Data Management Final Project

Group 6

Yue Jiang (yj4747), Shuheng Ma(sm64444), Yikang Wang(yw22279)

Yiqun Tian (yt5965), Rongzhi Xu (rx575)

Table of Contents

Introduction to airbnb

Airbnb is a vacation rental online marketplace company - offering millions of places to stay, all powered by local hosts.

Airbnb has helped hosts delight guests by providing convenient location, unique travel experiences, and immersion in local communities while keeping the financial benefits of tourism with the people who make it happen.

Data Management Strategy

Data defensive and offensive strategy are differentiated by a company's goal and core activities.

- For defensive strategy, it focuses on minimizing downside risk, using analytics to detect and limit fraud and building systems to prevent theft.
- Offensive strategy is about supporting business operations, generating profits and improving customer satisfactions.

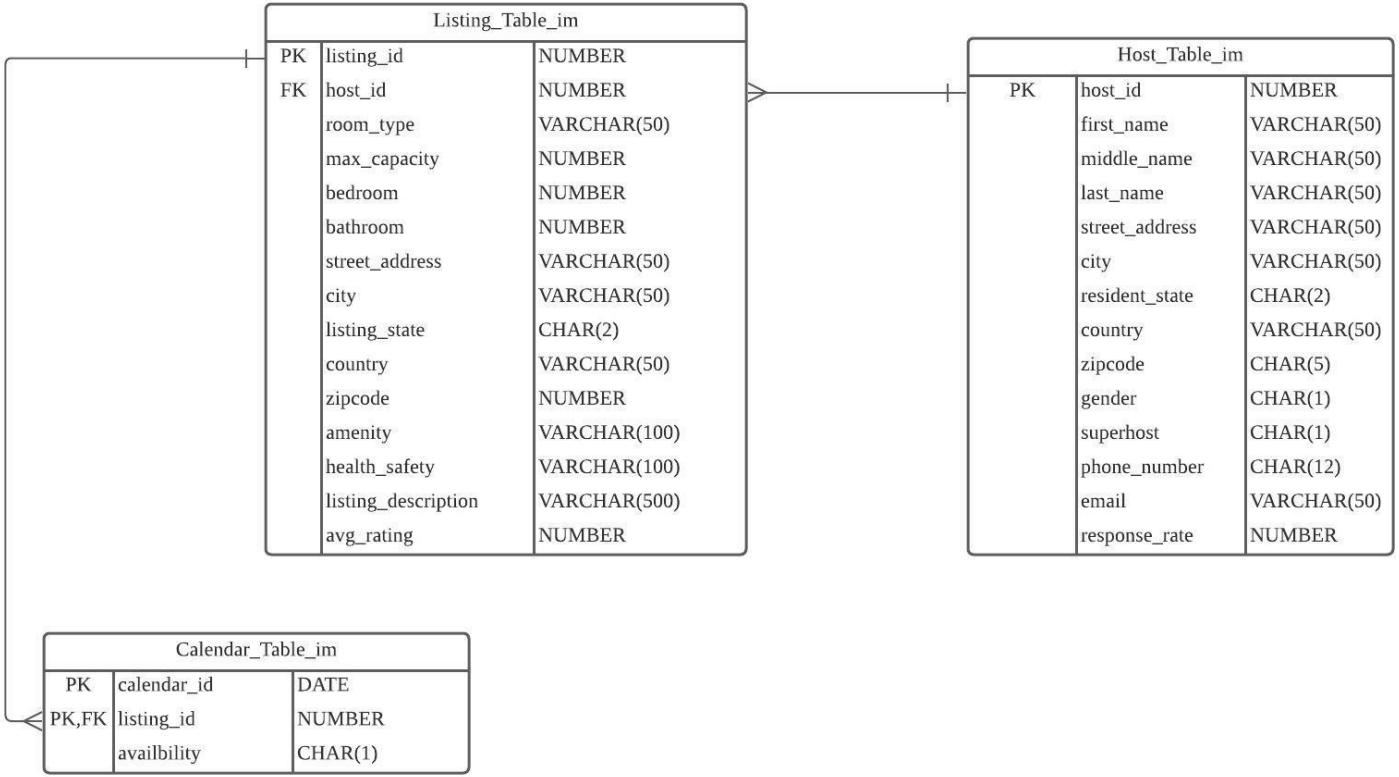
We believe Airbnb needs to devote equal attention to both defensive and offensive strategy to succeed.

- With millions of hosts' and customers' private information including, name, address, phone, credit card information etc., Airbnb is required to have a strong data **defense strategy.**
- However, Airbnb also operates in a dynamic market and requires to react quickly to market changes. It should also focus on a **offensive strategy** to react rapidly to competition and market changes.

