**Task 4 Submission:**

rm flight.db

sqlite3 flight.db

CREATE TABLE airports (

faa text primary key,

name text not null,

tzone text not null

)

;

.mode csv

.import /Users/naveenakesani/Documents/airports.csv airports --skip 1

SELECT \* FROM airports;

CREATE TABLE airlines (

carrier text primary key,

name text not null

)

;

.import /Users/naveenakesani/Documents/airlines.csv airlines --skip 1

SELECT \* FROM airlines;

CREATE TABLE flights (

id integer primary key autoincrement,

origin text not null,

destination text not null,

duration integer not null,

foreign key (origin) references airports(faa),

foreign key (destination) references airports(faa)

)

;

.import /Users/naveenakesani/Documents/flights.csv flights --skip 1

.mode markdown

SELECT \* FROM flights;

.table

.exit

ls flight.db

**Task 5 Submission:**

Update 1:

UPDATE airports

SET tzone = "America/Chicago"

WHERE tzone LIKE "American/Chicago";

SELECT tzone

FROM airports

WHERE tzone = "American/Chicago"

Update 2:

UPDATE flights

SET origin = "BQK", destination = “BQK”

WHERE origin LIKE "BQN" or destination LIKE “BQN”;

SELECT origin, destination

FROM flights

WHERE origin = "BQN" or destination = “BQN”;

Query 3:

Select count(\*) as Num\_of\_Records from airlines;

Select \* from airlines;

Query 4:

Select count(faa) as Number\_of\_FAA from airports;

Select \* from airports;

Query 5:

Select \* from airports

Where tzone = “America/Chicago”;

Query 6:

Select \* from flights

Where origin = “JFK”;

**Task 6 Submission:**

In terminal, in sqlite3:

Current status:

pragma foreign\_keys

;

Enable FK:

pragma foreign\_keys = 1

;

Insert into airports table:

INSERT INTO airports

(faa, name, tzone)

VALUES

("AMS", "AMS-Amsterdam Airport Schiphol", "Central European Summer Time");

Insert into flights:

INSERT INTO flights

(origin, destination, duration)

VALUES

("ORD", "AMS", 530)

;

Query 1:

SELECT airports.\*, flights.\*

FROM flights JOIN airports

ON flights.destination = airports.faa

WHERE destination = “ORD”;

**Task 7 Submission:**

Query 1:

SELECT \*

FROM flights

WHERE destination LIKE “%f%”;

Query 2:

SELECT \*

FROM flights

WHERE destination = “DFW” AND duration > 200;

Query 3:

SELECT airports.\*, flights.\*

FROM flights JOIN airports

ON flights.origin = airports.faa;

Run all queries in SQLite:

.open /Users/naveenakesani/Documents/flight\_db/flight.db

.headers on

.mode markdown

.read /Users/naveenakesani/Documents/flight\_db/Task7Queries.sql

Export to CSV file Q1:

.headers on

.mode csv

.separator ,

.once Task7Q1.csv

.read /Users/naveenakesani/Documents/flight\_db/Task7Q1.sql

Export to CSV file Q2:

.headers on

.mode csv

.separator ,

.once Task7Q2.csv

.read /Users/naveenakesani/Documents/flight\_db/Task7Q2.sql

Export to CSV file Q3:

.headers on

.mode csv

.separator ,

.once Task7Q3.csv

.read /Users/naveenakesani/Documents/flight\_db/Task7Q3.sql

**Task 8 Submission:**

Dump schema:

sqlite3 flight.db .schema > dump\_flight\_schema.sql

Replicate schema in new db:

rm flight\_empty.db

cat dump\_flight\_schema.sql | sqlite3 flight\_empty.db

When error pops up:

CREATE TABLE sqlite\_sequence(name,seq);

rm flight\_empty.db

cat dump\_flight\_schema.sql | sqlite3 flight\_empty.db

Run the following:

Cd Documents

Cd flight\_db

sqlite3 flight.db .schema > dump\_flight\_schema.sql

rm flight\_empty.db

cat dump\_flight\_schema.sql | sqlite3 flight\_empty.db

-- comment out the last create table statement

rm flight\_empty.db

cat dump\_flight\_schema.sql | sqlite3 flight\_empty.db

sqlite3 flight.db .dump > dump\_flight.sql

cat dump\_flight.sql | sqlite3 flight\_replica.db

sqlite3

.open /Users/naveenakesani/Documents/flight\_db/flight\_replica.db

.mode markdown

.table

select \* from airlines;

select \* from airports;

select \* from flights;

