UMBC

AN HONORS UNIVERSITY IN MARYLAND

CMSC 461, Database Management Systems
Spring 2018

Course Logistics & Chapter 1 – Introduction

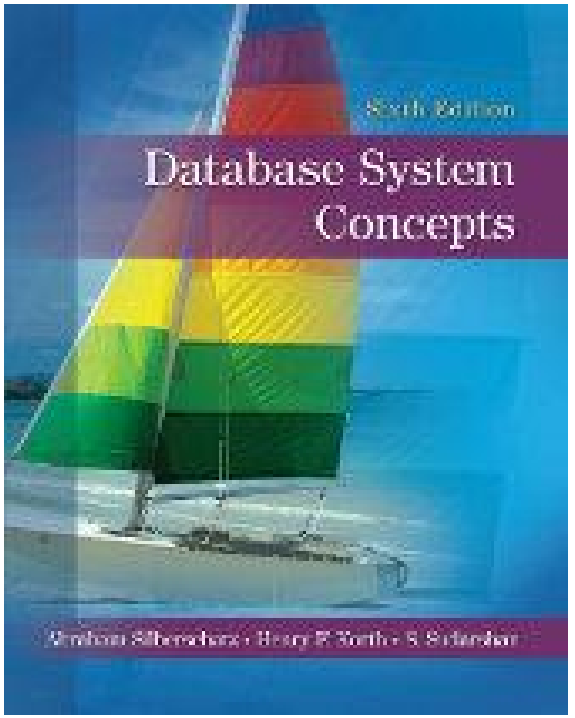
These slides are based on “Database System Concepts” book and slides, 6th edition , and the 2009 CMSC 461 slides by Dr. Kalpakis

Dr. Jennifer Sleeman https://www.csee.umbc.edu/~jsleem1/courses/461/spr18/index.html

The Book

Abraham Silberschatz, Henry H. Korth, and S. Sudarashan, **Database System Concepts**, Sixth Edition, McGraw-Hill.

ISBN-13: 978-0073523323
ISBN-10: 0073523321



Course Logistics

- Review course web site
  - https://www.csee.umbc.edu/~jsleem1/courses/461/spr18/index.html
- Review syllabus
- Review schedule
- Introduction to Slack
  - https://umbc-cmsc-461-spr-18.slack.com

“The world of databases is exciting, changing, and challenging”

Exciting...

- Databases are 'everywhere'
- Web Search
- Data Mining
- Medical data
- Scientific data

Changing...

- Relational vs non-relational
- Key-Value based databases
- Document based databases
- Graph-based databases

Challenging...

- More and more data
- Big Data problems
- Harder problems to model
- Interoperability

What does this mean for you...

- Job opportunity
- Cool research problems

Lecture Outline

1) Database and DBMS
2) Purpose of Database Systems
3) Data
4) Languages
5) Relational Databases
6) Database Design
7) Storage and Querying
8) Transactions
9) Architecture

What is a database?

What is a database?

- Collection of data
- Organized
- Relevant information to enterprise

What is a database management system (DBMS)?

What is a Database Management System (DBMS)?

- Includes the database & a set of programs to access it
- Define, store, retrieve data
- Convenience & Efficiency

Database Management System

- Large bodies of information
- Structures for storage
- Mechanisms for manipulating information
- Security
- Reliability
- Concurrency

Types of Applications

- Banking: transactions
- Airlines: reservations, schedules
- Universities: registration, grades
- Sales: customers, products, purchases
- Online retailers: order tracking, customized recommendations
- Manufacturing: production, inventory, orders, supply chain
- Human resources: employee records, salaries, tax deductions

University Example

- Information about:
- - Instructors
- - Students
- - Courses
- - Departments
- Storing information in files
- A number of applications to manipulate data
- Over time more files, more applications

University Example

Anything wrong with this approach?
What could go wrong?

