

DATA REPLICATION SOLUTION IN BOTH DIRECTIONS

// About Proginov...



- **Est. 1996 near Nantes**
- **A staff of 300 people**
- **2 private datacenters**
- **More than 14 000 OpenEdge databases in production**



// About Nantes...



// ... and about us



/ Clément Brodu

- **OpenEdge Software Developer at Proginov since 2016**
- **Framework Team since 2017**
- **DevOps Team since 2018**



/ Patrice Perrot

- **OpenEdge Software developer since 2001**
- **Software Optimisation since 2005**
- **At Proginov since 2008**
- **DBA since 2010**

Data Replication

// Context

/ Apps in production with OpenEdge 10 and OpenEdge 11

/ Start new apps with OpenEdge 12

/ Need to centralize data/metrics in a centralized databases

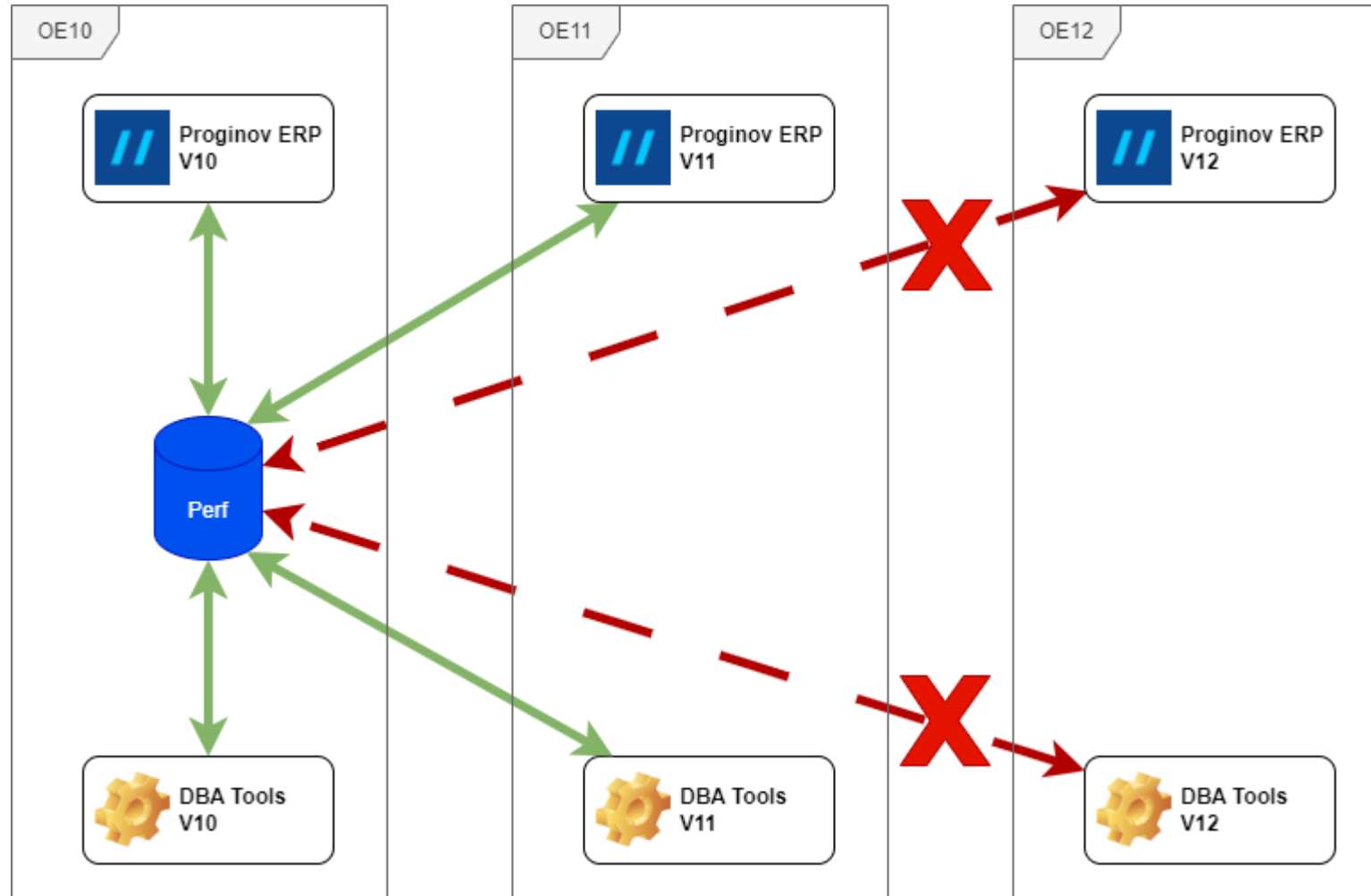
/ No compatibility between 3 OpenEdge versions

- OE12 client can connect to OE12 and OE11 databases
- OE11 client can connect to OE11 and OE10 databases



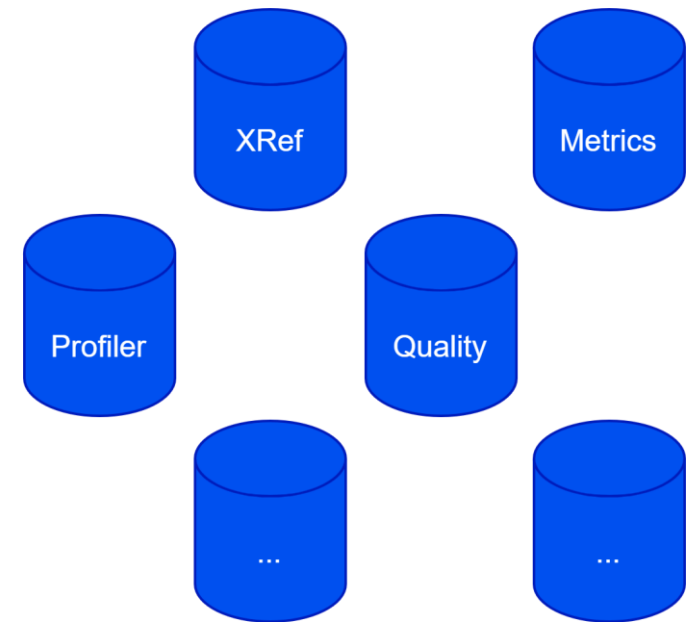
// Context

/ Feed and consult a centralized database



// Constraints

- / 6 databases – 465 tables
- / Many (many) tools, screen, batch
- / Tools runnable everywhere for DBA
- / Data provided from everywhere
- / Edit Virtual System Table (VST)



// Constraints

/ We were in 2022...

/ 40 days before production

- 30 June 2022

/ Databases with 1,5 TB of data

- Automatic data purge after 3 years
- Not allowed to consume more than 2,2 Tb

/ Need Bidirectional replication, but not full replication

- Need recent data everywhere (6 month)
- Older data could be only in 1 databases





// Options

/ Replication with Triggers ?

- ✗ DISABLE TRIGGERS instruction
- ✗ Hard to maintain

/ Replication with Pro2 ?

- ✗ Unidirectional
- ✗ Need change table definition on database
- ✗ based on Triggers (OE10)

/ Webservices and PASOE or AppServer

- ✗ Many programs, procedures and windows to modify
- ✗ Need a completely new architecture
- ✗ Need to update the VST
- ✗ Some processes are based on lock

/ Homemade replication System

- ✓ Change the connected database, it's ready to go
- ⚠ Need to create all the replication logic

// What was the plan

/ Add a new OE 11 database

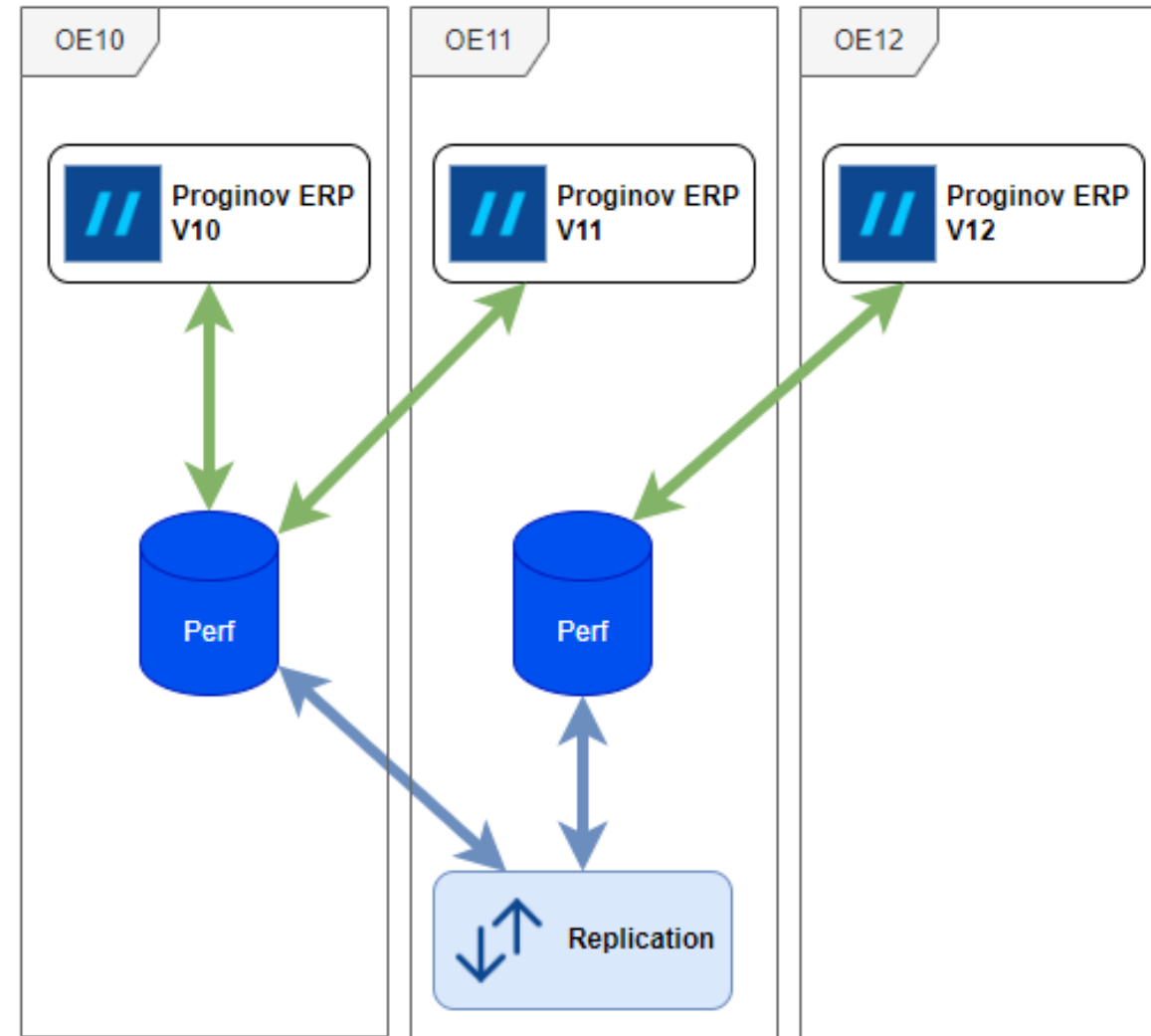
- Can be used from OE 12

/ Create batch process in OE 11

- Can access both databases
- Replicate data

/ Find a Change Tracking System

- For OpenEdge 11
- For OpenEdge 10



// What was the plan

```
changeType = GetChangeType().
CASE changeType:
  WHEN "Create" THEN
    DO:
      newPrimaryKey = GetNewPrimaryKey().
      FIND FIRST Item11 WHERE Item11.Itemnum = newPrimaryKey NO-LOCK NO-ERROR.
      CREATE Item10.
      BUFFER-COPY Item11 TO Item10.
    END.
  WHEN "Update" THEN
    DO:
      oldPrimaryKey = GetOldPrimaryKey().
      newPrimaryKey = GetNewPrimaryKey().
      FIND FIRST Item10 WHERE Item10.Itemnum = oldPrimaryKey EXCLUSIVE-LOCK NO-ERROR.
      FIND FIRST Item11 WHERE Item11.Itemnum = newPrimaryKey NO-LOCK NO-ERROR.
      BUFFER-COPY Item11 TO Item10.
    END.
  WHEN "Delete" THEN
    DO:
      oldPrimaryKey = GetOldPrimaryKey().
      FIND FIRST Item10 WHERE Item10.Itemnum = oldPrimaryKey EXCLUSIVE-LOCK NO-ERROR.
      DELETE Item10.
    END.
END CASE.
```



