

# Submission Document - Design a Data Warehouse for Reporting and OLAP

Please include all of the required screenshots and SQL queries in this document. Resize the images as necessary to ensure the texts are readable.

## A. Staging

1. Data architecture diagram showing all 8 files pointing to staging database to Operational Data Store (ODS) to Data Warehouse (DWH) to Reporting

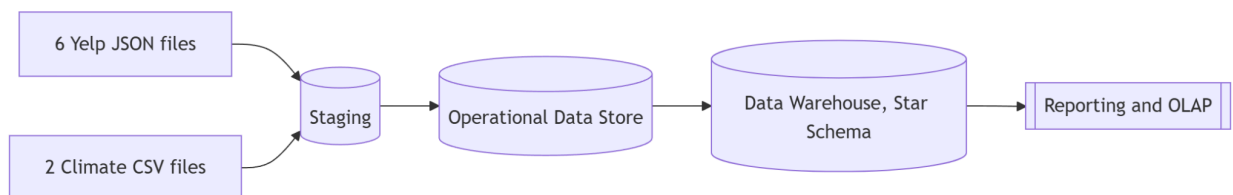


Figure 1. High-Level Abstraction of Architecture Pipeline

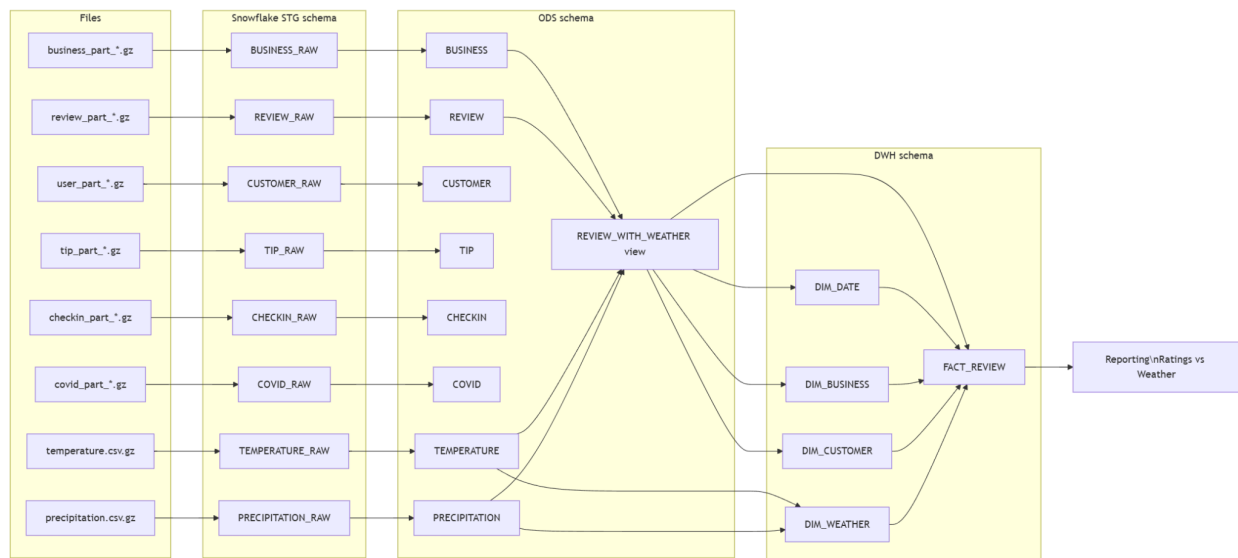


Figure 2. Medium-Level Abstraction of Architecture Pipeline

2. Screenshot showing the tables in the staging schema after extracting 6 Yelp files

```

+-----+-----+
| T           | N           |
+-----+-----+
| BUSINESS_RAW | 150346      |
| REVIEW_RAW   | 6990280     |
| CUSTOMER_RAW | 1230309     |
| CHECKIN_RAW  | 131930      |
| TIP_RAW      | 908915      |
| COVID_RAW    | 209795      |
+-----+-----+
6 Row(s) produced. Time Elapsed: 0.296s
Goodbye!

