What is a Database System?

- A collection of:
  - Software
  - Files
  - Abstractions
  - Views

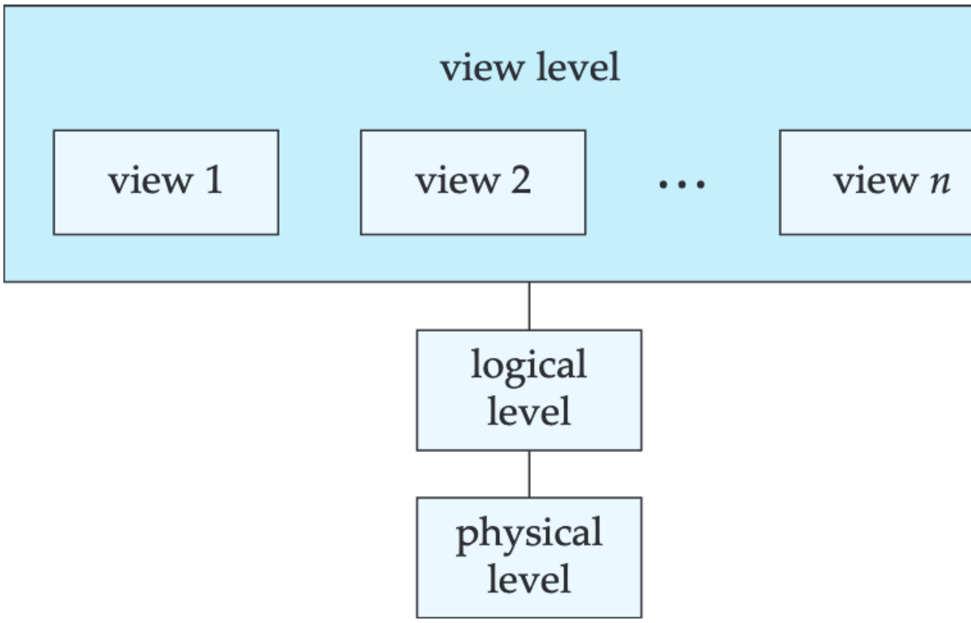
- Enables users to
  - Access data
  - Manipulate data
  - Otherwise manage data
  - Create reports

Why is Abstraction Important?

A major purpose of a database system is to provide user abstract view of the data.

- Data models
  - A collection of conceptual tools for describing d relationships, data semantics, and consistency
- Data abstraction
  - Hide the complexity of data structures to repre the database from users through several levels abstraction.

An Architecture For a Databas



Data Models

- A collection of tools for describing
  - Data
  - Data relationships
  - Data semantics
  - Data constraints

- Relational model (data is presented as tables)

