Lecture - 1

Tuesday

Date 16-07-19
Page

ADBMS

Advance Database Management System.

Data – Collection of object which can be recollect.
It is a raw entity
It is bits which form up information.
It is accepted by the computer system.
e.g. Glass, Each student's test score is one piece of data.

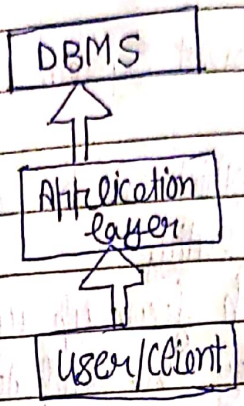
Information – Process of collection of data.
e.g. Glass is full of Water,
The average score of a class or of the entire school is information that can be derived from the given data.

Information is the message that is being conveyed, whereas data are plain facts.
Once the data is processed, organized, structured or presented in a given context, it can become useful. Then data will become information Knowledge.

Database – It is defined as collection of interrelated data and DBMS is software used to manipulate data stored into database to generate information.
or set of records or programs to organize data.
records - one student have only one record.

1-tier DBMS – Programmers communicate directly with the database for quick response, used for local application development.
→ When the database is directly available to the user for using it to store data.

2-tier DBMS -

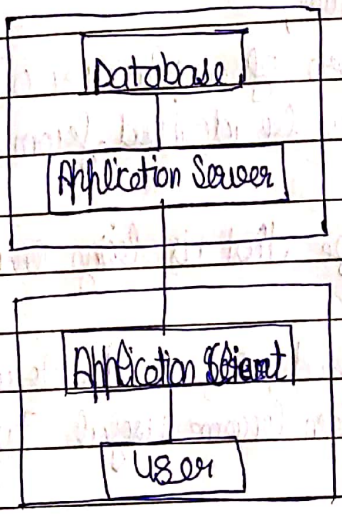


↳ Simple Client Server model

Disadvantage → ~~Sustainability~~ Scalability

→ Reliability

3 tier -



for large Web application.

Client can't directly communicate with the server.

* Logical representation or overall design of a database is called schema.

Three Schema Architecture : Three-level

↳ This framework is used to describe the structure of a specific database system.

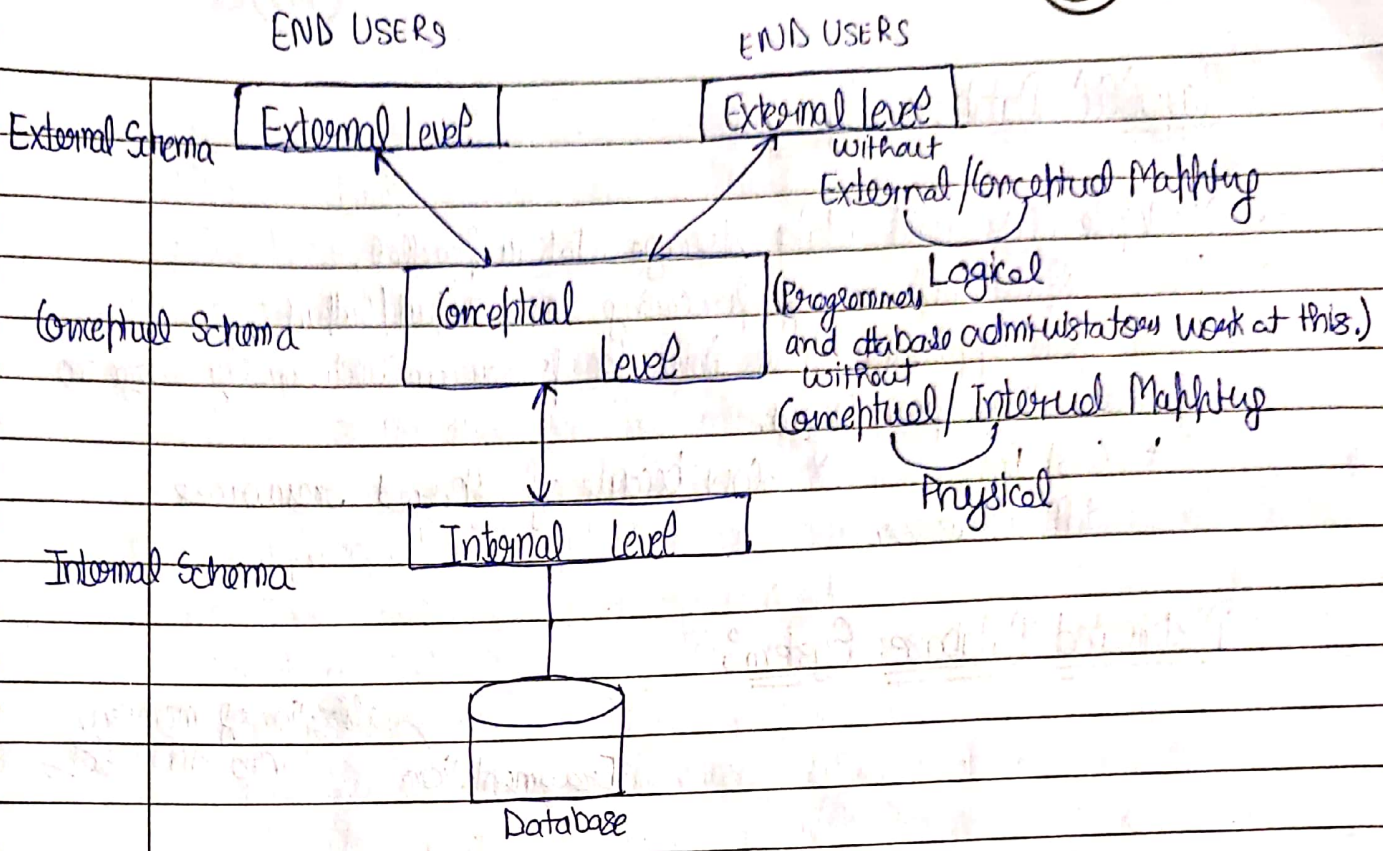
↳ Seprate user application and physical database.

