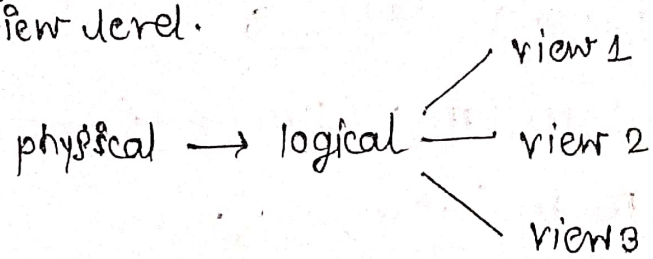
DBMS

file system
- no concurrency
- redundancy
- inconsistency
- harder access data

} DBMS as soln.

abstraction: hiding irrelevant details from users.

① ~~copy~~ physical level (actual store in disk).
② logical level (define schema)
③ view level.



schema - logical structure of db (blueprint).

instance - actual data stored in db (CRUD on instance).

data ~~int~~ models (tools)
- relational model
- ER model
- object based model
- network and hierarchical model.

DML - data manipulation language.
- access and manipulate
- CRUD operations.

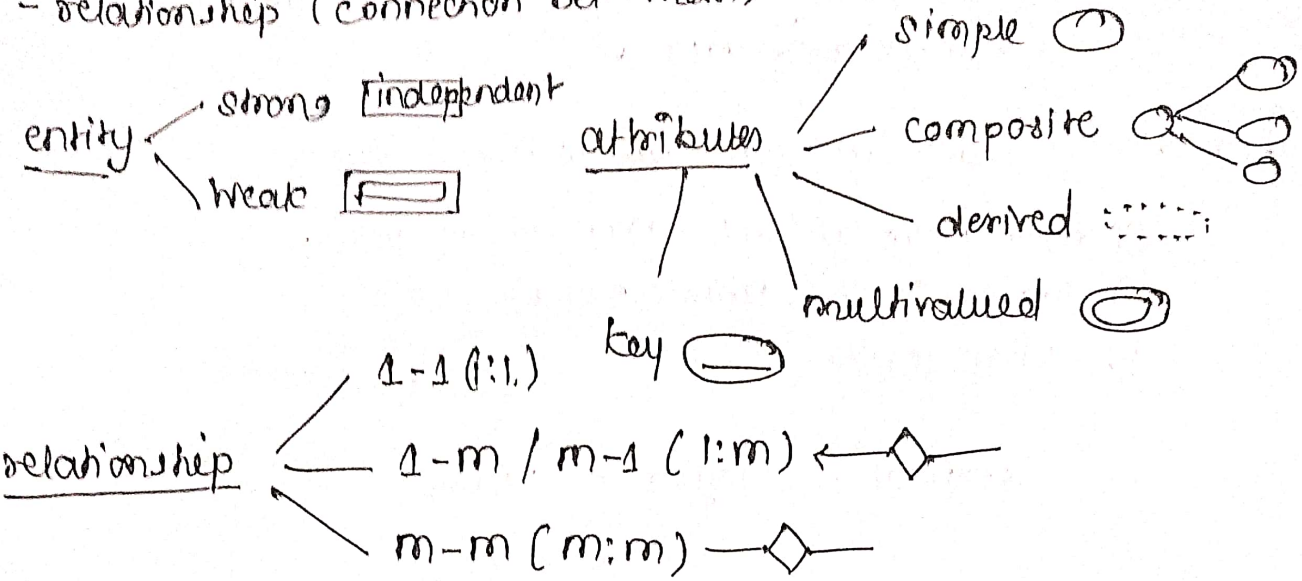
~~disk sto~~

DDL - data definition language
- manage schema and structure.
- create, alter, drop and truncate.

Architecture
1. tier architecture (db and UI in same system)
2. tier architecture client → server + db
3. tier architecture client → api (UI) → server + db.

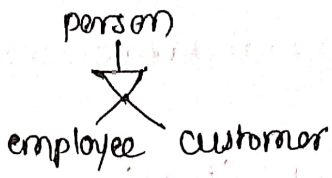
DB design ER model

- entities (objects)
- attributes (properties).
- relationships (connection betn them).



