Catalogue schema migration tool

PRESENTED BY CEDRIC CAFFY

Content

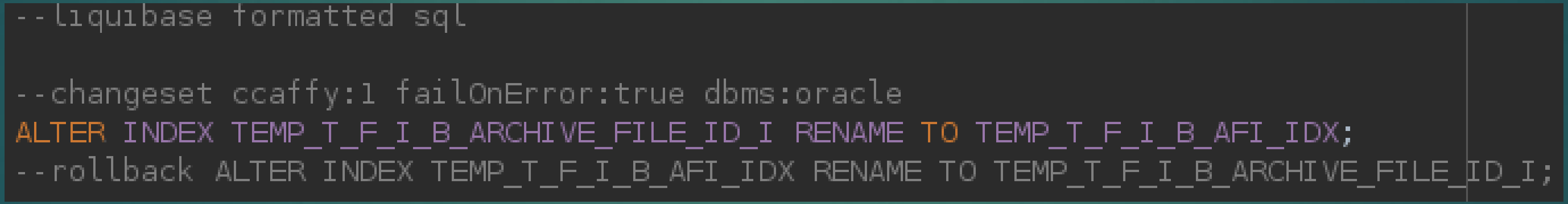
1. Liquibase
   1. How it works
      1. Changelog file
      2. Configuration file
      3. Tracking tables
   2. Run the tool
      1. Command line format
   3. Migration 1.0 to 1.1
      1. Structure of the migration folder
      2. Changelog file
      3. Update schema
      4. Rollback
2. Migration procedures
   1. Backward-compatible modifications
   2. Backward-incompatible modifications
   3. Migration on different database types
3. GO / NO GO

1. Liquibase

1.1 How it works

1.1.1 Changelog file

- Text file that contains SQL Statements + Liquibase-related metadata



```
--liquibase formatted sql

--changeset ccaffy:1 failOnError:true dbms:oracle
ALTER INDEX TEMP_T_F_I_B_ARCHIVE_FILE_ID_I RENAME TO TEMP_T_F_I_B_AFI_IDX;
--rollback ALTER INDEX TEMP_T_F_I_B_AFI_IDX RENAME TO TEMP_T_F_I_B_ARCHIVE_FILE_ID_I;
```

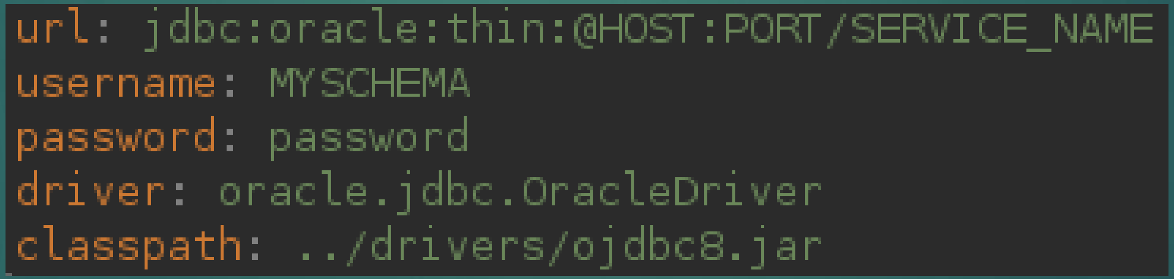
- ChangeSet
  - Set of changes that Liquibase executes within **one transaction**
  - Format : `--changeset userName:id_changeSet [attributes]`
  - Advice : Do not define more than one logical change per changeset

1. Liquibase

1.1 How it works

1.1.2 Configuration file

- Database-related .properties file



```
url: jdbc:oracle:thin:@HOST:PORT/SERVICE_NAME
username: MYSCHEMA
password: password
driver: oracle.jdbc.OracleDriver
classpath: ../drivers/ojdbc8.jar
```