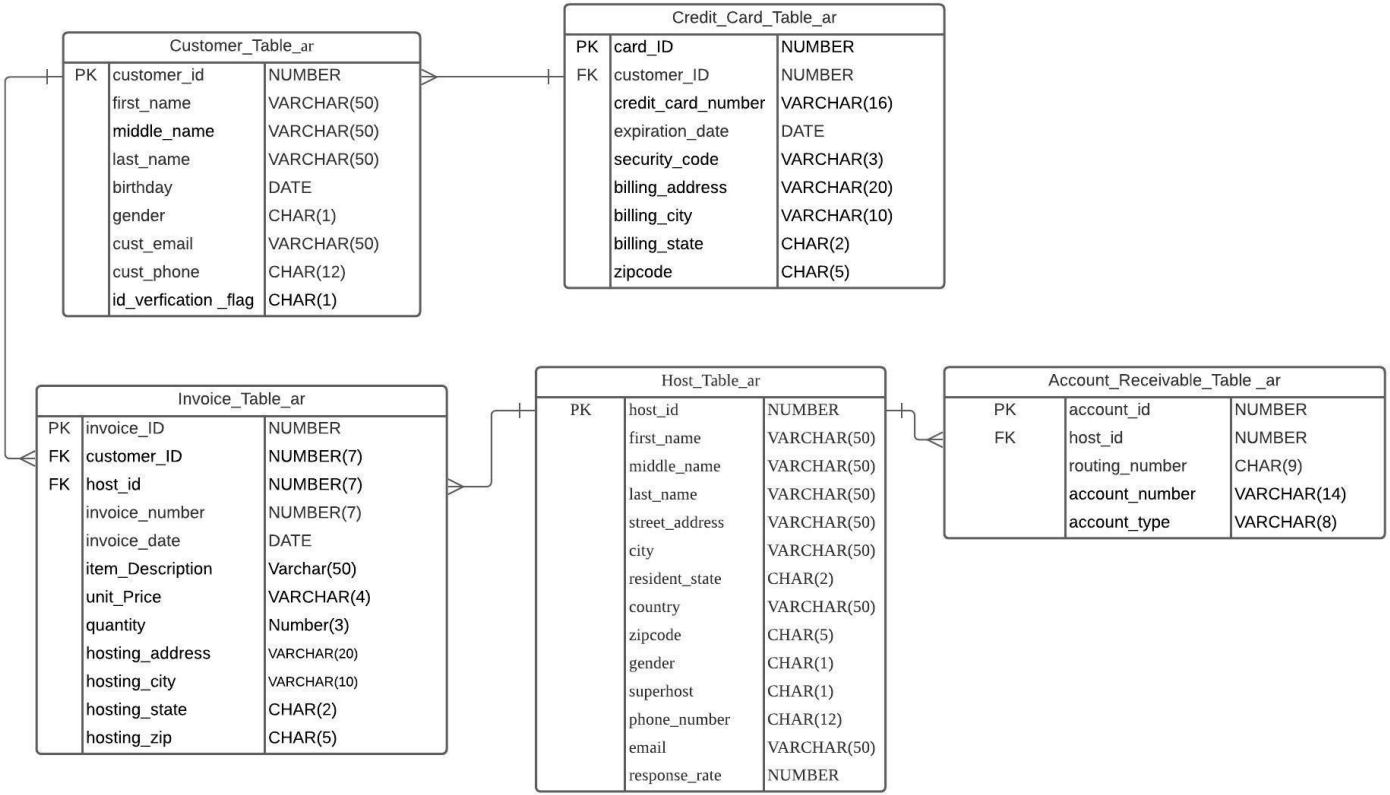
Transactional Database

- Inventory Management
- Customer Reservation
- Account Receivable

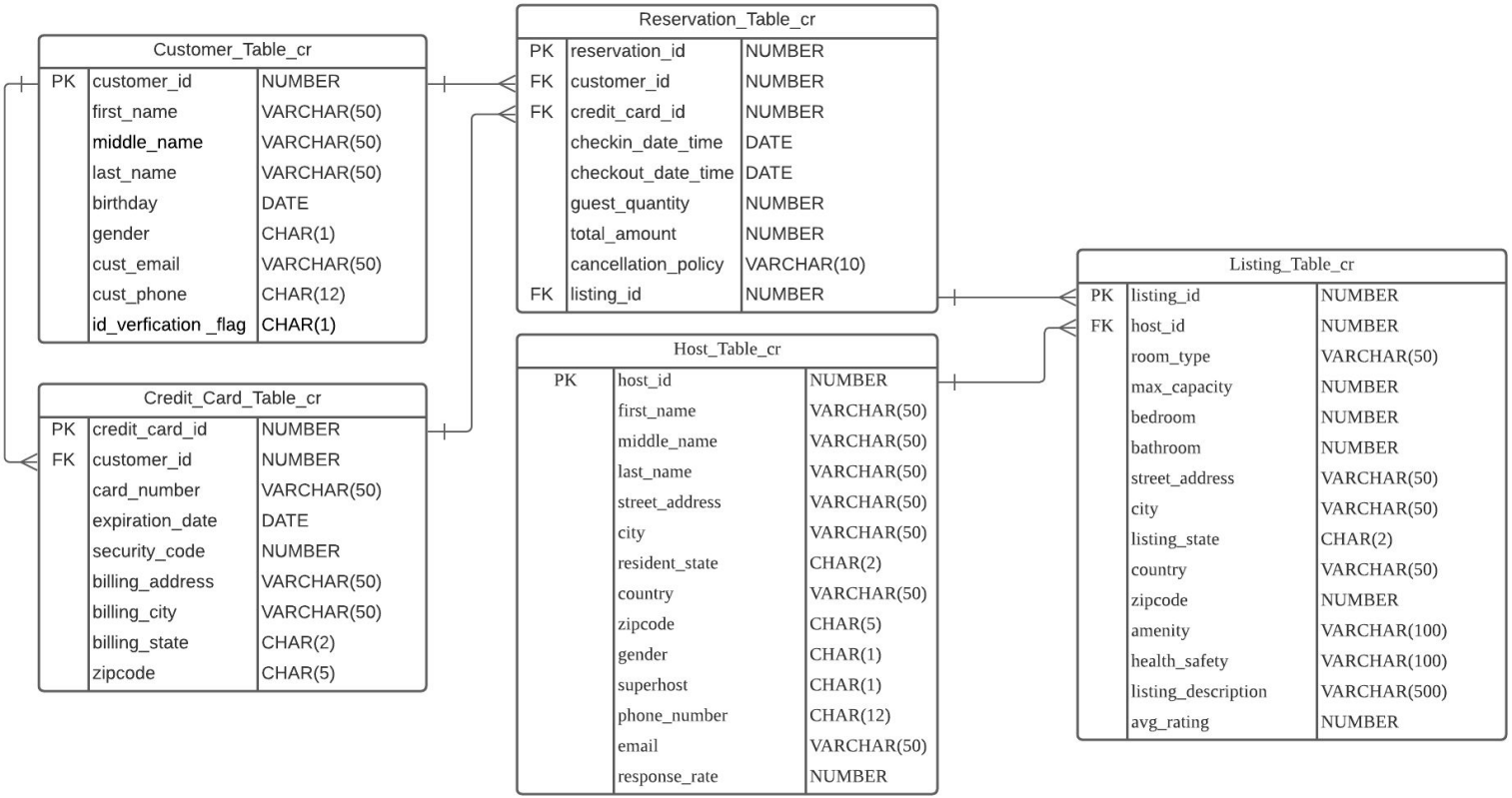
Transactional Database - Inventory Management