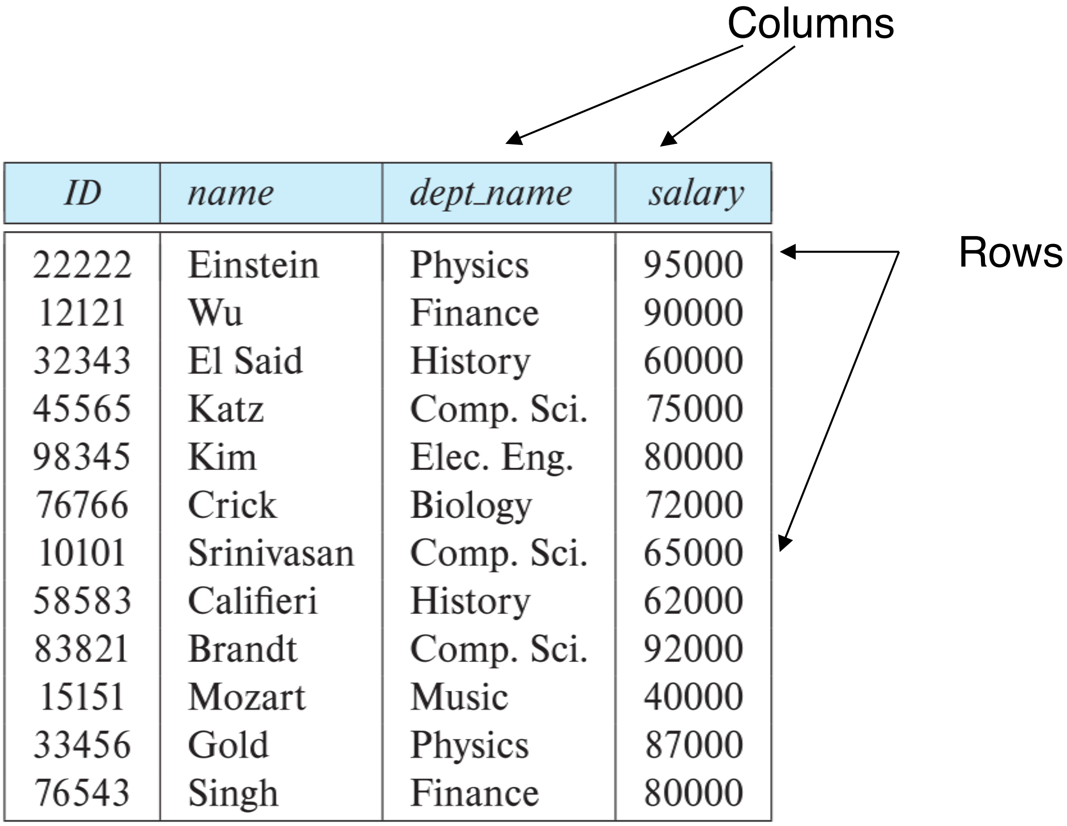
- Entity-Relationship data model (mainly for databas
  - Object-based data models (Object-oriented and Object-rela

- Semi-structured data model (XML, JSON)

- Other older models:
  - Network model
  - Hierarchical model

Relational Model

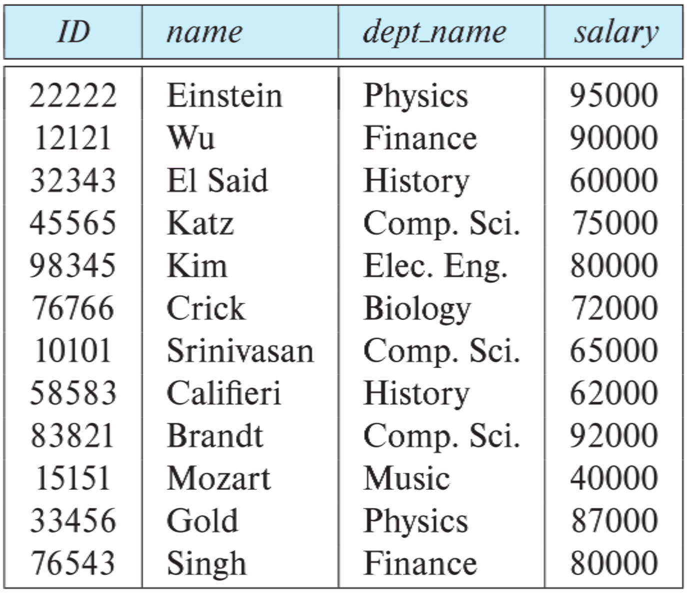
- All the data is stored in various tables.
- Example of tabular data in the relational model


Columns
Rows

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

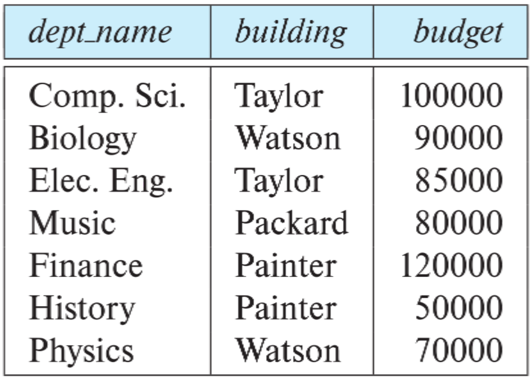
(a) The *instructor* table

A Sample Relational Database



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

Instances and Schemas

- Similar to types and variables in programming lang

- **Logical Schema** – the overall logical structure of t
  - Example: The database consists of information of customers and accounts in a bank and the re between them
    - Analogous to type information of a variable

- **Physical schema** – the overall physical structure database

- **Instance** – the actual content of the database at a point in time
  - Analogous to the value of a variable

Physical Data Independence

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

So how do we set up and access th
in a relational database?