1. Liquibase

1.1 How it works

1.1.3 Tracking tables

- DATABASECHANGELOG
  - Stores information about the *modifications* applied to the database
- DATABASECHANGELOGLOCK
  - Ensure only one instance of Liquibase is running at one time

1. Liquibase

1.2 Run the tool

1.2.1 Command line format

```
liquibase --defaultsFile=path_to_properties --changeLogFile=path_to_changelog_file <command>
```

- *<command>:*
  - *updateSQL*
  - *update*
  - *rollbackCountSQL*
  - *rollbackCount*

1. Liquibase

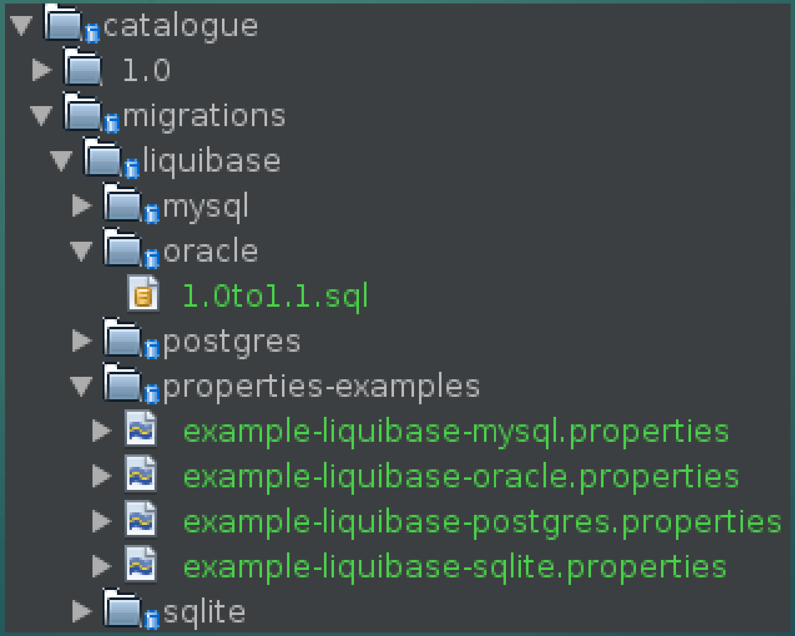
1.3 Migration 1.0 to 1.1

- 1 INDEX renaming
- 2 CONSTRAINTS renaming
- Creation of 1 UNIQUE CONSTRAINT
- Update SCHEMA_VERSION_MINOR

