Sardar Patel Institute of Technology
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058, India
(Autonomous College Affiliated to University of Mumbai)
End Semester Examination
November 2018

Max. Marks: 60 Duration: 3 Hrs
Class: T.E. Semester: V
Course Code: IT53 Branch: Information Technology
Name of the Course: Advanced Database Systems

Synoptic

Q1 a. In Parallel databases how shared memory and shared disk architectures are used? Give the advantages and disadvantages of both.

Ans:

1. Shared Memory Architecture 1 mark

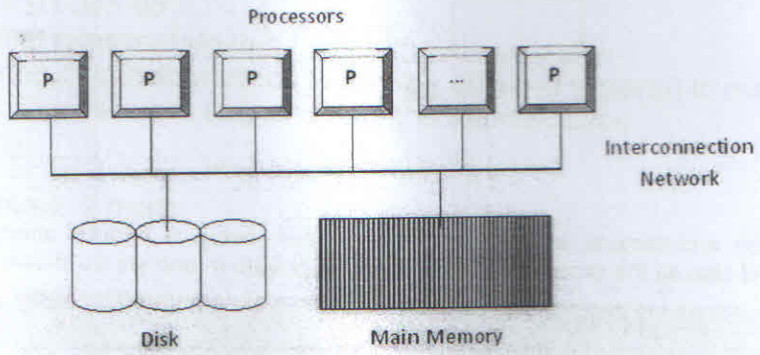


Figure 1 - Shared Memory Architecture

In Shared Memory architecture, single memory is shared among many processors as show in Figure 1. As shown in the figure, several processors are connected through an interconnection network with Main memory and disk setup. Here interconnection network is usually a high speed network (may be Bus, Mesh, or Hypercube) which makes data sharing (transporting) easy among the various components (Processor, Memory, and Disk).

Advantages: ***½ mark***

- Simple implementation
- Establishes effective communication between processors through single memory addresses space.
- Above point leads to less communication overhead.

Disadvantages: ***½ mark***

- Higher degree of parallelism (more number of concurrent operations in different processors) cannot be achieved due to the reason that all the processors share the same interconnection network to connect with memory. This causes Bottleneck in interconnection network (Interference), especially in the case of Bus interconnection network.

- Addition of processor would slow down the existing processors.

- Cache-coherency should be maintained. That is, if any processor tries to read the data used or modified by other processors, then we need to ensure that the data is of latest version.

- Degree of Parallelism is limited. More number of parallel processes might degrade the performance.

2. Shared Disk Architecture 1 mark

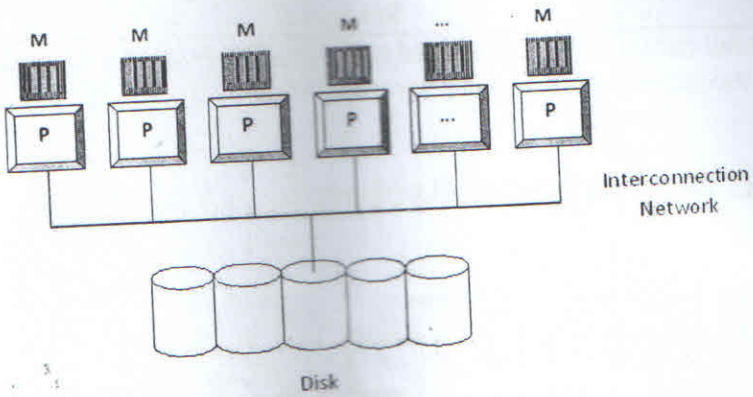


Figure 2 - Shared Disk Architecture

In Shared Disk architecture, single disk or single disk setup is shared among all the available processors and also all the processors have their own private memories as shown in Figure 2.

Advantages: ½ mark

- Failure of any processors would not stop the entire system (Fault tolerance)
- Interconnection to the memory is not a bottleneck. (It was bottleneck in Shared Memory architecture)
- Support larger number of processors (when compared to Shared Memory architecture)

Disadvantages: ½ mark

- Interconnection to the disk is bottleneck as all processors share common disk setup.
- Inter-processor communication is slow. The reason is, all the processors have their own memory. Hence, the communication between processors need reading of data from other processors' memory which needs additional software support.

