CSE 233

Database System Overview

Data Management

An evolving, expanding field:

- Classical stand-alone databases (Oracle, DB2, SQL Server)
- Computer science is becoming data-centric:

  web knowledge harvesting, crowd sourcing, cloud computing, scientific databases, networks, data mining, streaming sensor monitoring, social networks, bioinformatics, geographic information systems, digital libraries, data-driven business processes

- Classical database concepts and algorithms continue to provide the core technology

What is a database?

- Persistent data
- Query and update language for accessing and modifying data
- Query optimization
- Transactions and concurrency control

What kind of data?

Emphasis: many instances of similarly structured data

Examples:

- Airline reservations: database (large set of similar records)
- Computerized library: information retrieval
- Medication advisor: expert system

Top Level Goals of a Database System

- Provide users with a meaning-based view of data
  - shield from irrelevant detail → abstract view
- Support operations on data
  - queries, updates
- Provide data control
  - integrity, protection
  - concurrency, recovery

Database System

- Tailored to specific application

Database Management System

- Generalized DB system
  - used in variety of application environments
  - common approach to
    - data organization
    - data storage
    - data access
    - data control
  - e.g. Ingres/Postgres, DB2, Oracle, SQL Server, MySQL, etc.

Levels of Abstraction

- **Logical level:** describes data stored in database in terms close to the application

  type *customer* = **record**

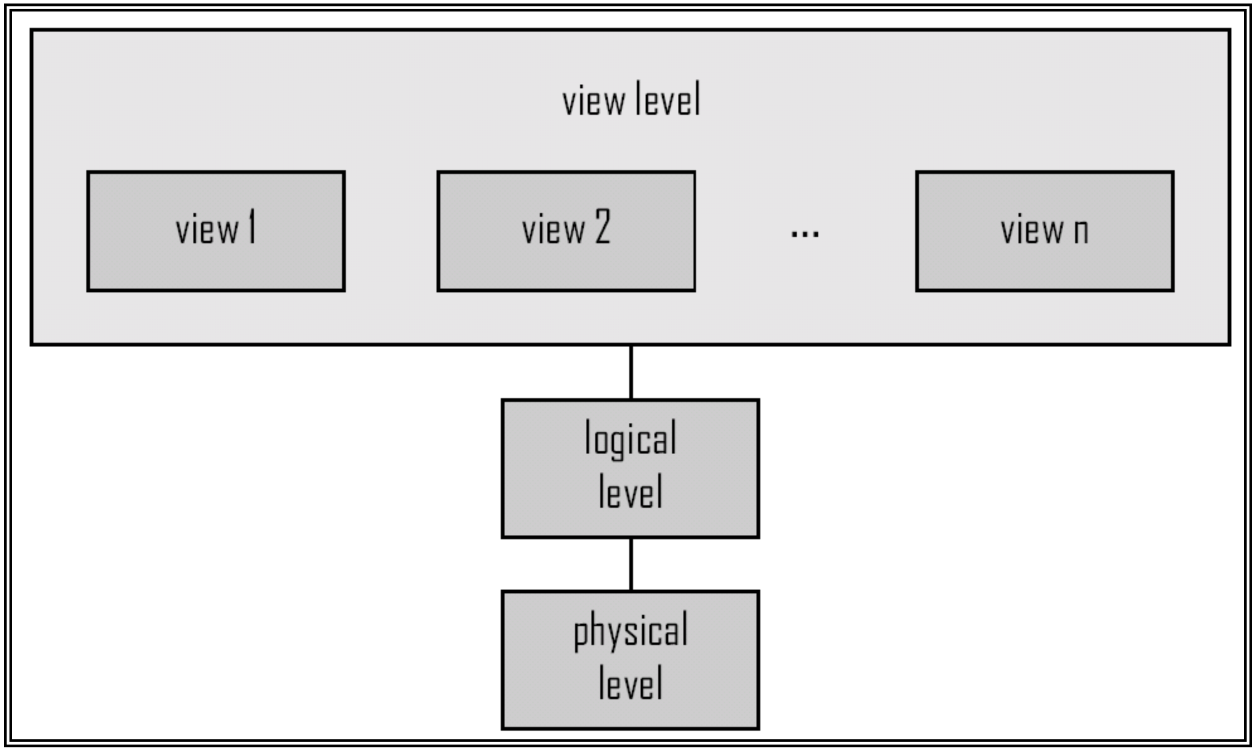
  customer_id : string;
  customer_name : string;
  customer_street : string;
  customer_city : integer;

  end;

- **Physical level:** describes how the data is stored.

