Article
Anton Umnikov · Dec 4, 2020 7m read

Open Exchange

Reading AWS S3 data on COVID as SQL table in IRIS

IRIS External Table is an InterSystems Community Open Source Project, that allows you to use files, stored in the local file system and cloud object storage such as AWS S3 as SQL Tables.

It can be found on Github https://github.com/intersystems-community/IRIS-ExternalTable Open Exchange https://openexchange.intersystems.com/package/IRIS-External-Table and is included in InterSystems Package Manager ZPM.

To instal External Table from GitHub use:

```
git clone https://github.com/antonum/IRIS-ExternalTable.git
iris session iris
USER>set sc = ##class(%SYSTEM.OBJ).LoadDir("<path-to>/IRIS-
ExternalTable/src", "ck",,1)
```

To install with ZPM Package Manager use:

```
USER>zpm "install external-table"
```

Working with local files

Let ' s create a simple file that looks like this:

```
a1,b1
a2,b2
```

Open your favorite editor and create the file or just use a command line in linux/mac:

```
echo $'a1,b1\na2,b2' > /tmp/test.txt
```

In IRIS SQL Create table to represent this file:

```
create table test (col1 char(10),col2 char(10))
```

Convert table to use external storage:

```
CALL EXT.ConvertToExternal(
```

```
    'test',
    '{
        "adapter":"EXT.LocalFile",
        "location":"/tmp/test.txt",
        "delimiter": ","
    }')
```

And finally – query the table:

```
select * from test
```

If everything works out as expected you should see output like:

```
col1    col2
a1  b1
a2  b2
```

Now get back to the editor, change the content of the file and rerun the SQL query. Ta-Da!!! You are reading new values from your local file in SQL.

```
col1    col2
a1  b1
a2  b99
```

Reading data from S3

At <https://covid19-lake.s3.amazonaws.com/index.html >you can get access to constantly updating data on COVID, stored by AWS in the public data lake.

Let ' s try to access one of the data sources in this data lake: s3://covid19-lake/rearc-covid-19-nyt-data-in-usa/csv/us-states

If you have AWS command line tool installed – you can repeat the steps below. If not – skip straight to SQL part. You don ' t need anything AWS – specific to be installed on your machine to follow along with SQL part.

```
$ aws s3 ls s3://covid19-lake/rearc-covid-19-nyt-data-in-usa/csv/us-states/
2020-12-04 17:19:10     510572 us-states.csv

$ aws s3 cp s3://covid19-lake/rearc-covid-19-nyt-data-in-usa/csv/us-states/us-
states.csv .
download: s3://covid19-lake/rearc-covid-19-nyt-data-in-usa/csv/us-states/us-
states.csv to ./us-states.csv

$ head us-states.csv
date,state,fips,cases,deaths
2020-01-21,Washington,53,1,0
2020-01-22,Washington,53,1,0
2020-01-23,Washington,53,1,0
2020-01-24,Illinois,17,1,0
2020-01-24,Washington,53,1,0
2020-01-25,California,06,1,0
```

```
2020-01-25,Illinois,17,1,0
2020-01-25,Washington,53,1,0
2020-01-26,Arizona,04,1,0
```

So we have a file with a fairly simple structure. Five delimited fields.

To expose this S3 folder as the External Table – first, we need to create a "regular" table with the desired structure:

```
-- create external table
create table covid_by_state (
    "date" DATE,
    "state" VARCHAR(20),
    fips INT,
    cases INT,
    deaths INT
)
```

Note that some field names like “ Date ” are reserved words in IRIS SQL and need to be surrounded by double quotes.
Then – we need to convert this “ regular ” table to the “ external ” table, based on AWS S3 bucket and CSV type.

```
 -- convert table to external storage
call EXT.ConvertToExternal(
    'covid_by_state',
    '{
    "adapter":"EXT.AWSS3",
    "location":"s3://covid19-lake/rearc-covid-19-nyt-data-in-usa/csv/us-states/",
    "type": "csv",
    "delimiter": ",",
    "skipHeaders": 1
    }'
)
```

If you'll look closely - EXT.ExternalTable procedures arguments are table name and then JSON string, containing multiple parameters, such as location to look for files, adapter to use, delimiter etc. Besides AWS S3 External Table supports Azure BLOB storage, Google Cloud Buckets and the local filesystem. GitHub Repo contains references for the syntax and options supported for all the formats.

