Questions:

1. Two new flights are scheduled to leave from *BOS* to *ORD* airports.
   * The first flight (*flightId 1111*) will be on airplane *AP098640* departing on-time at 2018-11-05 12:00:00 and arriving on-time at 2018-11-05 14:30:00.
   * The second flight (*flightId 1112*) will be on airplane *AP432379* departing on-time at 2018-11-05 22:00:00 and arriving on-time at 2018-11-06 00:30:00.

**SQL QUERY:**

set xact\_abort on;

begin transaction addtrans1

insert into FlightRoute\_t(flightNumber,departAirport,arriveAirport,scheduledDepartTime,scheduledArrivalTIme)

values(1111,'BOS','ORD','12:00:00','14:30:00'),(1112,'BOS','ORD', '22:00:00','00:30:00')

insert into FlightSchedule\_t(flightNumber,flightDate,statusID,airplaneID,delayDepartTime,delayArrivalTime)

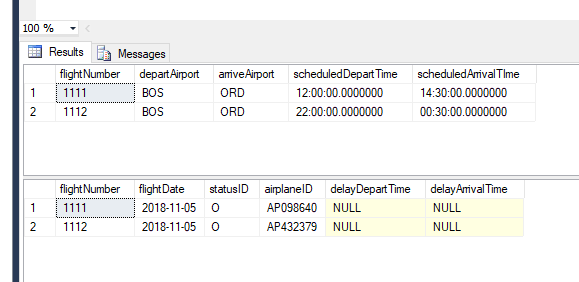
values(1111,'2018-11-05','O','AP098640',NULL,NULL),(1112,'2018-11-05','O','AP432379',NULL,NULL)

commit transaction addtrans1;

SELECT \* FROM FlightRoute\_t where flightNumber='1111' or flightNumber='1112'

SELECT \* FROM FlightSchedule\_t where flightNumber='1111' or flightNumber='1112'

**SCREENSHOT:**



1. Due to forecasted inclement weather, the new flights’ **statuses** have changed.
   * The earlier new flight from *BOS* to *ORD (flightId 1111)* has been ***delayed*** by 10 hours.
   * The later new flight from *BOS* to *ORD (flightId 1112)* hasbeen ***canceled***.
   * **NOTE:** DO NOT commit this transaction *(i.e. COMMIT TRANSACTION …;)*

**SQL QUERY:**

set xact\_abort on;

begin transaction updatetrans1

update FlightSchedule\_t

set FlightSchedule\_t.statusID='D'

from FlightSchedule\_t

where flightNumber=1111;

update FlightSchedule\_t

set statusID='C'

from FlightSchedule\_t

where flightNumber=1112;

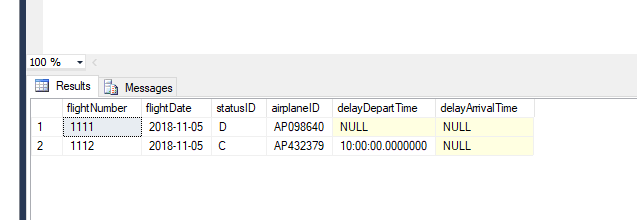
update FlightSchedule\_t

set delayDepartTime = '10:00:00.0000000'

from FlightSchedule\_t

where flightNumber=1112;

**SCREENSHOT:**

****

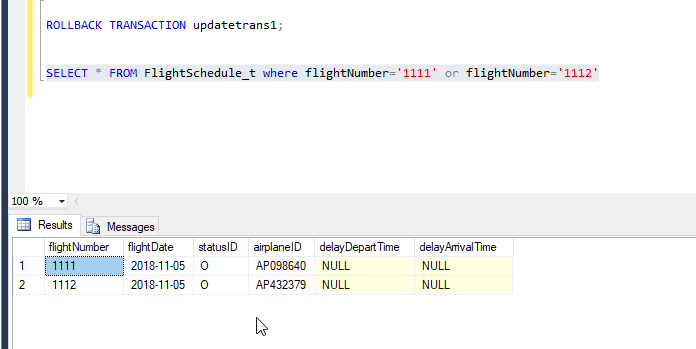
1. The weather has improved and the two flight are back to the original statuses (*question 1*).
   * ROLLBACK the last transaction (2)
   * ***NOTE****: If you committed the transaction in (2), you’ll be unable to Rollback.* Revert instead using a new transaction with update statements.