Data Definition Language (DDL

- Specification notation for defining the database sch

  Example:

  ```
  create table instructor (
          ID          char(5),
          name        varchar(20),
          dept_name   varchar(20),
          salary      numeric(8,2))
  ```

- DDL compiler generates a set of table templates st
  data dictionary

- Data dictionary contains metadata (i.e., data about
  - Database schema
  - Integrity constraints
    - Primary key (ID uniquely identifies instructo
  - Authorization
    - Who can access what

Data Manipulation Language (DM

- Language for accessing and updating the data organize appropriate data model
  - DML also known as query language

- There are basically two types of data-manipulation lang
  - **Procedural DML** -- require a user to specify what data ar how to get those data.
  - **Declarative DML** -- require a user to specify what data a specifying how to get those data.

- Declarative DMLs are usually easier to learn and use th procedural DMLs.

- Declarative DMLs are also referred to as non-procedura

- The portion of a DML that involves information retrieval **query** language.

SQL Query Language

- SQL query language is nonprocedural. A query takes tables as input and always returns a single table.

- Example to find all instructors in Comp. Sci. dept

```
select name
from instructor
where dept_name = 'Comp. Sci.'
```

- To be able to compute complex functions SQL is usua in some higher-level language

- Application programs generally access databases thr
  - Language extensions to allow embedded SQL
  - Application program interface (e.g., ODBC/JDBC) SQL queries to be sent to a database

Database Access from Application P

- SQL does not support actions such as input from u to displays, or communication over the network.

- Such computations and actions must be written in **language**, such as C/C++, Java or Python, with en SQL queries that access the data in the database.

- **Application programs** -- are programs that are us interact with the database in this fashion.

Designing a Relational Databas

Database Design

The process of designing the general structure of the dat

- Logical Design – Deciding on the database schema.
  - Business decision – What attributes should we rec database?
  - Computer Science decision – What relation schem have and how should the attributes be distributed various relation schemas?

- Physical Design – Deciding on the physical layout of t

The Database Engine

Database Engine

- A database system is partitioned into modules that each of the responsibilities of the overall system.

- The functional components of a database system c divided into
  - The storage manager,
  - The query processor component,
  - The transaction management component.

Storage Manager

- A program module that provides the interface betw level data stored in the database and the applicatio and queries submitted to the system.

- The storage manager is responsible to the followin
  - Interaction with the OS file manager
  - Efficient storing, retrieving and updating of data

- The storage manager components include:
  - Authorization and integrity manager
  - Transaction manager
  - File manager
  - Buffer manager

Storage Manager (Cont.)