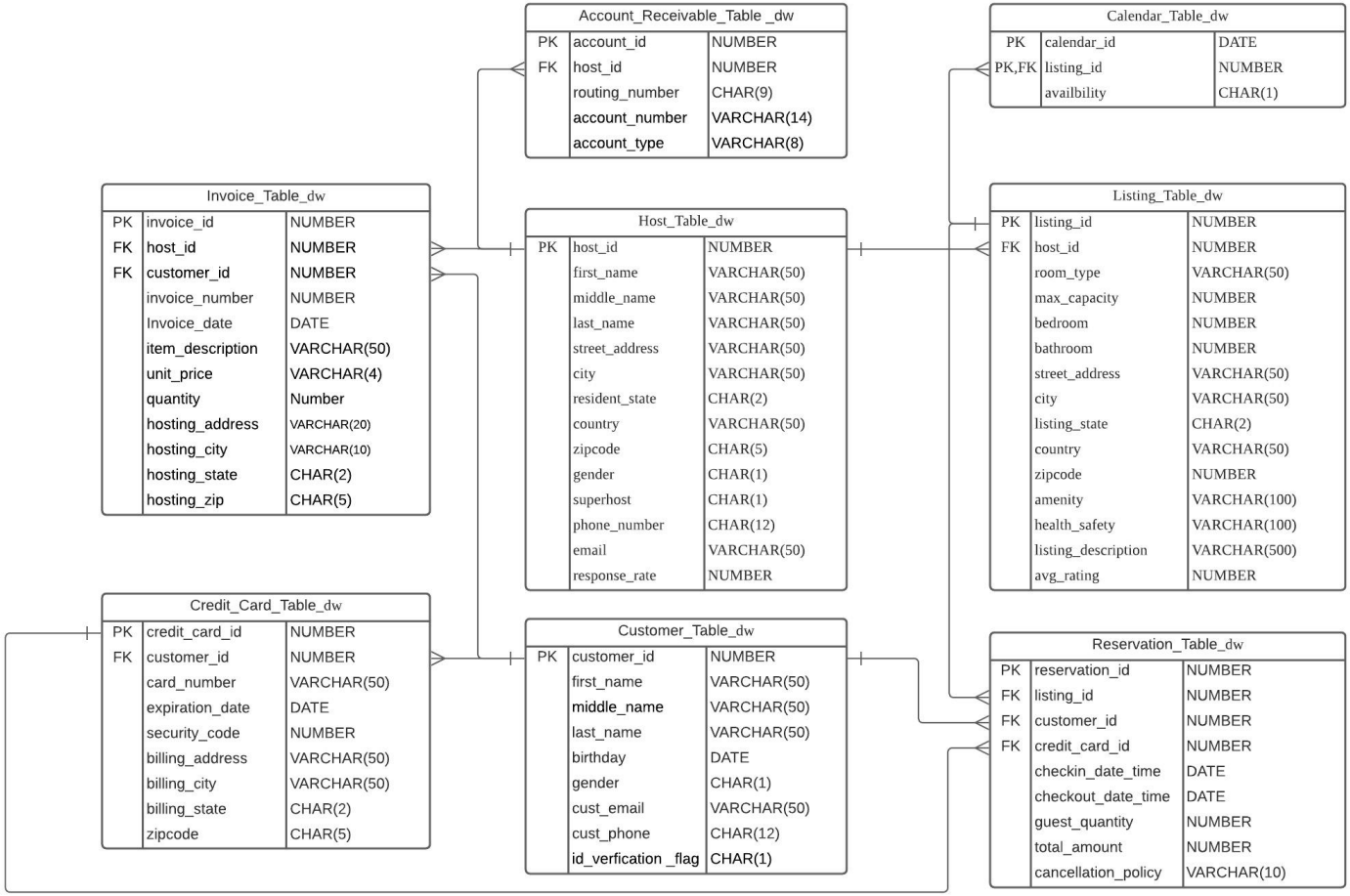
Transactional Database - Account Receivable



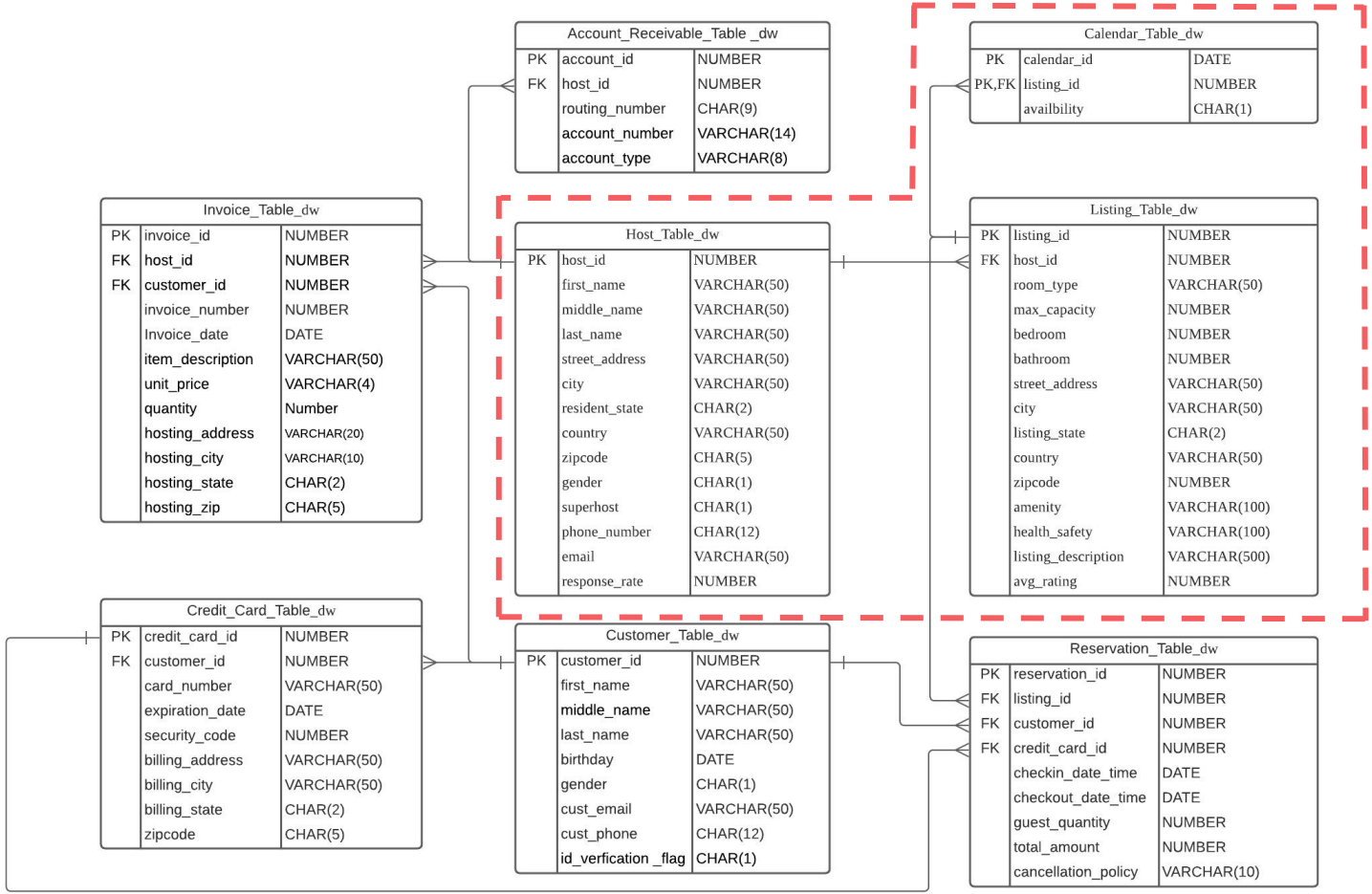
Transactional Database - Customer Reservation



