**INFSCI 2710 Database Management, Fall 2018**

**Homework 1: Relational Algebra, SQL SOLUTION**

**100 pts**

Consider the following schema:

city (id, city, state, country)

planes (plane-number, model, capacity, create-year)

pilot (ssn, home-city, fullname, day-of-birth, month-of-birth, year-of-birth, salary)

flight (flight-number, captain-ssn, plane-number, takeoff-city, landing-city, time-takeoff, time-landing)

Underlines attributes are the primary keys; attributes in italic style are foreign keys.

**Hint**: In relational algebra question joining tables without attributes of same name, you must specify matching condition explicitly, e.g.:

**Q1 [20 pt]** Write SQL DDL statements to create the above tables. Make sure that you capture the primary and foreign key constraints (if applicable), choose appropriate domain (data) type and constraints for each attribute.

Answer:

CREATE TABLE city   
  (   
     id      *INTEGER* NOT NULL,   
     city    *CHAR*(100) NOT NULL,   
     state   *CHAR*(200) NULL,   
     country *CHAR*(100) NOT NULL,   
     PRIMARY KEY (id)   
  );   
  
CREATE TABLE planes   
  (   
     plane-number *INT* NOT NULL,   
     model        *CHAR*(45) NOT NULL,   
     capacity     *INTEGER* NOT NULL,   
     create-year  *INTEGER* NOT NULL,   
     PRIMARY KEY (plane-number)   
  );   
  
CREATE TABLE pilot   
  (   
     ssn            *INTEGER* NOT NULL,   
     home-city      *INT* NOT NULL,   
     fullname       *CHAR*(200) NOT NULL,   
     day-of-birth   *INTEGER* NOT NULL,   
     month-of-birth *INTEGER* NOT NULL,   
     year-of-birth  *INTEGER* NOT NULL,   
     salary         *INTEGER* NOT NULL,   
     PRIMARY KEY (ssn),   
     FOREIGN KEY (home-city) REFERENCES city (id)    
  );   
  
CREATE TABLE flight   
  (   
     flight-number *INTEGER* NOT NULL,   
     captain-ssn   *INTEGER* NOT NULL,   
     plane-number  *INTEGER* NOT NULL,   
     takeoff-city  *INTEGER* NOT NULL,   
     landing-city  *INTEGER* NOT NULL,   
     time-takeoff  *DATETIME* NOT NULL,   
     time-landing  *DATETIME* NOT NULL,   
     PRIMARY KEY (flight-number),   
     FOREIGN KEY (captain-ssn) REFERENCES pilot (ssn) ,   
     FOREIGN KEY (plane-number) REFERENCES planes (plane-number) ,   
     FOREIGN KEY (takeoff-city) REFERENCES city (id) ,   
     FOREIGN KEY (landing-city) REFERENCES city (id)    
  );

**Q2 [10 pt]** Specify a relational algebra expression and an SQL query to find all cities in Germany.

Answer:

SELECT \*   
FROM   city   
WHERE  country = 'Germany'

**Q3 [10 pt]** Specify a relational algebra and a SQL expression to find all pilots who **live** in Germany.

Answer:

SELECT pilot.\*   
FROM   city, pilot   
WHERE  country = 'Germany'   
AND pilot.home-city = city.id

**Q4 [10 pt]** Specify a relational algebra and an SQL expressions to find all pilots who **fly** to **or**

from Germany. Answer:

SELECT pilot.\*   
FROM   pilot,flight,city   
WHERE  flight.captain-ssn = pilot.ssn   
       AND flight.takeoff-city = city.id   
       AND city.country = 'Germany'   
UNION   
(SELECT pilot.\*   
 FROM   pilot,flight,city   
 WHERE  flight.captain-ssn = pilot.ssn   
        AND flight.landing-city = city.id   
        AND city.country = 'Germany')

**Q5 [10 pt]** Specify a relational algebra expression and an SQL query to find plane models that serve (fly to and from) only US cities.

Answer:

SELECT \*

FROM planes, flight, city

WHERE flight.plane-number = planes.plane-number

AND (flight.landing-city = city.id OR flight.takeoff-city = city.id )

AND city.country = 'us'

AND planes.model not IN (

SELECT planes.model

FROM planes, flight, city

WHERE flight.plane-number = planes.plane-number

AND (flight.landing-city = city.id OR flight.takeoff-city = city.id )

AND city.country <> 'us'

)

**Q6 [15 pt]** Specify a relational algebra expression and an SQL query to find plane models that serve both US and international cities.

Answer:

SELECT \*

FROM planes, flight, city

WHERE flight.plane-number = planes.plane-number

AND ( flight.landing\_city = city.id

OR flight.takeoff\_city = city.id )

AND city.country = 'us'

AND planes.model IN (

SELECT planes.model

FROM planes, flight, city

WHERE flight.plane-number = planes.plane-number

AND ( flight.landing\_city = city.id OR flight.takeoff\_city = city.id )

AND city.country <> 'us'

)

**Q7 [10 pt]** Consider the following relational algebra expressions:

1. Select the expression(s) from above that return the name of the pilot and the country where he or she is from.

Answer: b, c

1. Select expressions from above that are equivalent (e.g. return the same answer).

Answer: (b, c); (a, d)

**Q8 [15 pt]** Consider the following relational algebra expression:

1. How many attributes will the result have?

Answer: 14 (plane-number will only appear once, because of the natural join)

1. Write in English what question the expression is trying to answer (e.g. describe what would be the result of the expression).

Answer: Find information of planes and flights that depart from Pittsburgh.

1. Write an equivalent version of the relational algebra expression that is the most efficient. Specify why your expression is more efficient.

Answer:

1. Translate all expressions into SQL.

Answer:

SELECT \*   
FROM   city,   
       flight,   
       planes   
WHERE  city = ’pittsburgh’   
       AND city.id = flight.takeoff-city   
       AND planes.plane-number = flight.plane-number