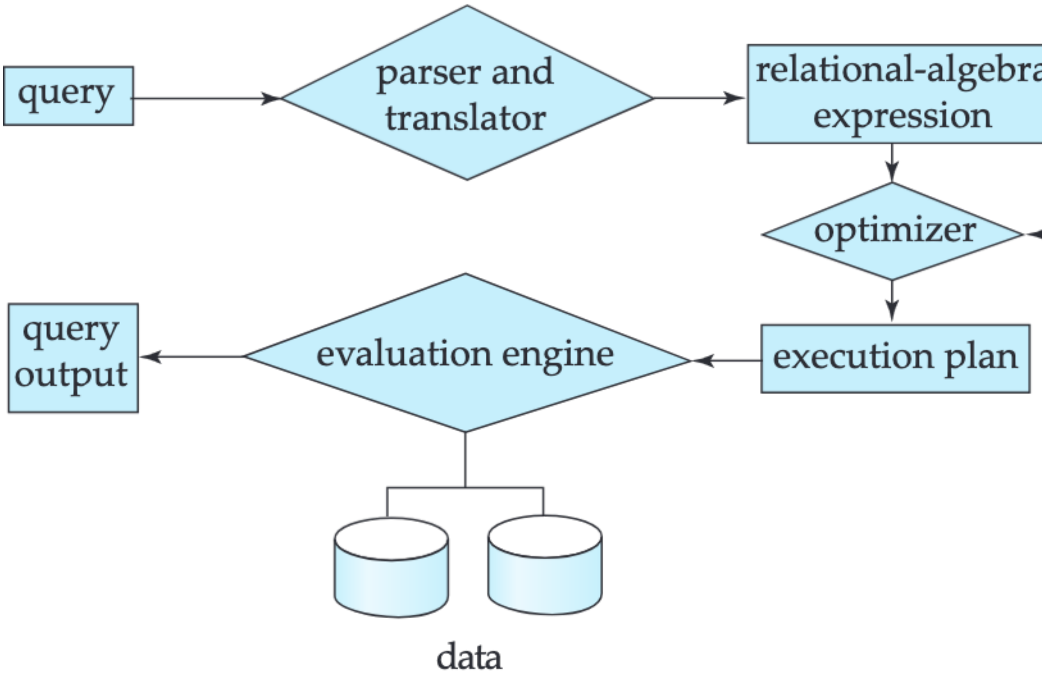
- The storage manager implements several data stru
  part of the physical system implementation:
  - Data files -- store the database itself
  - Data dictionary -- stores metadata about the st
    the database, in particular the schema of the da
  - Indices -- can provide fast access to data items
    database index provides pointers to those data
    hold a particular value.

Query Processor

- The query processor components include:
  - DDL interpreter -- interprets DDL statements a the definitions in the data dictionary.
  - DML compiler -- translates DML statements in a language into an evaluation plan consisting of l instructions that the query evaluation engine ur
    - The DML compiler performs query optimiza it picks the lowest cost evaluation plan from various alternatives.
  - Query evaluation engine -- executes low-level i generated by the DML compiler.

Query Processing

1. Parsing and translation
2. Optimization
3. Evaluation



Transaction Management

- A **transaction** is a collection of operations that per
  single logical function in a database application

- **Transaction-management component** ensures t
  database remains in a **consistent** (correct) state d
  system failures (e.g., power failures and operating
  crashes) and transaction failures.