Q1 b. Compare NOSQL and RDBMS.

Any 4 points 4 marks

- Higher degree of parallelism (more number of concurrent operations in different processors) cannot be achieved due to the reason that all the processors share the same interconnection network to connect with memory. This causes Bottleneck in interconnection network (Interference), especially in the case of Bus interconnection network.

- Addition of processor would slow down the existing processors.

- Cache-coherency should be maintained. That is, if any processor tries to read the data used or modified by other processors, then we need to ensure that the data is of latest version.

- Degree of Parallelism is limited. More number of parallel processes might degrade the performance.

2. Shared Disk Architecture 1 mark

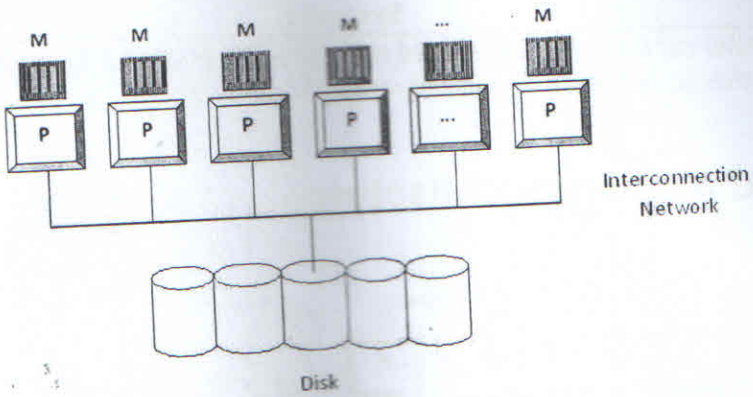


Figure 2 - Shared Disk Architecture

In Shared Disk architecture, single disk or single disk setup is shared among all the available processors and also all the processors have their own private memories as shown in Figure 2.

Advantages: ½ mark

- Failure of any processors would not stop the entire system (Fault tolerance)
- Interconnection to the memory is not a bottleneck. (It was bottleneck in Shared Memory architecture)
- Support larger number of processors (when compared to Shared Memory architecture)

Disadvantages: ½ mark

- Interconnection to the disk is bottleneck as all processors share common disk setup.
- Inter-processor communication is slow. The reason is, all the processors have their own memory. Hence, the communication between processors need reading of data from other processors' memory which needs additional software support.

Q1 b. Compare NOSQL and RDBMS.

Any 4 points 4 marks

Q1c. What is temporal database? With appropriate example describe a types of temporal databases.

