Solutions for Exercise 1: Schema Fragmentation

Solution for 1(a)

	Application 1 Row_PK[,Row_PK...] (Monthly Access)	Application 2 Row_PK[,Row_PK...] (Monthly Access)
Q1	1,3,6,7 (70)	5,8 (100)
Q2	1,2(100)	1,2 (100)
Q3	3,6,7(60)	3,4,6 (100)
Q4	*2, 4, 5, 8 (30)*	*7 (100)*
Q5	*0*	*1,3,6,7(20)*

Solution for 1(b)

The tuple access frequencies by both applications can be summarized as follows:

App	1	2	3	4	5	6	7	8
A1	170	130	130	30	30	130	130	30
A2	120	100	120	100	100	120	120	100

The tuples with uniform access frequencies by both applications should be placed in the same fragment. Thus, the above table can be partitioned into 4 fragments. The list of fragments with member tuples is:

- F1= {4,5,8}
- F2={3,6,7}
- F3={1}
- F4={2}

MinFrag algorithm

Add player = Beckham (ok)

Add goals <=1 (ok)

Add goals <=4 (no)

Add fouls > 10 (ok), now one can realize that either of goals <=1 or fouls > 10 becomes redundant

Final result { player= Beckham ,fouls >10}

OR {player=Beckham, goals<=1}

The minpredicates and minfragmetns are:

player=Beckham, goals<=1 ---> **{1}**

player=Beckham, goals>1 ---> **{2}**

player!=Beckham, goals<=1 ---> **{3,6,7}**

player!=Beckham, goals>1 ---> **{4,5,8}**

Alternatively, the following solutions are also correct:

{ player= Beckham, fouls <= 10}

OR {player=Beckham, goals>1}

Solution for 2(a)

The Matrix Q (which queries access which attributes)

	Car_type	Owner_name	Year	City
Q1	0	0	1	0
Q2	1	0	1	0
Q3	0	0	0	1
Q4	0	1	0	1

The Matrix M (how many times a query is posted from a site)

	P1	P2	P3
Q1	1	0	0
Q2	15	5	10
Q3	0	1	0
Q4	0	10	10

Then the Affinity Matrix A would be

	Car_type	Owner_name	Year	City
Car_type	30	0	30	0
Owner_name	0	20	0	20
Year	30	0	31	0
City	0	20	0	21

Solution for 2(b)

We have to run the BEA algorithm to first, determine the best attribute order.

<u>Step 1</u>

Let's choose the **Car_type** column first into the output matrix. It should be noted that the column **Year** can also be placed into this matrix. So our initial matrix would be:

	Car_type	Year		
Car_type	30	30		
Owner_name	0	0		
Year	30	31		
City	0	0		

<u>Step 2</u>

Now let's pick up the **Owner_name** attribute and find out the best position for it.

Option 1:

	Owner_name	Car_type	Year	
Car_type	0	30	30	
Owner_name	20	0	0	
Year	0	30	31	
City	20	0	0	

cont(_,Owner_name,Car_type)

= bond(_,Owner_name) + bond(Car_type, Owner_name) – bond(_,Car_type)

= 0 + 0 - 0 = 0

Option 2:

	Car_type	Owner_name	Year	
Car_type	30	0	30	
Owner_name	0	20	0	
Year	30	0	31	
City	0	20	0	

cont(Car_type, Owner_name, Year)

= bond(Car_type, Owner_name) + bond(Owner_name, Year) – bond(Car_type,Year)

= 0 + 0 – 1830

= -1830

Option 3:

	Car_type	Year	Owner_name	
Car_type	30	30	0	
Owner_name	0	0	20	
Year	30	31	0	
City	0	0	20	

cont(Year, Owner_name, _)

= bond(Year, Owner_name) + bond(Owner_name,_) - bond(Year,_)

= 0 + 0 - 0 = 0

Hence, the position of Owner_name should be either at the beginning or the end of the initial selection from Step 1. Lets choose the last position.

	Car_type	Year	Owner_name	
Car_type	30	30	0	
Owner_name	0	0	20	
Year	30	31	0	
City	0	0	20	

Step 3:

Now lets decide the position for the attribute **City.**

Modeling on the same lines, there are 4 possible positions for this attribute.

Here are their contributions:

cont(_, City, Car_type) = 0 + 0 – 0 = **0**

cont(Car_type, City, Year) = 0 + 0 – 1830 = **-1830**

cont(Year, City, Owner_name) = 0 + 820 -0 = **820**

cont(Owner_name, City, _) = 820 + 0 – 0 = **820**

Hence, there are two best positions for the City attribute. Let's choose the last one.

The resulting attribute ordering (both in the columns and the rows) will be:

	Car_type	Year	Owner_name	City
Car_type	30	30	0	0
Year	30	31	0	0
Owner_name	0	0	20	20
City	0	0	20	21

See the resulting clusters, which results in two fragments :

F1={Number, Car_type, Year}

F2= {Number, Owner_name, City}

Note the addition of the primary key (Number) to both the fragments.