UNIT-1

- Database System Concepts
- Relation models and Database Design using ER Model

Introduction to Database Management System

- A database-management system (DBMS) is a
  - collection of interrelated data and
  - a set of programs to access those data.
- The collection of data, usually referred to as the database, contains information relevant to an enterprise.
- The primary goal of a DBMS is to provide a way to store and retrieve database information that is both convenient and efficient.
-

- Database systems are designed to manage large bodies of information.
- Management of data involves
  - defining structures for storage of information and
  - providing mechanisms for the manipulation of information
- the database system must ensure the safety of the information stored
- If data are to be shared among several users, the system must avoid possible anomalous results.

Database-System Applications

- **Enterprise Information**
  - **Sales:** For customer, product, and purchase information.
  - **Accounting:** For payments, receipts, account balances, assets, and other accounting information.
  - **Human resources:** For information about employees, salaries, payroll taxes, and benefits, and for generation of paychecks.

- **Manufacturing:** For management of the supply chain and for tracking production of items in factories, inventories of items in warehouses and stores, and orders for items

- **Banking and Finance :**
  - Banking: For customer information, accounts, loans, and banking transactions.
  - **Credit card transactions:** For purchases on credit cards and generation of monthly statements
  - **Finance:** For storing information about holdings, sales, and purchases of financial instruments such as stocks and bonds

- **Universities:** For student information, course registrations, and grades
- **Airlines:** For reservations and schedule information.
- **Telecommunication:** For keeping records of calls, texts, and data usage, generating monthly bills, maintaining balances on prepaid calling cards, and storing information about the communication networks.

- **Web-based services**
  - **Social-media:** For keeping records of users, connections between users , posts made by users, rating/like information about posts, etc.
  - **Online retailers**: For keeping records of sales data and orders as for any retailer, but also for tracking a user’s product views, search terms, etc.,
  - **Online advertisements:** For keeping records of click history to enable targeted advertisements, product suggestions, news articles, etc.

Document databases: For maintaining collections of new articles, patents, published research papers, etc.

• **Navigation systems:** For maintaining the locations of various places of interest along with the exact routes of roads, train systems, buses, etc.

Purpose of Database Systems

Keeping organizational information in a **file-processing system** has a number of major **disadvantages:**

- **Data redundancy and inconsistency:**
  - **Multiple file formats, duplication of information in many files**
  - **redundancy leads to higher storage and access cost**
  - **it may lead to data inconsistency**
- **Difficulty in accessing data:**
  - **conventional file-processing environments do not allow needed data to be retrieved in a convenient and efficient manner**
  - **Need to write new programs to carry out each new task.**

- **Data isolation:**
  - Because data are scattered in various files,
  - and files may be in different formats,
  - writing new application programs to retrieve the appropriate data is difficult
- **Integrity problems:**
  - **when new constraints are added, it is difficult to change the programs to enforce them**
  - **when constraints involve several data items from different files.**
- **Atomicity problems:**
  - **if a failure occurs, the data be restored to the consistent state that existed prior to the failure**
  - **it must happen in its entirety or not at all**

- **Concurrent-access anomalies:**
    - Concurrent access needed for performance
    - uncontrolled concurrent access can leads to inconsistency
    - Example: Two people reading a balance and updating it at the same time
- **Security problems:**
    - Hard to provide user access to some data, but not all data

Database systems offer solutions to all the above problems

View of Data

- A database system is a collection of interrelated data and a set of programs that allow users to access and modify these data.

