INDEX

My Self Imran Khalid, 15DCS0019,
Dip. in Comp. Engg (3rd year, 6th sem).

S. No.	Date	Title	Pg. No.	Remarks

Advanced DBMS

DCO-601

Advanced Data Base Management System (DCO-601)

University Polytechnic, Jamia Millia Islamia, N.D- 25

Q → How google stores the data?

EID	Contact NO.	Hours	Name	EmpID	Location
E1	C1	72	suresh	C1D1	Delhi
E2	C2	48	Ramesh	C1D1	Delhi
E3	C3	24	Ganesh	C1D2	mumbai
E4	C4	24	Brijesh	C1D2	Mumbai

- Super Keys → Candidate Keys & primary key

Book -ID	Book Name	Author
B1D1	DB computer	Korth
B1D2	DB computer	cello
B1D3	computer Net.	chu
B1D4	Algo. computer	Bhu
B1D5	E concept	ullu

- super key → unique alternative or group of unique alternative.

- Candidate Keys → minimal of superkey or super keys without redundancy or super key that cannot we further divided when groups are divided then things should not uniquely identify records.

Primary key → Any one of the candidate key choosen is called primary key.

[Database Model]

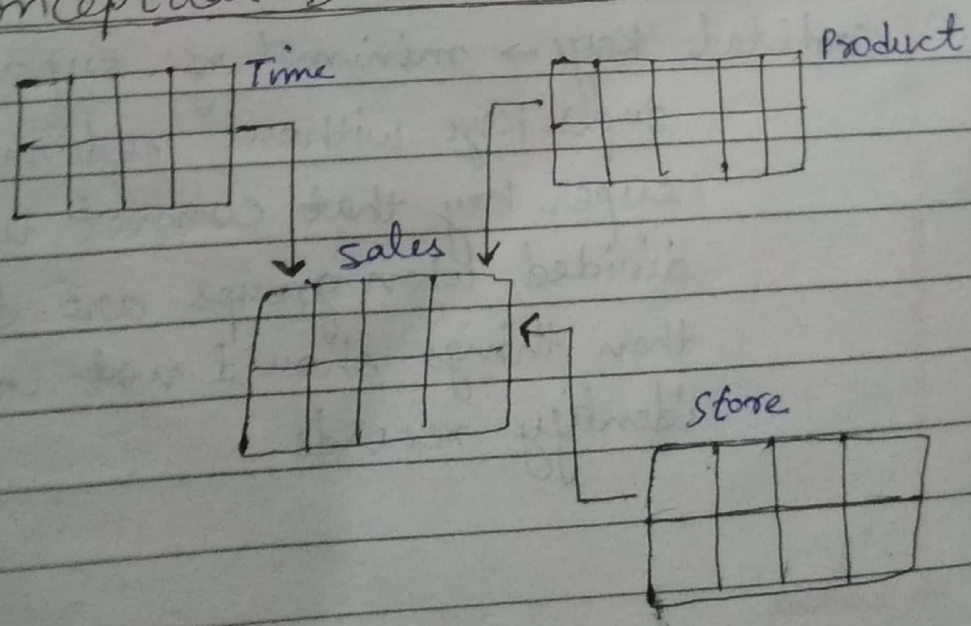
Abstract Representation

Types of DM –

1. Conceptual Database Model
2. Logical DM
3. Physical DM

A database model is a correction of concepts or notation for describing Data. Data Relationship Data and Data Constraints.
These data models also include a set of basic operations for manipulating data in the database

(1.) Conceptual Database Model :-



A high level conceptual data modal provide concepts for presenting data in ways that are close to the way people presence data. This modal identifies highest level relationship between different entities.
The features of C.D.M are :
1.) highly Abstract.
2.) Easy to understand.
3) These models can be easily enhanced.
4) only entities are visible.
5) No attributes is specified.
6) No primary key is specified.

(2.) Logical Data Modal :-
A logical Data Modal describes the data in as much detail as possible without suggesting as to how they will be physically implement in the database.
The features of LDBM are :
1.) Includes all entity and relationship among them.
2.) All attributes for each entity are specified.
3) The primary key for each entity is specified.
4) Normalization occur at this level of modelling.
5.) It is more detailed than the conceptual model.

6.) This modal is independent of the types of database to be used.

7.) Modification of this model requires the conceptual relatively more efforts than the conceptual data modal.

8.) Data modelling tools like is ERwin and PD (Power Designer) are used to create logical Data model.

3.

(3.) Physical DM

It represents how the modal will be built in a particular database.

A PDM shows all the table, structure, column name, data type, constraints, primary key, foreign key, Relationship between tables.

The features of PDM are:

1.) Specification of all tables and columns.

2.) D-Normalization may occur at this level based on user requirement.

3.) This model is different for different RDBMS.

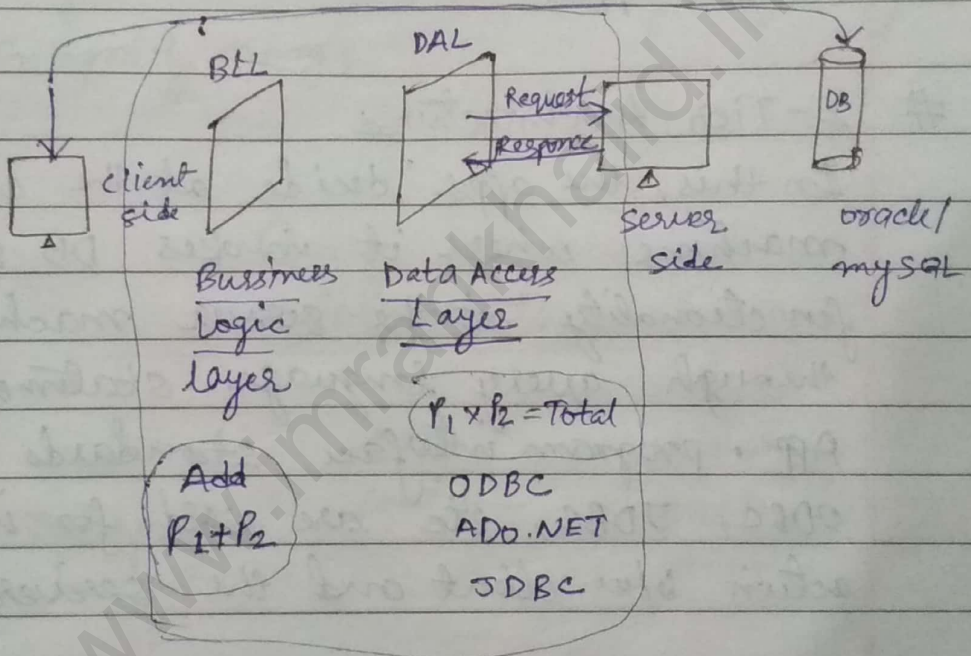
4.) It is difficult to port a particular physical data model to a different database once a designed is finalized.

Date 02/02/2018 Page no.

5.) Toole like ERWin & power desing are used to code logical data model to physical data model.

Database Architecture

① 2 Tier / level architecture
② 3 Tier / level Architecture



3-Tier

3-Tier Architecture

In a 3-Tier Arch. of a Database, the client ~~bank~~ machine act as a front-end and doesn't contain any direct database call instead the client communicates with an application server through a form / any interface. This app. server in teams communicate

Date / / Page no.

with a DB system to access data, the bussiness logic determines what action to carryout and under what conditions. This bussiness Logic Layer is embedded in the application server. 3-Tier Applications are more appropriate for the apps that run on the web it also finds its implementation in large apps.