```

### 3. Screenshot showing the tables after extracting 2 files into the staging schema

*The screenshot should have two tables - temperature and precipitation.*

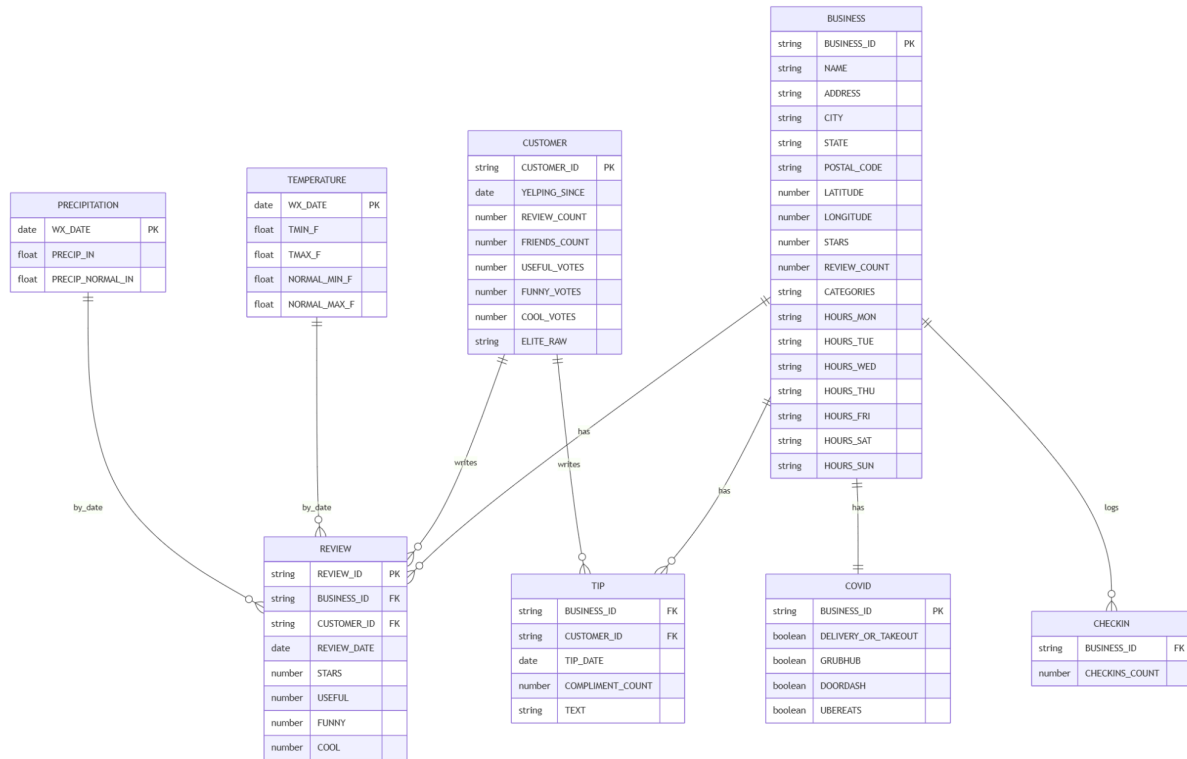
```

+-----+-----+
| T           | N           |
+-----+-----+
| TEMPERATURE_RAW | 28241      |
| PRECIPITATION_RAW | 28241      |
+-----+-----+
2 Row(s) produced. Time Elapsed: 0.202s
Goodbye!

```

## B. Operational Data Store (ODS)

1. ER diagram that includes one-to-one and one-to-many relationships for tables: Business, Customer, Tips, Review, Precipitation, Covid, Check\_in, Temperature



## 2. SQL queries that transform staging to ODS

USE ROLE ACCOUNTADMIN;

USE WAREHOUSE WH\_YELP\_XS;

USE DATABASE YELP\_WEATHER;

-- BUSINESS

INSERT OVERWRITE INTO ODS.BUSINESS

SELECT

```

t.V:business_id::STRING          AS BUSINESS_ID,
t.V:name::STRING                 AS NAME,
t.V:address::STRING              AS ADDRESS,
t.V:city::STRING                 AS CITY,
t.V:state::STRING                AS STATE,
t.V:postal_code::STRING          AS POSTAL_CODE,
TRY_TO_NUMBER(TO_VARCHAR(t.V:latitude)) AS LATITUDE,
TRY_TO_NUMBER(TO_VARCHAR(t.V:longitude)) AS LONGITUDE,
TRY_TO_NUMBER(TO_VARCHAR(t.V:stars)) AS STARS,
TRY_TO_NUMBER(TO_VARCHAR(t.V:review_count)) AS REVIEW_COUNT,
TRY_TO_NUMBER(TO_VARCHAR(t.V:is_open)) AS IS_OPEN,

```

```

t.V:categories::STRING          AS CATEGORIES,
t.V:hours:Monday::STRING    AS HOURS_MON,
t.V:hours:Tuesday::STRING   AS HOURS_TUE,
t.V:hours:Wednesday::STRING AS HOURS_WED,
t.V:hours:Thursday::STRING  AS HOURS_THU,
t.V:hours:Friday::STRING    AS HOURS_FRI,
t.V:hours:Saturday::STRING  AS HOURS_SAT,
t.V:hours:Sunday::STRING    AS HOURS_SUN,
t.V                          AS RAW
FROM STG.BUSINESS_RAW t;