External level / View

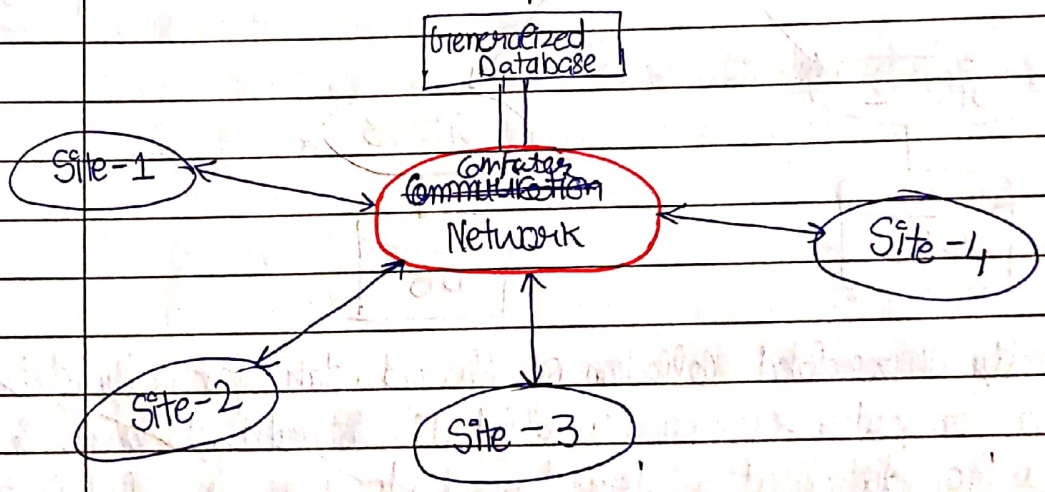
Conceptual level

Physical / Internal level

Data Independence – change one level of the database without changing the next higher level



Architecture of Generalized/Centralized Database:



Disadvantage →

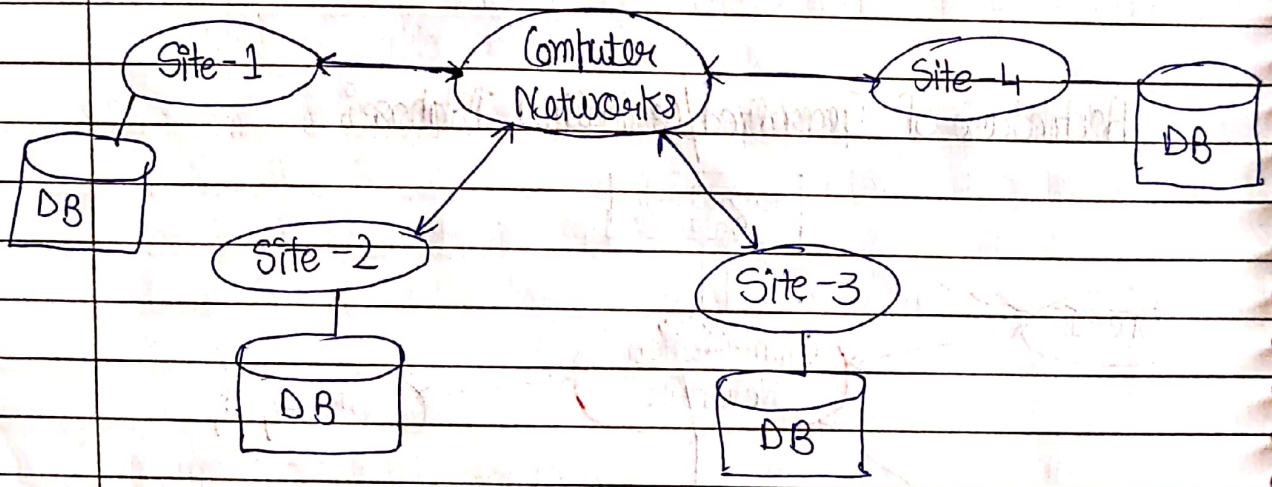
- When the central site computer or database system goes down, then every one (users) is blocked from using the system until the system comes back. – Bottle neck
  (no. of users are more – accessing the database is slow)
- Communication costs from the terminals to the central site can be expensive.
- less Reliability (If system fails, not getting access to database).

Parallel Database System:

↳ Multiple CPUs and data storage disk in parallel.
↳ Hence speed improve of processing and Input/Output
↳ Used for application that have to process an extremely large no. of transactions per second.
  * cost for diff. Processor
  * slow because of shared resources

Distributed Database System:

Fragmentation → allocation of memory into diff. sets



A logically interrelated collection of shared data physically distributed over a computer network is called as distributed database. & principle → makes the distribution transparent to user is called Distributed database.

↳ It consists of a single logical database that is split into a no. of fragments.
↳ Each fragment is stored on one or more computers under the control of separate DBMS., with the computers connected by a communication.
↳ These machines are spread geographically and connected together.
↳ Fragments allocates to diff. sites

by a variety of communication networks.

Advantages →

1. No more Bottle Neck
2. more Reliable (Copies of every database/records)
3. Better performance and great efficiency
4. Lower communication cost
5. Easily expandable (As data volumes and transaction rates increase, users can grow the system incrementally).

Disadvantage →

Recovery from failures is more difficult than others.

Data at each site is under control of DBMS.
DBMS are insight as its own rights i.e. It can handle local application independently.
Each database in dbms in a distributed system must have atleast one local application.

It can be classified into two parts – Heterogenous DDBMS
Homgenous DDBMS

Heterogenous DDBMS –

1. Different sites used dissimilar schema and software.
2. System may be composed of relational, network, hierarchical and object oriented DBMS's. (variety of DBMS).
3. Query Processing is complex due to dissimilar schema.
4. A site may not be aware of other site and there is limited operation and processing user request.
5. It has different type of interface.