2-Tier Architecture

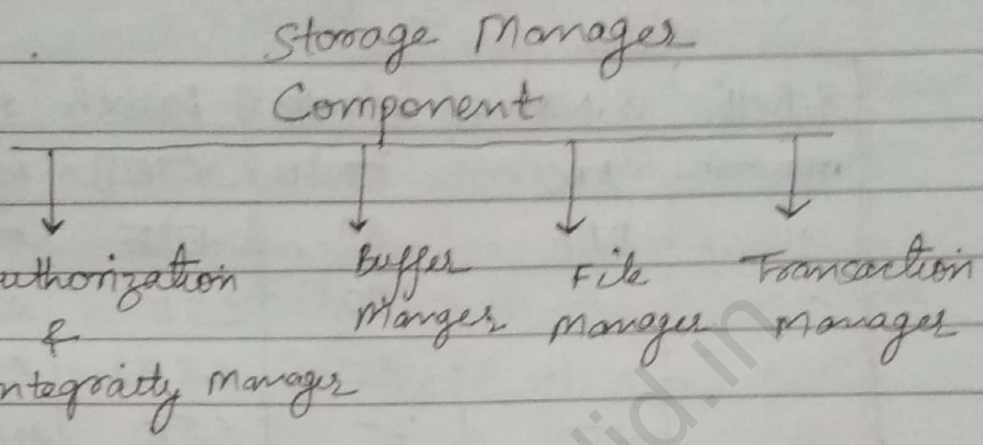
In this, the apps. decide at the client machine where it invokes DB system functionality at the server machine through query language statements App. program interface standards like ODBC, JDBC, etc are used for interaction b/w client and the server.

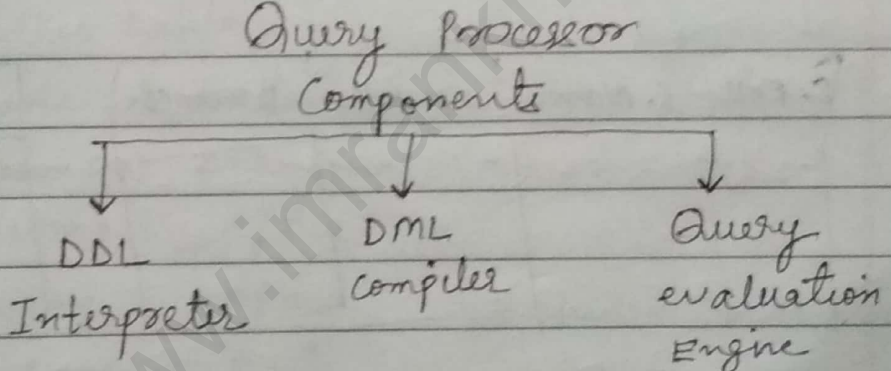
- Functional components of DB system

1. Storage Manager
2. Querying / Query Processor

Storage Manager, is the component of the database system that provides the interface b/w the low level data stored in the DB and App.

programs and queries seperated to the system. It is responsible for storing, updating, retrieving the database.





Meta Data

1. schema
2. Constraints
3. Indices

Normalization

Q → what are the stages involved in a DB design process?
(from scratch → End)

Date / /

Page no.

Anomalies

1. Insertion Anomaly
2. Delition Anomaly
3. Updation Anomaly

S.Roll	S.Name	marks	Branch	HOD
19DES0001	ABC	70	CSE	MSD
" 2	DEF	80	CSE	MSD
" 3	- -	90	CSE	MSD
"	- -	85	CSE	MSD
"	-	75	CSE	MSD

S. Roll	S. Name	marks	Branch

Branch	HOD

Normalization is a process of organizing data in a database, its primary intention is to decompose a given table into various relations that helps in reducing data redundancy and other undesirable characteristics by like insertion, Updation and deletion anomolies, the purpose of normalization is to reduce a

Data → singular

Date / / Page no.

redundancy as much as possible and to ensure that the data ~~is~~ that store are logically related.

1 - Normal Form

A table is said to be in 1-N form if all the values stored in the table are atomic.

No two attributes in the given table should be identical.

Functional Dependency

- Lakes Database

Name	Continent	Area	length
Capsian sea	Asia - Africa		
victoria	Africa		
Tangang	Africa		
Aral sea	Arab		
Florida sea	Arab		

	SID	SName	Subject	Score	Teacher
t_1	1	ABC	Java	70	Mr. A
t_2	2	ABC	C++	80	Mr. B
t_3	3	DEF	C	90	Mr. C
t_4	4	XYZ	DBMS	80	Mr. D
t_5	5	MNO	Java	100	Mr. E

FD → schema
↳ changes rarely

Date 16/02/2018

Page no.

Diff. types of constraints imposed on all permissible data of a relation or set of relations can be determined through the use of Functional Dependency. These FD arise naturally due to requirements & restrictions that exist in the real world and that needs to be stored in diff. relations.

Given a relation (R) and two set of attributes A, B. the attribute A functionally determines attribute B w.r.t this relation (R). $[A \to B]$. if and only if (iff) for any two ~~peoples~~ tuples $t1, t2$ of R whenever $t_1(A) \neq t_2(A)$ then $t_1(B) = t_2(B)$

① $R(A, B, C, D, E)$

$A \to B$

$AB \to C$

$BC \to E$

$BD \to E$

$[AD]^+$ → closure of AD

↑ Not on right side of R.

$[AD]^+ = \{ADBCE\}$

$[A]^+ \to \{A, B, C, F\}$ Not candidate key

$[D]^+ \to \{D\}$

Date / / Page no.

② R (A B C D E F)

A → C
C → D
D → B
E → F

$[AE]^+$ → closure of AE

✓ CK

$[AE]^+ = \{AECDBF\}$

$[A]^+ \Rightarrow \{ACDB\}$ ×

$[E]^+ = \{F\}$ ×

Prime Attributes = A, E
Non Prime = {B, C, D, E}

③ R (A, B, C, D, E, F, G, H)

CH → G
A → BC
B → CFH
E → A
F → EG

$[D]^+ \rightarrow \{D\}$ ×

$[DA]^+ \rightarrow \{DA\ BC\ FH\ EG\}$
$[DB]^+ \rightarrow \{DB\ CFH\ EG\ A\}$
$[DC]^+$ ×

CK = {DA, DB, DE, DF

DELTA My NoteBook

Date / /

Page no.

④ R(ABCDE)

$AB \rightarrow C$

$C \rightarrow D$

$D \rightarrow E$

$A \rightarrow B$

$C \rightarrow A$

Proceed with first combination (AB),

$[AB]^+ \rightarrow \{ABCDE\}$

$A^+ \rightarrow ABCDE$ $\quad A^+ \rightarrow R$

$[\because C \rightarrow A$
so, $C \rightarrow R]$

$CK = \{A, C\}$

Prime Attributes → A, C

Non-Prime Attributes → B, D, E

⑤ R (A, B, C, D, E)

$A \rightarrow B$

$AB \rightarrow C$

$B \rightarrow F$

$D \rightarrow C$

$E \rightarrow A$

⑥ R (ABCDEF)

$A \rightarrow C$

$B \rightarrow D$

$C \rightarrow E$

$D \rightarrow E$

$E \rightarrow A$

$F \rightarrow B$

Date / / Page no.

- **Rules of 1-N Form**

1. Each coloman of the table must be single value, that is, they should not contain multiple values or non-atomic values.

2. At Each colourm, the values stored must be of the same type.

3. All the attributes must have a unique name.

4. The order of records in the given relation doesn't matter.

[Example]

2-Normal Form

A DB is said to be in 2-N form if the following conditions are satisfied:

1) It is in 1-N form
2) All the Non-Key Attributes are fully functionally dependent on the primary key.

20/02/2018

Date / X /

Page no.

