BKM_DATS: Databázové systémy
5. Exchanging Data

Vlastislav Dohnal

Contents

- Exporting data from DB
- Importing data to DB
- Backups

Why data exchange

- cooperation between two or more **proprietary** systems of different platforms and architectures
- a data export to a common / human-readable format is typically provided
  - a widely supported format is CSV (comma-separated values)
    - see Wikipedia
- Here are common scenarios when CSV can be useful:
  - Customers pay a service company through a bank. Then, the bank provides payment records to the service company using a CSV file.
  - An operational system built in-house needs to be integrated into an ERP system.
  - A biometrics system (door lock) needs to be integrated into a human resources system for attendance purposes.

Exporting Data

- Standard database backup tools can be used
- Specific SQL command
  - COPY table TO file
  - COPY (query) TO file
    - instead of file, stdout can also be used.
  - The option `WITH (FORMAT CSV, HEADER TRUE)` produces a CSV file (for Excel, e.g.)
    - `COPY weather TO STDOUT WITH (FORMAT CSV, HEADER TRUE)`
  - https://www.postgresql.org/docs/16/sql-copy.html

Importing Data

- Standard database recovery tools can be used
- SQL command
  - `COPY table FROM file`
  - Reads the file and for each line inserts the data into the table.
    - First item is inserted into the first attribute, etc.
    - You may give the attributes to fill by:
      - `COPY table (A,B,C) FROM file`
  - Once again, the format and header can be used in options

Attaching Files as Tables

- In general, this feature is available in many RDBMSes under the name “foreign table”.
    - or import functionality from various sources is available
- Foreign Data Wrapper mechanism in PostgreSQL
  - allows to access external sources as regular “local” tables
- postgres_fdw
  - allows to attach a table from another PostgreSQL server
  - read-write access
- file_fdw
  - attaches a file (output of COPY command) as a table, so a CSV file too.
  - read-only access
- Needs to be activated
  - CREATE EXTENSION *extension_name;*

Attaching Files as Tables - postgres_fdw

- See documentation
  - https://www.postgresql.org/docs/16/postgres-fdw.html

-
  ```
  CREATE EXTENSION postgres_fdw;
  ```
  - actives the module

-
  ```
  CREATE SERVER db_fi
       FOREIGN DATA WRAPPER postgres_fdw
       OPTIONS (host 'db.fi.muni.cz', dbname 'pgdb', port
  '5432’);
  ```
  - Connection information only, excluding user and password

-
  ```
  CREATE USER MAPPING FOR CURRENT_USER SERVER db_fi
       OPTIONS ("user" ‘xxloginxx', password ‘xxpwdxx’);
  ```
  - Set credentials to access the remote server for “CURRENT_USER” / “PUBLIC” / particular user.

Attaching Files as Tables - postgres_fdw

-
  ```
  IMPORT FOREIGN SCHEMA xxsch FROM SERVER db_fi INTO
  xxlocal;
  ```
  - Import all tables in schema xxsch on the remote server into xxlocal schema.

-
  ```
  CREATE FOREIGN TABLE predmet ()
      INHERITS (xxsch.predmet) SERVER db_fi;
  ```
  - Creates a table backed by the remote server.
  - https://www.postgresql.org/docs/16/sql-createforeigntable.html

Attaching Files as Tables - file_fdw

- See documentation
  - https://www.postgresql.org/docs/16/file-fdw.html
- `CREATE EXTENSION file_fdw;`
  - actives the module
- `CREATE SERVER csv_file FOREIGN DATA WRAPPER file_fdw;`
  - create a “remote server”
-
  ```
  CREATE FOREIGN TABLE csv
      ( columns, …)
      SERVER csv_file;
      OPTIONS (filename ‘xxx’, format ‘csv’, header ‘1’);
  ```
  - Creates a table backed by a CSV file stored in the local server.

Backup & Restore

- DBMSes have tools for database backup & restore
  - Typically, you may select what to backup; and also, what to restore
- PostgreSQL
  - `pg_dump dbname >dumpfile`
    - a client app that connect to DB server and get the contents of the database
    - output can be read by newer server versions or different architecture (32-bit vs 64-bit)
    - it does not dump information about roles or tablespaces (because those are cluster-wide rather than per-database).
  - `psql dbname <dumpfile`
    - restore the data from dumpfile into the dbname database
    - dbname must exist in advance; if not, do first:
      - `createdb -T template0 dbname`
  - `pg_dumpall`

Backup & Restore

- PostgreSQL
  - `pg_dumpall  >dumpfile`
    - backs up each database in a given cluster, and also preserves roles and tablespace definitions
    - the option `--globals-only` can be used to dump just roles and tablespace definition.
  - `psql -f dumpfile postgres`
    - should be executed as a superuser

  - Dumpfiles are bulky, so it is wise to compress them.

- pgAdmin (web) app
  - Check context menus on databases, tables, schemas for Backup and Restore