Bidirectional Replication

(Part II)

by Ahmed Baraka

Introduction to Oracle Data Guard

In this lecture, we are going to talk about the basic concepts of Oracle Data Guard

Objectives

By the end of this lecture, you should be able to:

- Configure Conflicts Detection and Resolution (CDR) rules

Bidirectional Configuration Challenges

- Application Design
- Sequence numbers and identity data types
- Truncate table operations
- Triggers and Cascaded Deletes
- Loop detection
- Conflict prevention, detection, and resolution

Possible Conflicts

Operation	Possible Conflict
INSERT	Record already in the target
UPDATE	Record doesn't exist in target
	Record exists but different
DELETE	Record doesn't exist in target
	Record exists but different

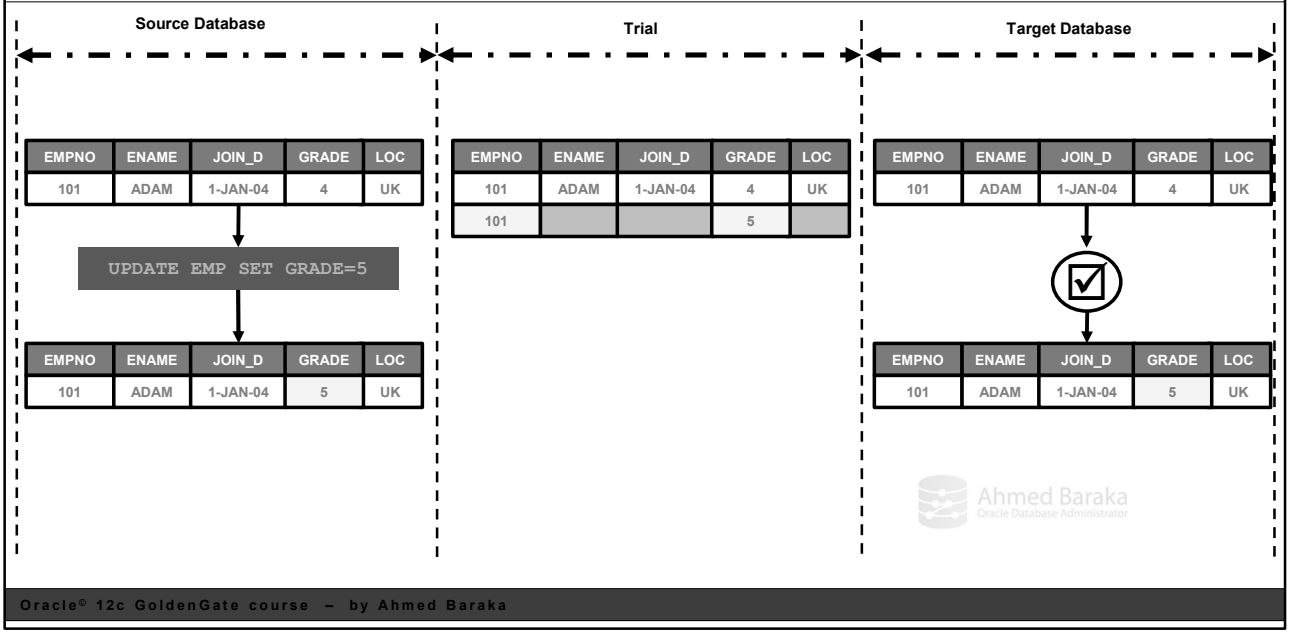
About Handling Conflicts in Oracle GoldenGate

- Conflict is inevitable in most cases because of the asynchronous nature of the GoldenGate synchronization mechanism
- It can be minimized:
  - Low network latency
  - Apply data policy on your system
- It should always be monitored
- Handled by Conflict Detection and Resolution (CDR) feature