File System as a Database

- Supported by operating system
- Permanent records stored in files
- Different programs extract records from files and add to files

File System as a Database

- Data redundancy and inconsistency
  - Duplication of information, many programs, different languages, higher storage and access cost, data inconsistency
- Difficulty in accessing data
  - New programs for each new task, lack of convenience and efficiency

File System as a Database

- Data isolation
  - Data scattered, hard to retrieve
- Integrity problems
  - Hard to enforce, programs define
- Atomicity problems
  - Failure of programs during execution

File System as a Database

- Concurrent-access anomalies
  - Multiple users, updating simultaneously
  - Inconsistent data
- Security problems
  - Restricting access to users
  - Programs added ad hoc manner

Capabilities of a DBMS

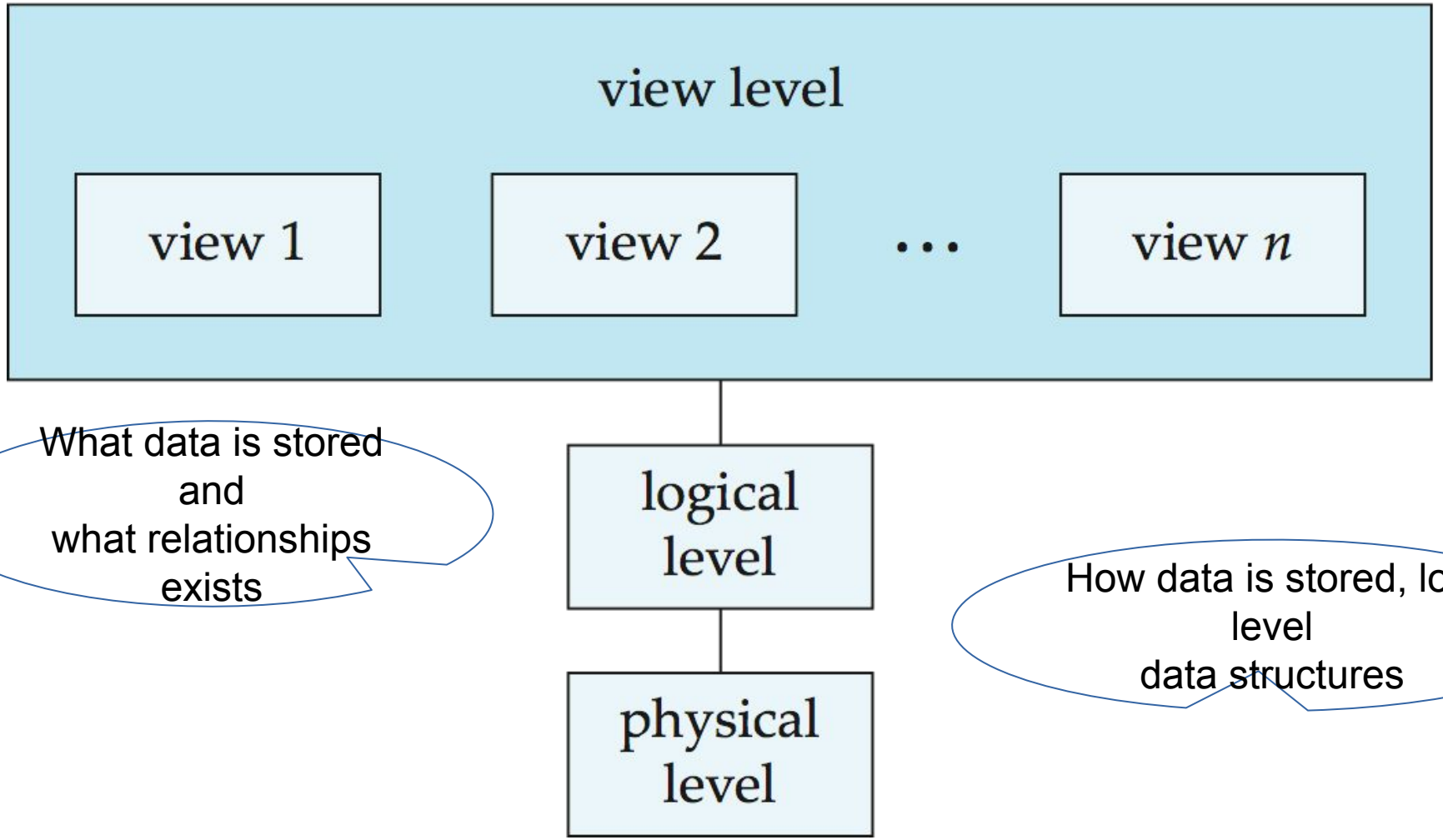
- DBMS are distinguished from other S/W systems due to their ability to:
  - Manage persistent data
  - Access large amounts of data efficiently
- Common capabilities of commercial DBMS systems
  - Support a data model, through which the user can view the data
  - Support a high-level language to define the structure of the data and access the data

