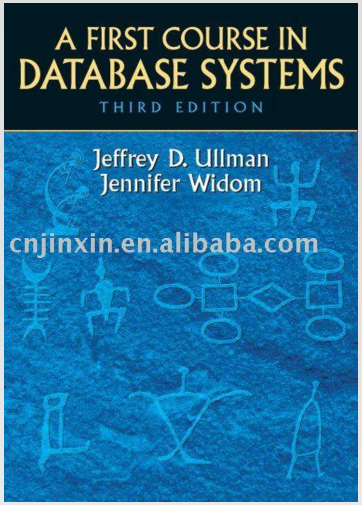
CMPT 354
Database Systems I

Spring 2012

Instructor: Hassan Khosravi

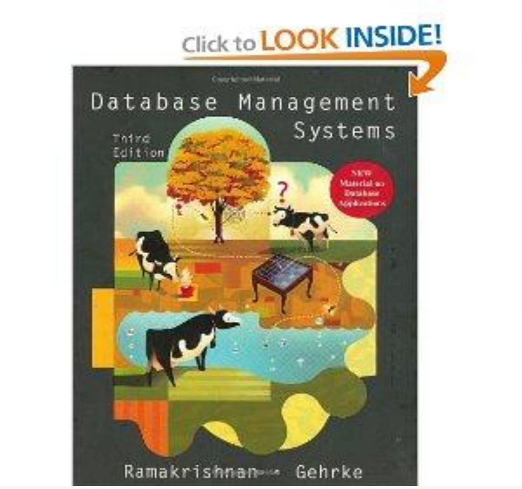
Textbook

- *First Course in Database Systems, 3rd Edition*.
  - Jeffry Ullman and Jennifer Widom

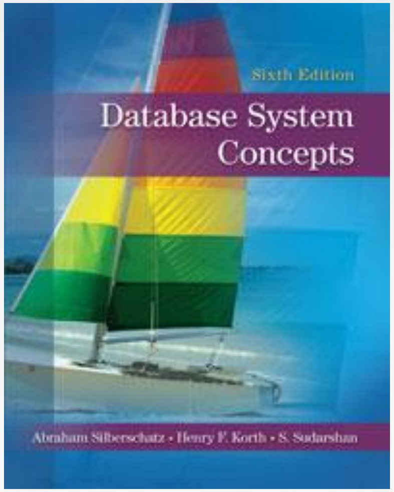


  - Other text books

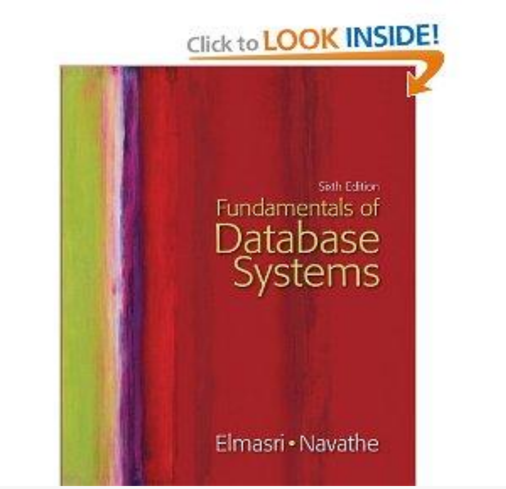
Ramakrishnan



SILBERSCHATZ



Ramez Elmasri



Course Grading

- Three quizzes 9% each
  - The quizzes are from the exercises in the textbook (excluding the double exclamations)
  - around 30 minutes
  - First quiz 25th of January
  - Second quiz 22nd of February
  - Third quiz 28th of March
- One project 15%
  - Given out midway through the course
  - You have until final day of classes (11th April) to do it
  - Its done in pairs
  - It requires a report and a demo presentation.
- Midterm 6th of March in class (20%)
- Final (40%) 17th April 19:00 - 22:00
  - You must be able to attend the final exam!

Teaching Style

- **Motivate the students**. I feel it is the duty of the instructor to present the subject in a motivating and engaging manner.
  - **Get the students involved.**
- **The main objective is to cover the text book**
- **Dealing with unfortunate timing of the class**
  - 3 hours in the evening, creativity to stop people from falling asleep
  - I will use Dr. Widom’s lecture slides as summaries
- **Set clear and realistic goals.** Students respond best to goals that are both challenging and achievable.
- **Encourage Team work:** I believe there should be emphasis on collaboration, planning and being able to clearly express ideas. Complex scientific projects are rarely the work of an individual; students must learn to organize and work as teams.
- **Final grade:** Normal distribution
- **Always respect the students.**

Content of CMPT 354

- Course Website is http://www.cs.sfu.ca/~hkhosrav/personal/db/354-2012.html
- About CMPT 354
  - Introduction the world of database systems
  - Relational database modeling
  - Design theory for relational databases
  - Higher level database models (E/R models)
  - Algebraic and logical query languages
  - The database language SQL
  - Constraints and triggers
  - Views and indexes
  - The semi-structured data model
  - Advance topics in relational databases
    - Security and authorization
    - On-Line Analytic Processing OLAP
    - Data mining