- **View level:** customized, restructured information. Views can also hide information (such as an employee's salary) for security purposes.

Basic Architecture of a Database System



Data Independence – logical and physical levels are independent

Data Models

- A collection of concepts and tools for describing the data
  relationships, semantics, constraints…

  +

- A language for querying and modifying the data

- Relational model
- Entity-Relationship data model (mainly for database design, no query language)
- Object-based data models (Object-oriented and Object-relational)
- Semi-structured data model (XML)
- Other older models:
  - Network model
  - Hierarchical model

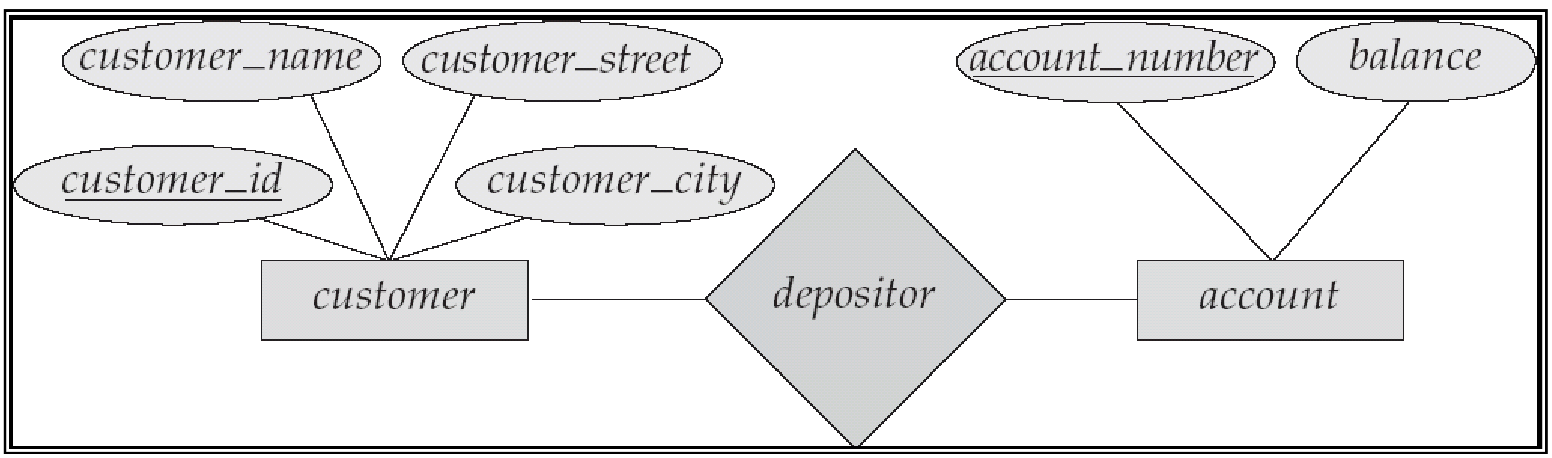
Schemas and Instances

Similar to types and values of variables in programming languages

- Schema – the logical structure of the database
  - Example: The database consists of information about a set of customers and accounts and the relationship between them)
  - Analogous to type of a variable in a program
- Instance – the actual content of the database at a particular point in time
  - Analogous to the value of a variable

Example: Entity-Relationship Model

- Models an application as a collection of *entities* and *relationships*
  - 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
- Represented diagrammatically by an *entity-relationship diagram:*



Example: Relational Model

customer_id	*customer_name*	*customer_street*	*customer_city*

(a) The *customer* table

account_number	*balance*

(b) The *account* table

customer_id	*account_number*

(c) The *depositor* table

Schema

Example: Relational Model

customer_id	customer_name	customer_street	customer_city
192-83-7465	Johnson	12 Alma St.	Palo Alto
677-89-9011	Hayes	3 Main St.	Harrison
182-73-6091	Turner	123 Putnam Ave.	Stamford
321-12-3123	Jones	100 Main St.	Harrison
336-66-9999	Lindsay	175 Park Ave.	Pittsfield
019-28-3746	Smith	72 North St.	Rye

(a) The *customer* table

account_number	balance
A-101	500
A-215	700
A-102	400
A-305	350
A-201	900
A-217	750
A-222	700