Date 17/07/19
Page

Wednesday

Homogeneous DDBMS -

1. It has indentical database or dbms and also have identical operating system.
2. It has style Interface
3. Each site is aware of all other sites and cooperate with other sites to process under request.
4. They are easier to design and manage.

Lecture - 2

Fragmentation occurs in a dynamic memory allocation system when many of free blocks are too small to satisfy any request.

- It is a task of dividing a table into a set of smaller table.
- The subset of table are fragmentation.
- Data can be stored in different computers by fragmenting the whole database into several pieces called fragments. Each piece is stored at a different site.
- It is logical data units stored at various sites in a distributed database system.

Advantages of Fragmentation :

1. Usage : In general, application work with views rather than entire relations. Therefore, for data distribution, it seems appropriate to work with subsets of relation as the unit of distribution.
2. Efficiency : Data is stored close to where it is most frequently used. In addition, data that is, not needed by local application is not stored.
3. Parallelism : With fragments as the unit of distribution, a transaction can be divided into several sub queries that

operate on fragments. This should increase the degree of concurrency, a parallelism, in the system, thereby allowing transactions that can do so safely to execute in parallel.

4. Security: Data not required by local application is not stored, and consequently not available to unauthorized users.

Disadvantages of fragmentation:

1. Performance: The performance of global application that requires data from several fragments located at different sites may be slower.
2. Integrity: If control may be more difficult if data and functional dependencies are fragmented and located at different sites.

Types: (Semi-Join operations)
1. Horizontal Fragmentation (Rows) (Attribute must be same)
   a) Primary Horizontal fragmentation (Basic operators >, <, >=, <=, =)
   b) Derived Horizontal fragmentation (AND, OR, NOT)
2. Vertical Fragmentation (Columns) (Natural-Join operations) (Rows must be same)
   a) Grouping (Bottom-up) (By adding new attributes)
   b) Splitting (Top-down) (Big to Small) (divided into no of fragmentation)
3. Hybrid Fragmentation

Horizontal Fragmentation:

1. Groups the tuples of a table in accordance to values of one or more fields.
2. It also confirmed to the rule of reconstruction.
3. Each HF must have all columns of original Base tables.
4. All fragments are stored at diff. site and each fragmentation has diff. tuple.

LECTURE - 3

Date 19/07/19
Page

Friday

Architecture – Organizing the Database

- Client - Server
  - Peer to Peer
  - Multilevel
    - 2-tier
      - Master - Slave
      - Client - Server
    - 3-tier

Same side Client-Server

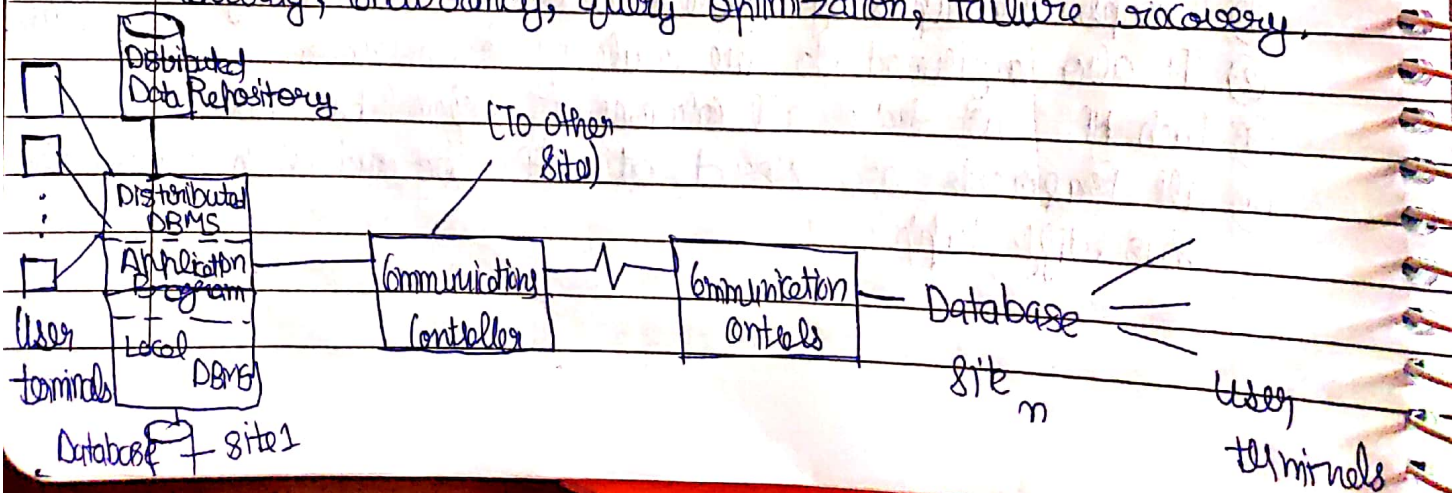
All the sites act as distributed system

Single Client - Single Server
Multiple Client - Multiple Server
- Sub
- Main
  - Only Responding the Request

Global Schema → Logical Representation of whole database.

Distributed DBMS architecture

Functions :-
- Locate data with a distributed data dictionary.
- Determine location from which to retrieve data and process query components.
- DBMS translation between nodes with different local DBMSs (using middleware).
- Data consistency (via multiphase commit protocols).
- Global primary key control
- Scalability.
- Security, concurrency, query optimization, failure recovery.



Local Transaction Steps:
1. Application makes request to distributed DBMS.
2. DDBMS checks distribute data repository for location of data. find that is local.
3. DDBMS sends request to local DBMS.
4. local DBMS processes request.
5. Local DBMS sends results to application.