Database Evolution

- What is a database?
  - A collection of information that exists over a long period of time.
- Database refers to a collection of data that is managed by DataBase Management System(DBMS)
- DBMS is expected to
  1. Allow users to create new database
  2. Give users ability to query (question) and modify the data
  3. Support the storage of very large amounts of data with efficient access to (2)
  4. Enable durability – enable recovery in case of failures or intentional misuse.
  5. Control access from many users without allowing unexpected interaction among users (*Isolation*), without allowing partial action on data (Atomicity).

Database Management System (DBMS)

- Database Applications:
  - Banking: all transactions
  - Airlines: reservations, schedules
  - Universities: registration, grades
  - Sales: customers, products, purchases

Question: Why have database systems (and not just directly use a file system)?

Early Database Management Systems

- In the early days, database applications were built directly on top of file systems
- File systems allow storage of large amount of data (3) over long period of time however
  - They do not directly support querying and modifying data (2)
  - Their support for database creation is limited to creation of files (1)
  - You can lose data that has not been backed up (4)
  - Atomicity of updates (5)
    - Failures may leave database in an inconsistent state with partial updates carried out
    - Example: Transfer of funds from one account to another should either complete or not happen at all
  - Uncontrolled concurrent accesses (Isolation) can lead to inconsistencies
    - Example: Two people reading a balance and updating it at the same time

Early Database Management Systems

- Drawbacks of using file systems to store data cont.
  - Data redundancy and inconsistency
  - Difficulty in accessing data
    - Need to write a new program to carry out each new task
  - Integrity problems
    - Integrity constraints (e.g. account balance > 0) become "buried" in program code rather than being stated explicitly
    - Hard to add new constraints or change existing ones
  - Security problems
    - Hard to provide user access to some, but not all, data

- Database systems offer solutions to all the above problems

Interesting Stuff About Databases

- Databases used to be about stuff like employee records, bank records, etc.

  → They still are.

- But today, the field also covers all the largest sources of data, with many new ideas.
  - Web search.
  - Data mining.
  - Scientific and medical databases.
  - Integrating information.

Bigger and Bigger Systems

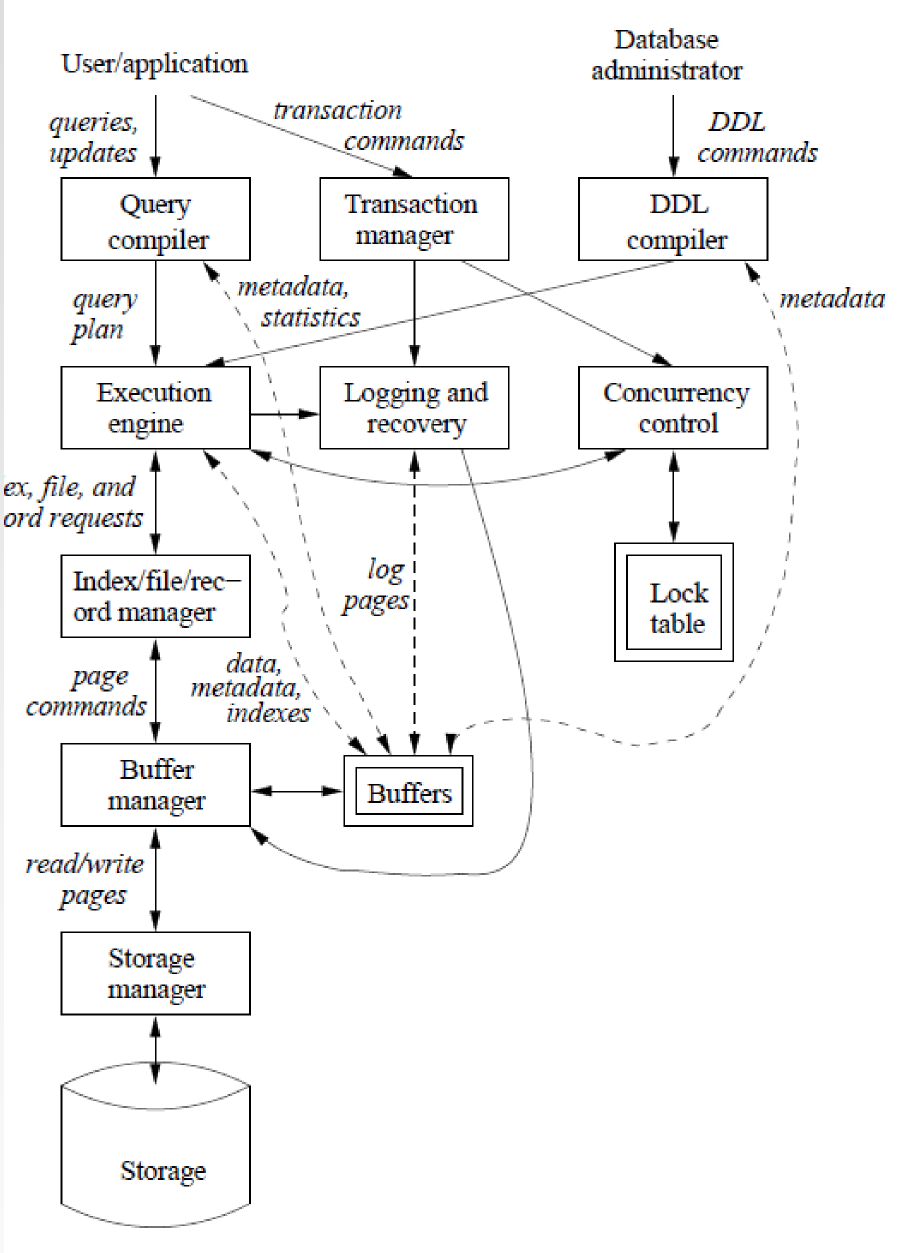
- Google holds 1 petabyte (1,000,000 gigabyte) data
- Satellites send down petabytes of information
- A picture is actually worth way more than a thousand words. Flickr stores millions of pictures and supports search for them
- Youtube holds millions of movies and they are easily accessible

