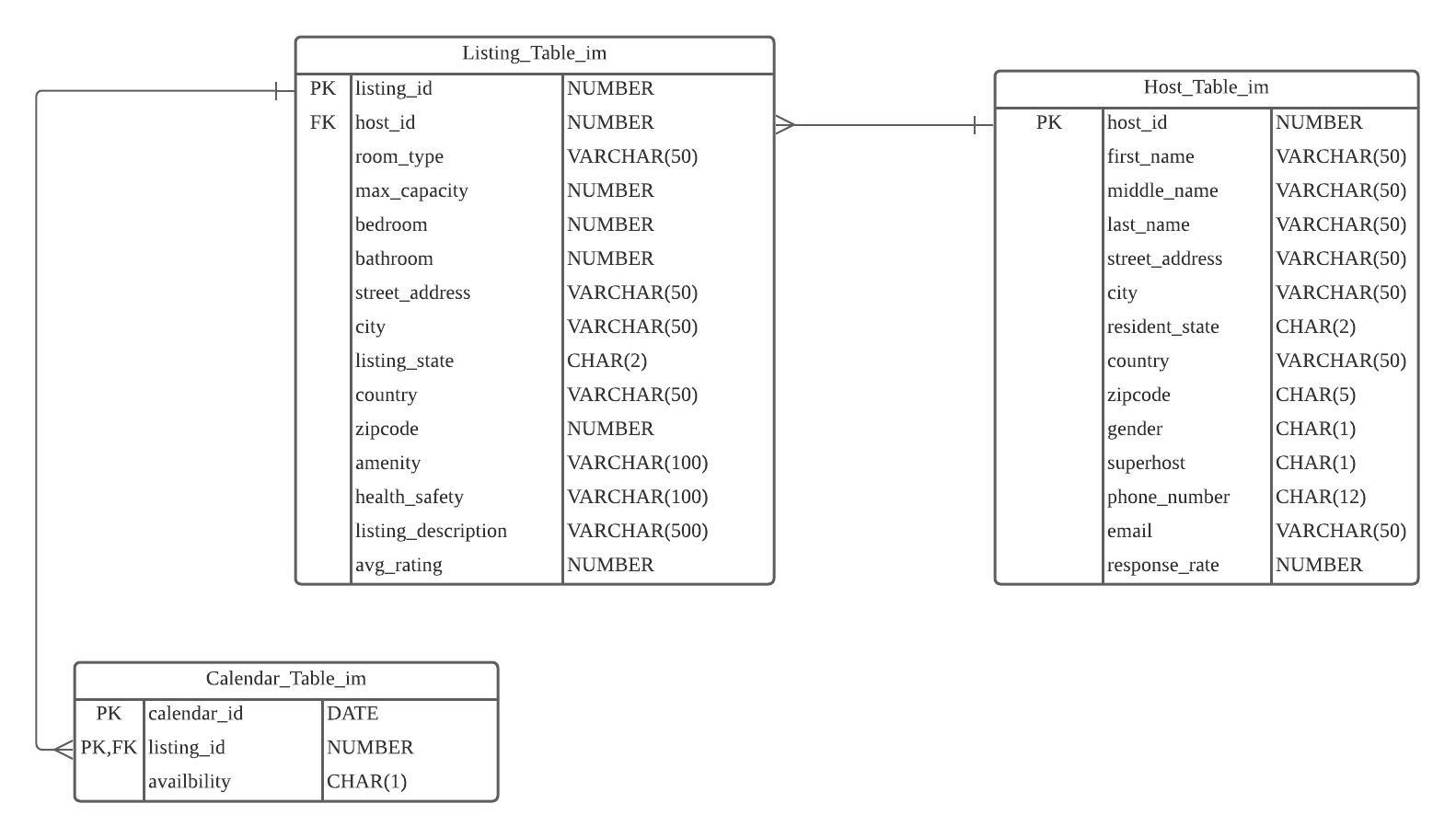
# Requirement

Submit a word document that includes the following:

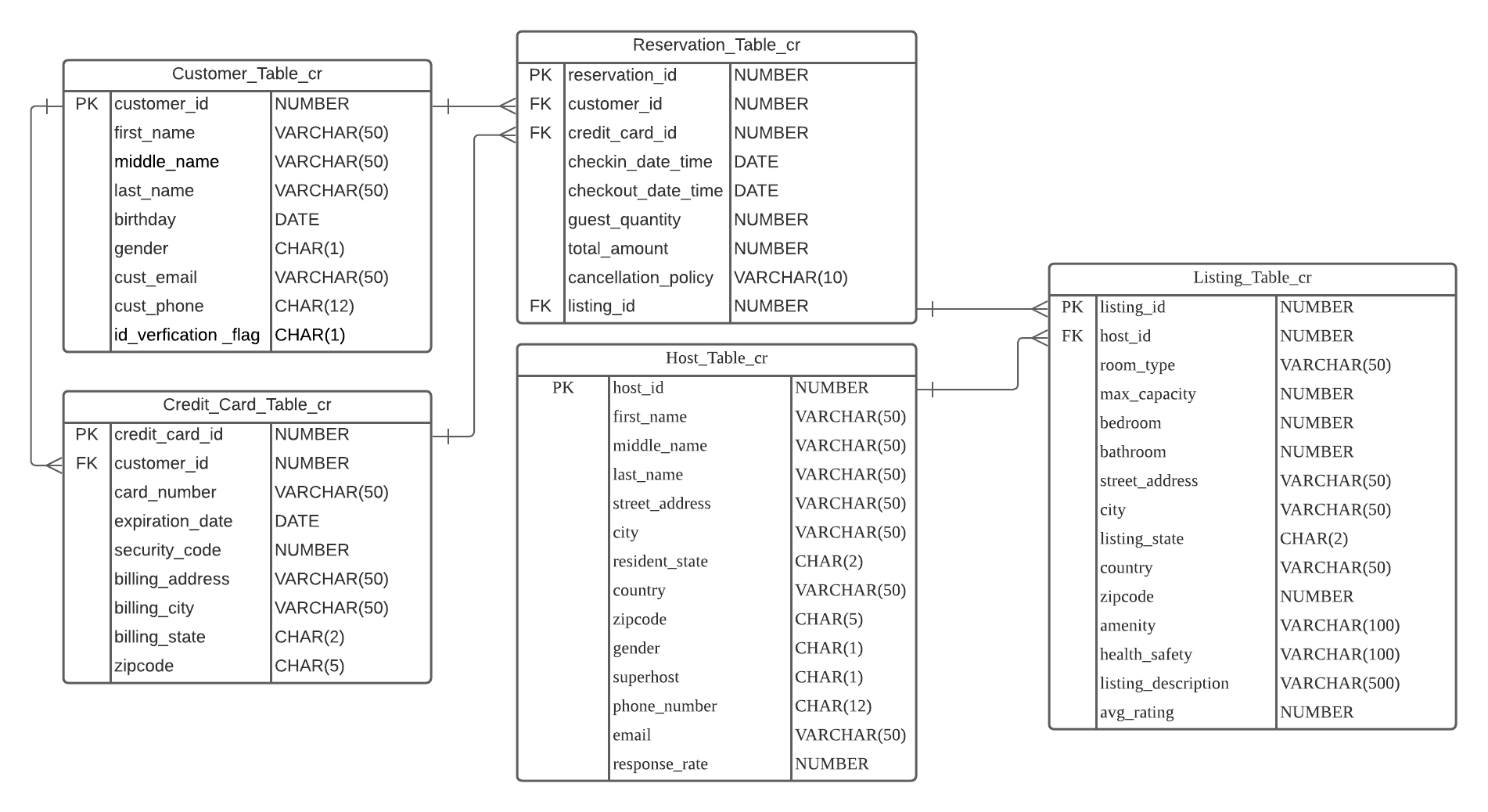
* 1. 4 LucidChart data models (3 transactional applications + 1 data warehouse)
  2. DDL for the four data models
  3. ETL code to populate the data models
  4. The model for your data lake
  5. The code for your analysis patterns

