Mount

```
# Mount the Azure DataLake  Storage container to Databricks filesystem
dbutils.fs.mount(
  source = "wasbs://bigdataproject@renujatoth.blob.core.windows.net",
  mount_point = "/mnt/bigdataproject",
  extra_configs = {"fs.azure.account.key.renujatoth.blob.core.windows.net":"+S3X0zq0epeHrZIUb6iZV1pcZPV6u9gxGdKSD3r6JhLqrFZukPko/MCSzhoA65PPWeVr6+2PI6Dq+AStO+WN/Q=="})
```

```
Out[2]: True
```

```
# Unmount the Azure Datalake container
dbutils.fs.unmount("/mnt/bigdataproject")
```

```
/mnt/bigdataproject has been unmounted.
Out[1]: True
```

```
%fs
ls /mnt/bigdataproject/bronze/raw/Source_Data
```

```
FileNotFoundException: /mnt/bigdataproject/bronze/raw/Source_Data
```

Creating a Database

```
%sql
--creating database
Create Database if not exists bigdata
```

OK

```
%sql
use bigdata;
```

OK

Bronze Table

```
%sql
----creating table from csv
drop table if exists bigdata.bronze_building_data_csv;
CREATE TABLE bigdata.bronze_building_data_csv
USING csv
OPTIONS(path='/mnt/bigdataproject/bronze/raw/Source_Data.csv',
header='true',
inferSchema='true'
);
```

OK

```
%sql
show tables
```

Table

	database	tableName	isTemporary
1	bigdata	bronze_building_data_csv	false

1 row

Verifying PK and filtering data where PK have duplicates

```
%sql
--Verifying PK and filtering data where PK have duplicates
select count(*),ID from bigdata.bronze_building_data_csv group by ID having count(*)=1
```

Table

	count(1)	ID
1	1	3247107
2	1	1832422
3	1	1893630

4	1	3104844
5	1	3033217
6	1	1707615
7	1	1885091

10,000 rows | Truncated data

Removing Unwanted Columns and trim for all the columns to make sure data is properly alligned

```
%sql
--select only required columns and add trim functions for all the columns to make sure data is properly alligned
DROP TABLE IF EXISTS bigdata.bronze_data_delta;

CREATE TABLE bigdata.bronze_data_delta
USING DELTA
OPTIONS (
  'path' = 'dbfs:/mnt/bigdataproject/bronze/raw/bronze_data_delta',
  'delimiter' = ' ',
  'header' = 'true',
  'inferSchema' = 'true'
)
AS
SELECT trim(id) AS id,
  trim(`permit#`) AS permit_number, -- Changed alias to avoid special characters in column name
  trim(permit_type) AS permit_type,
  trim(review_type) AS review_type,
  trim(application_start_date) AS application_start_date,
  trim(issue_date) AS issue_date,
  trim(processing_time) AS processing_time,
  trim(street_number) AS street_number,
  trim(`street direction`) AS street_direction, -- Enclosed column name with backticks
  trim(street_name) AS street_name,
  trim(work_description) AS work_description,
  trim(building_fee_paid) AS building_fee_paid,
  trim(zoning_fee_paid) AS zoning_fee_paid,
  trim(other_fee_paid) AS other_fee_paid,
  trim(subtotal_paid) AS subtotal_paid,
  trim(building_fee_unpaid) AS building_fee_unpaid,
  trim(zoning_fee_unpaid) AS zoning_fee_unpaid,
  trim(other_fee_unpaid) AS other_fee_unpaid,
  trim(subtotal_unpaid) AS subtotal_unpaid,
  trim(building_fee_waived) AS building_fee_waived,
  trim(zoning_fee_waived) AS zoning_fee_waived,
  trim(other_fee_waived) AS other_fee_waived,
  trim(subtotal_waived) AS subtotal_waived,
  trim(total_fee) AS total_fee,
  trim(contact_1_type) AS contact_type,
  trim(contact_1_name) AS contact_name,
  trim(contact_1_city) AS contact_city,
  trim(contact_1_state) AS contact_state,
  trim(contact_1_zipcode) AS contact_zipcode,
  trim(reported_cost) AS reported_cost,
  trim(pin1) AS pin,
  trim(community_area) AS community_area,
  trim(census_tract) AS census_tract,
  trim(ward) AS ward
from bronze_building_data_csv where ID in (select distinct ID from (select count(*),ID from bigdata.bronze_building_data_csv group by ID having count(*)=1))
```