More Interesting Stuff

- Database programming centers around limited programming languages.
  - One of the only areas where non-Turing-complete languages make sense.
- You may not notice it, but databases are behind almost everything you do on the Web.
  - Google searches.
  - Queries at Amazon, eBay, etc.

Overview of DBMS

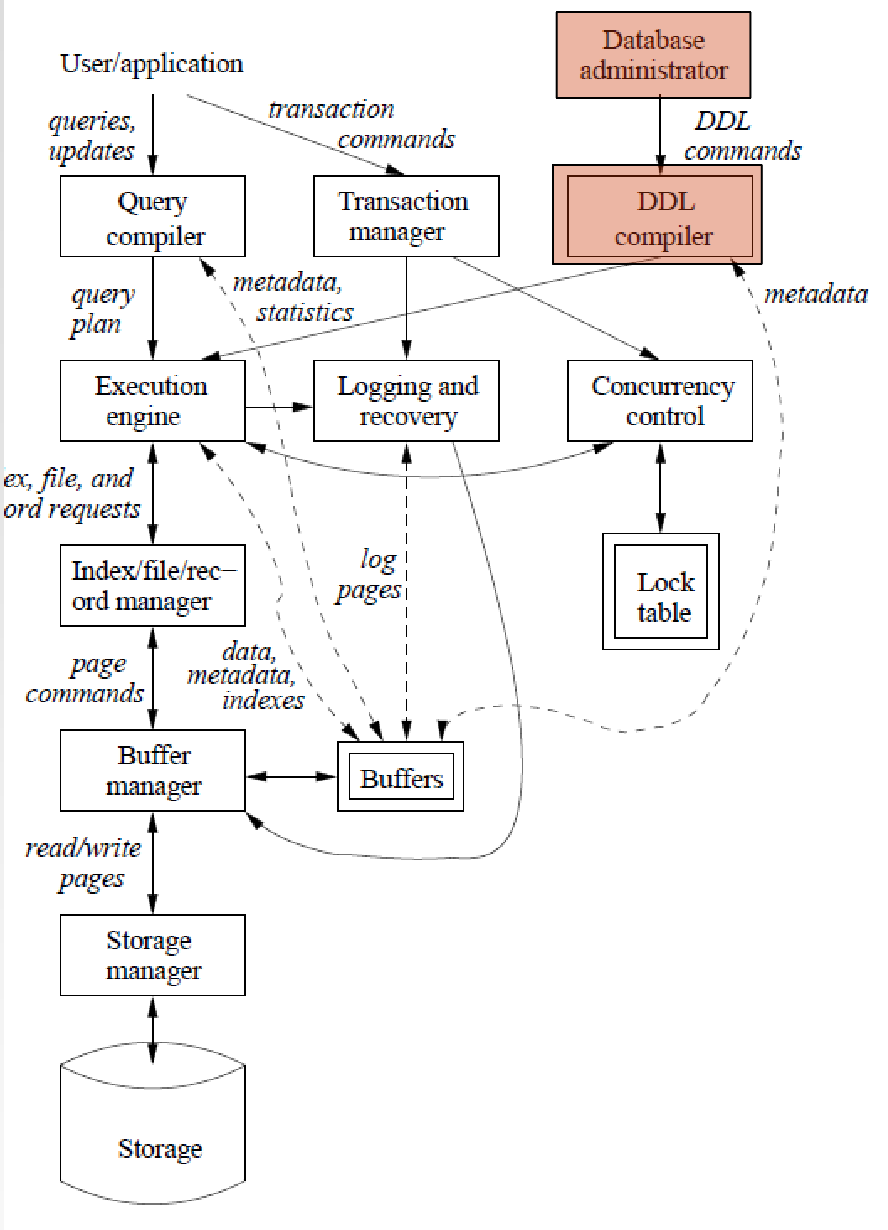
- Single boxes represent system components
- Double boxes represent in memory data structure
- Solid line indicate control and data flow
- Dashed lines indicate data flow only