Capabilities of a DBMS

- Provide correct, concurrent access to the data by many users
- Provide for the integrity (validity) of the data
- Limit access to the data by unauthorized users
- Recover from system failures without losing data

View of Data

Highest level of abstraction, reduction of complexity,
multiple views, simplifies interaction



Instances and Schemas

- **Instance** - collection information stored at that point in time
- **Schema** - overall design of the database
  - Several, partitioned according to abstraction level
  - Physical, logical, subschemas (view level)

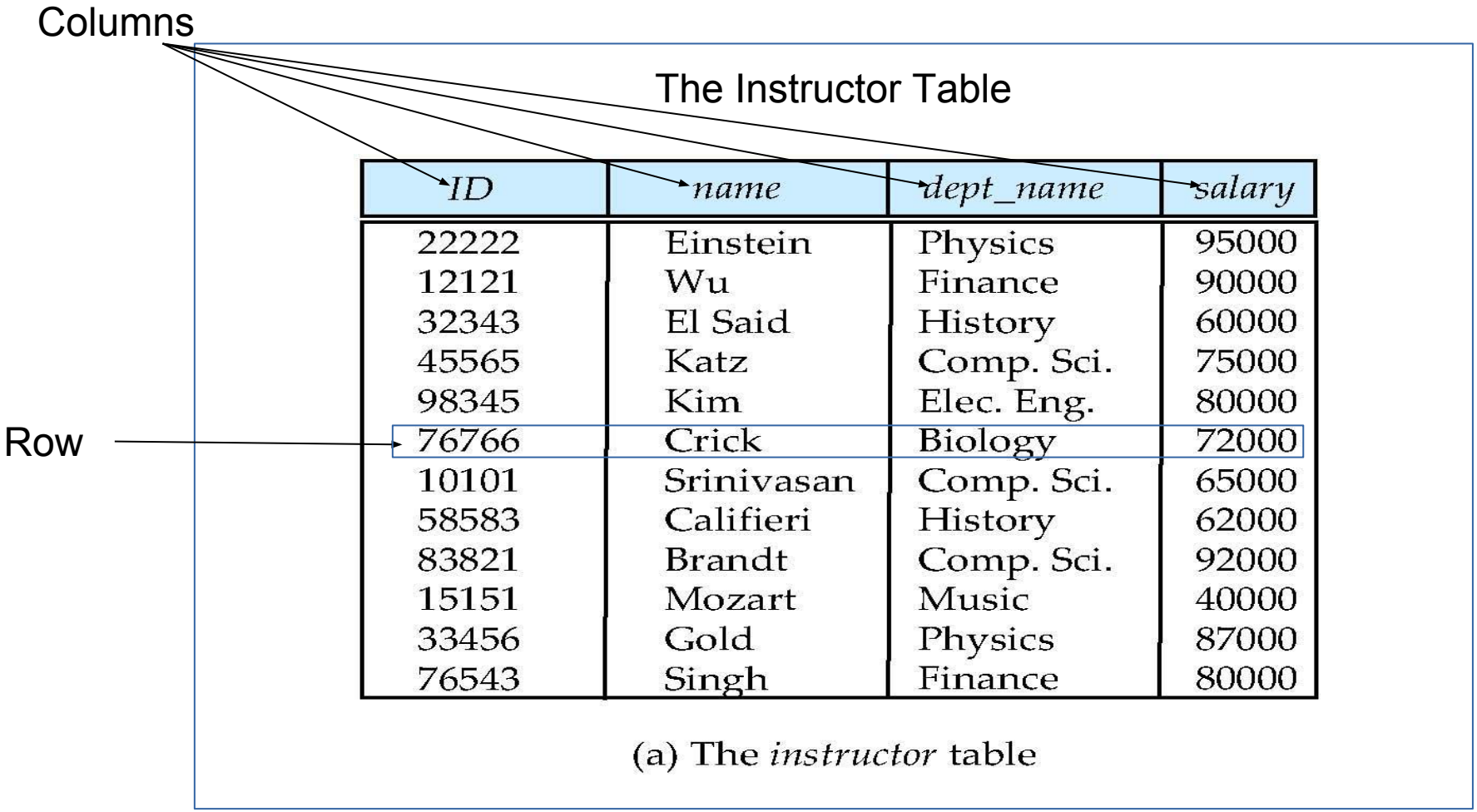
Instances and Schemas

- **Physical Data Independence -** ability to modify physical schema without changing the logical schema
  - Applications depend on logical schema
  - Interfaces between various levels should be well defined to changes in one part do not influence others
- **Logical Data Independence** - the ability to modify logical schema without changing the view schema of the database

Data Models

- **Data Model –** a collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints
  - Describes design of database at physical, logical and view levels
- Primary Data Models
  - Relational model
  - Entity-Relationship model

Relational Model



The Instructor Table

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

Relational Model

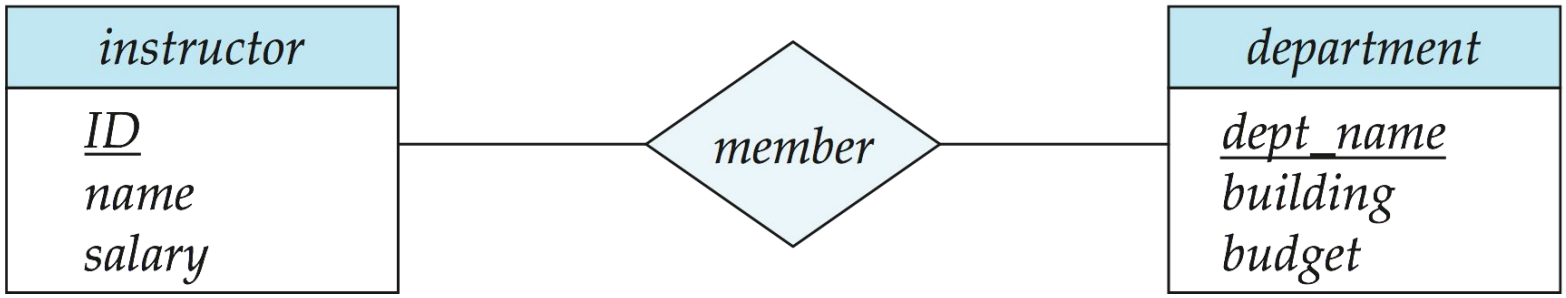
ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

dept_name	*building*	*budget*
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

(b) The *department* table

Entity-Relationship Model



Data Definition Language

- A formal (computer) language for defining a database schema