```

-- CUSTOMER (Yelp "user")

INSERT OVERWRITE INTO ODS.CUSTOMER

SELECT

```

t.V:user_id::STRING          AS CUSTOMER_ID,
COALESCE(
  TRY_TO_DATE(TO_VARCHAR(t.V:yelping_since), 'YYYY-MM'),
  TRY_TO_DATE(TO_VARCHAR(t.V:yelping_since), 'YYYY-MM-DD'),
  TRY_TO_TIMESTAMP_NTZ(TO_VARCHAR(t.V:yelping_since))::DATE
)
  AS YELPING_SINCE,
TRY_TO_NUMBER(TO_VARCHAR(t.V:review_count))          AS REVIEW_COUNT,
ARRAY_SIZE(TRY_PARSE_JSON(t.V:friends))              AS FRIENDS_COUNT,
TRY_TO_NUMBER(TO_VARCHAR(t.V:useful))                AS USEFUL_VOTES,
TRY_TO_NUMBER(TO_VARCHAR(t.V:funny))                AS FUNNY_VOTES,
TRY_TO_NUMBER(TO_VARCHAR(t.V:cool))                 AS COOL_VOTES,
t.V:elite::STRING          AS ELITE_RAW,
t.V                        AS RAW
FROM STG.CUSTOMER_RAW t;

```

-- REVIEW

INSERT OVERWRITE INTO ODS.REVIEW

SELECT

```

t.V:review_id::STRING          AS REVIEW_ID,
t.V:business_id::STRING       AS BUSINESS_ID,
t.V:user_id::STRING           AS CUSTOMER_ID,
TRY_TO_NUMBER(TO_VARCHAR(t.V:stars))          AS STARS,
TRY_TO_NUMBER(TO_VARCHAR(t.V:useful))          AS USEFUL,
TRY_TO_NUMBER(TO_VARCHAR(t.V:funny))          AS FUNNY,
TRY_TO_NUMBER(TO_VARCHAR(t.V:cool))           AS COOL,

```

```

COALESCE(
  TRY_TO_DATE(TO_VARCHAR(t.V:date), 'YYYY-MM-DD'),
  TRY_TO_TIMESTAMP_NTZ(TO_VARCHAR(t.V:date))::DATE
)
          AS REVIEW_DATE,
t.V
          AS RAW
FROM STG.REVIEW_RAW t;

```

```

-- TIP
INSERT OVERWRITE INTO ODS.TIP
SELECT
  t.V:business_id::STRING          AS BUSINESS_ID,
  t.V:user_id::STRING             AS CUSTOMER_ID,
  COALESCE(
    TRY_TO_DATE(TO_VARCHAR(t.V:date), 'YYYY-MM-DD'),
    TRY_TO_TIMESTAMP_NTZ(TO_VARCHAR(t.V:date))::DATE
  )
    AS TIP_DATE,
  TRY_TO_NUMBER(TO_VARCHAR(t.V:compliment_count)) AS COMPLIMENT_COUNT,
  t.V:text::STRING                AS TEXT,
  t.V
    AS RAW
FROM STG.TIP_RAW t;

```

```

-- CHECKIN (counts of timestamps)
INSERT OVERWRITE INTO ODS.CHECKIN
SELECT
  t.V:business_id::STRING          AS BUSINESS_ID,
  IFF(t.V:date IS NULL, 0,
    ARRAY_SIZE(SPLIT(TO_VARCHAR(t.V:date), ','))) AS CHECKINS_COUNT,
  t.V
    AS RAW
FROM STG.CHECKIN_RAW t;

```

```

-- COVID flags from free-form strings
INSERT OVERWRITE INTO ODS.COVID
SELECT
  t.V:business_id::STRING AS BUSINESS_ID,
  IFF(UPPER(TO_VARCHAR(t.V:delivery_or_takeout)) IN ('TRUE','T','YES','Y','1'), TRUE, FALSE)
AS DELIVERY_OR_TAKEOUT,
  IFF(UPPER(TO_VARCHAR(t.V:Grubhub)) IN ('TRUE','T','YES','Y','1'), TRUE, FALSE) AS
GRUBHUB,

```

```

    IFF(UPPER(TO_VARCHAR(t.V:DoorDash))      IN ('TRUE','T','YES','Y','1'), TRUE, FALSE) AS
DOORDASH,
    IFF(UPPER(TO_VARCHAR(t.V:UberEats))      IN ('TRUE','T','YES','Y','1'), TRUE, FALSE) AS
UBEREATS,
    t.V                                     AS RAW
FROM STG.COVID_RAW t;