Query returned no results

```
%sql
select count(*) from bigdata.bronze_data_delta
```

Table

	count(1)
1	753593

1 row

Silver Table

Full load for one time

```
%sql

DROP TABLE IF EXISTS bigdata.silver_data_delta;

CREATE TABLE bigdata.silver_data_delta
USING DELTA
OPTIONS (
  'path' = 'dbfs:/mnt/bigdataproject/silver/silver_data_delta',
  'delimiter' = ' ',
  'header' = 'true',
  'inferSchema' = 'true'
)
AS
```

```
AS
select id,
permit_number,
permit_type,
review_type,
application_start_date,
issue_date,
processing_time,
street_number,
street_direction,
street_name,
work_description,
building_fee_paid,
zoning_fee_paid,
other_fee_paid,
subtotal_paid,
building_fee_unpaid,
zoning_fee_unpaid,
other_fee_unpaid,
subtotal_unpaid,
building_fee_waived,
zoning_fee_waived,
other_fee_waived,
subtotal_waived,
total_fee,
contact_type,
contact_name,
contact_city,
contact_state,
contact_zipcode,
reported_cost,
pin,
community_area,
census_tract,
ward,
current_timestamp as efft_start_date,
to_timestamp('9999-12-31', 'yyyy-MM-dd') AS efft_end_date,
'Y' as is_active
from bigdata.bronze_data_delta
```

Query returned no results

```
%sql
select * from bigdata.silver_data_delta
```

Table

	id	permit_number	permit_type	review_type	application_start_date	issue_date	processing_time	street_number	street_direction	st
1	1830273	100190752	PERMIT - SIGNS	SIGN PERMIT	2007-10-16	2008-03-05	141	6349	S	CC
2	3056831	100832738	PERMIT - EASY PERMIT PROCESS	EASY PERMIT	2019-07-29	2019-07-29	0	248	W	CL
3	2060019	100296510	PERMIT - ELECTRIC WIRING	EASY PERMIT WEB	2009-06-15	2009-06-15	0	10000	N	BE
4	1671981	100108067	PERMIT - ELECTRIC WIRING	EASY PERMIT WEB	2006-04-17	2006-04-17	0	7535	N	HE
5	2832499	100696131	PERMIT - ELECTRIC WIRING	EASY PERMIT WEB	2017-03-20	2017-03-20	0	150	N	RI
6	3053734	100830913	PERMIT - ELECTRIC WIRING	EASY PERMIT WEB	2019-07-17	2019-07-17	0	4001	N	RA
7	1775627	100161330	PERMIT - RENOVATION/ALTERATION	STANDARD PLAN REVIEW	2007-06-12	2007-06-18	6	1431	N	N