Database Definition Commands

- Database administrator (DBA) – responsible for the structure or schema of database
  - Example: A university DBA decides on a table with student, course , grade columns. Grade can only be (A, B, C, D)

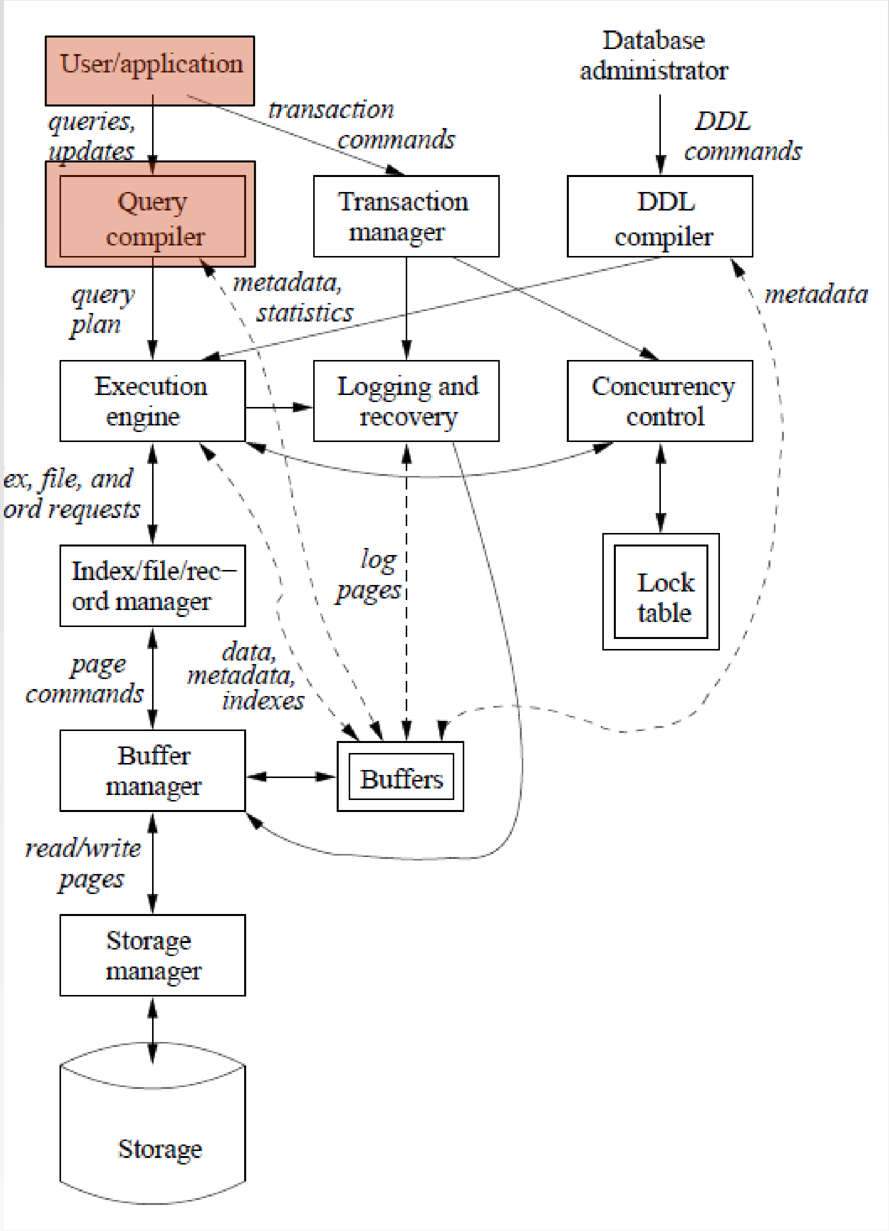
- DBA use data-definition language (DDL) which are processed in DDL compiler