# 4 LucidChart data models (3 transactional applications + 1 data warehouse)

## Inventory Management

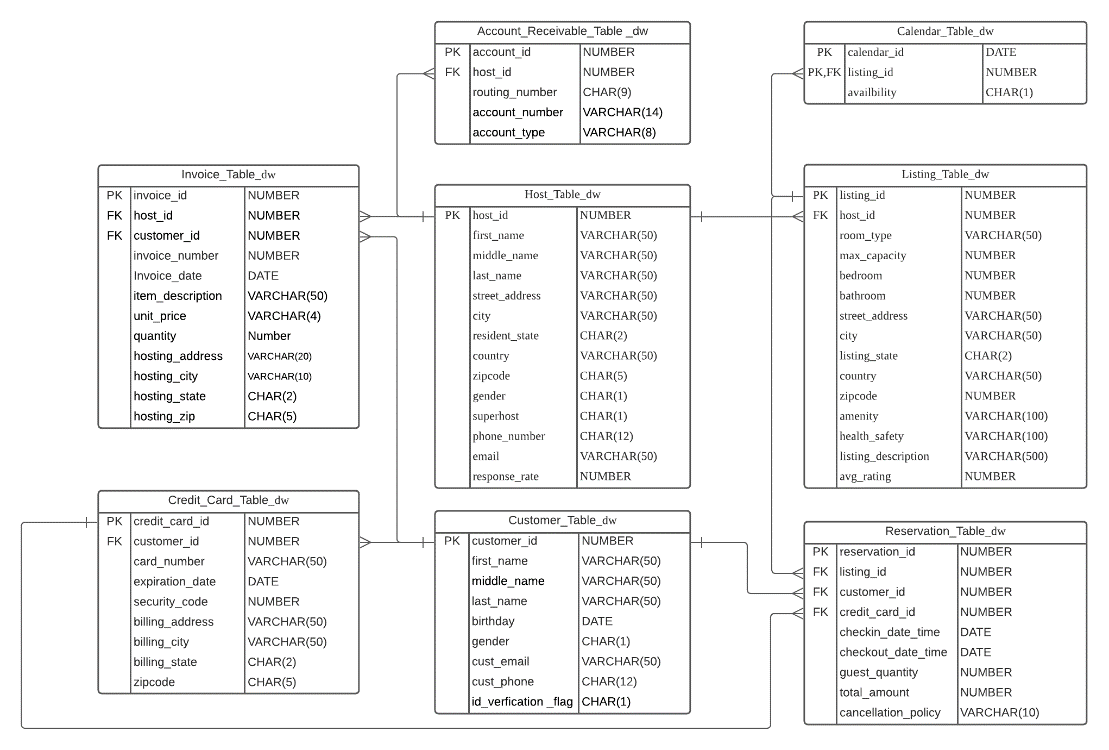


## Customer Reservation



## Account Receivable

## Data Warehouse



# DDL for the four data models

## Transactional\_Database\_Inventory\_Management

-- Drop Sequence: Drop all the sequence for all the tables(exclude linking table)

Drop Sequence host\_id\_seq\_im ;

Drop Sequence listing\_id\_seq\_im ;

-- Drop Table

DROP TABLE Calendar\_Table\_im;

DROP TABLE Listing\_Table\_im;

DROP TABLE Host\_Table\_im;

-- Create Sequence: Create sequence for all the established tables's PK(exclude linking table)

CREATE Sequence host\_id\_seq\_im

START WITH 1000000 INCREMENT BY 1;

CREATE Sequence listing\_id\_seq\_im

START WITH 1000000 INCREMENT BY 1;

-- Create Table

CREATE TABLE Host\_Table\_im (

host\_id NUMBER DEFAULT host\_id\_seq\_im.nextval PRimARY KEY,

first\_name VARCHAR(50) NOT NULL,

middle\_name VARCHAR(50),

last\_name VARCHAR(50) NOT NULL,

street\_address VARCHAR(50) NOT NULL,

city VARCHAR(50) NOT NULL,

resident\_state CHAR(2) NOT NULL,

country VARCHAR(50) NOT NULL,

zipcode CHAR(5) NOT NULL,

gender CHAR(1) NOT NULL,

superhost CHAR(1) DEFAULT 'N',

phone\_number CHAR(12) NOT NULL,

email VARCHAR(50) NOT NULL UNIQUE,

response\_rate NUMBER,

CONSTRAINT email\_length\_check

CHECK (LENGTHB(email) >= 7)

);

CREATE TABLE Listing\_Table\_im (

listing\_id NUMBER DEFAULT listing\_id\_seq\_im.nextval PRimARY KEY,

host\_id NUMBER,

room\_type VARCHAR(50) NOT NULL,

max\_capacity NUMBER NOT NULL,

bedroom NUMBER NOT NULL,

bathroom NUMBER NOT NULL,

street\_address VARCHAR(50) NOT NULL,

city VARCHAR(50) NOT NULL,

listing\_state CHAR(2) NOT NULL,

country VARCHAR(50) NOT NULL,

zipcode NUMBER NOT NULL,

amenity VARCHAR(100),

health\_safety VARCHAR(100),

listing\_description VARCHAR(500),

avg\_rating NUMBER,

CONSTRAINT listing\_table\_host\_table\_fk

FOREIGN KEY (host\_id)

REFERENCES Host\_Table\_im(host\_id)

);

CREATE TABLE Calendar\_Table\_im (

calendar\_id DATE,

listing\_id NUMBER,

room\_availability CHAR(1) DEFAULT 'Y',

CONSTRAINT calendar\_table\_pk\_listing\_table

PRimARY KEY(calendar\_id,listing\_id),

CONSTRAINT calendar\_table\_listing\_table\_fk

FOREIGN KEY (listing\_id)

REFERENCES Listing\_Table\_im(listing\_id)

);

-- Insert Data

INSERT INTO Host\_Table\_im

(first\_name,last\_name,street\_address,city,resident\_state,country,zipcode,gender,phone\_number,email,response\_rate)

VALUES ('shuheng','ma','706','austin','tx','usa','78705','M','123-456-1234','1124498677@qq.com',1.2);

INSERT INTO Host\_Table\_im

(first\_name,last\_name,street\_address,city,resident\_state,country,zipcode,gender,phone\_number,email,response\_rate)

VALUES ('shuheng2','ma2','7062','austin2','tx','usa','78725','M','123-456-1222','1124228677@qq.com',1.2);

INSERT INTO Listing\_Table\_im

(host\_id,room\_type,max\_capacity,bedroom,bathroom,street\_address,city,listing\_state,country,zipcode,amenity,health\_safety,listing\_description,avg\_rating)

VALUES (1000000,'big',5,3,2,'706','austin','tx','usa',78705,'na','nope','pay for it',3.5);

INSERT INTO Listing\_Table\_im

(host\_id,room\_type,max\_capacity,bedroom,bathroom,street\_address,city,listing\_state,country,zipcode,amenity,health\_safety,listing\_description,avg\_rating)

VALUES (1000001,'small',1,1,1,'7062','austin','tx','usa',78705,'na','nope','pay for it',5.2);

INSERT INTO Calendar\_Table\_im

(calendar\_id,listing\_id,room\_availability)

VALUES (TO\_DATE('2023/03/10','yyyy/mm/dd'),1000000,'Y');

INSERT INTO Calendar\_Table\_im

(calendar\_id,listing\_id,room\_availability)

VALUES (TO\_DATE('2019/03/11','yyyy/mm/dd'),1000001,'Y');

## Transactional\_Database\_Customer\_Reservation

-- Drop Sequence & Table

DROP TABLE Reservation\_Table\_CR;

DROP TABLE Credit\_Card\_Table\_CR;

DROP TABLE Listing\_Table\_CR;

DROP TABLE Host\_Table\_CR;

DROP TABLE Customer\_Table\_CR;

DROP SEQUENCE customer\_id\_seq\_CR;

DROP SEQUENCE credit\_card\_id\_seq\_CR;

DROP SEQUENCE listing\_id\_seq\_CR;