Ans :Temporal DB ½ marks

Valid time with example 1 marks

Transaction time with example 1.5 marks

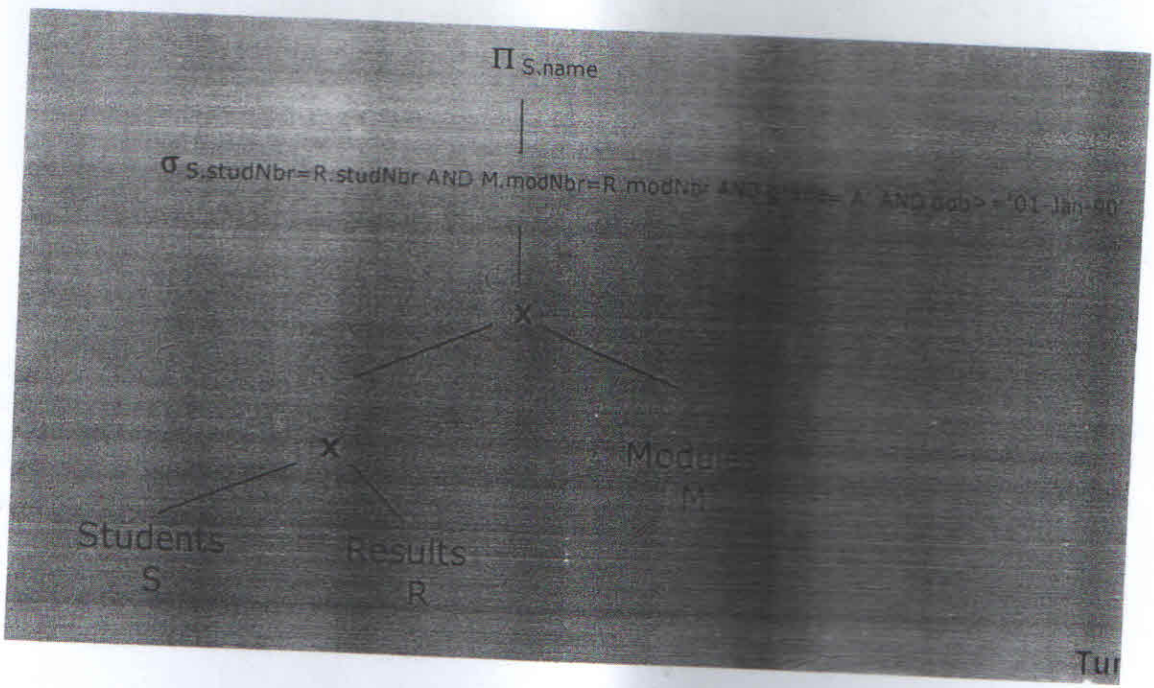
Bi_temporal time with example 1 marks

Q2a. Consider the following three linked relations:
Students (studNbr, name, dob)
Modules (modNbr, title)
Results (studNbr*, modNbr*, grade)
Suppose we have the following query:

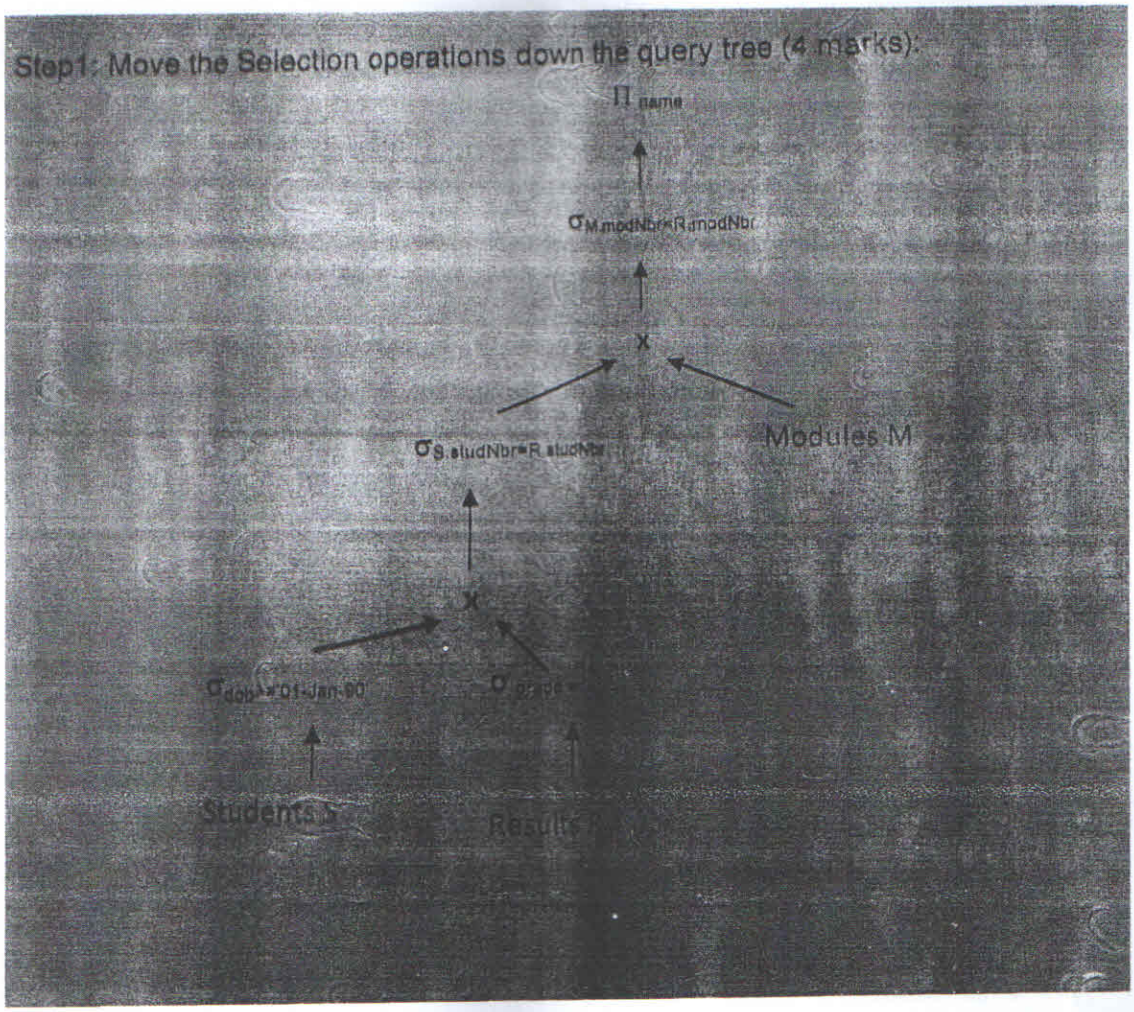
```
SELECT S.name FROM Students S, Modules M, Results R
WHERE S.studNbr = R.studNbr
AND M.modNbr = R.modNbr
AND R.grade = 'A'
AND S.dob >= '01-Jan-90';
```

