RapidsDB

A BORRUI DATA COMPANY

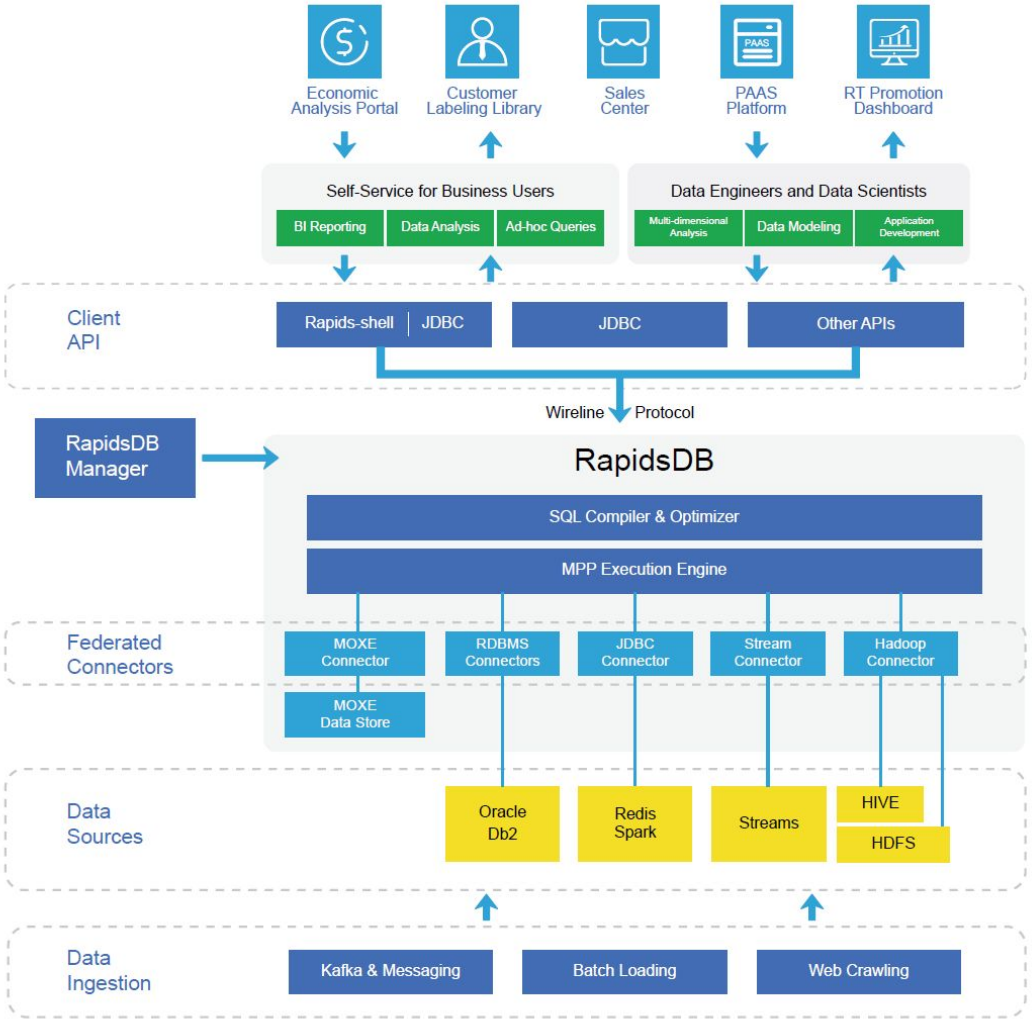
ISO BigData

CODE DEMO

World Leading Real-time Big Data Analytics Platform

OverView

- Rapids Shell Login
- MYSQL / RPDSQL Connector
- HDFS / HIVE Connector
- S3 Data Import
- CSV Data Import
- JSON Data Import
- TPCH Query
- Python Code Example
- GEO Polygon Query