**SQL QUERY:**

ROLLBACK TRANSACTION updatetrans1;

SELECT \* FROM FlightSchedule\_t where flightNumber='1111' or flightNumber='1112'

**SCREENSHOT:**

****

1. There is a new flightRoute between Portland, OR (*cityID C010*) and Seattle, WA (*cityID C011*).
   * The Portland airport is airportId *PDX* and is named *Portland International Airport*.
   * The Seattle airport is airportId *SEA* and is named *Seattle–Tacoma International Airport*.
   * The flight from PDX to SEA *(flightID 1003*) is scheduled to depart at *08:00:00 and arrive at 09:15:00.*
   * The flight from SEA to PDX (*flightID 1004*) is scheduled to depart at *14:00:00 and arrive at 15:15:00*.
   * ***NOTE:*** *No flights have been scheduled yet for this flightRoute.*

**SQL QUERY:**

set xact\_abort on;

begin transaction addtrans3

insert into states\_t values ('OR','Oregon');

insert into states\_t values ('WA','Washington');

insert into City\_t values ('C010','OR','Portland');

insert into City\_t values ('C011','WA','Seattle');

INSERT INTO Airport\_t VALUES ('PDX','C010','Portland International Airport');

insert into Airport\_t values ('SEA','C011','Seattle–Tacoma International Airport');

INSERT INTO FlightRoute\_t VALUES('1003','PDX','SEA','08:00','09:15');

INSERT INTO FlightRoute\_t VALUES('1004','SEA','PDX','14:00','15:15');

commit transaction addtrans3;

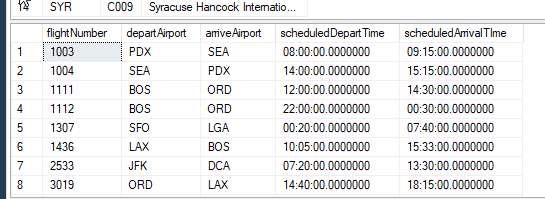
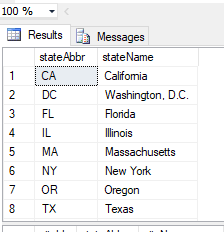
select \* from states\_t;

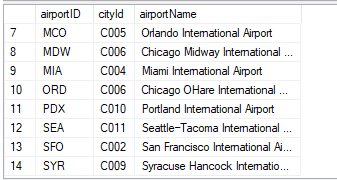
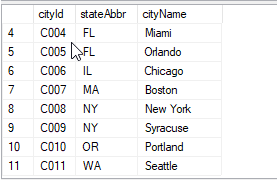
select \* from City\_t;

select \* from Airport\_t;

select \* from FlightRoute\_t;

**SCREENSHOT:**

****

****

Views

Directions:

Create a view based on the criteria in question (1). Take a screenshot of the SQL query which generated the view and take a screenshot of the result table when you **SELECT \* FROM *view***.

1. An operations analyst is interested in understanding flight capacity at each airport. In order to do so, they need to continually monitor how many departing and arriving flights occur at each airport.

To assist the analyst, create a **VIEW** called *Airport\_Capacity* with the following columns:

* *airportId, airportName, cityId, cityName, stateAbbr, StateName*
* View contains the total number of departing and arriving flights for each airport
* The view **excludes** airports which have no departing and arriving flights

**SQL QUERY:**

go

create view view1 as

select t3.airportID,t3.airportName,t3.cityId,t2.cityName,t2.stateAbbr,t1.stateName

from states\_t as t1

inner join City\_t t2 on t1.stateAbbr = t2.stateAbbr

inner join Airport\_t t3 on t2.cityId = t3.cityId;

go

go

create view view2 as

select COUNT(flightNumber) as departing,departAirport from FlightRoute\_t group by departAirport;

go

go

create view view3 as

select COUNT(flightNumber) as arriving, arriveAirport from FlightRoute\_t group by arriveAirport;

go

go

create view Airport\_Capacity as

select t1.\*,t2.departing,t3.arriving from view1 as t1

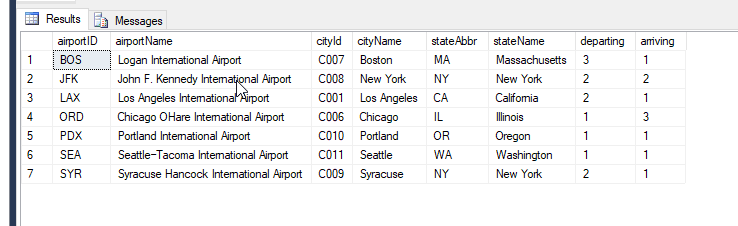
inner join view2 as t2 on t1.airportID=t2.departAirport

inner join view3 as t3 on t1.airportID= t3.arriveAirport;

go

select \* from

Airport\_Capacity;

**SCREENSHOT:**

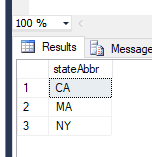
1. Query the *Airport\_Capacity* view and return the distinct states (*DISTINCT stateAbbr*) where the number of departures is greater than or equal to the number of arrivals.

**SQL QUERY:**

select distinct(stateAbbr) from Airport\_Capacity

where departing > arriving;

**SCREENSHOT:**

****