(i) Draw an initial relational algebra tree for the above query.
(ii) Apply a series of transformations to the tree obtained in part(i) to make the query more efficient. Discuss each step and state the heuristic used.

Ans: Initial subtree 2 marks

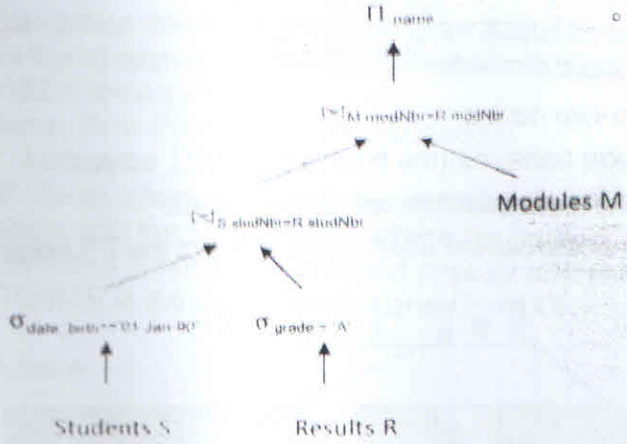


Step 1 2 marks

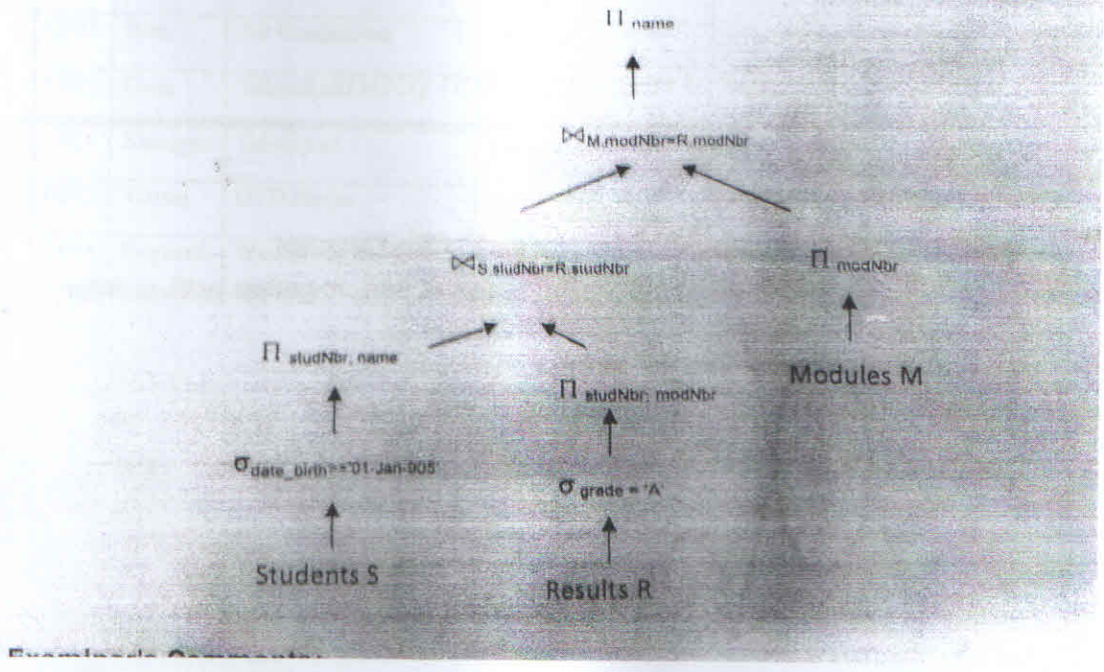


Step 2 and 3 2 marks

Step 2: combine the Cartesian Product with a subsequent Selection operation into a Join operation (4 marks)