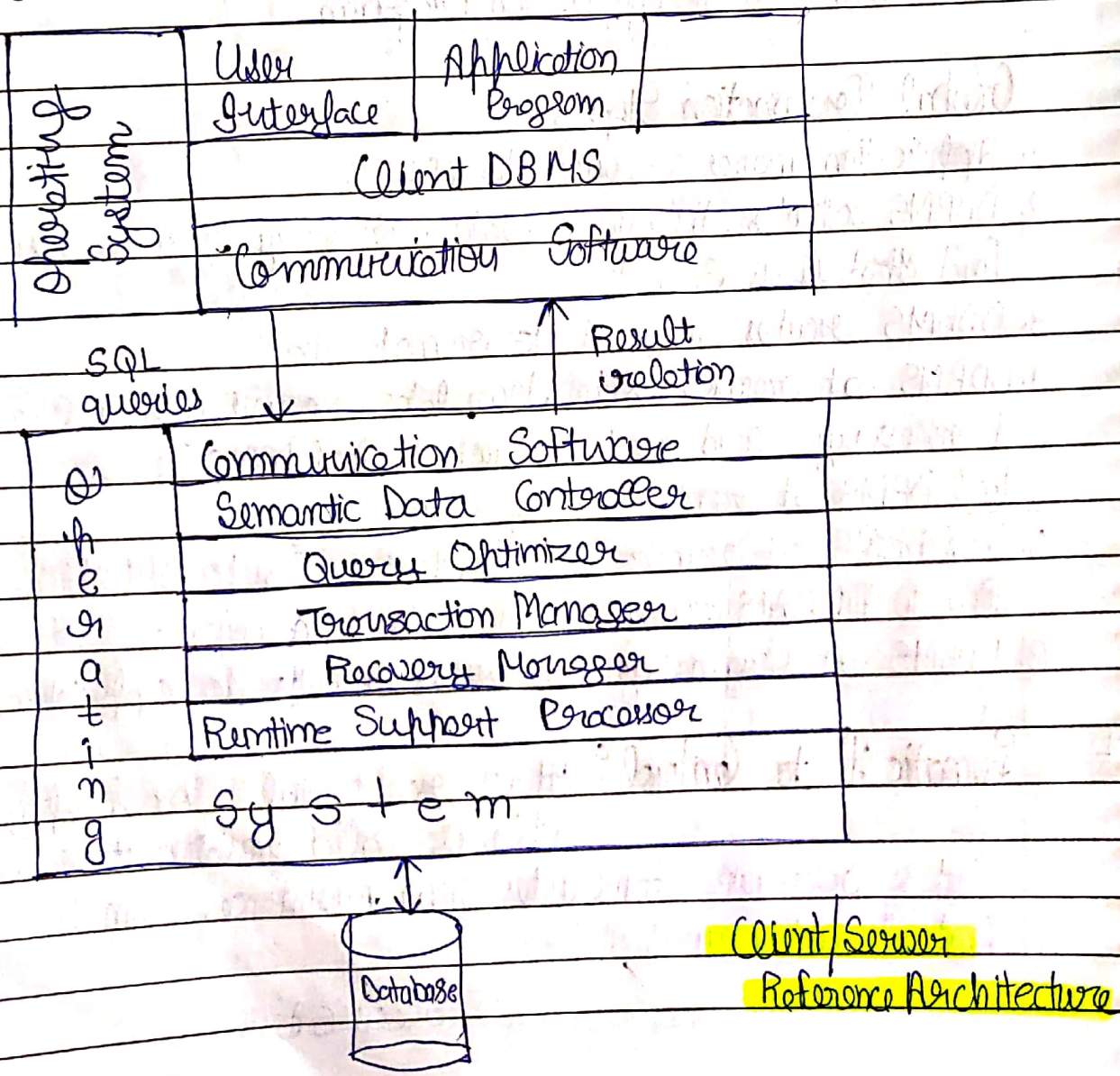
Application Program → DDBMS
DDBMS ↔ Database data repository
DDBMS → local DBMS
Local DBMS ↔ Database Site 1
Local DBMS → Application Program

Global Transaction Steps:
1. Application makes request to DDBMS.
2. DDBMS checks distribute data repository for location of data. find that it is remote.
3. DDBMS routes request to remote site.
4. DDBMS at remote site translates request for its local DBMS if necessary, and sends request to local DBMS.
5. local DBMS at remote site processes request.
6. local DBMS at remote site sends results to DDBMS at remote site.
7. Remote DDBMS sends results back to originating site.
8. DDBMS at originating site sends results to application.

* Semantic Data Control: It is responsible for keeping a database valid relative to a specified set of constraints. It includes view maintence, semantic integerity control and security control.
  - data protective
  - access control

Client/Server Systems:

- This provides two-level architecture which make it easier to manage the complexity of modern DBMS, where the functionality is divided into servers and clients.
- The server functions, primarily deals with data management, query processing, optimization, transaction and storage management.
- Client functions includes mostly user interface. (management the data that is cached to the client, management the transaction locks).
- This is quite common in relational systems where the communication between the clients and servers is at the level of SQL statements.