Score ID	S_ID	Sub_ID	Marks	Teacher
1	15	1	70	Mr. J
2	15	2	75	Mr. A
3	11	1	80	Mr. J

Not in 2-NF, so, break in 2 tables.

Proj_id	E_id	E_name	E_Dept	E_Rate/hr	Total_Hrs
100	123496	Heading	MIS	30	10
100	237991	John	Tech	40	20
100	3211721	Alex	Docum^n	60	30
100	921171	John	MIS	45	40
110	119331	Monty	Tech	30	6
110	32921	Rich	Docum^n	12	7
120	123669	Lopez	Tech	10	9
120	729912	Olwi			

PK → {Proj-ID, E-ID}

Prime Attributes → Proj_ID, E_ID

2NF

Non-Prime Attributes → E_name, E_Dept, E_Rate, Total_Hrs

E_ID	E_Name	E_Dept	E_Rate/Hr

Date / /

Page no.

Proj-ID	E-ID	Total_Hrs

(1)

A	B	C	D
E-ID	E-Name	E-Dept	E-Rate/Hr

To remove deletion Anamoly we have to remove Transitive dependency.

$A^k \rightarrow B$

$A^k \rightarrow C$

$A^k \rightarrow D$

$A^k \rightarrow C$

$C \rightarrow D$ Transitive

E-ID	E-Name	E-~~Rate/Hr~~ Dept

2NF / 3-NF

(1)

E-Dept	E-Rate/Hr

(2)

Date / / Page no.

$A \to B$ $\quad B \to C$

$B \not\to A$

$\Downarrow$ → Transitive Dependency

$A \to C$

$C \not\to A$

Q → Define BCNF.

Ans (5) → $R(A, B, C, D, E)$

$A \to B$

$AB \to C$ $\quad [AB]^+ = \{A, B, C, E, D\}$ ✓

$B \to E$ $\quad [BD]^+ = \{B, D, E, C, A\}$ ✓

$D \to C$ $\quad [ED]^+ = \{E, D, C, A, B\}$ ✓

$E \to A$

always start with max. combination.

$P.A = \{A, B, D, E\}$

$N.P.A = \{C\}$

6 → $R(ABCDEF)$

$A \to C$ $\quad [AF]^+ = \{A, F, C, E, B, D\}$

$B \to D$ $\quad [BF]^+ = \{$

$C \to E$

$D \to E$

$E \to A$

$F \to B$

Q → Define and explain about BCNF?

Date 23/02/2018 Page no.

• Employee R1

Emp-ID	E_Name	Title	D-ID
10	Anurag	sweeper	1
11	Saalim	Developer	2
12	Thulla	Nalla	Null

(child Table)

• Department R2

D-ID	D_Name	D_loc
1	NSS	Delhi
2	Poly	Thatta
3	XYZ	Jammu

(Parent Table)

Referential Integrity

Foreign Key

Given two relation R1 and R2 of the same database, a set of attributes of relation R1 is said to be foreign key of R1 w.r.t R2, if the following two conditions are satisfied:

1. The attributes in foreign key have the same domain as the set of attributes of relation R2, that have been defined as the primary key of R2.

2. The foreign key value of Relation R1 are neither null or must appear as the primary key value of relation R2.

→ The foreign key concept insures referencial integrity between the two relations.

Date / / Page no.

Q → schedule (S-ID, Class_No, S-Name, S-Project, Class-Time, Floor, Instructor)

primary key

S-ID → S-Name
S-ID → S-Project
Class_No → Class_Time
Class_No → Floor
class_No → Instructor

Step 1: Determine the candidate key
step 2: 2NF
step 3: 3 NF

Student (S-ID, S-Name, S-Project)
class (class-No, class-Time, Floor, Instructor)

Database Tuning / SQL

1. Always try to use the normalized form of database design.
2. While using select statement. try to fetch only the information, that is required and avoid using asterisk (*) in the queries.
3. Indices should be created in all the tables where we have frequent search operation. Indices should be avoided in the table where there is less search operation and more of insertion & updation.

Date 23/04/2018 Page no.

(4) While writing a query one should be carefull with the use of equality operator with real no. and datatype value.

(5) In SQL query pattern matching should be used judiciously and carefully.

(6) The SQL queries should be fine tuned. Depending upon the structure of queries and sub queries.

(7) For the queries that are executed on the regular basic, we must try to use procedures. (A procedure is a large group of SQL statements).

(8) Use of logical operator OR should be avoided in a query as much as possible.

(9) When performing a batch transaction try to perform "COMMIT" after a regular interval of record creation.

(10) Database should be defragmented on a regular basis.

Q1 Consider the relation schema R (Sales_Tran_No, Item_No, Quan_sold, price, seller, seller_region). Functional dependency are:

Sales_Tran_NO → Quan_sold

Item_NO → Price

Sales_Tran_NO → seller

Seller → seller_Region

Date 27.02.2018 Page no.

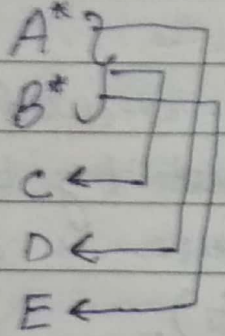
Determine the candidate Key for this relation & transform this into highest normal form.

Sol → CK → { sales_Trans_NO, Item_NO }

A B

PA PA

PA = prime attribute



A*, B* → C
A* → D
B* → E

2NF

sales_Trans_NO	Item_NO	Quan_sold

+

Item_NO	Price

+

Trans_NO	seller	seller_region

3NF convert it into 3NF

3NF

Trans_NO	seller

+

Seller	seller_region

* Non prime attribute depend upon another non-prime attribute → Transitive dependency

Q② → Consider a relation R (property-NO, sell-date, licence-NO, comission, Bonus), the following FD hold for this DB :-

Sell-Date → Bonus
licence-NO → comission

If we assume the attribute property-NO & Licence-No form composite key. what can be highest normal form for this relation.

Q③ → Consider a relation R (A, B, C, D, E, F) for which the functional Dependencies are

A → B
C → DF
A, C → E
D → F

Determine the candidate key for this relation and the highest normal form that can be attain.

Q④ → Consider a relation supplier (supplier-NO, part-NO, Supplier-Name, price, supplier-address) ; assume the following FD →

supplier-NO → supplier-Name
supplier-NO → supplier-address

Date 27/02/2018 Page no.

what are the following various data anomalies present in the given relational schema. Also find the highest normal form that can be attain.

Q⑤→ Find the highest normal form for the relation R(A,B,C,D), if the following FD hold:-

$A,B \rightarrow D$

$A,C \rightarrow B,D$

$B \rightarrow C$

Q⑥→ Consider the relation R(X,Y,W,Z) & FD→

$Y \rightarrow W$

$X,Y \rightarrow Z$

what are the ~~following~~ candidate key for this relation & Find the highest normal form that can be attain.

Date 16/03/2018 Page no.