Enterprise Data Warehouse



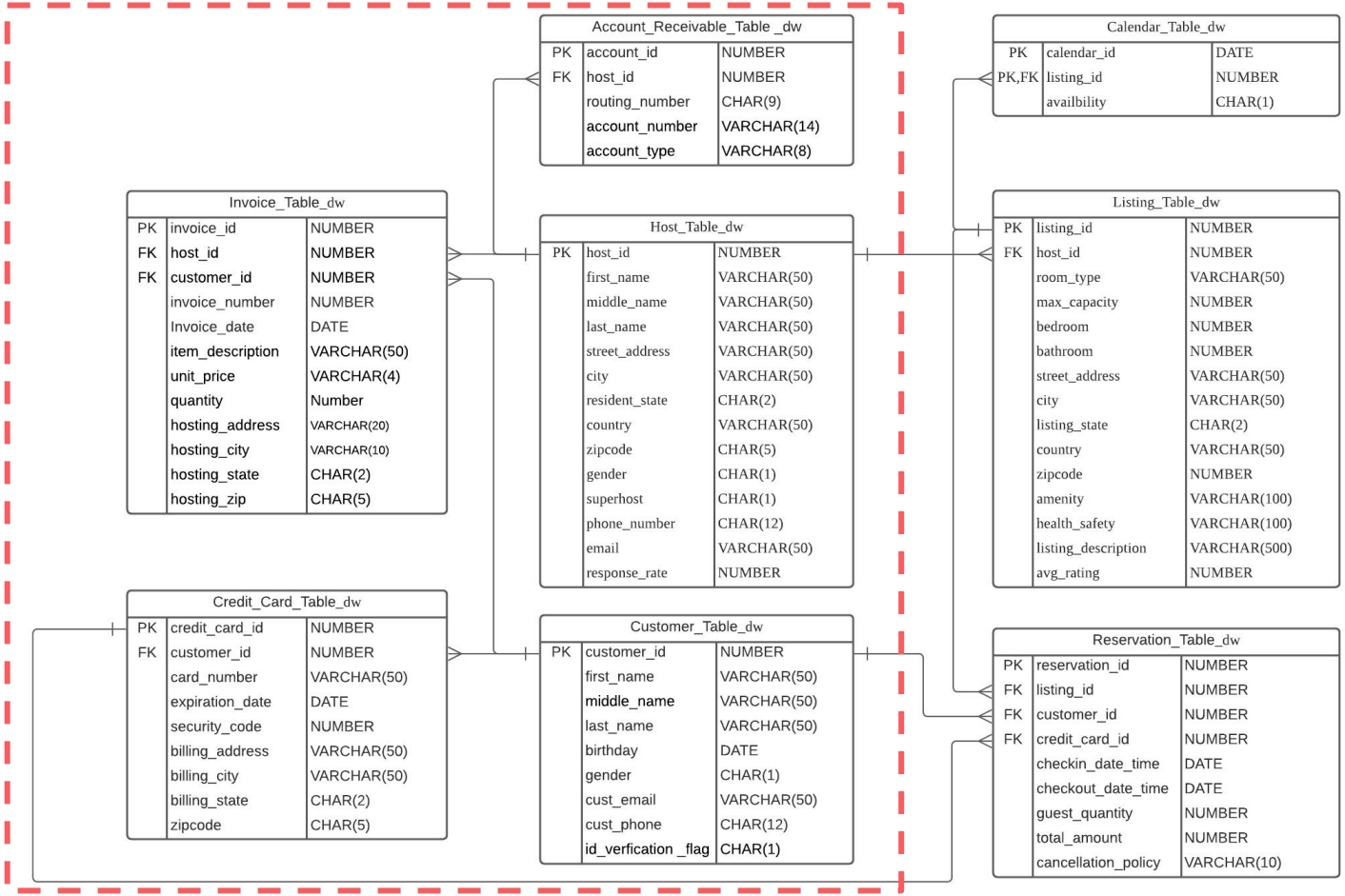
Enterprise Data Warehouse



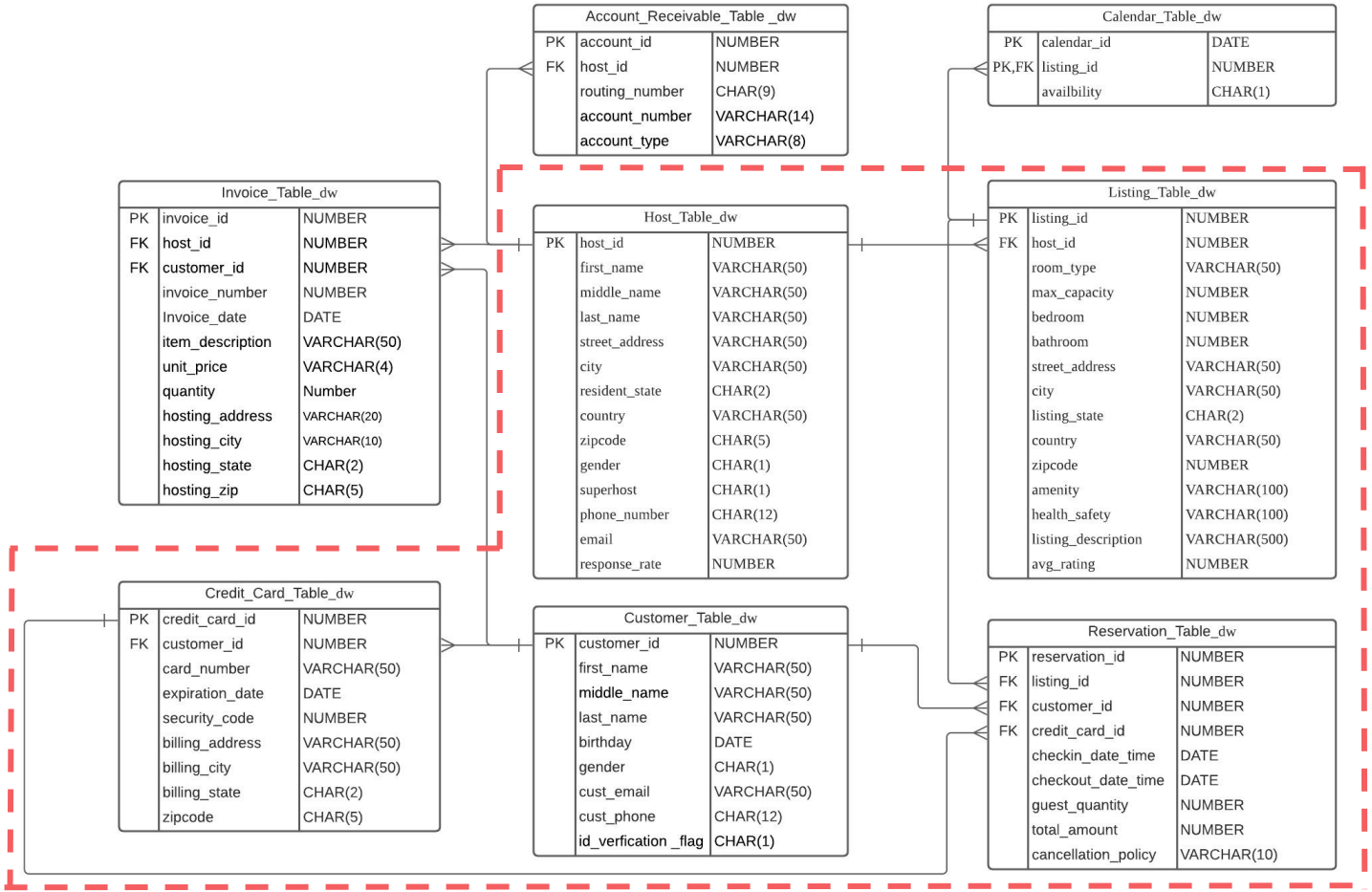
Inventory Management