Rapids Shell Login

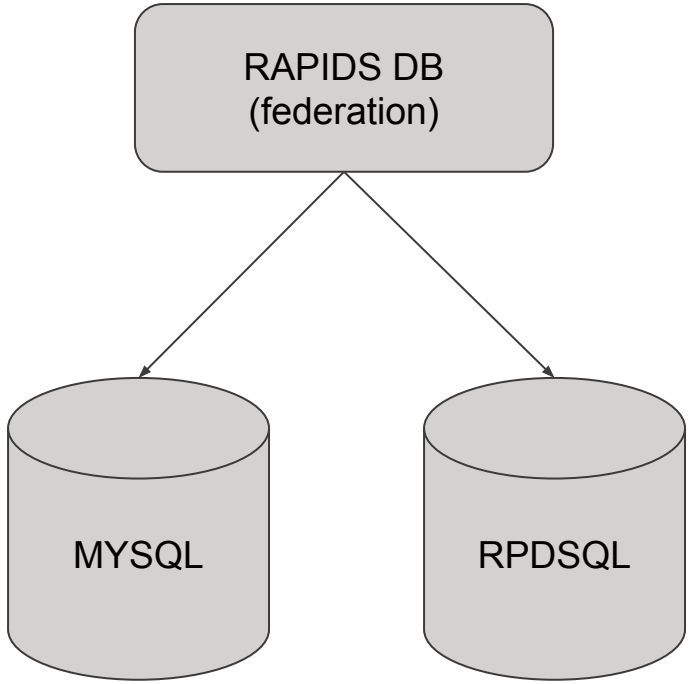
```
rapids-shell.sh -h ip -p port ##then input username and password

show connectors;

use connector CONNECTOR_NAME;
```

```
shineyear@fans-MacBook-Pro rapids-shell-4.0.6 % ./rapids-shell.sh -h 119.13.101.131 -p 54333
Please enter a username > rapids
Please enter the password for user 'RAPIDS' >
rapids > show connectors;
FEDERATION_NAME   CONNECTOR_NAME   CONNECTOR_TYPE     IS_ENABLED CONNECTOR_DDL
---------------   --------------   --------------     ---------- -------------
DEFAULTFED        METADATA         METADATA                 true CREATE CONNECTOR METADATA TYPE METADATA NODE * CATALOG * SCHEMA * TABLE *

DEFAULTFED        MOXE             MOXE                     true CREATE CONNECTOR MOXE TYPE MOXE WITH PARTITIONS_PER_NODE=2, MEM_PER_NODE='10GB' NODE * CATALOG * SCHEMA * TABLE *

DEFAULTFED        MYSQL_SSL_ON     MYSQL                    true CREATE CONNECTOR MYSQL_SSL_ON TYPE MYSQL WITH USER='rfamro', USE_SSL, PORT=4497, HOST='mysql-rfam-public.ebi.ac.uk', DATABASE='Rfam' NODE
 * CATALOG * SCHEMA * TABLE *
DEFAULTFED        RPDSQL_PUBLIC    JDBC                     true CREATE CONNECTOR RPDSQL_PUBLIC TYPE JDBC WITH USER='root', CONNECTIONSTRING='jdbc:mysql://119.8.186.55:3306/test' NODE * CATALOG * SCHEMA
 * TABLE *
DEFAULTFED        SF1              TPCHDB                   true CREATE CONNECTOR SF1 TYPE TPCHDB WITH SCALE_FACTOR=1 NODE * CATALOG * SCHEMA * TABLE *


5 row(s) returned (0.29 sec)
rapids >
```

MYSQL / RPDSQL Connector

```
CREATE CONNECTOR MYSQL_SSL_ON
TYPE MYSQL WITH
DATABASE='Rfam',
USER='rfamro',
PORT=4497,
USE_SSL='TRUE',
HOST='mysql-rfam-public.ebi.ac.uk'
NODE * CATALOG * SCHEMA * TABLE *;
```

