insert/Update Triggers: Check/Set Column Values
Check whether a row can be deleted or not
Convert numeric date into a real date value and vice versa
- **After** Trigger:
Update Summary Tables
Delete depending Rows (Cascading Delete)
Creating an Order
Send an SMS or other Message

For new tables implement Referential Integrities

- For existing tables add Referential Integrities whenever possible
→ but analyze your programs first





Instead Of Trigger


03.11.2022

QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page 29

QUSER
Think. Create. IBM Power Systems. Clear Choice.

IBM Champion since 2020



Instead Of Trigger

Special type of SQL Trigger

- Special Event: **Instead Of**
- Trigger program linked with a **SQL views**
 - Only **SQL trigger** → External trigger are not allowed
 - DDS described (joined) logical files are not allowed
 - Can only be used in composition with the **SQE**

What are Instead Of Triggers for?

- Set **Default Values** when inserting a row
- Allows Insert/Updates/Deletes on **non-updateable** (joined) **views**
 - The trigger program will update the underlying base tables
 - Facilitates redesigning your database

03.11.2022

QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page 30

QUSER
Think. Create. IBM Power Systems. Clear Choice.

IBM Champion since 2020



Instead Of Trigger - Example

```

Create Or Replace Table HSCOMMON10.ItemBas
(ItemNo Integer Not Null,
Description VarChar (50) Not Null,
Status Char (2) Not Null,
CrTTimeStmp TimeStamp Not Null with Default,
UpdTimeStmp TimeStamp Not Null with Default '0001-01-01-00.00.000000');

Create Or Replace Table HSCOMMON10.ItemDet
(ItemNo Integer Not Null,
Color Char (20) Not Null,
Material Varchar(50) Not Null,
Hight Dec (5, 3) Not Null,
Width Dec (5, 3) Not Null,
Depth Dec (5, 3) Not Null,
Comment VarChar(256) Not Null,
CrTTimeStmp TimeStamp Not Null with Default,
UpdTimeStmp TimeStamp Not Null with Default '0001-01-01-00.00.000000');

Create Or Replace View HSCOMMON10.ItemBV01
(ItemNo, Description, Status, Color, Material,
Hight, Width, Depth, Comment, Volume)
as Select b.ItemNo, Description, Status, Color, Material,
Hight, Width, Depth, Comment ,
cast(Hight * Width * Depth as Dec(15, 9))
from ItemBas b Left Outer Join ItemDet d on b.ItemNo = d.ItemNo;
    
```

• Table **ITEMBAS**
Item Basic Information

• Table **ITEMDET**
Item Detail Information

• View **ITEMBV01**
Joining Item Basic with Item Details



Instead Of Trigger – Example Insert into Read Only View

```

Insert into ItemBV01 (ItemNo, Description)
Values(100, 'Rocker Gnanpa Hugo');

Sun Apr 23 14:27:32 CEST 2017 | Run Selected...
Insert into ItemBV01 (ItemNo, Description) Values(100, 'Rocker Gnanpa Hugo')
SQL State: 42807
Vendor Code: -158
Message: [SQL0158] View or logical file ITEMBV01 in HSCOMMON10 read-only. Cause . . . . . : Update, delete, insert, merge, or truncate is not allowed. ITEMBV01 in HSCOMMON10 can be used only for read operations. A view or logical file can be used only for read operations if one or more of the following conditions are true: -- The view contains a DISTINCT keyword, GROUP BY clause, HAVING clause, CONNECT BY clause, or an aggregate function in the outer fullselect. -- The view or logical file references more than one table in the outer fullselect. -- The view contains a subquery that refers to the same table as the table of the outer fullselect. A view of this type may be used for inserting rows. -- The view contains UNION, EXCEPT, or INTERSECT in the outer fullselect. -- All the columns of the view are expressions, constants, or special registers. -- All the columns of the logical file are input only. -- The select list of the view omits a column of the based on table that does not allow null values or default values. Inserting into the view is not allowed. Recovery . . . . . : Change the statement to insert, delete, update, merge, or truncate data using the base table of view ITEMBV01. All columns of the table that do not allow null values or default values must be assigned a value when inserting a row into a table or view unless an INSTEAD OF trigger is defined for the view. Try the request again.
Failed statements: 1
    
```

• Insert Row into View **ITEMBV01** **without** Instead Of Trigger
→ Insert in an read only view not allowed