**Database Redesign:**

Create Database:

sqlite3 flight2.db

CREATE TABLE airports (

faa text primary key,

name text not null,

tzone text not null

)

;

CREATE TABLE airlines (

carrier text primary key,

name text not null

)

;

CREATE TABLE flights (

id integer primary key autoincrement,

origin text not null,

destination text not null,

duration integer not null,

airlinecarrier text not null,

foreign key (airlinecarrier) references airlines(carrier),

foreign key (origin) references airports(faa),

foreign key (destination) references airports(faa)

)

;

CREATE TABLE passengers (

passengerid integer primary key autoincrement,

fullname text not null,

flightid integer not null,

carrier text not null,

foreign key (flightid) references flights(id),

foreign key (carrier) references airlines(carrier)

)

;

Populate database:

.mode csv

.import /Users/naveenakesani/Documents/airports.csv airports --skip 1

.mode csv

.import /Users/naveenakesani/Documents/airlines.csv airlines --skip 1

.mode csv

.import /Users/naveenakesani/Documents/flights.csv flights --skip 1

.mode csv

.import /Users/naveenakesani/Documents/passengers.csv passengers --skip 1

Check the tables:

.table

.mode markdown

SELECT \* FROM airports;

SELECT \* FROM airlines;

SELECT \* FROM flights;

SELECT \* FROM passengers;

Database dump:

Cd Documents

Cd flight\_db

sqlite3 flight2.db .schema > dump\_flight2\_schema.sql

rm flight2\_empty.db

cat dump\_flight2\_schema.sql | sqlite3 flight2\_empty.db

-- comment out the last create table statement

rm flight2\_empty.db

cat dump\_flight2\_schema.sql | sqlite3 flight2\_empty.db

sqlite3 flight2.db .dump > dump\_flight2.sql

cat dump\_flight2.sql | sqlite3 flight2\_replica.db

sqlite3

.open /Users/naveenakesani/Documents/flight\_db/flight2\_replica.db

.mode markdown

.table

select \* from airlines;

select \* from airports;

select \* from flights;

select \* from passengers;

Query 1:

SELECT f.id, f.origin, f.destination, f.airlinecarrier, count(distinct p.passengerid) as Num\_Of\_Passengers

FROM passengers p JOIN flights f ON p.flightid = f.id

GROUP BY f.id, f.origin, f.destination, f.airlinecarrier

ORDER BY 1;

Query 2:

SELECT a.\*, count(DISTINCT f.id) as Num\_Of\_Flights

FROM airlines a JOIN flights f ON a.carrier = f.airlinecarrier

GROUP BY a.carrier

ORDER BY 1;

Query 3:

SELECT b.faa, b.name, b.tzone, count(f.origin) as Num\_Of\_Flights

FROM airports b JOIN flights f ON b.faa = f.origin

GROUP BY b.faa

ORDER BY 4 DESC

LIMIT 3;

Query 4:

SELECT b.\*, count(f.origin) as Num\_Of\_Flights

FROM airports b JOIN flights f ON b.faa = f.origin

GROUP BY b.faa

ORDER BY 1;

Query 5:

SELECT p.passengerid, p.fullname, f.id, f.origin, f.destination

FROM passengers p JOIN flights f ON p.flightid = f.id

ORDER BY 1;

Query 6:

SELECT p.passengerid, p.fullname, f.id, f.origin, b.name, f.destination, b2.name

FROM passengers p JOIN flights f ON p.flightid = f.id

JOIN airports b ON b.faa = f.origin

JOIN airports b2 ON b2.faa = f.destination

GROUP BY p.passengerid, p.fullname, f.id, f.origin, b.name, f.destination, b2.name

ORDER BY 1;

Query 7:

SELECT p.passengerid, p.fullname, f.id as flight\_id, f.origin, f.destination, f.duration as duration\_minutes

FROM passengers p JOIN flights f ON p.flightid = f.id

ORDER BY 1;