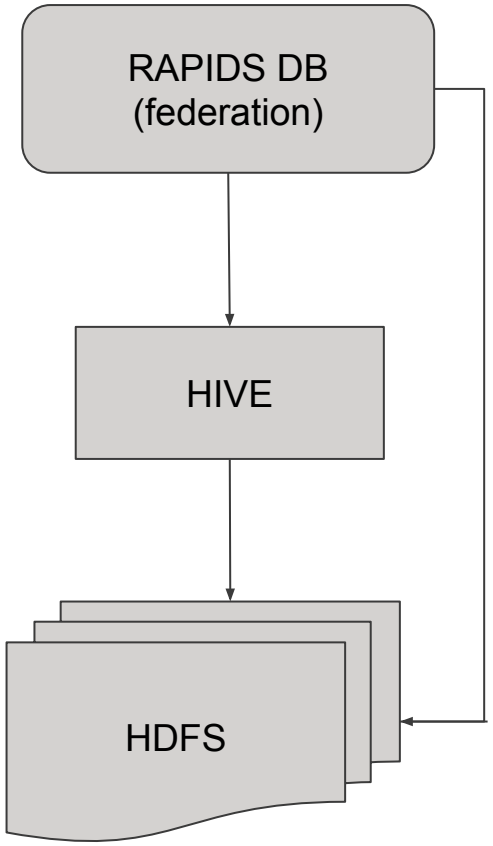
```
CREATE CONNECTOR RPDSQL_PUBLIC
TYPE JDBC WITH
CONNECTIONSTRING='jdbc:mysql://119.8.186.55:3306/test',
USER='root'
NODE * CATALOG * SCHEMA * TABLE *;
```



HDFS / HIVE Connector

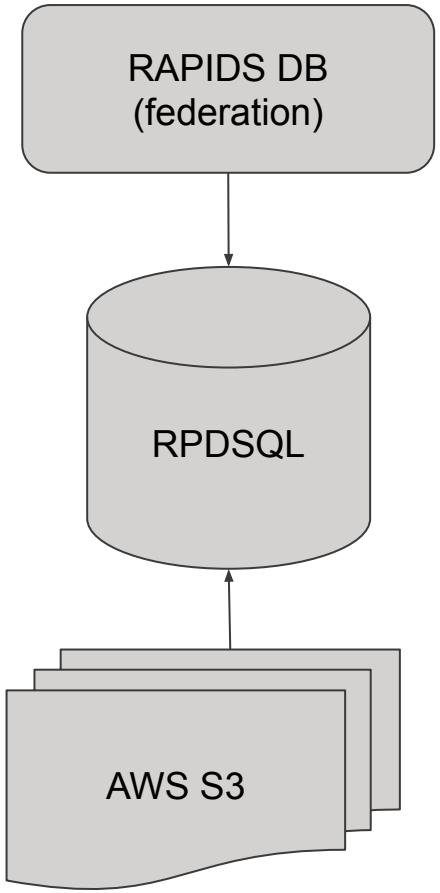
```
CREATE CONNECTOR HDFS5
TYPE HADOOP
WITH hdfs='hdfs://159.138.83.27:9000',
format='delimited',
delimiter=',',
user='hadoop',
partitions_per_node = '1'
CATALOG * SCHEMA *
TABLE TEST5 (C1 INTEGER, C2 VARCHAR)
WITH PATH='/test5';
```

```
CREATE CONNECTOR HIVE2
TYPE JDBC
WITH CONNECTIONSTRING='jdbc:hive2://127.0.0.1:10000/default'
NODE * CATALOG * SCHEMA * TABLE *;
```



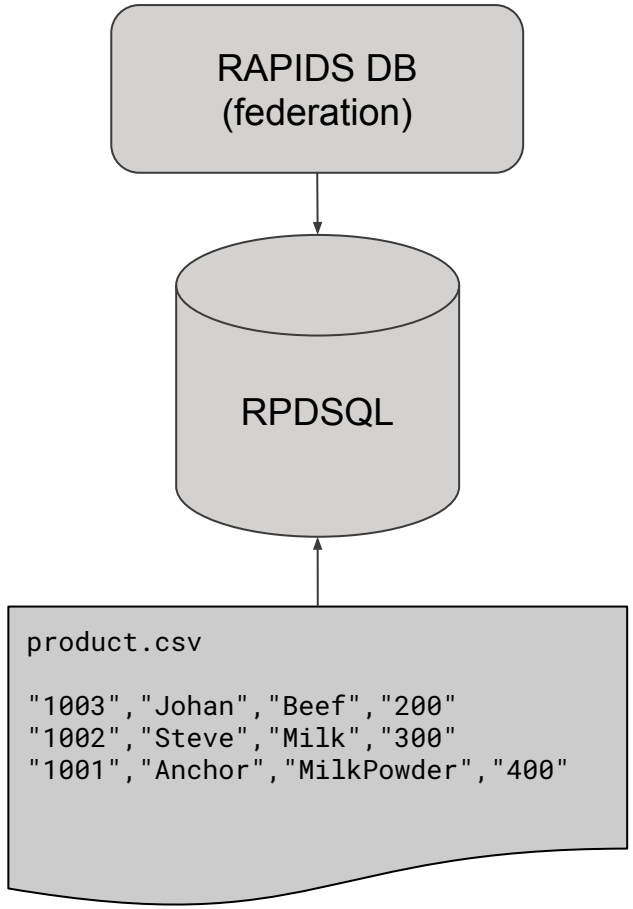
S3 Data Import

```
use connector RPDSQL_PUBLIC;

create table beta (num VARCHAR(20), price VARCHAR(20));


CREATE PIPELINE p AS LOAD DATA S3 'bucket-name'
CONFIG '{"region": "region-name"}'
CREDENTIALS '{
"aws_access_key_id": "key-id",
"aws_secret_access_key": "secret-key"
}'
SKIP DUPLICATE KEY ERRORS
INTO TABLE beta
FIELDS TERMINATED BY ',' ENCLOSED BY '' ESCAPED BY '\\'
LINES TERMINATED BY '\n' STARTING BY '';

test pipeline p;

show pipelines;

start pipeline p;
```