// What was the plan

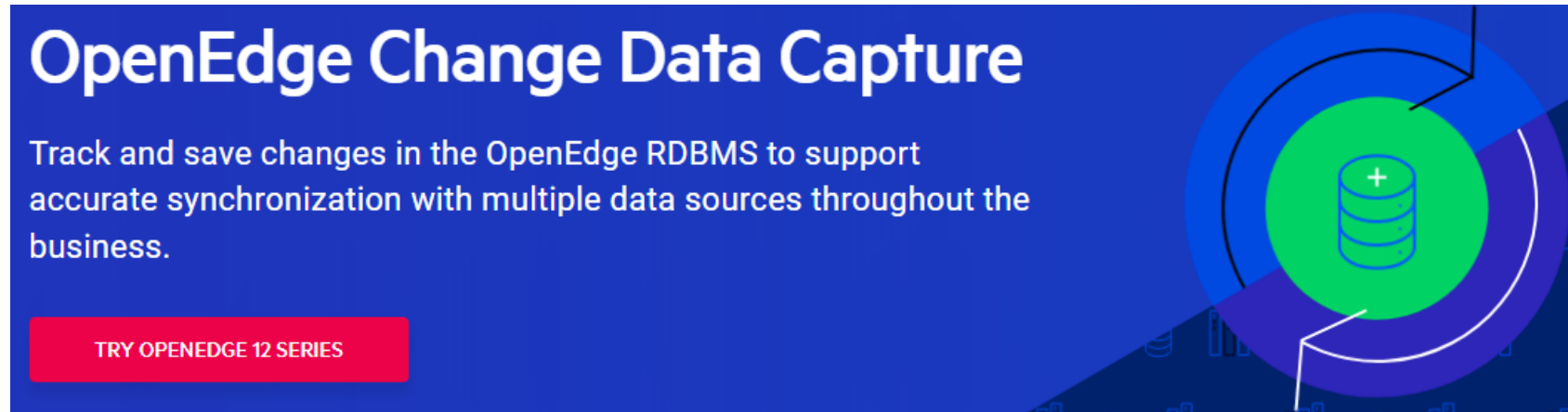
/ Find change tracking tool for OpenEdge

/ Tool need to provide

- Change type (Create / Update / Delete)
- Old Primary Key
- New Primary Key

// Change Tracking Tools

/ OpenEdge 11



OpenEdge Change Data Capture

Track and save changes in the OpenEdge RDBMS to support accurate synchronization with multiple data sources throughout the business.

[TRY OPENEDGE 12 SERIES](#)

The banner features a blue background with a central graphic of a green database cylinder with a white plus sign, surrounded by concentric circles in blue and purple. A red button with white text is positioned at the bottom left of the banner.

/ OpenEdge 10

- CDC not available
- **OpenEdge Auditing**



// CDC vs Auditing

/ Similar purpose and operation

- Use policies to define what to track and how

/ CDC

- ✗ Can't deal with lob
- ✓ Data in relational form
- ✓ Can add custom index

/ Auditing

- ⚠ Create many records per changes
- ⚠ Data in custom form, not easy to deal with
- ⚠ Data stored as string with American format (date MDY,...)
- ✓ Secured with permissions

Change Data Capture

// Enable CDC

/ Create add_cdc.st

```
add_cdc.st  
  
#  
d "CDC_Track_Data":200,64;512 .  
#  
d "CDC_Track_Idx":201,1;64 .
```

/ Enable CDC on database

```
# add CDC areas  
prostrct add %DbName% add_cdc.st  
# enable CDC  
proutil %DbName% -C enablecdc area CDC_Track_Data indexarea CDC_Track_Idx
```

// Enable CDC

/ Add policy to track change on table « Item »

```
initialize_cdc.p

service = new OpenEdge.DataAdmin.DataAdminService(ldbname(1)).
tableName = "Item".

tablepolicy = service:NewCdcTablePolicy(SUBSTITUTE("cdc-policy-&1", tableName)).
tbl = service:GetTable(tableName, "PUB").
ASSIGN
    tablePolicy:Table                = tbl
    tablePolicy:Level                = CdcTablePolicyLevelEnum:Maximum /* medium if the primary key is immutable */
    tablePolicy:State                = CdcTablePolicyStateEnum:Active
    tablePolicy:IdentifyingField     = YES
    tablePolicy:EncryptPolicy        = no
    tablePolicy:ChangeTable          = SUBSTITUTE("cdc-&1", tableName)
    tablePolicy:ChangeTableOwner    = "PUB"
    tablePolicy:DataArea              = service:GetArea("CDC_Track_Data")
    tablePolicy:IndexArea            = service:GetArea("CDC_Track_Idx")
    tablePolicy:Description          = SUBSTITUTE("CDC policy for table &1", tableName)
.
FOR FIRST _file WHERE _file-name = tableName NO-LOCK,
    EACH _field WHERE _file-rcid = Recid(_file) NO-LOCK: /* track changes on each fields in the table */
    fieldPolicy = service:NewCdcFieldPolicy().
    tablePolicy:FieldPolicies:add(fieldPolicy).
    fieldPolicy:Field = tbl:Fields:Find(_field._Field-Name).
    fieldPolicy:IdentifyingField = IF _field._Field-Name = "Itemnum" THEN 1 ELSE ?. /* position in primary key */
END.
service:CreateCdcTablePolicy(tablePolicy).
```


// Enable CDC

/ Add policy to track change on table « Item »

- New table « cdc_Item »

_Tran-id	_Tran-id	zzzzzzzz9
_Time-Stamp	_Time-Stamp	99/99/9999 HH:MM:SS.SSS+HH:MM
_Change-Sequence	_Change-Sequence	zzzzzzzz9
_Continuation-Position	_Continuation-Position	zzzzzzzz9
_ArrayIndex	_ArrayIndex	zzzzzzzz9
_Fragment	_Fragment	zzzzzzzz9
_Operation	_Operation	>>, >>9
Itemnum	Item Num	zzzzzzzzzz9
ItemName	Item Name	x(25)
Price	Price	->, >>>, >>9.99

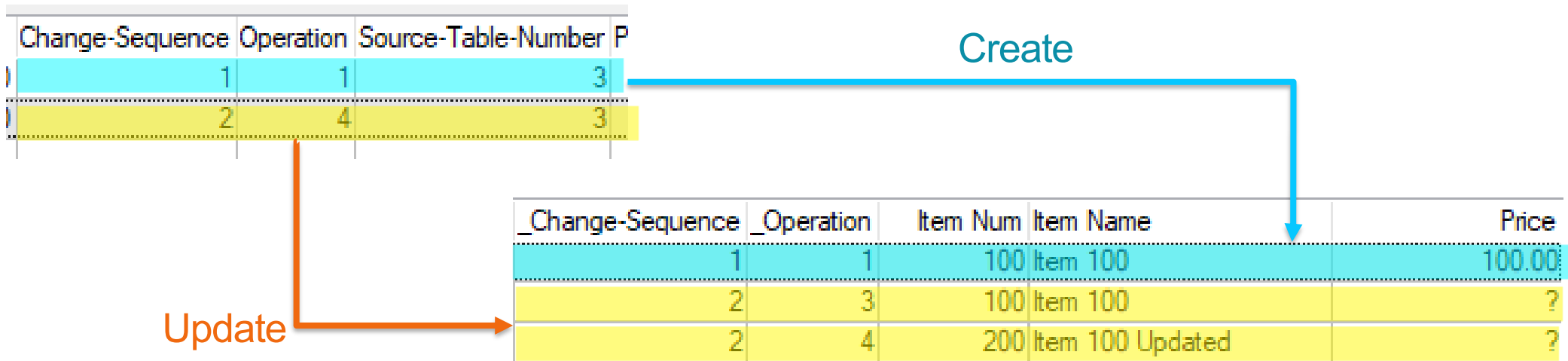
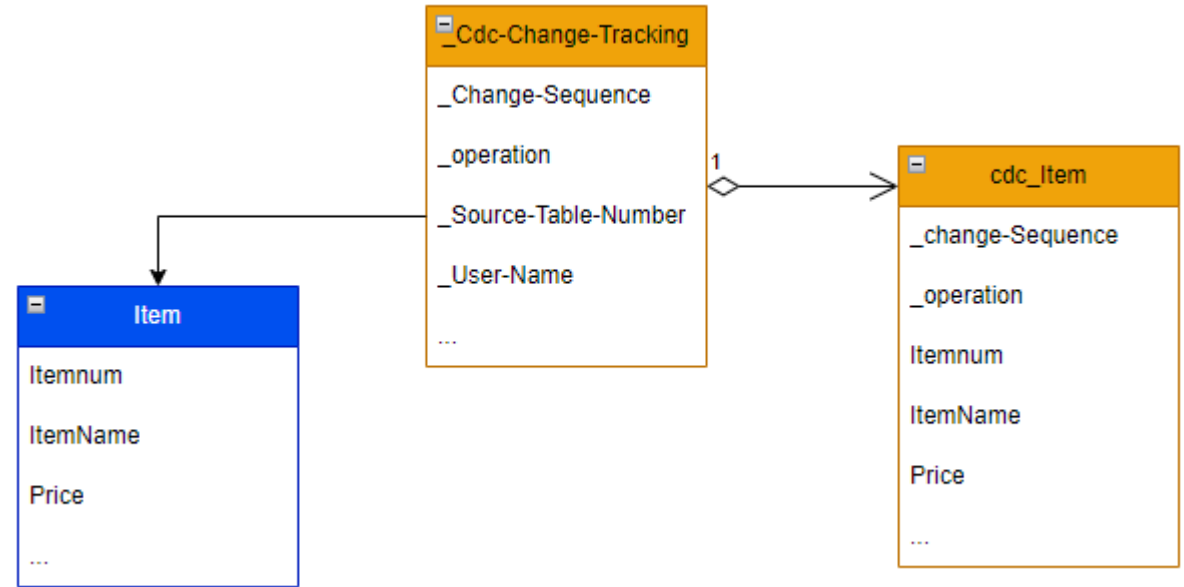
- ... and all other fields

- Linked to « _Cdc_Change-Tracking »

