Relational Model Introduction

Database Definition

—> Organized collection of inter-related data that models some aspect of the real world.

Database are the core component of most computer applications.

Flat File Strawman:

like csv (comma-seperated value) files

—> Use a seperated file per entity

—> The application has to parse files each time they want to read/update records.

xxx, xxx, xyx

File system Disadvantage:

- Data redundancy and inconsistency
- Integrity problem
- Atomicity problem
- Security problem

...

Database management system.

A DBMS is software allows applications to store and analyze information in a database.

General purpose: definition, creation, querying, update, administration of databases.

Data Model

- Relational ← most DBMSs
- Key / Value
- Graph
- Document
- Column-family

} NoSQL

- Array / Matrix ← Machine Learning

~
~

Relational Model

Structure : The definition of relations and contents

Integrity : Ensure the database's contents satisfy constraints.

Munipulation : How to access and modify a database's content.

---

relation : unsorted set that contain the relationship of attributes that represents entities.

tuple : a set of attribute values in the relation

Artist (name, year, country) ← relation

name	year	country
x	y	z

← a tuple

we can use "table" and "relation" interchangeably also "record" and "tuple".

A relation's primary key uniquely identifies a single tuple

A foreign key specifies that an attribute from one relation has to map to a tuple in another relation.

Data Munipulation Language (DML)

Store and retrieve information from a database.

Procedural

→ The query specifies the (high level) strategy the DBMS should use to find the desired result.

Non-Procedural

→ The query specifies only what data is wanted and not how to find it.

Relational Algebra

take one or more relations as its input, outputs a new relation

σ	Select
π	Projection
∪	Union
∩	Intersection
−	Difference
x	Product
⋈	Join

Relational Model : Query

The relational model is independent of any query language implementation.

SQL is what we use mostly.

Through the query language, we don't have to parse every line of the file. We can get the result exatly.

```
for line in file:
    record = parse(line)
    if "Juice Wrld" == record[0]:
        print int(record[1])
```

```
Select year from artists

where name = "Juice Wrld";
```

Relational algebra define the primitive for processing queries on a relational database

Advanced SQL

Relational Language

User only needs to specify the answer they want, not how to compute it.

The DBMS is responsible for efficient evaluation of the query. (Query optimizer)

SQL → Structured Query Language

Most DBMSs at least support SQL-92.

SQL is not a single language, it's a collection of:

- Data Munipulation Language (DML)
- Data Definition Language (DDL)
- Data Control Language (DCL)
- View definition
- Integrity & Referential Constraints
- Transactions

SQL is based on bags (duplicates) not sets (no duplicates)

Aggregates

Functions that return a single value from a bag of tuples.

- AVG(col)
- MIN(col)
- MAX(col)
- SUM(col)
- COUNT(col)

only be used in the SELECT output list.

COUNT, SUM, AVG support DISTINCT.

Non-aggregated values in SELECT output clause must appear in GROUP BY clause.

```
SELECT AVG(s.gpa) As avg-gpa, e.cid
FROM enrolled AS e, student AS s
WHERE e.sid = s.sid
AND avg-gpa > 3.9
GROUP BY e.cid
```

X → HAVING avg-gpa > 3.9

CMU SCS

String Operations

	String Case	String Quotes
SQL-92	Sensitive	Single Only
Postgres	Sensitive	Single Only
MySQL	Insensitive	Single/Double
SQLite	Sensitive	Single/Double
DB2	Sensitive	Single Only
Oracle	Sensitive	Single Only

```
WHERE UPPER(name) = 'KAYNE'        SQL-92
WHERE name = "KAYNE"               MySQL
```

54

LIKE is used for string matching

'%' -> Matches any substring

'_' -> Matches any one character

SQL standard says to use || operator to concatenate two or more strings together.

Date / Time Operation:

SELECT NOW( ); SELECT CURRENT-TIMESTAMP( );

postgres: SELECT DATE('2018-08-19') - DATE('2018-01-01');

MYSQL: SELECT DATEDIFF(~)

Output redirection

-> store query results in another table.

must not already be defined

same column

1) SELECT xxx INTO xxx From xxx ;
2) CREATE TABLE xxx (
   SELECT xxx FROM xxx);
3) Insert INTO xxx (SELECT xxx FROM xxx);

Output Control

ORDER BY <column> [ASC | DESC]

Default is ascending.

LIMIT <count> [offset]

-> Limit the number of tuples returned in output.

Nested Query

Queries containing queries.

It's difficult to optimize ;

All ANY = IN EXISTS

Window Functions

Perform a calculation across a set of tuples that related to a single row.