EER

① specialization - subgrouping
② generalization - reverse.
③ aggregation - relationships among relationships
- eliminate redundancy.



✱ Relational DB Design

- represent data in tables (relations).
- pitfalls - redundant data
  - insertion anomalies
  - deletion anomalies (unintended loss of data)
  - updation anomalies (single update req multi row update)

Normalisation : decompose large relations into small, to reduce redundancy

1NF - atomic value (unique)
- no multi valued attributes.

name	branch	rollno	language
cheran	AIML	40	python, C
Bharesh	Electrical	4Φ	Rust, C++

⇒

name	branch	rollno	lang.
cheran	AIML	40	python
cheran	AIML	40	C
bharesh	elec	4Φ	Rust
bharesh	elec	4Φ	C++

2NF - should 1NF
- no partial dependencies (PK → ⊗ duplicate value).

1NF ⇒

name	rollno (PK)	branch
cheran	40	AIML
bharesh	99	electrical

(rollno).

rollno (FK)	language
40	python
90	C
40	RUST
40	C++

3NF - should 2NF
- no transitive dependency.
- non prime attribute should depend on primary key (PK).

studentid	name	deptID	dept name
101	cheran	1	AIML
102	bharesh	2	EE

transitive dependency
studentid ← deptID ← dept name
depend.

studentid	name	deptId
101	cheran	1
102	bharesh	2

deptid	dept name
1	AIML
2	EE

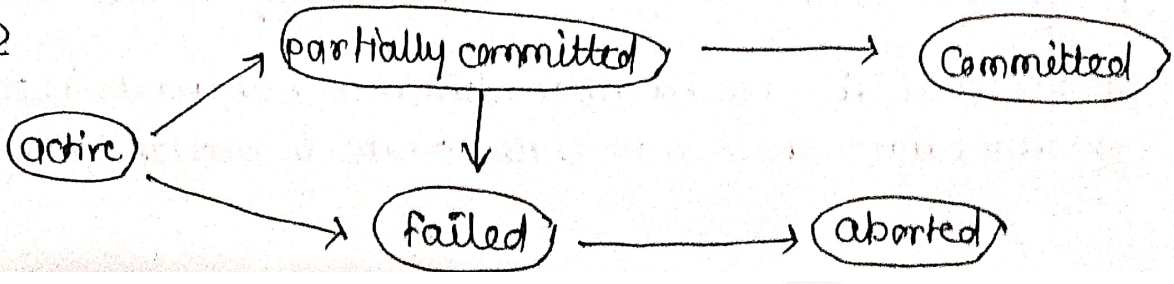
BCNF - should 3NF
- dependency A → B then A should super key.

(*) Transactions

- transaction is unit of work performed in db.

ACID - atomicity (all reflect or none)
- consistency (after execution consistency in db)
- isolation (seperate execution)
- durability (backup after system failure, changes saved).

States



Schedules

- sequence of instructions that specify chronological order of concurrent transactions are executed.

① serial scheduling - one transaction completed then another is executed.

T1
T2

② Concurrent scheduling - in beta execution.

T1
T2
T2
T1

Serializibility - preserve db consistency.
- final result align with sequential execution.

① conflict serializibility - transform in serial schedule by swap
② view serializibility - view equivalent to serial schedule.

Read - compatible instruction
Write - non-compatible instruction.

Lock based Protocol

- Lock : mechanism to control concurrent access to data item

① exclusive - read and write
- lock-X

② shared - read only
- lock-S

	S	E(X)
S	T	F
E(X)	F	F

T1 - S then T2 gets S

- deadlock ⇒ rollback and lock released.
- starvation.

soln: **Two phase locking protocol**

① growing phase - obtain locks, not release.
② shrinking phase - release locks, not obtain

Recovery (Log based recovery) - log records of update activities in db.

① deferred db - record all modification but update in partial commit
② immediate db - immediate update as written/change