- **create table *instructor***
- ***ID*** **char(5),**
- ***name*** **varchar(20),**
- ***dept_name*** **varchar(20),**
- ***salary*** **numeric(8,2));**
- DDL compiler generates a set of tables stored in a data dictionary

Data Definition Language

- Data dictionary contains metadata (data about data)
  - Integrity constraints
  - Primary key (ID uniquely identifies instructors)
  - Referential integrity (references constraint in SQL)
- dept_name value in any instructor row must appear in department relation
  - Authorization

Data Manipulation Language

- A language for accessing and manipulating the data organized by the appropriate data model
  - Retrieval
  - Insertion
  - Deletion
  - Modifications

Data Manipulation Language

- Also known as query language
- Two classes of query languages
  - **Procedural** - user specifies what data is required and how to get those data
  - **Declarative or Nonprocedural** - user specifies what data is required without specifying how to get those data
- SQL is the most widely used query language

SQL Examples

- Find the name of the instructor with ID 22222:

```
select name
from  instructor
where instructor.ID = '22222';
```

- Find the ID and building of instructors in the Physics dept.

```
select instructor.ID, department.building
from instructor, department
where instructor.dept_name =
department.dept_name and
department.dept_name = 'Physics'
```

SQL

- Application programs generally access databases through one of:
  - Language extensions to allow embedded SQL
  - Application program interface (e.g., ODBC/JDBC) which allow SQL queries to be sent to a database
- We will learn about SQL in Chapters 3, 4 and 5