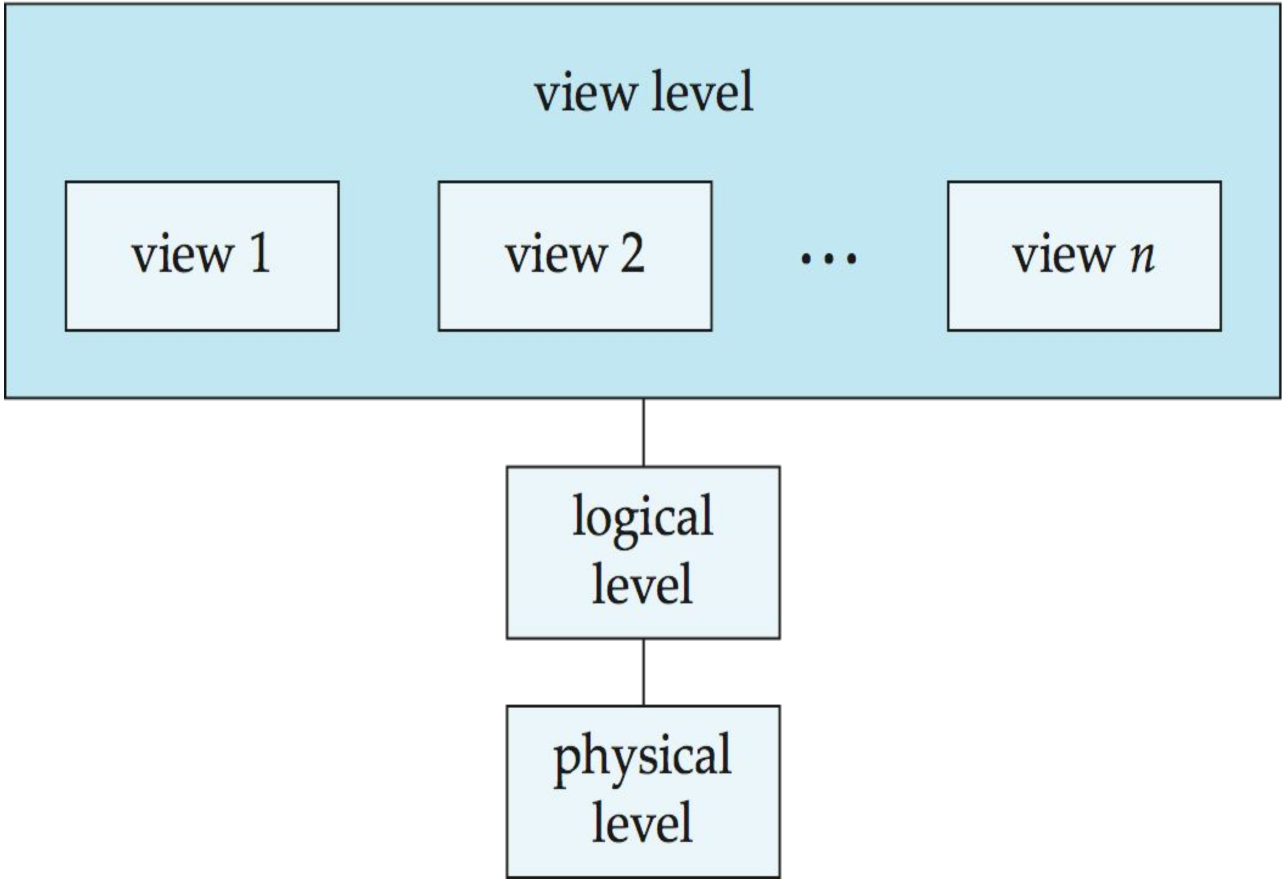
- A major purpose of a database system is to provide users with an abstract view of the data.

- the system hides certain details of how the data are stored and maintained.

Data Abstraction

- The need for efficiency has led database system developers to use complex data structures to represent data in the database.
- developers hide the complexity from users through several levels of data abstraction, to simplify users' interactions with the system
  - Physical Level
  - Logical Level
  - View Level

The Three Levels of Data Abstraction



Physical level.

- The lowest level of abstraction describes how the data are actually stored.
- The physical level describes complex low-level data structures in detail.

Logical level.

- The next-higher level of abstraction describes what data are stored in the database,
- and what relationships exist among those data.
- describes the entire database in terms of a small number of relatively simple structures.

View level.

- The highest level of abstraction describes only part of the entire database
- Many users of the database system need to access only a part of the database.
- The view level of abstraction exists to simplify their interaction with the system.
- The system may provide many views for the same database
- hide low-level implementation details from database-application developers.
- The database system allows application developers to store and retrieve data using the abstractions of the data model,
- and converts the abstract operations into operations on the low-level implementation.

- **department,** with fields dept name, building, and budget.
- **course,** with fields course id, title, dept name, and credits.
- **student,** with fields ID, name, dept name, and tot cred.

