Introduction

- **DBMS** = a (complex) software system used by a group of users - each with different information needs.
  
  1. Define the structure of the data
  2. Manage (insert, delete, update) the data
  3. Query the data.

- Efficiency is an important requirement.
- Maintain consistency (concurrency control) of data.

2 sources of commands to the DBMS:

1. **DBA** (defining structure or schema) at the Database
   - (managed by the DBMS)

2. User issuing commands interactively + Application Programs (no user control)
Overview DBMS

User/Apps

- queries
- updates

(CQL) query compiler

- query plan

Execution engine

Transaction manager

Logging + recovery

DDL compiler

Database definition

DDL commands

Access index file records

Index/file/recurse manage

Page commands

Buffer management

Buffers

Read/write pages

Storage manager

Storage

Lock table

Concurrency control

Metadata

Statistics

Log pages

Metadata

Data

Indexes
- Defining Data: Data Definition Language (DDL)

```sql
CREATE TABLE Student
(
  StudID char(4) NOT NULL,
  Name char(30) NOT NULL,
);
```

- Data Definitions are stored in Database "Catalog" (Meta Data file)

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Attr. Name</th>
<th>Attr. Type</th>
<th>SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student</td>
<td>StudID</td>
<td>Char</td>
<td>10</td>
</tr>
<tr>
<td>Student</td>
<td>Name</td>
<td>Char</td>
<td>3</td>
</tr>
</tbody>
</table>
Overview Query Processing

Query: `select ... from student ... where ...`

- **Parsed**
- **Optimized**

Better query plan (= sequence of actions to answer query)

Execution engine

Read/write data

Storage manager

Result

May require synchronization and locking !!!
Concurrency Control

- Transactions must appear to
  execute in isolation
  (see no effect of other transactions)

- \( T_1: x = x + 1 \)
  \( T_2: x = x - 1 \)

\[
\begin{align*}
\text{Read}(x) - &\quad x = 4 \\
&\quad x = x + 1 \\
&\quad x = 5^- \\
&\quad \text{Write}(x) \\
&\quad x = 5 \\
\text{Read}(x) - &\quad x = 4
\end{align*}
\]

But if not isolated \( \Rightarrow x = 3 \) or \( x = 5 \)

\( T_1 \rightarrow T_2 \rightarrow T_1 \)

\( x = 4 \)

Grammar solution: use locks.

problem: dead lock (lade)
Overview Transaction Processing:

Transaction: a group of operations that must be executed atomically and the effect of the execution must be:

1. Persistent (durable): even if the system fails
2. Consistent even when multiple transactions are processed simultaneously.

Transaction Processing (process):

Concurrency Control Manager (Scheduler):
- Make sure
  1. Transactions are atomic
  2. Transactions are isolated from each other.

Logging + Recovery Manager:
- Ensure persistence (durability) of committed transactions.
Logging: ensure durability
(in the face of system failure)

Eg: \[ x = 4 \]

\[ T_1: x = x + ! \]

Read \((x)\)
\[ x = 4. \]
\[ x = x + 1 \]

Write \((x)\)
\[ x = 5. \text{ (still in memory buffer, failed)} \]

Commit transaction.

Write \((x)\)
\[ x = 5 \text{ not updated.} \]

Use lugs.
Types of Data Used by the DBMS

1. User Data (content of the Database)
2. Meta Data (Data that describes the file structures of the user data).

<table>
<thead>
<tr>
<th>S</th>
<th>Student</th>
<th>StudID</th>
<th>Char 5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Student Name</td>
<td>char 30</td>
<td></td>
</tr>
</tbody>
</table>

(3) Statistics on (some) data files:
- # records
- Max value of some field
- Avg value etc.

(4) Indexes: data structures (stored in files) that support efficient access to user data.
Main-Memory Buffers and Buffer Manager.

Memory Hierarchy:

- Super slow: > 1 sec
- Slow: 50 ms
- Fast: 50 nsec
- Super fast: 1 nsec

Virtual memory (pages)
File System
Main memory
Cache

Tertiary storage
(Tapes, CD racks with mechanical arms)

Peta bytes
Terabyte
Giga bytes
Mega bytes

Disks
Facts: Data can ONLY be processed when it resides in Main Memory. Size is SMALL relative to the amount of data stored in Secondary/Secondary storage.

Buffer Manager: Software system responsible for:

1. Partition the available main memory into buffers (fixed size).

2. Manage the buffers so that data needed to process query is available as soon/frequent as possible in main memory.
Query + Query Processing:

- Query = asking the DBMS to retrieve information that satisfy certain criteria (boolean criteria).

- Query Processor:
  
  **Query Processor**
  
  **Query Compiler**
  
  **Query Pre-processor**
  
  **Query Optimizer**

- Query Parser:
  
  build tree structure from the "textual form" of the query (Parse tree).

- Query Pre-processor:
  
  semantic check transform parse tree to "tree of algebraic operators".

- Query Optimizer:
  
  transforms initial query plan to a "better" query plan.
Execution Engine

- executes queries and other transactions concurrently.

- will use:
  - buffer manager
  - transaction manager
  - concurrency manager.
Course Overview

1) Storage Management
   - Disk Structures
   - File Structures
   - Single Dim. Index (B-tree)
   - Multi-Dim. Index

2) Query Processing
   - Basic query execution
   - Query compiler
   - Query optimization

3) Transaction Processing
   - Concurrency control
     (locks, time stamps)
   - Logging, recovery