

1. List the name, age and EID of employees who are certified to fly *all* aircraft (AID) in the database that have a cruising range greater than 3000 miles.

```
AircraftOver3000 <- PI_AID (SIGMA_CruisingRange > 3000)
CertifiedEmp <- PI_EID(CERTIFIED_TO_FLY DIVIDE AircraftOver3000)
Result <- PI_EName, Age, EID (EMPLOYEES JOIN CertifiedEmp)
```

2. List the names and age of all pilots who have *never* had a maintenance issue on a flight where they were the pilot (415/615)

```
FD <- FLIGHTDETAILS
MI <- MAINTENANCE_ISSUE
IssueEID <- RHO(EID) PI_FD.PilotEID(FD JOIN MI)
GoodEID <- PI_EID (EMPLOYEES) - IssueEID
Result <- PI_EName, Age (EMPLOYEES JOIN GoodEID)
```

List the names and age of all pilots who have *ever* had a maintenance issue on a flight where they were the pilot (315)

```
PilotsWithMaintenance ← πPilotEID (FLIGHTDETAILS ⋈ MAINTENANCE_ISSUE)
PilotsWithMaintenance ← ρPilotEID -> EID (PilotsWithMaintenance)
Result <- πEName, Age (PilotsWithMaintenance ⋈ EMPLOYEES)
```

3 (315):

```
JARED_CERTIFIED <- PI_(AID) SELECT_(EName = "Jared Eisner") (EMPLOYEES
NATURAL_JOIN CERTIFIED_TO_FLY)
```

```
AIRCRAFT_NOTCERT <- PI_(AID) AIRCRAFT – JARED_CERTIFIED
```

```
RESULT <- PI_(EName, AName, AID) AIRCRAFT_NOTCERT NATURAL_JOIN AIRCRAFT
NATURAL_JOIN EMPLOYEES
```

3 (415):

ALL_PAIRS <- PI (EID, AID) (EMPLOYEES X AIRCRAFT)

CANT_FLY <- ALL_PAIRS – CERTIFIED_TO_FLY

RESULT <- PI (EName, AID, AName) (CANT_FLY NATURAL_JOIN EMPLOYEES
NATURAL_JOIN AIRCRAFT)

4 (Both):

{ t | EXISTS a IN AIRCRAFT (a[AName] = "Boeing787" ^

EXISTS fd IN FLIGHTDETAILS (fd[PlaneAID] = a[AID] ^

EXISTS f1 IN FLIGHTS (f1[FLNO] = fd[FNO] ^

EXISTS f2 IN FLIGHTS (f2[Dist] < 2000 ^

f1[FromCode] = f2[FromCode] ^

f1[ToCode] = f2[ToCode])))) ^

t[FromCode] = f1[FromCode] ^ t[ToCode] = f2[ToCode] ^ t[Distance] = f2[Distance]}


```
WHERE AIRCRAFT.AID = MAINTENANCE_ISSUE.PlaneAID
GROUP BY AIRCRAFT.AName ) as R )
```

5b: 415/615)

```
SELECT Typelssues.AName, Typelssues.Total
FROM ( SELECT MAX(Total) AS Total
FROM ( SELECT AIRCRAFT.AName, COUNT(MAINTENANCE_ISSUE.PlaneAID) AS
Total
FROM AIRCRAFT, MAINTENANCE_ISSUE
WHERE AIRCRAFT.aid = MAINTENANCE_ISSUE.PlaneAID
GROUP BY AIRCRAFT.AName ) AS TotalIssues ) AS MaxIssues,
( SELECT AIRCRAFT.AName, COUNT(MAINTENANCE_ISSUE.PlaneAID) AS Total
FROM AIRCRAFT, MAINTENANCE_ISSUE
WHERE AIRCRAFT.aid = MAINTENANCE_ISSUE.PlaneAID
GROUP BY AIRCRAFT.AName ) AS Typelssues
WHERE MaxIssues.Total = Typelssues.Total
```