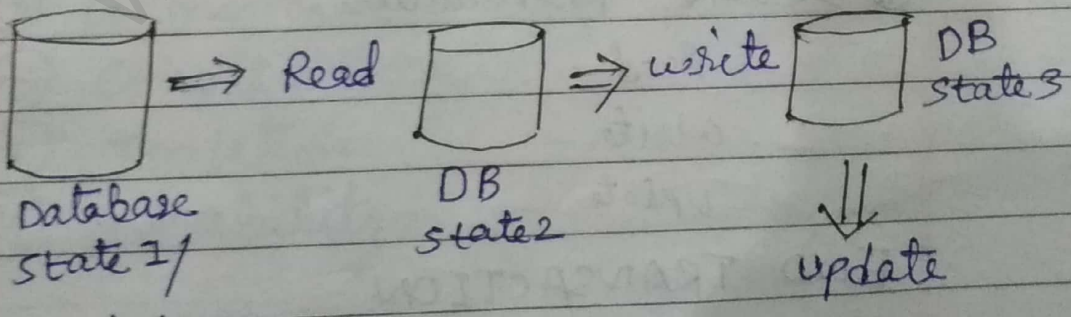
Transaction

- select
- update
- delete
- append

} combination is transaction (Group of statement)

Transaction Processing :- It is a logical unit of work database processing that includes one or more database access operations. It's a series of action that is carried out by single user or application program to access the content of the database

A transaction must be either completed or aborted

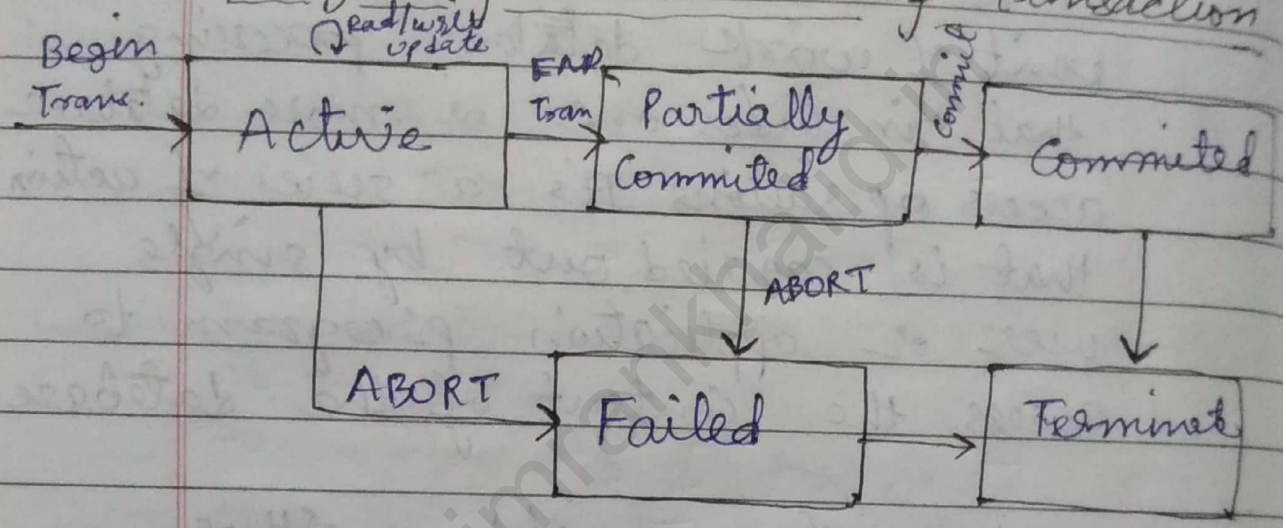


If 1st state is consistance than last is also consistance

Date / / Page no.

Transaction that changes the contents of the DB must alter the DB from one consistance state to another. A consistance DB state is one where all the integrity contraints are satisfied.

- Stages of Execution of transaction



① BEGIN Transaction
Read —
write —
Update —
END TRANSACTION

A transaction goes into an active state immediately after its start the execution. When the transaction ends it moves to the

partially commited state. At this point some recovering protocols must ensure that a system failure will not result in an inability, to record the changes of the transaction. Once this check is successful the transaction is said to the in the commited state. However transaction can go to a faibd state if it is aborted in its active state. The aborted transactions can be restarted automatically or as a completely new transaction.

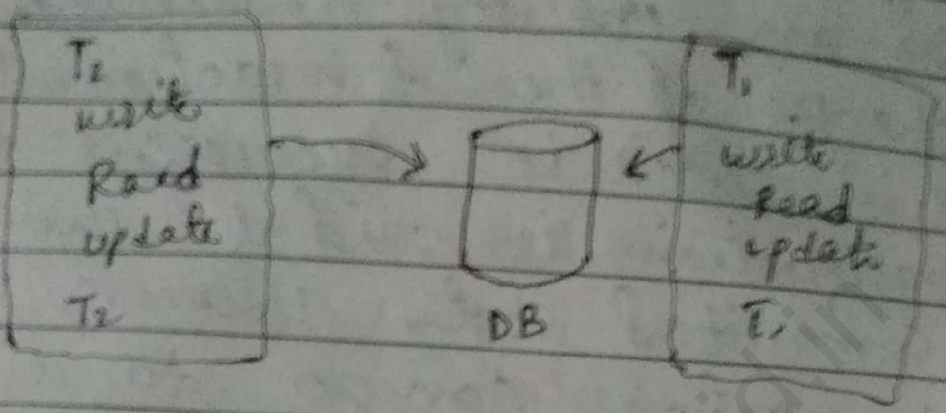
- Properties of Transaction (ACID property)

1. Atomicity
2. Consistancy
3. Isolation
4. Durability

```
BEGIN T1
    Read
    write
    Delete
    update
END T1
```

Atomicity (when all statement executed successfully)

consistancy → state that results after execution of transaction.



(No two same transaction at same DB) → Isolation.

- **Atomicity**

Its a property of transaction require that all operations for transaction must be completed or else the transaction is aborted In other words the transaction is treated as a single individual logical unit of work.

The atomicity of transaction is manage by the transaction recovery sub-system.

- Consistency
Its a property that ensures that a DB moves from one consistance state to another after a completion of transaction. The preservation of Consistancy is the responsibility of programmers to must ensure the integrity constraints of the database.

- Isolation
This property of transaction means that the data that is used by one transaction can't be used by any other transaction untill the first transaction has completed its execution.
The isolation property is enforced by concurrency control subsystem of DBMS.

- Durability

Date / /

Page

transaction

Q → what are the various / control statements? Explain with an example.

Q → what are the problems that arises due to the concurrent execution of various transactions.

↳ lost update
Uncommitted data
Unrepeatable Read

- Schedule

'Locks' shedule the transaction

- Shedule is a sequence of actions or operations that is constructed by merging the actions of several transactions taking into consideration ~~actual~~ action within each transaction.

- A serialisable schedule is a schedule that follows several transactions to execute in some order such that the effects are equivalent to executing them in some serial order.

Date 20/03/2018 Page no.

Rules of serializability

Permitable Actions A1

A pairs of actions A1, A2 are permitable if every execution of A1, A2 followed by A2 has the same result as the execution of A2 followed by A1 on the same data item. These data items are called granules, for the action read & write we need

Read / write is permitable

Read / write is not permitable since the result is different depending upon whether Read is first either Write is first.