DROP SEQUENCE reservation\_id\_seq\_CR;

DROP SEQUENCE host\_id\_seq\_CR;

-- Create Sequence & Table

CREATE Sequence customer\_id\_seq\_CR

START WITH 1000010 INCREMENT BY 1;

CREATE Sequence host\_id\_seq\_CR

START WITH 1000010 INCREMENT BY 1;

CREATE Sequence credit\_card\_id\_seq\_CR

START WITH 1000010 INCREMENT BY 1;

CREATE Sequence listing\_id\_seq\_CR

START WITH 1000010 INCREMENT BY 1;

CREATE Sequence reservation\_id\_seq\_CR

START WITH 1000010 INCREMENT BY 1;

CREATE TABLE Customer\_Table\_CR (

customer\_id NUMBER default customer\_id\_seq\_CR.NEXTVAL PRIMARY KEY,

first\_name VARCHAR(50)NOT NULL,

middle\_name VARCHAR(50),

last\_name VARCHAR(50)NOT NULL,

birthday DATE NOT NULL,

gender CHAR(1) NOT NULL,

cust\_email VARCHAR(50) NOT NULL UNIQUE,

cust\_phone CHAR(12) NOT NULL,

id\_verification\_flag CHAR(1) DEFAULT 'N'

);

CREATE TABLE Host\_Table\_CR (

host\_id NUMBER default host\_id\_seq\_CR.NEXTVAL PRIMARY KEY,

first\_name VARCHAR(50) NOT NULL,

middle\_name VARCHAR(50),

last\_name VARCHAR(50) NOT NULL,

street\_address VARCHAR(50) NOT NULL,

city VARCHAR(50) NOT NULL,

resident\_state CHAR(2) NOT NULL,

country VARCHAR(50) NOT NULL,

zipcode CHAR(5) NOT NULL,

gender CHAR(1) NOT NULL,

superhost CHAR(1) DEFAULT 'N',

phone\_number CHAR(12) NOT NULL,

email VARCHAR(50) NOT NULL UNIQUE,

response\_rate NUMBER,

CONSTRAINT email\_length\_check\_cr

CHECK (LENGTHB(email) >= 7)

);

CREATE TABLE Credit\_Card\_Table\_CR (

credit\_card\_id NUMBER default credit\_card\_id\_seq\_CR.NEXTVAL PRIMARY KEY,

customer\_id NUMBER,

card\_number VARCHAR(50) NOT NULL,

expiration\_date DATE NOT NULL,

security\_code NUMBER NOT NULL,

billing\_address VARCHAR(50) NOT NULL,

billing\_city VARCHAR(50) NOT NULL,

billing\_state CHAR(2) NOT NULL,

zipcode CHAR(5) NOT NULL,

CONSTRAINT credit\_card\_Table\_CR\_customer\_Table\_CR\_fk

FOREIGN KEY (customer\_id)

REFERENCES Customer\_Table\_CR(customer\_id)

);

CREATE TABLE Listing\_Table\_CR (

listing\_id NUMBER default listing\_id\_seq\_CR.NEXTVAL PRIMARY KEY,

host\_id NUMBER,

room\_type VARCHAR(50) NOT NULL,

max\_capacity NUMBER NOT NULL,

bedroom NUMBER NOT NULL,

bathroom NUMBER NOT NULL,

street\_address VARCHAR(50) NOT NULL,

city VARCHAR(50) NOT NULL,

listing\_state CHAR(2) NOT NULL,

country VARCHAR(50) NOT NULL,

zipcode NUMBER NOT NULL,

amenity VARCHAR(100),

health\_safety VARCHAR(100),

listing\_description VARCHAR(500),

avg\_rating NUMBER,

CONSTRAINT listing\_Table\_CR\_host\_Table\_CR\_fk

FOREIGN KEY (host\_id)

REFERENCES Host\_Table\_CR(host\_id)

);

CREATE TABLE Reservation\_Table\_CR (

reservation\_id NUMBER default reservation\_id\_seq\_CR.NEXTVAL PRIMARY KEY,

listing\_id NUMBER,

customer\_id NUMBER,

credit\_card\_id NUMBER,

checkin\_date\_time DATE NOT NULL,

checkout\_date\_time DATE NOT NULL,

guest\_quantity NUMBER NOT NULL,

total\_amount NUMBER NOT NULL,

cancellation\_policy VARCHAR(10) NOT NULL,

CONSTRAINT reservation\_Table\_CR\_listing\_Table\_CR\_fk

FOREIGN KEY (listing\_id)

REFERENCES Listing\_Table\_CR(listing\_id),

CONSTRAINT reservation\_Table\_CR\_customer\_Table\_CR\_fk

FOREIGN KEY (customer\_id)

REFERENCES Customer\_Table\_CR(customer\_id),

CONSTRAINT reservation\_Table\_CR\_ciredit\_card\_id\_fk

FOREIGN KEY (credit\_card\_id)

REFERENCES Credit\_Card\_Table\_CR(credit\_card\_id)

);

INSERT INTO Customer\_Table\_CR (first\_name,last\_name,birthday,gender,cust\_email,cust\_phone,id\_verification\_flag)

VALUES ('Nicole','Jiang',TO\_DATE('1998/03/10','yyyy/mm/dd'),'F','annabelle@outlook.com','512-202-0832','Y');

INSERT INTO Customer\_Table\_CR (first\_name,last\_name,birthday,gender,cust\_email,cust\_phone,id\_verification\_flag)

VALUES ('Ashley','Smith',TO\_DATE('1990/02/04','yyyy/mm/dd'),'F','AshleySmith@utexas.edu','513-000-2140','Y');

INSERT INTO Customer\_Table\_CR (first\_name,last\_name,middle\_name,birthday,gender,cust\_email,cust\_phone,id\_verification\_flag)

VALUES ('Tom','Johnson','Jack',TO\_DATE('1996/03/07','yyyy/mm/dd'),'M','TomJohnson@utexas.edu','325-490-0040','N');

INSERT INTO Customer\_Table\_CR (first\_name,last\_name,middle\_name,birthday,gender,cust\_email,cust\_phone,id\_verification\_flag)

VALUES ('Betty','Williams','Kai',TO\_DATE('1994/06/12','yyyy/mm/dd'),'F','BettyWilliams@utexas.edu','512-622-7890','Y');

INSERT INTO Customer\_Table\_CR (first\_name,last\_name,birthday,gender,cust\_email,cust\_phone,id\_verification\_flag)

VALUES ('Katherine','Miller',TO\_DATE('1989/05/18','yyyy/mm/dd'),'F','KatherineMiller@utexas.edu','737-490-2247','N');

INSERT INTO Customer\_Table\_CR (first\_name,last\_name,middle\_name,birthday,gender,cust\_email,cust\_phone,id\_verification\_flag)

VALUES ('Shawn','Davis','Mia',TO\_DATE('1992/08/17','yyyy/mm/dd'),'M','ShawnDavis@utexas.edu','452-213-2355','Y');

INSERT INTO Customer\_Table\_CR (first\_name,last\_name,birthday,gender,cust\_email,cust\_phone,id\_verification\_flag)

VALUES ('Ben','Jones',TO\_DATE('1996/06/06','yyyy/mm/dd'),'M','BenJones@utexas.edu','213-352-5643','N');

INSERT INTO Credit\_Card\_Table\_CR (card\_number,expiration\_date,security\_code,billing\_address,billing\_city,billing\_state,zipcode,customer\_id)

VALUES('1234123412341234',TO\_DATE('2025/08/13', 'yyyy/mm/dd'),351,'Greystone Dr','Austin','TX','78731',1000011);

INSERT INTO Credit\_Card\_Table\_CR (card\_number,expiration\_date,security\_code,billing\_address,billing\_city,billing\_state,zipcode,customer\_id)

VALUES('4734123412341234',TO\_DATE('2026/02/12', 'yyyy/mm/dd'),231,'Balckstone Dr','Austin','TX','78712',1000012);

