FRIDAY 263-102 • WK 38

20 SEPTEMBER 2019

S M T W T F S S M T W T F S
AUGUST

DBMS

Advantages:

1. Better Data Transfer: Manged data
2. Better Data Security
3. Better Data Integration
4. Minimized Data Inconsistency
   ↓
   Diff. versions of same data appear in diff. places
5. Faster data Access
6. Better Decision Making
7. Inc. end-user productivity
8. Simple

Keys

1. **Candidate**: Minimal set of attributes which can uniquely identify a tuple.
2. **Super**: Set of attributes which can uniquely identify a tuple
   * Adding more attributes to Candidate key
   * Candidate key is Super key but vice versa is not true.

2019

3. Primary Key : Unique identifier (only one)

4. Alternate : Candidate key other than primary key.

5. foreign : Primary key of the other table.

Data Redundancy → Having same data but at diff. places.

Normalizing → Minimizing Redundancy.

Normal Forms

1. 1NF – If it does not contain any composite or multi valued attribute.
   * Every attribute is singly valued attribute

2. 2NF – Must be in INF and no partial dependency
   ↓
   If proper subset of candidate key determines non-prime attribute (attributes which are not part of any candidate key)

* K^th highest Salary: SELECT Salary from Table
ORDER by Salary DESC
LIMIT k-1, 1

3. **3NF** - It is 2NF & no transitive dependency.

X → Y is in 3NF if
(i) X is a super key, OR
(ii) Y is a prime attribute (part of some candidate key)

Transitive dependency - If A→B & B→C, then A→C

* Non prime should not be derived from non prime attribute.

4. **BCNF** - It is in 3NF and LHS is super key. (i.e. non prime attribute can not derive prime att.)

2, A (NP) ↛ B (P)

SQL Commands

DDL	DML	DCL	TCL
(Data Definition)	(Data Manipulation)	(Data Control)	(Transaction Control)
To define db structure	Managing Data	Authorization	Manage changes by DML
* CREATE	* SELECT	* GRANT	* COMMIT
* ALTER	* INSERT	* REVOKE	* ROLLBACK
* DROP	* UPDATE		* SAVEPOINT
* RENAME	* DELETE		
	* MERGE		

Aggregate Functions

(i) Count
(ii) Sum
(iii) Avg
(iv) Min
(v) Max

Relational Algebra → Procedural Query Language

Takes Relation as i/p
& gives Relation as o/p

Operators

(i) Projection (Π) : Project reqd. column data from a rel^n.
* By default, it removes duplicates

(ii) Selection (σ) : Select reqd. tuples

(iii) Union ($\cup$) : Relation must have same set of attributes

(iv) Set Diff. ($-$)

(v) Rename (ρ) : $\rho(a/b)R$
↳ Rename 'b' by 'a'

2019

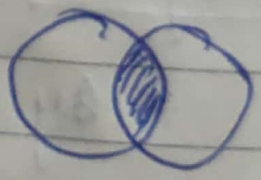
(vi) Cross Product (×) : Results all attributes of A followed by each attribute of B.

(vii) Natural Join (⋈) : Binary Operator
↳ Depends on equal common attribute

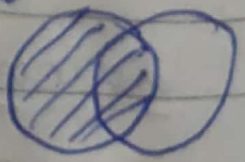
(viii) Conditional Join : Specify any condition.

Types Of JOINS in SQL

(i) INNER JOIN : Returns records having matching values in both tables



(ii) LEFT : All records from left Table & matched from right



(iii) RIGHT :

(iv) FULL :

(v) SELF JOIN : Each row is joined with itself

Syntax: SELECT a.col1, b.col2
FROM table_name a, table_name b
WHERE condition

Views → Virtual table where data is stored temporarily.
Provide Encapsulation

Cursor → Pointer to control the context area & also go through the records in the db.
↳ used in looping over all indexes.

Trigger → Prog. which gets executed in response to occurrence of some events.
↳ Special Type of SP

Transactions → Group a set of tasks into a single execution unit.

(i) SET
(ii) COMMIT
(iii) ROLLBACK
(iv) SAVEPOINT - To roll the transaction back to a certain point
(v) RELEASE SAVEPOINT - To remove sp

2019

ACID Properties

Atomicity - Entire trans. takes place at once or doesn't happen at all

Consistency - Data must be consistent before & after the transaction

Isolation - Multiple trans. occur independently without interference.

Durability - Changes of successful trans. occurs even if the system failure occurs.

Concurrency problems in DBMS Transactions

1. **Dirty Read | Temporary Update**
   ↳ when a trans. updates a an item & fails but the updated item is used by another trans.

Isolation level (IL) - Read Uncommitted

2. Incorrect Summary : Aggregate func. may calculate some values before the values have been updated & others after update.

3. Lost Update Prob : Update done to a data item by a trans. is lost as it is overwritten by the update done by other trans.