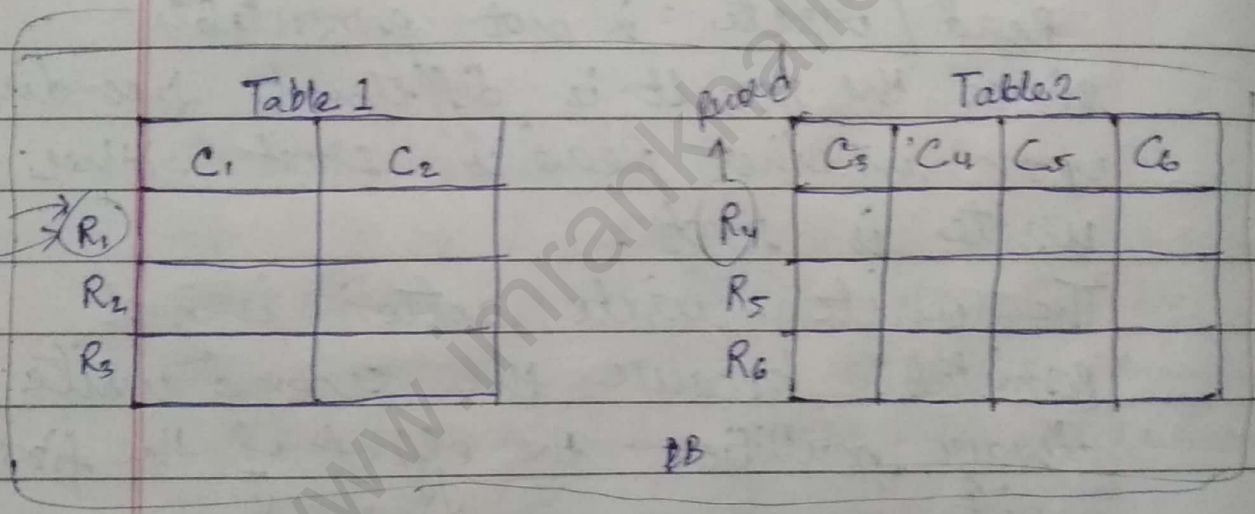
The write - write action is not permitable because the second write always nullifies the effect of the first write.

Locks

A lock is a variable associated with a data item that describes the status of the item w.r.t possible operations that can be applied to it. The locking mechanism prevents access to a DB record by a second transaction until all the operations of the

first transaction has been completed. It is a means of synchronizing the access of DB by concurrent transaction. Blocks are granted and released by lock manager. The principle data structure used by a lock manager is a lock table. The lock table consists of

① Transaction Identifier
② Granule Type
③ Mode of Locking



- **Granularity of Lock**

There are 4 types of locking granularity

1. **DB level**

At the DB level locking the entire DB is locked

2. **Table level locking**

The T.L.L

3. Record level locking

4. Column level locking

2. T.L.L

At this level, the entire table is locked. It is less restricted than the DB level locking and these are not suitable for multiuser DBMS.

3. R.L.L

In this mechanism, a particular row is locked. It is much less restricted the DB level locking. It improves the availability of data but requires a large overhead cost.

4. A/C L.L

It acquires lock on a particular attributes. It allows con-current transaction to access the same row as long as they require the use of different attributes within a row. It is more flexible the D.A but it has a large overhead cost.

Type of Lock Mechanism :-

1. Binary Lock
2. Shared / Exclusive Lock
3. Two phase Locking (2PL)
4. Three phase Locking (3PL)

(2) Shared / Exclusive Lock

If a Transaction Ti has obtained a shared mode lock shared lock (S) then Ti can read but can't write for the data item on which it has aquired the lock.

Exclusive Mode Lock

If a transaction Ti has acquired an exclusive lock on data item Q then it can both read and write on Q.

If we require that every transaction request a lock in an appropriate mode on a data item depending upon the types of operations that it will perform. The transaction makes a request to the concurrency control manager that grants the desired lock to the transaction.

Date 20/03/2018 Page no.

The use of these 2 locks mode allow multiple transactions to read a data item but limits the exclusive access to just 1 transaction at a time.

A transaction requests shared lock on data item Q by executing

lock-S(Q)

lock-X(Q)

.

A transaction can unlock the aquired lock on Q by using instruction unlock(Q).

T_1 : Lock-X(B);	T_2 : Lock-S(A);
read (B);	read (A);
B : B-50;	unlock (A);
write (B);	lock-S(B);
unlock(B);	read (B);
lock-X(A);	unlock (B);
Read (A);	display (A+B);
A: = A+50;	
write (A);	
unlock (A);	

schedule

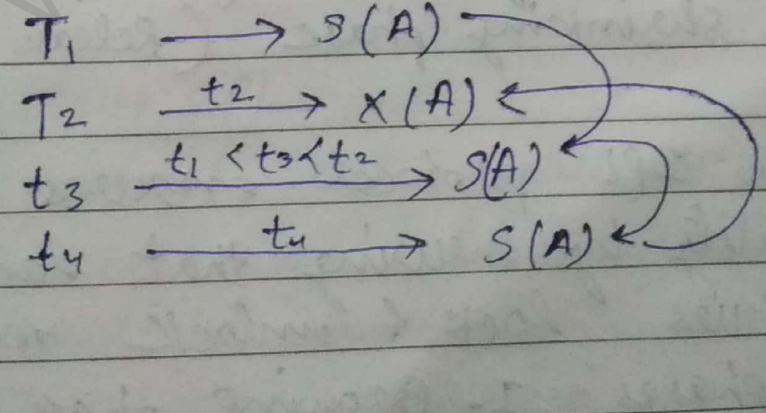
Date / / Page no.

Remove in cascading → delayed unlock

T_3:	T_4:
lock-X(B)	lock-X(A);
read(B);	read(B);
write(B);	lock-X(B);
lock-X(A);	read(A);
read(B);	display(A+B);
A=A+50;	unlock(A)
write(A);	unlock(B);
unlock(A)	
unlock(B);	

Date 23/03/18 Page no.

To remove inconsistency in transaction we can delay the unlocking of data items by a transaction in the end of the transaction. This is called the delayed unlock. But this again may the give rise to a situation where a given schedule reaches a states of deadlock, but deadlock deal unnecessary evil for transaction. A deadlock state is preffered over the inconsistency state of transaction Because deadlock can be resolved but inconsistency of data might give rise to further problem. To avoid these situation the system follows a set of rules called locking protocols.



starvation is a situation, lock can be granted by following method.
In transaction T_i requests a lock on data item Q in a particular mode M, then M can be shared or exclusive, a concurrency control manger grants a lock provided
1. There is no other transaction holding a lock on Q in a mode that conflicts with M.
2. There is no other transaction that is waiting for a lock on Q that has made its lock request before T_i.

2-phase locking Protocols (2PL)

1. Growing phase (acquire lock)
2. Shrinking phase (Release lock)

The 2PL protocol ensures the serializability by ensuring that each transaction issues lock & unlock request into 2 phases 1. Growing phase
in this phase a transaction may obtains lock but can't release any lock.
2. shrinking phase, in this

phase transaction release lock but can't acquire lock.
Initially a transaction is an growing phase, once a transaction declears a lock it enters the shrinking phase. The point where the transaction has obtained final lock is called the lock point of transaction.

1. Strict 2PL
The protocal that ensures ~~acquires 2PL~~ that all the locks in 2 Phases in addition it must be must ensure that all the exclusive that is acquired by the transaction must be held till the transaction commits.

2. Rigorous 2PL
The protocal that ensures that all the locks ux held in 2 phases it must be make sure that all the locks exclusive as well as shared that are acquired by the transaction must be held till the transaction commits.

Date / /

Page no.

Lock Conversion / upgradation

T_1	T_2
read (a_1);	read (a_1);
read (a_2);	read (a_2);
:	display (a_1+a_2)
:	
:	
read (a_8);	
write (a_1);	

Q1→ How a lock manager processes the two locking and the unlocking by a transaction? (5-6 steps)

Q2→ Explain the working of lock control manager.

Date / / Page no.

Schedule - S

lock - X(a_1)	lock-S(a_1)
read(a_1)	read-S(a_1)
:	
:	
lock - S(a_2)	lock-S(a_2)
read(a_2)	read(a_2)
upgrade(a_1)	
downgrade(a_1)	

Deadlock

1. Acquire all the locks by transaction
2. Pre emption and rollback.
3. Ordering of data item
4. Ordering of data + 2PL

- Time stamp of transaction
- lock-time-out

Date 27/03/2018 Page no.