Instead Of Trigger – Example - Create Instead Of Trigger

```

Create Trigger HSCCOMMON10.InsertItem
Instead Of Insert On HSCCOMMON10.ItemBV01
Referencing New As N
For Each Row
Mode DB2SQL

Begin Atomic
  Insert into HSCCOMMON10.ItemBas
  Values(N.ItemNo, N.Description,
        Case When Status = ' ' or Status is NULL
        Then 'A' Else N.Status End,
        Default, Default);

  If N.Color is not NULL or N.Material is not NULL
  or N.Hight is not NULL or N.Width is not NULL
  or N.Depth is not NULL or N.Comment is Not NULL
  Then Insert into HSCCOMMON10.ItemDet
  Values(N.ItemNo,
        Coalesce(N.Color, 'Colored'),
        Coalesce(N.Material, 'Various'),
        Coalesce(N.Hight, 0), Coalesce(N.Width, 0),
        Coalesce(N.Depth, 0), Coalesce(N.Comment, ''),
        Default, Default);

End If;
End
    
```

- Insert - Instead Of Trigger for View ITEMVBV01
- Table ITEMBAS is always written Status is set to A or passed value

- Table ITEMDET is only written if at least one column value is passed
- NULL Values are set to Default values



Instead Of Trigger – Example - Insert into Read Only View

```

Insert into ItemBV01 (ItemNo, Description)
Values(100, 'Rocker Granpa Hugo');

Insert into ItemBV01 (ItemNo, Description, Color, Comment)
Values(200, 'Sofa Aunt Trude', 'Antic Pink', 'Ultra Soft');
    
```

- Insert Row into View ITEMVBV01 with an Instead Of Trigger

```

[ Sun Apr 23 14:21:47 CEST 2017 ] Run Selected...
Insert into ItemBV01 (ItemNo, Description) Values(100, 'Rocker Granpa Hugo')
Statement ran successfully (1,069 ms = 1.069 sec)
1 rows were affected by the statement

[ Sun Apr 23 14:21:50 CEST 2017 ] Run Selected...
Insert into ItemBV01 (ItemNo, Description, Color, Comment) Values(200, 'Sofa Aunt Trude', 'Antic Pink', 'Ultra Soft')
Statement ran successfully (236 ms)
1 rows were affected by the statement
    
```

→ Rows are successfully inserted

Select * from ItemBV01;

| ITEMNO | DESCRIPTION | STATUS | COLOR | MATERIAL | HIGHT | WIDTH | DEPTH | COMMENT | VOLUME |
|--------|--------------------|--------|------------|-------------|-------|-------|-------|------------|-------------|
| 100 | Rocker Granpa Hugo | A | - | - | - | - | - | - | - |
| 200 | Sofa Aunt Trude | A | Antic Pink | ... Various | 0,000 | 0,000 | 0,000 | Ultra Soft | 0,000000000 |



Commitment Control

03.11.2022

QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page 35



IBM Champion since 2020



Commitment Control

Condensing multiple operation to a single **Transaction**

- A **Transaction** includes **multiple individual modifications** on objects, which are handled as **a single Action**
- **Examples** for Transactions:
 - Accounting: Credit / Debit
 - Order: Part Delivery not allowed

Commitment Control ensures

- **ALL** changes within a transaction are executed
- **NONE** of the changes within a transaction is executed
- When cancelling a transaction all changes within the transaction are reset

Begin/End of a Transaction:

Reset a Transaction:

COMMIT

ROLLBACK

03.11.2022

QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page 36



IBM Champion since 2020



Commitment Control and SQL

Prerequisites: Tables are **registered** in a **Journal**

STRCMTCTL is **automatically** executed by the SQL runtime

- **Attention:** Start with Default Values → CMTSCOPE – **Activation Group**



Set Commitment Properties:

- Interaktive SQL: F13=Service → 1. Change session attributes
→ Commitment Control = *CHG
- Access Client Solutions (ACS): Connection → JDBC Config./Setup → Server
→ Commit Mode = *CHG
- Insert / Update / Delete Statements: End Statement: **WITH Commitment Level (NC/CS...)**

03.11.2022

QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page 37



Commitment Control and SQL

Modifying Data in Tables

- Execute **COMMIT** first - **BEFORE** Insert / Update / Delete
- Run your Insert, Update, Delete Statement
- Check the modifications with a SELECT Statement
 - Wrong Result: **Reset with ROLLBACK**
 - Correct Result: **Confirm** modifications with **COMMIT**

03.11.2022

QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page 38



Row And Column Access Control

03.11.2022

QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page 39



IBM Champion since 2020



What is RCAC?