// How CDC works

/ 1 header per action

/ 1 or 2 « cdc_Item » per change



// Usage of CDC

/ Retrieve changes information

```
DEFINE BUFFER Cdc-Change-Tracking FOR sp2k11._Cdc-Change-Tracking.

ohelper = NEW CDCTrackingHelper ("Item" , BUFFER Cdc-Change-Tracking:HANDLE ).
sourceTableNumber = ohelper:SourceTableNumber.

FOR EACH Cdc-Change-Tracking WHERE Cdc-Change-Tracking._Source-Table-Number = sourceTableNumber NO-LOCK:

    CASE Cdc-Change-Tracking._Operation:
        WHEN 1 THEN changeType = "Create".
        WHEN 2 THEN changeType = "Delete".
        WHEN 4 THEN changeType = "Update".
    END CASE.

    FIND FIRST cdc_Item WHERE cdc_Item._change-Sequence = Cdc-Change-Tracking._change-Sequence
                        AND cdc_Item._operation          = Cdc-Change-Tracking._operation NO-LOCK NO-ERROR.
    newPrimaryKey = cdc_Item.Itemnum.
    oldPrimaryKey = cdc_Item.Itemnum.

    IF Cdc-Change-Tracking._operation = 4 THEN /* UPDATE - */
    DO:
        FIND FIRST cdc_Item WHERE cdc_Item._change-Sequence = Cdc-Change-Tracking._change-Sequence
                        AND cdc_Item._operation          = 3 NO-LOCK NO-ERROR. /* before values record */
        oldPrimaryKey = cdc_Item.Itemnum.
    END.

    /* Do replication ... */
END.
```

Auditing

// Enable Auditing

/ Create add_audit.st

```
add_audit.st

#
d "AuditData":210,64;512 .
#
d "AuditIndex":211,1;64 .
```

/ Enable Auditing on database

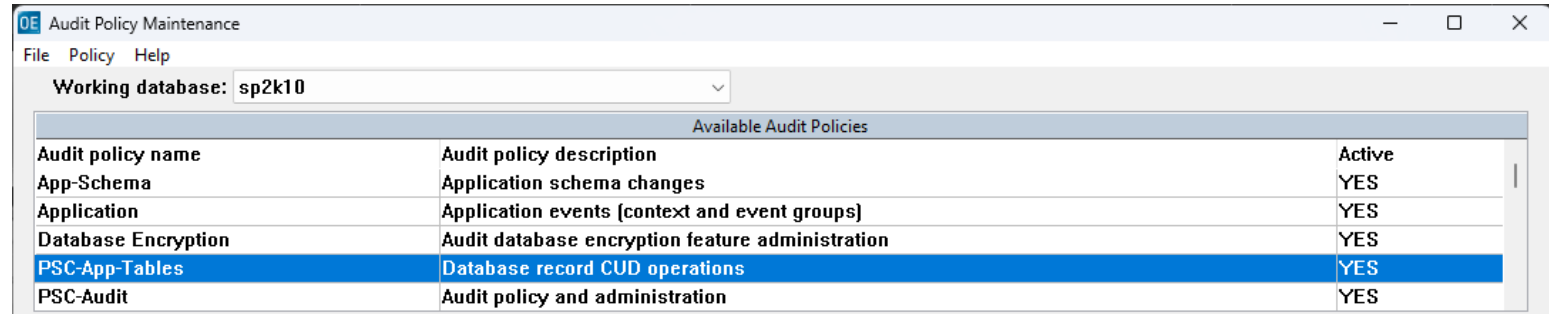
```
# add Auditing areas
prostrct add %DbName% add_audit.st
# enable Auditing
proutil %DbName% -C enableauditing area AuditData indexarea AuditIndex
```

/ Load default policies

- o auditing/policies.xml

// Enable Auditing

/ Set Table policy

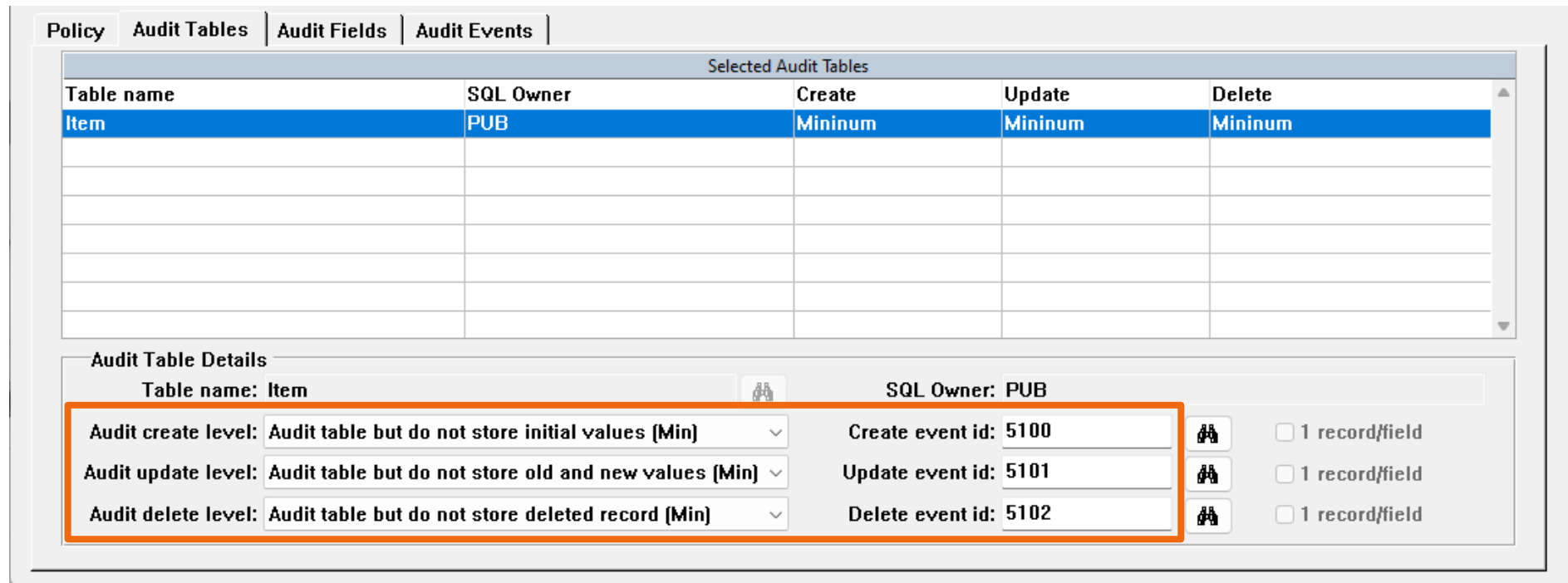


Audit Policy Maintenance

File Policy Help

Working database: sp2k10

Available Audit Policies		
Audit policy name	Audit policy description	Active
App-Schema	Application schema changes	YES
Application	Application events (context and event groups)	YES
Database Encryption	Audit database encryption feature administration	YES
PSC-App-Tables	Database record CUD operations	YES
PSC-Audit	Audit policy and administration	YES



Policy | Audit Tables | Audit Fields | Audit Events

Selected Audit Tables

Table name	SQL Owner	Create	Update	Delete
Item	PUB	Minimum	Minimum	Minimum

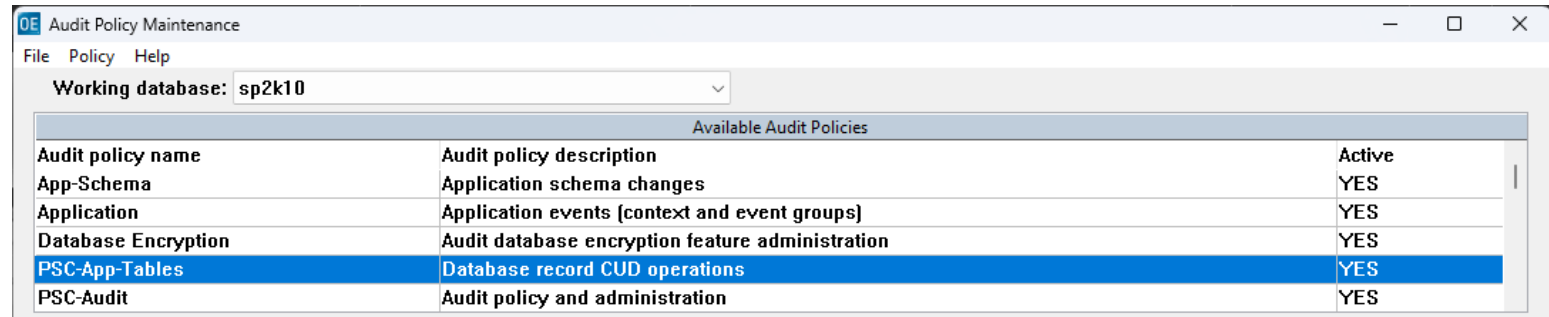
Audit Table Details

Table name: Item SQL Owner: PUB

Audit create level:	Audit table but do not store initial values (Min)	Create event id: 5100		<input type="checkbox"/> 1 record/field
Audit update level:	Audit table but do not store old and new values (Min)	Update event id: 5101		<input type="checkbox"/> 1 record/field
Audit delete level:	Audit table but do not store deleted record (Min)	Delete event id: 5102		<input type="checkbox"/> 1 record/field

// Enable Auditing

/ Set Field policy

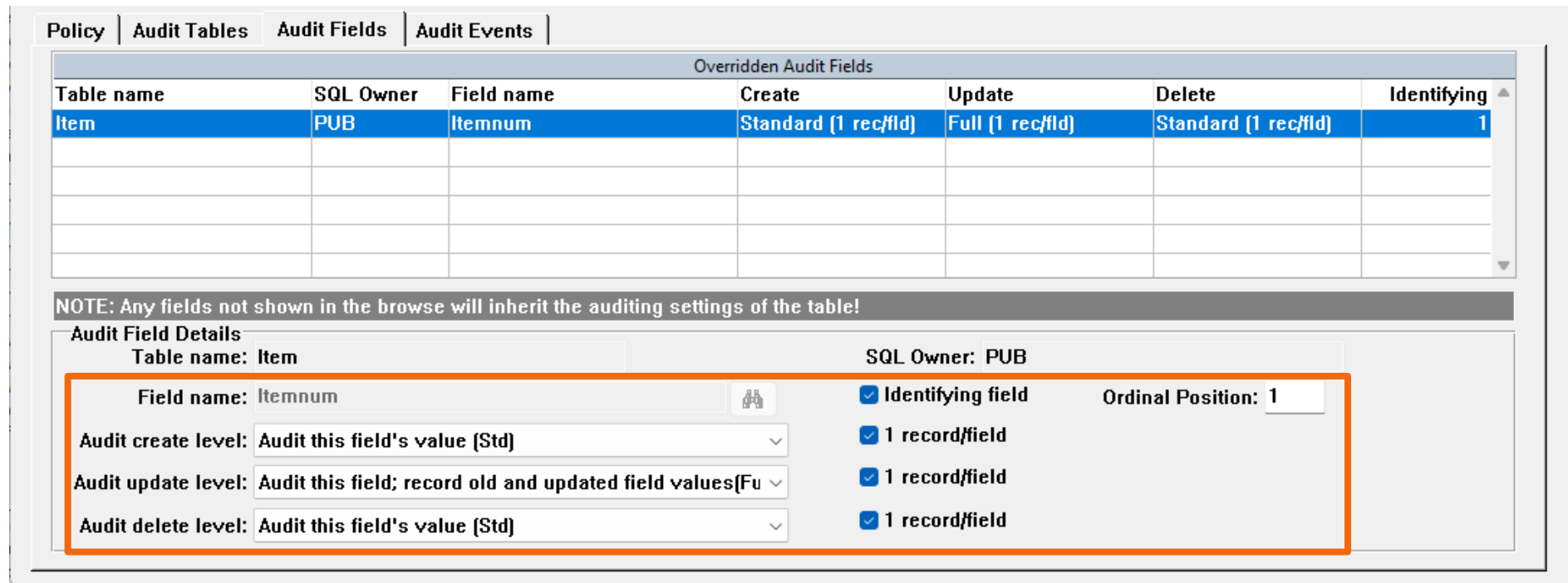


Audit Policy Maintenance

File Policy Help

Working database: sp2k10

Available Audit Policies		
Audit policy name	Audit policy description	Active
App-Schema	Application schema changes	YES
Application	Application events (context and event groups)	YES
Database Encryption	Audit database encryption feature administration	YES
PSC-App-Tables	Database record CUD operations	YES
PSC-Audit	Audit policy and administration	YES



Policy | Audit Tables | Audit Fields | Audit Events

Overridden Audit Fields						
Table name	SQL Owner	Field name	Create	Update	Delete	Identifying
Item	PUB	Itemnum	Standard (1 rec/fld)	Full (1 rec/fld)	Standard (1 rec/fld)	1

NOTE: Any fields not shown in the browse will inherit the auditing settings of the table!