```

```

-- TEMPERATURE CSV to typed ODS table
INSERT OVERWRITE INTO ODS.TEMPERATURE
SELECT
    TO_DATE(date_raw, 'YYYYMMDD') AS WX_DATE,
    TMIN_F                        AS TMIN_F,
    TMAX_F                        AS TMAX_F,
    NORMAL_TMIN_F                AS NORMAL_MIN_F,
    NORMAL_TMAX_F                AS NORMAL_MAX_F
FROM STG.TEMPERATURE_RAW
WHERE DATE_RAW IS NOT NULL;

```

```

-- PRECIPITATION CSV to typed ODS table
INSERT OVERWRITE INTO ODS.PRECIPITATION
SELECT
    TO_DATE(date_raw, 'YYYYMMDD') AS WX_DATE,
    PRECIP_IN                     AS PRECIP_IN,
    PRECIP_NORMAL_IN             AS PRECIP_NORMAL_IN
FROM STG.PRECIPITATION_RAW
WHERE DATE_RAW IS NOT NULL;

```

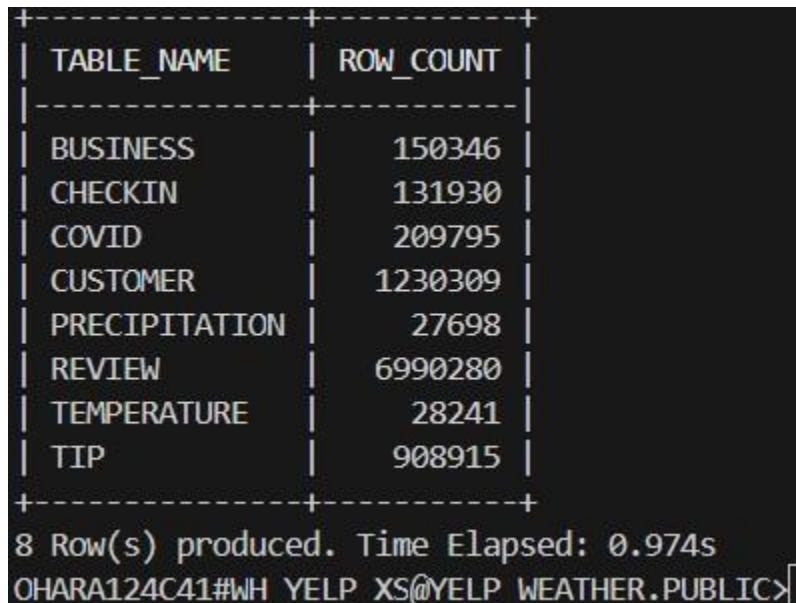
```

-- Evidence for ODS load
SELECT table_name, row_count
FROM INFORMATION_SCHEMA.TABLES
WHERE table_schema = 'ODS'
    AND table_name IN (

'BUSINESS','CUSTOMER','REVIEW','TIP','CHECKIN','COVID','TEMPERATURE','PRECIPITATION')
ORDER BY table_name;

```

3. Screenshot showing the queries were used successfully to transform the staging data to ODS



A screenshot of a SQL query result displayed in a terminal window. The result is a table with two columns: 'TABLE\_NAME' and 'ROW\_COUNT'. The table lists eight tables and their corresponding row counts. Below the table, it states '8 Row(s) produced. Time Elapsed: 0.974s'. The prompt 'OHARA124C41#WH YELP XS@YELP WEATHER.PUBLIC>' is visible at the bottom.

TABLE_NAME	ROW_COUNT
BUSINESS	150346
CHECKIN	131930
COVID	209795
CUSTOMER	1230309
PRECIPITATION	27698
REVIEW	6990280
TEMPERATURE	28241
TIP	908915

8 Row(s) produced. Time Elapsed: 0.974s  
OHARA124C41#WH YELP XS@YELP WEATHER.PUBLIC>

4. SQL queries that use JSON functions to transform staging data from a single JSON structure into multiple columns for ODS

```
SELECT
  V:business_id::string AS business_id,
  V:name::string      AS name,
  V:stars::number(3,2) AS stars
FROM STG.BUSINESS_RAW
LIMIT 3;
```

```
SELECT
  BUSINESS_ID,
  LATITUDE,
  LONGITUDE,
  HOURS_MON,
  RAW:hours AS raw_hours
FROM ODS.BUSINESS
LIMIT 3;
```

5. Screenshot showing the queries were used successfully to transform staging data from a single JSON structure into multiple columns for ODS

BUSINESS_ID	NAME	STARS
Pns2l4eNsf08kk83dixA6A	Abby Rappoport, LAC, CMQ	5.00
mpf3x-BjTdTEA3yCZrAYPw	The UPS Store	3.00
tUFrWirKiKi_TAnsVwINQQ	Target	3.50