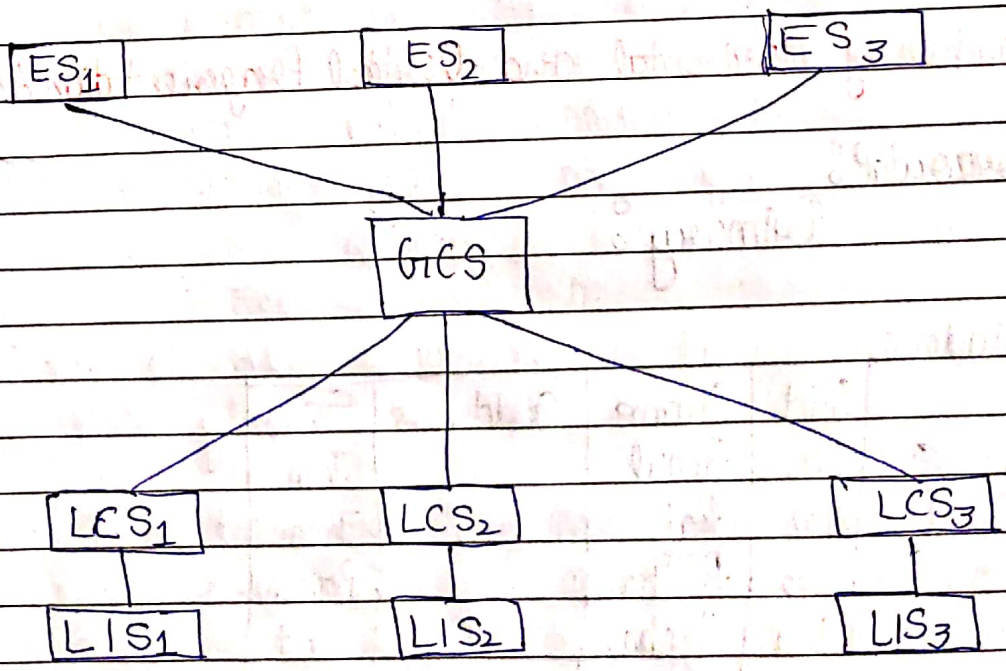


Client/Server Reference Architecture

Schema
↳ graphical defination of the database structure

Peer to Peer:
- Unlike Client/Server, peer to peer model doesn't differintiate between client & server.
- Instead each node can either be a client or server depending on whether the node is requesting or providing the service.
- This architecture generally has 4-level of Schema:

1. Local Conceptual Schema (LCS): describes the logical organization of data at each site.
2. Global Conceptual Schema (GCS): describes the enterprise's Global (Logical) view of the data.
3. Local Internal Schema (LIS): It is an individual internal Schema defination at each site. (Physical Data Organisation) at each site.
4. External Schema (ESs): (Work with the user view of data), support user applications and user acess to the database.



Peer-to-Peer Reference Database

(not in Syllabus)

Multi-DBMS architecture: It can be expressed through 6-level of schema:

1. Multi Database view level: Multiple User views comprising the work of this schema. (subset of integrated distributed System).
2. Multi Database Conceptual level: Integrated Multiple Database that comprises of Global, logical, Multidatabase structure definition.
3. MultiDatabase Internal level: Data distribution occurs at different sites and Multi-database to fold data mapping.
4. Local Database View level: Public view of local data.
5. Local Database Conceptual level: Work with local data occurance at each site.
6. Local Database Internal level: Physical data occurance at each site.

Examples of Horizontal and Vertical Fragmentation:

Horizontal:

Primary:

Student

	S.id	SName	SAddress	Ssub
F1	101	Rahul		JAVA
	102	Raj		JAVA
F2	103	Ritika		PYTHON
F3	104	Raghav		C++

F1 → Select * from Student where Ssub = "JAVA";

Derived: R_1 , R_2

R_1 → owner Table (Table that is fragmented)

R_2 → Manager Table (Attribute using) Or Member Table

R_1 (empid, eName, Sal)
PK → Primary Key

R_2 (empid, ProjectId, Role)
FK → Foreign Key
PK

DT <owner>

	DNo	DName	Budget	Loc.
F1	D1	CSE	7K	Mumbai
F2	D2	Mech	3K	Delhi
	D3	ETC	10K	Delhi
F3	D4	MCA	5K	Jaipur

PK

PT <Member>

PNo	PName	Budget	DNo
P1	DST	13K	D2
P2	AICE	15K	D3
P3	avn	16K	D2
P4	Architecture	13K	D1
P5	Medical	19K	D1

PK FK

Step 1 → Create Fragments of owner table

F1 → Select * from DT where loc = 'MUMBAI';

F1

DNo	DName	Budget	loc
D1	CSE	7K	Mumbai

Step 2 → Perform semi-join operation on PT relation and fragments.

Syntax: Create table Project1 as select PNo, PName, Budget, PT.DNo from PT, where PT.Dno = f.DNo.

Project1 PT ⋈ F1 PT ⋈ F3 Project3

Project2 PT ⋈ F2

PT.DNo	P.No.	PName	Budget
D1	P4	Architecture	13K
D1	P5	Medical	19K

Project1

22/07/17

LECTURE – 4

Monday

SQL Performance Tools: To check the performance of your database.

→ Directly Accessing Database is not good, any failure can occur.

→ Frequently Accessing Data from Database, to increasing the performance, view is used. (efficiency)
↳ into a view
↳ act as interface b/w database and enduser.

Simple View: working with single data

Complex View: working with multiple data.
↳ no need to access the Database

1. View is virtual or logical tables for the base (original) table.
2. View doesn't store data but it only store the query statement.
3. View will create with help of select statement.
4. View is used to retrieving data from table.

Two types of views are there:

1. Simple View: It is access the required data from a single Base table.

→ It can access a support or create DML operation or access it.

→ Simple is view is also called updatable view.
DML → Select, insert, update, delete
DDL → Create, drop, alter, truncate, rename, comment
DCL → Grant, Revoke
TCL → Commit, Rollback, Savepoint, Set transaction

Syntax of Simple view : (also for Complex view)
Create view <viewname> as Select * from <Table Name>
[where (condition)];
optional

e.g.

Emp

Eid	EName	Sal
101	Jones	25000
102	Scott	15000
103	Allen	30000
104	Smith	60000
105	Warner	48000

→ Create view v1 as Select * from Emp;
(as → not case sensitive)
→ insert into v1 values (106, 'ABC', 68000);
→ update v1
set Sal = 80000 where Eid = 103
→ delete from v1
where Eid = 106;

2. Complex view : They are applicable when we access the record data from multiple base table.
→ It doesn't support all DML operations on multiple base table at a time.

→ It is also called as non-updatetable view.

e.g.