Indexing

EmpID	E-name	Dept	salary
10101	ABC	Finance	60000
12121	BCD	Hiring	70000
15151	CBC	CS	90000
22222	DEF		
32343	EFG		
33456	GHI		
45565			
58583			
76543			
76543			
76766			
83821			

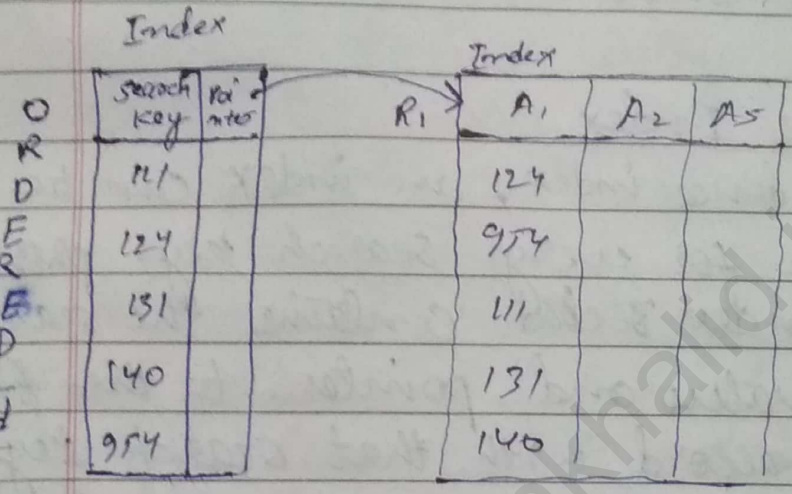
1. clustured | Primary Index
2. Un clustured
3. Ordered → Dence, Spare
4. Multilevel
5. Hash Index

Types of index to be selected depends upon the following factors:

1. Access type
2. Access Time
3. Insertion Time ← where to insert update Index

4. Deletion Time — Finding the point where data is delete. / After deletion update Index
5. space overhead

- Clustured or Primary Index



Clustured Indexes

A clustering inder is an index to search key also defines the sequential order of the table. If the choosen search is the primary key of the table then such as clustering index is called primary index.

The search key of clustering index is often the primary key although that is not a neccesity.

- **Ordered Index**

An ordered index stores the values of a search keys in sorted order and associates with each search key value, the records from the main table.

- **Dence Index**

In a dence index, an index can be appear for every search key present the index record contains the search key value and pointer to the first data record with that search key value. The rest of the records in the same search key would be stored sequentially one after the other.

- **Sparce Index**

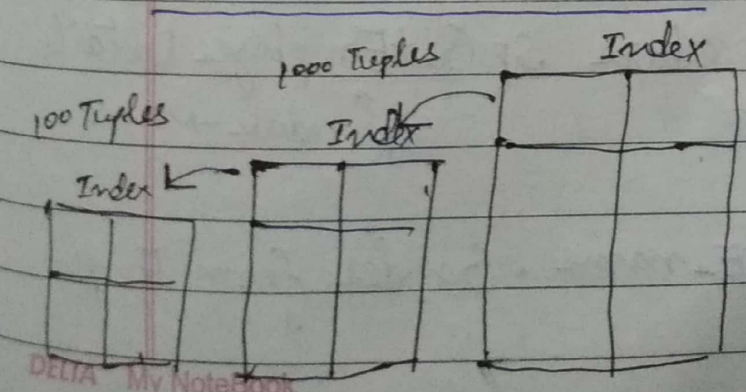
Emp_Id	E_name
10101	
12421	
15131	
22222	
32343	
33456	
45565	
58583	
76766	
83821	

Spare key	
10101	
32343	
58583	
83821	

In a spare index, a index entry appears for only some of the search key values. Each index entry contains a search key value and a pointer to the first data record with that search key value. To locate a record we find the index entry with a largest search key value that is less than or equal to value of the search key we are looking for. We start at the record pointed by that search key value and then follow the pointers until we find the desired record.

It is faster to locate a record with the dence index rather than a sparce index. But this sparce index required less space and they have less space overhead by performing insertion & deletion operation.

- Multi-level indexes



Date / / Page no.

create index <index_name>;

or

<table_name> (attribute_list);

No. of attributes

03/04/21

Procedure

Emp

E_ID	E_Name	D_ID	Gender	Contact
1	ABC	1	Male	9582
2	XYZ	1	Female	9583
3	CDE	2	Female	9581
4	MNO	1	Male	9572
5	PQR	2	Male	9570

- Procedure is a group of SQL statements that need to be executed multiple times through in an application. When a user has to write same query over & over again, we can save the query as a procedure that can be called just by its name.

- Creating a procedure

```
CREATE procedure Sp_GetEmployeeDetail
AS                          ↑ procedure-name
BEGIN
    select E_name, Gender from Emp
END
```

DELTA My NoteBook

Date / /

Page no

Procedure with parameter

```
CREATE PROCEDURE spEmpData_with parameter
@Gender varchar(20)
@E_ID int
As
BEGIN
  select E_ID, Gender, E_Name from Emp
  Gender = @Gender and
    E_ID = @ID
END
```

- Execution

(1.) EXEC spEmpData with parameter
'Male', '21'

or

```
EXEC spEmpData with parameter
  @D_ID = 2
  @Gender = 'Male'
```

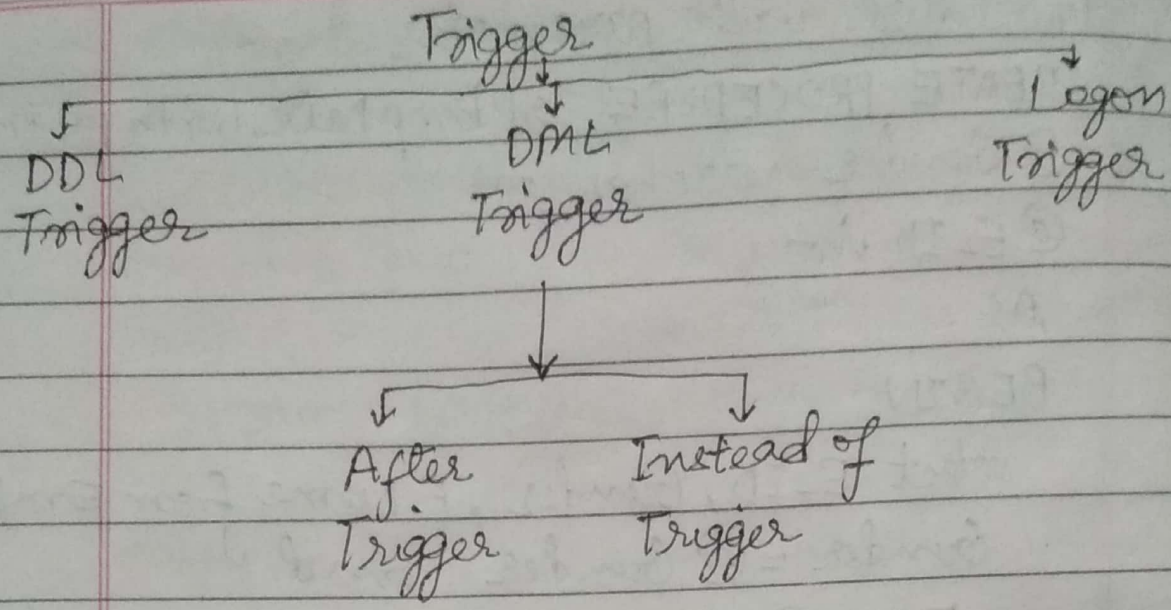
- To see the defination of the procedure →
sp_helptext <name of procedure>

- To Encrypt -

```
ALTER PROCEDURE spEmpData with parameter
@Gender varchar(25)
with ENCRYPTION
As
BEGIN select
```

Date / /

Page no.




```
CREATE Trigger tr_InsertData
ON emp
FOR INSERT
 As
BEGIN
     select * from ---
END
```

[logical table always form]

- (1) → How to decrypt?
- (2) → Advantages of procedure & Disadvantages.
- (3) → Can we use multiple select statement in procedure.