RCAC = Row and Column Access Control

- **Additional layer of Data Security** (available with Db2)
 - CL commands for securing (database) objects can be used additionally
- **Complementary to Object Level Security**
- **Limits access to only the required data**
 - Controls access to a table at the **row and/or column level**
 - ***ALLOBJ users can no longer freely access all of the data in the database**

Provides 2 different approaches

- Access permissions for **rows** → CREATE PERMISSION
- Provides masks for **column contents** → CREATE MASK

IBM Advanced Data Security feature for i

- **Must be installed** → No-charge feature, option 47
- Required on **both development and production systems**

03.11.2022

QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page 40



IBM Champion since 2020



Why to use RCAC?

Current methods for limiting data access

- By defining and using **SQL Views**
- Access rules are included within the **application logic** → programming required

Access restrictions can be circumvented

- By accessing the database **table directly**
 - With interactive SQL, Db2-WebQuery, Query/400, JDBC, ODBC native I/O, UPDDTA etc.
- Users with **object authority** (e.g. *ALLOBJ) still can view **ALL data**

With RCAC **ALL** data access can be controlled at **Column/Row Level**

- **Independent** of which access method is used i.e. SQL, native I/O, CL, ODBC
- **No dependency** on application logic → **Business logic is moved into the database**
- Facilitates **multi-tenancy**
 - several independent customers/business units can **share a single database table** without being aware of one another
 - ensures each user **only sees the rows and column values** they are **truly entitled** to view



Data Access with Row Permission - Example

```
CREATE OR REPLACE PERMISSION HSCCOMMON10.MYADDRESS_PERMCUSTNO ON HSCCOMMON10.MYADDRESS AS MACUSTNO
FOR ROWS
WHERE Substr(CustNo, 1, 1)
  between case
    When Verify_Group_for_User(Session_User, 'HAUSERB') = 1 Then '0'
    When Verify_Group_for_User(Session_User, 'QPGMR') = 1 Then '5'
  End
  and case
    When Verify_Group_for_User(Session_User, 'HAUSERB') = 1 Then '4'
    When Verify_Group_for_User(Session_User, 'QPGMR') = 1 Then '7'
  End
ENFORCED FOR ALL ACCESS
ENABLE ;
```

- Session_User = **HAUSERB**
- Can access all customer no beginning with **0, 1, 2, 3 or 4**

- Session_User = **HAUSER**
- Member of the **QPGMR** group profile
- Can access all customer no beginning with **5, 6 or 7**

```
Select * from HSCCOMMON10.MyAddress
```

| CUSTID | CUSTNO | CUSTNAME1 |
|--------|--------|----------------------------|
| 45 | 00100 | Pallhuber und Söhne |
| 46 | 00110 | Bahnleitner Gemischtwaren |
| 47 | 00120 | Ebäcko Nordrhein Westfalen |
| 48 | 00130 | Deutscher-Paket-Dienst |

```
Select * from HSCCOMMON10.MyAddress
```

| CUSTID | CUSTNO | CUSTNAME1 |
|--------|--------|-------------------------|
| 31 | 56453 | GWINNER WOHNDESIGN GMBH |
| 23 | 63820 | FIRMA MAYER GMBH |
| 29 | 63899 | HELLSTERN GMBH |
| 33 | 66215 | GÜNTHER NETZER GMBH |
| 36 | 66588 | ARBURG GMBH |



Data Access with Column Mask Example

```
CREATE or Replace MASK COMRCAC.PEREPLSALARY
ON COMRCAC.EMPLOYEE
FOR COLUMN SALARY
RETURN Case When Verify_Group_For_User(Session_User, 'HAUSERB') = 1
Then Case When salary < 100000 Then Salary Else -999 End
When Verify_Group_For_User(Session_User, 'QPGMR') = 1
and CostCenter = 344 Then Salary
When Verify_Group_For_User(Session_User, 'HAUSERHR') = 1
Then Salary Else -999 End
ENABLE;
```

```
Select Id, EmployeeNo, CostCenter,
Salary, LastName, FirstName
from Employee
Order By Salary Desc
```

| ID | EMPLOYEENO | COSTCENTER | SALARY |
|----|------------|------------|----------|
| 4 | 2000 | 100 | -999,00 |
| 2 | 1010 | 344 | 70000,00 |
| 1 | 1000 | 344 | 55000,00 |
| 3 | 1020 | 344 | 55000,00 |
| 5 | 2100 | 111 | 45000,00 |
| 6 | 1030 | 344 | 30000,00 |

