# SQOOP

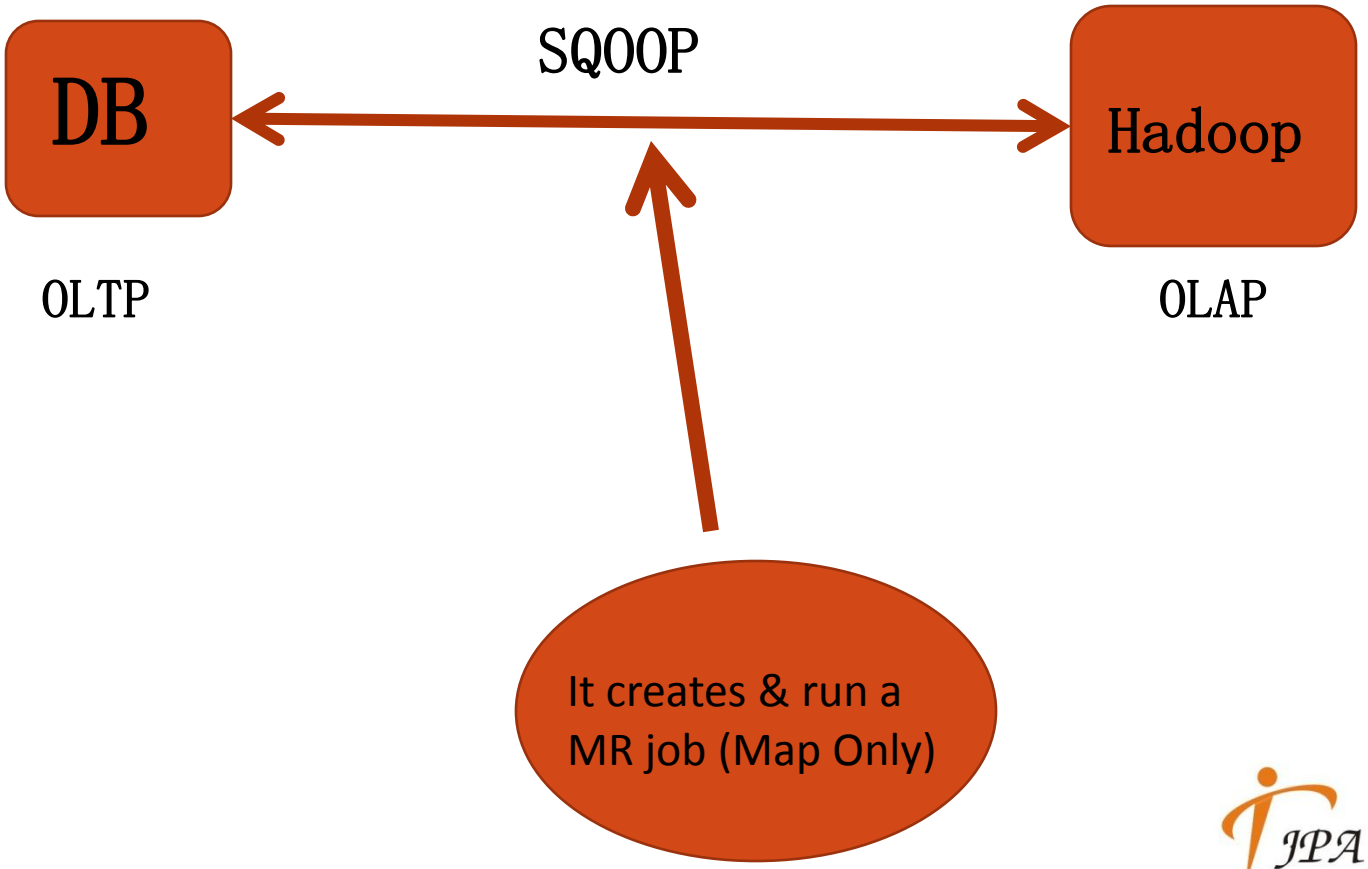
JPA Solutions
Center Of Excellence

# Introduction – Sql to Hadoop

- Bulk data transfer tool
- To import and export data from a relational database into Hadoop for processing
- Map only job.
- command-line tool
- Integrates with Hive & Hbase
- Support plugins via connector based architecture