(b) The *account* table

customer_id	account_number
192-83-7465	A-101
192-83-7465	A-201
019-28-3746	A-215
677-89-9011	A-102
182-73-6091	A-305
321-12-3123	A-217
336-66-9999	A-222
019-28-3746	A-201

(c) The *depositor* table

Instance

Data Definition Language (DDL)

- Specification language for defining the database schema

  Example: **create table** *account* (
  account-number **char**(10),
  balance **integer**)

- DDL compiler generates a set of tables stored in a *data dictionary*
- Data dictionary contains metadata (i.e., data about data)
  - Database schema
  - Integrity constraints
    - Domain constraints
    - Referential integrity (**references** constraint in SQL)
    - Assertions
  - Authorization information

Data Manipulation Language (DML)

- Language for accessing and modifying data

  DML also known as query/update language
- Two classes of languages
  - Procedural – user specifies what data is required and how to get that data
  - Declarative (nonprocedural) – user specifies what data is required

    without specifying how to get it
- SQL is the most widely used query language

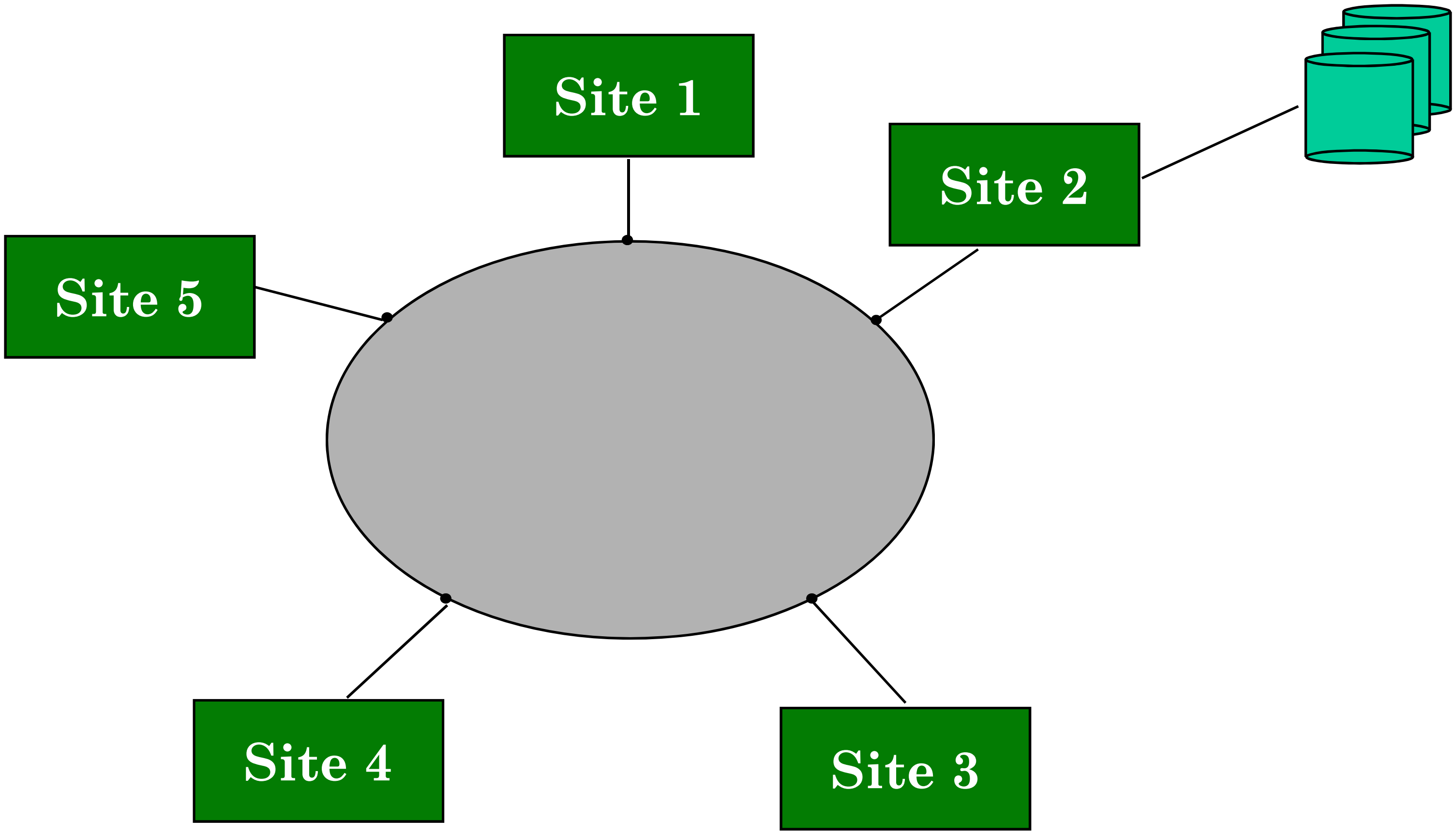
  primarily declarative

Database Architecture

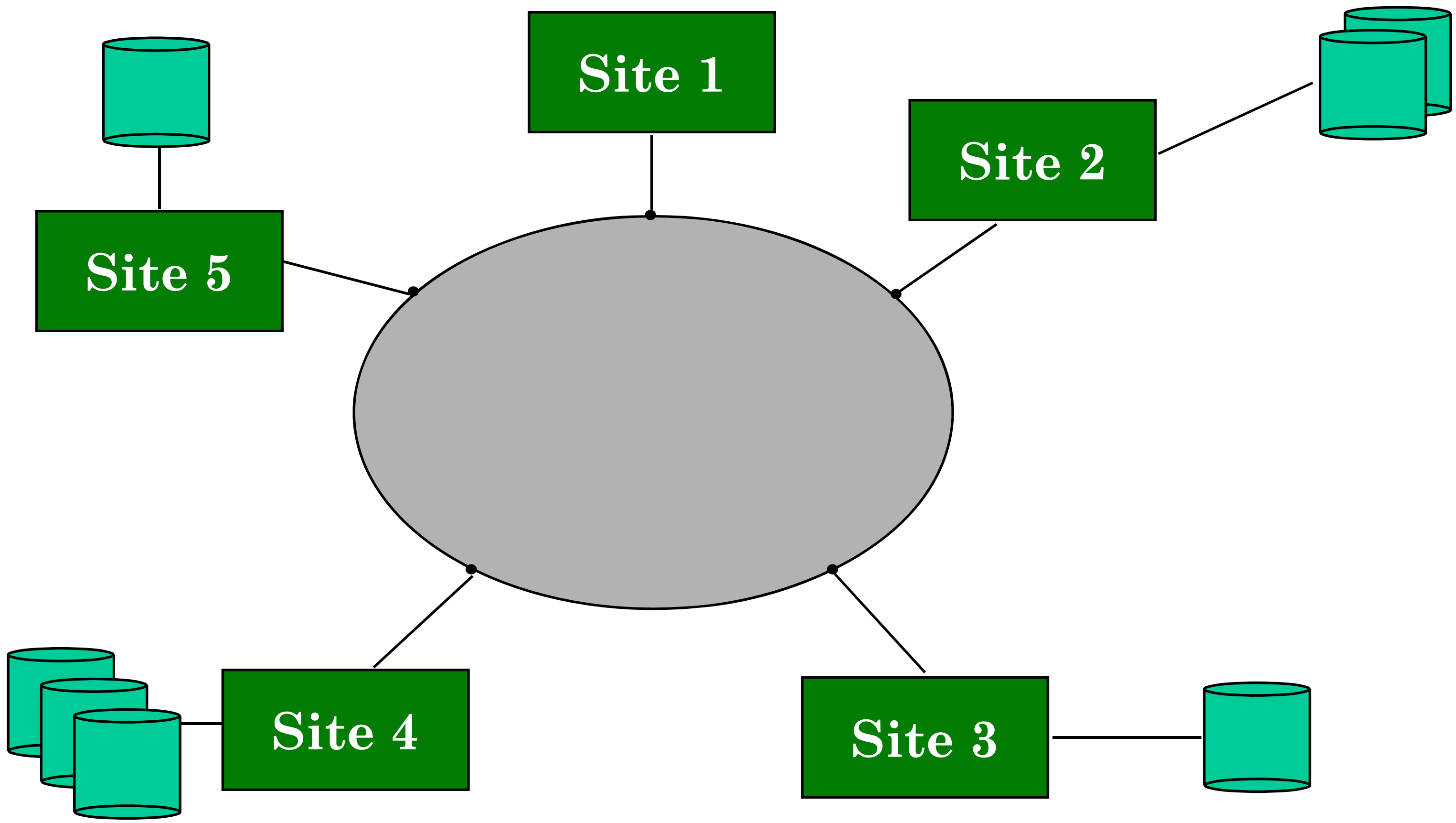
Different architectures for different settings:

- Centralized
- Parallel (multi-processor) cloud computing/map-reduce
- Client-server
- Distributed

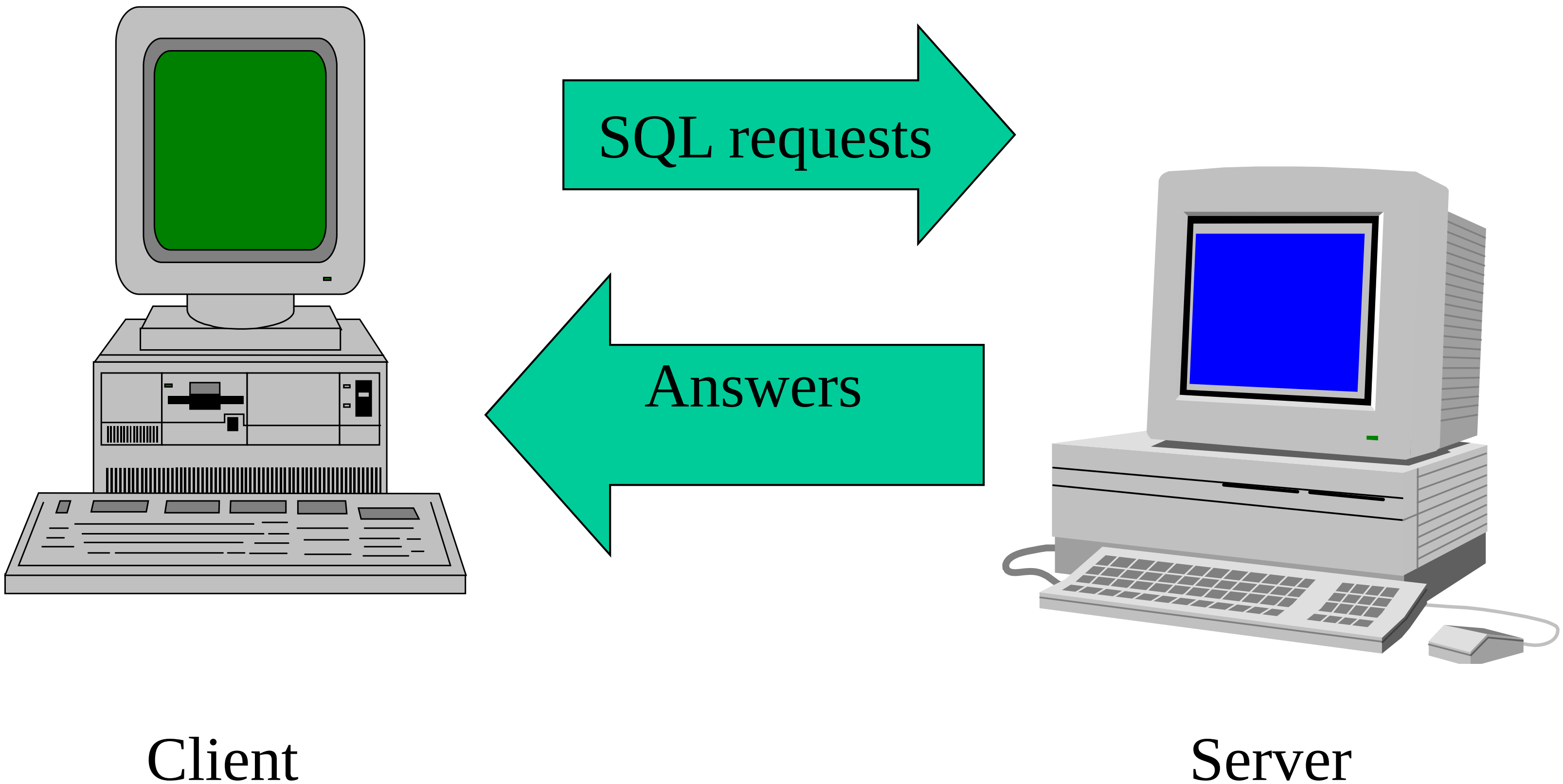
Centralized DBMS



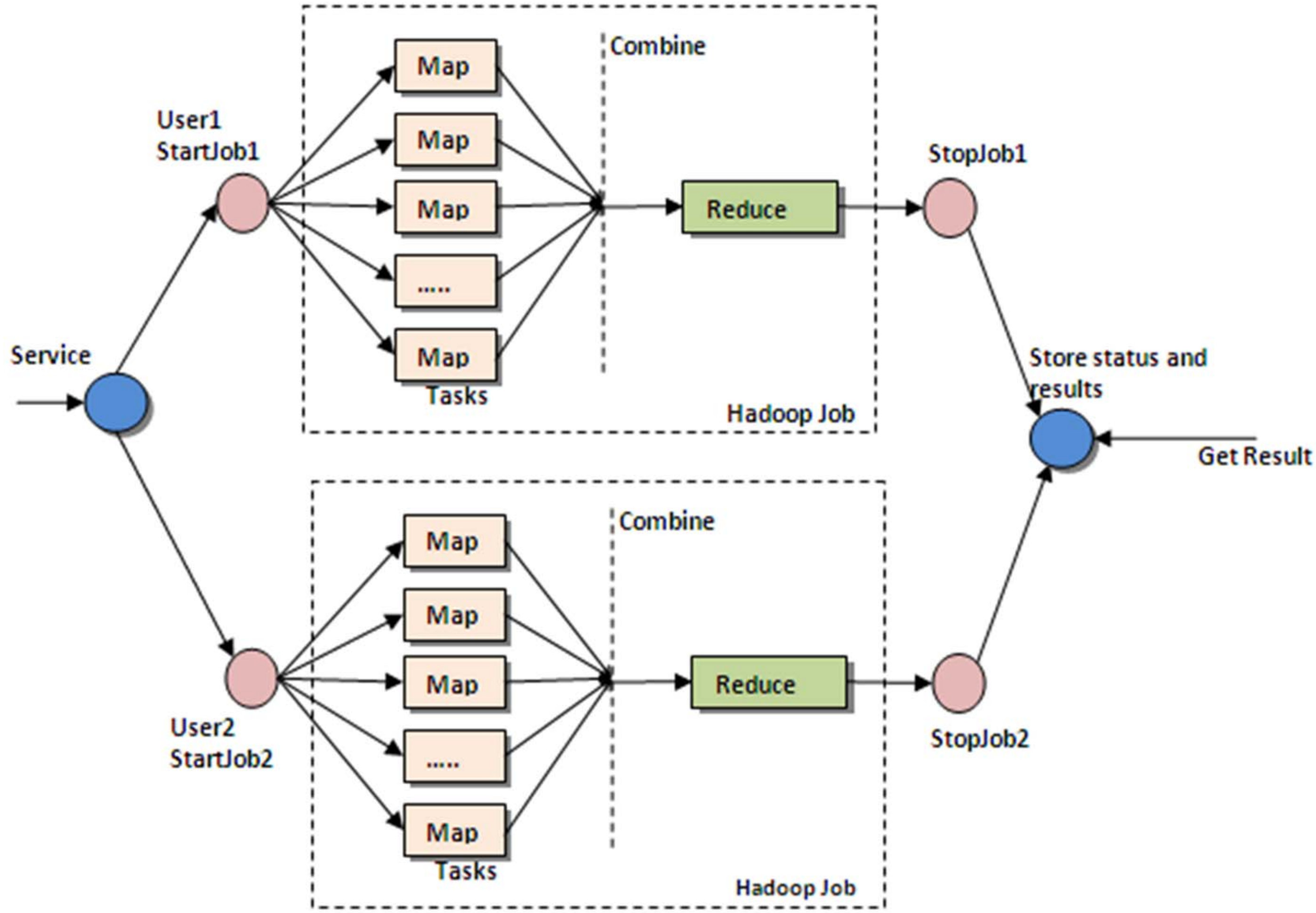
Distributed DBMS



Client/Server DBMS



Map-reduce


Service
User1
StartJob1
Map
Map
Map
.....
Map
Tasks
Combine
Reduce
Hadoop Job
StopJob1
Store status and
results
Get Result
User2
StartJob2
Map
Map
Map
.....
Map
Tasks
Combine
Reduce
Hadoop Job
StopJob2

Core database issues

- Data models, query languages
- Database design
- Query processing
- Storage management
- Transaction management
- Concurrency control

