

INSB BCA Collage IDAR.

SUBJECT:- Advance Database Management System

Subject Code:- BCA-302

Present By: Asst-Prof. Smit Trivedi

BASIC ADBMS

1.What is Data?

- Data is a collection of row facts and figure.
- It can be no alphabets ,symbols etc.

2.What is Database?

- Database is a collection of meaningful information that is organized so it can easily access, managed and update.

3.What is DBMS?

- DBMS is a computer software prewriting programs that is design managing all the database.
- Ex: Ms-Access , FoxPro, DB2 , Oracle

4.What is Table?

→ Table is a database object that hold (Store) user data.

5.Row And Record

→ A record is also collect a row of data is individual entry that exist in a table.

6.Column and attribute

→ A column is vertical entry that contains all information with specific field in a table .

7.What is RDBMS

→RDBMS stands Relational Database Management System.

→ RDBMS is a database management system based on relational model design by code.

→ Data is stored in form of row and column .

→Ex:- oracle,SQL server .

Difference between DBMS and RDBMS

DBMS

1.Relation between two tables or File are maintain program actually.

2.DBMS is support single user only.

3.It does not support client server architecture.

4.DBMS there is no security of data

5.In DBMS rows and column are independent.

RDBMS

1.Relation between two tables or file Can be specific the time of table creation.

2.RDBMS is support multiple user.

3.it support client server architecture.

4.RDBMS there are multiple level of security.

5.In RDBMS rows and column are dependent.

UNIT-1

Transaction Management

ACID Properties

❖ There are **FOUR** properties of as below:

1.A:- Atomicity

2.C:- consistency

3.I :- Isolation

4. D:- Durability

1. A:- Atomicity

- Transaction are automatic call or nothing.
- This properties of transaction required all operation should be Completed if not the transaction.

2. C:- Consistency

- Database consistence from one consistence state to another.
- That any change to value in an intense are consistence with change to Other value in a same instance .

3. I:- Isolation

- Transaction are isolated from one other .
- When many transaction running concurrently any update in one transaction should Not effect to other.

4. D :- Durability

- Once transaction is commit its update made in the database even If system crash.

Concurrency

- ✓ The terms concurrency returns to the fact that DBMS Typically allowed many transaction to access the same database At the same time.

Three major concurrency problem.

1.The Last Update Problem

2.Uncommitted Dependency Problem

3.Inconsistant Analysis Problem

1. Last Update Problem

Transaction A	Time	Transition B
	↓	
Retrieve t	t1	
	↓	
	t2	Retrieve t
	↓	
Update t	t3	
	↓	
	t4	Update t

2. Uncommitted Dependency Problem

Transaction A	Time	Transition B
	↓	
	t1	Update t
	↓	
Retrieve t	t2	
	↓	
Update t	t3	
	↓	
	t4	Rollback

3. Inconsistent Analysis Problem

Acc1=40	Acc2=50	Acc3=30
Transaction A	Time	Transaction B
	↓	
RetriveAcc1	t1	
Sum=40	↓	
Retrieve acc2 sum=90	t2	
	↓	
	t3	Retrieve acc3
	↓	
	t4	Update acc3=30→20
	↓	
	t5	Retrieve acc1
	↓	
	t6	Update acc1 40→50
	↓	
	t7	commit

Retrieve acc3

sum=110 not 120

Two phase commit

→ When transaction has completed its database processing successfully .

On receiving that commit request The coordinator goes through the following Two phase process :

1. Prepare
2. COMMIT.

1.Prepare

→ first it instructs all resource manages to get ready to “go either way” On the transaction each participant in the process.

→ assuming the forced write is successful the resource manager How replies “OK” to the coordinator otherwise it replies “NOT OK”.

2.COMMIT

→ when the coordinator has received replies from all participants , It forces a record to its on physical log recording its basis on regarding The transaction.

→ if all replies were “OK” that decision is “COMMIT”, if any reply was “NOT OK” the decision is rollback.

Locking

Lock: **Lock** is a mechanism to ensure data consistency. **SQL Server locks** objects when the transaction starts. When the transaction is completed, **SQL Server releases the locked** object. ...

Exclusive (X) Locks: When this **lock** type occurs, it occurs to prevent other transactions to modify or access a **locked** object.

There are **two** types of lock:

1. Exclusive lock or X-lock or write lock
2. Shared lock or S-lock or read lock

1.Exclusive lock or X-lock or write lock

→ If transition A hold x-lock on time t then request from transition B for a lock of ether type on time t can not be immediately granted.

2.Shared lock or S-lock or read lock

The transaction A hold shared lock on time t then

- i) A request from transaction B for an x-lock on time t can not be immediately Granted.
- ii) A request from transaction B for an s-lock on time t can be immediately Granted.

Deadlock

“ deadlock is a situation in which two or more transaction are in a Simultaneous wait state, each of them waiting for one of the other To realize a lock before it can proceed.”

→ We have also seen that locking can introduced problem of deadlock.

Transaction A	Time	Transaction B
Lock r1 exclusive	t1	
	t2	Lock r2 exclusive
Lock r2 request	t3	
Wait		
Wait	t4	Lock r1 request
Wait		Wait
wait		wait

Recovery

→ **Definition:** Database recovery is the process of restoring the database to a correct (consistent) state in the event of a failure. In other words, it is the process of restoring the database to the most recent consistent state that existed shortly before the time of system failure.

Types of Recovery:

1. Transaction recovery
2. System Recovery

1.Transaction Recovery:

→ Transaction is a logical unit of work.

→ The transaction begins (Start) with the execution of the beginning transaction operation and ends with the execution of the commit or rollback operation.

→ commit:

→ Commit is a successful end of transaction .

→ it tells the transaction manager that logical unit of work has been successfully completed.

→ Rollback:

→ the rollback is an unsuccessful end of transaction.

→ it tells the transaction manager something has gone wrong.

2. System Recovery

→ The system must be prepared to recover not only from purely local failure but also from global failure such as power outage.

→ Types of global failure:

1> System failure (Soft crash)

2> Media failure (Hard crash)

1> System Failure (soft crash)

→ A transaction has to abort when it fails to execute or when it reaches a point from where it can't go any further. ... **System errors** – Where the **database system** itself terminates an active transaction because the **DBMS** is not able to execute it, or it has to stop because of some **system** condition.

2> Media failure (Hard crash)

→ A common example of **media failure** is a disk head crash, which causes the loss of all files on a disk drive. All files associated with a database are vulnerable to a disk crash, including data files, online redo log files, and control files. The appropriate recovery from a **media failure** depends on the files affected.

Backward Recovery

- Backward recovery restores a journeyed database to a prior state.
- **Backward** processing starts by rolling back updates to a checkpoint (specified by -SINCE or -AFTER) prior to the desired state and replaying database updates forward till the desired state

forward recovery

- **forward recovery**, used together with after-imaging and backup, lets you **recover** from media failures. When a database disk fails, you can restore the most recent backup, then use roll-**forward recovery** to restore the database to the condition it was in before you lost the disk.