

# CORAL Database Copy Tools

Gitika Khare  
Computer Centre  
RRCAT  
Indore, India

# Scope

- ◆ Several experiment applications are based on LCG database abstraction layer CORAL for accessing relational data from several supported back-ends (*Oracle, MySql, SQLite*).
- ◆ To simplify deployment of CORAL based applications, a set of associated tools are required to allow copying of individual tables or complete schemas between existing databases and technologies.

# CoralTools

- ◆ The CoralTools package provides a set of export tools for the CORAL framework.
- ◆ The mapping between data types of source and destination schema follows the CORAL mapping rules.
- ◆ This would allow the CORAL based applications to run unchanged against data copies in a different database backend.
- ◆ The tools support schema and data copy between the relational databases – Oracle, MySQL and SQLite.
- ◆ The CoralTools package is developed in Python and is an implementation of the PyCoral interface which was developed using the python/C API.

# schemautils Python Package

The following set of CoralTools have been developed for the CORAL framework:

- ◆ `listobjects(schema)`
- ◆ `dumpobjectlist(schema)`
- ◆ `copyschema()`
- ◆ `copydata()`
- ◆ `copytableschema(tablename)`
- ◆ `copytabledata(tablename, selectionclause, selectionparameters)`
- ◆ `copytablelistschema(tablelist)`
- ◆ `copytablelistdata(tablelist)`

# Implementation Details

## Resolution of Table Hierarchies

- ◆ Export of a tree of hierarchically related objects from source to specified database required the knowledge of table dependency trees.
- ◆ A Python class is developed for providing schema object relationship information.
- ◆ The relational dependency class provides an implementation for building table dependency trees for a specific table and an entire schema. It scans all the tables in the schema to construct the dependencies between the tables.
- ◆ Circular dependency between the tables is not handled since it requires special treatment.
- ◆ This class is used for the implementation of CoralTools.

# Implementation Details

## Scalability

- ◆ The tools are targeted to copy of big volumes of data.
- ◆ Implementation uses bulk operations on both reading and writing side, taking care to balance the data transfer from the source to the destination (minimize the roundtrips), and the memory required on the client side for caching.
- ◆ For copying of data buffer protocol is used, which allows reading and writing of data into a buffer without additional memory requirement.

# UseCase 1

1. Export of all the objects from a specified schema from a source database into a specified database. Source schema objects in destination database do not exist.
  - 1.1 Schema only, no data
  - 1.2 Schema + data
  
2. Export of a tree of hierarchically related objects from a source database into a specified database. For a given specified table, the export considers:
  - a) all the upper level tables which are referenced by foreign key constraints.
  - b) all of dependent tables (referencing columns of this table by foreign key constraint).

# UseCase 2

User specifies only a single table; the rest of tables are automatically resolved, based on a) and b).

The export also considers

c. Data selection on the specified table columns.

2.1 Tables and related objects (constraints, indexes) do not exist in the destination database. The required tables and related objects are consistently created.

2.2 Tables and related objects exist, possibly with data. The existing set of tables involved is checked. The export is successful if no primary key violations are found during the new data insertion.



# Status

- ◆ The CoralTools have been developed and tested using the test cases with Oracle, MySQL and SQLite.
- ◆ The tools for Use Case 1 have been tested with a case with POOL-ORA application with Oracle, SQLite and MySQL database.

## Remaining Jobs

- ◆ Validation with acceptance test for CoralTools.
  - case with COOL application
  - case with POOL-ORA application
- ◆ Testing
  - Scalability test
  - Performance analysis
  - Optimization
- ◆ Documentation compliant with CORAL doc framework.