INSERT INTO Credit\_Card\_Table\_CR (card\_number,expiration\_date,security\_code,billing\_address,billing\_city,billing\_state,zipcode,customer\_id)

VALUES('6234123412341234',TO\_DATE('2021/12/30', 'yyyy/mm/dd'),325,'6th St','Houston','TX','78214',1000014);

INSERT INTO Credit\_Card\_Table\_CR (card\_number,expiration\_date,security\_code,billing\_address,billing\_city,billing\_state,zipcode,customer\_id)

VALUES('4134123522341234',TO\_DATE('2030/01/05', 'yyyy/mm/dd'),124,'Congress Dr','Dallas','TX','78351',1000013);

INSERT INTO Host\_Table\_CR (first\_name,last\_name,street\_address,city,resident\_state,country,zipcode,gender,phone\_number,email,response\_rate)

VALUES ('Nicole','Xu','706','austin','tx','usa','78705','M','123-456-1234','311244498677@qq.com',1.2);

INSERT INTO Host\_Table\_CR (first\_name,last\_name,street\_address,city,resident\_state,country,zipcode,gender,phone\_number,email,response\_rate)

VALUES ('Jiang','Li','7062','austin2','tx','usa','78725','F','123-456-1222','4232428677@qq.com',1.2);

INSERT INTO Host\_Table\_CR (host\_id,first\_name,last\_name,street\_address,city,resident\_state,country,zipcode,gender,phone\_number,email,response\_rate)

VALUES (1000000,'yiqun','Li','7062','austin2','tx','usa','78725','F','123-456-1222','4232s428677@qq.com',1.2);

INSERT INTO Listing\_Table\_CR (host\_id,room\_type,max\_capacity,bedroom,bathroom,street\_address,city,listing\_state,country,zipcode,amenity,health\_safety,listing\_description,avg\_rating)

VALUES (1000010,'big',5,3,2,'706','austin','tx','usa',78705,'na','nope','pay for it',4.8);

INSERT INTO Listing\_Table\_CR (host\_id,room\_type,max\_capacity,bedroom,bathroom,street\_address,city,listing\_state,country,zipcode,amenity,health\_safety,listing\_description,avg\_rating)

VALUES (1000011,'small',1,1,1,'7062','austin','tx','usa',78705,'na','nope','pay for it',4.9);

INSERT INTO Reservation\_Table\_CR (checkin\_date\_time,checkout\_date\_time,guest\_quantity,total\_amount,cancellation\_policy,listing\_id,customer\_id,credit\_card\_id)

VALUES (TO\_DATE('2021/12/30', 'yyyy/mm/dd'),TO\_DATE('2021/12/31', 'yyyy/mm/dd'),2,400,'strict',1000011,1000011,1000010);

INSERT INTO Reservation\_Table\_CR (checkin\_date\_time,checkout\_date\_time,guest\_quantity,total\_amount,cancellation\_policy,listing\_id,customer\_id,credit\_card\_id)

VALUES (TO\_DATE('2020/11/25', 'yyyy/mm/dd'),TO\_DATE('2020/11/26', 'yyyy/mm/dd'),1,200,'moderate',1000010,1000014,1000012);

INSERT INTO Reservation\_Table\_CR (checkin\_date\_time,checkout\_date\_time,guest\_quantity,total\_amount,cancellation\_policy,listing\_id,customer\_id,credit\_card\_id)

VALUES (TO\_DATE('2021/01/12', 'yyyy/mm/dd'),TO\_DATE('2021/01/13', 'yyyy/mm/dd'),1,150,'strict',1000011,1000013,1000013);

## Transactional\_Database\_Account\_Receivable

-- Drop Sequence: Drop all the sequence for all the tables(exclude linking table)

Drop Sequence account\_id\_seq\_ar ;

Drop Sequence invoice\_id\_seq\_ar ;

Drop Sequence credit\_card\_id\_seq\_ar ;

Drop Sequence host\_id\_seq\_ar ;

Drop Sequence customer\_id\_seq\_ar ;

-- Drop Table

DROP TABLE Account\_Receivable\_Table\_ar;

DROP TABLE Invoice\_Table\_ar;

DROP TABLE Credit\_Card\_Table\_ar;

DROP TABLE Host\_Table\_ar;

DROP TABLE Customer\_Table\_ar;

-- Create Sequence: Create sequence for all the established tables's PK(exclude linking table)

CREATE Sequence account\_id\_seq\_ar

START WITH 1000000 INCREMENT BY 1;

CREATE Sequence invoice\_id\_seq\_ar

START WITH 1000000 INCREMENT BY 1;

CREATE Sequence credit\_card\_id\_seq\_ar

START WITH 1000000 INCREMENT BY 1;

CREATE Sequence host\_id\_seq\_ar

START WITH 1000000 INCREMENT BY 1;

CREATE Sequence customer\_id\_seq\_ar

START WITH 1000000 INCREMENT BY 1;

-- Create

CREATE TABLE Customer\_Table\_ar (

customer\_id NUMBER DEFAULT customer\_id\_seq\_ar.nextval PRIMARY KEY,

first\_name VARCHAR(50) NOT NULL,

middle\_name VARCHAR(50),

last\_name VARCHAR(50) NOT NULL,

birthday DATE NOT NULL,

gender CHAR(1) NOT NULL,

cust\_email VARCHAR(50) NOT NULL UNIQUE,

cust\_phone CHAR(12) NOT NULL,

id\_verification\_flag CHAR(1) DEFAULT ('N')

);

CREATE TABLE Host\_Table\_ar (

host\_id NUMBER DEFAULT host\_id\_seq\_ar.nextval PRIMARY KEY,

first\_name VARCHAR(50) NOT NULL,

middle\_name VARCHAR(50),

last\_name VARCHAR(50) NOT NULL,

street\_address VARCHAR(50) NOT NULL,

city VARCHAR(50) NOT NULL,

resident\_state CHAR(2) NOT NULL,

country VARCHAR(50) NOT NULL,

zipcode CHAR(5) NOT NULL,

gender CHAR(1) NOT NULL,

superhost CHAR(1) DEFAULT 'N',

phone\_number CHAR(12) NOT NULL,

email VARCHAR(50) NOT NULL UNIQUE,

response\_rate NUMBER

);

CREATE TABLE Credit\_Card\_Table\_ar (

credit\_card\_id NUMBER DEFAULT credit\_card\_id\_seq\_ar.nextval PRIMARY KEY,

customer\_id NUMBER,

card\_number VARCHAR(50) NOT NULL,

expiration\_date DATE NOT NULL,

security\_code NUMBER NOT NULL,

billing\_address VARCHAR(50) NOT NULL,

billing\_city VARCHAR(50) NOT NULL,

billing\_state CHAR(2) NOT NULL,

zipcode CHAR(5) NOT NULL,

CONSTRAINT credit\_card\_table\_customer\_table\_fk

FOREIGN KEY (customer\_id)

REFERENCES Customer\_Table\_AR (customer\_id)

);

CREATE TABLE Invoice\_Table\_ar (

invoice\_id NUMBER DEFAULT invoice\_id\_seq\_ar.nextval PRIMARY KEY,

host\_id NUMBER,

customer\_id NUMBER,

invoice\_number NUMBER NOT NULL,

Invoice\_date DATE NOT NULL,

item\_description VARCHAR(50) NOT NULL,

unit\_price VARCHAR(4) NOT NULL,

quantity Number NOT NULL,

hosting\_address VARCHAR(50) NOT NULL,

hosting\_city VARCHAR(50) NOT NULL,

hosting\_state CHAR(2) NOT NULL,

hosting\_zip CHAR(5) NOT NULL,

CONSTRAINT invoice\_table\_host\_table\_fk

FOREIGN KEY (host\_id)

REFERENCES Host\_Table\_AR (host\_id),

CONSTRAINT invoice\_table\_customer\_table\_fk

FOREIGN KEY (customer\_id)

REFERENCES Customer\_Table\_AR(customer\_id)

);