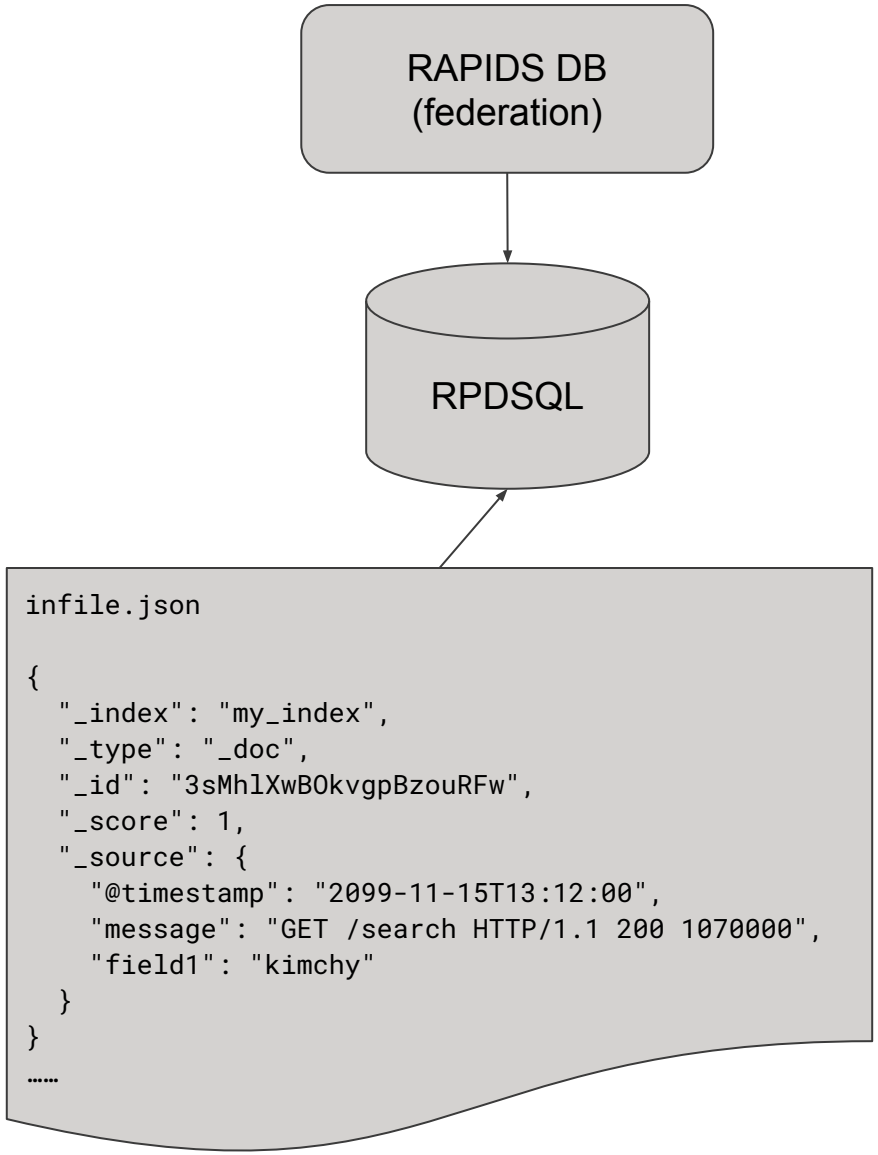
CSV Data Import

```
use connector RPDSQL_PUBLIC;

create table csv (
id VARCHAR(100) NOT NULL,
name VARCHAR(100) NOT NULL,
category VARCHAR(100) NOT NULL,
price VARCHAR(100) NOT NULL,
PRIMARY KEY ( name )
);

LOAD DATA LOCAL  INFILE 'product.csv' INTO
TABLE csv COLUMNS TERMINATED BY ',' ENCLOSED
BY '"';

select * from csv;
```



