

```

-- RAW.CONFORMED.DATEDIMENSIONRAW.CONFORMED.DATEDIMENSION
-- create databases
create database if not exists analytics;
create database if not exists raw;

--create schemas
create schema if not exists raw.conformed;
create schema if not exists raw.PUBS;

-- warehouse
use warehouse transforming;

-- define file formats
create or replace file format RAW.PUBLIC.PARQUET
  TYPE = parquet
  REPLACE_INVALID_CHARACTERS = TRUE;

create or replace file format RAW.PUBLIC.JSONARRAY
  TYPE = json
  STRIP_OUTER_ARRAY = TRUE;

create or replace file format RAW.PUBLIC.JSON
  TYPE = json
  STRIP_OUTER_ARRAY = FALSE;

create or replace file format RAW.PUBLIC.CSVHEADER
  TYPE = 'csv'
  FIELD_DELIMITER = ','
  SKIP_HEADER=1;

create or replace file format RAW.PUBLIC.CSV
  TYPE = csv
  FIELD_DELIMITER = ','
  PARSE_HEADER = FALSE
  SKIP_HEADER = 0;

-- create stages
-- varying file formats
CREATE or replace STAGE RAW.PUBLIC.externalworld_files
  URL = 'azure://externalworld.blob.core.windows.net/files/';

-- these are all parquet file formats
CREATE or replace STAGE RAW.PUBLIC.externalworld_database

```

```
URL = 'azure://externalworld.blob.core.windows.net/database/'
FILE_FORMAT = RAW.PUBLIC.PARQUET ;
```

```
-- stage the date dimension
```

```
CREATE or REPLACE TABLE raw.conformed.datedimension (
  datekey int
  ,date date
  ,datetime timestamp
  ,year int
  ,quarter int
  ,quartername varchar(2)
  ,month int
  ,monthname varchar(3)
  ,day int
  ,dayofweek int
  ,dayname varchar(3)
  ,weekday varchar(1)
  ,weekofyear int
  ,dayofyear int
) AS
WITH CTE_MY_DATE AS (
  SELECT DATEADD(DAY, SEQ4(), '1970-01-01 00:00:00') AS MY_DATE
  FROM TABLE(GENERATOR(ROWCOUNT=>365*30))
)
SELECT
  REPLACE(TO_DATE(MY_DATE)::varchar, '-', '')::int as datekey,
  TO_DATE(MY_DATE) as date
  ,TO_TIMESTAMP(MY_DATE) as datetime
  ,YEAR(MY_DATE) as year
  ,QUARTER(MY_DATE) as quarter
  ,CONCAT('Q', QUARTER(MY_DATE)::varchar) as quartername
  ,MONTH(MY_DATE) as month
  ,MONTHNAME(MY_DATE) as monthname
  ,DAY(MY_DATE) as day
  ,DAYOFWEEK(MY_DATE) as dayofweek
  ,DAYNAME(MY_DATE) as dayname
  ,case when DAYOFWEEK(MY_DATE) between 1 and 5 then 'Y' else 'N' end as weekday
  ,WEEKOFYEAR(MY_DATE) as weekofyear
  ,DAYOFYEAR(MY_DATE) as dayofyear
  FROM CTE_MY_DATE
;
```

```
-- stage pubs discounts
```

```
create or replace table RAW.pubs.discounts
(
    discounttype varchar,
    stor_id varchar,
    lowqty smallint,
    higqty smallint,
    discount decimal
);
copy into RAW.pubs.discounts
    FROM '@RAW.PUBLIC.externalworld_database/pubs.discounts.parquet'
    MATCH_BY_COLUMN_NAME='CASE_INSENSITIVE';
```

```
-- stage pubs stores
create or replace table RAW.pubs.stores
(
    stor_id varchar,
    stor_name varchar,
    stor_address varchar,
    city varchar,
    state varchar,
    zip varchar
);
copy into RAW.pubs.stores
    FROM '@RAW.PUBLIC.externalworld_database/pubs.stores.parquet'
    MATCH_BY_COLUMN_NAME='CASE_INSENSITIVE';
```

```
-- stage pubs SALES
create or replace table RAW.pubs.sales
(
    stor_id varchar,
    ord_num varchar,
    ord_date varchar,
    qty smallint,
    payterms varchar,
    title_id varchar
);
copy into RAW.pubs.sales
    FROM '@RAW.PUBLIC.externalworld_database/pubs.sales.parquet'
    MATCH_BY_COLUMN_NAME='CASE_INSENSITIVE';
```

```
-- stage the PUBS.SALES.1995orders
```

```
COPY INTO RAW.PUBS.SALES
(stor_id, ord_num, ord_date, qty, payterms, title_id)
FROM '@RAW.PUBLIC.externalworld_files/pubs.1995orders.csv'
FILE_FORMAT = (TYPE = CSV SKIP_HEADER = 0);
```

```
-- stage the PUBS.SALES.1996orders
copy into RAW.PUBS.SALES
(stor_id,ord_num,ord_date,qty,payterms,title_id)
from '@RAW.PUBLIC.externalworld_files/pubs.1996orders.csv'
FILE_FORMAT = (TYPE = CSV SKIP_HEADER = 0);
```

```
-- stage pubs AUTHORS
create or replace table RAW.pubs.AUTHORS
```

```
(
  au_id varchar,
  au_lname varchar,
  au_fname varchar,
  phone varchar,
  address varchar,
  city varchar,
  state varchar,
  zip varchar,
  Contract boolean
);
copy into RAW.pubs.authors
FROM '@RAW.PUBLIC.externalworld_database/pubs.authors.parquet'
MATCH_BY_COLUMN_NAME='CASE_INSENSITIVE';
```

```
-- stage pubs TITLEAUTHOR
create or replace table RAW.pubs.titleauthor
(
  au_id varchar,
  title_id varchar,
  au_ord int,
  royaltypers int
);
copy into RAW.pubs.titleauthor
FROM '@RAW.PUBLIC.externalworld_database/pubs.titleauthor.parquet'
MATCH_BY_COLUMN_NAME='CASE_INSENSITIVE';
```

```

-- stage pubs TITLES
create or replace table RAW.pubs.titles
(
    title_id varchar,
    title varchar,
    type varchar,
    pub_id varchar,
    price decimal,
    advance decimal,
    royalty int,
    ytd_sales int,
    notes varchar,
    pubdate varchar
);
copy into RAW.pubs.titles
    FROM '@RAW.PUBLIC.externalworld_database/pubs.titles.parquet'
    MATCH_BY_COLUMN_NAME='CASE_INSENSITIVE';

-- stage pubs ROYSCHED
create or replace table RAW.pubs.roysched
(
    title_id varchar,
    lorange int,
    hirange int,
    royalty int
);
copy into RAW.pubs.roysched
    FROM '@RAW.PUBLIC.externalworld_database/pubs.roysched.parquet'
    MATCH_BY_COLUMN_NAME='CASE_INSENSITIVE';

-- stage pubs PUBLISHERS
create or replace table RAW.pubs.publishers
(
    pub_id varchar,
    pub_name varchar,
    city varchar,
    state varchar,
    country varchar
);
copy into RAW.pubs.publishers
    FROM '@RAW.PUBLIC.externalworld_database/pubs.publishers.parquet'
    MATCH_BY_COLUMN_NAME='CASE_INSENSITIVE';