CREATE TABLE Account\_Receivable\_Table\_ar (

account\_id NUMBER DEFAULT account\_id\_seq\_ar.nextval PRIMARY KEY,

host\_id NUMBER,

routing\_number CHAR(9) NOT NULL,

account\_number VARCHAR(14) NOT NULL,

account\_type VARCHAR(8) NOT NULL,

CONSTRAINT account\_receivable\_table\_host\_table\_fk

FOREIGN KEY (host\_id)

REFERENCES Host\_Table\_AR(host\_id)

);

INSERT INTO Customer\_Table\_AR (customer\_id, first\_name, last\_name, birthday, gender, cust\_email, cust\_phone)

VALUES (1000030, 'Yiqun', 'Tian', TO\_DATE('1997-02-03', 'YYYY-MM-DD'), 'M', 'yiqun.tian@utexas.edu', '444-333-1111');

INSERT INTO Customer\_Table\_AR (customer\_id, First\_name, Last\_name, birthday, gender, cust\_email, cust\_phone)

VALUES (1000031, 'Michael', 'Jordan', TO\_DATE('1963-02-17', 'YYYY-MM-DD'), 'M', 'michael.j@utexas.edu', '877-888-3331');

INSERT INTO Customer\_Table\_AR (customer\_id, First\_name, Middle\_name, Last\_name, birthday, gender, cust\_email, cust\_phone)

VALUES (1000032, 'Kevin', 'Wayne', 'Durant', TO\_DATE('1988-09-29', 'YYYY-MM-DD'), 'M', 'kevin.durant@utexas.edu', '744-331-4422');

ALTER SESSION SET NLS\_DATE\_FORMAT = 'mm/yy';

INSERT INTO Credit\_Card\_Table\_AR (credit\_card\_id, customer\_id, card\_number, expiration\_date, security\_code, billing\_address, Billing\_city, Billing\_state, zipcode)

VALUES (1000030, 1000030, '1234567890123456', '08/24', '123', '913 Pearl Road', 'Tempe', 'AZ', '85288');

INSERT INTO Credit\_Card\_Table\_AR (credit\_card\_id, customer\_id, card\_number, expiration\_date, security\_code, billing\_address, Billing\_city, Billing\_state, zipcode)

VALUES (1000031, 1000031, '1111567890123000', '03/26', '263', '933 N Sunny Road', 'Tempe', 'AZ', '85288');

INSERT INTO Host\_Table\_AR (host\_id, first\_name, middle\_name, last\_name, street\_address, city, resident\_state, country, zipcode, gender, phone\_number, email, response\_rate)

VALUES (1000010, 'John', 'Wayne', 'Stevens', '777 W curry Road', 'Austin', 'TX', 'USA', 78700, 'M', '480-477-1131', 'steve121ns.j@gmail.com', 1.5);

INSERT INTO Host\_Table\_AR (host\_id, first\_name, middle\_name, last\_name, street\_address, city, resident\_state, country, zipcode, gender, phone\_number, email, response\_rate)

VALUES (1000000, 'John', 'Wayne', 'Stevens', '777 W curry Road', 'Austin', 'TX', 'USA', 78700, 'M', '480-477-1131', 'steve1ns.j@gmail.com', 1.5);

INSERT INTO Host\_Table\_AR (host\_id, first\_name, middle\_name, last\_name, street\_address, city, resident\_state, country, zipcode, gender, phone\_number, email, response\_rate)

VALUES (1000030, 'John', 'Wayne', 'Stevens', '777 W curry Road', 'Austin', 'TX', 'USA', 78700, 'M', '480-477-1131', 'stevens.j@gmail.com', 1.5);

INSERT INTO Host\_Table\_AR (host\_id, first\_name, last\_name, street\_address, city, resident\_state, country, zipcode, gender, phone\_number, email, response\_rate)

VALUES (1000031, 'Bain', 'Stevens','675 W Sunny Road', 'Austin', 'TX', 'USA', 78770, 'M', '480-477-1131', 'stevedfrns.j@gmail.com', 2.1);

INSERT INTO Invoice\_Table\_AR (host\_id, customer\_id, invoice\_number, Invoice\_date, item\_description, unit\_price, quantity, hosting\_address, hosting\_city, hosting\_state, hosting\_zip)

VALUES (1000030, 1000030, 1000000, '08/24', 'comfortable, three bedrooms condo', 79, 2,'913 N Pearl Road', 'Phoenix', 'AZ', 85220);

INSERT INTO Invoice\_Table\_AR (host\_id, customer\_id, invoice\_number, Invoice\_date, item\_description, unit\_price, quantity, hosting\_address, hosting\_city, hosting\_state, hosting\_zip)

VALUES (1000031, 1000031, 1000001, '07/14', 'Large, three bedrooms beach house', 150, 7,'983 N Chandler Road', 'Austin', 'TX', 78772);

INSERT INTO Account\_Receivable\_Table\_AR

(host\_id,routing\_number,account\_number,account\_type)

VALUES (1000030,123456789,12345678912345,'checking');

INSERT INTO Account\_Receivable\_Table\_AR

(host\_id,routing\_number,account\_number,account\_type)

VALUES (1000031,123456789,12345678912345,'checking');

## Data\_Warehouse

-- Drop Table

DROP TABLE Account\_Receivable\_Table\_dw;

DROP TABLE Calendar\_Table\_dw;

DROP TABLE Invoice\_Table\_dw;

DROP TABLE Reservation\_Table\_dw;

DROP TABLE Listing\_Table\_dw;

DROP TABLE Credit\_Card\_Table\_dw;

DROP TABLE Host\_Table\_dw;

DROP TABLE Customer\_Table\_dw;

-- Create

CREATE TABLE Customer\_Table\_dw (

customer\_id NUMBER PRIMARY KEY,

first\_name VARCHAR(50)NOT NULL,

middle\_name VARCHAR(50),

last\_name VARCHAR(50)NOT NULL,

birthday DATE NOT NULL,

gender CHAR(1) NOT NULL,

cust\_email VARCHAR(50) NOT NULL UNIQUE,

cust\_phone CHAR(12) NOT NULL,

id\_verification\_flag CHAR(1) DEFAULT 'N'

);

CREATE TABLE Host\_Table\_dw (

host\_id NUMBER PRIMARY KEY,

first\_name VARCHAR(50) NOT NULL,

middle\_name VARCHAR(50),

last\_name VARCHAR(50) NOT NULL,

street\_address VARCHAR(50) NOT NULL,

city VARCHAR(50) NOT NULL,

resident\_state CHAR(2) NOT NULL,

country VARCHAR(50) NOT NULL,

zipcode CHAR(5) NOT NULL,

gender CHAR(1) NOT NULL,

superhost CHAR(1) DEFAULT 'N',

phone\_number CHAR(12) NOT NULL,

email VARCHAR(50) NOT NULL UNIQUE,

response\_rate NUMBER

);

CREATE TABLE Credit\_Card\_Table\_dw (

credit\_card\_id NUMBER PRIMARY KEY,

customer\_id NUMBER,

card\_number VARCHAR(50) NOT NULL,

expiration\_date DATE NOT NULL,

security\_code NUMBER NOT NULL,

billing\_address VARCHAR(50) NOT NULL,

billing\_city VARCHAR(50) NOT NULL,

billing\_state CHAR(2) NOT NULL,

zipcode CHAR(5) NOT NULL,

CONSTRAINT credit\_card\_table\_customer\_dw\_fk

FOREIGN KEY (customer\_id)

REFERENCES Customer\_Table\_dw(customer\_id)

);

CREATE TABLE Listing\_Table\_dw (

listing\_id NUMBER PRIMARY KEY,

host\_id NUMBER,

room\_type VARCHAR(50) NOT NULL,

max\_capacity NUMBER NOT NULL,

bedroom NUMBER NOT NULL,

bathroom NUMBER NOT NULL,

street\_address VARCHAR(50) NOT NULL,

city VARCHAR(50) NOT NULL,

listing\_state CHAR(2) NOT NULL,

country VARCHAR(50) NOT NULL,

zipcode NUMBER NOT NULL,

amenity VARCHAR(100),

health\_safety VARCHAR(100),

listing\_description VARCHAR(500),

avg\_rating NUMBER,

CONSTRAINT listing\_table\_host\_dw\_fk

FOREIGN KEY (host\_id)