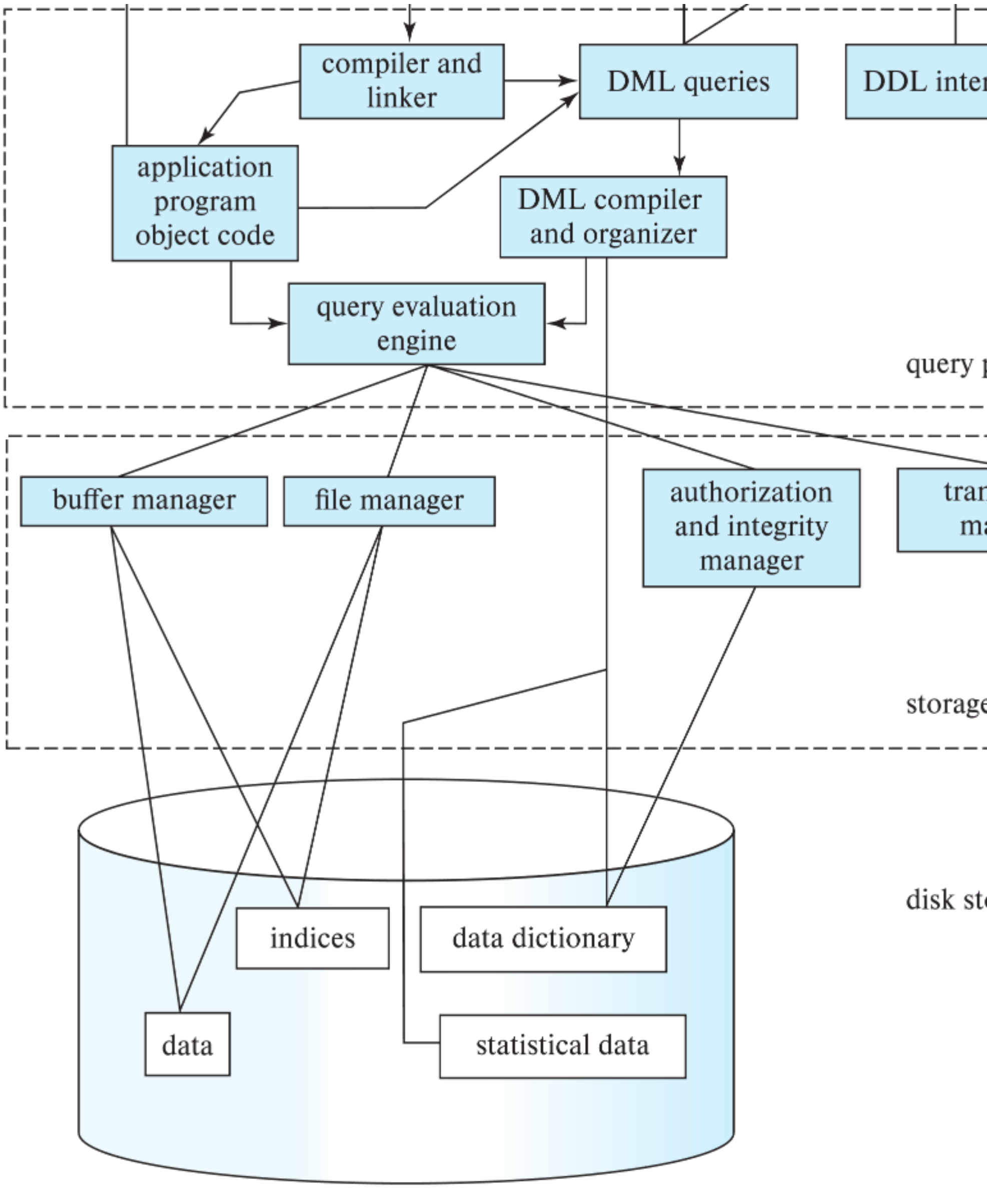
- **Concurrency-control manager** controls the intera
  among the concurrent transactions (**isolation**), to e
  consistency of the database.

A	Atomic
C	Consistent
I	Isolated
D	Durable

Database Architecture

- Centralized databases
  - One to a few cores, shared memory
- Client-server,
  - One server machine executes work on behalf o client machines.
- Parallel databases
  - Many core shared memory
  - Shared disk
  - Shared nothing
- Distributed databases
  - Geographical distribution
  - Schema/data heterogeneity