```

```

-- stage pubs PUB_INFO
create or replace table RAW.pubs.pub_info
(
    pub_id varchar,
    logo varchar,
    pr_info varchar
);
copy into RAW.pubs.pub_info
    FROM '@RAW.PUBLIC.externalworld_database/pubs.pub_info.parquet'
    MATCH_BY_COLUMN_NAME='CASE_INSENSITIVE';

```

```

-- stage pubs EMPLOYEE
create or replace table RAW.pubs.employee
(
    emp_id varchar,
    fname varchar,
    minit varchar,
    lname varchar,
    job_id smallint,
    job_lvl int,
    pub_id varchar,
    hire_date varchar
);
copy into RAW.pubs.employee
    FROM '@RAW.PUBLIC.externalworld_database/pubs.employee.parquet'
    MATCH_BY_COLUMN_NAME='CASE_INSENSITIVE';

```

```

-- stage pubs JOBS
create or replace table RAW.pubs.jobs
(
    job_id smallint,
    job_desc varchar,
    min_lvl int,
    max_lvl int
);
copy into RAW.pubs.jobs
    FROM '@RAW.PUBLIC.externalworld_database/pubs.jobs.parquet'
    MATCH_BY_COLUMN_NAME='CASE_INSENSITIVE';

```

```

--
select TABLE_CATALOG, TABLE_SCHEMA, TABLE_NAME, ROW_COUNT

```

```
from RAW.INFORMATION_SCHEMA.TABLES  
where TABLE_SCHEMA='PUBS' order by TABLE_NAME;
```