Like an aggregation but tuples are not grouped into a single output tuples.

Aggregation Functions

Special window Functions

-> ROW_NUMBER()

-> RANK()

OVER specifies how to group together tuples

PARTITION BY specifies group

SQL is not a dead language.

You should alway strive to compute your answer as a single SQL statement.

Database Storage

outline :

- Relational database
- Storage
- Execution
- Concurrency Control
- Recovery
- Distributed Database
- Potpourri

Query planning

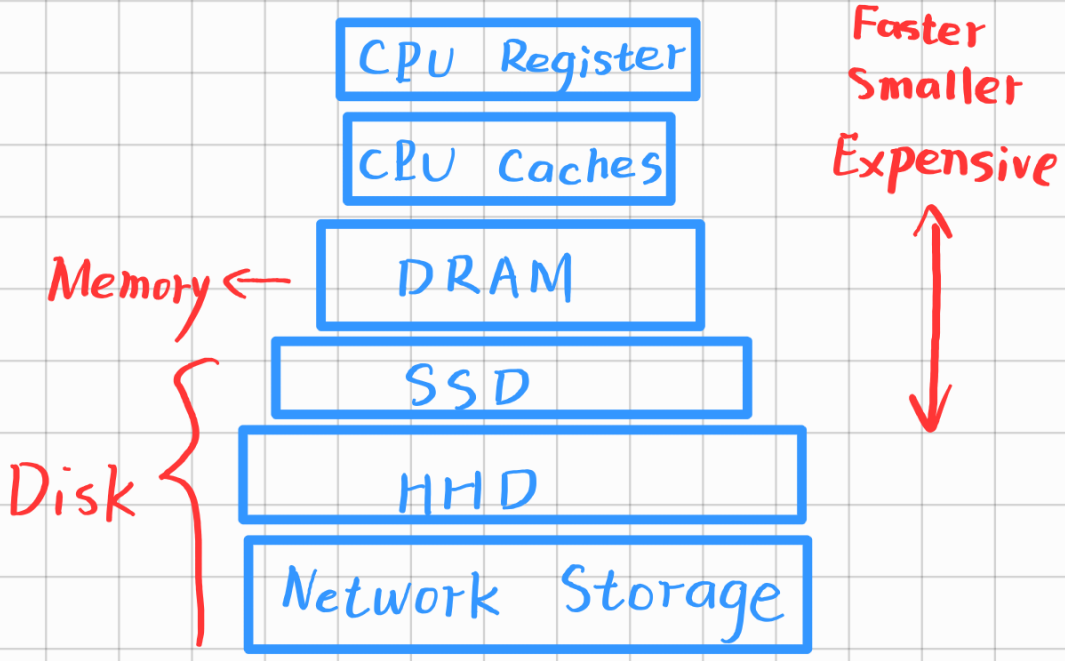
Operator Execution

Access Methods

Buffer Pool Manager

Disk Manager

DBMS's components manage the movement of data between non-volatile and volatile storage.



Reading/writing to disk is expensive!

Database Storage.

Problem 1: How the DBMS represents the database in files on disk.

Problem 2: How the DBMS manage its memory and move data back-and-forth from disk.

File Storage
Page Layout
Tuple Layout

File Storage

File -> Page -> metadata, tuple ...

The DBMS stores a database as one or more files on disk.

-> OS doesn't know anything about the contents of these files.

Storage Manager

is responsible for maintaining a database's file.

It organize the files as a collection of **pages**.

↓

Fixed-size block of data

-> contain tuples, meta-data, indexes, log records.

Most system do not mix page types.

Some system require a page to be self-contained

Each Page is given a unique identifier.

-> DBMS uses an indirection layer to map page ids to physical location.

Different DBMSs manage pages in files on disks in different ways:

- Heap File Organization
- Sequential / Sorted File Organization
- Hashing File Organization

