CERN

Catalogue schema verification and migration tool

What is done and discussions to move forward

Cedric Caffy

The catalogue schema verification tool

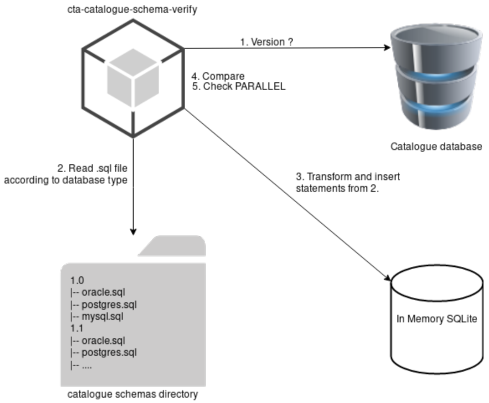
The catalogue schema migration tool

The catalogue schema verification tool

- What it does
  - Checks INDEX names
  - Checks TABLE names
  - Checks COLUMN names and types
  - Checks CONSTRAINT names
    $\Rightarrow$ NOT for MySQL
    $\Rightarrow$ NOT NOT NULL constraints for PostgreSQL
  - Display WARNING for Oracle PARALLEL TABLES
- What it does not
  - Does not check for triggers $\Rightarrow$ no triggers should be implemented in the Catalogue

The catalogue schema verification tool

- How does it work ?

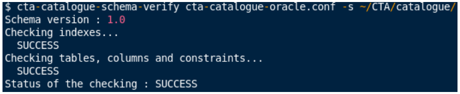


cta-catalogue-schema-verify
1. Version ?
4. Compare
5. Check PARALLEL
Catalogue database
2. Read .sql file according to database type
3. Transform and insert statements from 2.
In Memory SQLite
catalogue schemas directory

```
1.0
|-- oracle.sql
|-- postgres.sql
|-- mysql.sql
1.1
|-- oracle.sql
|-- postgres.sql
|-- ....
```

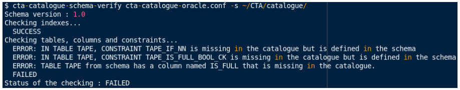
The catalogue schema verification tool

- The output



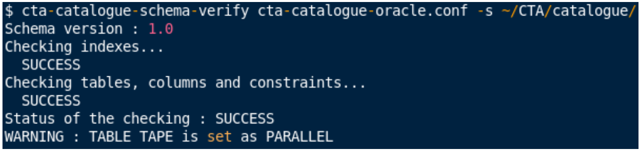
```
$ cta-catalogue-schema-verify cta-catalogue-oracle.conf -s ~/CTA/catalogue/
Schema version : 1.0
Checking indexes...
  SUCCESS
Checking tables, columns and constraints...
  SUCCESS
Status of the checking : SUCCESS
```

Schema verification SUCCESS



```
$ cta-catalogue-schema-verify cta-catalogue-oracle.conf -s ~/CTA/catalogue/
Schema version : 1.0
Checking indexes...
  SUCCESS
Checking tables, columns and constraints...
  ERROR: IN TABLE TAPE, CONSTRAINT TAPE_IF_NN is missing in the catalogue but is defined in the schema
  ERROR: IN TABLE TAPE, CONSTRAINT TAPE_IS_FULL_BOOL_CK is missing in the catalogue but is defined in the schema
  ERROR: TABLE TAPE from schema has a column named IS_FULL that is missing in the catalogue.
  FAILED
Status of the checking : FAILED
```

Missing IS_FULL column in table TAPE

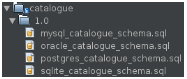


```
$ cta-catalogue-schema-verify cta-catalogue-oracle.conf -s ~/CTA/catalogue/
Schema version : 1.0
Checking indexes...
  SUCCESS
Checking tables, columns and constraints...
  SUCCESS
Status of the checking : SUCCESS
WARNING : TABLE TAPE is set as PARALLEL
```

Table TAPE has been set as PARALLEL

The catalogue schema verification tool

- What needs to be discussed
  - Where do we put the folder containing all the versions of the schema ?
    - Create a RPM of the cta-catalogue-schema-verify tool and stick the folder with it.
      $\Rightarrow$ This folder will be in the same place as the executable
    - Other ideas ?



```
catalogue
  1.0
    mysql_catalogue_schema.sql
    oracle_catalogue_schema.sql
    postgres_catalogue_schema.sql
    sqlite_catalogue_schema.sql
```

The database schema migration tool

- Problem to solve
  - Schema modifications $\Rightarrow$ new schema version !
    - Schema version format : MAJOR.MINOR
    - MAJOR changes ONLY if schema modifications are not backward compatible with previous schema version
  - How do we do catalogue schema migration from one version to the next one ?

The catalogue schema migration tool

- CASTOR's way of schema migration : PL/SQL file !
  - 1. Check database schema version
  - 2. Update tables, procedures, etc... (COMMIT after each update)
  - 3. Recompile all procedures, triggers...
    - If all is successful, update a table UpgradeLog with status COMPLETE
    - Else ROLLBACK current update and update UpgradeLog by increasing a failure counter, list the failed-to-compile objects

The catalogue schema migration tool

- What would change between one schema version and the next one ?
  - Creation/deletion/modification of tables, columns, constraints, sequences
  - Data changes within tables
  - Creation/deletion/modification of PL/SQL procedures

The catalogue schema migration tool

- The migration of the schema should
  - be idempotent
  - do verification before each update (e.g: does a column exists before updating its type ?)
  - be a step by step migration (one modification at a time)
  - be rollbackable
  - be traceable
  - be easy to execute
- The migration tool should be the same during the whole life of CTA

The catalogue schema migration tool

- What kind of tool should we use to migrate the catalogue schema from one version to the next one ?
  - Four ways
    - Use a framework/language-dependant library
    - Use an independent Database-Migration-Focused software
    - Develop a tool ourselves
    - Use simple PL/SQL scripts because after all, modifications will be little ones

The catalogue schema migration tool

- Use a framework/language-dependent library
  - Python (alembic), Ruby (Active Record)
  - No built-in support for Oracle databases ⇒ We would need to use an ORM (Object-relational mapping) tool and plug a migration framework to it.

The catalogue schema migration tool

- Use an independent Database-Migration-Focused software
  - Liquibase ▸ Link
  - Flyway ▸ Link

The catalogue schema migration tool

- Liquibase
  - Open-source command-line tool based on Java. Apache v2 License
  - Compatible with our 4 database types
  - Easy-to-use : *liquibase –changeLogFile=migration1.0To1.1.sql update*
  - changeLogFile = sql file for doing the migration + Liquibase-related metadata
  - Rollback changes
  - Adds two Liquibase-related tables to the database

The catalogue schema migration tool

- Flyway
    - Command-line tool based on Java. Apache v2 License
    - Compatible with our 4 database types
    - Easy-to-use : *flyway migrate*
    - Migration scripts have to be put in a specific folder. (Naming convention).
    - Adds one history table to the database : flyway_schema_history
    - A lot of functionalities only in PRO versions :-(

The catalogue schema migration tool

- Use simple PL/SQL scripts because after all, modifications will be little ones
    - What do you think ?

The catalogue schema migration tool

- Conclusion

Migration strategy	Pros	Cons
Framework/language-dependent lib	Use of exisiting lib to do migrations	No built-in support for Oracle
Database-Migration software	Easy migrations, well defined migration organization, Rollback	Timeline for our 4 databases support ? Changing technology might be difficult
Develop a tool ourselves	We depend on nobody except us	Time consuming. Safe ?
PL/SQL Scripts	Very easy to execute	Migration failure management

CERN