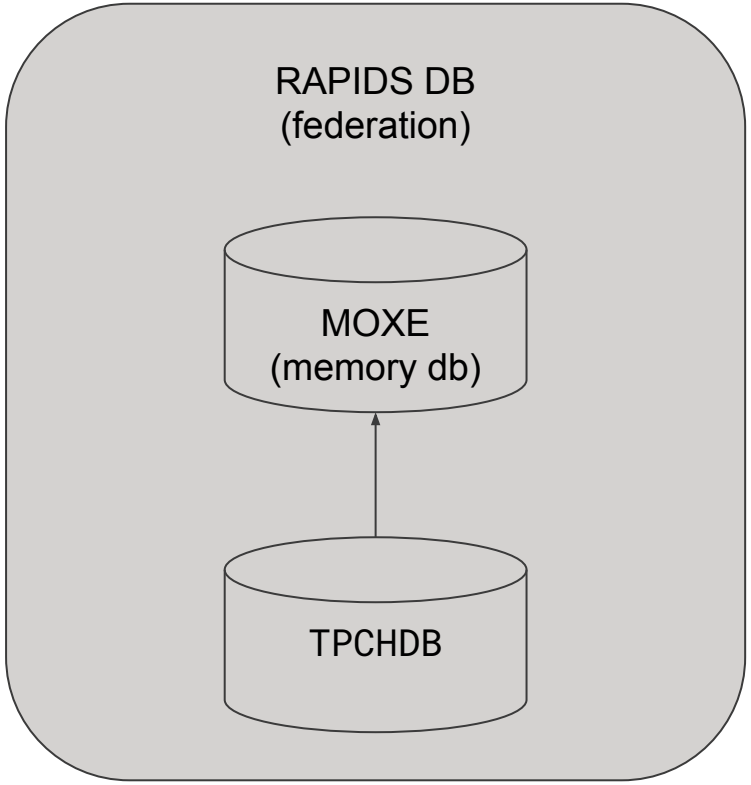
JSON Data Import

```
create table test (
_id varchar(10),
_index varchar(20),
_score int,
field varchar(100),
message varchar(50),
timest datetime(6),
raw json,
KEY(_id),
KEY(timest)
);

LOAD DATA LOCAL INFILE "infile.json" INTO TABLE test FORMAT JSON (
_id <- _id default NULL,
_index <- _index default NULL,
_score <- _score DEFAULT 0,
@avar <- _source default NULL,
raw <- % default NULL
)
SET
field = json_extract_string(@avar, 'field1'),
message = json_extract_string(@avar, 'message'),
timest = to_date(json_extract_string(@avar, '@timestamp'), 'YYYY-MM-DDTHH24:MM:SS');

select json_extract_string(raw, "_id") from test;
select json_extract_string(raw::_source, "message") from test;

update test set raw = JSON_SET_STRING(raw::_source, "field1", "rice") where _id =
'3sMhlXwBOkvgpBzouRFw';
```