Audit Field Details

Table name: Item SQL Owner: PUB

Field name: Itemnum Identifying field Ordinal Position: 1

Audit create level: Audit this field's value (Std) 1 record/field

Audit update level: Audit this field; record old and updated field values(Fu) 1 record/field

Audit delete level: Audit this field's value (Std) 1 record/field

// Enable Auditing

/ Set User Permissions

Dialog box: Edit Audit Permissions (sp2k10)

UserId	Permission
SYNC_USER	Audit Data Archiver
SYNC_USER	Audit Data Reporter

UserId: SYNC_USER
Permission: Audit Data Archiver
Grantor: DBA_ADMIN_USER
Comments:
 Can Grant Permissions for Audit Data Archiver

Buttons: Done, Grant, Save, Cancel, Revoke, Help

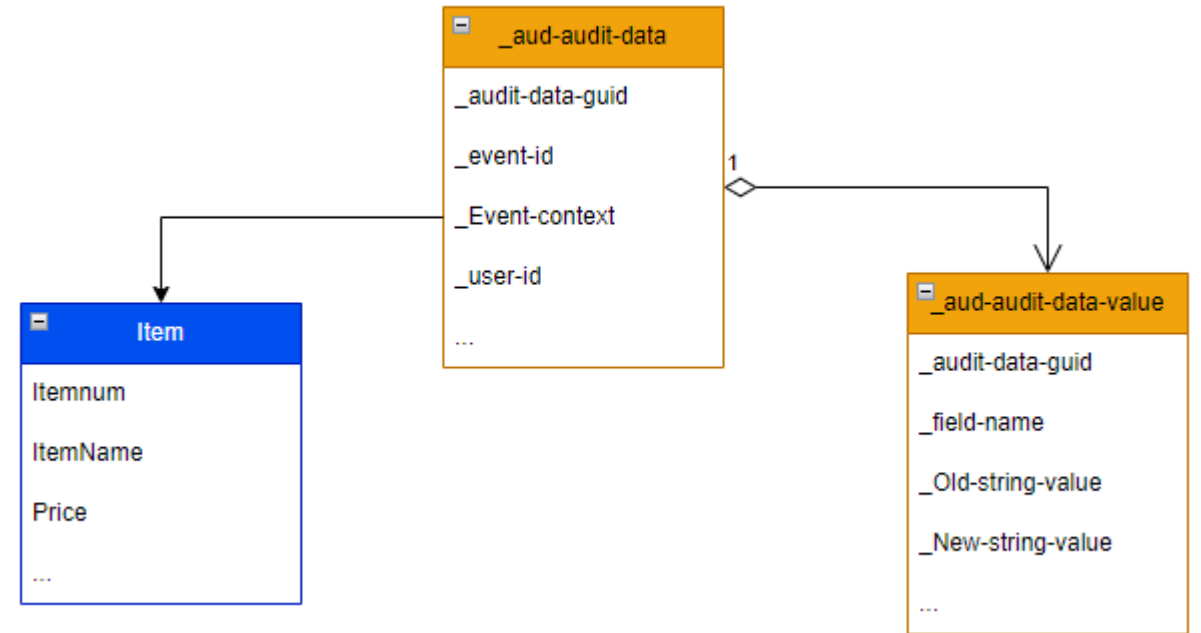
User ID:

// How Audit Works

/ 1 header per change

/ 1 « aud-audit-data-value »

- o per field of primary key



Audit data guid	Event id	Event context
ryYJqrQRJXvEX44662WDQ	5100	PUB.ItemI100
ryYJqrQRJXvEX44YN2WDQ	5101	PUB.ItemI200

Create

Update

Audit data guid	Field name	Data type code	Old string value	New string value
ryYJqrQRJXvEX44662WDQ	Itemnum	4		100
ryYJqrQRJXvEX44YN2WDQ	Itemnum	4	100	200

// Usage of Auditing

/ Retrieve changes information

```
DEFINE BUFFER b_aud-audit-data FOR sp2k10._aud-audit-data.
DEFINE BUFFER b_aud-audit-data-value FOR sp2k10._aud-audit-data-value.

change_event:
FOR EACH b_aud-audit-data
WHERE b_aud-audit-data._Event-context BEGINS SUBSTITUTE('Pub.&1&2', "Item", CHR(6)) NO-LOCK
BY _Audit-date-time :

CASE b_aud-audit-data._Event-id:
WHEN 5100 THEN changeType = "Create".
WHEN 5102 THEN changeType = "Delete".
WHEN 5101 THEN changeType = "Update".
END CASE.

/* get key from _event-context if change is not on identifying fields */
ASSIGN
identifyingFields = ENTRY(2, b_aud-audit-data._Event-context, CHR(6)).
oldPrimaryKey    = INTEGER (ENTRY(1, identifyingFields, CHR(7))).
newPrimaryKey     = oldPrimaryKey.

/* get key from _aud-audit-data-value if change is on identifying fields */
FOR EACH b_aud-audit-data-value WHERE b_aud-audit-data-value._Audit-data-guid = b_aud-audit-data._Audit-data-guid
AND b_aud-audit-data-value._Field-name = 'Itemnum'
AND b_aud-audit-data._Event-id = 5101 /* UPDATE */ NO-LOCK:

ASSIGN
oldPrimaryKey = INTEGER(b_aud-audit-data-value._old-string-value)
newPrimaryKey = INTEGER(b_aud-audit-data-value._new-string-value).

END.

/* Do replication ... */
END.
```

Problems to manage

// Increase -L

Lock Table Entries (-L)

/ 1 change = 4 records (average)

Without Auditing

Create/Update/Delete **Item**

With Auditing

Create/Update/Delete **Item**

- Create **_aud-Audit-Data**

- Create **_aud-Audit-Data-Value ***

Without CDC

Create/Update/Delete **Item**

With CDC

Create/Update/Delete **Item**

- Create **_Cdc-Change-Tracking**

- Create **Cdc-Item ***

/ **More records to write = more locks**

/  **multiply “-L” by 6**

// Transaction & Data integrity

/ Ensure transaction are completed before replication

/ Check if CDC/Auditing records are locked

- If locked, the transaction is not finished



```
change_event:
FOR EACH Cdc-Change-Tracking WHERE Cdc-Change-Tracking._Source-Table-Number = sourceTableNumber NO-LOCK :
  /* try exclusive lock to make sure the transaction is not in progress */
  FIND FIRST Locked_Cdc-Change-Tracking WHERE ROWID(Locked_Cdc-Change-Tracking) = ROWID(Cdc-Change-Tracking )
  EXCLUSIVE-LOCK NO-WAIT NO-ERROR.

  IF AVAILABLE Locked_Cdc-Change-Tracking THEN
    /* OK */
```



// Avoid infinite loop

/ Create / Update / Deletes

- from DB11 to DB10 will create new records in Auditing tables
- from DB10 to DB11 will create new records in CDC tables

/ Use a dedicated “SyncUser”

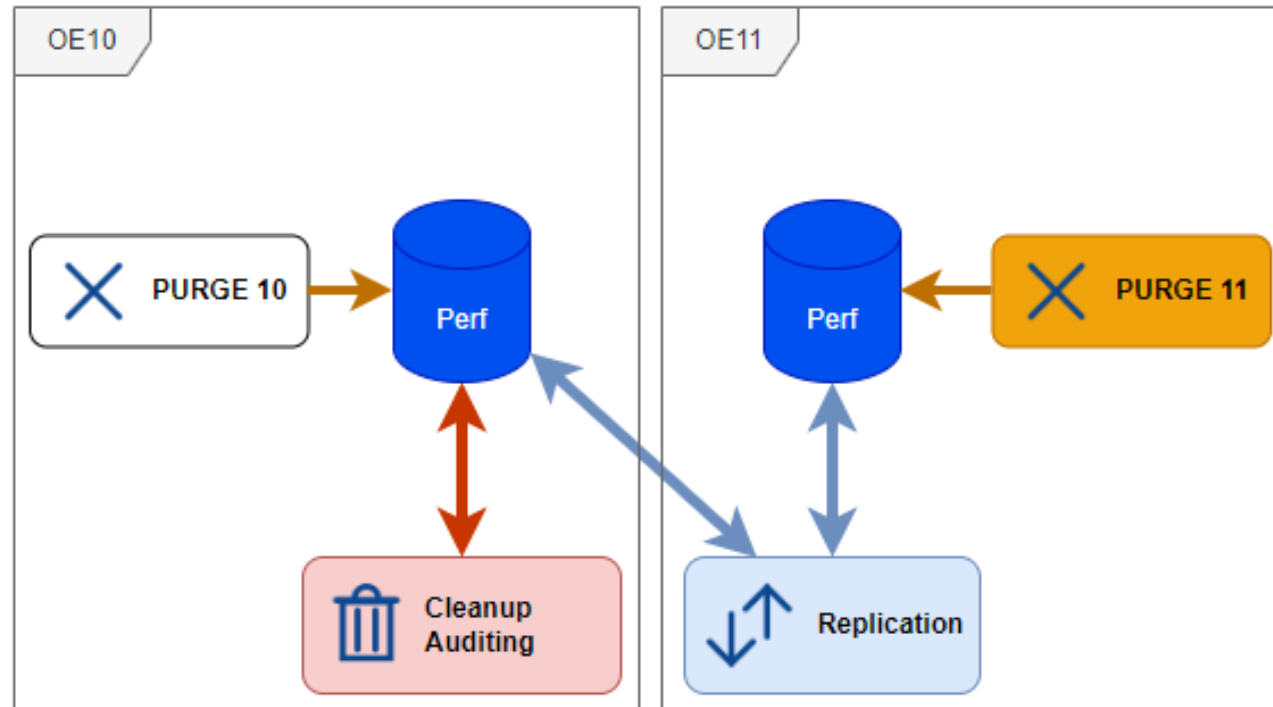
- Ignore everything from this user

// Adapt existing purge process

/ Add purge processes for new database (OE 11)

- Same principle than OE10
- Shorter retention (~6month)

/ Use the “SyncUser” to ignore deletion and avoid propagation of removed data



// Sequences management

/ Each database has its own sequences

- Both could create data with same Id:

```
Item.Itemnum = NEXT-VALUE(NextItemNum)
```

/ Define how to deal with sequences between databases

- Increment 2 by 2
- Even on OpenEdge 10 / odd on OpenEdge 11

/ Resync sequences to stay with close values

SEQUENCE				
Seq copiée	Val Mast	Inc Mast	Val Slave	Inc Slave
cb_num_correct	51	2	50	2
cb_num_extract	81	2	80	2
cb_uniq_id	81	2	80	2
...