3 Row(s) produced. Time Elapsed: 0.770s

BUSINESS_ID	LATITUDE	LONGITUDE	HOURS_MON	RAW_HOURS
Pns2l4eNsf08kk83dixA6A	34.000000	-120.000000	NULL	null
mpf3x-BjTdTEA3yCZrAYPw	39.000000	-90.000000	0:0-0:0	{ "Friday": "8:0-18:30", "Monday": "0:0-0:0", "Saturday": "8:0-14:0", "Thursday": "8:0-18:30", "Tuesday": "8:0-18:30", "Wednesday": "8:0-18:30" }
tUFrWirKiKi_TAnsVwINQQ	32.000000	-111.000000	8:0-22:0	{ "Friday": "8:0-23:0", "Monday": "8:0-22:0", "Saturday": "8:0-23:0", "Sunday": "8:0-22:0", "Thursday": "8:0-22:0", "Tuesday": "8:0-22:0", "Wednesday": "8:0-22:0" }

3 Row(s) produced. Time Elapsed: 0.818s  
OHARA124C41#WH YELP XS@YELP WEATHER.PUBLIC>

6. Screenshot showing different sizes/row\_counts of raw, staging, and ODS tables in database

TABLE_SCHEMA	TABLE_NAME	ROW_COUNT	BYTES
ODS	BUSINESS	150346	41465856
ODS	CHECKIN	131930	358139392
ODS	COVID	209795	17512448
ODS	CUSTOMER	1230309	4217672192
ODS	PRECIPITATION	27698	260096
ODS	REVIEW	6990280	7103526400
ODS	TEMPERATURE	28241	354304
ODS	TIP	908915	190510592
STG	BUSINESS_RAW	150346	11576320
STG	CHECKIN_RAW	131930	84423680
STG	COVID_RAW	209795	5270528
STG	CUSTOMER_RAW	1230309	1375520768
STG	PRECIPITATION_RAW	27698	165376
STG	REVIEW_RAW	6990280	2056817664
STG	TEMPERATURE_RAW	28241	222208
STG	TIP_RAW	908915	48301568

16 Row(s) produced. Time Elapsed: 2.958s

## 7. SQL queries that integrate the climate and Yelp datasets

```
USE ROLE ACCOUNTADMIN;
USE WAREHOUSE WH_YELP_XS;
USE DATABASE YELP_WEATHER;
USE SCHEMA ODS;
```

```
CREATE OR REPLACE VIEW ODS.REVIEW_WITH_WEATHER AS
SELECT
  r.REVIEW_ID,
  r.CUSTOMER_ID,
  r.BUSINESS_ID,
  b.NAME AS BUSINESS_NAME,
  b.CITY,
  b.STATE,
  r.REVIEW_DATE,
```

```
r.STARS,  
t.TMIN_F,  
t.TMAX_F,  
t.NORMAL_MIN_F,  
t.NORMAL_MAX_F,  
p.PRECIP_IN,  
p.PRECIP_NORMAL_IN  
FROM ODS.REVIEW r  
JOIN ODS.BUSINESS b USING (BUSINESS_ID)  
LEFT JOIN ODS.TEMPERATURE t ON t.WX_DATE = r.REVIEW_DATE  
LEFT JOIN ODS.PRECIPITATION p ON p.WX_DATE = r.REVIEW_DATE  
WHERE b.STATE IS NOT NULL;
```

```
-- Evidence for integration  
SELECT COUNT(*) AS REVIEW_WITH_WEATHER_ROWS FROM  
ODS.REVIEW_WITH_WEATHER;  
SELECT STATE, COUNT(*) AS REVIEW_ROWS  
FROM ODS.REVIEW_WITH_WEATHER  
GROUP BY STATE  
ORDER BY REVIEW_ROWS DESC  
LIMIT 10;
```