Enterprise Data Warehouse

Account Receivable



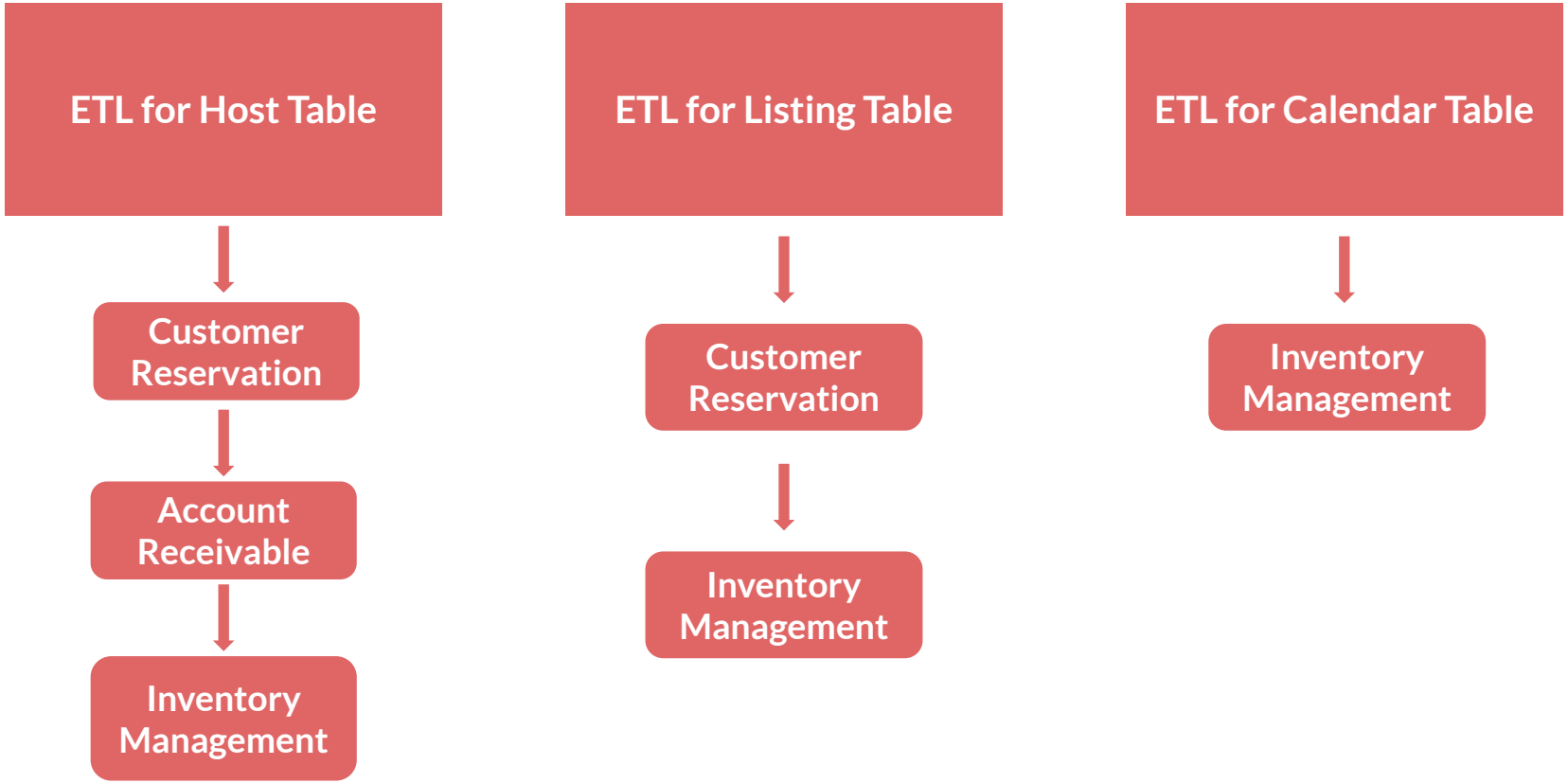
Enterprise Data Warehouse



Customer Receivation

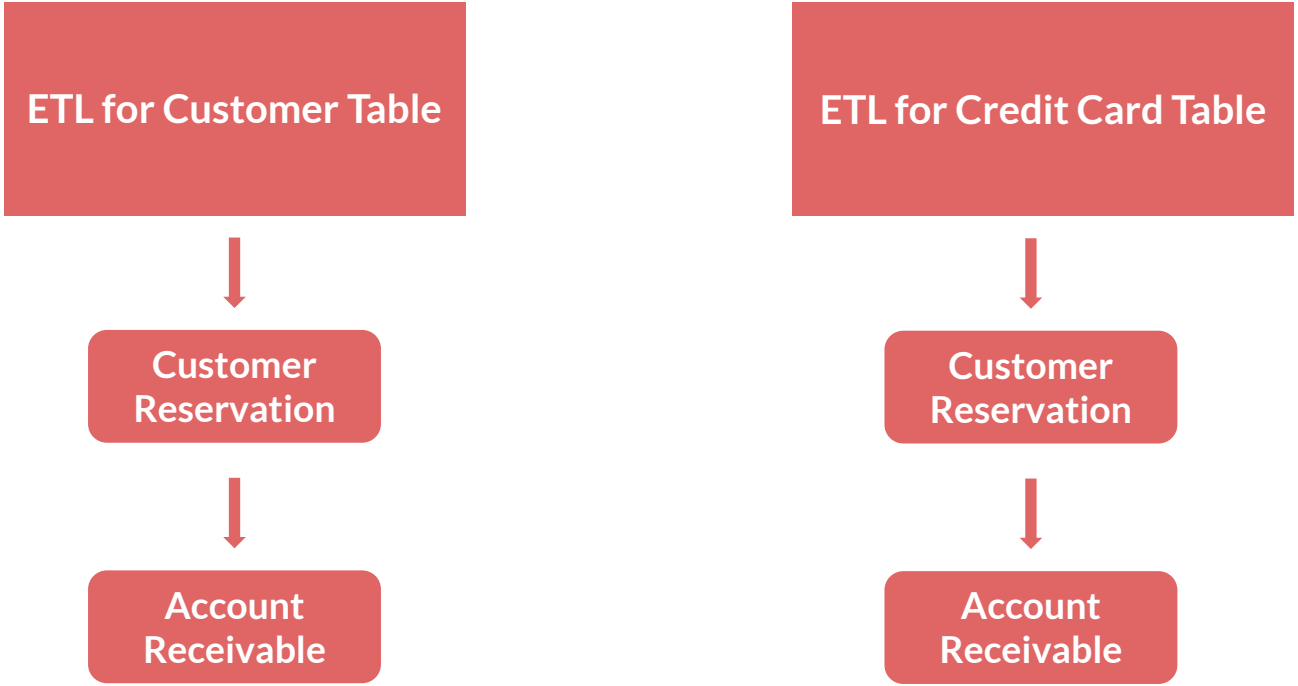
Extract Transform Load(ETL):

Inventory Management as first Priority



Extract Transform Load(ETL):

Account Receivable as second Priority



Extract Transform Load(ETL):