Query Processing

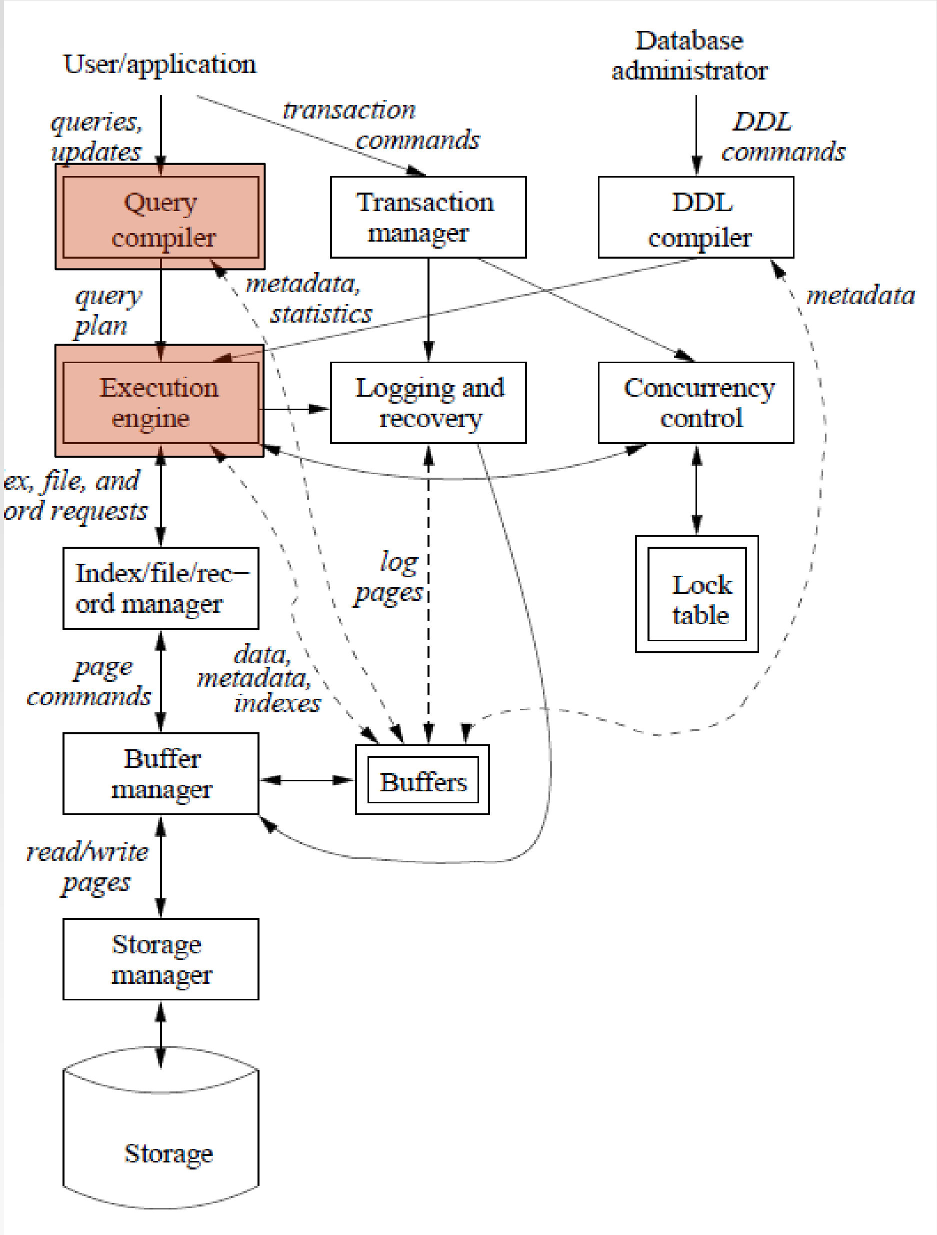
1. Conventional user query or modify data
    1. Example: a user Jack may want to take the course 101 for Spring 2012.

- Majority of interactions are queries or updates using data manipulation language (DML) which are parsed and optimized by Query compiler.
- The query compiler translates the query into an internal form called a query plan



