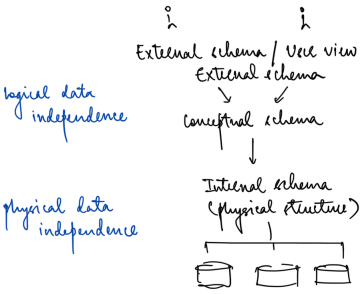
ESA PREP – DBMS

Three schema architecture



Characteristics of Database Approach :

1. Self describing nature – metadata
2. Insulation b/w programs & data
   Data abstraction
3. Support multiple views of data
4. Sharing of data, multi user transaction processing
5. Controlling data redundancy.

field ⟶ record ⟶ file ⟶ flat file.

Constraints :
1. Inherent / Implicit constraint
   – based on data model itself
2. Schema based / Explicit constraints
   – using facilities provided by model
3. Application / semantic constraints

(i) Domain constraint
(ii) Key constraint — constraints on Null values
(iii) Integrity constraints
- Entity integrity
- Referential integrity

DBMS languages

DDL	DML	DCL	TCL
· Create	· Insert	· Grant	· Commit
· Alter	· Select	· Revoke	· Rollback
· Drop	· Update		· Savepoint
· Truncate	· Delete		
· Comment			
· Rename			

Primary Data types

1. Numeric data type: INTEGER, INT, SMALL INT
   FLOAT, REAL, DOUBLE PRECISION
2. Character string
   Fixed length : CHAR(n), CHARACTER(n)
   Varying length : VARCHAR
   CHARACTER VARYING
   CHAR VARYING
3. Bit String : BIT
   BIT VARYING
4. Boolean
5. Date
6. Timestamp
7. Interval

Large Objects ⟶ book review clob (10KB)
image blob (10MB)
movie blob (2GB)

Data structures in large objects :

Datatypes for large objects –

TinyBlob	tinytext
Blob	text
mediumblob	mediumtext
longblob	longtext

setBlob ()
setBinaryStream()

setClob
setCharacterStream()

java.sql.Clob Interface.

SET operations in SQL –

1. Union – removes duplicates
2. UnionAll
3. Intersect
4. Minus
5. Except

CONSTRAINT TYPES IN SQL –

① Key constraint
primary key value cannot be duplicated

② Entity integrity
primary key value cannot be null

③ Referential integrity
foreign key must have a value that is present as a primary key somewhere or NULL.

AGGREGATE FUNCTIONS – Used to summarize info from multiple tuples into a single tuple summary.

- COUNT
- SUM
- MAX
- MIN
- AVG

→ HAVING clause to select entire groups.

→ The WITH clause allows a user to define a table that will only be used in a particular query

Cross tabulation → Pivot table
Data cube – multidimensional generalization of a cross tab

OLAP – Online Analytical Processing

Pivoting – Changing dimensions used in cross tab
Slicing – Creating a cross tab for fixed values
aka dicing.
Roll up – finer granularity → coarser granularity
Drill down – coarser granularity → finer granularity

Group by clause ⟹ Cube, Rollup func.

TRIGGER → 3 components that make it a rule for an active database.

→ Event
→ Condition
→ Action

ECA

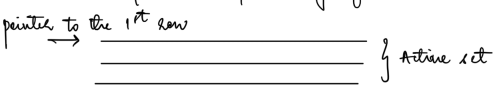
VIEWS

a. Query modification approach
b. View materialization
1. immediate update
2. lazy update
3. periodic update

PROCEDURE – A subroutine or a subprogram.

A collection of pre-compiled SQL statements stored in the database.

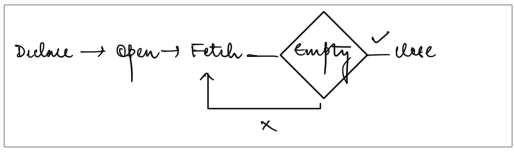
CURSOR – A mechanism that enables traversal over the records of a database.
– facilitate processing of the data



Temporary work area created in system memory

+ There are implicit & explicit cursors.

Properties
(i) Read only
(ii) Non-scrollable [top to bottom only]
(iii) Asensitive



How to use a cursor in mySQL

Informal guidelines for Relational Schemas

1. Making sure that the semantics of the schema is clear in the schema.
2. Reducing redundant information in the tuples
3. Reducing NULL values in the tuples
4. Disallowing the possibility for generating spurious tuples

G1> Informally, each tuple in a relation should represent one entity or instance

Attributes of different entities should not be mixed in the same relation.

Use only foreign keys to refer to other entities.

Entity & relationship attributes should be kept apart as much as possible.

G2> The schema should not suffer from insertion / update / delete anomalies.

G3> Relations should be designed that their tuples will have as few NULL values as possible.

G4> Relations should be designed to satisfy the lossless join condition.

No spurious tuples should be generated by a natural join of these relations.

FUNCTIONAL DEPENDENCIES

Constraints that are derived from the meaning and interrelationships of the data attributes.

+ functional dependency (FD) is a relationship that exists b/w two attributes, where one attribute can directly or indirectly be derived from the other.

A set of attributes X functionally determines a set of attributes Y if the value of X determines a unique value of Y.

y is functionally dependent on x

Normalization

A process of organising data in the database to avoid data redundancy & update / insert / delete anomalies.

Refer to Note : DBMS U4 for U4 notes.

SECRECY | INTEGRITY | AVAILABILITY (CIA)

Threats to the database

- loss of confidentiality
- loss of integrity
  - unauthorized changes
- loss of availability
  - loss of access

refers to protection of data from unauthorized disclosure

Security Mechanisms

- Discretionary Access control

  Based on the concept of access rights / privileges for objects + grant / revoke priv.