Step 3 Move the Projection operations down the query tree (5 marks):

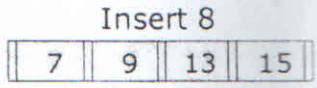


In B+ tree for insertion write the overflow conditions for splitting leaf node and non-leaf node , if the node contains 7,9,13,15 and this node is full. Insert 8 in this node.

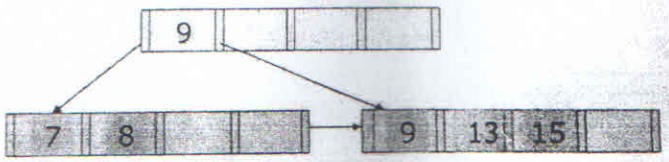
Ans: 3 marks

B+ Tree Insertion

- Overflow
  - When number of search-key values exceed n



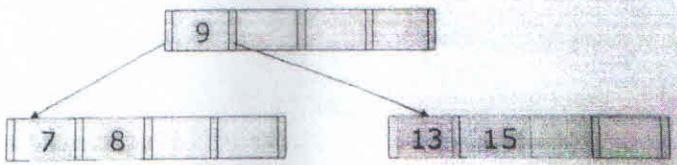
- Leaf Node
  - Split into two nodes:
    - 1st node contains (the first) $\lfloor (n+1)/2 \rfloor$ values
    - 2nd node contains the remaining values
    - Copy the smallest search-key value of the 2nd node to parent node



3 marks

B+ Tree Insertion (cont.)

- Non-Leaf Node
  - Split into two nodes:
    - 1st node contains (the first) $\lceil (n+1)/2 \rceil - 1$ values
    - Move the smallest of the remaining values, together with pointer, to the parent
    - 2nd node contains the remaining values



Q2b. While designing the distributed database what types of replications are required. Justify with example and give the advantages and disadvantages of replication.

Ans: 3 Types of replications with example and advantages and disadvantages 2 marks each

Q3 a. Consider an organization maintains the information about its customers. They store information about the customer in CUSTOMER table and the customer addresses in C_ADDRESS table as follows;
CUSTOMER(CId, CName, Prod Purchased, Shop Location)
C_ ADDRESS(CId, C_Address)
The table CUSTOMER stores information about the customer, the product purchased from their shop, and the shop location where the product is purchased. C_ Address stores information about permanent and present addresses of the customer. Here, CUSTOMER is the owner relation and C_ ADDRESS is the member relation.
Figure 1: CUSTOMER table

CID	CNAME	PROD_PURCHASED	SHOP_LOCATION
C001	Ram	Air Conditioner	Mumbai
C002	Guru	Television	Chennai
C010	Murugan	Television	Coimbatore
C003	Yuvraj	DVD Player	Pune
C004	Gopinath	Washing machine	Coimbatore

Figure 2: C ADDRESS table

CID	C_ADDRESS
C001	Bandra, Mumbai
C001	XYZ, Pune
C002	T.Nagar, Chennai
C002	Kovil street, Madurai
C003	ABX, Pune
C004	Gandhipuram, Ooty
C004	North street, Erode
C010	Peelamedu, Coimbatore

1. How would you fragment CUSTOMER relation on the shop location attribute. 3 marks
2. How would you fragment C_ ADDRESS based on the fragment created on CUSTOMER relation. Fragment the relation C_ADDRESS, based on different location. 3 Marks

If the organization would go for fragmenting the relation CUSTOMER on the shop_location attribute, it needs to create 4 fragments using horizontal fragmentation technique as given in Figure 3 below.