❑ At the **physical level**, an instructor, department, or student record can be described as a block of consecutive bytes

  ❑ Databases use a type of data structure called an **index** to support efficient retrieval of records

❑ At the **logical level**, each such record is described by a type definition

  ❑ **type** customer = **record**
  name : string;
  street : string;
  city : integer;
  end;

  ❑ The interrelationship of these record types is also defined at the logical level;

- At the view level,
  - users see a set of application programs that hide details of the data types.
  - several views of the database are defined, and a database user sees some or all of these views
  - views also provide a security mechanism to prevent users from accessing certain parts of the database

Instances and Schemas

- The collection of information stored in the database at a particular moment is called an instance of the database.
- The overall design of the database is called the database schema.
- Database systems have several schemas, partitioned according to the levels of abstraction.
  - The **physical schema** describes the database design at the physical level,
  - the **logical schema** (**Conceptual Schema**)describes the database design at the logical level.
- A database may also have several schemas at the view level are called **subschemas**

- **Physical Data Independence** – the ability to modify the physical schema without changing the logical schema
    - Applications depend on the logical schema
    - In general, the interfaces between the various levels and components should be well defined so that changes in some parts do not seriously influence others.

Data Models

- A collection of conceptual tools for describing
  - data,
  - data relationships,
  - data semantics,
  - and consistency constraints.
- The data models can be classified into four different categories:
  - Relational model
  - Entity-Relationship model
  - Semi structured data model (XML)
  - Object-based data models (Object-oriented and relational)

Relational Model

- The relational model uses a collection of tables to represent both data and the relationships among those data.
- Each table has multiple columns, and each column has a unique name.
- Tables are also known as relations.
- The relational model is an example of a record-based model.
  - Each table contains records of a particular type.
  - Each record type defines a fixed number of fields, or attributes.
  - The columns of the table correspond to the attributes of the record type.
- The relational data model is the most widely used data 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

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

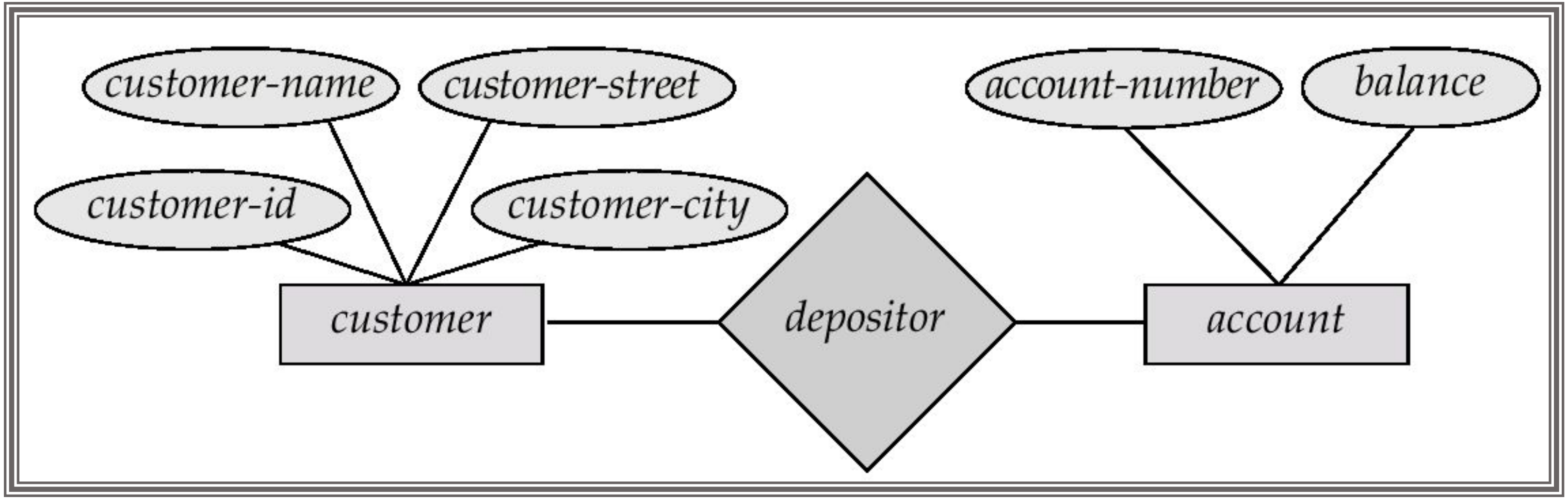
(a) The *instructor* table

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.