Database Design

- Mainly focused on database schema
  - Models enterprise
- Characterize data needs of users – results in specification of user requirements
- Conceptual Design Phase:
  - Choose a data model
  - Translate requirements into a conceptual schema
  - Review, fix redundancy
  - Focus on describing data and relationships

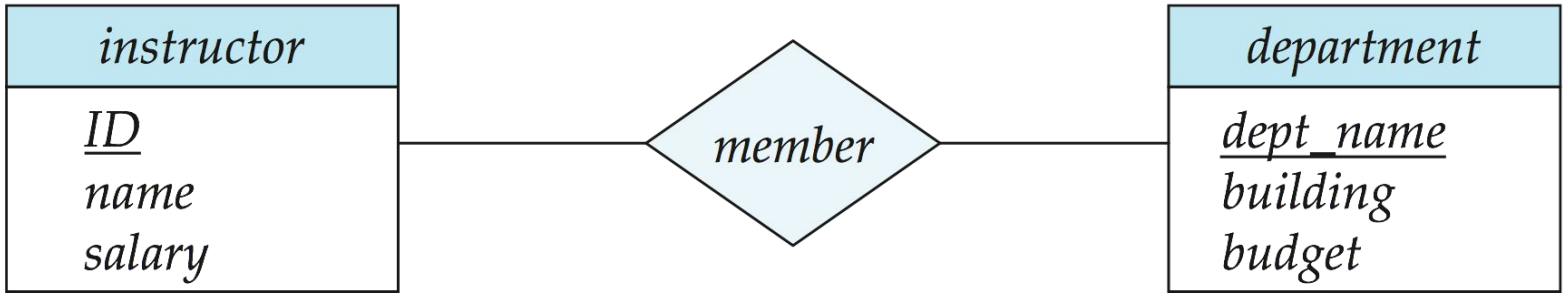
Database Design

- How does one decide which attributes should be captured and how to group the attributes to form tables?
  - Business decisions
  - Entity Relationship Model
  - Normalization

Database Design

- Logical Design Phase
  - Map high level conceptual schema to implementation of the data model
- Physical Design Phase
  - Physical features, file organization, internal storage structures

Entity-Relationship Model



- Entity: a “thing” or “object” in the enterprise that is distinguishable from other objects
  - Described by a set of attributes
- Relationship: an association among several entities

Normalization

- Generate set of relation schemas
  - Store information but remove unnecessary redundancy
- Appropriate normal form
- Use functional dependencies
- Bad design properties:
  - Repetition of information
  - Inability to represent certain information

Normalization – Bad Design Example

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

dept_name	*building*	*budget*
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

(b) The *department* table

Alternative table that combines the instructor table and the department table

ID	*name*	*salary*	*dept_name*	*building*	*budget*
22222	Einstein	95000	Physics	Watson	70000
12121	Wu	90000	Finance	Painter	120000
32343	El Said	60000	History	Painter	50000
45565	Katz	75000	Comp. Sci.	Taylor	100000
98345	Kim	80000	Elec. Eng.	Taylor	85000
76766	Crick	72000	Biology	Watson	90000
10101	Srinivasan	65000	Comp. Sci.	Taylor	100000
58583	Califieri	62000	History	Painter	50000
83821	Brandt	92000	Comp. Sci	Taylor	100000
15151	Mozart	40000	Music	Packard	80000
33456	Gold	87000	Physics	Watson	70000
76543	Singh	80000	Finance	Painter	120000

What is wrong here?

Storage Manager

- Storage manager provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system
- Responsible for interaction with the file manager
- Translates various DML statements into low-level file system commands
- Responsible for storing, retrieving, and updating data