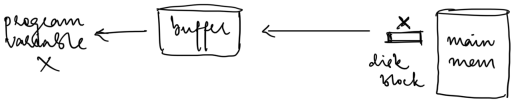
- Mandatory Access control

  Classify data & users into security classes & then implement appropriate security policy.

  Enforce multilevel security.

TRANSACTION PROCESSING

read_item(X)



DBMS buffers - several in DB cache
- buffer replacement policy

Concurrency control

- Lost-Update problem

- Interleaved operations
- Results in incorrect value of DB items

- Temporary update / Dirty read problem

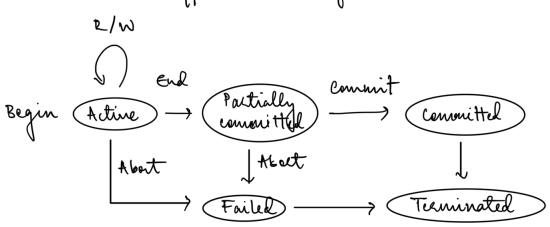
- T1 updates a transaction but fails;
  T2 reads temporarily incorrect value of X.

- Incorrect summary problem

- T1 is calculating an aggregate fn on some items in the DB.
- T2 is updating some items in the DB.
- Incorrect values may be aggregated.

- Unrepeatable Read problem

- T reads the same item twice.
- T' changes the item between the reads.
- T receives different values for 2 reads.



System log
- keeps track of transactions
- sequential append only file

- periodically backed up
- not affected by failure.

[start_transaction, T]

[write_item, T, X, old_value, new_value]

Desirable properties of transactions

Atomicity – T is performed in entirety

Consistency preservation – DB from 1 state to an.

Isolation – Not infering w/ other transactions

Durability / Permanency – changes persist

Two operations conflict if changing their order results in a different outcome.

- Cascadeless schedule
- Strict schedule
- Serializable schedules

SQL commands that support transaction

- Commit, Rollback, Savepoint
- SQL TCL used w/ DML for in/up/del.

Lock modes:

(i) Shared mode: Read lock (X)

(ii) Exclusive mode: Write lock (X)

Variations of 2 phase locking –

1. Binary
2. Conservative
3. Strict 2PL
4. Rigorous 2PL

NoSQL characteristics

- Scalability
- Availability
- Replication models
- Sharding of files – Horizontal partioning
- High performance data access
- Not requiring a schema
- No powerful query language
- Versioning

SARS – NoV is a high performing DB.

1. Document database – Mongo DB
2. Key Value DBs – Berkeley DB
3. Wide Column DBs – Cassandra
4. Graph DBs – Neo4j

SQL	NoSQL
Relational DBMS	Distributed DBMS
Vertically scalable	Horizontally scalable.
Fixed schema	Dynamic schema
Not suited for	Hierarchical data
Complex queries	Not suited

MONGO DB — DOCUMENT DATABASE

- Embedded Data Model
- Normalized data model.

```
use ESA db
db
show dbs
db.dropDatabase()
```

```
db.createCollection("ESA")
show collections

db.student.insert({'Name':'Deepthi'})
  creates collection student, inserts document

db.student.drop()

db.employee.find({'salary':'9000','skill':'network'})
db.employee.find({'age':{$lt:'30'}})
  return documents where date is b/w 1940 & 1960
db.employee.find({age:{$gt:newDate('1940'),
                       $lt:newDate('1960')}})

db.employee.update({'skill':'mongodb'},{$set:{'sal':10}})
```

DYNAMO DB – KEY VALUE DB

Primary key
- single attribute – hash type
- pair of attributes – hash & range

SSDs, high availability, scalability, performance

HBASE – COLUMNAR DB

- compression
- aggregation
- scalability
- Fast to load & query

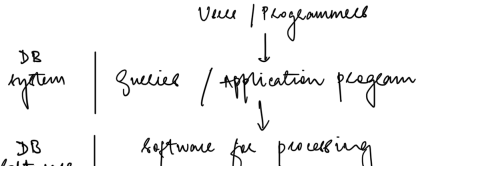
```
create 'EMPLOYEE', 'Name', 'Address', 'Details'

put 'EMPLOYEE', 'Row1', 'Name:Fname', 'Deep'
put 'EMPLOYEE', 'Row1', 'Details:Job', 'Engineer'
```

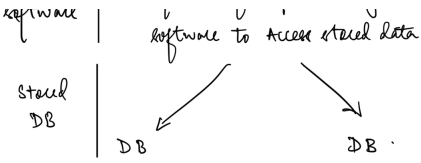
SYNTAX

```
create <table name>, <column family> ....
put <table name>, <column family>:<column qualifier>
                                  , <value>

scan <table name>
get <table name>, <rowid>
```

NEO4j – GRAPH DB

```
create (n)

match (n) return (n) limit 2
match (n) where id(n) = 1 return n
                          delete n

sudo service neo4j stop
sudo rm -rf /var/lib/neo4j/data*

create (n:Person)

match (n) where n:Person:Indian return n

match (n) where id(n)= 0
remove n:manager
set n:director
return n

create (x:Book {title:'THGTTG'})
return x
```





- Self describing nature
- Insulation b/w prog & data
  data abstraction
- support multiple views
- multi user transaction processing

The internal level has an internal schema which describes the storage structure of the database

The conceptual schema — describes the structure of the whole DB for a community of users.
- Hides details of physical storage structures