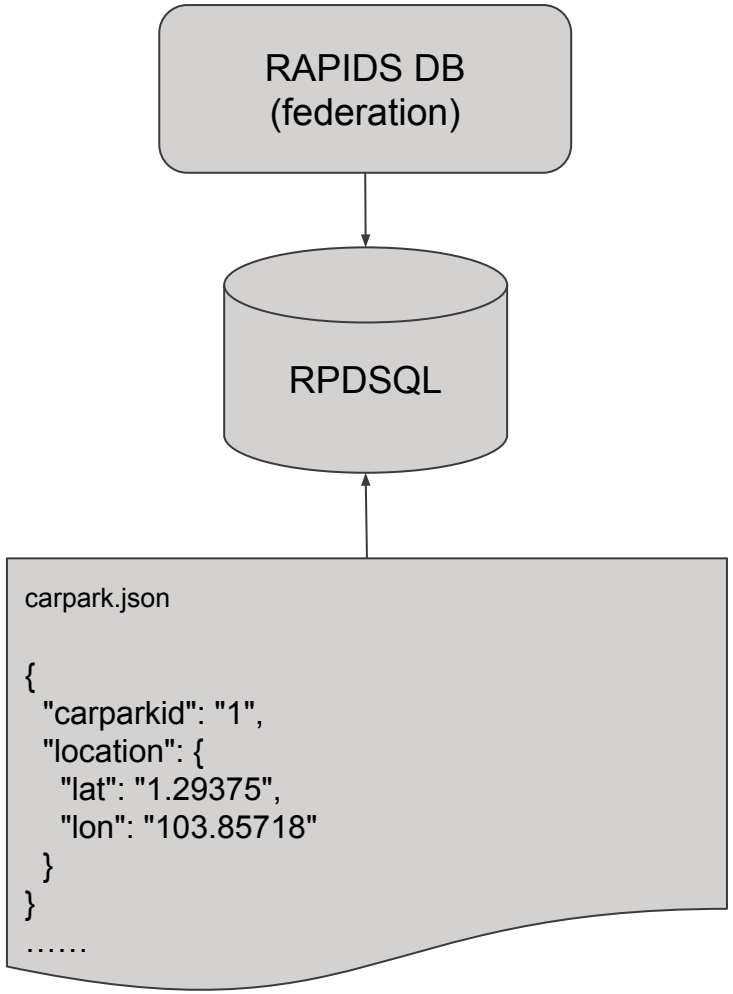
TPCH Query

```
CREATE CONNECTOR SF1 TYPE TPCHDB WITH SCALE_FACTOR='1';

CREATE CONNECTOR MOXE TYPE MOXE WITH
PARTITIONS_PER_NODE='2', MEM_PER_NODE='10GB';

#CREATE TABLES

insert into moxe.lineitem select * from sf1.lineitem;
insert into moxe.orders select * from sf1.orders;
insert into moxe.partsupp select * from sf1.partsupp;
insert into moxe.part select * from sf1.part;
insert into moxe.supplier select * from sf1.supplier;
insert into moxe.customer select * from sf1.customer;
insert into moxe.nation select * from sf1.nation;
insert into moxe.region select * from sf1.region;

#RUN SQL QUERY
```



