
DataFax 3.7 SQL Loader

Eric Bosch
Clinical DataFax Systems Inc.

Rationale

- Make DataFax data available for export to SQL-based databases
- Provide some of the functionality that was promised with DataFax 4.0
- 3 phase solution
 - DataFax 3.7 includes implementation of phase 1

Requirements

- Easy to use
 - Leave the SQL / relational expertise to DBAs
- Target platforms: Oracle 8+, PostgreSQL 7.4+, MySQL4 .0+

Requirements

- Requires no additional DataFax configuration
- Works with a complete, ongoing, or new study
- Extract all primary data for each plate from DataFax
 - CRF image id only: CRF images are excluded
- Extract all QC notes
- Extract all reasons for data change

Requirements

- Create target schema, if necessary
- Create tables, one for each plate, in the target schema
- Each field of a plate becomes a column in the table
- Overwrite existing tables with same name
 - Privileges are preserved

Requirements

- Can be run repeatedly, each execution completely replacing the previous
- Transfer a DataFax study of 250,000 records to the target schema in 10 minutes or less
- No other changes to the target schema are permitted

Data Transformation

- DataFax data is imported as VARCHAR data into the target schema
- Field sizes from DataFax are applied to the target schema
 - Data which are larger than the field size are treated as invalid data
- Coded data is extracted in uncoded format
- Missing data is extracted in uncoded format
- Partial dates are extracted without imputation

Data Typing – Plate Data

Field Name	PostgreSQL	Oracle	MySQL
DFSTATUS	INT2	NUMBER(p)	INT2
DFVALID	INT2	NUMBER(p)	INT2
DFRASTER	VARCHAR(n)	VARCHAR2(n)	VARCHAR(n)
DFSTUDY	INT2	NUMBER(p)	INT2
DFPLATE	INT2	NUMBER(p)	INT2
DFSEQ	INT4	NUMBER(p)	INT4
DFPID	INT4	NUMBER(p)	INT4
8 ~ NF-3	VARCHAR(n)	VARCHAR2(n)	VARCHAR(n)
DFSCREEN	INT2	NUMBER(p)	INT2
DFCREATE	TIMESTAMP	DATE	DATETIME
DFMODIFY	TIMESTAMP	DATE	DATETIME

Data Typing – QC/Reason Data

DF Type	PostgreSQL	Oracle	MySQL
choice	INT2	NUMBER(p)	INT2
check	INT2	NUMBER(p)	INT2
integer	INT4	NUMBER(n)	INT4
number(nn:nn)	TIME	VARCHAR2(n)	TIME
float	NUMERIC(p,s)	NUMBER(p,s)	DECIMAL(p,s)
vas	NUMERIC(p,s)	NUMBER(p,s)	DECIMAL(p,s)
character	CHAR(n)	VARCHAR2(n)	CHAR(n)
string	VARCHAR(n)	VARCHAR2(n)	VARCHAR(n)
date	DATE	DATE	DATE
datetime	TIMESTAMP	DATE	DATETIME
timestamp	TIMESTAMP	DATE	DATETIME

Table Names

- QC notes -> DFQC
- Reason for data change -> DFREASON
- Plate data -> DFPLATE_###
- Table Type:

PostgreSQL	Oracle	MySQL
Relational	Relational (not Object)	InnoDB (not MyISAM)

Field Names

- Fields 1~7 and NF-2~NF -> uses DataFax generic name
- Fields 8~NF-3 -> uses DataFax unique name
- Field names matching an SQL keyword get an _ (underscore) appended

Field Names

- Any non-alphanumeric characters are replaced with _
- If field name starts with a digit, DF_ is prepended
- Field names are truncated to 30 chars
- A sequence number is appended to each non-unique field name

Operation

- **Command-line interface**

```
DFsqlload [-flavour oracle|postgresql|mysql] [-q]
          [-d drfname] DFstudy#
          server:database:schema[.tablespace][:username:password]
```

- **Default database type is Oracle**

Operation

1. Is named 'schema' defined in 'database' on 'server'?
 - If not, create the schema and grant privileges
2. Read DataFax definition from DFschema
3. Backup table and column privileges, if any

Operation

4. Drop existing tables
5. Generate and execute SQL statements to create tables
6. Restore privileges from backup, if any
7. Use DFexport.rpc to generate files of primary records for import
8. Import records

Database Specifics

- Oracle ignores 'server'
 - Performs lookup of 'database' in tnsnames.ora
- PostgreSQL and MySQL ignore 'tablespace'
 - Tablespace is an Oracle concept

Credentials

- Username and password can be specified on command-line or read from:
 - PostgreSQL: Read ~/.pgpass
 - MySQL: Read ~/.my.cnf
 - Oracle: Read ~/.orapass or use "external credentials"

Phase 2

- Optionally import typed data, replacing VARCHARs of phase 1
- Drop and recreate tables only if DataFax plate definition has changed
- Modules become separate tables

Phase 3

- Existing phase 2 solution is preserved
- Optional, write-only, real-time updates by DataFax server to SQL database
- Keeps a queue of failed/pending transactions if database is unavailable