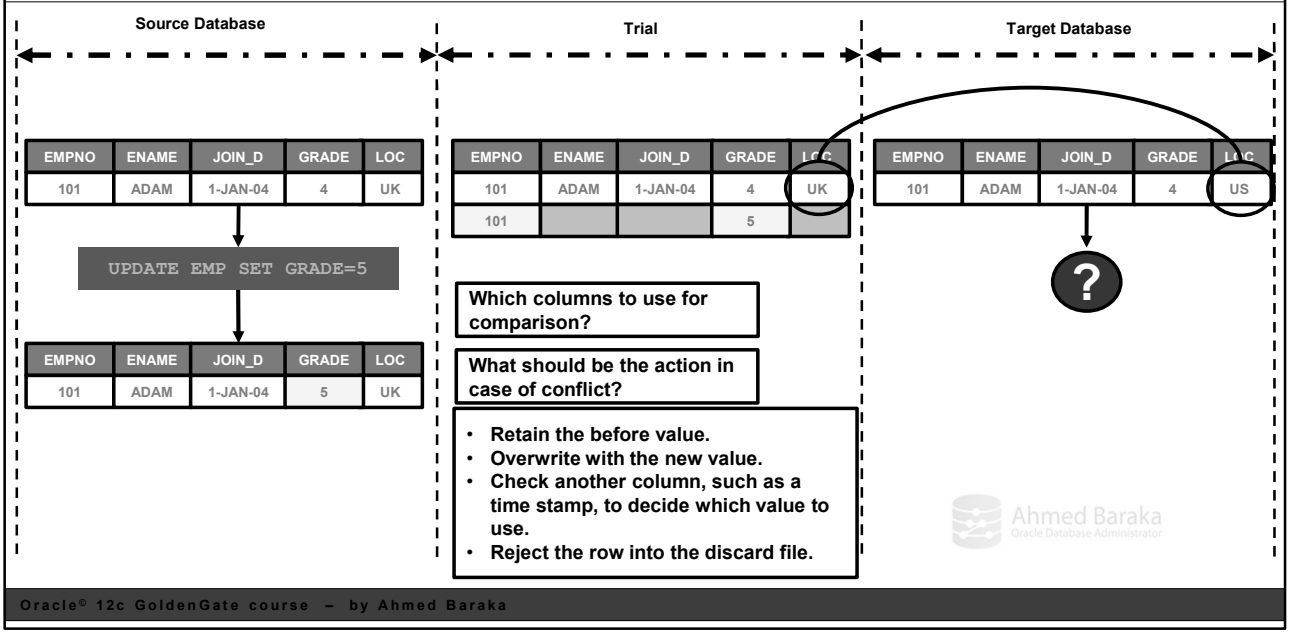
Conflicts Detection

Operation	Possible Conflict	Detection Method
INSERT	Record already in the target	Unique values
UPDATE	Record doesn't exist in target	No data found
	Record exists but different	Compare Columns (before image is required)
DELETE	Record doesn't exist in target	No data found
	Record exists but different	Compare Columns

Conflict Detection



Conflict Detection



Configuring Oracle GoldenGate CDR

- Make the required column values available to Extract
- Configure the parameter files for conflict resolution
- Configure the parameter files for error handling

Make the required column values available to Extract

- Use the ADD TRANDATA command to instruct the database to include the before image in its supplemental logging data

```
ADD TRANDATA [container.]owner.table
 [ALLCOLS] [, COLS (columns)]
```

```
ADD TRANDATA sales.orders
ADD TRANDATA sales.orders ALLCOLS
ADD TRANDATA sales.orders, COLS (quantity, desc)
```

Configuring Oracle GoldenGate CDR

- Make the required column values available to Extract
- Configure the parameter files for conflict resolution
- Configure the parameter files for error handling

Conflict Resolution Parameters

Parameter/Option	Process	Description
LogAllSupCOLS (12c)	E	For Extract, capture the BEFORE image For Replicat, include the BEFORE image in its operation.
GetUpdateBefores	E and R	
TABLE .. **GetBeforeCols**	E	To specify which columns to be included in the BEFORE image data (not needed when LogAllSupCOLS is used)
MAP… **CompareCols**	R	Which columns to use for update or delete conflict detect.
MAP… **ResolveConflict**		To specify how to resolve detected conflicts

About the `GetBeforeCols` Option

- Specify which columns to capture by the Extract (11g)
- Not needed when `LogAllSupCOLS` is used

```
TABLE .. GETBEFORECOLS(
 {ON UPDATE | ON DELETE}
 {ALL | KEY | KEYINCLUDING (col[,...]) | KEYANDMOD |
  | ALLEXCLUDING (col[,...]) } [,...] )
```

```
TABLE orders, GETBEFORECOLS (
 ON UPDATE KEYINCLUDING (quantity, desc),
 ON DELETE KEYINCLUDING (quantity, desc));
```