UPD seq master		UPD seq Slave	
cb_num_correct	cb_num_correct	cb_num_correct	cb_num_correct
Cur Val: 51		Cur Val: 50	
Inc: 2		Inc: 2	



// Choose unique index

/ Some tables do not have a unique index

/ Determine which index should be used

- “should be unique” index 😊

/ Identify it with comment in “_Desc”

/ Allow to generate audit and CDC policies and replication scripts



// Queue Management

/ Scale and prioritize replication to avoid latency

- Define table with higher importance
- Define the right number of batch processes
- Adjust the replication logic between table

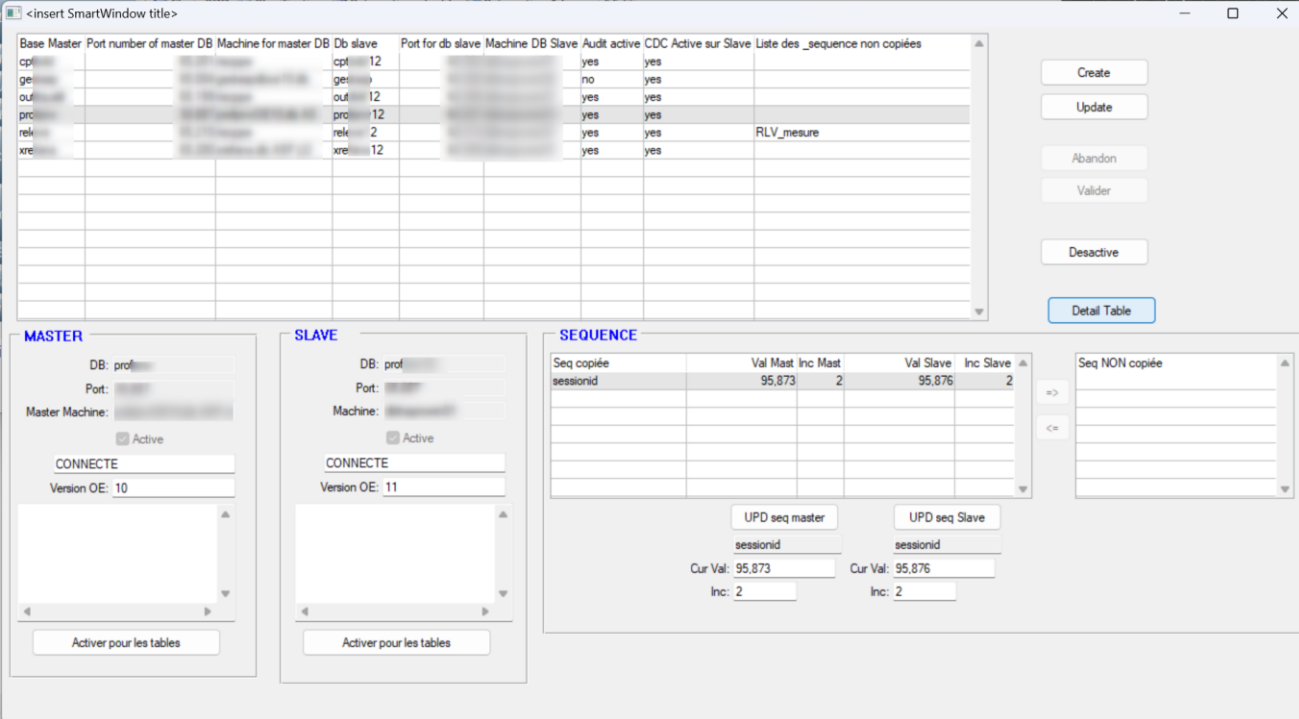
// Administrate Replication

/ Make it easy, automate everything you can !

- Enable CDC / Auditing
- Policy Creation / Update
- Replication script generation

/ New Database for settings

/ Administration console



Base Master	Port number of master DB Machine for master DB	Db slave	Port for db slave Machine DB Slave	Audit active	CDC Active sur Slave	Liste des _sequence non copiées
cpt		12		yes	yes	
ge				no	yes	
out		12		yes	yes	
pro		12		yes	yes	
rel		2		yes	yes	RLV_mesure
xre		12		yes	yes	

MASTER

DB: prof
Port:
Master Machine:
 Active
CONNECTE
Version OE: 10
Activer pour les tables

SLAVE

DB: prof
Port:
Machine:
 Active
CONNECTE
Version OE: 11
Activer pour les tables

SEQUENCE

Seq copiée	sessionid	Val Mast	Inc Mast	Val Slave	Inc Slave	Seq NON copiée
		95.873	2	95.876	2	

UPD seq master
sessionid
Cur Val: 95.873
Inc: 2

UPD seq Slave
sessionid
Cur Val: 95.876
Inc: 2

// Propagate replication programs

/ Every replication settings are saved in our admin database

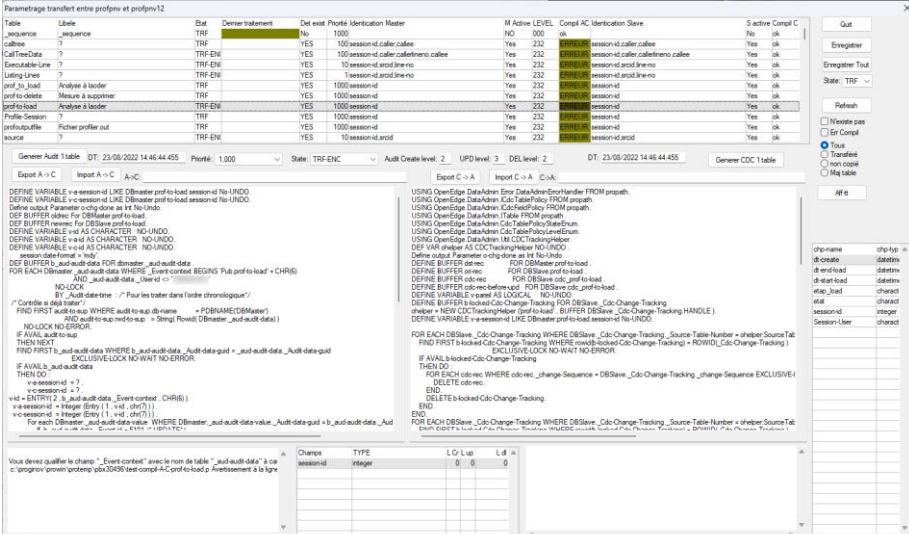
/ Replication programs are generated and editable from console

/ Programs are stored as blob in database

/ Programs are versioned in database

/  **Batch session**

- COPY-LOB in temp directory
- Execute

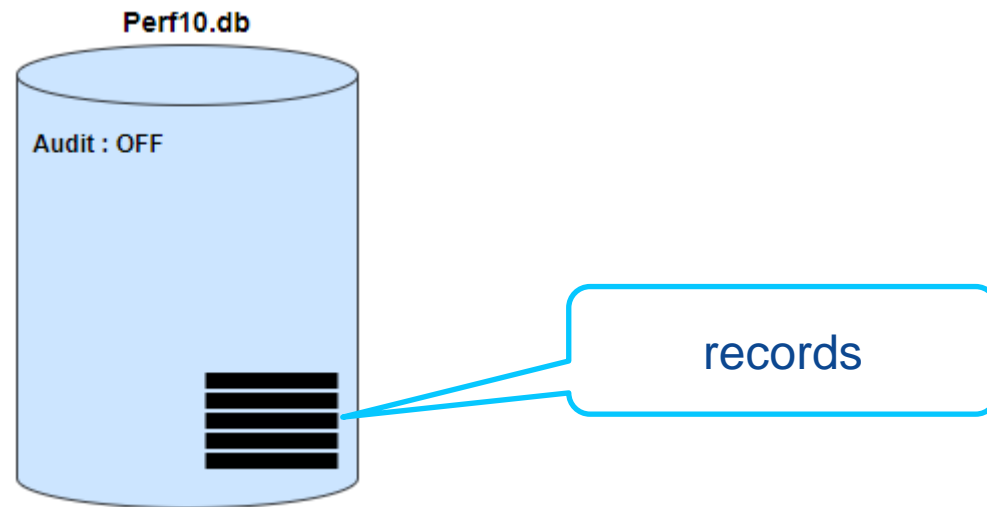


The screenshot displays a software interface for managing replication programs. At the top, there is a table titled 'Paramétrage transfert entre profvis et profvis12'. The table has columns for 'Table', 'Libelle', 'Etat', 'Dernier traitement', 'Del exist', 'Priorité', 'Identification Master', 'M Active LEVEL', 'Compl AC Identification Slave', and 'S active Compl C'. Below the table, there is a section for 'Générer Audit 1 table' with various parameters like 'DT: 23/09/2022 14:46:44.455', 'Priorité: 1.000', 'Etat: TRF-ENC', 'Audit Create level: 2', 'UPD level: 3', 'DEL level: 2', and 'DT: 23/09/2022 14:46:44.455'. The main part of the interface is a SQL script editor showing a complex script for generating audit tables. The script includes comments in French and SQL code for defining variables, creating tables, and performing data operations. At the bottom, there is a small table with columns 'Champs', 'TYPE', 'L.O.L up', and 'L.d', containing values 'sessionid', 'integer', '0', '0', and '0'.