**8. Screenshot showing evidence that the SQL queries managed to integrate the datasets**

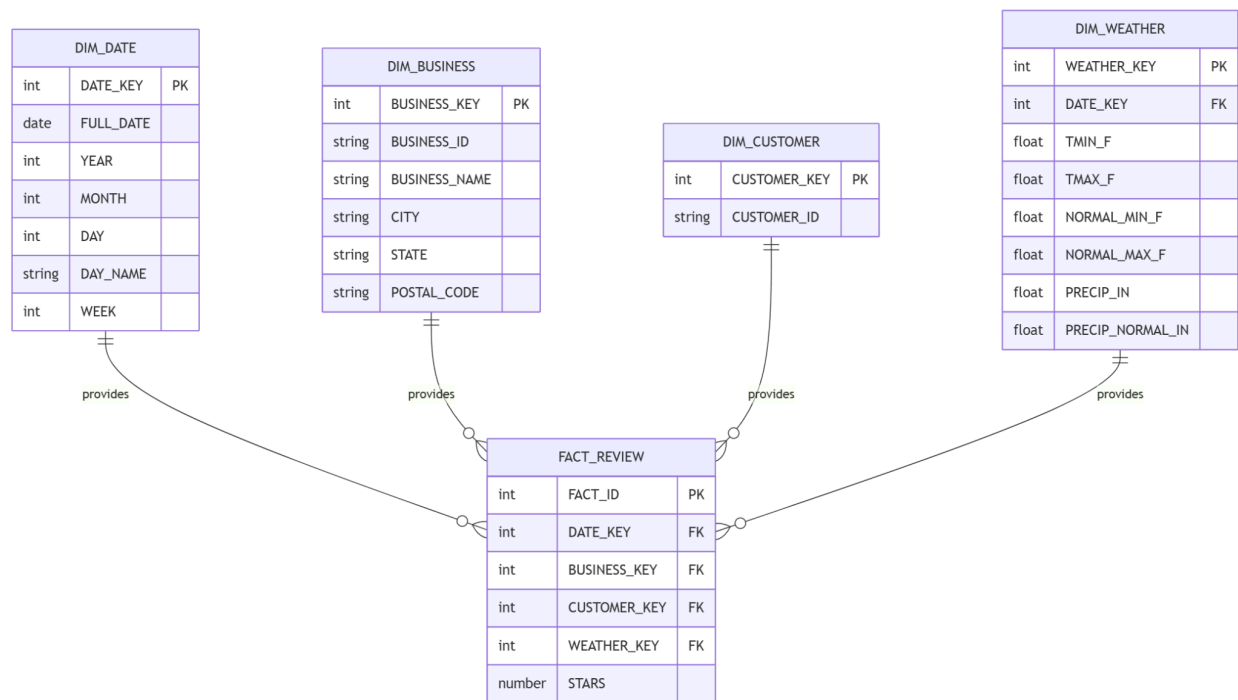
```

+-----+
| REVIEW_WITH_WEATHER_ROWS |
+-----+
| 6990280 |
+-----+
1 Row(s) produced. Time Elapsed: 2.476s
+-----+-----+
| STATE | REVIEW_ROWS |
+-----+-----+
| PA    | 1598960     |
| FL    | 1161545     |
| LA    | 761673      |
| TN    | 614388      |
| MO    | 502385      |
| IN    | 489752      |
| AZ    | 431708      |
| NV    | 430678      |
| CA    | 348856      |
| NJ    | 260897      |
+-----+-----+
10 Row(s) produced. Time Elapsed: 1.229s
OHARA124C41#WH_YELP_XS@YELP_WEATHER.ODS>

```

## C. Data Warehouse (DWH)

1. Diagram of star schema with several dimensions and a fact table that connects dimensions



## 2. SQL queries that moves the data from ODS to DWH

```
USE ROLE ACCOUNTADMIN;
USE WAREHOUSE WH_YELP_XS;
USE DATABASE YELP_WEATHER;
USE SCHEMA DWH;
```

```
-- Date dimension from distinct review dates
```

```
CREATE OR REPLACE TABLE DIM_DATE (
  DATE_KEY INT PRIMARY KEY,
  FULL_DATE DATE,
  YEAR INT,
  MONTH INT,
  DAY INT,
  DAY_NAME STRING,
  WEEK INT
);
```

```
INSERT OVERWRITE INTO DIM_DATE
SELECT
```

```
TO_NUMBER(TO_CHAR(REVIEW_DATE, 'YYYYMMDD')) AS DATE_KEY,  
REVIEW_DATE,  
YEAR(REVIEW_DATE),  
MONTH(REVIEW_DATE),  
DAY(REVIEW_DATE),  
DAYNAME(REVIEW_DATE),  
WEEK(REVIEW_DATE)  
FROM (SELECT DISTINCT REVIEW_DATE FROM ODS.REVIEW_WITH_WEATHER)  
WHERE REVIEW_DATE IS NOT NULL  
ORDER BY DATE_KEY;
```

-- Business dimension

```
CREATE OR REPLACE TABLE DIM_BUSINESS (  
  BUSINESS_KEY INT AUTOINCREMENT PRIMARY KEY,  
  BUSINESS_ID STRING UNIQUE,  
  BUSINESS_NAME STRING,  
  CITY STRING,  
  STATE STRING  
);
```

```
INSERT OVERWRITE INTO DIM_BUSINESS (BUSINESS_ID, BUSINESS_NAME, CITY, STATE)  
SELECT DISTINCT BUSINESS_ID, BUSINESS_NAME, CITY, STATE  
FROM ODS.REVIEW_WITH_WEATHER;
```