Only source from one transactional database

ETL for Account Receivable Table

↓

Account Receivable

ETL for Invoice Table

↓

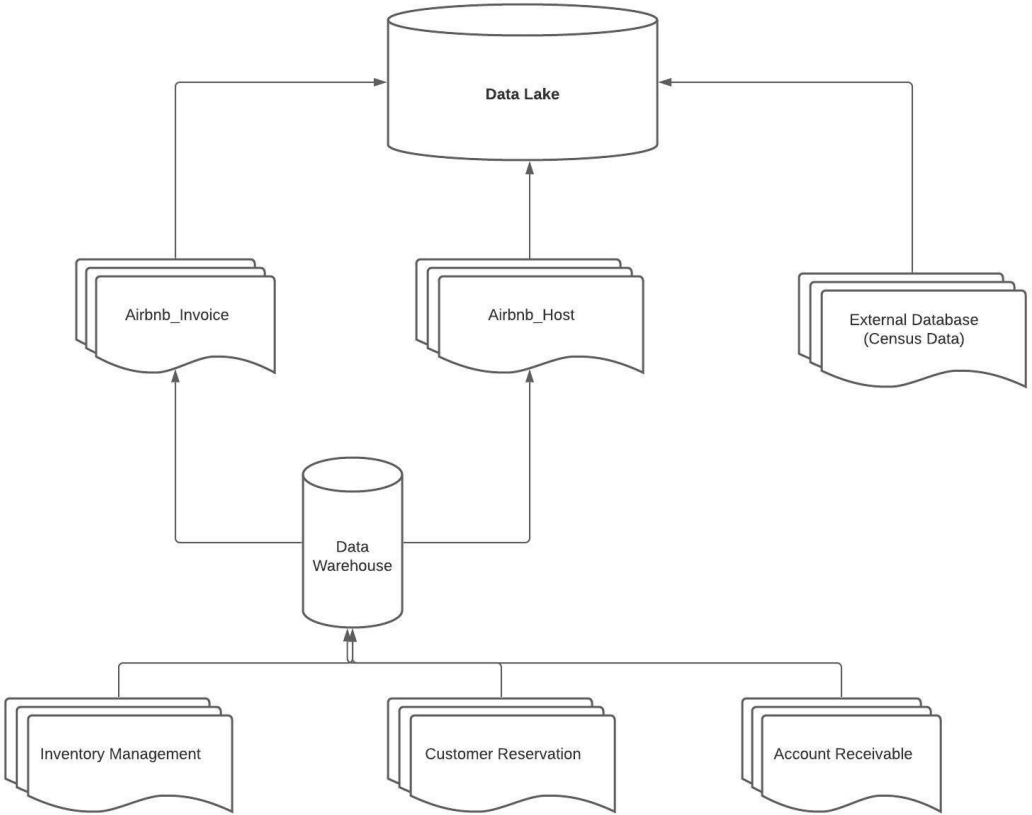
Account Receivable

ETL for Reservation Table

↓

Customer Reservation

Data Lake


Data Lake
Airbnb_Invoice
Airbnb_Host
External Database
(Census Data)
Data
Warehouse
Inventory Management
Customer Reservation
Account Receivable

Data Lake: Additional Data (Census data)

	A	B	C	D
1	City	Year	GDP	Population
2	Austin	2018	147000	0.964
3	Houston	2018	490000	2.326
4	San_Antor	2018	133000	1.487
5	Dallas	2018	521000	1.345

Analysis

Read Data

```python
host_df = spark.read.format("csv").load("dbfs:/FileStore/shared_uploads/mshch96@utexas.edu/DataLake_Host.csv", header=True, inferSchema=True)
host_df.show()
```

▸ (3) Spark Jobs

▸ host_df: pyspark.sql.dataframe.DataFrame = [host_id: integer, first_name: string ... 12 more fields]

host_id	first_name	middle_name	last_name	street_address	city	resident_state	country	zipcode	gender	superhost	phone_number	email	response_rate
1000000	Shuheng	null	Ma	706 West	Austin	TX	USA	78705	M	N	123-456-7891	1124499221@qq.com	1.5
1000001	Nico	null	Jiang	21st	Austin	TX	USA	78705	F	Y	122-456-7891	1124429221@qq.com	4.2
1000002	Tom	Jack	Williams	210E Greystone Dr	Houston	TX	USA	78711	M	Y	124-564-3455	masdsad@163.com	3.5
1000003	Betty	Kai	Miller	840 Balckstone Dr	Dallas	TX	USA	76651	M	N	123-428-7891	lfdfdad@163.com	1.5
1000004	Katherine	null	Davis	560 6th St	San Antonio	TX	USA	76689	M	N	122-456-9380	wqenjs@gmail.com	1.0
1000005	Shawn	Mia	Jones	901 Congress Dr	Houston	TX	USA	73340	F	N	179-374-4456	w3ewrewnjs@gmail.com	1.9

Analysis

Read Data

```
invoice_df = spark.read.format("csv").load("dbfs:/FileStore/shared_uploads/mshch96@utexas.edu/DataLake_Invoice-1.csv", header=True, inferSchema=True)
invoice_df.show()
```

▸ (3) Spark Jobs

▸ invoice_df: pyspark.sql.dataframe.DataFrame = [invoice_id: integer, host_id: integer ... 9 more fields]

invoice_id	host_id	invoice_number	Invoice_date	item_description	unit_price	quantity	hosting_address	hosting_city	hosting_state	hosting_zip
1000000	1000000	1000000	11/27/2018	comfortable, thre...	79	2	706 West	Austin	TX	78705
1000001	1000001	1000001	10/5/2001	Large, three bedr...	420	3	21st	Austin	TX	78705
1000002	1000002	1000002	1/20/2005	Single bedroom; f...	350	5	210E Greystone Dr	Houston	TX	78711
1000003	1000003	1000003	3/21/2007	Double bedroom; c...	120	4	840 Balckstone Dr	Dallas	TX	76651
1000004	1000004	1000004	5/25/2008	Single large bedr...	115	4	560 6th St	San Antonio	TX	76689
1000005	1000005	1000005	8/21/2010	Big house with ga...	320	8	901 Congress Dr	Houston	TX	73340

Analysis

Read Data

```
Texas_census_df = spark.read.format("csv").load("dbfs:/FileStore/shared_uploads/mshch96@utexas.edu/Additional_Data.csv", header=True, inferSchema=True)
Texas_census_df.show()

```

(3) Spark Jobs

Texas_census_df: pyspark.sql.dataframe.DataFrame = [City: string, Year: integer ... 2 more fields]

```
+-----------+----+------+----------+
|       City|Year|   GDP|Population|
+-----------+----+------+----------+
|     Austin|2018|147000|     0.964|
|     Huston|2018|490000|     2.326|
|San Antonio|2018|133000|     1.487|
|     Dallas|2018|521000|     1.345|
+-----------+----+------+----------+
```

Analysis

Create View

```
host_df.createOrReplaceTempView("host")
invoice_df.createOrReplaceTempView("invoice")
Texas_census_df.createOrReplaceTempView("texas_census")
```