5c: 315)

```
SELECT EMPLOYEES.ENAME, EMPLOYEES.Age, C.c
FROM EMPLOYEES, (SELECT MAX(c) AS m
FROM (SELECT SUM(FLIGHTS.Distance) AS c
FROM FLIGHTS, CODE_NAMES, FLIGHTDETAILS,
WHERE FLIGHTDETAILS.FNO = FLIGHTS.FLNO
AND FLIGHTS.FromCode = CODE_NAMES.Code
AND CODE_NAMES.StateName = "Maryland"
```

```

        GROUP BY FLIGHTDETAILS.PilotEID) as B),
    (SELECT D.PilotEID, SUM(F.Distance) AS c
     FROM FLIGHTS, CODE_NAMES, FLIGHTDETAILS,
     WHERE FLIGHTDETAILS.FNO = FLIGHTS.FLNO
     AND FLIGHTS.FromCode = CODE_NAMES.Code
     AND CODE_NAMES.StateName = "Maryland"
     GROUP BY FLIGHTDETAILS.PilotEID) as C
WHERE C.c = B.m
AND EMPLOYEES.EID = C.PilotEID;

```

5c: 415/615)

```

SELECT T.ENAME, T.AGE, T.sum_distance
FROM (SELECT MAX(MarylandInfo.MarylandDistance) AS MaxDistance
      FROM (SELECT SUM(FLIGHTS.Distance) AS MarylandDistance, EMPLOYEES.EID
            FROM FLIGHTS, FLIGHTDETAILS, EMPLOYEES, CODE_NAMES
            WHERE CODE_NAMES.StateName = "Maryland"
            AND (CODE_NAMES.Code = FLIGHTS.FromCode
                 OR CODE_NAMES.Code = FLIGHTS.ToCode)
            AND FLIGHTS.FLNO = FLIGHTDETAILS.FNO
            AND FLIGHTDETAILS.PilotEID = EMPLOYEES.EID
            GROUP BY EMPLOYEES.EID ) AS MarylandInfo) AS MAX,
 (SELECT SUM(FLIGHTS.Distance) AS MarylandDistance, EMPLOYEES.EID,
  EMPLOYEES.ENAME, EMPLOYEES.AGE
  FROM FLIGHTS, FLIGHTDETAILS, EMPLOYEES, CODE_NAMES
  WHERE CODE_NAMES.StateName = "Maryland"
  AND (CODE_NAMES.Code = FLIGHTS.FromCode

```

```
OR CODE_NAMES.Code = FLIGHTS.ToCode)
AND FLIGHTS.FLNO = FLIGHTDETAILS.FNO
AND FLIGHTDETAILS.PilotEID = EMPLOYEES.EID
GROUP BY EMPLOYEES.EID ) AS MarylandInfo
WHERE MarylandInfo.MarylandDistance = MAX.MarylandDistance
```

5d: 315)

```
SELECT FromCode, ToCode, min(Price) as minDirectPrice
FROM FLIGHTS
GROUP BY FromCode, ToCode
```

5d: 415)

```
SELECT MIN(F.Price)
FROM FLIGHTS F
WHERE F.Distance > 5000
GROUP BY F.FromCode, F.ToCode
```

6a: (Both)

FLIGHTS	FLNO	FromCode	ToCode	DepTime	ArrTime	Price	Distance
	_f1 _f2	'BWI' _c	_c 'SFO'	_t2	_t1		_d1 _d2

CODE_NAMES	Code	CityName	StateName
	_c	_cityname	

EMPLOYEES	EID	EName	Age	Salary

AIRCRAFT	AID	AName	CruisingRange

CERTIFIED_TO_FLY	EID	AID

RESULT	FlightNum1	FlightNum2	Connection	Distance	
P.	_f1	_f2	_cityname	_d1+_d2	

Conditions
_t2 > _t1 AND _d1 + _d2 < 3200

6b: 315) (6 points) A customer wants to fly from BWI to SFO on exactly 3 connecting flights (e.g. a flight from BWI to ORD, another from ORD to DEN and another flight from DEN to SFO). List all 3-flight options from BWI to SFO including their departure time from BWI, their arrival time in SFO and both the airport codes and name of the first city where the flights connect.