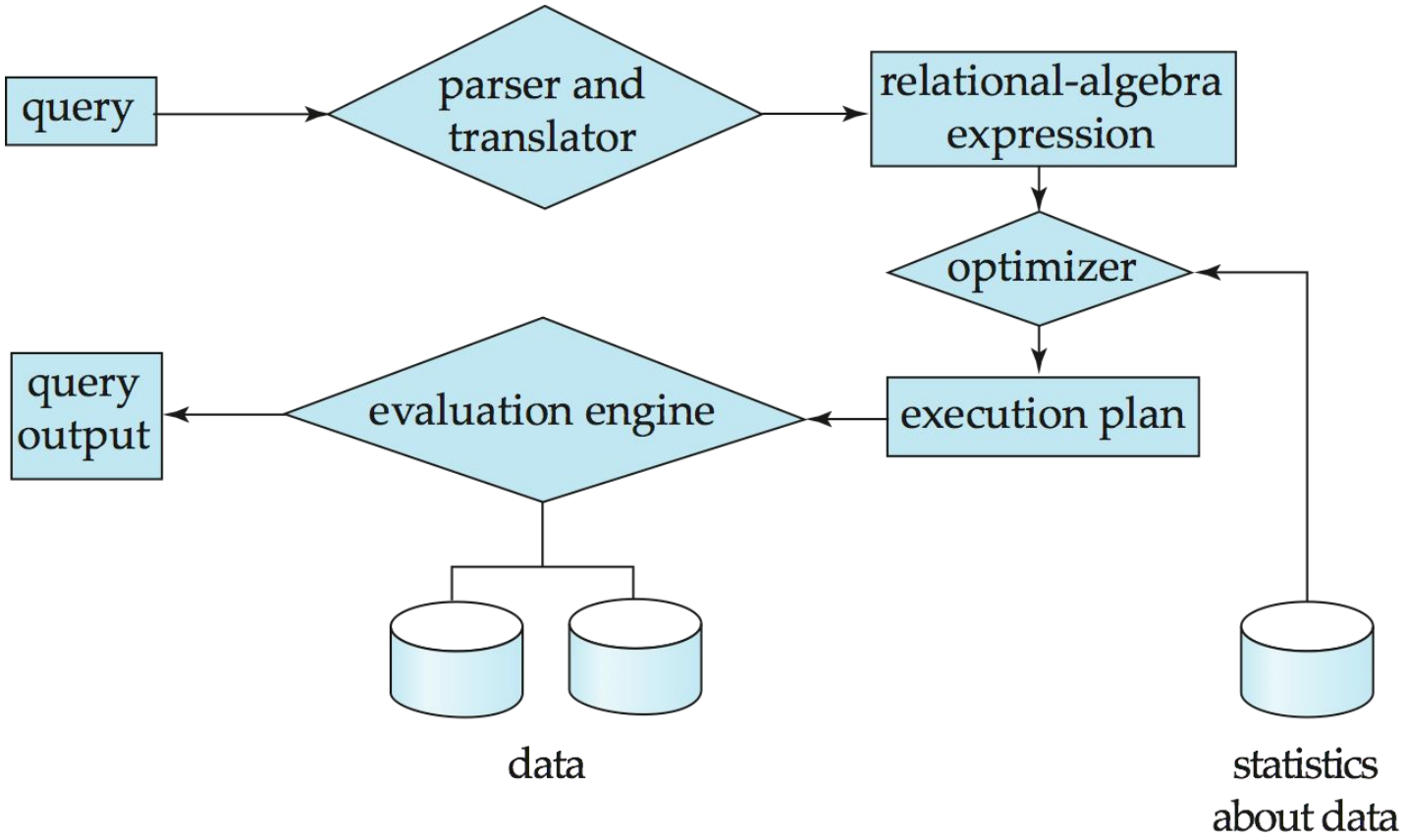
Storage Manager

- Components:
  - Authorization and integrity management
  - Transaction management
  - File management
  - Buffer management
- Implements:
  - Data files
  - Data dictionary
  - Indices

Query Processing

- **DDL interpreter** – interprets DDL statements and records definitions in the data dictionary
- **DML compiler** – translates DML statements into an evaluation plan
- **Query evaluation engine** – executes low level instructions generated by the DML compiler

Query Processing



Transaction Management

- A **transaction** is a collection of operations that performs a single logical function in a database application
- **Transaction-management component** ensures that the database remains in a consistent (correct) state despite system failures (e.g., power failures and operating system crashes) and transaction failures.