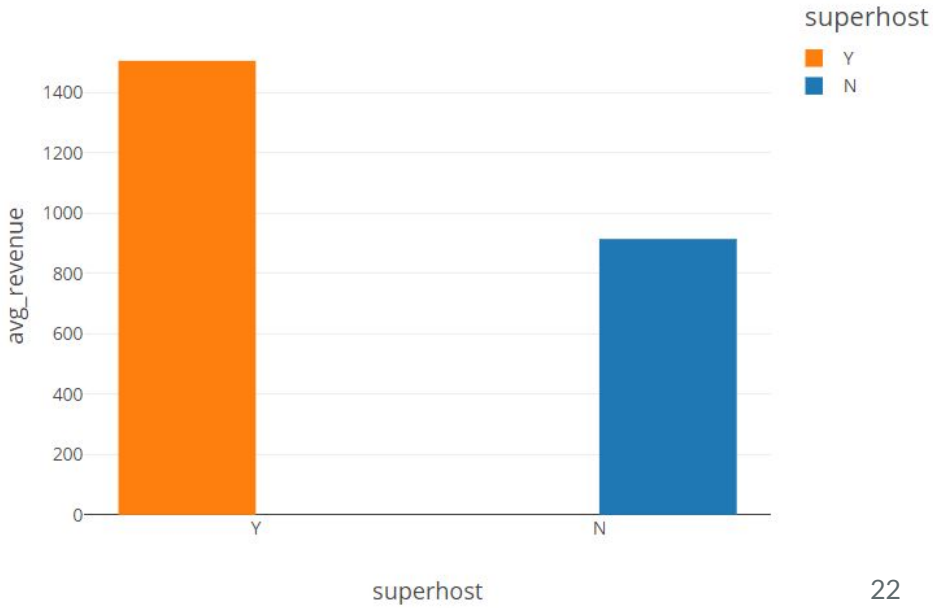
Data Lake - Insight Analysis 1

Analysis of Avg Superhost Revenue

```
%sql

SELECT avg(i.unit_price * i.quantity) as avg_revenue, h.superhost
FROM host h INNER JOIN invoice i
  ON h.host_id = i.host_id
GROUP BY h.superhost;

```



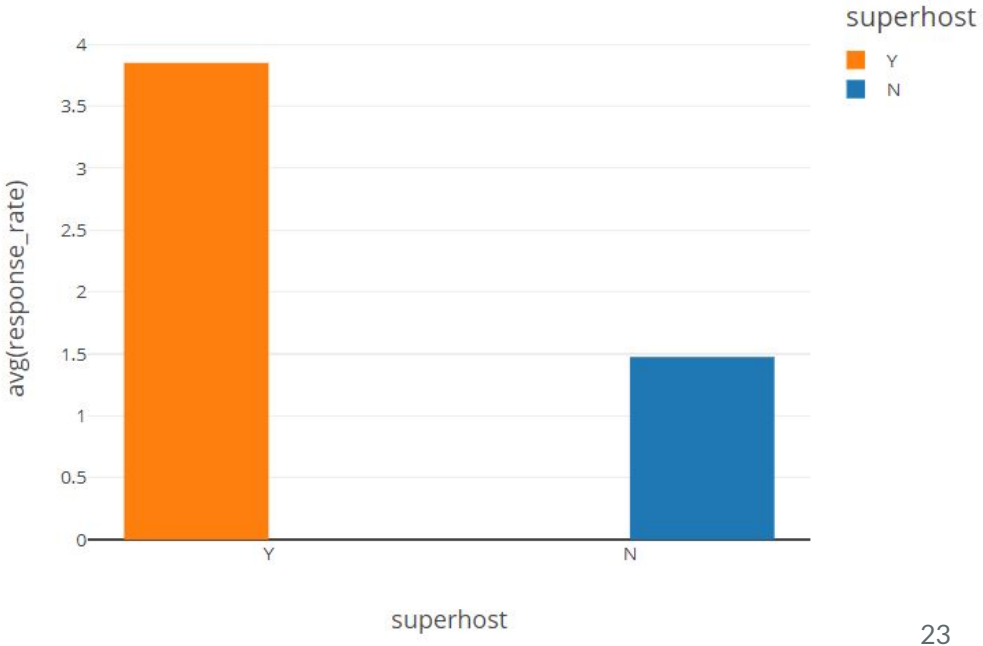
Data Lake - Insight Analysis 2

Analysis of Superhost Response Rate

```
%sql

SELECT superhost, avg(response_rate)
FROM host
Group By superhost;

```

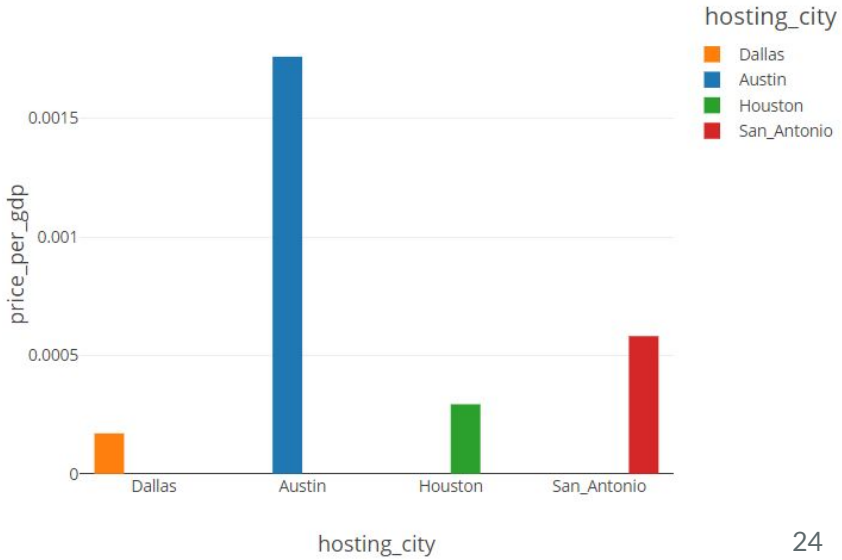


Data Lake - Insight Analysis 3

Analysis of US City Average Daily Renting Price Distribution

```
%sql

SELECT avg(i.unit_price / t.GDP / t.Population) as price_per_gdp, i.hosting_city
FROM invoice i left join texas_census t
On i.hosting_city = t.city
Group By i.hosting_city;
```



Critical Reflection: Lesson Learned

- We have learned how to compose data warehouse from seperate transactional databases, how to utilize ETL to populate data, and how to use spark to draw business insights.
- The most valuable knowledge we have learned from the project is the Airbnb company's data structures as well as all the data techniques.
- We could use these data techniques mentioned above to establish data warehouse for future companies/industries and provide business insights given certain requirements with dataset.

Critical Reflection: Future Improvement

- We could have used MongoDB as NoSQL database for Data Lake
- We could use MongoDB instead of Spark to draw and visualize business insights, since MongoDB is capable of handling much larger size of data compared to spark as well as more functions.

Thank you!