-- Customer dimension

```
CREATE OR REPLACE TABLE DIM_CUSTOMER (  
  CUSTOMER_KEY INT AUTOINCREMENT PRIMARY KEY,  
  CUSTOMER_ID STRING UNIQUE  
);
```

```
INSERT OVERWRITE INTO DIM_CUSTOMER (CUSTOMER_ID)  
SELECT DISTINCT CUSTOMER_ID  
FROM ODS.REVIEW_WITH_WEATHER;
```

-- Weather dimension (daily weather facts)

```
CREATE OR REPLACE TABLE DIM_WEATHER (  
  WEATHER_KEY INT AUTOINCREMENT PRIMARY KEY,  
  DATE_KEY INT,  
  TMIN_F FLOAT,
```

```
TMAX_F FLOAT,  
NORMAL_MIN_F FLOAT,  
NORMAL_MAX_F FLOAT,  
PRECIP_IN FLOAT,  
PRECIP_NORMAL_IN FLOAT  
);
```

```
INSERT OVERWRITE INTO DIM_WEATHER (DATE_KEY, TMIN_F, TMAX_F, NORMAL_MIN_F,  
NORMAL_MAX_F, PRECIP_IN, PRECIP_NORMAL_IN)  
SELECT DISTINCT  
TO_NUMBER(TO_CHAR(COALESCE(t.WX_DATE, p.WX_DATE), 'YYYYMMDD')) AS DATE_KEY,  
t.TMIN_F,  
t.TMAX_F,  
t.NORMAL_MIN_F,  
t.NORMAL_MAX_F,  
p.PRECIP_IN,  
p.PRECIP_NORMAL_IN  
FROM ODS.TEMPERATURE t  
FULL OUTER JOIN ODS.PRECIPITATION p  
ON p.WX_DATE = t.WX_DATE  
WHERE COALESCE(t.WX_DATE, p.WX_DATE) IS NOT NULL;
```

```
-- Fact table keyed by dimensions  
CREATE OR REPLACE TABLE FACT_REVIEW (  
FACT_ID INT AUTOINCREMENT PRIMARY KEY,  
DATE_KEY INT,  
BUSINESS_KEY INT,  
CUSTOMER_KEY INT,  
WEATHER_KEY INT,  
STARS NUMBER(2,1)  
);
```

```
INSERT OVERWRITE INTO FACT_REVIEW (DATE_KEY, BUSINESS_KEY, CUSTOMER_KEY,  
WEATHER_KEY, STARS)  
SELECT  
d.DATE_KEY,  
b.BUSINESS_KEY,  
c.CUSTOMER_KEY,  
w.WEATHER_KEY,
```

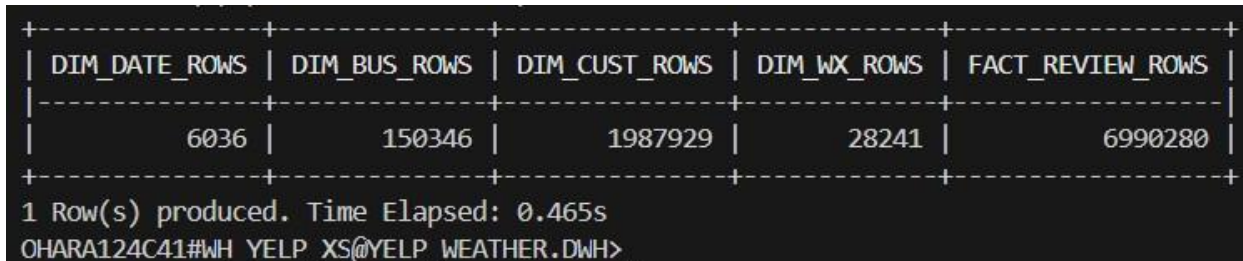
```

r.STARS
FROM ODS.REVIEW_WITH_WEATHER r
JOIN DIM_DATE d ON d.FULL_DATE = r.REVIEW_DATE
JOIN DIM_BUSINESS b ON b.BUSINESS_ID = r.BUSINESS_ID
JOIN DIM_CUSTOMER c ON c.CUSTOMER_ID = r.CUSTOMER_ID
LEFT JOIN DIM_WEATHER w ON w.DATE_KEY = d.DATE_KEY;

-- Evidence for DWH load
SELECT
  (SELECT COUNT(*) FROM DIM_DATE) AS DIM_DATE_ROWS,
  (SELECT COUNT(*) FROM DIM_BUSINESS) AS DIM_BUS_ROWS,
  (SELECT COUNT(*) FROM DIM_CUSTOMER) AS DIM_CUST_ROWS,
  (SELECT COUNT(*) FROM DIM_WEATHER) AS DIM_WX_ROWS,
  (SELECT COUNT(*) FROM FACT_REVIEW) AS FACT_REVIEW_ROWS;