Emp-PS

Eid	EName	Sal
101	Jones	25000
102	Scott	15000
103	Allen	30000
104	Smith	60000
105	Warner	48000
106	ABC	68000

Emp-BS

Eid	EName	Sal
101	Jones	25000
109	PQR	90000
110	STV	88000

Syntax: Create view <Viewname> as select * from <tablename>;

→ Create view cv1 as select * from emp_ps U select * from emp_bs;
Join
Output contain 8 Records
→ select * from cv1;
→ Other operations are not allowed except select, Give error - Constant or derived field.

* Diff. between Complex and Simple View in CSA - Lecture 6.

Tuesday

LECTURE-5

23/07/19

Index:

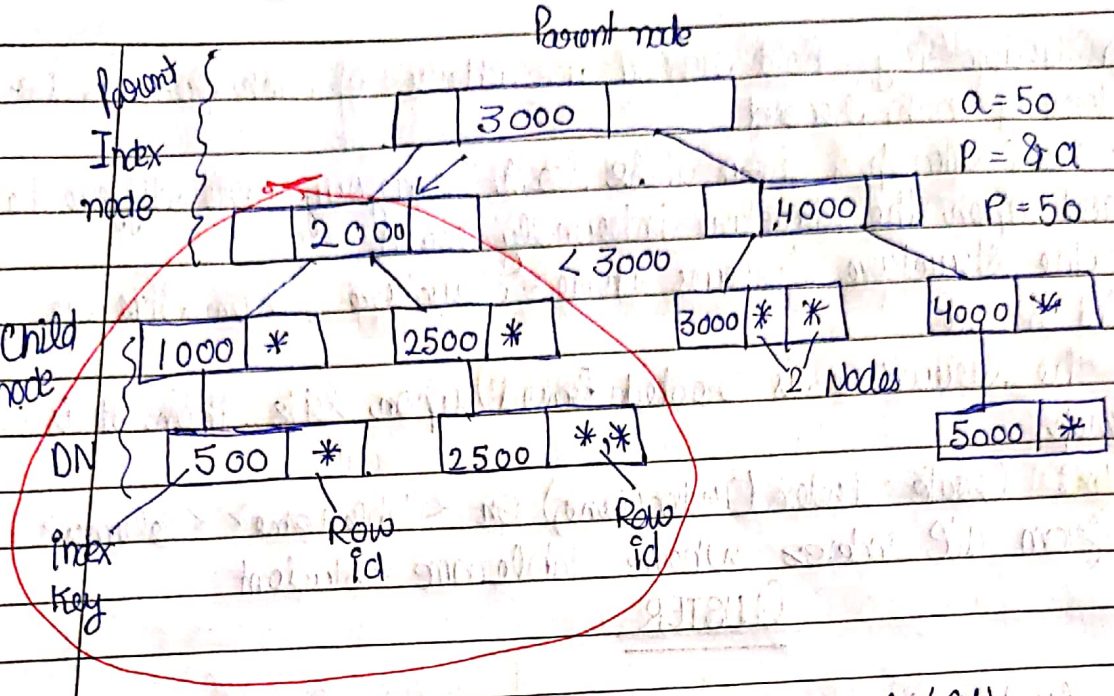
select * from emp where sal = 3000;

Emp (not index)

	Sal	ename	emp
1	5000		
✓ 2	3000		
3	1000		
4	2000	SD	
✓ 5	3000		

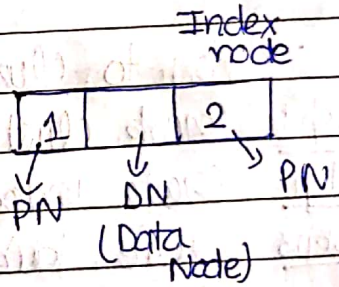
time consuming (To check sal = 3000).

→ Index ↑ performance and efficiency of the database system.
→ Index is applied to a particular column.
It is categorised into 2 types:
(i) B+ TREE
(ii) BITMAP



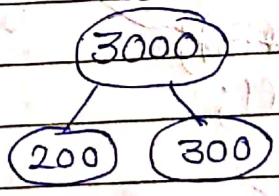
Syntax: Create index (x) on emp (sal)
(x) → index row
emp → table
sal → column

1) Table Scan → linear
2) Index Scan → Non-linear data-structure



Pointer → Holds the address of another variable.
Row-id → Record-Id

Syntax:



1) Set Auto-trace on explain → Show Index
2) Index
→ It is a database subject that makes data retrieval faster.
→ DB index works the same way as an index in text book.
→ Index created on column and that col is called index-key.

→ Each index-key hold row-id i.e. address of each record, i.e. address of each record.
→ B-Tree, Bitmap, these both category are differentiated based upon the structure internally created.
→ If the structure follows tree-structure, then this is called B-tree.
→ If the structure is created based upon bits then it is called Bitmap.

Syntax: Create index (indexname) on <Tablename> <Column>

Show / Select * from all_indexes where tablename = student

CLUSTER

A cluster is a db object that stores data related to two or more tables in a single disc space.

→ How to cluster the table.

Step1 Create cluster.
Step2 Create index on cluster.
Step3 Create table.

```
Create cluster <cluster name> (col.name Datatype);
create cluster c1 (dno number(3));
Create index ci on cluster c1;
                 (index name)

Create table dept (dno number(2), dname varchar(20),
                   cluster c1 (dno));
Create table emp (emp_no number(2), ename varchar(20),
                  cluster c1 (empno));
Select row_id, d_no., d_name from dept;
Select row_id, emp_no, ename, d_no from emp;
```

Mon
~~Wednesday~~

Date 29/07/19
Page

LECTURE-6

Sequence:

QUERY PERFORMANCE TOOLS:

1. INDEX
2. VIEW
3. SEQUENCE
4. CLUSTER

- SIMPLE (Create only one index)
- COMPOSIT (From more than one diff. indexes)
- UNIQUE (Primary Index) (Id)
- NON-UNIQUE (normal Index)
- BIT MAP (Oracle 9i support) → FAIL UNIQUE INDEX doesn't
- B-TREE (Searching and Order of log(n))
- FUNCTION (UPname, ahhley function)

Complexity
$O(n) - n^2$

In the case of query performance tools, It will increase the performance of query based on efficiency, Time Complexity, Searching Time of the data, using tools, We can increase the performance of searching data: ① Index
② View
③ Sequence
④ Cluster.