- The entity-relationship (E-R) data model uses a collection of basic objects, called entities, and relationships among these objects.
- An entity is a “thing” or “object” in the real world that is distinguishable from other objects.
- The entity-relationship model is widely used in database design



- **Semi-structured Data Model.**
    - The semi-structured data model permits the specification of data where individual data items of the same type may have different sets of attributes
    -
    - JSON and Extensible Markup Language (XML) are widely used semi-structured data representations
- Object-Based Model
    - Object oriented Model can be seen as extending the E-R model with notions of encapsulation, methods, and object entity

Database Languages

A database system provides

- DDL(Data Definition Language):
    - to specify the database schema
- DML(Data Manipulation Language):
    - to express database queries and updates

DDL(Data Definition Language):

- We specify a database schema by a set of definitions expressed by a special language called a data-definition language (DDL).

- The DDL is also used to specify additional properties of the data.

- We specify the storage structure and access methods used by the database system by a set of statements in a special type of DDL called a **data storage and definition language.**

- The DDL provides facilities to specify such constraints
  - **Domain Constraints**: A domain of possible values must be associated with every attribute
  - **Referential Integrity:** ensure that a value that appears in one relation for a given set of attributes also appears in a certain set of attributes in another relation
  - **Authorization:** to differentiate among the users as far as the type of access they are permitted on various data values in the database.
    - read authorization
    - Insert authorization
    - Update authorization
    - Delete authorization

- The output of the DDL is placed in the **data dictionary**, which contains metadata—that is, data about data.

- The data dictionary is considered to be a special type of table that can be accessed and updated only by the database system itself

- The database system consults the data dictionary before reading or modifying actual data.

- **The SQL Data-Definition Language**
  - SQL provides a rich DDL that allows to define tables with data types and integrity constraints.
  -
  - the following SQL DDL statement defines the department table:
    - **create table department (dept_name char (20), building char (15), budget number(12,2));**

Data-Manipulation Language(DML)

- A data-manipulation language (DML) is a language that enables users to access or manipulate data
  - Retrieval of information stored in the database.
  - insertion of new information into the database.
  - Deletion of information from the database.
  - Modification of information stored in the database

- There are basically two types of data-manipulation language:
  - **Procedural DML**s require a user to specify what data are needed and how to get those data.
  - **Declarative DML**s (also referred to as **nonprocedural DML**s) require a user to specify what data are needed without specifying how to get those data
    - the database system has to figure out an efficient means of accessing data

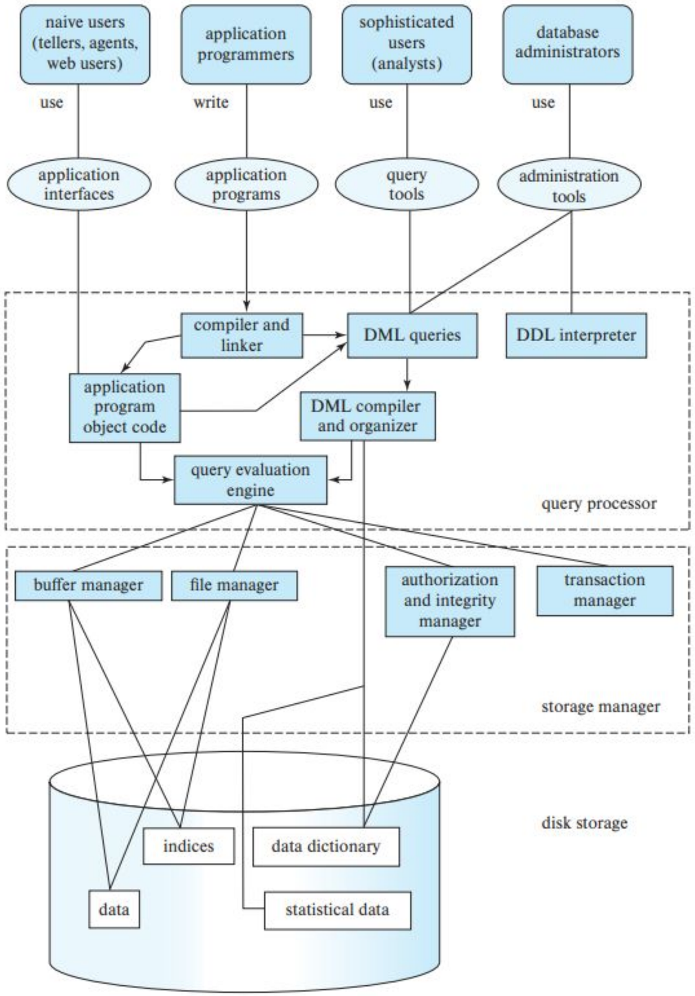
- Declarative DMLs are usually easier to learn and use than are procedural DMLs.
- A query is a statement requesting the retrieval of information.
- The portion of a DML that involves information retrieval is called a query language.
- DML also known as query language