4,290 rows | Truncated data

For Incremental Load

```
%sql
MERGE INTO bigdata.silver_data_delta AS target
USING bigdata.bronze_data_delta  AS source
ON target.ID = source.ID
WHEN MATCHED and
source.permit_number<>target.permit_number and
source.permit_type<>target.permit_type and
source.review_type<>target.review_type and
source.application_start_date<>target.application_start_date and
source.issue_date<>target.issue_date and
source.processing_time<>target.processing_time and
source.street_number<>target.street_number and
source.street_direction<>target.street_direction and
source.street_name<>target.street_name and
source.work_description<>target.work_description and
source.building_fee_paid<>target.building_fee_paid and
source.zoning_fee_paid<>target.zoning_fee_paid and
source.other_fee_paid<>target.other_fee_paid and
source.subtotal_paid<>target.subtotal_paid and
source.building_fee_unpaid<>target.building_fee_unpaid and
source.zoning_fee_unpaid<>target.zoning_fee_unpaid and
source.other_fee_unpaid<>target.other_fee_unpaid and
source.subtotal_unpaid<>target.subtotal_unpaid and
source.building_fee_waived<>target.building_fee_waived and
source.zoning_fee_waived<>target.zoning_fee_waived and
source.other_fee_waived<>target.other_fee_waived and
source.subtotal_waived<>target.subtotal_waived and
source.total_fee<>target.total_fee and
source.contact_type<>target.contact_type and
source.contact_name<>target.contact_name and
source.contact_city<>target.contact_city and
```

```
source.contact_state<>target.contact_state and
source.contact_zipcode<>target.contact_zipcode and
source.reported_cost<>target.reported_cost and
source.pin<>target.pin and
source.community_area<>target.community_area and
source.census_tract<>target.census_tract and
source.ward<>target.ward
THEN
  UPDATE SET
    target.efft_end_date = current_timestamp,
    target.is_active = 'N'

WHEN NOT MATCHED THEN
  INSERT (id,permit_number,
permit_type,
review_type,
application_start_date,
issue_date,
processing_time,
street_number,
street_direction,
street_name,
work_description,
building_fee_paid,
zoning_fee_paid,
other_fee_paid,
subtotal_paid,
building_fee_unpaid,
zoning_fee_unpaid,
other_fee_unpaid,
subtotal_unpaid,
building_fee_waived,
zoning_fee_waived,
other_fee_waived,
subtotal_waived,
total_fee,
contact_type,
contact_name,
contact_city,
contact_state,
contact_zipcode,
reported_cost,
pin,
community_area,
census_tract,
ward,efft_start_date,efft_end_date,is_active)
  VALUES (source.id,source.permit_number,
source.permit_type,
source.review_type,
source.application_start_date,
source.issue_date,
source.processing_time,
source.street_number,
source.street_direction,
source.street_name,
source.work_description,
source.building_fee_paid,
source.zoning_fee_paid,
source.other_fee_paid,
source.subtotal_paid,
source.building_fee_unpaid,
source.zoning_fee_unpaid,
source.other_fee_unpaid,
source.subtotal_unpaid,
source.building_fee_waived,
source.zoning_fee_waived,
source.other_fee_waived,
source.subtotal_waived,
source.total_fee,
source.contact_type,
source.contact_name,
source.contact_city,
source.contact_state,
source.contact_zipcode,
source.reported_cost,
source.pin,
source.community_area,
source.census_tract,
source.ward,
current_timestamp,to_timestamp('9999-12-31', 'yyyy-MM-dd') ,'Y');
```

Table

	num_affected_rows	num_updated_rows	num_deleted_rows	num_inserted_rows
1	0	0	0	0

1 row

Gold Layer

Creating Dimension Table

Dimension - Permit

Dimension - Permit

```
%sql
DROP TABLE IF EXISTS dim_permit;

CREATE TABLE dim_permit (
  permit_key BIGINT GENERATED BY DEFAULT AS IDENTITY,
  permit_type STRING
)
USING DELTA
OPTIONS (
  'header' = 'true',
  'inferSchema' = 'true'
)
LOCATION 'dbfs:/mnt/bigdataproject/gold/dim_permit';
```

OK

```
%sql
INSERT INTO dim_permit ( permit_type)
SELECT DISTINCT permit_type
FROM bigdata.silver_data_delta;
```

Table

	num_affected_rows	num_inserted_rows
1	11	11

1 row

```
%sql
select * from dim_permit
```

Table

	permit_key	permit_type
1	1	PERMIT - RENOVATION/ALTERATION
2	2	PERMIT - WRECKING/DEMOLITION
3	3	PERMIT - PORCH CONSTRUCTION
4	4	PERMIT - SIGNS
5	5	PERMIT - ELEVATOR EQUIPMENT
6	6	PERMIT - FOR EXTENSION OF PMT
7	7	PERMIT - EASY PERMIT PROCESS