Go to Production

// Create a new database

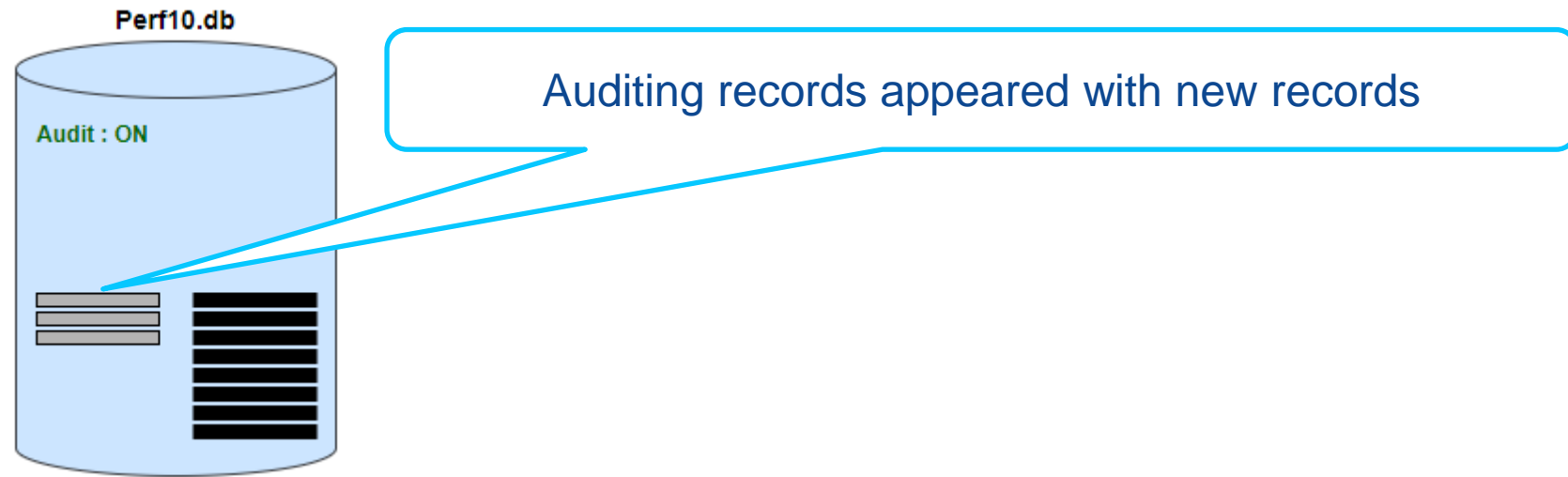
/ Existing OpenEdge 10 Database



// Create a new database

/ Existing OpenEdge 10 Database

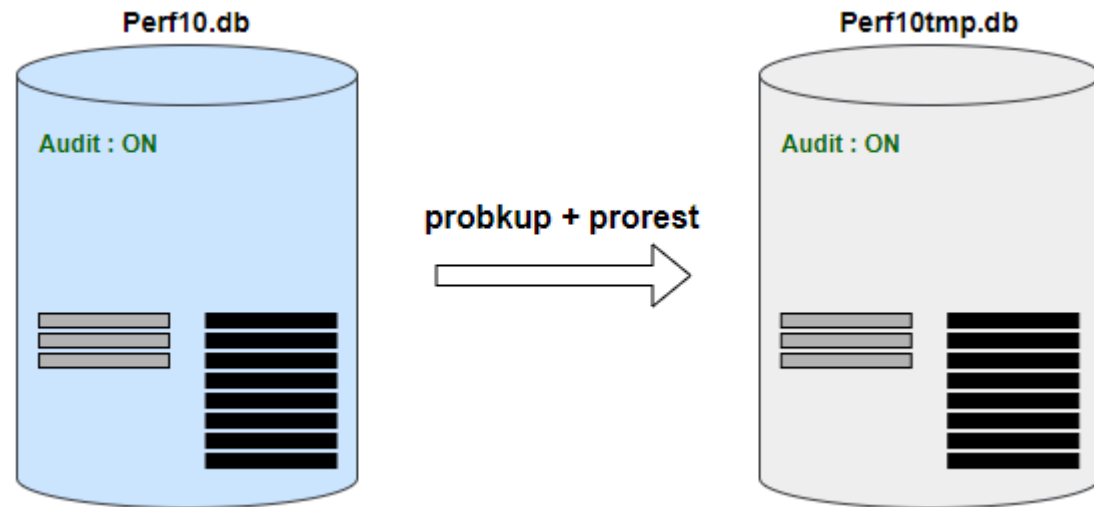
- Enable Auditing



// Create a new database

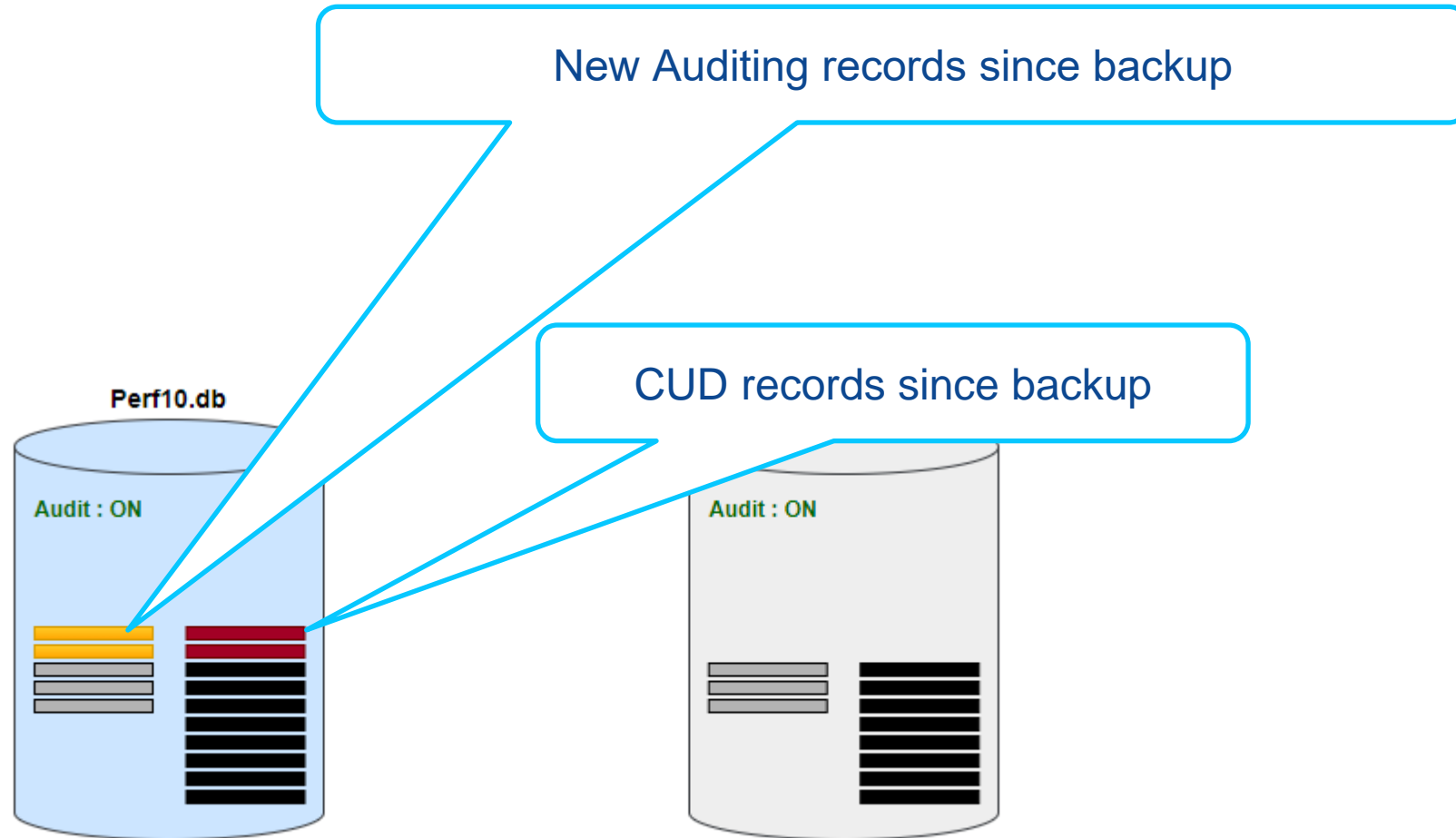
/ Create a temporary OpenEdge 10 database

- Duplicate the db with “probkup” + “prorest”



// Create a new database

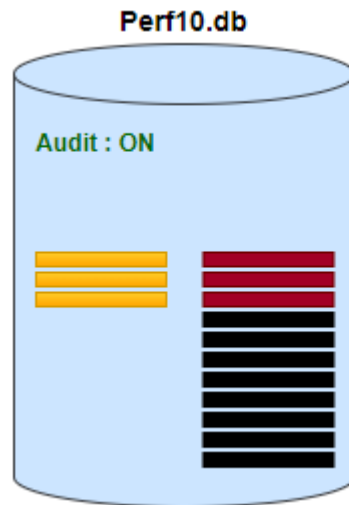
/ Create a temporary OpenEdge 10 database



// Create a new database

/ Create a temporary OpenEdge 10 database

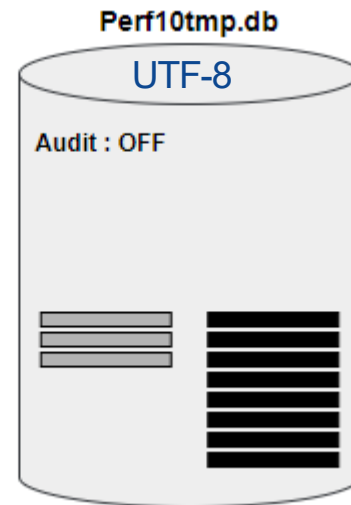
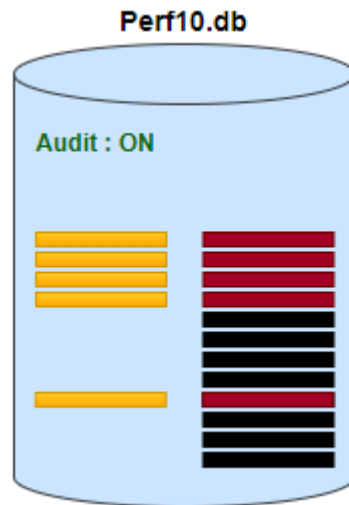
- Remove all audit records in existing database
 - Remove CUD happened before backup in auditing table



// Create a new database

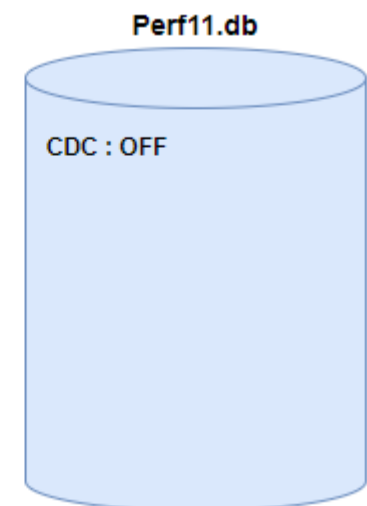
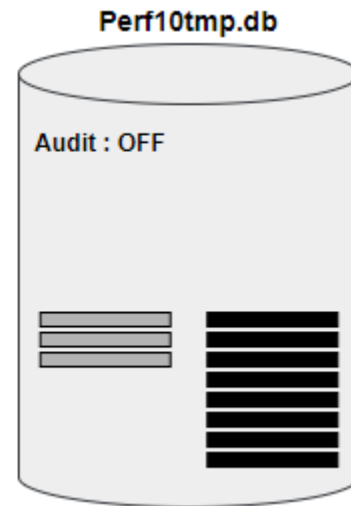
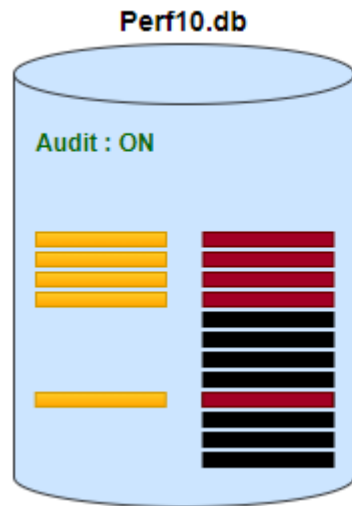
/ Create a temporary OpenEdge 10 database

