Problem 2.1 (*) Given relation EMP as in Fig. 2.2, let p1: TITLE < "Programmer" and p2: TITLE > "Programmer" be two simple predicates. Assume that character strings have an order among them, based on the alphabetical order.

(a) Perform a horizontal fragmentation of relation EMP with respect to {p1, p2}.

(b) Explain why the resulting fragmentation (EMP1, EMP2) does not fulfill the correctness rules of fragmentation.

(c) Modify the predicates p1 and p2 so that they partition EMP obeying the correctness rules of fragmentation. To do this, modify the predicates, compose all minterm predicates and deduce the corresponding implications, and then perform a horizontal fragmentation of EMP based on these minterm predicates. Finally, show that the result has completeness, reconstruction, and disjointness properties

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

ASG

ENO	PNO	RESP	DUR
E1	P1	Manager	12
E2	P1	Analyst	24
E2	P2	Analyst	6
E3	P3	Consultant	10
E3	P4	Engineer	48
E4	P2	Programmer	18
E5	P2	Manager	24
E6	P4	Manager	48
E7	P3	Engineer	36
E8	P3	Manager	40

PROJ

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal
P2	Database Develop.	135000	New York
P3	CAD/CAM	250000	New York
P4	Maintenance	310000	Paris

PAY

TITLE	SAL
Elect. Eng.	40000
Syst. Anal.	34000
Mech. Eng.	27000
Programmer	24000

Fig.2.2

Solution:

a)

EMP_1: TITLE < "Programmer"

ENO	ENAME	TITLE
E_1	J. Doe	Elect. Eng.
E_3	A. Lee	Mech. Eng.
E_6	L. Chu	Elect. Eng.
E_7	R. Davis	Mech. Eng.

EMP_2: TITLE > "Programmer"

ENO	ENAME	TITLE
E_2	M Smith	Syst. Anal.
E_5	B. Casey	Syst. Anal.
E_8	J. Jones	Syst. Anal.

b) The resulting fragmentation is incomplete since E4, who is a Programmer cannot be found in either fragment.

c) This can be accomplished by changing p1 as TITLE ≤ "Programmer" or changing p2 as TITLE ≥ "Programmer". If p1 is changed, E4 will be added to EMP1; if p2 is modified, E4 will be added to EMP2.

****

Problem 2.2 (*) Consider relation ASG in Fig. 2.2. Suppose there are two applications that access ASG. The first is issued at five sites and attempts to find the duration of assignment of employees given their numbers. Assume that managers, consultants, engineers, and programmers are located at four different sites. The second application is issued at two sites where the employees with an assignment duration of less than 20 months are managed at one site, whereas those with longer duration are managed at a second site. **Derive the primary horizontal fragmentation** of ASG using the foregoing information.

Solution:

- For the first application, we have the following simple predicates:
  - p1: RESP = "Manager"
  - p2: RESP = "Consultant"
  - p3: RESP = "Engineer"
  - p4: RESP = "Programmer"
- For the second application we have:
  - p5: DUR < 20
  - p6: DUR > 20
- Accordingly, we can form the following minterms:
  - m1: RESP = "Manager" ∧ DUR < 20
  - m2: RESP = "Manager" ∧ DUR > 20
  - m3: RESP = "Consultant" ∧ DUR < 20
  - m4: RESP = "Consultant" ∧ DUR > 20 → No result
  - m5: RESP = "Engineer" ∧ DUR < 20 → No result
  - m6: RESP = "Engineer" ∧ DUR > 20
  - m7: RESP = "Programmer" ∧ DUR < 20
  - m8: RESP = "Programmer" ∧ DUR > 20 → No result

- Note that m4, m5, and m8 are empty, so in the end we get the following fragments:

ASG_1

ENO	PNO	RESP	DUR
E_1	P_1	Manager	12

ASG_2

ENO	PNO	RESP	DUR
E_5	P_2	Manager	24
E_6	P_3	Manager	48
E_8	P_4	Manager	40

ASG_3

ENO	PNO	RESP	DUR
E_3	P_3	Consultant	10

ASG_6

ENO	PNO	RESP	DUR
E_3	P_4	Engineer	48
E_7	P_3	Engineer	36

ASG_7

ENO	PNO	RESP	DUR
E_4	P_2	Programmer	18

Problem 2.3(*) Consider relations EMP and PAY in Fig. 2.2. EMP and PAY are horizontally fragmented as follows:

EMP1 = σ_{TITLE}=“Elect. Eng.”(EMP)

EMP2 = σ_{TITLE}=“Syst. Anal.”(EMP)

EMP3 = σ_{TITLE}=“Mech. Eng.”(EMP)

EMP4 = σ_{TITLE}=“Programmer”(EMP)