11 rows

Dimension - Contact

```
%sql
drop table if exists dim_Contact;
create or replace table dim_Contact (
  contact_id bigint GENERATED BY DEFAULT AS IDENTITY (START WITH 1 INCREMENT BY 1)
  ,contact_type string
  ,contact_name string
  ,contact_city string
  ,contact_state string
  ,contact_zipcode string)
USING DELTA
OPTIONS (
  'header' = 'true',
  'inferSchema' = 'true')
LOCATION 'dbfs:/mnt/bigdataproject/gold/dim_Contact'
```

OK

```
%sql
insert into dim_Contact(contact_type ,contact_name,contact_city ,contact_state ,contact_zipcode)
select distinct contact_type ,contact_name,contact_city ,contact_state ,contact_zipcode from bigdata.silver_data_delta;
```

Table

	num_affected_rows	num_inserted_rows
1	160086	160086

1 row

```
%sql
select * from dim_Contact order by contact_id limit 5
```

Table

	contact_id	contact_type	contact_name	contact_city	contact_state	contact_zipcode
1	1	CONTRACTOR-ELECTRICAL	KUO CONSTRUCTION, INC.	CHICAGO	IL	60616-

2	2	CONTRACTOR-ELECTRICAL	ELITE ELECTRIC CO. INC.	MELROSE PARK	IL	60160
3	3	CONTRACTOR-GENERAL CONTRACTOR	ALPHA RESTORATION SERVICES INC	BLUE ISLAND	IL	60406-
4	4	OWNER	WRIGHTWOOD 1516 LLC	CHICAGO	IL	60647
5	5	CONTRACTOR-ELECTRICAL	GURTZ ELECTRIC COMPANY	ARLINGTON HEIGHTS	IL	60005

5 rows

Dimension - Location

```
%sql
drop table if exists dim_location;
create or replace table dim_location (
  location_id bigint GENERATED BY DEFAULT AS IDENTITY (START WITH 1 INCREMENT BY 1)
  ,street_number int
  ,street_direction string
  ,street_name string
  ,community_area string
  ,census_tract string
  ,ward string)
USING DELTA
OPTIONS (
  'header' = 'true',
  'inferSchema' = 'true')
LOCATION 'dbfs:/mnt/bigdataproject/gold/dim_location'
```

OK

```
%sql
insert into dim_location(street_number, street_direction, street_name, community_area, census_tract, ward)
select distinct street_number, street_direction, street_name, community_area, census_tract, ward
from bigdata.silver_data_delta;
```

Table

	num_affected_rows	num_inserted_rows
1	327770	327770

1 row

```
%sql
select * from dim_location  order by location_id limit 5
```

Table

	location_id	street_number	street_direction	street_name	community_area	census_tract	ward
1	1	1632	W	WRIGHTWOOD	7	70700	32
2	2	624	W	43RD	61	842600	11
3	3	3918	S	WELLS	37	3701	3
4	4	4319	N	RICHMOND	null	null	null
5	5	2233	N	KENMORE	7	71000	43

5 rows

Creating Fact Table

Fact - Permit

```
%sql
drop table if exists fact_permit;
create or replace table fact_permit (
permit_key string,
location_id string,
contact_id string,
id string,
permit_number string,
application_start_date string,
issue_date string,
processing_time string,
work_description string,
building_fee_paid string,
zoning_fee_paid string,
other_fee_paid string,
subtotal_paid string,
building_fee_unpaid string,
zoning_fee_unpaid string,
other_fee_unpaid string,
subtotal_unpaid string,
building_fee_waived string,
zoning_fee_waived string,
other_fee_waived string,
subtotal_waived string,
```

```
total_fee string
)
    USING DELTA
    LOCATION 'dbfs:/mnt/bigdataproject/gold/fact_permit';
```