- Session_User = HAUSERB
- Can see all salaries lower than 100000

```
Select Id, EmployeeNo, CostCenter,
Salary, LastName, FirstName
from Employee
Order By Salary Desc
```

| ID | EMPLOYEENO | COSTCENTER | SALARY |
|----|------------|------------|----------|
| 4 | 2000 | 100 | -999,00 |
| 2 | 1010 | 344 | 70000,00 |
| 1 | 1000 | 344 | 55000,00 |
| 3 | 1020 | 344 | 55000,00 |
| 5 | 2100 | 111 | -999,00 |
| 6 | 1030 | 344 | 30000,00 |

- Session_User = HAUSER
- Member of Group Profile QPGMR
- Sees only salaries of cost center 344

Employee Master and Credit Card No - Examples

```
Select EmployeeNo, EMCCID "CredCard Id",
LastName, FirstName, Street, ZipCode, City
from EmplCred
Order By EMCCID;
```

• Employee Master

| EMPLOYEENO | CredCard Id | LASTNAME | FIRSTNAME | STREET | ZIPCODE | CITY |
|------------|-------------|----------|-----------|---------------------|---------|---------------|
| 2100 | 2 | Gerber | Kim | Am Bach 3 | 85051 | Ingolstadt |
| 1000 | 5 | Fischer | Fritz | Oberfeldstr. 16 | 76149 | Karlsruhe |
| 1010 | 11 | Meier | Anna | Frankfurter Str. 55 | 63128 | Dietzenbach |
| 1020 | 15 | Bauer | Stefan | An der Havel 234 | 10785 | Berlin |
| 1030 | 17 | Moser | Ben | Waldstr 1 | 77880 | Sasbach |
| 2000 | 19 | Schmidt | Anton | Seestr. 7 | 17192 | Waren/Mueritz |

- Credit Card Id within the Employee Master
- Whether a credit card no is shown completely or only the last 4 digits depends on information located within the Employee Master (Cost Center)

```
Select CCID "CredCard Id", CredCardNo, CVV,
ExpirYear, ExpirMonth
from CREDCARD;
```

• Credit Cards

| CardId | CREDCARDNO | CVV | EXPIRYEAR | EXPIRMONTH |
|--------|--------------------|-----|-----------|------------|
| 1 | 6011276617038831 | 765 | 2022 | 3 |
| 2 | 5140103430432676 | 221 | 2022 | 9 |
| 3 | 5294479349873539 | 396 | 2021 | 5 |
| 4 | 5302985801465861 | 324 | 2022 | 9 |
| 5 | 5487796361329240 | 323 | 2024 | 9 |
| 6 | 4013987007326163 | 373 | 2021 | 1 |
| 7 | 74042051746869564 | 614 | 2021 | 7 |
| 8 | 84047043193104590 | 207 | 2021 | 2 |
| 9 | 4070199796656947 | 793 | 2021 | 12 |
| 10 | 4092983562516516 | 460 | 2022 | 7 |
| 11 | 14108554684486359 | 210 | 2024 | 11 |
| 12 | 124131077380803421 | 833 | 2022 | 10 |
| 13 | 14134875449544952 | 376 | 2021 | 10 |
| 14 | 4500484439016885 | 602 | 2022 | 3 |
| 15 | 4502245604298774 | 263 | 2024 | 3 |
| 16 | 4508969691406214 | 586 | 2021 | 7 |
| 17 | 4529749847757685 | 127 | 2023 | 3 |
| 18 | 4539759236009026 | 716 | 2021 | 5 |
| 19 | 4563063840841158 | 503 | 2024 | 11 |