1. Liquibase

1.3 Migration 1.0 to 1.1

1.3.1 Structure of the migration folder

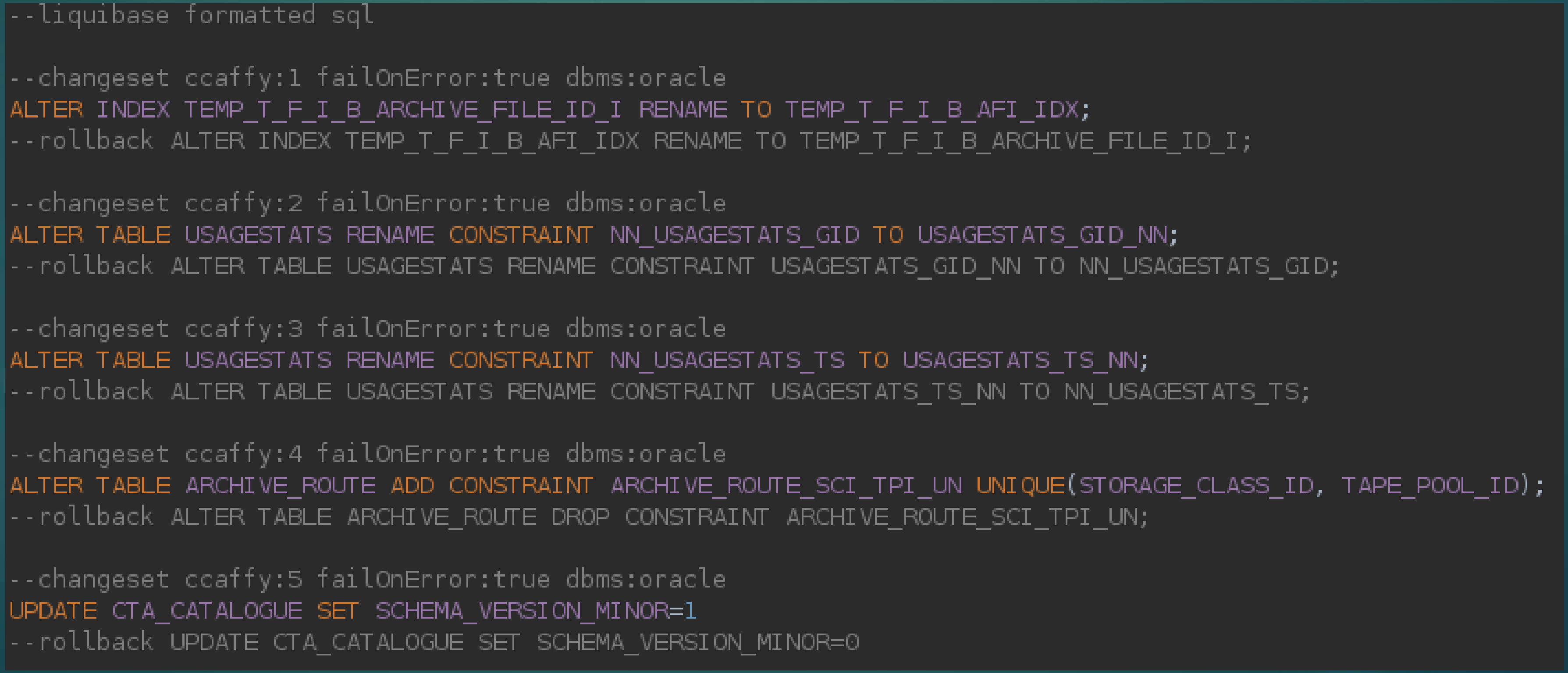


```
▼ catalogue
  ▶ 1.0
  ▼ migrations
    ▼ liquibase
      ▶ mysql
      ▼ oracle
          1.0to1.1.sql
      ▶ postgres
      ▼ properties-examples
        ▶ example-liquibase-mysql.properties
        ▶ example-liquibase-oracle.properties
        ▶ example-liquibase-postgres.properties
        ▶ example-liquibase-sqlite.properties
      ▶ sqlite
```