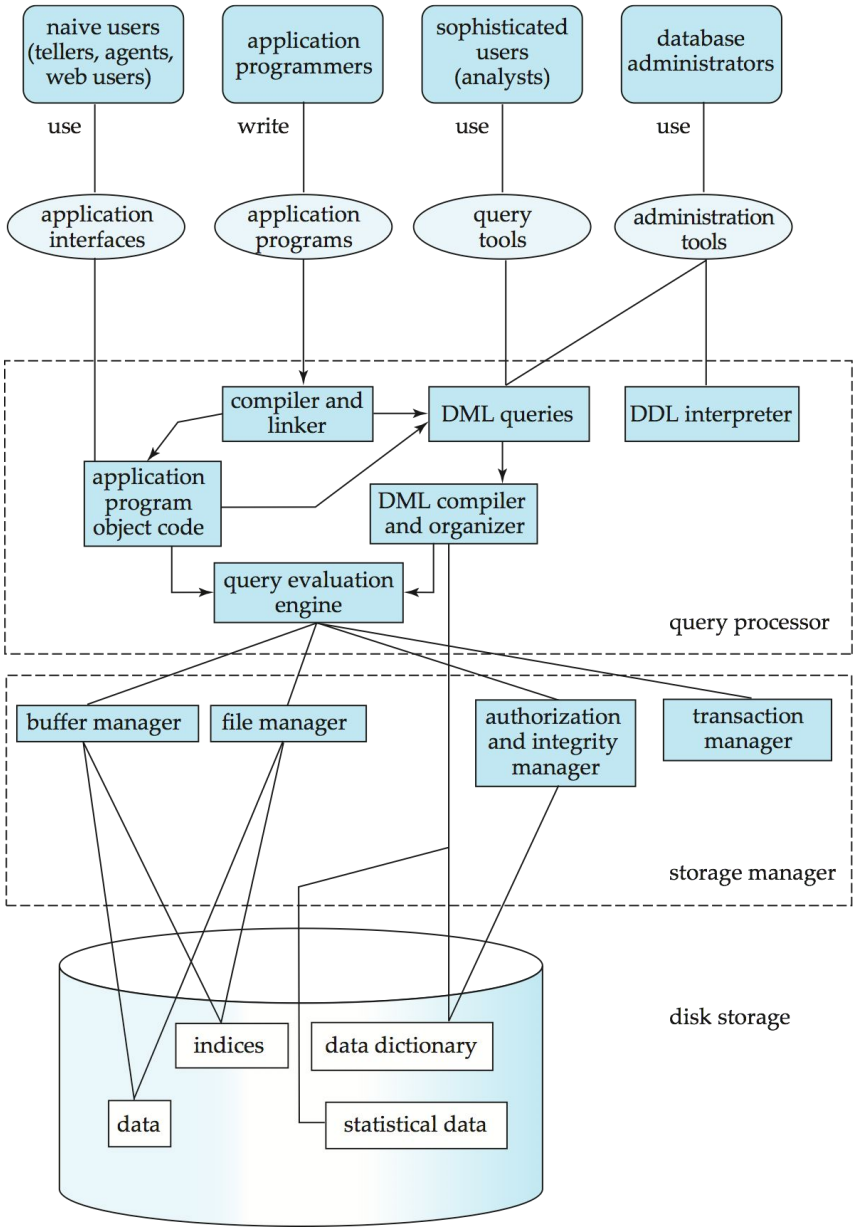
Transaction Management

- **Recovery manager** - ensures atomicity (all or none) and durability (persistence)
- **Failure recovery** – detect failures and restore database to previous state
- **Concurrency-control manager** controls the interaction among the concurrent transactions, to ensure the consistency of the database.

Database Architecture

- The architecture of a database systems is greatly influenced by the underlying computer system on which the database is running:
  - Centralized
  - Client-server
  - Parallel (multi-processor)
  - Distributed

Database Architecture



Database Users

- Users:
  - Naive – unsophisticated
  - Programmers – write applications
  - Sophisticated – interact without programs, use database query language
  - Specialized – write specialized applications

Database Administrator (DBA)

- Coordinates all the activities of the database system
- Needs good understanding of the enterprise's information resources and needs

Database Administrator (DBA)

- Duties include:
  - Schema definition
  - Storage structure and access method definition
  - Schema and physical organization modification
  - Granting user authority to access the database
  - Specifying integrity constraints
  - Acting as liaison with users
  - Monitoring performance and responding to changes in requirements