REFERENCES Host\_Table\_dw(host\_id)

);

CREATE TABLE Reservation\_Table\_dw (

reservation\_id NUMBER PRIMARY KEY,

listing\_id NUMBER,

customer\_id NUMBER,

credit\_card\_id NUMBER,

checkin\_date\_time DATE NOT NULL,

checkout\_date\_time DATE NOT NULL,

guest\_quantity NUMBER NOT NULL,

total\_amount NUMBER DEFAULT 0,

cancellation\_policy VARCHAR(10) NOT NULL,

CONSTRAINT reservation\_dw\_listing\_dw\_fk

FOREIGN KEY (listing\_id)

REFERENCES Listing\_Table\_dw(listing\_id),

CONSTRAINT reservation\_dw\_customer\_dw\_fk

FOREIGN KEY (customer\_id)

REFERENCES Customer\_Table\_dw(customer\_id),

CONSTRAINT reservation\_dw\_ciredit\_card\_dw\_fk

FOREIGN KEY (credit\_card\_id)

REFERENCES Credit\_Card\_Table\_dw(credit\_card\_id)

);

CREATE TABLE Invoice\_Table\_dw (

invoice\_id NUMBER PRIMARY KEY,

host\_id NUMBER,

customer\_id NUMBER,

invoice\_number NUMBER NOT NULL,

Invoice\_date DATE NOT NULL,

item\_description VARCHAR(50) NOT NULL,

unit\_price VARCHAR(4) NOT NULL,

quantity Number NOT NULL,

hosting\_address VARCHAR(20) NOT NULL,

hosting\_city VARCHAR(10) NOT NULL,

hosting\_state CHAR(2) NOT NULL,

hosting\_zip CHAR(5) NOT NULL,

CONSTRAINT invoice\_dw\_host\_dw\_fk

FOREIGN KEY (host\_id)

REFERENCES Host\_Table\_dw(host\_id),

CONSTRAINT invoice\_dw\_customer\_dw\_fk

FOREIGN KEY (customer\_id)

REFERENCES Customer\_Table\_dw(customer\_id)

);

CREATE TABLE Calendar\_Table\_dw (

calendar\_id DATE,

listing\_id NUMBER,

room\_availability CHAR(1) DEFAULT 'Y',

CONSTRAINT calendar\_dw\_pk\_listing\_dw

PRIMARY KEY(calendar\_id,listing\_id),

CONSTRAINT calendar\_dw\_listing\_dw\_fk

FOREIGN KEY (listing\_id)

REFERENCES Listing\_Table\_dw(listing\_id)

);

CREATE TABLE Account\_Receivable\_Table\_dw (

account\_id NUMBER PRIMARY KEY,

host\_id NUMBER,

routing\_number CHAR(9) NOT NULL,

account\_number VARCHAR(14) NOT NULL,

account\_type VARCHAR(8) NOT NULL,

CONSTRAINT account\_receivable\_dw\_host\_dw\_fk

FOREIGN KEY (host\_id)

REFERENCES Host\_Table\_dw(host\_id)

);

## ETL code to populate the data models

-- Create DW

-- Create View for Inventoray Management

Create or Replace View Host\_Table\_im\_view as

Select \*

From Host\_Table\_im;

Create or Replace View Listing\_Table\_im\_view as

Select \*

From Listing\_Table\_im;

Create or Replace View Calendar\_Table\_im\_view as

Select \*

From Calendar\_Table\_im;

-- Customer Reservation View

Create or Replace View Host\_Table\_CR\_view as

Select \*

From Host\_Table\_CR;

Create or replace view Customer\_Table\_CR\_view as

select t.\*

from Customer\_Table\_CR t;

Create or replace view Host\_Table\_CR\_view as

select t.\*

from Host\_Table\_CR t;

Create or replace view Credit\_Card\_Table\_CR\_view as

select t.\*

from Credit\_Card\_Table\_CR t;

Create or replace view Listing\_Table\_CR\_view as

select t.\*

from Listing\_Table\_CR t;

Create or replace view Reservation\_Table\_CR\_view as

select t.\*

from Reservation\_Table\_CR t;

-- Account

Create or Replace View Account\_Receivable\_Table\_ar\_view as

Select \*

From Account\_Receivable\_Table\_ar;

Create or Replace View Invoice\_Table\_ar\_view as

Select \*

From Invoice\_Table\_ar;

Create or Replace View Credit\_Card\_Table\_ar\_view as

Select \*

From Credit\_Card\_Table\_ar;

Create or Replace View Host\_Table\_ar\_view as

Select \*

From Host\_Table\_ar;

Create or Replace View Customer\_Table\_ar\_view as

Select \*

From Customer\_Table\_ar;

-- Create Procedure

-- Customer ETL

Create or replace procedure customer\_etl\_proc

As

Begin

-- customer reservation

MERGE INTO Customer\_Table\_dw dw

USING Customer\_Table\_CR\_view v

ON (dw.customer\_id = v.customer\_id)

WHEN MATCHED

THEN UPDATE SET

dw.first\_name = v.first\_name,

dw.middle\_name = v.middle\_name,

dw.last\_name = v.last\_name,

dw.birthday = v.birthday,

dw.gender = v.gender,

dw.cust\_email = v.cust\_email,

dw.cust\_phone = v.cust\_phone,

dw.id\_verification\_flag = v.id\_verification\_flag

WHEN NOT MATCHED

THEN INSERT (customer\_id,first\_name,middle\_name,last\_name, birthday, gender,cust\_email,cust\_phone,id\_verification\_flag)

VALUES (v.customer\_id,v.first\_name, v.middle\_name, v.last\_name,

v.birthday, v.gender, v.cust\_email,v.cust\_phone,v.id\_verification\_flag);

-- account

MERGE INTO Customer\_Table\_dw a

USING Customer\_Table\_ar\_view b

ON (a.customer\_id = b.customer\_id)

WHEN MATCHED

THEN UPDATE SET

a.first\_name = b.first\_name,

a.middle\_name = b.middle\_name,

a.last\_name = b.last\_name,

a.birthday = b.birthday,

a.gender = b.gender,

a.cust\_email = b.cust\_email,

a.cust\_phone = b.cust\_phone,

a.id\_verification\_flag = b.id\_verification\_flag

WHEN NOT MATCHED

THEN INSERT (customer\_id,first\_name,middle\_name,last\_name, birthday, gender,cust\_email,cust\_phone,id\_verification\_flag)

VALUES (b.customer\_id,b.first\_name, b.middle\_name, b.last\_name,

b.birthday, b.gender, b.cust\_email,b.cust\_phone,b.id\_verification\_flag);

END;

/

-----------------------------------

-- Host ETL

CREATE OR REPLACE PROCEDURE Host\_Table\_im\_etl\_proc

AS

BEGIN

-- Customer Reservation

MERGE INTO Host\_Table\_dw dw

USING Host\_Table\_CR\_view v

ON (dw.host\_id = v.host\_id)

WHEN MATCHED

THEN UPDATE SET

dw.first\_name = v.first\_name,

dw.middle\_name = v.middle\_name,

dw.last\_name = v.last\_name,

dw.street\_address = v.street\_address,

dw.city = v.city,

dw.resident\_state = v.resident\_state,

dw.country = v.country,

dw.zipcode = v.zipcode,

dw.gender = v.gender,

dw.superhost = v.superhost,

dw.phone\_number = v.phone\_number,

dw.email = v.email,

dw.response\_rate = v.response\_rate

WHEN NOT MATCHED

THEN INSERT (host\_id,first\_name,middle\_name,last\_name,

street\_address, city,resident\_state,country,

zipcode,gender,superhost,phone\_number,email,response\_rate)

VALUES (v.host\_id,v.first\_name, v.middle\_name, v.last\_name,

v.street\_address, v.city, v.resident\_state,v.country,

v.zipcode,v.gender,v.superhost,v.phone\_number,v.email,v.response\_rate);