- Other Proginov operation (convert to UTF-8,...)
 - Need to disable and remove all Auditing stuff



// Create a new database

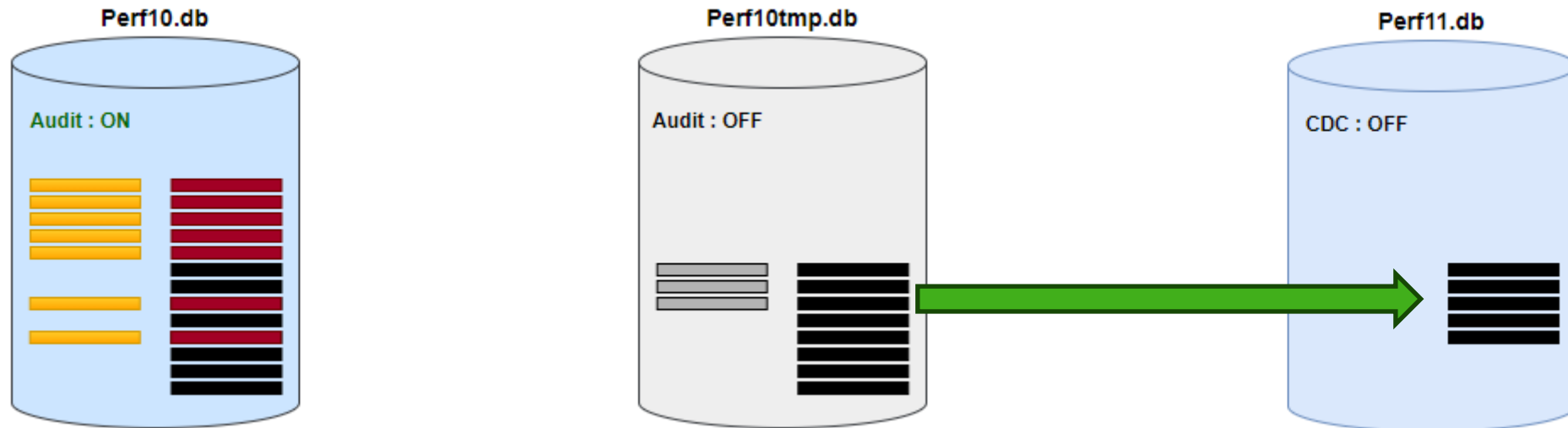
/ Create an empty OpenEdge 11 Database



// Create a new database

/ Create an empty OpenEdge 11 Database

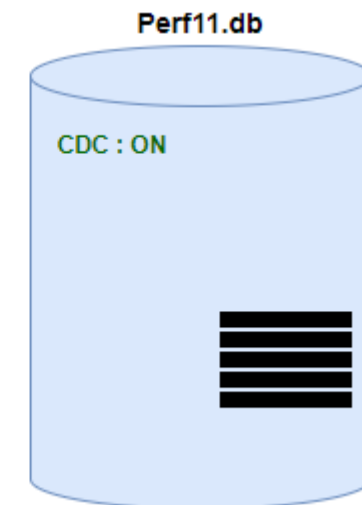
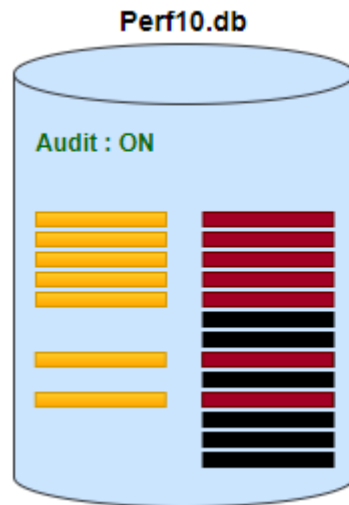
- Import recent data (~6 month) for each table (buffer copy or binary dump-load)



// Create a new database

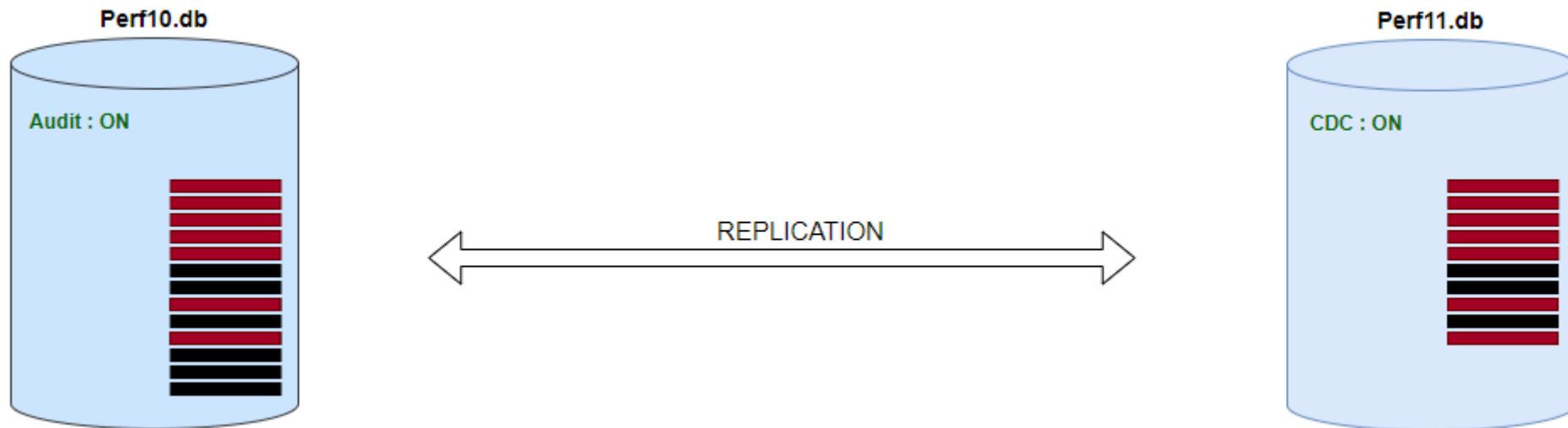
/ Create an empty OpenEdge 11 Database

- o Enable CDC



// Create a new database

/ Start real time replication !



And then... deal with reality

// 0 changes, really ?

✓ Implement with zero changes to the application, just configure and run

/ Not that easy ⚠

/ CDC or Auditing create new records for every CUD

Without Auditing

Create/Update/Delete **Item**

With Auditing

Create/Update/Delete **Item**

- Create **_aud-Audit-Data**

- Create **_aud-Audit-Data-Value ***

Without CDC

Create/Update/Delete **Item**

With CDC

Create/Update/Delete **Item**

- Create **_Cdc-Change-Tracking**

- Create **Cdc-Item ***

/ 1 change = 4 records

/ Same problems with large transactions



// Long Transaction and BI

/ In case of a transaction that lasts a long time

- All Create Update Delete will be added to the BI
- ... And all the Auditing / CDC changes !

/ The more processes there are, the more BI is likely to increase

/ Replication processes come into play !

/ Transaction must be as quick as possible

// Lock Table Entries

/ Pay attention to -L with massive deletions

- Delete 13970 records of OrderLine

Case	Most Locks	BI (Mb)	AI (Mb)
OE10	1	8.8	8.6
OE10 Audit	125 733	61.9	61.6
OE11	1	8.8	8.6
OE11 CDC	27 944	21.1	20.6

/ Before, only 1 lock at a time

/ Now, each deletion add 2 locks until the end of transaction

/ Multiply « -L » by 40 instead of 6

- 20K -> 120 K -> 800K

/ Do smaller Transaction (200 per 200 for example) !



```
DO TRANSACTION:  
  FOR EACH Item EXCLUSIVE-LOCK:  
    DELETE Item.  
  END.  
END.
```

Progress Documentation > Startup Command and Parameter Reference > ... >

Lock Table Entries (-L)

// Index Entry Locks (IEL) – The Phantom Keys

/ Replication process deletes many record in auditing and CDC tables

/ Every value of Unique Index are kept by Progress (as Phantom Key)

- Used in case of transaction rollback, to ensure we can rollback

/ Those IEL will have big impact on performance

machine:	<input type="text" value="dblrxprowin01"/>	Base:	<input type="text" value="profpnv12"/>	Proenv:	<input type="text" value="proenv1176"/>	
Directory:	<input type="text" value="/asp/bases/poolprod/cptbdd/dboe11/"/>					<input type="button" value="idxcompact"/>
Index:	<input type="text" value="_Cdc-Change-Tracking._Sequence-Id"/>					<input type="button" value="perf09"/>
read IDX:	<input type="text" value="1,287,988"/>	read Table:	<input type="text" value="0"/>	Qry-rec:	<input type="text" value="356"/>	

/ Use IdxCompact and other tricks (see last year conference)

// Reduce the number of changes

/ Use variables !

increase prices by 10%

```
/* Before : 1 Update per OrderLine */
DO TRANSACTION:
    FIND FIRST Order WHERE Order.OrderNum = 1 EXCLUSIVE-LOCK NO-ERROR.
    ASSIGN Order.TotalAmount = 0.
END.

FOR EACH OrderLine WHERE OrderLine.OrderNum = Order.OrderNum EXCLUSIVE-LOCK:
    ASSIGN
        OrderLine.Price = OrderLine.Price * 1.1
        Order.TotalAmount = Order.TotalAmount + OrderLine.Price * OrderLine.Qty.
END.

/* After : 1 Update */
DEFINE VARIABLE vTotalAmount AS DECIMAL NO-UNDO.

DO TRANSACTION:
    FIND-FIRST Order WHERE Order.OrderNum = 1 EXCLUSIVE-LOCK NO-ERROR.
END.

FOR EACH OrderLine WHERE OrderLine.OrderNum = Order.OrderNum EXCLUSIVE-LOCK:
    ASSIGN
        OrderLine.Price = OrderLine.Price * 1.1
        vTotalAmount = vTotalAmount + OrderLine.Price * OrderLine.Qty.
END.

DO TRANSACTION:
    ASSIGN Order.TotalAmount = vTotalAmount.
    RELEASE Order.
END.
```

// Reduce the number of changes

/ Use temp-table...

```
/* Before : 1 Update per order */
FOR EACH Order NO-LOCK:
    FIND FIRST Customer WHERE Customer.CustNum = Order.CustNum EXCLUSIVE-LOCK NO-ERROR.
    ASSIGN Customer.TotalOrdersAmount = Customer.TotalOrdersAmount + Order.TotalAmount.
    RELEASE Customer.
END.

/* After : 1 Update per customer */
DEFINE TEMP-TABLE ttCustomerOrdersAmount NO-UNDO
    FIELD CustNum AS INTEGER
    FIELD TotalOrdersAmount AS DECIMAL
    INDEX CustNum IS PRIMARY CustNum.

FOR EACH Order NO-LOCK:
    FIND FIRST ttCustomerOrdersAmount WHERE ttCustomerOrdersAmount.CustNum = Order.CustNum EXCLUSIVE-LOCK NO-ERROR.
    IF NOT AVAILABLE ttCustomerOrdersAmount THEN
        DO:
            CREATE ttCustomerOrdersAmount.
            ASSIGN ttCustomerOrdersAmount.CustNum = Order.Ordernum.
        END.
    ttCustomerOrdersAmount.TotalOrdersAmount = ttCustomerOrdersAmount.TotalOrdersAmount + Order.TotalAmount.
END.

FOR EACH ttCustomerOrdersAmount NO-LOCK:
    FIND FIRST Customer WHERE Customer.CustNum = ttCustomerOrdersAmount.CustNum EXCLUSIVE-LOCK NO-ERROR.
    ASSIGN Customer.TotalOrdersAmount = ttCustomerOrdersAmount.TotalOrdersAmount.
    RELEASE Customer.
END.
```