1. INDEX: It is used by query to find the data from table quickly. It is internal peices of sql. ~~Explicit~~ In generally, We can create an index for a particular table.

Syntax:

a) create a table of full time emp ~~emp~~ (FTE) where data is emp_id, emp_name, gender, designation, salary.
b) insert the data (minimum 5 entries).

① Simple Index: create index <index_name> ON <table_name> (attribute);

31/07/19
Wednesday

LECTURE - 7

It will create in a particular table and sql engine by default create based on the primary key value.

Rules → a) Manipulation performed very less, in which we create index. (Bad effect on performance of database).

I ← Implicit (Generate automatically)
✓ Explicit (User-defined)

→ DROP index <index_name>;

(2) Composit Index : create index <index_name> ON <table_name> (empname, sal);
→ DROP index <index_name>;

(3) Unique Index : can be created on unique column (means don't have duplicate value). Once unique index is created the duplicate value not accept in that particular column. Unique index ensure entity integrity in a table.

Syntax :
→ create unique index <index_name> on <table_name> (at input).
→ drop unique index <table_name>

(4) BIT-MAP : used for millions of rows for uniqueness
→ Create BITMAP INDEX (index_name) ON <table_name> (column_name)

(5) FUNCTION BASED INDEX :
→ create index <index_name> on <table_name> (<function>, column);
(upper (emp_name));

⑥ Multi Index on table:
We can create more indexes in a particular table but it will decrease performance of SQL query.

Cluster Index:

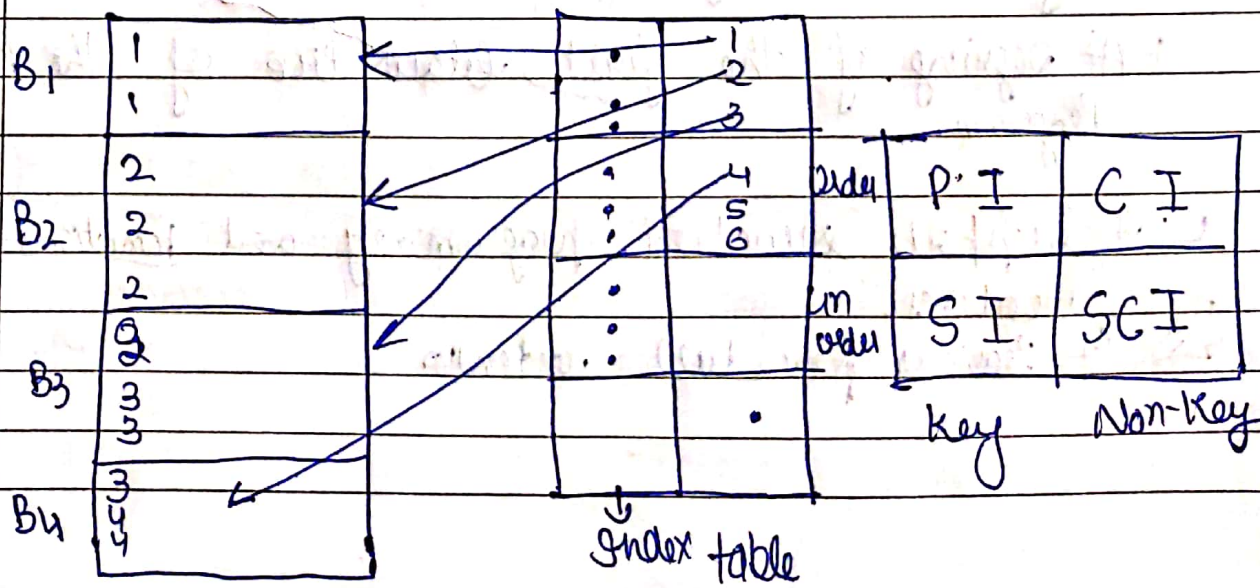
Cluster is a schema object that stores data from one or more table each have more attribute i.e.
2 types:
① Index Cluster ② Hash Cluster

→ Create cluster
→ Create index using cluster
→ insert data using cluster index
→ drop

→ Create cluster <cluster_name> (d ~~number~~ number (2));
→ Create index <index_name> on cluster <cluster_name>.
→ Create table <dept> (insert data)
→ Create table <emp> (insert data)

B|T TREE

Sql > select rowID, Dno, Dname from <tablename>;
Sql > " " empname, eno. from <tablename>;



06/Aug/19

LECTURE - 8

Tuesday

PL/SQL

Procedural Extension language for SQL

Structure define and way of calling the function

1. Block Header (It defines the Procedure and Function) acc. to Requirement
2. Declaration Header (Required no. of variables, constants, identifiers) used in program
3. Execution Section (Execute the Program)
4. Exception Section (Runtime errors)

END;

As Begin Exception

class
System.out.println
object class function

```
sprint : 'Hello World'

DECLARE
Message varchar2(20) := 'Hello, World!';
BEGIN
dbms_output.Put_line(Message);
END;
```

Assignment operator

To Run the Code from the SQL command line, you may need to type '/'

At beginning of the first blank line of the PL/SQL Program.