OK

```
%sql
insert into fact_permit
select a.id,
a.permit_number,
a.application_start_date,
a.issue_date,
a.processing_time,
a.work_description,
a.building_fee_paid,
a.zoning_fee_paid,
a.other_fee_paid,
a.subtotal_paid,
a.building_fee_unpaid,
a.zoning_fee_unpaid,
a.other_fee_unpaid,
a.subtotal_unpaid,
a.building_fee_waived,
a.zoning_fee_waived,
a.other_fee_waived,
a.subtotal_waived,
a.total_fee,b.permit_key,
c.location_id,
d.contact_id from bigdata.silver_data_delta a
join dim_permit b on a.permit_type=b.permit_type
join dim_location c on a.street_number=c.street_number and
a.street_direction=c.street_direction and
a.street_name=c.street_name and
a.community_area=c.community_area and
a.census_tract=c.census_tract and
a.ward=c.ward
join dim_Contact d on a.contact_type=d.contact_type  and
a.contact_name=d.contact_name and
a.contact_city=d.contact_city  and
a.contact_state=d.contact_state  and
a.contact_zipcode=d.contact_zipcode
```

Table

	num_affected_rows	num_inserted_rows
1	633353	633353

1 row

Merging location

```
%sql
MERGE INTO bigdata.dim_location AS a
USING (select distinct street_number,street_direction,street_name ,community_area ,census_tract,ward from bigdata.silver_data_delta) AS d on
ifnull(d.street_number,'NA')=ifnull(a.street_number ,'NA') and
ifnull(d.street_direction,'NA') =ifnull(a.street_direction,'NA')  and
ifnull(d.street_name,'NA') =ifnull(a.street_name,'NA')   and
ifnull(d.community_area,'NA') =ifnull(a.community_area,'NA')   and
ifnull(d.census_tract,'NA') =ifnull(a.census_tract,'NA') and
ifnull(d.ward,'NA') =ifnull(a.ward,'NA')
WHEN NOT MATCHED THEN
  INSERT (street_number,street_direction,street_name,community_area ,census_tract,ward)
  VALUES (d.street_number,
d.street_direction,
d.street_name,
d.community_area,
d.census_tract,
d.ward);
```

Table

	num_affected_rows	num_updated_rows	num_deleted_rows	num_inserted_rows
1	0	0	0	0

1 row

Merging permit_details

```
%sql
MERGE INTO bigdata.dim_permit AS a
USING (select distinct permit_type from bigdata.silver_data_delta) AS d on
ifnull(d.permit_type,'NA')=ifnull(a.permit_type ,'NA')
WHEN NOT MATCHED THEN
  INSERT (permit_type)
  VALUES (d.permit_type);
```

Table

Table

	num_affected_rows	num_updated_rows	num_deleted_rows	num_inserted_rows
1	0	0	0	0

1 row

Merging Contact

```
%sql
MERGE INTO bigdata.dim_Contact AS a
USING (select distinct contact_type ,contact_name,contact_city ,contact_state ,contact_zipcode from bigdata.silver_data_delta) AS d on
ifnull(d.contact_type,'NA')=ifnull(a.contact_type ,'NA') and
ifnull(d.contact_name,'NA') =ifnull(a.contact_name,'NA')  and
ifnull(d.contact_city,'NA') =ifnull(a.contact_city,'NA')   and
ifnull(d.contact_state,'NA') =ifnull(a.contact_state,'NA')   and
ifnull(d.contact_zipcode,'NA') =ifnull(a.contact_zipcode,'NA')
WHEN NOT MATCHED THEN
  INSERT (contact_type,contact_name,contact_city,contact_state,contact_zipcode)
  VALUES (d.contact_type,
d.contact_name,
d.contact_city,
d.contact_state,
d.contact_zipcode);
```