Database Architecture (Centralized/Shared-Memor

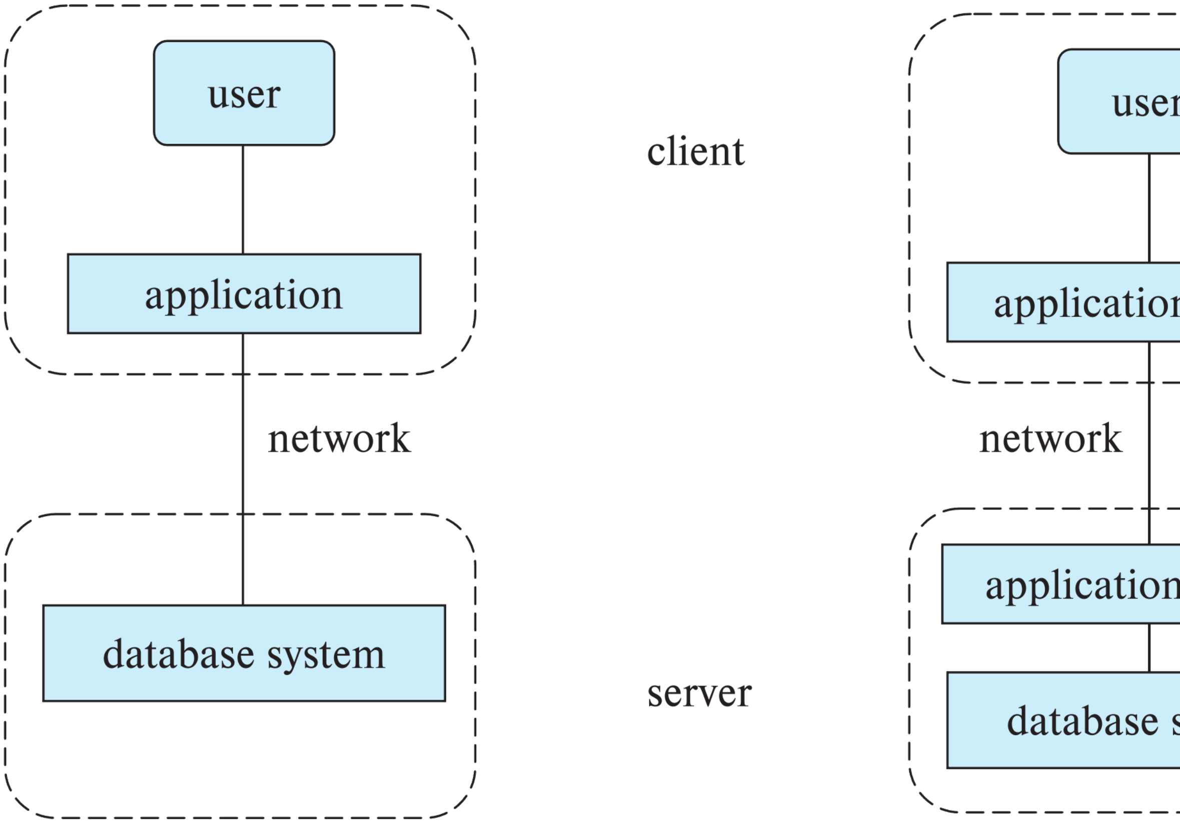


Database Applications

Database applications are usually partitioned into two or

- Two-tier architecture -- the application resides at the where it invokes database system functionality at the
- Three-tier architecture -- the client machine acts as a does not contain any direct database calls.
  - The client end communicates with an application s through a forms interface.
  - The application server in turn communicates with system to access data.