~~It supports structured programming and functions and procedures.~~

~~It provides procedural extension~~

↳ It supports structural Programming through function and Procedure.
↳ It supports Object oriented Programming concept.
↳ It supports the development of Web Application and Server Pages.
↳ PL/SQL is not a Case Sensitive and uses same datatype as SQL.
↳ Block Headers define the type of Block i.e. Procedures & Functions. And also its defines the way it is called.
↳ Declaration Section: It contains defination PL/SQL identifiers such as Variable, Constants and so on.
↳ Decletion statement always start with Keyword DECLARE
↳ Excaition Section: It contains executable Statements that has allowed to manipulate the variables that are declare in declaration section.
↳ It always begin with the Keyword BEGIN and ends with the Keyword END.
↳ This is the mandatory section of PL/SQL. It supports all DML commands and SQL * built in functions.
↳ It also supports DDL command using native dynamic SQL or DBMS_SQL Package built in

SET SERVEROUTPUT ON; (write in Starting)

PL/SQL Operators:-

	Operators	Operation / Description
1)	+	Addition
2)	-	Subtraction
3)	* /	Mul, div

Operator	Operations
%	Attribute Indegator
'	Character string delimitator
.	Component Selector
(,)	Expression or list delimitator
:	Host variable Indicator
,	Seperator
=, >, <, =>, <=	Relational Operator
@	Remote Access Indicator
;	Statement termiciator
:=	Assignment Operator
=>	Association Operator
||	Concatation Operator
**	Exponticial operator
/*, */	Multiline Comments
--	Single-line Comment
..	Range Operator
<>, <=, >=, ^=	Not-Equal to Operator

7/08/19 Wednesday

LECTURE-9

WAP to print the numbers from 1 to 5.

Loop & Initialization
↳ Increment / Decrement
↳ Condition check

```
Declare
  i number;
Begin - - - - - - - - - - - -
  i := 1;
Loop
  dbms_output.Put_line (i);
  i := i+1;
  exit when i>5;
end loop;
end;
```

Initialization — Increment — Condition check

8/8/19 Thursday

LECTURE-10

Variable_name datatype [NOTNULL := value];

```
SQL > SET Serveroutput ON;
SQL > Declare
        Var1 integer;
        Var2 Real;
        Var3 varchar(20);
      Begin
        Null;
      End;
```

In both

	Static	Dynamic
a	a=5	&a
b	b=6	&b
c=a+b		

```
Declare
a number := &a;
b number := &b;
c number
Begin
    c := a+b;
DBMS_output_line(c)
   End;
/
```

⇒ Set Serveroutput on is used to display the buffer used by the dbms output.

⇒ After End; '/' It tells SQL *+ to execute the Block.

Exception Handling:

In PL/SQL Exception uses two types:

1. System defined Exception
2. User defined Exception

1. System defined Exception: This Exception is defined by Oracle. There are different types of System defined Exception are;
   1. Zero_divide
   2. Value_error
   3. Invalid_number
   4. No_data_found
   5. Too_many_rows
   6. dup_val_on_index
   7. Cursor_already_open
   8. Invalid_cursor

1. Zero_divide → This Exception arises when you try to divide a number by 0.

Eg. a:=&a;	10	10
b:=&b;	5	0
c:= a/b;	2	Zero-Divide Exception.

2. Value_error → This Exception arises if the variable size is mismatch or data-type is mis-match.

e.g. ① X number(4) := 10000;
② X number(4) := 'abc';

3. Invalid_number → This Exception, it arises when you try to perform any invalid mathematical operation or invalid numeric operation.

e.g.

sysdate + 10 ✓
6-feb-2019 + 10 ✗

4. No_Data_found → This Exception arises if data not found in table.

e.g.

Vemo := &empno
select ename into Vename from employee where empno = Vemo;

5. Too_many_rows → It arises if the select statement try to fetch more than one record.

e.g.

Vdno := &deptno
select dname into Vdname from employee where deptno = Vdno;

6. dup_val_on_index → It arises if you try to insert the duplicate value to primary column.

e.g. Create table emp (eno number Primary key);
Insert into emp values (100);
Insert into emp values (100);

(2) User-defined Exception → Exception define by user, there are two types of user-defined Exception:

(1) Rise Statement
(2) Rise application error

a) Rise exception → We Rise the name of Exception.
b) In Rise application error exception, the exception will generate by code.

If user has arise the exception in rise statement then this exception is handled by user itself.
Otherwise its optional.

19 August 2019

LECTURE - 13

Monday

Semi-Structured Database ← XML (Xtendable Markup Language)
↓
e.g. Relational database
↓
Structured

↓
opening and closing tags

Extensible Markup Language
It is define by www. consortium, it is derived from standard generalized markup language (GML), enclosing the text in between two tags - opening and closing tab. e.g.
< Employee name - Nikita > </ Employee name >

↳ XML tags are user defined tags whereas HTML tag are pre-defined tags.
↳ XML tag is case sensitive. Whereas HTML is case insensitive.
↳ XML tag is used to define or describe the data. Whereas HTML tag is used to display the data.
↳ XML is platform and language independent and having one version 1.0.
↳ XML database is used to store huge amount of information in the XML format.
↳ There are two major type of XML database → ① XML enabled ② Native XML

① XML enabled → It is nothing but the extension provided for the conversion of XML document. This is a relational database where data is stored in tables consisting of rows and columns.

② Native XML → It is based on the container rather than table format. It can store large amount of XML document and data native XML database has an advantage over the XML enabled database.

It is highly capable to store, query and maintain XML document than XML enabled database.

Structured Data :- The Data has a structure and is well organized either in the form of tables or, in some other way called structured data. Searching and accessing information from such type of data is very ~~data~~ easy. e.g. Relational Database, spreadsheet.

Unstructured Data :- It is a data which is not organised in a pre-defined manner or doesn't have a pre-defined data model. e.g. image and graphics, pdf file, word document, audio, video, emails, power-point presentation etc.

Semi-Structured Data :- It is an information that doesn't reside in a relational database but that have some organizational property that make it easier to analysis. e.g. XML and non-SQL Database are considered as semi-structured database.