ANSWER IN QBE

QBE Shortcut: \neg -> NOT

FLIGHTS	<u>FLNO</u>	FromCode	ToCode	DepTime	ArrTime	Price	Distance
	_f1 _f2 _f3	BWI _t1 _t2	_t1 _t2 SFO	_d1 _d2 _d3	_a1 _a2 _a3		

CODE_NAMES	Code	CityName	StateName
	_t1	_name	

EMPLOYEES	EID	EName	Age	Salary

AIRCRAFT	AID	AName	CruisingRange

RESULT	FLNOs	Code	City	DepTime	ArrTime
P.	(_f1, _f2, _f3)	_t1	_name	_d1	_a3

Conditions
$D1 < a1 < d2 < a2 < d3 < a3$

6b: 415/615) (6 points) A customer wants to fly from BWI to SFO on 2 or 3 connecting flights (e.g. a flight from BWI to ORD, another from ORD to DEN and another flight from DEN to SFO). List all 2 or 3-flight options from BWI to SFO including their departure time from BWI, their arrival time in SFO and both the airport codes and name of the first city where the flight connects. ANSWER IN QBE

QBE Shortcut: \neg -> NOT

FLIGHTS	<u>FLNO</u>	FromCode	ToCode	DepTime	ArrTime	Price	Distance
	_f1	BWI	_a	_dep	_t1		
	_f2	_a	_b	_t2	_t3		
	_f3	_b	SFO	_t4	_arr1		
	_f4	_a	SFO	_t5	_arr2		

CODE_NAMES	Code	CityName	StateName
	_a	_name	

EMPLOYEES	EID	EName	Age	Salary

AIRCRAFT	AID	AName	CruisingRange

RESULT	Flights	Code	CityName	Departure	Arrival
P.	(_f1, _f2, _f3) (_f1, _f4)	_a	_name	_dep	_arr1 _arr2

Conditions
_dep < _t1 < _t2 < _t3 < _t4 < _t5 and _t4 < _arr1 and _t5 < arr2

6c: 315) . (6 points) List the city, state and airport code of cities in California which are possible to reach from BWI with 2 flights (one connection) but for which HopAir offers no nonstop flights.

QBE Shortcut: \neg -> NOT

FLIGHTS	<u>FLNO</u>	FromCode	ToCode	DepTime	ArrTime	Price	Distance
NOT		BWI _a BWI	_a _b _b				

CODE_NAMES	Code	CityName	StateName
P.	_b	_name	California

EMPLOYEES	EID	EName	Age	Salary

AIRCRAFT	AID	AName	CruisingRange

CERTIFIED_TO_FLY	EID	AID

RESULT					

Conditions

6c: 415/615) (6 points) List the flight number of all nonstop flights from a city in Maryland to a city in California which could possibly be piloted by an certified employee who is less than 23 years old or makes less than 20,000 a year.

QBE Shortcut: \neg -> NOT

FLIGHTS	<u>FLNO</u>	FromCode	ToCode	DepTime	ArrTime	Price	Distance
	P. _fn	_a	_b				_dis

CODE_NAMES	Code	CityName	StateName
	_a _b		Maryland California

EMPLOYEES	EID	EName	Age	Salary
	_e		_age	_sal

AIRCRAFT	AID	AName	CruisingRange
	_a		_cruise

CERTIFIED_TO_FLY	EID	AID
	_e	_a

Conditions
_cruise > _dis and (_age < 23 or _sal (this should be weeklySalary) * 52 < 20000

7a (Both 315 and 415/615)

	Derivable from FD's above (type YES/NO) You don't need to give a justification or derivation.
$B \rightarrow C$?	YES
$A \rightarrow C$?	NO
$AB \rightarrow A$?	YES
$ABD \rightarrow E$?	YES
$ABC \rightarrow D$?	YES
$AB \rightarrow E$?	YES
$AC \rightarrow E$?	NO
$A \rightarrow E$?	NO
$B \rightarrow E$?	YES
$C \rightarrow E$?	NO

7b

$AB \rightarrow C$

$B \rightarrow D$

$AB \rightarrow CD \rightarrow E$

A cannot derive B and B cannot derive A

Thus the candidate key is AB