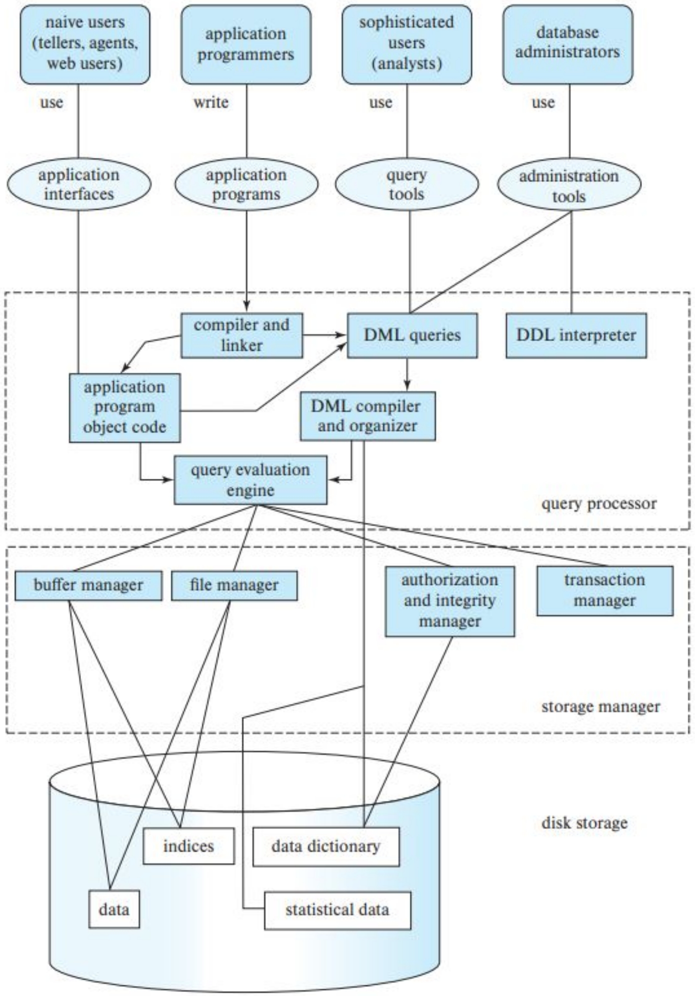
The SQL Data-Manipulation Language

- The SQL query language is nonprocedural.
- A query takes input as several tables and always returns a single table.
- Example:
  - select instructor.name from instructor where instructor.dept_name = 'History';

Database Architecture

- The architecture of a database system that runs on a centralized server machine
- different types of users interact with a database,
- different components of a database engine are connected to each other
- The centralized architecture is applicable to shared-memory server architectures




naive users (tellers, agents, web users)
application programmers
sophisticated users (analysts)
database administrators
use
write
use
use
application interfaces
application programs
query tools
administration tools
compiler and linker
DML queries
DDL interpreter
application program object code
DML compiler and organizer
query evaluation engine
query processor
buffer manager
file manager
authorization and integrity manager
transaction manager
storage manager
disk storage
indices
data dictionary
data
statistical data

- The functional components of a database system can be broadly divided into
  - storage manager,
  - query processor
  - transaction management

Storage Manager

- It provides the interface between the **low-level data** stored in the database and the **application programs and queries** submitted to the system.