Figure 3: Horizontal fragments of Figure 1 on Shop_Location attribute

$CUSTOMER_1$

CID	CNAME	PROD_PURCHASED	SHOP_LOCATION
C001	Ram	Air Conditioner	Mumbai

$CUSTOMER_2$

CID	CNAME	PROD_PURCHASED	SHOP_LOCATION
C002	Guru	Television	Chennai

$CUSTOMER_3$

CID	CNAME	PROD_PURCHASED	SHOP_LOCATION
C010	Murugan	Television	Coimbatore
C004	Gopinath	Washing machine	Coimbatore

$CUSTOMER_4$

CID	CNAME	PROD_PURCHASED	SHOP_LOCATION
C003	Yuvraj	DVD Player	Pune

Now, it is necessary to fragment the second relation C_ADDRESS based on the fragment created on CUSTOMER relation. Because, in any other way, if we fragment the relation C_ADDRESS, then it may end in different location for different data. For example, if C_ADDRESS is fragmented on the last digit of the CID attribute, it will end up with more number of fragments and the data may not be stored in the same location where customer information are stored. That is, customer 'Ram' information is stored in Mumbai and his address information might be stored somewhere else. To avoid such confusion, the table C_ADDRESS which is actually a member table of CUSTOMER, must be fragmented into four fragments and based on the CUSTOMER table fragments given in Figure 3. This type of fragmentation based on owner relation is called Derived Horizontal Fragmentation. This will work for relations where an equi-join is required for joining two relations. Because, an equi-join can be represented as set of semi-joins.

The fragmentation of C_ADDRESS is done as follow as set of semi-joins as follows.

$$\mathbf{C\_ADDRESS_1 = C\_ADDRESS \ltimes CUSTOMER_1}$$
$$\mathbf{C\_ADDRESS_2 = C\_ADDRESS \ltimes CUSTOMER_2}$$
$$\mathbf{C\_ADDRESS_3 = C\_ADDRESS \ltimes CUSTOMER_3}$$
$$\mathbf{C\_ADDRESS_4 = C\_ADDRESS \ltimes CUSTOMER_4}$$

This will result in four fragments of C_ADDRESS where the customer address of all customers of fragment $CUSTOMER_1$ will go into $C\_ADDRESS_1$, and the customer address of all customers of fragment $CUSTOMER_2$ will go into $C\_ADDRESS_2$, and so on. The resultant fragment of C_ADDRESS will be the following.

Figure 4: Derived Horizontal fragments of Figure 2 as a member relation of the owner relation's fragments from Figure 3

$C\_ADDRESS_1$

CID	C_ADDRESS
C001	Bandra, Mumbai
C001	XYZ, Pune

$C\_ADDRESS_2$

CID	C_ADDRESS
C002	T.Nagar, Chennai
C002	Kovil street, Madurai

$C\_ADDRESS_3$

CID	C_ADDRESS
C004	Gandhipuram, Ooty
C004	North street, Erode
C010	Peelamedu, Coimbatore

$C\_ADDRESS_4$

CID	C_ADDRESS
C003	ABX, Pune

0R

Create any scenario for distributed database and with the help of that scenario explain deadlock detection technique in distributed database.
Ans:

Sceneraios 2 marks

Local wait for graph 2 marks

Global wait for graph 2 marks

Q3b. Discus the following Object Oriented data modelling concepts providing an example of each concept.
Object Identity and
Object Structure

Ans: Object identity characteristics with example 3 marks

Object Structure with example 3 marks

Q4a. Write the xpath quries for the following with output
1. selects all the book titles 2 marks

/bookstore/book/title

2. selects the title of the first book node under the bookstore element 2 marks

/bookstore/book[1]/title

3. selects the text from all the price nodes

/bookstore/book/price[text()]

Q4b. What is graph database? Explain the property of graph data model for social network.

Graph database 1 marks

Property of graph data model for social network 5 marks

OR

What is Hypergraph? With the help of example explain the directed hyperedge.

Ans: Hypergraph definition 1 marks

Example of explain the directed hyperedge. 5 marks

Q5a. What is the need of image database? How images are stored in database?

Need 3 marks

How images stored 3 marks

Q5b. what are the advantages and disadvantages of distributed databases in designing distributed database and what are the issues in designing distributed database?

Ans:

Advantages and disadvantages 3 marks

Issues 3 marks