1. Liquibase

1.3 Migration 1.0 to 1.1

1.3.2 Changelog file

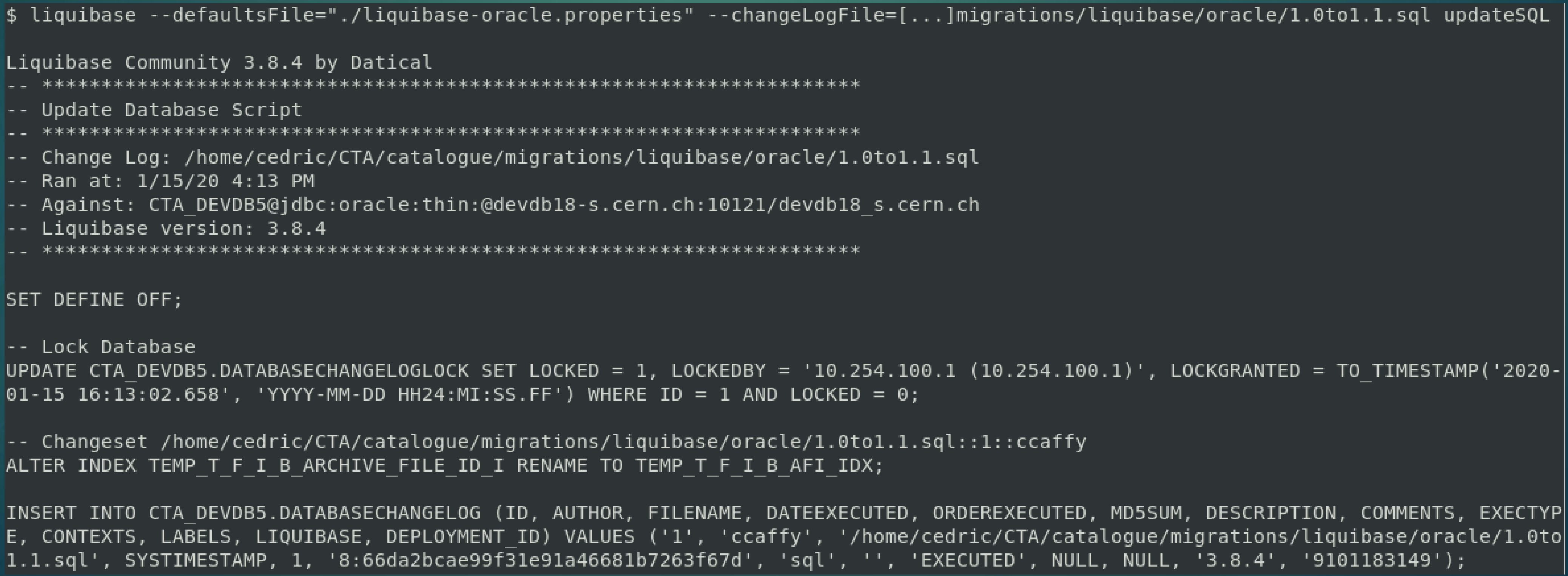


```
--liquibase formatted sql

--changeset ccaffy:1 failOnError:true dbms:oracle
ALTER INDEX TEMP_T_F_I_B_ARCHIVE_FILE_ID_I RENAME TO TEMP_T_F_I_B_AFI_IDX;
--rollback ALTER INDEX TEMP_T_F_I_B_AFI_IDX RENAME TO TEMP_T_F_I_B_ARCHIVE_FILE_ID_I;

--changeset ccaffy:2 failOnError:true dbms:oracle
ALTER TABLE USAGESTATS RENAME CONSTRAINT NN_USAGESTATS_GID TO USAGESTATS_GID_NN;
--rollback ALTER TABLE USAGESTATS RENAME CONSTRAINT USAGESTATS_GID_NN TO NN_USAGESTATS_GID;

--changeset ccaffy:3 failOnError:true dbms:oracle
ALTER TABLE USAGESTATS RENAME CONSTRAINT NN_USAGESTATS_TS TO USAGESTATS_TS_NN;
--rollback ALTER TABLE USAGESTATS RENAME CONSTRAINT USAGESTATS_TS_NN TO NN_USAGESTATS_TS;

--changeset ccaffy:4 failOnError:true dbms:oracle
ALTER TABLE ARCHIVE_ROUTE ADD CONSTRAINT ARCHIVE_ROUTE_SCI_TPI_UN UNIQUE(STORAGE_CLASS_ID, TAPE_POOL_ID);
--rollback ALTER TABLE ARCHIVE_ROUTE DROP CONSTRAINT ARCHIVE_ROUTE_SCI_TPI_UN;

--changeset ccaffy:5 failOnError:true dbms:oracle
UPDATE CTA_CATALOGUE SET SCHEMA_VERSION_MINOR=1
--rollback UPDATE CTA_CATALOGUE SET SCHEMA_VERSION_MINOR=0
```