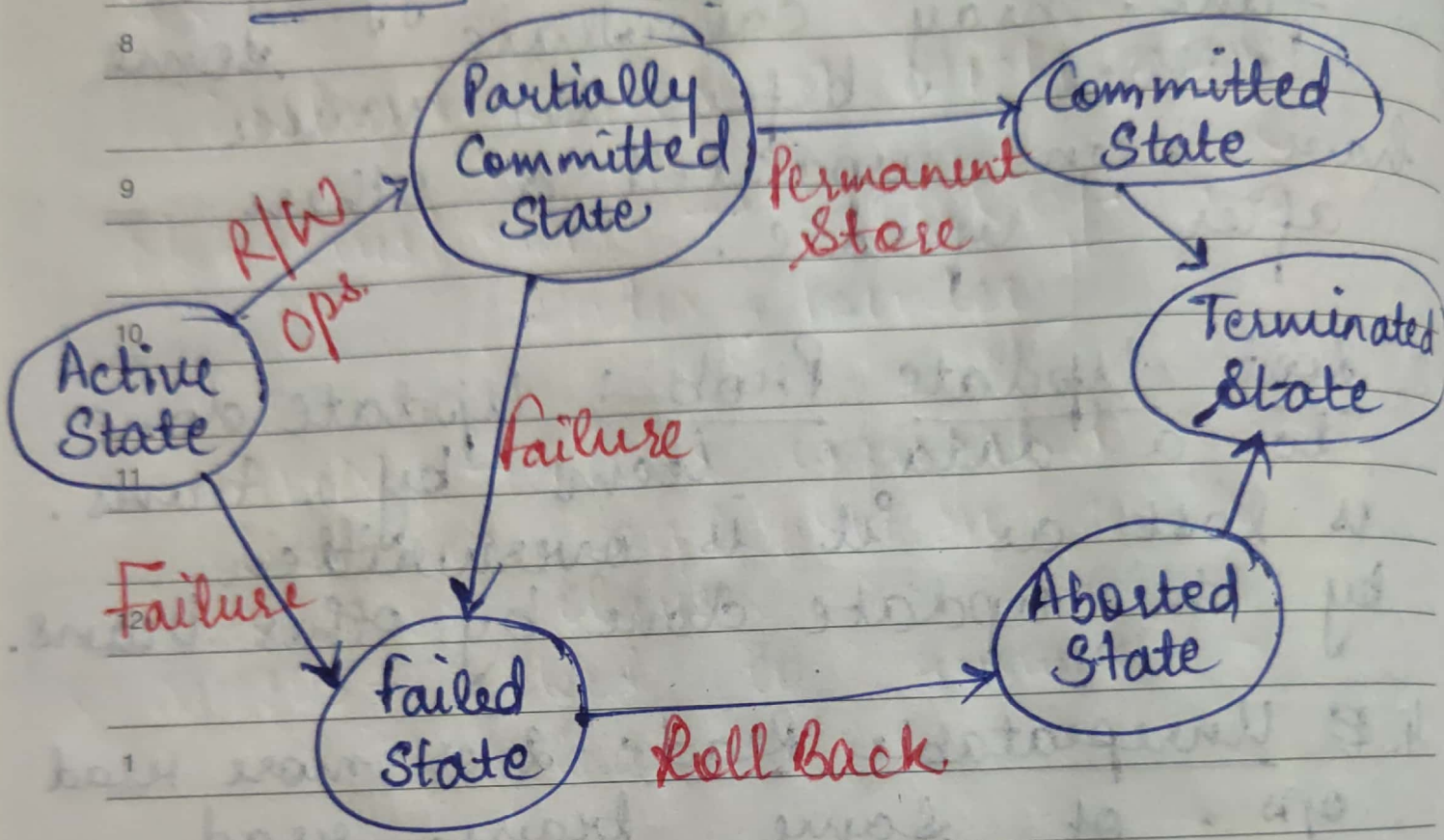
4. Unrepeatable Read : 2 or more read ops. of same trans. read diff. value of same variable.

5. Phantom Read : A trans. reads a variable once but then other trans. deletes it so reading same variable again says it does not exist.

Highest Isolation Level → Serializable

30

Transaction States in DBMS



Types Of Serializable Schedule

1. Conflict Serializable :- if it can be transformed into serial schedule by swapping non conflicting ops.

Graph has no cycle.

Easy to Achieve

- Belong to diff. trans.
- operate on same data item
- Atleast one of them is write opr.

(2.) View Serializable :- if order of operations is same in both schedules.

* Cascading Rollback - A single trans. failure leads to a series of trans. rollbacks.

* Cascadeless Schedule - Allows only committed read ops. However, uncommitted write ops.

*4NF : It is in BCNF & should not have Multi-valued dependency.
↳ More than one value is dependent on same value.
eg. Person →→ Mobile, Person →→ food

*5NF: It is in 4NF & not have property of Join Dependency.
↳ Lossless Join Decomposition
↳ Joining must be lossless
↳ (Project Join NF)
2nd condition: Cannot be further lossless decomposed.

02 OCTOBER 2019 SEPTEMBER

~~REST~~ ~~API~~ → Speedy Retrieval

Clustered Index → Dictionary

Non-Clustered → Index of a book
[Index, Pointer]

* HAVING - Filter aggregated records
Used in combination with the GROUP BY clause.

* MINUS - Used to remove duplicates

* DELETE - Delete some rows
↳ Doesn't free the space

* TRUNCATE - Delete all rows
↳ Frees the space

* DROP - Delete table from db

Stored Procedure - provides security for users who can't access data can access via SP.

Diddy
→ Occupies more memory
→ Can't be used anywhere except db