Table

	num_affected_rows	num_updated_rows	num_deleted_rows	num_inserted_rows
1	0	0	0	0

1 row

Merging Fact - Permit

```
df = spark.sql("""select a.id,
a.permit_number,
a.application_start_date,
a.issue_date,
a.processing_time,
a.work_description,
a.building_fee_paid,
a.zoning_fee_paid,
a.other_fee_paid,
a.subtotal_paid,
a.building_fee_unpaid,
a.zoning_fee_unpaid,
a.other_fee_unpaid,
a.subtotal_unpaid,
a.building_fee_waived,
a.zoning_fee_waived,
a.other_fee_waived,
a.subtotal_waived,
a.total_fee,b.permit_key,
c.location_id,
d.contact_id from bigdata.silver_data_delta a
join dim_permit b on ifnull(a.permit_type,'NA')=ifnull(b.permit_type,'NA')
join dim_location c on ifnull(a.street_number,'NA')=ifnull(c.street_number,'NA') and
ifnull(a.street_direction,'NA')=ifnull(c.street_direction,'NA') and
ifnull(a.street_name,'NA')=ifnull(c.street_name,'NA') and
ifnull(a.community_area,'NA')=ifnull(c.community_area,'NA') and
ifnull(a.census_tract,'NA')=ifnull(c.census_tract,'NA') and
ifnull(a.ward,'NA')=ifnull(c.ward,'NA')
join dim_Contact d on ifnull(a.contact_type,'NA')=ifnull(d.contact_type,'NA')  and
ifnull(a.contact_name,'NA')=ifnull(d.contact_name,'NA') and
ifnull(a.contact_city,'NA')=ifnull(d.contact_city,'NA')  and
ifnull(a.contact_state,'NA')=ifnull(d.contact_state,'NA')  and
ifnull(a.contact_zipcode,'NA')=ifnull(d.contact_zipcode,'NA')""")
df.createOrReplaceTempView("df_tmp_view")
```

```
%sql
MERGE INTO fact_permit f
USING df_tmp_view t
ON ifnull(f.id,'NA') = ifnull(t.id,'NA')
WHEN NOT MATCHED THEN
  INSERT (id,
    permit_number,
    application_start_date,
    issue_date,
    processing_time,
    work_description,
    building_fee_paid,
    zoning_fee_paid,
    other_fee_paid,
    subtotal_paid,
    building_fee_unpaid,
    zoning_fee_unpaid,
    other_fee_unpaid,
```

```
    subtotal_unpaid,
    building_fee_waived,
    zoning_fee_waived,
    other_fee_waived,
    subtotal_waived,
    total_fee,permit_key,
    location_id,
    contact_id
  ) VALUES (t.id,
    t.permit_number,
t.application_start_date,
t.issue_date,
t.processing_time,
t.work_description,
t.building_fee_paid,
t.zoning_fee_paid,
t.other_fee_paid,
t.subtotal_paid,
t.building_fee_unpaid,
t.zoning_fee_unpaid,
t.other_fee_unpaid,
t.subtotal_unpaid,
t.building_fee_waived,
t.zoning_fee_waived,
t.other_fee_waived,
t.subtotal_waived,
t.total_fee,
t.permit_key,
t.location_id,
t.contact_id
  );
```

Table

	num_affected_rows	num_updated_rows	num_deleted_rows	num_inserted_rows
1	753593	0	0	753593

1 row

```
%sql
select * from  fact_permit
```

Table

	permit_key	location_id	contact_id	id	permit_number	application_start_date	issue_date	processing_time	work_description
1	1	312602	146516	3199221	100931497	2021-07-20	2021-07-26	6	"SPR 2019 CBRC: REPLACE EXISTING WIRELESS ANTENNAS. REMOVE (6) RRU(S) ANTENNA(S), (6) RRU(S), (2) BREAKOUT BOXES, (2) HCS 2.0HYBRID TRUNK CABL (6) ANTENNA(S), (3) RRU(S) & (3) TMA(S) TO BE RELOCATED; (6) 7/8""COAX CAB

1 row