// Reduce the number of changes

/ Use temp-table... and transactional packet

- Better for BI, AI and performance

```
/* After : 1 Update per customer */
DEFINE TEMP-TABLE ttCustomerOrdersAmount NO-UNDO
  FIELD CustNum AS INTEGER
  FIELD TotalOrdersAmount AS DECIMAL
  INDEX CustNum IS PRIMARY CustNum.

FOR EACH Order NO-LOCK:
  FIND FIRST ttCustomerOrdersAmount WHERE ttCustomerOrdersAmount.CustNum = Order.CustNum EXCLUSIVE-LOCK NO-ERROR.
  IF NOT AVAILABLE ttCustomerOrdersAmount THEN
  DO:
    CREATE ttCustomerOrdersAmount.
    ASSIGN ttCustomerOrdersAmount.CustNum = Order.Ordernum.
  END.
  ttCustomerOrdersAmount.TotalOrdersAmount = ttCustomerOrdersAmount.TotalOrdersAmount + Order.TotalAmount.
END.

ii = 500.
DO WHILE ii >= 500 TRANSACTION:
  FOR EACH ttCustomerOrdersAmount NO-LOCK ii = 1 TO 500:
    FOR EACH Customer WHERE Customer.CustNum = ttCustomerOrdersAmount.CustNum EXCLUSIVE-LOCK:
      ASSIGN Customer.TotalOrdersAmount = ttCustomerOrdersAmount.TotalOrdersAmount.
      DELETE ttCustomerOrdersAmount.
    END.
  END.
END.
```

// Reduce the number of changes

/ Use **BUFFER-COPY** with **ASSIGN**

- ASSIGN then BUFFER-COPY = 1 create and 1 update
- BUFFER-COPY with ASSIGN = 1 create



```
/* Before : 1 creation & 1 update per TItem */
FOR EACH TItem NO-LOCK:
  CREATE Item.
  ASSIGN Item.Itemnum = GetNextNum(). /* CREATE*/
  BUFFER-COPY TItem EXCEPT Itemnum TO Item. /* UPDATE */
  RELEASE Item.
END.

/* After : 1 creation per TItem */
FOR EACH TItem NO-LOCK:
  RELEASE Item.
  /* buffer-copy will do the create if Item is not available */
  BUFFER-COPY TItem TO Item ASSIGN Item.Itemnum = GetNextNum().
END.
RELEASE Item.
```

// Settings Audit / CDC

/ After 3 months analysis watching changes on identifying fields

/ If identifying fields are immutable, change CDC level from Maximum to Medium

- The before value in records will not be created
- It might save time !

Table 1. Description of CDC levels

Level	Description
Minimal (0)	This level indicates a change occurred. No record values will be recorded.
Minimal with Fieldmap (1)	Similar to the Minimal (0) level, this level indicates a change occurred, but also includes a field map value indicating which fields changed. No record values will be recorded.
Medium (2)	This level records the current (after) value of all CUD operations.
Maximum (3)	This level records both the previous (before) and current (after) values of all CUD operations.

Conclusion



// Can it fit to your need ?

/ No high frequency update on same records on both sides

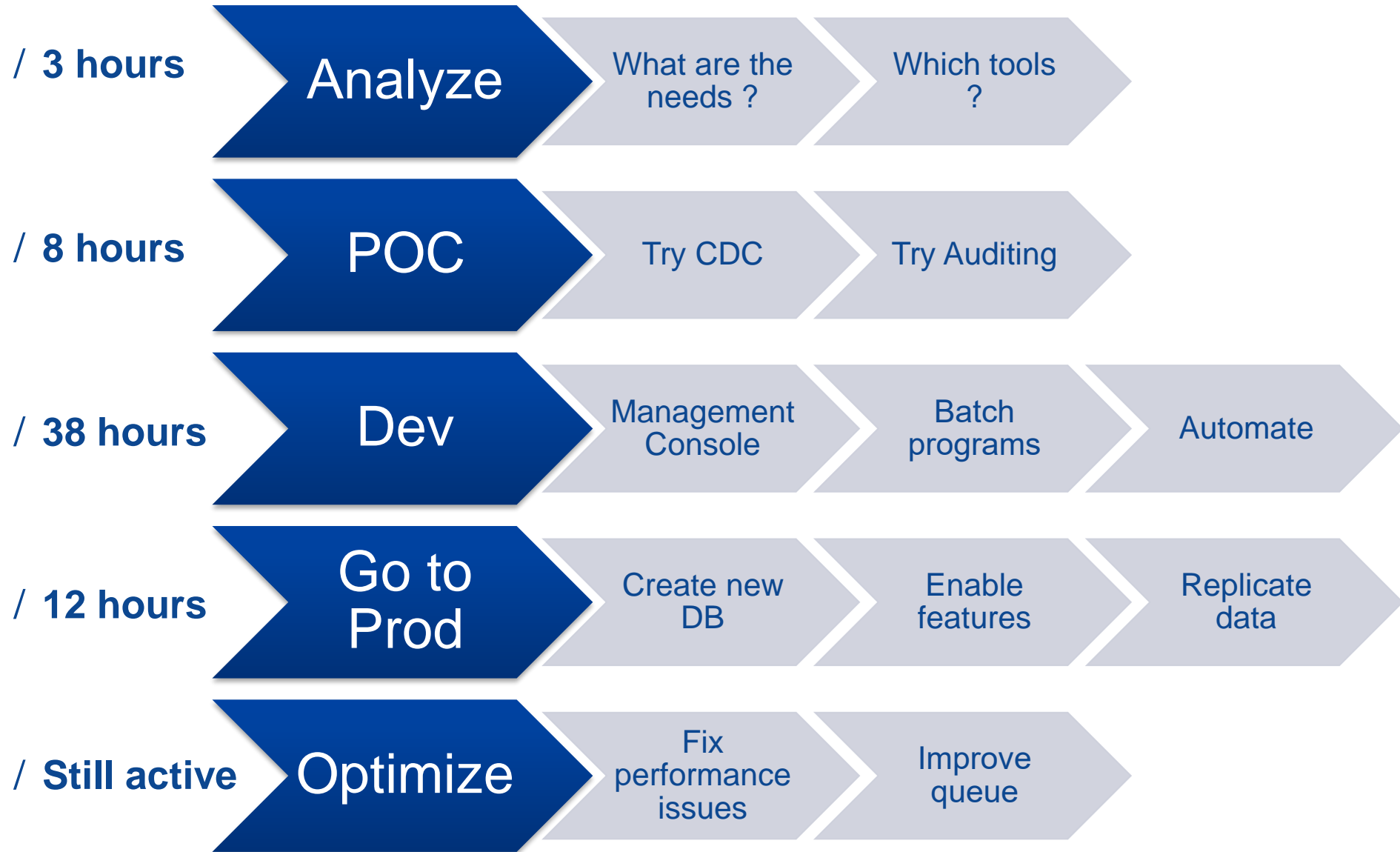
- Good for analysis or metrics

/ No critical data, still a custom tool

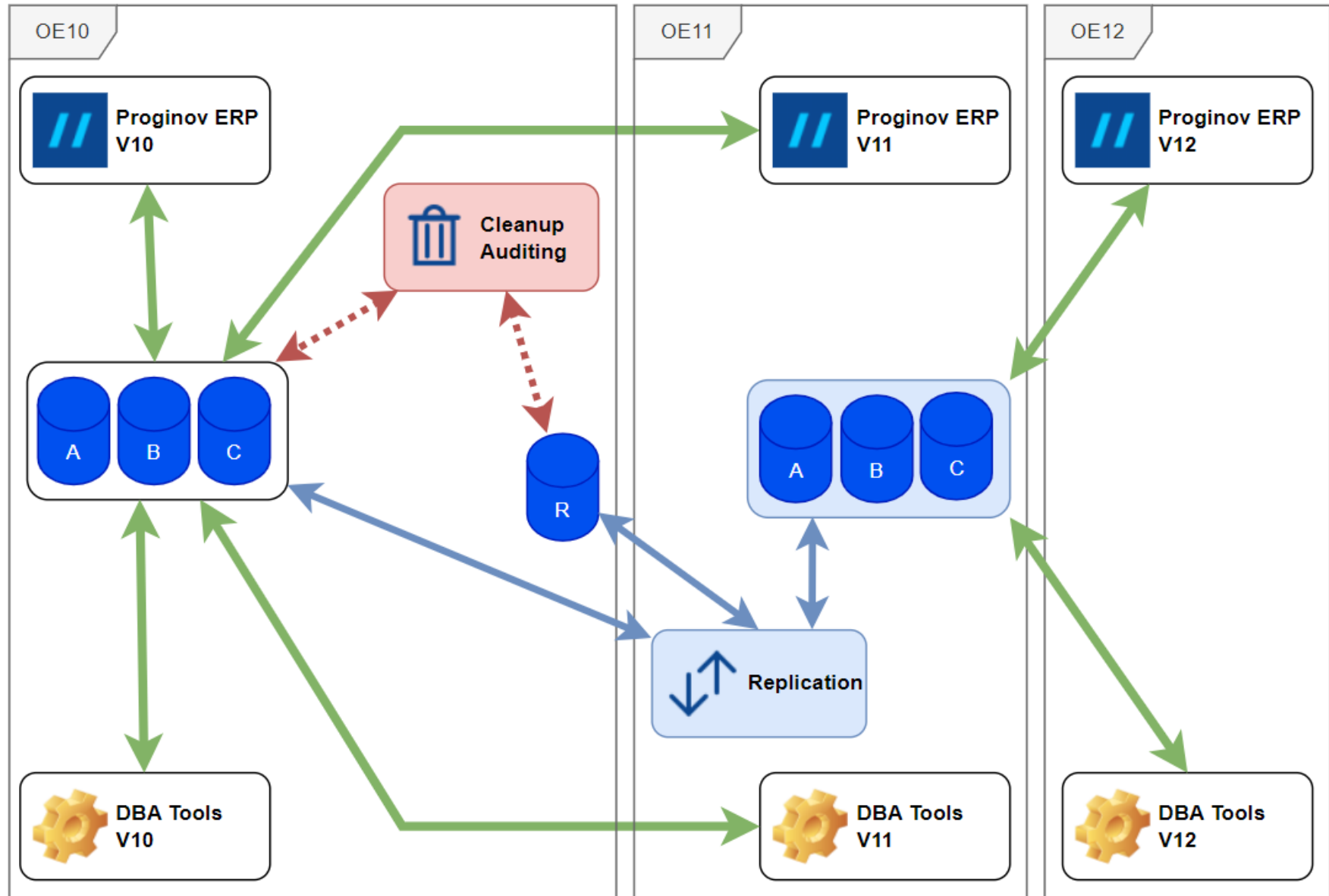
/ For us, average of 16 000 records / minute (2 months average)

- 50 replication batch sessions
- 25 cleanup processes
- Could be faster without network constraints

// How much time ?



// Result



Any questions ?





+33 (0)2 51 70 93 93
www.proginov.com