- The storage manager is responsible for the interaction with the file manager.
    - The raw data are stored on the disk using the file system provided by the operating system.

- The storage manager translates the various DML statements into low-level file-system commands

- the storage manager is responsible for storing, retrieving, and updating data in the database

The storage manager components include:

- **Authorization and integrity manager**, which tests for the satisfaction of integrity constraints and checks the authority of users to access data.

- **Transaction manager**, which ensures that the database remains in a consistent (correct) state despite system failures, and that concurrent transaction executions proceed without conflicts.

- **File manager**, which manages the allocation of space on disk storage and the data structures used to represent information stored on disk.

- **Buffer manager**, which is responsible for fetching data from disk storage into main memory, and deciding what data to cache in main memory.

- The storage manager implements several data structures as part of the physical system implementation:

    - **Data files**, which store the database itself.

    - **Data dictionary,** which stores metadata about the structure of the database, in particular the schema of the database.

    - **Indices,** which can provide fast access to data items.
        - a database index provides pointers to those data items that hold a particular value.
        - For example, we could use an index to find the instructor record with a particular ID, or all instructor records with a particular name.

The Query Processor

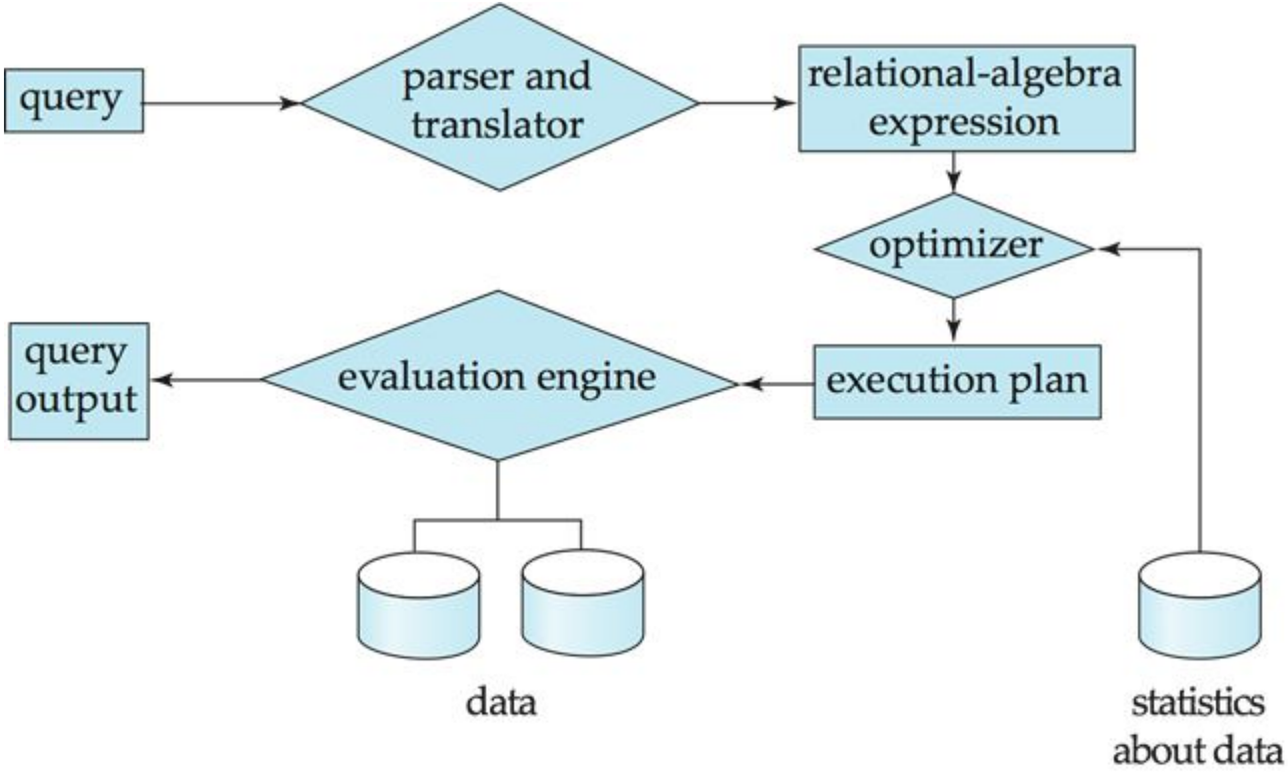
The query processor components include:

- **DDL interpreter**, which interprets DDL statements and records the definitions in the data dictionary.