1. Liquibase

1.3 Migration 1.0 to 1.1

1.3.3 Update the schema

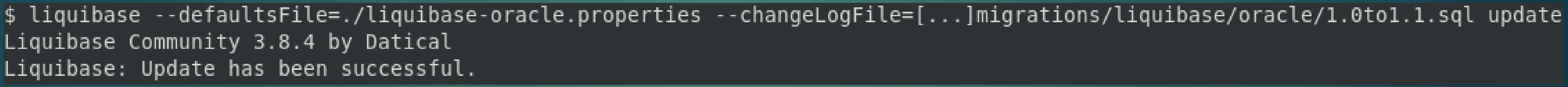


```
$ liquibase --defaultsFile="./liquibase-oracle.properties" --changeLogFile=[...]migrations/liquibase/oracle/1.0to1.1.sql updateSQL

Liquibase Community 3.8.4 by Datical
-- *********************************************************************
-- Update Database Script
-- *********************************************************************
-- Change Log: /home/cedric/CTA/catalogue/migrations/liquibase/oracle/1.0to1.1.sql
-- Ran at: 1/15/20 4:13 PM
-- Against: CTA_DEVDB5@jdbc:oracle:thin:@devdb18-s.cern.ch:10121/devdb18_s.cern.ch
-- Liquibase version: 3.8.4
-- *********************************************************************

SET DEFINE OFF;

-- Lock Database
UPDATE CTA_DEVDB5.DATABASECHANGELOGLOCK SET LOCKED = 1, LOCKEDBY = '10.254.100.1 (10.254.100.1)', LOCKGRANTED = TO_TIMESTAMP('2020-
01-15 16:13:02.658', 'YYYY-MM-DD HH24:MI:SS.FF') WHERE ID = 1 AND LOCKED = 0;

-- Changeset /home/cedric/CTA/catalogue/migrations/liquibase/oracle/1.0to1.1.sql::1::ccaffy
ALTER INDEX TEMP_T_F_I_B_ARCHIVE_FILE_ID_I RENAME TO TEMP_T_F_I_B_AFI_IDX;

INSERT INTO CTA_DEVDB5.DATABASECHANGELOG (ID, AUTHOR, FILENAME, DATEEXECUTED, ORDEREXECUTED, MD5SUM, DESCRIPTION, COMMENTS, EXECTYP
E, CONTEXTS, LABELS, LIQUIBASE, DEPLOYMENT_ID) VALUES ('1', 'ccaffy', '/home/cedric/CTA/catalogue/migrations/liquibase/oracle/1.0to
1.1.sql', SYSTIMESTAMP, 1, '8:66da2bcae99f31e91a46681b7263f67d', 'sql', '', 'EXECTUED', NULL, NULL, '3.8.4', '9101183149');
```