About the `CompareCols` Option

- Specify the columns to detect conflicts
- Only scalar data types are supported


```
COMPARECOLS( {ON UPDATE | ON DELETE}
 {ALL | KEY | KEYINCLUDING (col[,...]) | ALLEXCLUDING
  (col[,...]) } [,...] )
```

`CompareCols` Option Examples


```
MAP orders, TARGET tgt.orders
COMPARECOLS (
 ON UPDATE KEYINCLUDING (quantity, desc),
 ON DELETE KEYINCLUDING (quantity, desc));
```

```
MAP src, TARGET tgt
COMPARECOLS (ON UPDATE ALLEXCLUDING (comment));
```

About the `ResolveConflict` Option

- To specify how Replicat handles conflicts


```
MAP.. RESOLVECONFLICT (
 {INSERTROWEXISTS | UPDATEROWEXISTS | UPDATEROWMISSING |
 DELETEROWEXISTS | DELETEROWMISSING}
 ( {DEFAULT | resolution_name},
   {USEMAX (resolution_column) |
    USEMAXEQ (resolution_column)  |
    USEMIN (resolution_column) |
    USEMINEQ (resolution_column) |
    USEDELTA | DISCARD | OVERWRITE | IGNORE}
 ) [, COLS (column[,...])])
```

Conflict Resolution Methods

Conflict-handler Logic	Description
USEMAX/MIN(EQ)	If the value in the trail record is greater/less than (or equal to) the value of the current record
USEDELTA	Update current record with the value of the difference between the before and after image values
DISCARD	Write the trail record in the discard file
OVERWRITE	Apply the trail record
IGNORE	Do not apply to the current record, nor to the discard file

Conflict Resolution Methods

Operation	Conflict Type	Resolution Method	Effect
INSERT	InsertRowExists	Overwrite	Change INSERT into UPDATE
		Ignore	Ignore the trail record and retain the current record
		Discard	Write trail record in discard file and retain the current record
		UseMin, UseMax	Change INSERT into UPDATE
UPDATE	UpdateRowMissing	Overwrite	Change the missing UPDATE to an INSERT
		Ignore	Ignore the trail record and retain the current record
		Discard	Write trail record in discard file and retain the current record
	UpdateRowExists	Overwrite	Apply the update from the trail record
		Ignore	Ignore the trail record and retain the current record
		Discard	Write trail record in discard file and retain the current record
		UseMin, UseMax	Apply the trail record as an update
		UseDelta	Update with the difference between the before and after values
DELETE	DeleteRowMissing	Ignore	Ignore the trail record and retain the current record
		Discard	Write trail record in discard file and retain the current record
	DeleteRowExists	Overwrite	Apply the delete from the trail record
		Ignore	Ignore the trail record and retain the current record
		Discard	Write trail record in discard file and retain the current record

CDR Example 1

```
MAP sales.src, TARGET sales.tgt,
COMPARECOLS (ON UPDATE ALL, ON DELETE ALL),
RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX (last_mod_time))),
RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(last_mod_time))),
RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)),
RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)),
RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
```

CDR Example 2

```
MAP sales.src, TARGET sales.tgt,
COMPARECOLS (ON UPDATE KEYINCLUDING (phone, salary, last_mod_time),
             ON DELETE KEYINCLUDING (phone, salary, last_mod_time)),
RESOLVECONFLICT ( UPDATEROWEXISTS,
                  (delta_res_method, USEDELTA, COLS (salary)),
                  (DEFAULT, USEMAX (last_mod_time))
                );
```

CDR Example 3

```
MAP sales.src, TARGET sales.tgt,
COMPARECOLS (ON UPDATE ALLEXCLUDING (comment)),
RESOLVECONFLICT (
 UPDATEROWEXISTS,
 (delta_res_method, USEDELTA, COLS (salary, balance)),
 (max_res_method, USEMAX (last_mod_time), COLS (address, last_mod_time)),
 (DEFAULT, IGNORE));
```

Summary

In this lecture, you should have learnt the following:

- Configure Conflicts Detection and Resolution (CDR) rules