Midterm Solution

Question 1:

You are hired by a big bank. It has a head quarter and a number of branches over Hong Kong. Each of those branches has its own databases managed by a DBMS. Some of them use Oracle and some of them use SQL Server. You are asked to design a distributed database system for the bank so that a branch can access data at other branches and officers a the header quarter can access data at all branches.

a) (5%) You propose that the system will have no global schema. List two reasons that you can use to justify your decision.

Solution:
Reasons for the proposal:
1. The DBMS is heterogeneous. Therefore, providing global schema lead to high cost and complexity.
2. Whenever a new branch is set up, it creates its own schema and then shares it with all other branches. It is much more costly to define the global schema again and figure out the new local schema for the new branch in a top-down approach. Without the global schema, we can do it in a bottom-up approach.

b) (5%) You are expecting your colleagues to challenge you. List two reasons that your colleagues may use to against your proposal.

Solution:
Reasons against the proposal:
1. Update of a data may not lead to an update of the same data in other branches since there is no global schema. It needs to synchronization / consistency / integrity problem.
2. With help of the integrated view provided by the global schema, database users do not need to know where the data is. This provides higher level of transparency and hence eases the implementation of front end system.

Question 2: (30%)

A supply-part database consists of three relations as shown below.

SUPPLIER		
SNO	SNAME	COUNTRY
S1	SN1	USA
S2	SN2	INDIA
S3	SN3	CHINA
S4	SN4	CHINA
S5	SN5	INDIA
S6	SN6	USA

SUPPLY		
SNO	PNO	QTY
S1	P1	60
S1	P3	70
S2	P2	60
S3	P3	55
S3	P4	96
S4	P2	65
S6	P2	70
S6	P4	96

PARTS		
PNO	PNAME	PRICE
P1	PC	10000
P2	CAMERA	8000
P3	VIDEO	5000
P4	HI-HI	3000

There are three applications:
Q1: Print SNO of suppliers who supply parts with price less than 6000.
Q2: For each supplier in USA, print SNAME, and PNO of the parts that s/he supplies.
Q3: For each supplier, print SNO, SNAME, the number of parts s/he supplies.

a) (10%) Determine a set of simple predicates that is complete minimal. Justify your answer.

Solution:
Change to application statement to SQL:

```
Q1:   select SNO
      from PARTS, SUPPLY
      where PARTS.PNO = SUPPLY.PNO
      and PARTS.PRICE < 6000
Q2:   select SNAME, PNO
      from SUPPLIER, SUPPLY
      where SUPPLIER.SNO = SUPPLY.SNO
      and SUPPLIER.COUNTRY = "USA"
Q3:   select SNO, SNAME, COUNT(*)
      from SUPPLIER, SUPPLY
      where SUPPLIER.SNO = SUPPLY.SNO
      group by SUPPLIER.SNO
```

Simple predicate from application:
Q1: price < 6000
Q2: country = "USA"
Q3: null

Since "price < 6000" is related to relation PARTS and "country = "USA"" is related to relation SUPPLIER, we need two sets of complete and minimal simple predicates.

Set of complete and minimal simple predicate for relation PARTS:
{"price < 6000"}
Set of complete and minimal simple predicate for relation SUPPLIER:
{"country = "USA""}

Checking for completeness:

The minterm predicates for PARTS is {"price < 6000", "price >= 6000"}
The minterm predicates for SUPPLIER is {"country = "USA"", country ≠ "USA"}
PARTS 1 = $\sigma_{\text{price} < 6000}$ PARTS
PARTS 2 = $\sigma_{\text{price} >= 6000}$ PARTS
SUPPLIER 1 = $\sigma_{\text{country} = \text{"USA"}}$ SUPPLIER
SUPPLIER 2 = $\sigma_{\text{country} \neq \text{"USA"}}$ SUPPLIER

Fragments		Q1	Q2	Q3
SUPPLIER 1	S1		X	X
	S6		X	X
SUPPLIER 2	S2			X
	S3			X
	S4			X
	S5			X
PARTS 1	P3	X		
	P4	X		
PARTS 2	P1			
	P2			

Since all the turples in all fragments have the same probability to be accessed by all applications, the two set of simple predicates are complete.

Checking for minimality:

For relation PARTS: The addition of predicate "price < 6000" divide PARTS into two fragments which accessed differently by Q1.
Therefore, "price < 6000" is relevant and so {"price < 6000"} is minimal for PARTS.

For relation SUPPLIER: The addition of predicate "country < "USA"" divide SUPPLIER into two fragments which accessed differently by Q2.
Therefore, "country < "USA"" is relevant and so {"country < "USA""} is minimal for SUPPLIER.

b) Derive a horizontal fragmentation step by step, and show the contents of each fragmentation.