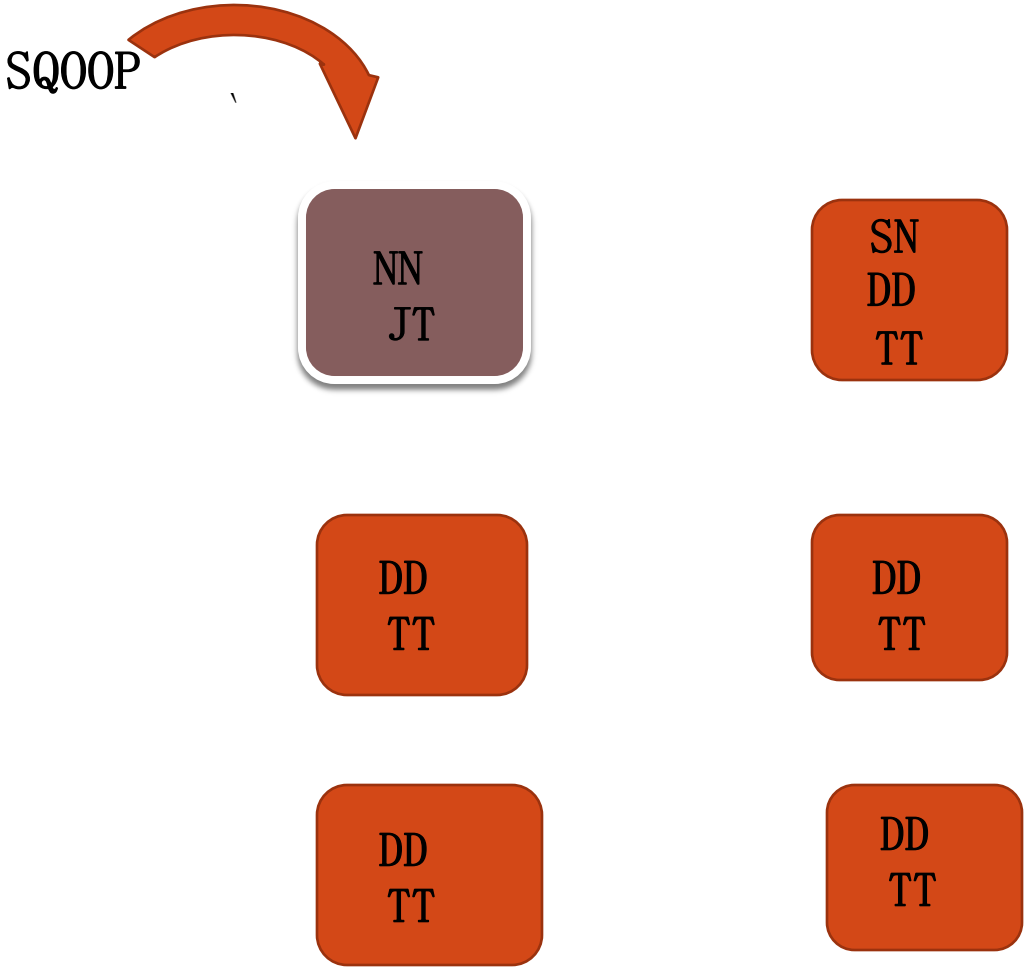
JPA Solutions
Center Of Excellence

# Basic Architecture




IPA Solutions
Center Of Excellence

# Real Time/ Production Side




IPA Solutions
Center Of Excellence

# File Formats

- Two file formats:
  - Delimited text
  - SequenceFiles

- Delimited Text File
  - Default import format, explicitly as --as-textfile
  - Default delimiters are comma(,) for fields , a newline (\n) for records.