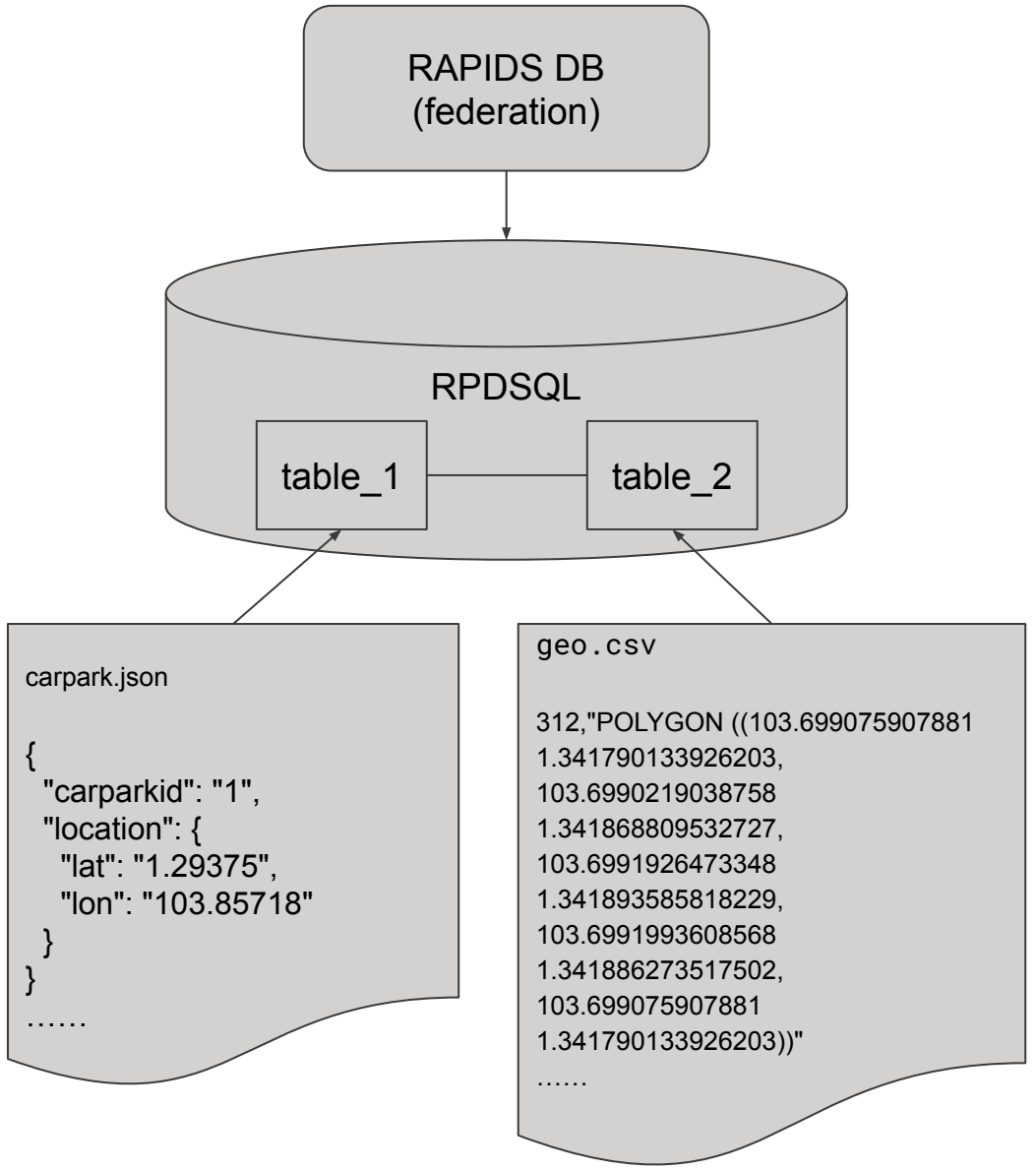
Python Code Example

```
#import pyRDP
import pyRDP as pyRDP
import json

#open json file
with open('carpark.json') as f:
  data = json.load(f)

#open connections
conn = pyRDP.connect(host = "domain or ip", port = 4333, user = 'RAPIDS',
password = 'rapids', catalog = 'connector_name', schema = "database_name")
cursor = conn.cursor()

#process data
for i in data:
    carparkid = i["carparkid"]
    lat = i["location"]["lat"]
    lon = i["location"]["lon"]

#execute SQL
sql = "INSERT INTO table_name (carparkid, location, lat, lon) VALUES
('"+carparkid+"', 'POINT("+lon+" "+lat+")', "+lat+", "+lon+")"
cursor.execute(sql)

#close connection
conn.close()
```

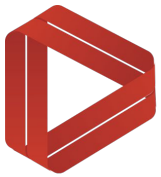


GEO Polygon Query

```
create table table_1 (
  carparkid varchar(10),
  location GEOGRAPHYPOINT,
  lat double,
  lon double,
  index(location)
);

create table table_2 (
  FEATID varchar(100) default null,
  GEOMETRY GEOGRAPHY default null,
  index (GEOMETRY) with (resolution = 8),
  index (FEATID)
);

load data local infile "geo.csv" into table table_2 FIELDS TERMINATED BY
'|' ENCLOSED BY '"';

SELECT carparkid, location FROM table_1 WHERE
ROUND(GEOGRAPHY_DISTANCE("POINT(1.85718 2.29375)", location), 0) < 5000;

SELECT c.carparkid, h.FEATID FROM table_1 c, table_2 h WHERE
GEOGRAPHY_CONTAINS(h.GEOMETRY, c.location);
```



Thank You !



RapidsDB

A BORRUI DATA COMPANY

Intelligent Data, Enabling Future !

doc.rapidsdb.sg

Please email your questions to: yylai@rapidsdb.sg

@RapidsDB