

CSCI-341 Database Systems

Fall 2019 Test 2-1, Total points: 120, plus 5 extra credit

Part 1 - SQL: (80pts total) For each of the following problems, write a single SQL statement to do what is asked. Read carefully and work carefully. All questions refer to the airline database whose relational model is attached. The MySQL function to get the current date/time is called *now()*.

1. (3pts) List the flight number, leg number and scheduled arrival time of all **flight legs** that arrive at the airport whose code is 'TSE'.

```
select FlightNum, LegNum, SchedArrTime
from FLIGHT_LEG
where ArrAirportCode = 'TSE';
```

2. (3pts) Bill Smith is on the no-fly list. Delete all *future* reservations of people named 'Bill Smith'.

```
delete from SEAT_RESERVATION
where CustomerName like 'Bill Smith'
and Date > now();
```

3. (4pts) The airline named 'Air Astana' wants to be more competitive. Reduce all the fares of this airline by 15%.

```
update FARE
set Amount = 0.85 * Amount
where FlightNum in (select FlightNum
                    from FLIGHT
                    where Airline like 'Air Astana');
```

4. (4pts) List all airplane types who are manufactured by **companies** whose company name begins with the letter 'B', and show the results in alphabetical order.

```
select Name
from AIRPLANE_TYPE
where Company like 'B%'
order by Name;
```

5. (4pts) Count the number of **flight legs** that arrive in a city named 'Almaty'.

```
select count(*)
from FLIGHT_LEG FL, AIRPORT A
where FL.ArrAirportCode = A.Code
and City like 'Almaty';
```

6. (6pts) Find the flight number, airline and minimum fare amount for each flight, and sort the results by the airline name.

```
select FL.FlightNum, FL.Airline, min(F.Amount)
from FLIGHT FL, FARE F
where FL.FlightNum = F.FlightNum
group by FL.FlightNum
order by FL.Airline;
```

7. (5pts) Find the **flight legs** (flight number and leg number) that either arrive at or depart from the airport whose code is 'ORF'.

```
select FlightNum, LegNum
from FLIGHT_LEG
where ArrAirportCode = 'ORF'
or DeptAirportCode = 'ORF';
```

8. (7pts) For each flight leg, list the flight number, leg number, departure **city** and arrival **city**, and sort the results by the departure **city** name.

```
select FL.FlightNum, FL.LegNum,
       AC1.City as 'DeptCity', AC2.City as 'ArrCity'
from FLIGHT_LEG FL, AIRPORT AC1, AIRPORT AC2
where FL.ArrAirportCode = AC2.Code
and FL.DeptAirportCode = AC1.Code
order by AC1.City;
```

9. (5pts) Create a list of **all** of the airplane types, and, for each one, give the type name, max seats, company, and the count of the number of airports it can land at.

```
select AT.Name, AT.MaxSeats, AT.Company, count(*)
from AIRPLANE_TYPE AT left outer join CAN_LAND CL
on AT.Name = CL.AirplaneTypeName
group by AT.Name;
```

10. (7pts) For each flight leg, find the flight number, leg number and the **average number of seats filled** on this leg.

```
select FL.FlightNum, FL.LegNum, avg(A.TotalSeats - LI.AvailableSeats)
from FLIGHT_LEG FL, LEG_INSTANCE LI, AIRPLANE A
where FL.FlightNum = LI.FlightNum
and FL.LegNum = LI.LegNum
and LI.AirplaneID = A.ID
group by FL.FlightNum, FL.LegNum;
```

11. (7pts) Find the airplane types that can land at airport 'TSE' but which have **never** landed there.

```
select CL.AirplaneTypeName
from CAN_LAND CL
where CL.AirportCode = 'TSE'
and not exists (select AirplaneTypeName
                from LEG_INSTANCE LI, AIRPLANE A
                where LI.AirplaneID = A.ID
                and A.AirplaneTypeName = CL.AirplaneTypeName
                and LI.ArrAirportCode = 'TSE');
```

12. (11pts) For **flight legs**, list the airline, flight number, leg number and the count of the number of times that a flight leg has arrived late, but include only flight legs that have had less than 3 late arrivals.

```
select F.Airline, FL.FlightNum, FL.LegNum, count(*) as 'Late'
from FLIGHT F, FLIGHT_LEG FL, LEG_INSTANCE LI
where F.FlightNum = FL.FlightNum
and FL.FlightNum = LI.FlightNum
and FL.LegNum = LI.LegNum
and FL.SchedArrTime < LI.ActualArrTime
group by FlightNum, LegNum
having Late < 3;
```

13. (4pts) Find the flight number and leg number of every **flight leg** that arrives at an airport where airplanes of type '737Max' can land.

```
select FL.FlightNum, FL.LegNum
from FLIGHT_LEG FL, CAN_LAND CL
where FL.ArrAirportCode = CL.AirportCode
and AirplaneTypeName = '737Max';
```

14. (10pts) Find the flight number, leg number and date of all **future** flight (legs) that 'Hillary Clinton' and 'Donald Trump' will take together.

```
select LI.FlightNum, LI.LegNum, LI.Date
from LEG_INSTANCE LI, SEAT_RESERVATION SR1, SEAT_RESERVATION SR2
where LI.FlightNum = SR1.FlightNum
and LI.LegNum = SR1.LegNum
and LI.Date = SR1.Date
and LI.FlightNum = SR2.FlightNum
and LI.LegNum = SR2.LegNum
and LI.Date = SR2.Date
and SR1.CustomerName = 'Hillary Clinton'
and SR2.CustomerName = 'Donald Trump'
and LI.Date > now();
```

BONUS: For each flight leg, list the flight number, leg number, arrival airport code and the percentage of times that the flight leg arrived late.

```
select TA.FlightNum, TA.LegNum, TA.ArrAirportCode,
       ifnull(late_flights,0)/all_flights*100
from (select FL.FlightNum, FL.LegNum, FL.ArrAirportCode,
            count(*) as all_flights
      from FLIGHT_LEG FL, LEG_INSTANCE LI
      where FL.FlightNum = LI.FlightNum
      and FL.LegNum = LI.LegNum
      group by FL.FlightNum, FL.LegNum) as TA
left outer join (select FL.FlightNum, FL.LegNum, FL.ArrAirportCode,
                      count(*) as late_flights
                from FLIGHT_LEG FL, LEG_INSTANCE LI
                where FL.FlightNum = LI.FlightNum
                and FL.LegNum = LI.LegNum
                and SchedArrTime < ActualArrTime
                group by FL.FlightNum, FL.LegNum) as TL
on TA.FlightNum = TL.FlightNum and TA.LegNum = TL.LegNum;
```

Part 2 - Database Models: (40pts) Look at the attached ER model and draw the relational model that corresponds to it using the same relational model notation that the attached airline model is drawn in.

CLINIC [CID, DID(FK)]
CLINIC_PHONE [CID, ClinicPhoneNumber]
DOCTOR [DID, DoctorName]
TREATS [DID(FK), PID(FK)]
PATIENT [PID, PatientEmail, DID(FK)]
CHILD [PID, ChildName]