- Sequence File --as-sequencefile

IPA Solutions
Center Of Excellence

# Supported Databases

- Uses JDBC compatibility layer to talk with the databases
- Sample List of DBs
  - MySQL
  - MS SQL
  - PostgreSQL
  - Oracle
- Need to add vendor specific drivers in $SQOOP_HOME/lib

# Installation

- Download Sqoop-*.tar.gz
- tar -xvf sqoop-*.*.tar.gz
- export HADOOP_HOME=/some/path/hadoop-dir
- Please add the vendor Specific JDBC jar to $SQOOP_HOME/lib
- Change to Sqoop Bin folder
  - ./sqoop help

# Sqoop Commands

- sqoop help
  - Or We can use:  sqoop COMMAND [ARGS]
- Available commands:
  - codegen
    - Generate code to interact with database records
  - create-hive-table
    - Import a table definition into Hive
  - eval
    - Evaluate a SQL statement and display the results
  - export
    - Export an HDFS directory to a database table help List available commands
  - import
    - Import a table from a database to HDFS
  - import-all-tables
    - Import tables from a database to HDFS
  - list-databases
    - List available databases on a server
  - list-tables
    - List available tables in a database version Display version information

IPA Solutions
Center Of Excellence

# Mysql Connectivity

- mysql -u root -p
- Enter password:root
- show databases;
- use test;
- mysql>CREATE TABLE patient( pid INT(10),name VARCHAR(20),durg VARCHAR(20),tot_amt INT(10));
- mysql>insert into patient values(1,'saravanan','avil',100);
- mysql>insert into patient values(2,'senthil','metacin',200);
- mysql>insert into patient values(3,'Gowtham','paracetamol',300);
- mysql>select * from patient;

# Sqoop Evaluate

- Evaluate a SQL statement

- bin/sqoop eval --connect jdbc:mysql://localhost/test -username root -password root -query "SELECT * FROM patient"
- bin/sqoop eval --connect jdbc:mysql://localhost/test -username root -password root --query "SELECT * FROM patient LIMIT 2"
- bin/sqoop eval --connect jdbc:mysql://localhost/test -username root -password root --query "INSERT INTO patient VALUES(4, ‘amudhan’,'avil',400)H

# Sqoop List

- Sqoop-list-databases

- bin/sqoop list-databases --connect jdbc:mysql://localhost/information_schema -username root -password root

- Sqoop-list-tables
  bin/sqoop list-tables --connect jdbc:mysql://localhost/test username root -password root

# Sqoop Import: mysql to hdfs

- bin/sqoop import --connect jdbc:mysql://localhost/test -username root -password root --table patient -m 1

  - Imports “patient” table into HDFS directory
    - Data imported as text or SequenceFiles
  - Sqoop generates java file(patient.java) for our use
    - Instead we can use codegen
  - `bin/hadoop dfs -cat /user/username/patient/part-00000`
    - All values re displayed
    - These files can be used as input to MR jobs.

# Cont.

- Increasing parallelism (number of mappers)
- bin/sqoop import --connect jdbc:mysql://localhost/test -username root -password root --table patient --split-by column name(pid) -m 2

- target-directory
- bin/sqoop import -connect jdbc:mysql://localhost/test -username root -password root --table patient --target-dir /user/output -m 1

- mysql to hdfs import-all-tables
- bin/sqoop import-all-tables --connect jdbc:mysql://localhost/test -username root -password root -m 1

# Hive Integration

- bin/sqoop-import --connect jdbc:mysql://localhost/test -username root -password root --table patient --hive-table patientthive --create-hive-table --hive-import -m 1

Other Hive Options

--hive-import

--hive-overwrite

--hive-partition-key

# Hbase Integration

- bin/sqoop import --connect jdbc:mysql://localhost/test --username root --password root -table patient --hbase-table patienthbase2 --column-family datasqoop --hbase-row-key pid --hbase-createtable -m 1

Options

--column-family <family>

--hbase-create-table

--hbase-row-key <col>

--hbase-table <table-name>

JPA Solutions
Center Of Excellence

# Sqoop Export

- Exports a set of files from HDFS back to an RDBMS
- The target table must already exist in the database
- The input files are read and parsed into a set of records according to the user-specified delimiters.
- Does not export from HBase

# Cont.

- hdfs to mysql

- bin/sqoop export --connect jdbc:mysql://localhost/test -username root -password root --table patient --export-dir /user/amudhan/pati5 Sqoop_students_datadotz

- hive to mysql:

- bin/sqoop export --connect jdbc:mysql://localhost/test --table patient --export-dir /user/hive/warehouse/patient --username root --password root -m 1

# Miscellaneous

- sqoop-merge
- sqoop-codegen
- sqoop-job
- sqoop-metastore

# THANK YOU

IPA Solutions
Center Of Excellence