DBMS.

→ What is DB?
→ What is DBMS? → managing DB. SQL - communicate to DB.
→ DB VS DBMS.

Types of DB.

SQL DB - Relational, Analytical(OLAP)

NOSQL DB - Column-family, graph, document, key value

→ What r Indexes.
→ How is table scan different from index scan.
→ pages in DB. → pages are stored in B tree in memory
→ DB partitioning
→ DB sharding
→ Exclusive lock Vs shared lock.
→ ER diagrams. to → schema
→ Types of keys in DB.
key
↳ primary key, candidate key, foreign key, composite key, unique key, super key, secondary key, surrogate key
→ anomalies in DB.
↳ update, delete, insert
→ Normalization

→ Functional dependency
→ Normal forms.
→ Transactions
LACID, BASES, recovery, states.

→ ~~R~~

=

DBMS: Crash Course - Vamsi Bharani

Imp Terms:

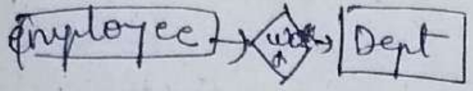
→ Relation - table
→ Tuple - row
→ Attribute - column
→ Domain - finite set of values for a attribute.
→ Schema - design of table.
→ Degree of relation - no. of attributes.
→ intension - schema
→ extension - table.

DQL - Data Query Language - select.
DDL - Data Definition Language - schema, instance, alter DB, rename, create, drop, truncate
DML - Data Manipulation language - insert, delete, update,
DCL - Data Ctrl language - grant permission or access, revoke.

E-R Model.

entity - real world object. (nouns) - person, employee etc.
entity set - collection of entities.
Attributes - properties of those entities.
Relationship - (verb).

→ Types of attributes :- multivalued, single attribute, composite attribute, complex attr, derived attribute.



→ one to one
one to many
many to one
many to many

→ Constraints (Restrictions)
- Domain → sno|name|pho X   sno|fn|ls|pho ✓ (single attribute undali)
- Entity integrity → primary key
- Referential integrity - foreign key (null values undochu) or ref prevu table.

→ Normalization.

4 points :- ① non-atomic → multi/comp attribu
② partial dependence → (A, B) → prim   C A mida kani B kani depend ithe partial depe
③ transitive dependence A → B, B → C so A → C
④ dependence internally

(A, B) ← oka set   (C, D) ← another set : okavela B & C okadani meda okati depend ithe

1) 1NF. → non-atomic avvakudadhu

2) 2NF → 1NF + no partial dependen

3) 3NF → 2NF + no transitive dependen

4) BCNF (3.5 NF) → 3NF + no internal dependen

Indexing:

selection of right block/page becomes faster.

→ primary
→ secondary
→ clustering

Indexing type is done based on key (primary key) or ordering (asc/desc).

Transactions:

collection of diff operation. helps to solve single logical unit of work

ACID properties: → ensured by Transaction manager (s/w).

A - Atomicity (all or none.) anni collections jargali ledha anni yedhi jaragakudhu.

ensured by user/app programs ← C - Consistency - correctness transaction correct ga undali. money detection fuda addition

ensured by concurrency ctrl manager ← I - Isolation - ye transaction ki a transaction cheskovali $T_1, T_2, T_3 \ldots$ anni separate ga work cheyali

D - Durability. operations anni permanent ga undali
↓
ensured by recovery manager.

→ Concurrency ctrl.

↳ ||el or simultaneously ga multiple transactions run avvadam

Types of concurrency ctrl techniques.

1. Lock based protocol:-
   - → simple lock
   - → 2 phase lock
   - → graph based protocol
2. Time stamp based protocol
   - → Timestamp ordering
   - → Thomas write rule
3. Multi-granularity protocol.
4. Multi version protocol.
   - → multiversion 2Phase lock
   - → multiversion timestamp ordering

Simple lock.

- shared lock - read - lock_R
- exclusive lock - write - lock_W.

	shared	exclusive
shared	✓	✗
exclusive	✗	✗

shared ithe malli shared cheyocha
kani migatha cheyalem.

2PL (2 phase lock) → both shared & exclusive will be there but differ on how we obtain.

→ (1) growing phase → kavalisina locks ni acquire chesukuntundi
(2) shrinking phase → akkaglenidhi locks ki vadhilesidhi

3 types of 2PL.

- strict → acquire chesina exclusive locks ni transaction complete ayya varaku vadalakudadhu.)
- rigorous → both exclusive lock & shared lock are not released until transaction done.
- conservation.

↓

Then before committing we can release locks.
opp. to rigorous.