Query Processing

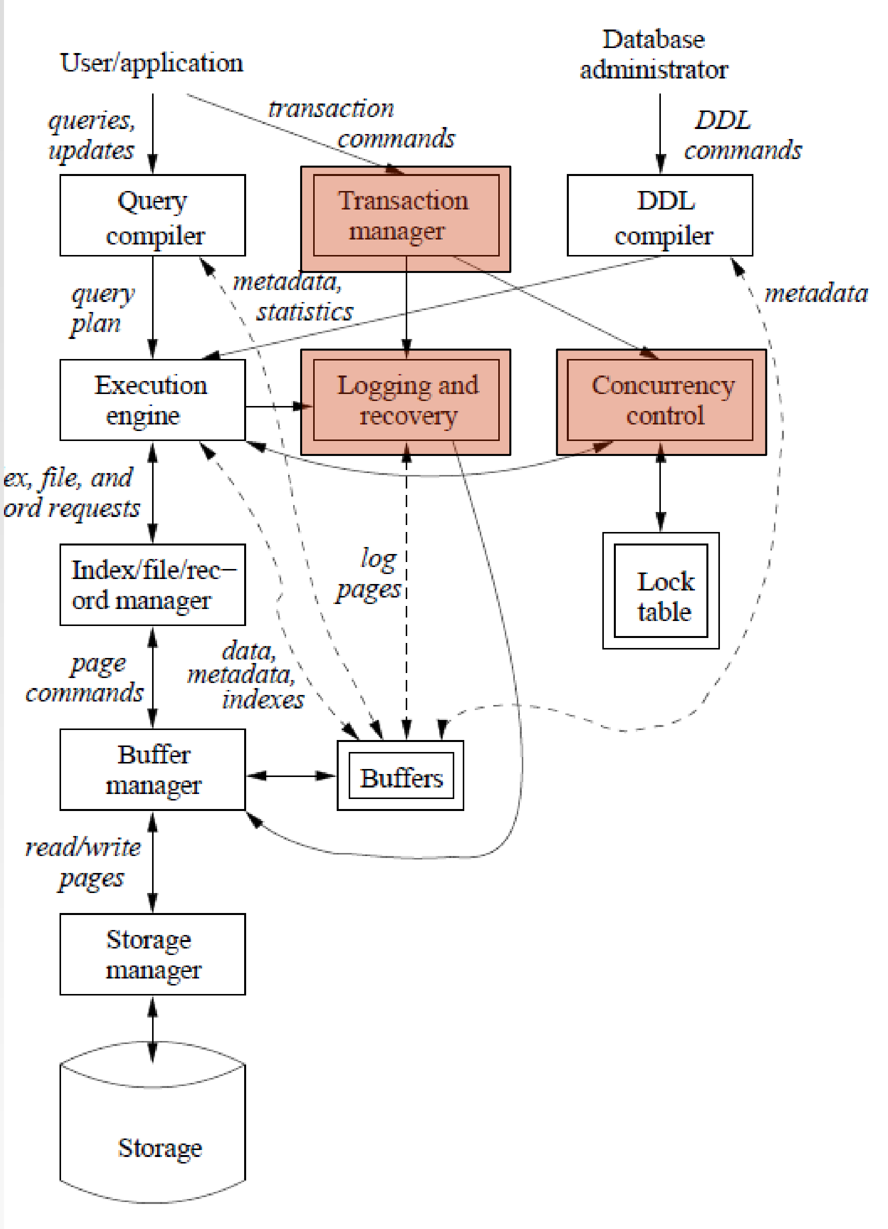
- The query compiler consists of three major units
  - Query parser: builds tree of the structure from text
  - Query preprocessor: performs semantic checks on the query
  - Query optimizer: transforms the initial query plan into the best available sequence of operations.
- The query plan is passed to the Execution engine which has the responsibility of executing each of the steps



Transaction Process

- Queries and DML actions are also handled by Transaction Manager. They are grouped into transactions which are units that most be executed *atomically* and in *isolation* from one another. Concurrency control manager is responsible for this.

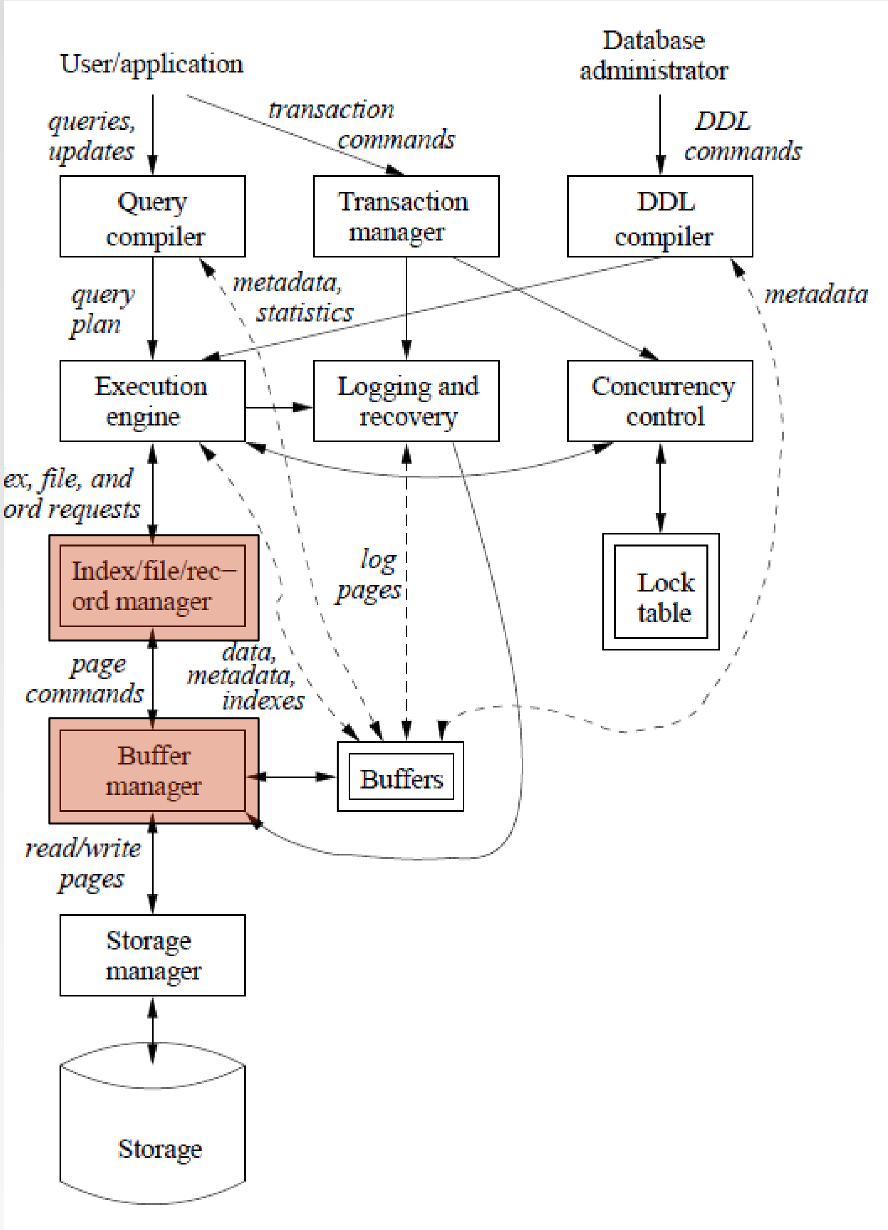
- The transaction most be durable – if completed most be persevered even if the system fails right after completion of transaction. Logging and recovery manager is responsible for this.