- **DML compiler**, which translates DML statements in a query language into an evaluation plan consisting of low-level instructions that the query-evaluation engine understands.

- A query can usually be translated into any number of alternative evaluation plans that all give the same result
  - Equivalent expressions
  - Different algorithms for each operation
- The DML compiler also performs **query optimization**; that is, it picks the lowest cost evaluation plan from among the alternatives.
- **Query evaluation engine**, which executes low-level instructions generated by the DML compiler.

1. Parsing and translation
2. Optimization
3. Evaluation



Transaction Management

- A transaction is a collection of operations that performs a single logical function in a database application.
- Each transaction is a unit of both **atomicity** and **consistency**

- Transactions do not violate any database consistency constraints.
  - If the database was consistent when a transaction started, the database must be consistent when the transaction successfully terminates.

- ensures that the database remains in a consistent state despite system failures and transaction failures.

- **recovery manager:**Ensuring the atomicity and durability properties is the responsibility of the database system itself
    - In the absence of failures, all transactions complete successfully, and atomicity is achieved easily

- **Concurrency-control manager**
    - controls the interaction among the concurrent transactions, to ensure the consistency of the database

Database Users and Administrators

- People who work with a database can be categorized as database users or database administrators.

Database Users and User Interfaces:

- There are four different types of database-system users, differentiated based on their interaction with the system
  - Na¨ıve users
  - Application programmers
  - Sophisticated users
  - Database Administrator

Naïve users :

- Naïve users are unsophisticated users who interact with the system by using predefined user interfaces, such as web or mobile applications.

- The typical user interface for naïve users is a forms interface, where the user can fill in appropriate fields of the form.

- Naïve users may also view read reports generated from the database.

Application programmers :

- These users are computer professionals who write application programs.
- Application programmers can choose from many tools to develop user interfaces

Sophisticated users :

- These users interact with the system without writing programs.

- they form their requests either using a database query language or by using tools such as data analysis software.

- Analysts who submit queries to explore data in the database fall in this category.

Database Administrator

A person who has central control over the system is called a database administrator (DBA)

The functions of a DBA include:

- **Schema definition:**
  - The DBA creates the original database schema by executing a set of data definition statements in the DDL.

- **Storage structure and access-method definition:**
  - The DBA may specify some parameters pertaining to the physical organization of the data and the indices to be created.

- **Schema and physical-organization modification:**
  - The DBA carries out **changes to the schema and physical organization** to reflect the changing needs of the organization, or to alter the physical organization to **improve performance**

- **Granting of authorization for data access.**
  - By granting different types of authorization, the database administrator can regulate which parts of the database various users can access.

- **Routine maintenance:**
  - Periodically backing up the database onto remote servers, to prevent loss of data
  - Ensuring that enough free disk space is available for normal operations, and upgrading disk space as required.
  - Monitoring jobs running on the database and ensuring that performance is not degraded

History of Database Systems

- 1950s and early 1960s:
  - Data processing using magnetic tapes for storage
  - Tapes provided only sequential access
  - Punched cards for input
- Late 1960s and 1970s:
  - Hard disks allowed direct access to data
  - Network and hierarchical data models in widespread use
  - Ted Codd defines the relational data model
    - Would win the ACM Turing Award for this work
    - BM Research begins System R prototype
    - UC Berkeley begins Ingres prototype
- High-performance (for the era) transaction processing

- 1980s:
    - Research relational prototypes evolve into commercial systems
    - SQL becomes industrial standard
    - Parallel and distributed database systems
    - Object-oriented database systems
- 1990s:
    - Large decision support and data-mining applications
    - Large multi-terabyte data warehouses
    - Emergence of Web commerce
- Early 2000s:
    - XML and XQuery standards
    - Automated database administration
- Later 2000s:
    - Giant data storage systems
    - Google BigTable, Yahoo PNuts, Amazon, ..