Beyond the Core

- Deductive databases
- Temporal databases
- Multimedia databases
- Geographic information systems
- Data warehouses
- Real-time and active databases
- XML databases
- Database-driven Web applications/services
- Data analytics (aka Big Data)

Databases at UCSD

- Prof. Alin Deutsch
- Prof. Arun Kumar
- Prof. Yannis Papakonstantinou
- Prof. Victor Vianu

Database group Web site: http://db.ucsd.edu/

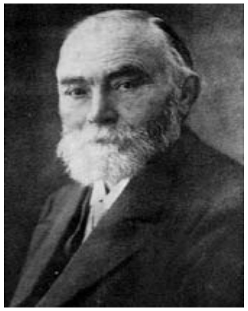
papers, seminars, bragging….

- Intersections with other CSE groups
  - storage
  - multimedia
  - machine learning
  - IR/ data mining
  - networks

Database Theory

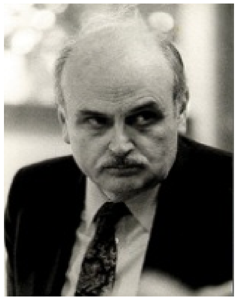
CSE 233

Relational db: theory ⟶ practice



Frege: FO logic

Tarski: algebra for FO



Codd: relational databases

Databases: implemented logic!

- FO lies at the core of modern database systems

“Databases = FO on every desk!”

- Relational query languages are based on FO:

SQL, QBE

- More powerful query languages (all the way to XML) are based on extensions of FO

Why is FO so successful as a query language?

- easy to use syntactic variants

  SQL, QBE

- efficient implementation via relational algebra

  amenable to analysis and simplification

- potential for perfect scaling to large databases

  very fast response can be achieved

  using parallel processing

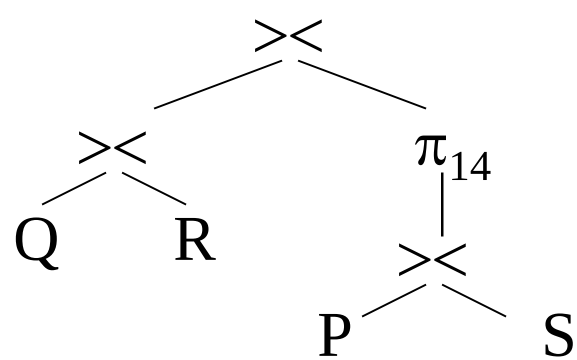
Journey of a Query

SQL ~ FO — select … from … where

Relational Algebra — $\pi_{13}(P >< Q) >< \ldots$

Query Rewriting — $\pi_{14}(P >< S) >< Q >< R$

Query Execution Plan



Execution

Physical Level

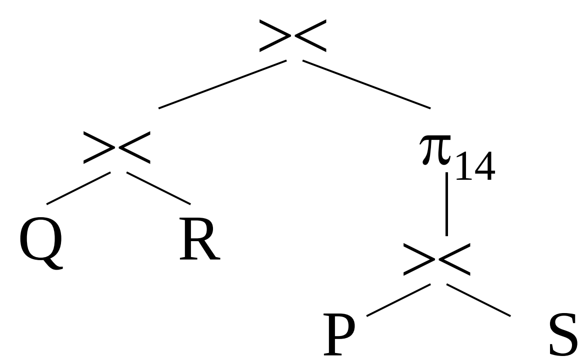
Journey of a Query

SQL ~ FO — select … from … where

Relational Algebra — $\pi_{13}(P >< Q) >< \ldots$

Query Rewriting — $\pi_{14}(P >< S) >< Q >< R$

Query Execution Plan



Execution

Physical Level

Most spectacular: theoretical potential for perfect scaling!

- perfect scaling: given sufficient resources, performance does not degrade as the database becomes larger
- key: parallel processing
- cost: number of processors polynomial in the size of the database (FO $\subseteq$ AC_0)
- role of algebra: operations highlight parallelism

Outline

- FO (aka CALC), relational algebra
- Static analysis for query processing
- Dependency theory
- Extending FO with recursion: Datalog and fixpoint logics
- Expressiveness and complexity
  - Ehrenfeucht-Fraisse games, 0/1 laws
  - The quest for a language for PTIME
- Highly expressive languages

Other topics (if time)

- Incomplete information
- Complex objects
- Selected research topics