Employee Master and Credit Card No - Example

```

Create Or Replace Mask COMRCAC.COLM_CREDCARD_CREDCARDNO
On COMRCAC.CREDCARD For Column CREDCARDNO
Return Case When Verify_Group_For_User(Session_User, 'HAUSERHR') = 1
Then CredCardNo
When Verify_Group_For_User(Session_User, 'HAUSERB') = 1
and (Select CostCenter
      From EmplCred
      Where EMCCID = CCID
      Fetch First Row Only) between 100 and 200
Then CredCardNo
When Verify_Group_For_User(Session_User, 'OPGMR') = 1
and (Select CostCenter
      from EmplCred
      Where emccid = CCID
      Fetch First Row Only) = 344
Then CredCardNo
Else Repeat('*', Length(Trim(CredCardNo)) - 4) concat Right(Trim(CredCardno), 4)
End
Enable;
    
```

• Cost Center from the Employee Master must be checked

• The last 4 digits of the credit card are always readable

```

ALTER TABLE COMRCAC.CREDCARD
ACTIVATE COLUMN ACCESS CONTROL ;
    
```

• Activating Column Access Control



Credit Card Table with Column Masks - Example

```

Select CCID "CredCard Id", CredCardNo, CVV,
      ExpirYear, ExpirMonth
from CREDCARD;
    
```

| CredCard Id | CREDCARDNO | CVV | EXPIRYEAR | EXPIRMONTH |
|-------------|------------------|-----|-----------|------------|
| 1 | *****8831 | 765 | 2022 | 3 |
| 2 | *****2676 | 221 | 2022 | 9 |
| 3 | *****3539 | 396 | 2021 | 5 |
| 4 | *****5861 | 324 | 2022 | 9 |
| 5 | 5487796361329240 | 323 | 2024 | 9 |
| 6 | *****6163 | 373 | 2021 | 1 |
| 7 | *****9564 | 614 | 2021 | 7 |
| 8 | *****4590 | 207 | 2021 | 2 |
| 9 | *****6947 | 793 | 2021 | 12 |
| 10 | *****6516 | 460 | 2022 | 7 |
| 11 | 4108554684486359 | 210 | 2024 | 11 |
| 12 | *****3421 | 833 | 2022 | 10 |
| 13 | *****4952 | 376 | 2021 | 10 |
| 14 | *****6885 | 602 | 2022 | 3 |
| 15 | 4502245604298774 | 263 | 2024 | 3 |
| 16 | *****6214 | 586 | 2021 | 7 |
| 17 | 4529749847757685 | 127 | 2023 | 3 |
| 18 | *****9026 | 716 | 2021 | 5 |
| 19 | *****1158 | 503 | 2024 | 11 |

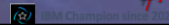


Any Questions?

03.11.2022

QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page 47



Special Thanks to

Holger Scherer – RZKH Rechenzentrum Kreuznach

- For providing an IBM i-System enabling the creation of the samples/code used in my presentations
- <http://www.rzkh.de>



• Your data is save! ... in the bunker



QUSER – 2022-11-10 - Data Centric - Moving Business Logic into the Database - Birgitta Hauser

Page 48



Speaker's Biography

Birgitta Hauser
Diplom-Betriebswirt (BA)
Database and Software Architect

Birgitta Hauser worked on the IBM i and its predecessors since 1992. She graduated with a business economics diploma, and started programming on the AS/400 in 1992. She worked and works as traditional RPG Programmer but also as Database and Software Engineer, focusing on IBM i application and database modernization.

Currently she is self-employed and works in Consulting and Application and Database Modernization on IBM i and Db2 for i.

She also works in education as a trainer for RPG and SQL developers.

Since 2002 she has frequently spoken at the COMMON User Groups and other IBM i and Power Conferences in Germany, other European Countries, USA and Canada.

In addition, she is co-author of two IBM Redbooks and also the author of several articles and papers focusing on RPG and SQL for the ITP Verlag (a German publisher), IT Jungle Guru and IBM DeveloperWorks.

In 2015 she received the John Earl Speaker Scholarship Award. In 2018 she received the Al Barsa Memorial Scholarship Award.



IBM Champion since 2020



Thank you!

Data Centric - Thinking? Yes i will do that!

If you are interested in more detailed individual Workshops on-site or remote,
 Please contact me directly

Birgitta Hauser – Modernization – Education – Consulting on IBM i

Diplom-Betriebswirt (BA)
 Database and Software Architect

Hauser@ModEdCon.com
<https://ModEdCon.com/>