-- Account

MERGE INTO Host\_Table\_dw a

USING Host\_Table\_ar\_view b

ON (a.host\_id = b.host\_id)

WHEN MATCHED THEN

UPDATE SET

a.first\_name = b.first\_name,

a.middle\_name = b.middle\_name,

a.last\_name = b.last\_name,

a.street\_address =b.street\_address,

a.city =b.city,

a.resident\_state=b.resident\_state,

a.country = b.country,

a.zipcode = b.zipcode,

a.gender = b.gender,

a.superhost = b.superhost,

a.phone\_number = b.phone\_number,

a.email = b.email,

a.response\_rate = b.response\_rate

WHEN NOT MATCHED THEN

INSERT (host\_id,first\_name,middle\_name,last\_name,street\_address,city,resident\_state,country,zipcode,gender,superhost,phone\_number,email,response\_rate)

VALUES (b.host\_id,b.first\_name,b.middle\_name,b.last\_name,b.street\_address,b.city,b.resident\_state,b.country,b.zipcode,b.gender,b.superhost,b.phone\_number,b.email,b.response\_rate);

-- Inventory Management

MERGE INTO Host\_Table\_dw a

USING Host\_Table\_im\_view b

ON (a.host\_id = b.host\_id)

WHEN MATCHED THEN

UPDATE SET

a.first\_name = b.first\_name,

a.middle\_name = b.middle\_name,

a.last\_name = b.last\_name,

a.street\_address =b.street\_address,

a.city =b.city,

a.resident\_state=b.resident\_state,

a.country = b.country,

a.zipcode = b.zipcode,

a.gender = b.gender,

a.superhost = b.superhost,

a.phone\_number = b.phone\_number,

a.email = b.email,

a.response\_rate = b.response\_rate

WHEN NOT MATCHED THEN

INSERT (host\_id,first\_name,middle\_name,last\_name,street\_address,city,resident\_state,country,zipcode,gender,superhost,phone\_number,email,response\_rate)

VALUES (b.host\_id,b.first\_name,b.middle\_name,b.last\_name,b.street\_address,b.city,b.resident\_state,b.country,b.zipcode,b.gender,b.superhost,b.phone\_number,b.email,b.response\_rate);

END;

/

-----------------------------------

-- Listing

CREATE OR REPLACE PROCEDURE Listing\_Table\_im\_etl\_proc

AS

BEGIN

-- Customer Reservation

MERGE INTO Listing\_Table\_dw dw

USING Listing\_Table\_CR v

ON (dw.listing\_id = v.listing\_id)

WHEN MATCHED

THEN UPDATE SET

dw.host\_id = v.host\_id,

dw.room\_type = v.room\_type,

dw.max\_capacity = v.max\_capacity,

dw.bedroom = v.bedroom,

dw.bathroom = v.bathroom,

dw.street\_address = v.street\_address,

dw.city = v.city,

dw.listing\_state = v.listing\_state,

dw.country = v.country,

dw.zipcode = v.zipcode,

dw.amenity = v.amenity,

dw.health\_safety = v.health\_safety,

dw.listing\_description = v.listing\_description,

dw.avg\_rating = v.avg\_rating

WHEN NOT MATCHED

THEN INSERT (listing\_id,host\_id,room\_type,max\_capacity,bedroom,

bathroom, street\_address,city,listing\_state,country,zipcode,amenity,health\_safety,listing\_description,avg\_rating)

VALUES (v.listing\_id,v.host\_id,v.room\_type, v.max\_capacity, v.bedroom,

v.bathroom, v.street\_address, v.city,v.listing\_state,v.country,v.zipcode,v.amenity,v.health\_safety,v.listing\_description,v.avg\_rating);

-- Inventory Management

MERGE INTO Listing\_Table\_dw a

USING Listing\_Table\_im b

ON (a.listing\_id = b.listing\_id)

WHEN MATCHED THEN

UPDATE SET

a.host\_id = b.host\_id,

a.room\_type = b.room\_type,

a.max\_capacity = b.max\_capacity,

a.bedroom = b.bedroom,

a.bathroom = b.bathroom,

a.street\_address = b.street\_address,

a.city = b.city,

a.listing\_state = b.listing\_state,

a.country = b.country,

a.zipcode = b.zipcode,

a.amenity = b.amenity,

a.health\_safety = b.health\_safety,

a.listing\_description = b.listing\_description,

a.avg\_rating = b.avg\_rating

WHEN NOT MATCHED THEN

INSERT(listing\_id, host\_id, room\_type, max\_capacity, bedroom, bathroom,street\_address, city, listing\_state, country, zipcode, amenity, health\_safety, listing\_description, avg\_rating)

VALUES(b.listing\_id, b.host\_id, b.room\_type, b.max\_capacity, b.bedroom, b.bathroom,b.street\_address, b.city, b.listing\_state, b.country, b.zipcode, b.amenity, b.health\_safety, b.listing\_description, b.avg\_rating);

END;

/

-----------------------------------

-- Calendar

CREATE OR REPLACE PROCEDURE Calendar\_Table\_im\_etl\_proc

AS

BEGIN

-- insert and update from curr\_user\_view

MERGE INTO Calendar\_Table\_dw a

USING Calendar\_Table\_im\_view b

ON (a.calendar\_id = b.calendar\_id and a.listing\_id = b.listing\_id)

WHEN MATCHED THEN

UPDATE SET

a.room\_availability = b.room\_availability

WHEN NOT MATCHED THEN

INSERT (calendar\_id,listing\_id,room\_availability)

VALUES (b.calendar\_id,b.listing\_id,b.room\_availability);

-- insert and update from prospective\_view

END;

/

-----------------------------------------

-- Credit Card

Create or replace procedure Credit\_Card\_etl\_proc

As

Begin

-- Customer Reservation

MERGE INTO Credit\_Card\_Table\_dw dw

USING Credit\_Card\_Table\_CR\_view v

ON (dw.credit\_card\_id = v.credit\_card\_id)

WHEN MATCHED

THEN UPDATE SET

dw.customer\_id = v.customer\_id,

dw.card\_number = v.card\_number,

dw.expiration\_date = v.expiration\_date,

dw.security\_code = v.security\_code,

dw.billing\_address = v.billing\_address,

dw.billing\_city = v.billing\_city,

dw.billing\_state = v.billing\_state,

dw.zipcode = v.zipcode

WHEN NOT MATCHED

THEN INSERT (credit\_card\_id,customer\_id,card\_number,expiration\_date,

security\_code, billing\_address,billing\_city,billing\_state,zipcode)

VALUES (v.credit\_card\_id,v.customer\_id, v.card\_number, v.expiration\_date,

v.security\_code, v.billing\_address, v.billing\_city,v.billing\_state,v.zipcode);

-- Account

MERGE INTO Credit\_Card\_Table\_dw a

USING Credit\_Card\_Table\_ar\_view b

ON (a.credit\_card\_id = b.credit\_card\_id)

WHEN MATCHED

THEN UPDATE SET

a.customer\_id = b.customer\_id,

a.card\_number = b.card\_number,

a.expiration\_date = b.expiration\_date,

a.security\_code = b.security\_code,

a.billing\_address = b.billing\_address,

a.billing\_city = b.billing\_city,

a.billing\_state = b.billing\_state,

a.zipcode = b.zipcode

WHEN NOT MATCHED

THEN INSERT (credit\_card\_id,customer\_id,card\_number,expiration\_date,

security\_code, billing\_address,billing\_city,billing\_state,zipcode)

VALUES (b.credit\_card\_id,b.customer\_id, b.card\_number, b.expiration\_date,

b.security\_code, b.billing\_address, b.billing\_city,b.billing\_state,b.zipcode);

END;

/

-------------------------------------------

-- Reservation

Create or replace procedure Reservation\_etl\_proc

As

Begin

-- Customer Reservation

MERGE INTO Reservation\_Table\_dw dw

USING Reservation\_Table\_CR v

ON (dw.reservation\_id = v.reservation\_id)