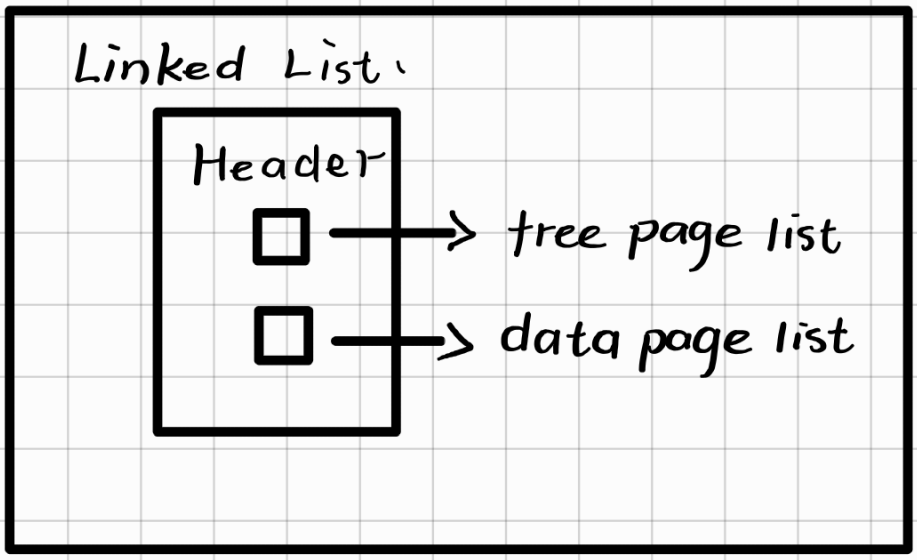
Database Heap

A Heap file is an unsorted collection of pages where tuples are stored in random order.

Need meta-data to keep track of what pages exist and which ones have free space.

To represent a heap file:

- Linked List
- Page Directory (Better approach)



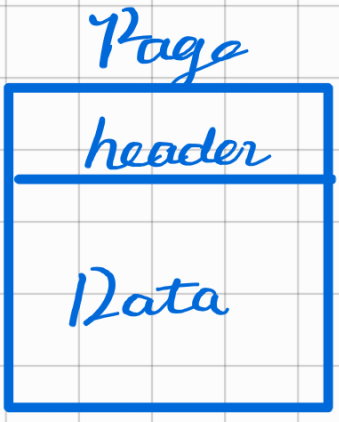
Page Directory

DBMS maintains special pages that tracks the location of data pages in the database files

Page Layout

every page contains a header of metadata about the pages's content :

- Page Size
- Checksum
- DBMS version
- Transaction Visibility
- Compression Information



how to organize the data stored inside the page.

Two approaches :

- Tuple-oriented
- Log-structure

Tuple Storage

Strawman Idea:
keep track of the number of tuples in a page and then just append a new tuple to the end

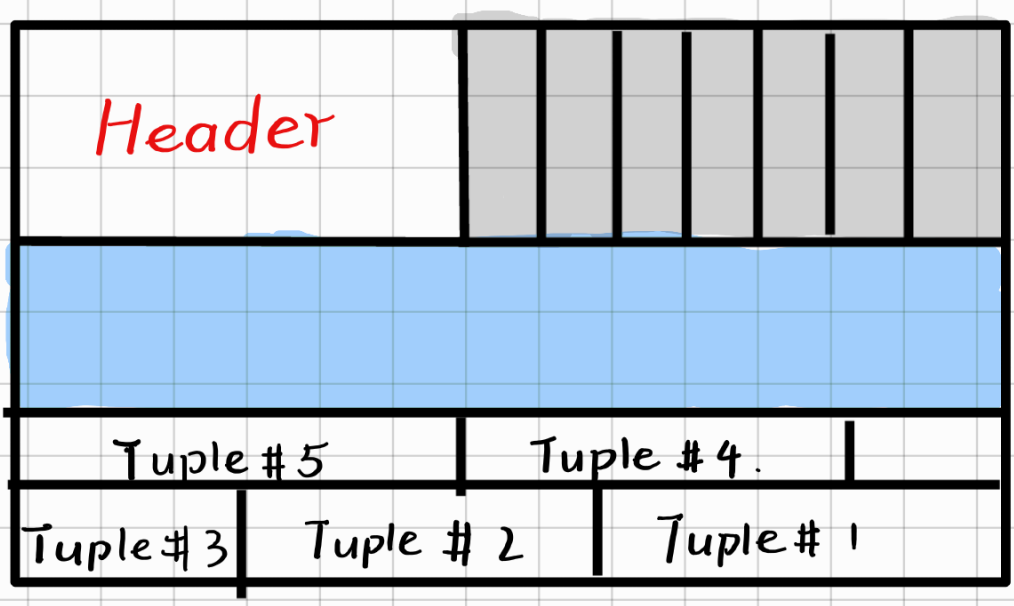
Slotted Pages:

- The most common layout scheme.

The slot array maps "slots" to the tuples' starting position offset.

The header keep track of:

- The number of used slots
- The offset of the starting location of the last slot used



Each Tuple is assigned a unique record ID to be tracked.

→ Most common : Page_ID + offset / slot

→ Can also contain location info.