Query Execution

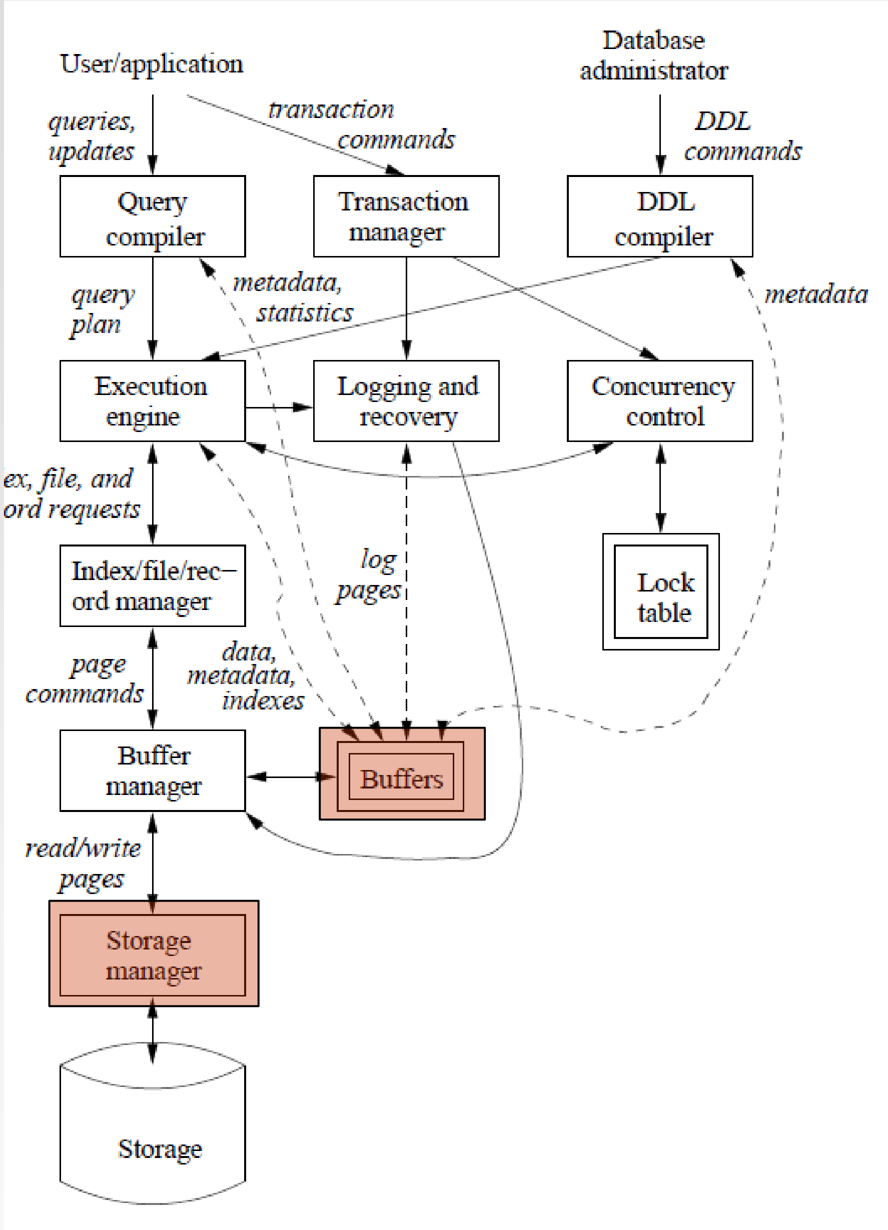
- The execution engine issues a sequence of request for small pieces of data (records or tuples). The request is submitted to buffer management

- The request for data are passed to the Buffer management. The data is usually stored on secondary storage (hard drive). However to perform any operation on data, it must be in the main memory. The buffer manager communicates with Storage manger to get this data.