WHEN MATCHED

THEN UPDATE SET

dw.listing\_id = v.listing\_id,

dw.customer\_id = v.customer\_id,

dw.credit\_card\_id = v.credit\_card\_id,

dw.checkin\_date\_time = v.checkin\_date\_time,

dw.checkout\_date\_time = v.checkout\_date\_time,

dw.guest\_quantity = v.guest\_quantity,

dw.total\_amount = v.total\_amount,

dw.cancellation\_policy = v.cancellation\_policy

WHEN NOT MATCHED

THEN INSERT (reservation\_id,listing\_id,customer\_id,credit\_card\_id,

checkin\_date\_time, checkout\_date\_time,guest\_quantity,total\_amount,cancellation\_policy)

VALUES (v.reservation\_id,v.listing\_id, v.customer\_id, v.credit\_card\_id,

v.checkin\_date\_time, v.checkout\_date\_time, v.guest\_quantity,v.total\_amount,v.cancellation\_policy);

END;

/

-------------------------------

-- Account

CREATE OR REPLACE PROCEDURE Account\_Receivable\_Table\_ar\_etl\_proc

AS

BEGIN

-- Account

MERGE INTO Account\_Receivable\_Table\_dw a

USING Account\_Receivable\_Table\_ar b

ON (a.account\_id = b.account\_id)

WHEN MATCHED THEN

UPDATE SET

a.host\_id = b.host\_id,

a.routing\_number = b.routing\_number,

a.account\_number = b.account\_number,

a.account\_type = b.account\_type

WHEN NOT MATCHED THEN

INSERT(account\_id,host\_id,routing\_number,account\_number,account\_type)

VALUES(b.account\_id,b.host\_id, b.routing\_number, b.account\_number, b.account\_type);

END;

/

------------------------------

-- Invoice

CREATE OR REPLACE PROCEDURE Invoice\_Table\_ar\_etl\_proc

AS

BEGIN

-- Account

MERGE INTO Invoice\_Table\_dw a

USING Invoice\_Table\_ar\_view b

ON (a.invoice\_id = b.invoice\_id)

WHEN MATCHED THEN

UPDATE SET

a.host\_id = b.host\_id,

a.customer\_id = b.customer\_id,

a.invoice\_number = b.invoice\_number,

a.Invoice\_date = b.Invoice\_date,

a.item\_description = b.item\_description,

a.unit\_price = b.unit\_price,

a.quantity = b.quantity,

a.hosting\_address = b.hosting\_address,

a.hosting\_city = b.hosting\_city,

a.hosting\_state = b.hosting\_state,

a.hosting\_zip = b.hosting\_zip

WHEN NOT MATCHED THEN

INSERT (invoice\_id, host\_id, customer\_id, invoice\_number,

Invoice\_date, item\_description, unit\_price, quantity, hosting\_address,

hosting\_city, hosting\_state, hosting\_zip)

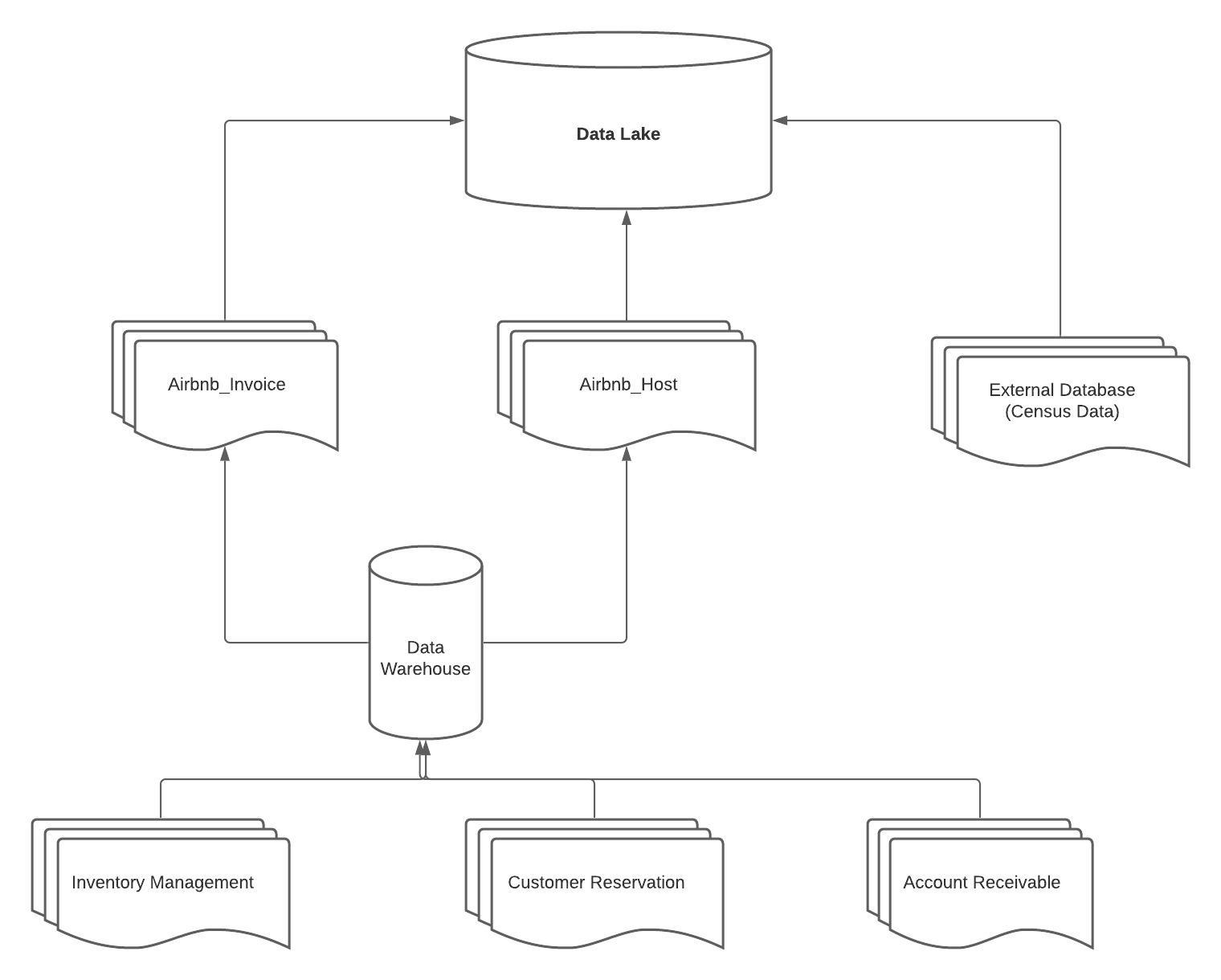
VALUES (b.invoice\_id, b.host\_id,b.customer\_id, b.invoice\_number, b.Invoice\_date,

b.item\_description, b.unit\_price, b.quantity,b.hosting\_address,b.hosting\_city, b.hosting\_state, b.hosting\_zip);

END;

/

# The model for your data lake



# The code for your analysis patterns

**Read data:**

host\_df = spark.read.format("csv").load("dbfs:/FileStore/shared\_uploads/mshch96@utexas.edu/DataLake\_Host.csv", header=True, inferSchema=True)

host\_df.show()

invoice\_df = spark.read.format("csv").load("dbfs:/FileStore/shared\_uploads/mshch96@utexas.edu/DataLake\_Invoice-2.csv", header=True, inferSchema=True)

invoice\_df.show()

Texas\_census\_df = spark.read.format("csv").load("dbfs:/FileStore/shared\_uploads/mshch96@utexas.edu/Additional\_Data-2.csv", header=True, inferSchema=True)

Texas\_census\_df.show()

**Convert DF to View:**

host\_df.createOrReplaceTempView("host")

invoice\_df.createOrReplaceTempView("invoice")

Texas\_census\_df.createOrReplaceTempView("texas\_census")

**Insights 1:**

%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;

**Insights 2:**

%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;

**Insights 3:**

%sql

SELECT superhost, avg(response\_rate)

FROM host

Group By superhost;