```

### 3. Screenshot showing evidence that the SQL queries managed to move the data from ODS to DWH



DIM_DATE_ROWS	DIM_BUS_ROWS	DIM_CUST_ROWS	DIM_WX_ROWS	FACT_REVIEW_ROWS
6036	150346	1987929	28241	6990280

1 Row(s) produced. Time Elapsed: 0.465s  
 OHARA124C41#WH\_YELP\_XS@YELP\_WEATHER.DWH>

### 4. SQL queries that produce a report showing the business name, temperature, precipitation, and ratings

```

USE ROLE ACCOUNTADMIN;
USE WAREHOUSE WH_YELP_XS;
USE DATABASE YELP_WEATHER;
USE SCHEMA DWH;

SELECT
  b.BUSINESS_NAME,
  d.FULL_DATE AS REVIEW_DATE,
  COALESCE(w.TMIN_F, w.NORMAL_MIN_F) AS TEMP_MIN_F,

```

```
COALESCE(w.TMAX_F, w.NORMAL_MAX_F) AS TEMP_MAX_F,  
w.NORMAL_MIN_F AS NORMAL_MIN_F,  
w.NORMAL_MAX_F AS NORMAL_MAX_F,  
COALESCE(w.PRECIP_IN, w.PRECIP_NORMAL_IN) AS PRECIP_IN,  
w.PRECIP_NORMAL_IN AS PRECIP_NORMAL_IN,  
f.STARS AS RATING  
FROM FACT_REVIEW f  
JOIN DIM_BUSINESS b ON f.BUSINESS_KEY = b.BUSINESS_KEY  
JOIN DIM_DATE d ON f.DATE_KEY = d.DATE_KEY  
LEFT JOIN DIM_WEATHER w ON f.WEATHER_KEY = w.WEATHER_KEY  
ORDER BY REVIEW_DATE DESC, BUSINESS_NAME  
LIMIT 50;
```

**5. Screenshot showing the report produced by the SQL queries above**

BUSINESS_NAME	REVIEW_DATE	TEMP_MIN_F	TEMP_MAX_F	NORMAL_MIN_F	NORMAL_MAX_F	PRECIP_IN	PRECIP_NORMAL_IN	RATING
101 Taiwanese Cuisine	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	5.0
1200 Chophouse	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	5.0
14 Days to Close	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	5.0
2022 Boba and Badges with Ni Hao Tea	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	5.0
2022 Boba and Badges with Ni Hao Tea	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	5.0
3 J's Plumbing Service	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	5.0
3 Rivers Solid Surface & Remodeling	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	1.0
317 Burger	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	4.0
51st Deli	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	5.0
7-Eleven	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	1.0
717 Parking Enterprises	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	1.0
717 Parking Enterprises	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	1.0
86 West	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	1.0
99 Cents Only Stores	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	4.0
9th Street Bottle Shop	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	4.0
A La Mousse	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	5.0
A Radiant Property Inspection	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	5.0
AA Animal ER Center	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	1.0
ABC Pizza House	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	4.0
ACS Irrigation	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	4.0
AFC URGENT CARE SEMINOLE	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	1.0
AMC Dine-In Thoroughbred 20	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	1.0
AMC Plymouth Meeting Mall 12	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	5.0
AMC Sundial 12	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	3.0
ASIE	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	5.0
AT&T Store	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	1.0
AT&T Store	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	1.0
Acree Plumbing, Air & Electric	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	1.0
Active Life Chiropractic	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	5.0
Aesthetics Med Spa	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	5.0
Affordable Chimney Sweep	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	5.0
Affordable Dentures & Implants- Tampa	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	1.0
Alamo Rent A Car	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	1.0
Alamodak Restaurant and Hookah Lounge	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	5.0
Alan Reed	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	5.0
Alaska Airlines	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	1.0
Alaska Airlines	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	1.0
Alcazar Tapas Bar	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	5.0
Alessia Patisserie & Cafe	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	5.0
Alessia Patisserie & Cafe	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	5.0
Alessia Patisserie & Cafe	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	3.0
Alessia Patisserie & Cafe	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	5.0
Alessia Patisserie & Cafe	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	5.0
Alessia Patisserie & Cafe	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	5.0
All Hours Air	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	4.0
All Hours Air	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	5.0
All Terrain Landscape Creations	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	5.0
Allan Domb Real Estate	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	5.0
Allan Domb Real Estate	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	5.0
Alma Cafe	2022-01-19	58.3	39.5	58.3	39.5	0.35	0.35	5.0

50 Row(s) produced. Time Elapsed: 2.163s  
CHARA124C41#H1 YELP\_XS@YELP\_WEATHER.DWH>