Two-tier and three-tier architectu



user

application

network

database system

client

server

user

application

network

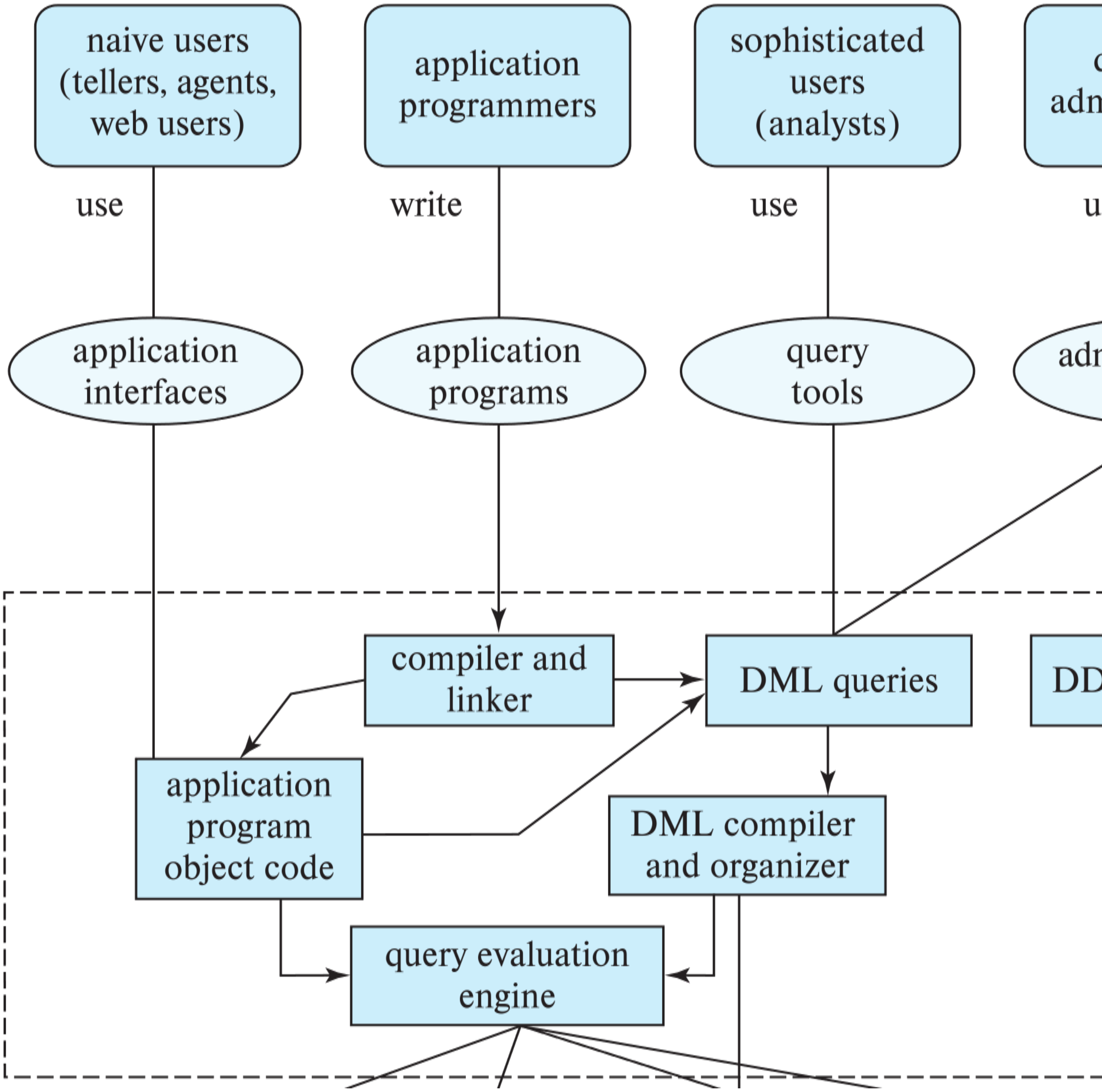
application

database s

(a) Two-tier architecture

(b) Three-tier ar

Database Users May See the Data Di



Database Administrator

A person who has central control over the system is calle **administrator (DBA).** Functions of a DBA include:

- Schema definition
- Storage structure and access-method definition
- Schema and physical-organization modification
- Granting of authorization for data access
- Routine maintenance
- Periodically backing up the database
- Ensuring that enough free disk space is available for operations, and upgrading disk space as required
- Monitoring jobs running on the database

History of Database Systems

- 1950s and early 1960s:
  - Data processing using magnetic tapes for stora
    - 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 widesp
  - Ted Codd defines the relational data model
    - Would win the ACM Turing Award for this w
    - IBM Research begins System R prototype
    - UC Berkeley (Michael Stonebraker) begins
    - Oracle releases first commercial relational d
  - High-performance (for the era) transaction proc

History of Database Systems (Co

- 1980s:
  - Research relational prototypes evolve into com systems
    - SQL becomes industrial standard
  - Parallel and distributed database systems
    - Wisconsin, IBM, Teradata
  - Object-oriented database systems

- 1990s:
  - Large decision support and data-mining applica
  - Large multi-terabyte data warehouses
  - Emergence of Web commerce

History of Database Systems (Co

- 2000s
  - Big data storage systems
    - Google BigTable, Yahoo PNuts, Amazon,
    - “NoSQL” systems.
  - Big data analysis: beyond SQL
    - Map reduce and friends
- 2010s
  - SQL reloaded
    - SQL front end to Map Reduce systems
    - Massively parallel database systems
    - Multi-core main-memory databases