Date 10 / 4 / 2018 Page no.

Database snapshot

Database snapshot is a read only static view of a SQL server database. It is transaction consistent with a source databases. It always resides on the same server as the source database.

Multiple snapshots of a particular database is possible as feasable, each snapshot of a DB persist unless it is specifically drop by the database administrate / user.

- Creating

* → DB_Name = CustDetails

```
CREATE DATABASE CustDetails-DBSS
ON
(Name = "CustDetails", FileName = "C:\use\DBSS).
AS SNAPSHOT OF CustDetails;
```

- Restoring

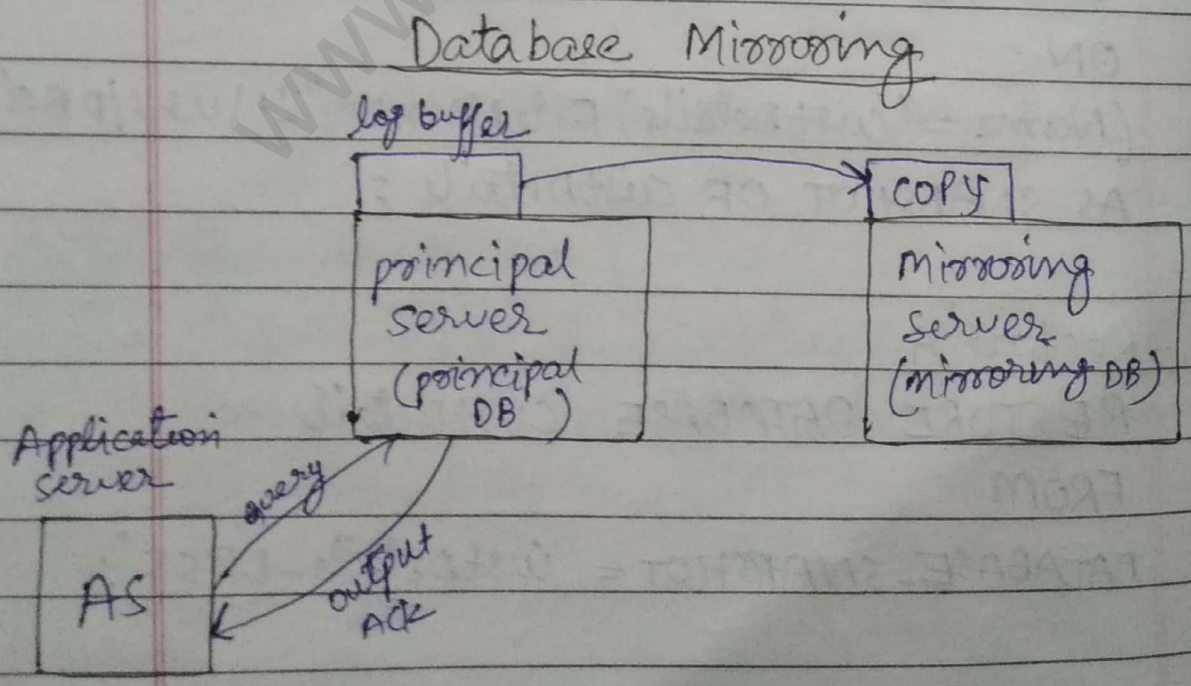
```
RESTORE DATABASE CustDetails
FROM
DATABASE_SNAPSHOT = "CustDetails-DBSS";
```

Page no.

→ What are the drawback of creating / limitations of creating a snapshot.

- DB snapshots can be used to provide a backup to a existing Database such that in case of any circumstance change to the main database the original data can be restored using its snapshot.

- If we want to perform certain manipulation on a DB that need to permanent we can perform those manipulation on the snapshot to get the desired outcome this way the original DB doesn't get affected.



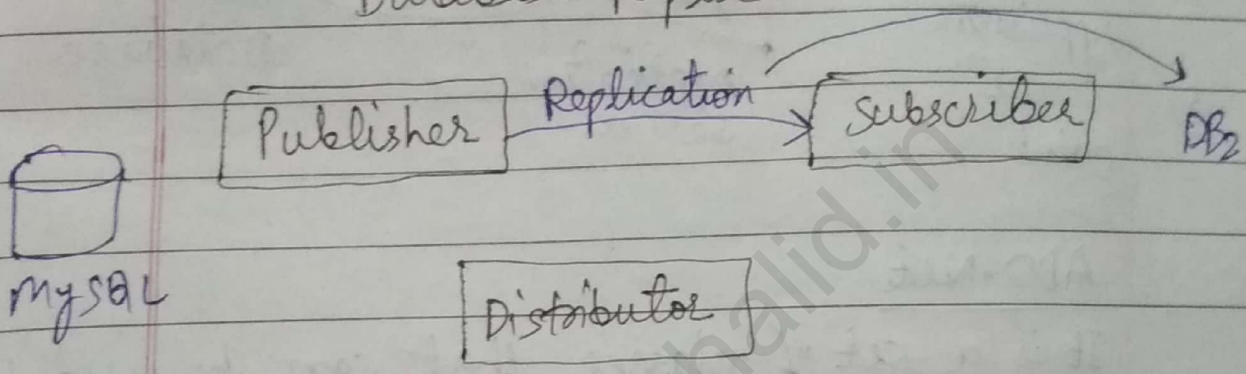
Date / /

Page no.

Modes of operations

1. High scalability (Dynamic)
2. High safety
3. High performance

Database Replication

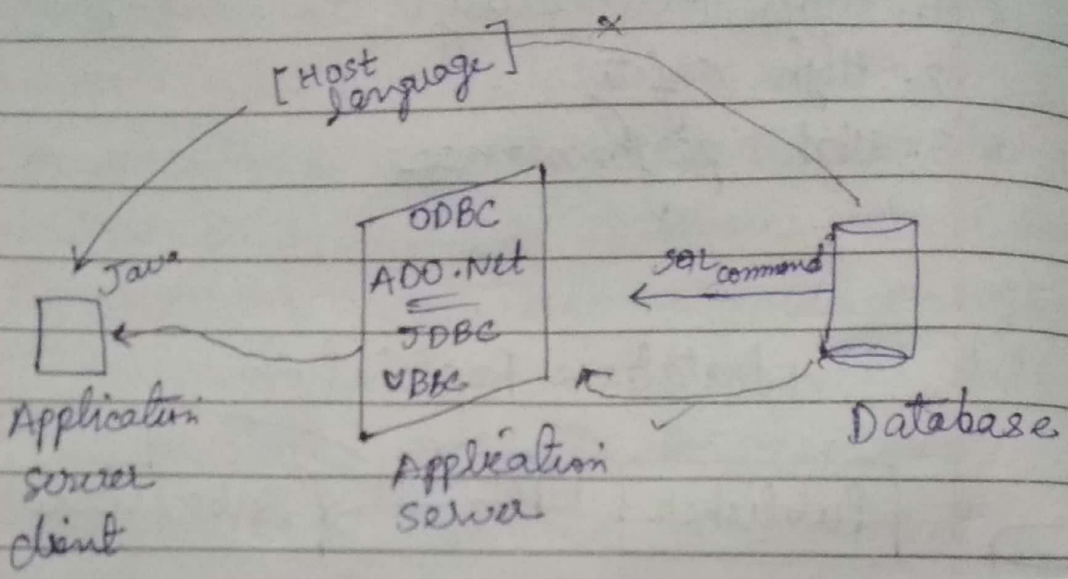


Type

1. Snapshot Replication
2. Transactional "
3. Merge "

24/04/2018 Page no.