Solution:

For relation PARTS:
1. complete and minimal simple predicate: p1 = "price < 6000"
2. minterm predicate: p1, ¬p1.
3. Implications among simple predicate: none
4. Removal of predicate due to contradiction: none
5. Do the horizontal fragmentation:
   PARTS 1 = $\sigma_{\text{price} < 6000}$ PARTS
   PARTS 2 = $\sigma_{\text{price} >= 6000}$ PARTS

PARTS 1		
PNO	PNAME	PRICE
P3	VIDEO	5000
P4	HI-HI	3000

PARTS 2		
PNO	PNAME	PRICE
P1	PC	10000
P2	CAMERA	8000

For relation SUPPLIER:

6. complete and minimal simple predicate: p1 = "country = "USA""
7. minterm predicate: p1, ¬p1.
8. Implications among simple predicate: none
9. Removal of predicate due to contradiction: none
10. Do the horizontal fragmentation:

    SUPPLIER 1 = $\sigma_{\text{country = "USA"}}$ SUPPLIER

    SUPPLIER 2 = $\sigma_{\text{country} \neq \text{"USA"}}$ SUPPLIER

SUPPLIER 1		
SNO	SNAME	COUNTRY
S1	SN1	USA
S6	SN6	USA

SUPPLIER 2		
SNO	SNAME	COUNTRY
S2	SN2	INDIA
S3	SN3	CHINA
S4	SN4	CHINA
S5	SN5	INDIA

For relation SUPPLY, do derived fragmentation:

Temp1 = SUPPLY $\ltimes$ SUPPLIER1

Temp2 = SUPPLY $\triangleright\!\!\ltimes$ SUPPLIER2

SUPPLY1 = Temp1 $\ltimes$ PARTS1

SUPPLY2 = Temp1 $\ltimes$ PARTS2

SUPPLY3 = Temp2 $\triangleright\!\!\ltimes$ PARTS1

SUPPLY4 = Temp2 $\triangleright\!\!\ltimes$ PARTS2

SUPPLIER1		
SNO	PNO	QTY
S6	P4	96

SUPPLILER2		
SNO	PNO	QTY
S1	P1	60
S6	P2	70

SUPPLIER3		
SNO	PNO	QTY
S1	P3	70
S3	P3	55
S3	P4	96

SUPPLIER4		
SNO	PNO	QTY
S2	P2	60
S4	P2	65

Question 3 (30%)

In the Bond Energy Algorithm for vertical partitioning, we are given a use-matrix U[1..n, 1..k], and an access frequency matrix ACC[1..n, 1..m], where n is the number of queries, k is the number of attributes, and m is the number of sites in a distributed system.

a) (10%) Compute the AA for the following input.

U =

	A1	A2	A3	A4
q1	0	0	1	0
q2	1	1	1	0
q3	1	0	0	1
q4	0	1	1	1
q5	1	1	1	1

ACC =

	S1	S2	S3
q1	10	20	0
q2	5	0	10
q3	0	35	5
q4	0	10	0
q5	0	15	0

Solution:

AA =

	A1	A2	A3	A4
A1	70	30	30	55
A2	30	40	40	25
A3	30	40	70	25
A4	55	25	25	65

b) (20%) Assume two columns A_i and A_j have been placed next to each other in the CA matrix. Write the procedure that determine the location where the third attribute A_k should be placed based on contributions to the global affinity measure as defined in the Bond Energy Algorithm.

Solution:

Assume without loss of generality that the given order is (A_i, A_j)

Contribution[1] := Cont(NULL, A_k, A_i);
Contribution[2] := Cont(A_i, A_k, A_j);
Contribution[3] := Cont(A_j, A_k, NULL);
If Contribution [1] > Contribution [2] and Contribution [1] > Contribution [3] then
    Place A_k at the left hand side of A_i
  else if Contribution [2] > Contribution [1] and Contribution [2] > Contribution [3]
    then Place A_k between A_i and A_j
  else Place A_k at the right hand side of A_j

```
Function Cont(a, b, c)
  Begin
      If (a==0) or (c==0)
          then return 0;
      return (2*Bond(a, b) + 2*Bond(b, c) – 2*Bond(a, c));
  End
```

Function Bond(A_m, A_n)

```
Begin
    Sum = 0;
    // Assume the no of attributes is N
    For r := 1 to N do
        Sum += aff[A_r, A_m]*aff[A_r, A_n];
    Return sum;
End
```

Question 4 (30%)

a) (10%) You are given a fragment F, a set of sites S_1, S_2, … , S_m, and a set of queries, Q_1, Q_2, …, Q_n. Workload of the system is represented by two arrays: Size[1..n], and Freq[1..n, 1..m]. Size[i] is the size of data to be transferred for query i when the fragment is not allocated at the site where the query is issued. Freq[i,j] is the frequency of query i issued at site j. Write the procedure (in pseudo-code form) that determines the site where F should be allocated so that the total data transferring cost is minimized.

Solution:

```
Begin
    MinCost := MAX;
    For  site  := 1 to m do
        Cost := GetCost(site);
        If cost < Min_cost then
            MinCost := Cost;
            MinSite := site;
        End
    End;
    Return minSite;
end

Function GetCost (site)    /* compute the cost if F is allocated at the site)
begin
    Cost := 0;
    For  thisSite := 1 to m do
        For  query := 1 to n do
            If  thisSite != site then  Cost += Freq[query, thisSite]*Size[query];
    Return cost;
end.
```

b) (20%) Follow the above question. If you are also given an array Update[1..n], where Update[i] represents the update cost of query i (Update[i] = 0 indicates that query i does not involve any updates). Write the procedure (in pseudo-code form) that determines whether F should be replicated an another site.

Solution:

```
Begin
    Let LocatedSite is the site determined in a)
    For site := 1 to m do
        If  site != LocatedSite and Benificial(site) then return True
    Return False;
End.

Function Benificial (Site)
Begin
    UpdateCost = 0;
    Saving = 0;
    For query := 1 to k do
        For site := 1 to m do
            UpdateCost += Freq[query, Site]*Update[query];
        Saving += Freq[query, Site]*size[query];
    End;
    Return (Saving > UpdateCost);
End;
```

In this algorithm, when the two cost is equal, we do not replicate the data since we should not make some effort for no reward.