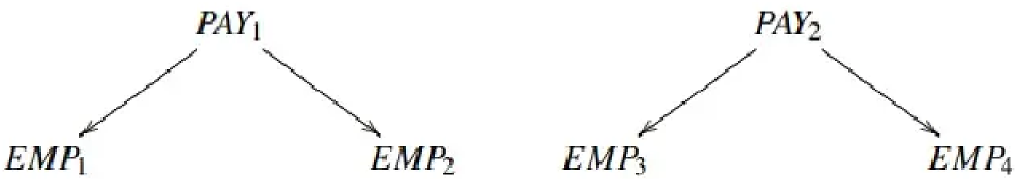
PAY1 = $\sigma_{SAL}\geq$30000(PAY)

PAY2 = $\sigma_{SAL}<$30000(PAY)

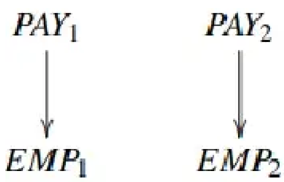
Draw the join graph of EMP$_{TITLE}$ PAY. **Is the graph simple or partitioned**? If it is partitioned, **modify the fragmentation of either EMP or PAY so that the join graph** of EMP $_{TITLE}$ PAY is **simple**.

Solution:

- The join graph is the following:



- This is not simple, it is partitioned.
- To make it simple, Let us change EMP fragmentation as follows:
  - EMP1: σ_{TITLE}=“Elect. Eng.”∧ TITLE=“Syst. Anal.” (EMP)
  - EMP2: σ_{TITLE}=“Mech. Eng.”∧ TITLE=“Programmer” (EMP)
  - This results in:



Problem 2.5 (**) Given relation PAY as in Fig. 2.2, let p1 : SAL < 30000 and p2 : SAL ≥ 30000 be two simple predicates. **Perform a horizontal fragmentation of PAY with respect to these predicates** to obtain PAY1 and PAY2. **Using the fragmentation of PAY, perform further derived horizontal fragmentation for EMP**. **Show completeness, reconstruction, and disjointness of the fragmentation of EMP.**

Solution:

- PAY1:

TITLE	SAL
Mech. Eng.	27000
Programmer	24000

And PAY2:

TITLE	SAL
Elect. Eng.	40000
Syst. Anal.	34000

- **Derived horizontal fragmentation for EMP:**
  - EMP1 = EMP ⊳< $_{TITLE}$PAY1

ENO	ENAME	TITLE
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E7	R. Davis	Mech. Eng.

  - EMP2 = EMP ⊳< $_{TITLE}$PAY2

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng
E2	M. Smith	Syst. Anal.
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E8	J. Jones	Syst. Anal.

- **Completeness**: all tuples in the original relation EMP can be found in the resulting relations EMP1, EMP2.
- **Reconstruction**: The original relation EMP can be reconstructed by the union operator on the resulting fragments: EMP1 and EMP2.
- **Disjointness**: EMP1 and EMP2 do not have any common tuples.

Problem 2.7 (**) Write an algorithm for derived horizontal fragmentation

Solution:

```
Algorithm HORIZ_FRAG
Input: SIDU = attribute-predicate matrix;
Output:  ACF = array of cost frequency
Begin
  for i from 1 to TotalNumberOfAtrributes do
     for j from 1 to TotalNumberOfPredicates[i] do
        begin
                max[i][j]=0;
                for k from 1 to TotalNumberOfSites do
                begin
      /* calculating sum of all applications  cost of predicate j of attribute i at site k */
                        for l from 1 to TotalNumberOfApplications[k] do
                        begin
                                cost[i][j][k][l] = S + I + D + U
                                sum[i][j][k] = sum[i][j][k] + cost[i][j][k][l]

                          end_for
                          /* find out at which site cost of predicate j is maximum */
                          if sum[i][j][k] > max[i][j] then
                          begin
                                  max[i][j]=sum[i][j][k]
                                  position[i][j]=k
                          end_if
         /* calculating the other sum of cost applications diferent of current site k */
                          restSum=0
                          for l from 1 to TotalNumberOfApplications[k] do
                                  if l ≠ k then
                                  begin
                                          restCost[i][j][k][l] = S + I + D + U
                                          restSum = restSum + restCost[i][j][k][l]
                                  end_if
                end_for
                /* actual cost for predicate j of attribute i */
                ACF[i][j] = sum[i][j][position[i][j]] - restSum
        end_for
  End
```

Problem 2.15 (**) Describe how the following can be properly modeled in the database allocation problem.