And finally – query the table:

```
-- query the table
select top 10 * from covid_by_state order by "date" desc

[SQL]USER>>select top 10 * from covid_by_state order by "date" desc
2.  select top 10 * from covid_by_state order by "date" desc

date    state   fips    cases   deaths
2020-12-06  Alabama 01  269877  3889
2020-12-06  Alaska  02  36847   136
2020-12-06  Arizona 04  364276  6950
2020-12-06  Arkansas    05  170924  2660
2020-12-06  California  06  1371940 19937
2020-12-06  Colorado    08  262460  3437
```

```
2020-12-06  Connecticut 09  127715  5146
2020-12-06  Delaware    10  39912   793
2020-12-06  District of Columbia    11  23136   697
2020-12-06  Florida 12  1058066 19176
```

It takes understandably more time to query data from the remote table, then it is for the " IRIS native " or global based table, but it is completely stored and updated on the cloud and being pulled into IRIS behind the scenes.

Let ' s explore a couple of more features of the External Table.

%PATH and tables, based on multiple files

In our example folder in the bucket contains just one file. More often then not it would have multiple files of the same structure where filename identifies either timestamp or deviceid of some other attribute that we ' ll want to use in our queries.

%PATH field is automatically added to every External Table and contains full path to the file where row was retrieved from.

```
select top 5 %PATH,* from covid_by_state
```

```
%PATH   date    state   fips    cases   deaths
s3://covid19-lake/rearc-covid-19-nyt-data-in-usa/csv/us-states/us-
states.csv    2020-01-21  Washington  53  1   0
s3://covid19-lake/rearc-covid-19-nyt-data-in-usa/csv/us-states/us-
states.csv    2020-01-22  Washington  53  1   0
s3://covid19-lake/rearc-covid-19-nyt-data-in-usa/csv/us-states/us-
states.csv    2020-01-23  Washington  53  1   0
s3://covid19-lake/rearc-covid-19-nyt-data-in-usa/csv/us-states/us-
states.csv    2020-01-24  Illinois    17  1   0
s3://covid19-lake/rearc-covid-19-nyt-data-in-usa/csv/us-states/us-
states.csv    2020-01-24  Washington  53  1   0
```

You can use this %PATH field in your SQL queries as any other field.

ETL data to " Regular Tables "

If your task is to load data from S3 into an IRIS table – you can use the External Table as an ETL tool. Just do:

```
INSERT INTO internal_table SELECT * FROM external_table
```

In our case, if we want to copy COVID data from S3 into the local table:

```
--create local table
create table covid_by_state_local (
    "date" DATE,
    "state" VARCHAR(100),
    fips INT,
    cases INT,
    deaths INT
)
```

```
--ETL from External to Local table
INSERT INTO covid_by_state_local SELECT TO_DATE("date",'YYYY-MM-
DD'),state,fips,cases,deaths FROM covid_by_state
```

JOIN between IRIS – native and External table. Federated queries

External Table is an SQL table. It can be joined with other tables, used in subselects and UNIONs. You can even combine the “ Regular ” IRIS table and two or more external tables from different sources in the same SQL query.

Try creating a regular table such as matching state names to state codes like Washington – WA. And Join it with our S3-Based table.

```
create table state_codes (name varchar(100), code char(2))
insert into state_codes values ('Washington','WA')
insert into state_codes values ('Illinois','IL')

select top 10 "date", state, code, cases from covid_by_state join state_codes on stat
e=name
```

Change ‘ join ’ to ‘ left join ’ to include rows for which state code is not defined. As you can see - the result is a combination of data from S3 and your native IRIS table.

Secure access to data

AWS Covid Data Lake is public. Anyone can read data from it without any authentication or authorization. In real life you want access your data in secure way that would prevent strangers from peeking at your files. Full details of AWS Identity and Access Management (IAM) is outside of the scope of this article. But the minimum, you need to know is that you need at least AWS Account Access Key and Secret in order to access private data in your account. <https://docs.aws.amazon.com/general/latest/gr/aws-sec-cred-types.html#ac... >

AWS Uses Account Key/Secret authentication to sign requests. https://docs.aws.amazon.com/general/latest/gr/aws-sec-cred-types.html#ac...

If you are running IRIS External Table on EC2 instance, the recommended way of dealing with authentication is using EC2 Instance Roles https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon... IRIS External Table would be able to use permissions of that role. No extra setup required.

On a local/non EC2 instance you need to specify AWSACCESSKEYID and AWSSECRETACCESSKEY by either specifying environment variables or installing and configuring AWS CLI client.

```
export AWS_ACCESS_KEY_ID=AKIAEXAMPLEKEY
export AWS_SECRET_ACCESS_KEY=111222333abcdefghigklmnopqrst
```

Make sure that the environment variable is visible within your IRIS process. You can verify it by running:

```
USER>write $system.Util.GetEnviron("AWS_ACCESS_KEY_ID")
```

It should output the value of the key.

or install AWS CLI, following instruction here https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2-linux.html and run:

```
aws configure
```

External Table then will be able to read the credentials from aws cli config files. Your interactive shell and IRIS process might be running under different accounts - make sure you run aws configure under the same account as your IRIS process.

#Best Practices #Cloud #CSV #Interoperability #SQL #InterSystems IRIS
Check the related application on InterSystems Open Exchange