3-Tier



ADO.Net

It is a set of classes that can be used to interact with the data sources like any database or xml file. This data from the DB is then used by the corresponding application. ADO stands for Activic X Data Objects.
The .Net applications that use ADO.Net connects to the DB, execute the commands and retrieve the data to be used by .Net applications.

4-Stages

1. Connecting to a database
2. Configuring the commands
3. Executing the Commands
4. Retriving the data and displaying over application.

for VB

Date Using System.Data.SqlClient; Page no.

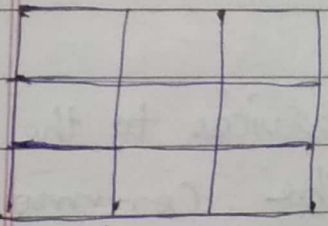
```
SqlConnection con = new SqlConnection
        ("data source = .", database = sample,
         Integrated security = SSPI")
SqlCommand cmd = new SqlCommand ("select
        * from sample", con);
```

Sample

S-ID	S-Name	Branch
01	ABC	CSE
02	XYZ	MEC
03	MNO	ECE

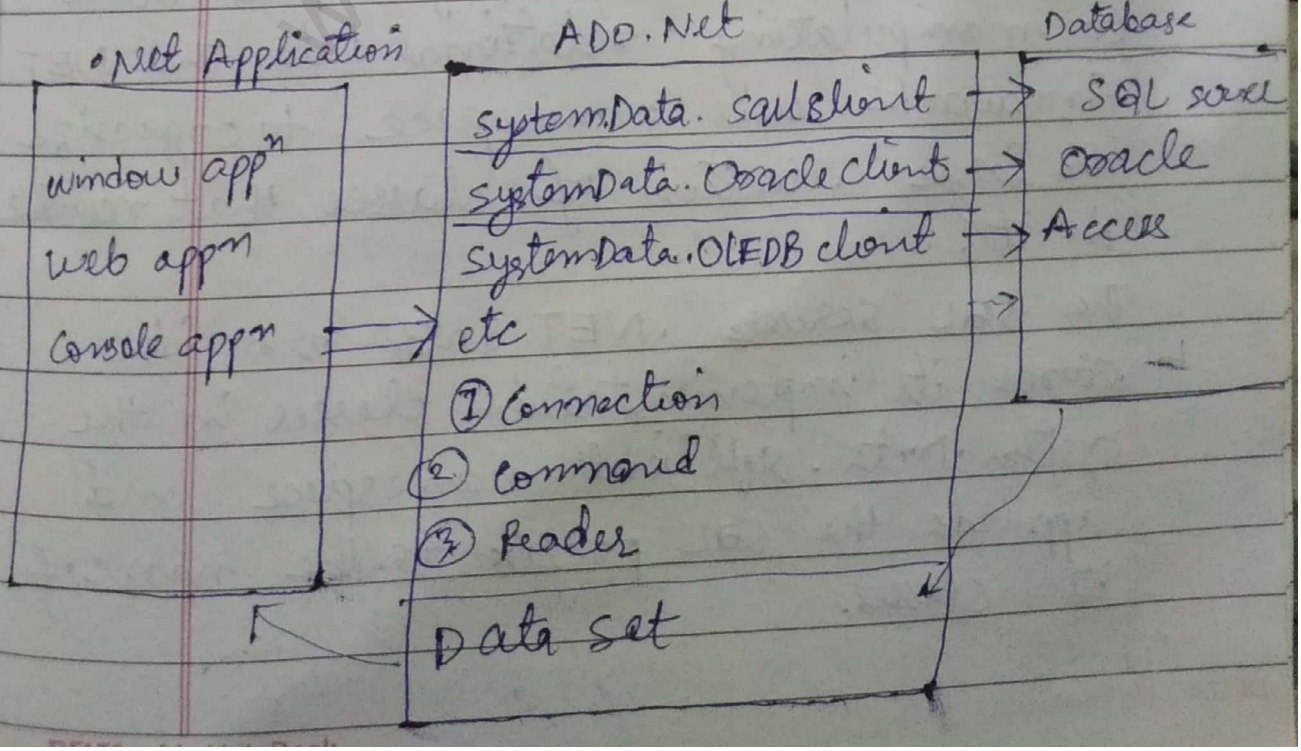
SQL server

```
Con.Open ();
SqlDataReader rdr = cmd.ExecuteReader ();
Gridview.DataSource = rdr;
Gridview.DataBind ();
Con.close ();
```



.Net App.

SSPI → security support provided Interface



Date 27/04/2018 Page no.

Universal Data Access

Universal Data Access is microsoft model or framework for a single uniform application program interface to different databases.

These different databases can be relational as well as non-relational.

A universal data access consist of a high level interface, data objects and some lower level services like Ole DB

ADO.NET

The ADO.NET provide services to the client code running under the common language runtime. The client interacts with the data sources through .NET data providers. The SQL server .Net data provider provides a high speed route to manipulating relational data. The .NET framework uses namespace to organize the large collection of classes that reside in it.

The SQL server .NET data provider stores its implementation classes in the system.Data.sqlClient namespace and appends the SQL prefix to the name of its classes.

The OleDB .NET data provider resides in the system.Data.OleDBClient namespace.

ODBC (Open DataBase Connectivity)

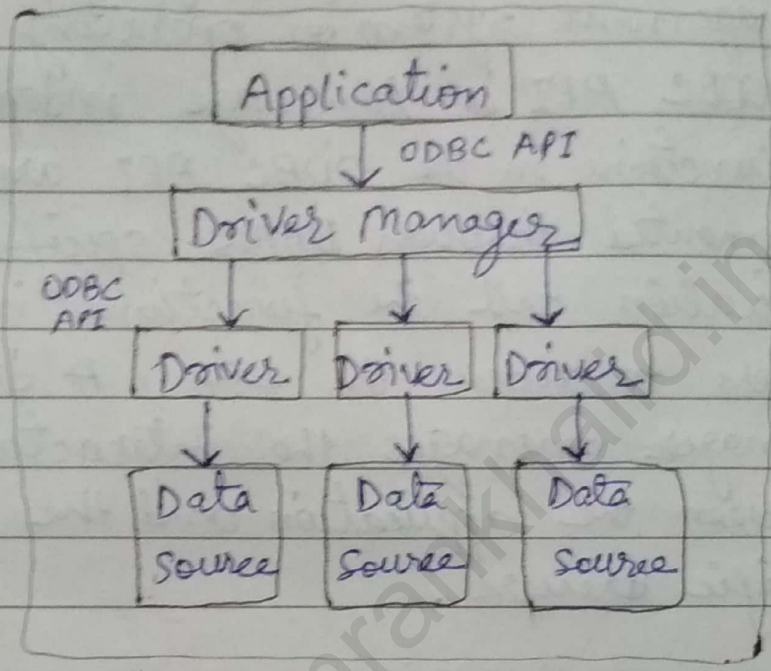


Fig. Component Diagram for ODBC

There are 4 different components for ODBC. ODBC is an open standard application programming interface that allows application programming to access any database.

How ODBC Works

The ODBC consist of 4 components that allow programs use sql requests that access databases without knowing the proprietary interfaces to the databases.

Date 27/04/2018 Page no.

ODBC handles the SQL request and converts it into a request that can be understood by the database. ODBC is a specification for a database API, this API is independent of any one database management system or operating system. The ODBC API is language independent. The functions in the ODBC API are implemented through DBMS specific drivers. Application call the functions in these drivers to access the data. A driver manager manages the interaction between the application and the database specific drivers.