(a) Relationships among fragments

(b) Query processing

(c) Integrity enforcement

(d) Concurrency control mechanisms

Solution:

a)The relationships among fragments can be properly modeled in the database allocation problem by using a graph data structure. The fragments can be represented as nodes in the graph, and the relationships among the fragments can be represented as edges in the graph.

b)Query processing can be modeled in the database allocation problem by using a query processing algorithm. The query processing algorithm can be used to find the fragments that are relevant to the query, and to generate the results of the query.

c)Integrity enforcement can be modeled in the database allocation problem by using an integrity enforcement algorithm. The integrity enforcement algorithm can be used to enforce the integrity constraints of the database.

d)Concurrency control can be modeled in the database allocation problem by using a concurrency control algorithm. The concurrency control algorithm can be used to control the access of the fragments to the database.

****

Problem 2.16 (**) Consider the various heuristic algorithms for the database allocation problem.

(a) What are some of the reasonable criteria for comparing these heuristics? Discuss.

(b) Compare the heuristic algorithms with respect to these criteria.

Solution:

(a) Here's some reasonable criteria for comparing these heuristics.

Two heuristic evaluation approaches were examined in this comparison research. **Usability** testing was conducted to confirm whether the administration tool was user-friendly. These concepts will be applied in detail the methods utilized. The research questions will be restated first:

RQ1: "When can Usability Evaluation Methods be used in a cost-effective and formative manner? Implementing a website while maintaining high levels of pleasure and efficiency?"

RQ2: "Are Gerhardt-Powals principles more effective than Nielsen's heuristics at producing more high-severity usability problems in less time?"

A definition of usefulness is required by RQ1. What criteria should be used to determine whether a webpage has a high level of user satisfaction?

(b) A step-by-step technique for addressing a specific problem in a finite number of steps is known as an algorithm. Given the same parameters, an algorithm's outcome is predictable and repeatable. A heuristic is an educated guess that serves as a starting point for further investigation. Heuristic evaluation involves someone looking at the user interface and identifying issues. Potential consumers check out the user interface with real tasks during usability testing. The issues discovered during usability testing are genuine issues, as each one was experienced by at least one user. The fundamental difference is that a representative heuristic makes judgments on items and people based on stereotypes. The availability heuristic, on the other hand, uses recent events to help predict future events.

****
****

Problem 3.1. Define in SQL-like syntax a view of the engineering database V(ENO,ENAME, PNO, RESP), where the duration is 24. Is view V updatable? Assume that relations EMP and ASG are horizontally fragmented based on access frequencies as follows:

Site 1 Site 2 Site 3

EMP1 EMP2

ASG1 ASG2

where

$EMP1 = \sigma TITLE6=$"Engineer"(EMP)

$EMP2 = \sigma TITLE =$ "Engineer" (EMP)

$ASG1 = \sigma 0<DUR<36(ASG)$

$ASG2 = \sigma DUR \geq 36(ASG)$

At which site(s) should the definition of V be stored without being fully replicated to increase locality of reference?

Solution:

In new folder

****

Problem 3.2 Express the following query: names of employees in view V who work on the CAD/CAM project.

Solution:

In new folder

****

Problem 3.3 (*) Assume that relation PROJ is horizontally fragmented as

$PROJ1 = \sigma_{PNAME} =$ "CAD/CAM" (PROJ)

$PROJ2 = \sigma_{PNAME}=$ "CAD/CAM" (PROJ)

Modify the query obtained in Problem 3.2 to a query expressed on the fragments.

Solution:

In new folder

****

Problem 3.4 (**) Propose a distributed algorithm to efficiently refresh a snapshot at one site derived by projection from a relation horizontally fragmented at two other sites. Give an example query on the view and base relations which produces an inconsistent result.

Solution:

Step-by-Step explanation

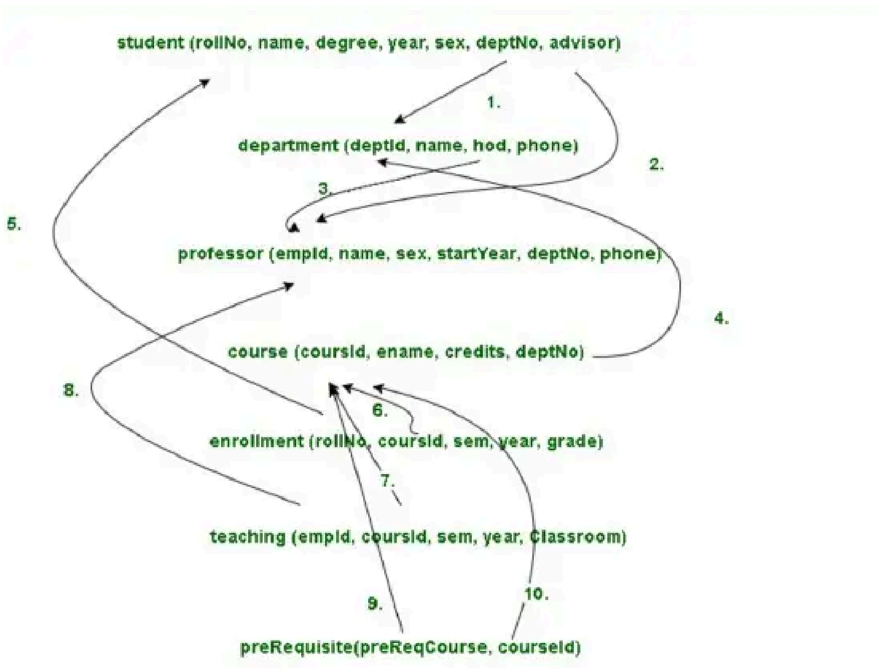
Solution:

a. i. Send the query name(employee) to the Boca plant.
ii. Have the Boca location send back the answer.
b. i. Compute the average at New York.
ii. Send answer to San Jose.
c.
i. Send the query to find the highest salaried employee to Toronto, Edmonton, Vancouver, and Montreal.
ii. Compute the queries at those sites.
iii. Return answers to San Jose.
d. i. Send the query to find the lowest salaried employee to New York.
ii. Compute the query at New York.
iii. Send answer to San Jos

****

Problem 3.6 Propose a relation schema for storing the access rights associated with user groups in a distributed database catalog, and give a fragmentation scheme for that relation, assuming that all members of a group are at the same site.

Solution:



FRAGMENTATION :
In this model tables are divided by two parts or pieces, referred to as partitions or fragments that could be stored at various locations. This is because it is rare that all data in a table will be required at the same location. Additionally, fragmentation enhances parallelism and helps with disaster recovery. In this case, there is just one replica of each fragment in this system i.e. no redundant data.
The three techniques for fragmentation are:
Vertical fragmentation
Horizontal fragmentation
Hybrid fragmentation
Vertical Fragmentation
In vertical fragmentation the columns or fields in a table get divided into fragments. In order to ensure the integrity of the reconstruction each fragment must contain one primary field(s) within the table. Vertical fragmentation can be utilized to protect data from unauthorized access.
Let us say that the University database records the students who have registered in the Student table with this schema.
STUDENT

Regd_No	Name	Course	Address	Semester	Fee	Marks

Step 4
The fees information are kept in the section of accounts. In this instance the designer would break the database in the following manner:
CREATE TABLE STD_FEES AS
SELECT Regd_No, Fees
FROM STUDENT;
Horizontal Fragmentation
Horizontal fragmentation is the process of dividing the tuples of tables according to the numbers of fields or values. Horizontal fragmentation also needs to adhere to the principle of reconstructiveness. Each horizontal fragment should have all columns from the original base table.
For instance in the student schema in the student schema, if the data of all students in the Computer Science Course needs to be kept at the School of Computer Science, then the designer will vertically split the database according to the following pattern:
CREATE COMP_STD AS
SELECT * FROM STUDENT
WHERE COURSE = "Computer Science";
Hybrid Fragmentation
When a hybrid method is employed, it's a mix of vertical and horizontal fragmentation techniques is used. It is the most adaptable method of fragmentation as it creates fragments with only the minimum of extra information. However, re-creating the original table can be costly.
Hybrid fragmentation is possible in two ways:
First, create the horizontal fragments. Later, create vertical fragments using one or more horizontal fragments.
In the beginning, you will need to generate the vertical fragments. Later, you can create horizontal fragments using one or more vertical fragments.
****

Problem 3.7 (**) Give an algorithm for executing the REVOKE statement in a distributed DBMS, assuming that the GRANT privilege can be granted only to a group of users where all its members are at the same site.

Solution:

The main difficulty with this approach is that the revoking process must be recursive. For example, if A, who granted B who granted C the GRANT privilege on object O, wants to revoke all the privileges of B on O, all the privileges of C on O must also be revoke.

Bell and Lapaduda model

1. A subject S is allowed to read an object of security level I only if level(S) ≥ I called the (No READ UP).
2. A subject S is allowed to write an object of security level I only if class(S) ≤ I , (no Write Down)

Steps:

1. Authentication information is maintained at a central site for global users which can then be authenticated only once and then accessed from multiple sites.
2. The information for authenticating users (user name and password) is repli-cated at all sites in the catalog. Local programs, initiated at a remote site, must also indicate the user name and password
3. All sites of the distributed DBMS identify and authenticate themselves similar to the way users do. Intersite communication is thus protected by the use of the site password. Once the initiating site has been authenticated, there is no need for authenticating their remote users.