1. Liquibase

1.3 Migration 1.0 to 1.1

1.3.3 Update the schema

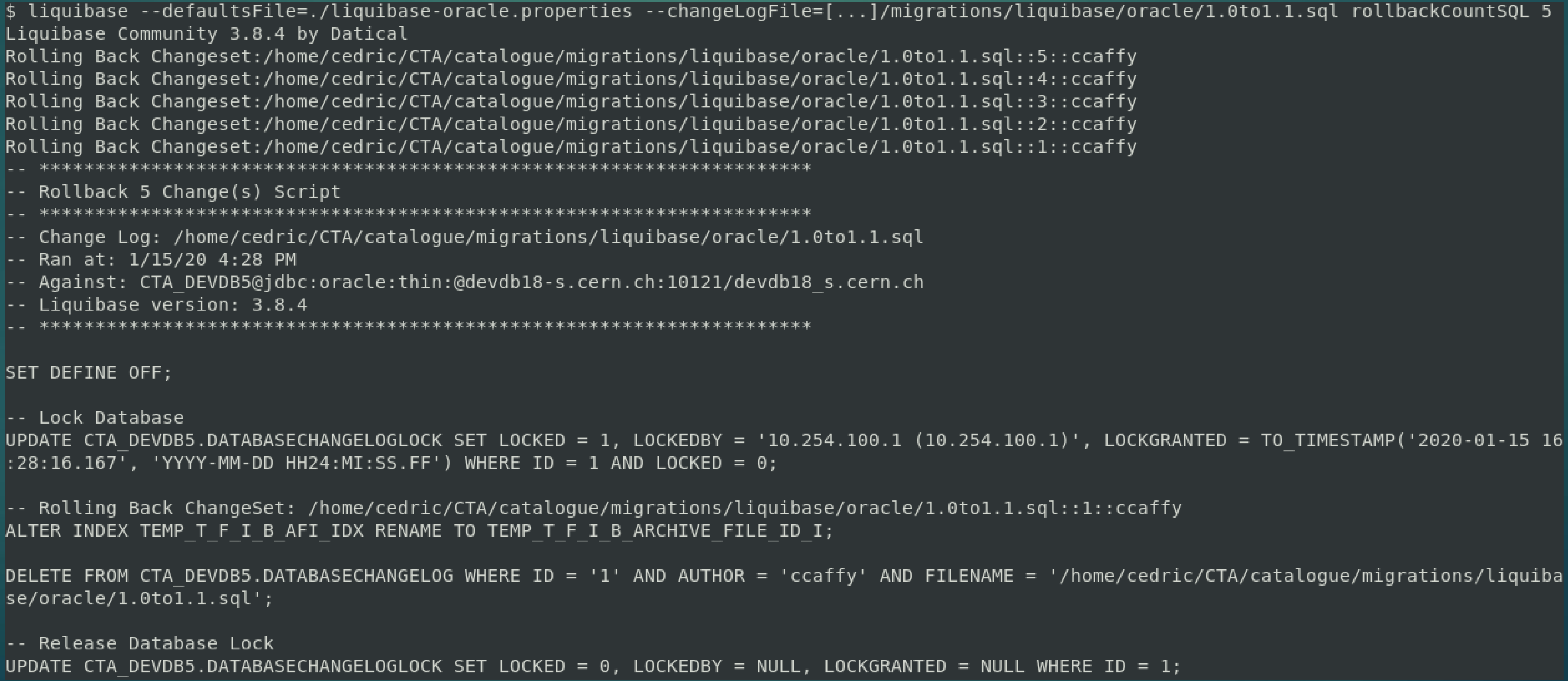


```
$ liquibase --defaultsFile=./liquibase-oracle.properties --changeLogFile=[...]migrations/liquibase/oracle/1.0to1.1.sql update
Liquibase Community 3.8.4 by Datical
Liquibase: Update has been successful.
```

1. Liquibase

1.3 Migration 1.0 to 1.1

1.3.4 Rollback



```
$ liquibase --defaultsFile=./liquibase-oracle.properties --changeLogFile=[...]/migrations/liquibase/oracle/1.0to1.1.sql rollbackCountSQL 5
Liquibase Community 3.8.4 by Datical
Rolling Back Changeset:/home/cedric/CTA/catalogue/migrations/liquibase/oracle/1.0to1.1.sql::5::ccaffy
Rolling Back Changeset:/home/cedric/CTA/catalogue/migrations/liquibase/oracle/1.0to1.1.sql::4::ccaffy
Rolling Back Changeset:/home/cedric/CTA/catalogue/migrations/liquibase/oracle/1.0to1.1.sql::3::ccaffy
Rolling Back Changeset:/home/cedric/CTA/catalogue/migrations/liquibase/oracle/1.0to1.1.sql::2::ccaffy
Rolling Back Changeset:/home/cedric/CTA/catalogue/migrations/liquibase/oracle/1.0to1.1.sql::1::ccaffy
-- *********************************************************************
-- Rollback 5 Change(s) Script
-- *********************************************************************
-- Change Log: /home/cedric/CTA/catalogue/migrations/liquibase/oracle/1.0to1.1.sql
-- Ran at: 1/15/20 4:28 PM
-- Against: CTA_DEVDB5@jdbc:oracle:thin:@devdb18-s.cern.ch:10121/devdb18_s.cern.ch
-- Liquibase version: 3.8.4
-- *********************************************************************

SET DEFINE OFF;

-- Lock Database
UPDATE CTA_DEVDB5.DATABASECHANGELOGLOCK SET LOCKED = 1, LOCKEDBY = '10.254.100.1 (10.254.100.1)', LOCKGRANTED = TO_TIMESTAMP('2020-01-15 16
:28:16.167', 'YYYY-MM-DD HH24:MI:SS.FF') WHERE ID = 1 AND LOCKED = 0;

-- Rolling Back ChangeSet: /home/cedric/CTA/catalogue/migrations/liquibase/oracle/1.0to1.1.sql::1::ccaffy
ALTER INDEX TEMP_T_F_I_B_AFI_IDX RENAME TO TEMP_T_F_I_B_ARCHIVE_FILE_ID_I;

DELETE FROM CTA_DEVDB5.DATABASECHANGELOG WHERE ID = '1' AND AUTHOR = 'ccaffy' AND FILENAME = '/home/cedric/CTA/catalogue/migrations/liquiba
se/oracle/1.0to1.1.sql';

-- Release Database Lock
UPDATE CTA_DEVDB5.DATABASECHANGELOGLOCK SET LOCKED = 0, LOCKEDBY = NULL, LOCKGRANTED = NULL WHERE ID = 1;
```