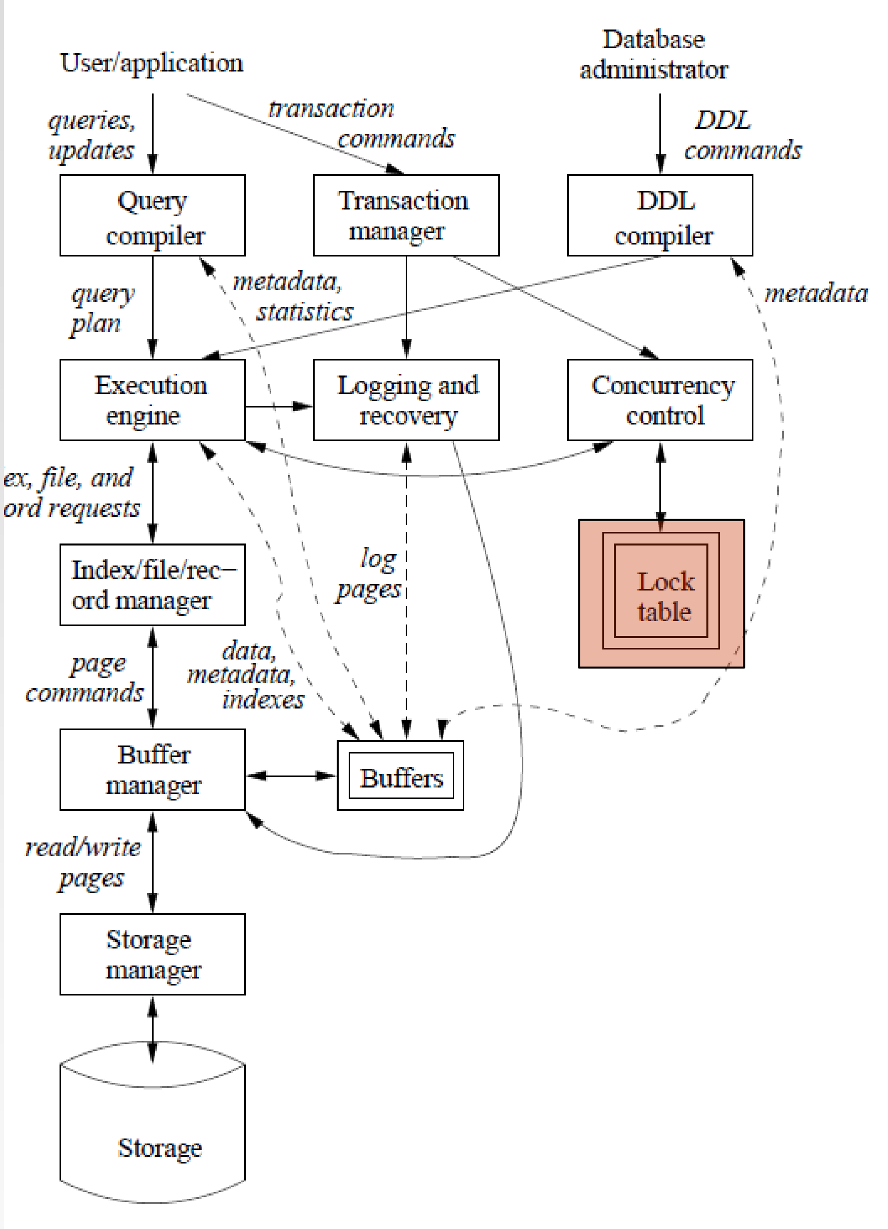
Storage and Buffer Management

- The storage manger keeps tack of the location of files on the disk and provides the buffer manager with the file
- The buffer manger is responsible for partitioning the main memory into page sized buffers. All DBMS components that need information from disk interact with buffer and buffer management directly, or through execution engine. This information includes
  - Data
  - Metadata
  - Log records
  - Statistics



Concurrency Control and Deadlock Resolution

- Concurrency manger must assure that the individual actions of multiple transaction are executed such that the effect is the same as running them one at a time.
- It usually works by maintaining locks on the records. Example moving money from one account to another
- Its possible to get into a situation where all the transactions are waiting for each other due to locks. Lock table calls roll back or abort on some of the transactions to resolve deadlock.



Quick Summary

- Lecture given by Dr. Widom

End of Introduction