2. Migration procedures

2.1 Backward-compatible modifications

- The new database schema can be used by the old and the new version of CTA
  - Add a TABLE or a VIEW
  - Add a COLUMN
  - Remove a COLUMN that is not used by the old nor the new version of CTA
  - Remove a CONSTRAINT

2. Migration procedures

2.1 Backward-compatible modifications

- Migration procedure
  1. Backup database ?
  2. Execute update with Liquibase
  3. Use the `cta-catalogue-schema-verify` tool in order to verify the migration is successful

2. Migration procedures

2.2 Backward-incompatible modifications

- Database schema is modified in a way that it can not be used by the old version of CTA
  - Rename a COLUMN, a TABLE or a VIEW
  - Change the data type of a COLUMN
  - Remove a COLUMN, a TABLE or a VIEW that is still used by the old version of CTA

2. Migration procedures

2.2 Backward-incompatible modifications

- Rename a COLUMN, a TABLE or a VIEW

(COLUMN taken as example)

1. Add the COLUMN the with the new name and same datatype as the old one
2. Copy data from the old COLUMN to the new one

   => Add trigger to sync between both columns ?

   => If triggers not supported : two versions of CTA should be released
3. Update CTA_CATALOGUE version
4. Update all CTA components with the new version of CTA

2. Migration procedures

2.2 Backward-incompatible modifications

- Change the data type of a COLUMN
    - Same as previously, but we need to convert the values in between

2. Migration procedures

2.2 Backward-incompatible modifications

- Remove a COLUMN, a TABLE or a VIEW
  1. Update **all** CTA components
     - With the old schema version number associated to it
  2. Run the migration tool
  3. Update all CTA components in order to change the schema version it is supposed to run against

2. Migration procedures

2.2 Backward-incompatible modification

- Conclusion
  - Complexity of these modifications require a case-by-case migration plan
    - Create *transition version* of CTA that can run against "in-between" schema
    - Inform the user that a complex migration can be run only once and will not be rollbackable
    - Each complex migration could be documented in eoscta-docs
      - Create topic like "Migration from database schema version 1.6 to 2.0"

2. Migration procedures

2.3 Migration on different database types

- An easy migration on one database type can be a more tricky migration in another database type
  - Example : Renaming a CONSTRAINT "x" to "y" on table A
    - ORACLE
      - ALTER TABLE A RENAME CONSTRAINT x TO y;
    - SQLITE
      - Create a new table A-BIS that is the same as A
      - Copy content of A in A-BIS
      - DROP TABLE A
      - CREATE TABLE A with new constraint y
  - Suggestion : We do not create migration scripts as long as the related database type is not in production

3. GO / NO GO

- Team decision
  - GO ? NO GO ?
- If GO, I will create a documentation summarizing and explaining the migration procedures in eoscta-docs