CSE 132A

Database Systems Principles

Prof. Victor Vianu

Logistics

- Lectures: in person (WLH 2005), podcast
- Attendance optional but strongly recommended
- Slides posted prior to lecture
- Weekly discussion section: in person and podcast
- TA office hours and instructor office hours:
  in person and by zoom
- Discussion board (Piazza)
- Exams: in person only!
- Everything will be posted on Canvas or Piazza

Data Management

An evolving, expanding field:

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

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

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

CSE132A: Database Systems Principles

- Core concepts and techniques in database systems
- Databases from the viewpoint of user and designer
- A lot of SQL, but also peeks under the hood:
  query processing, schema design, transactions
  and concurrency control
- Some basic theory: formal languages underlying SQL
  relational algebra and relational calculus
- Basic background for follow-up courses

  132B: Database applications (A. Deutsch)

  132C: Database implementation (A. Kumar)

  135: Online analytics applications (Y. Papakonstantinou)

  190: Beyond relational databases (A. Deutsch)

Requirements

- Two Gradiance SQL Labs and 3 written homeworks (13%)
- Two programming assignments (SQL and JDBC) (27%)
- Midterm (30%)
- Final (30%)

Academic Integrity

Everyone taking the class is assumed familiar with the Integrity of Scholarship policy posted on the class Web site

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, security
  - concurrency, recovery

Levels of Abstraction

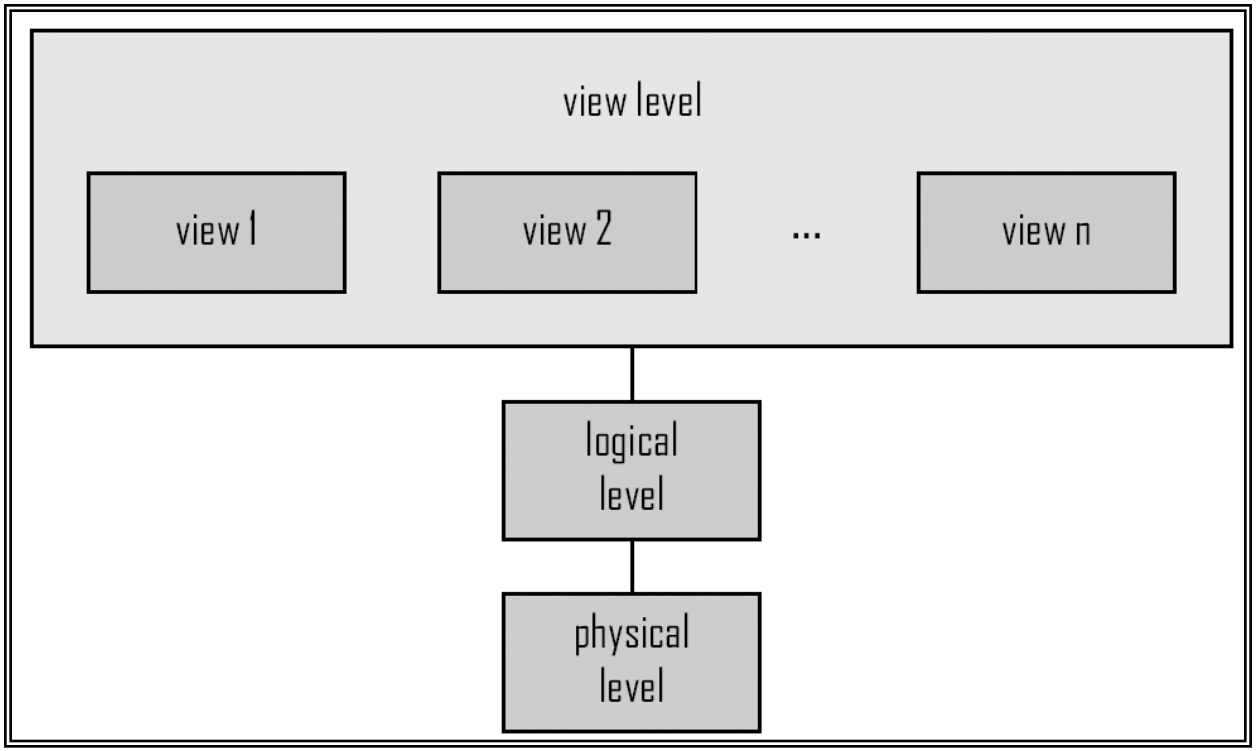
- **Logical level:** describes data stored in database in terms close to the application

flight	**reservation**	**…**
flight number	passenger name	
origin	flight number	
destination	class	
departure time	seat	

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

- **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

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.

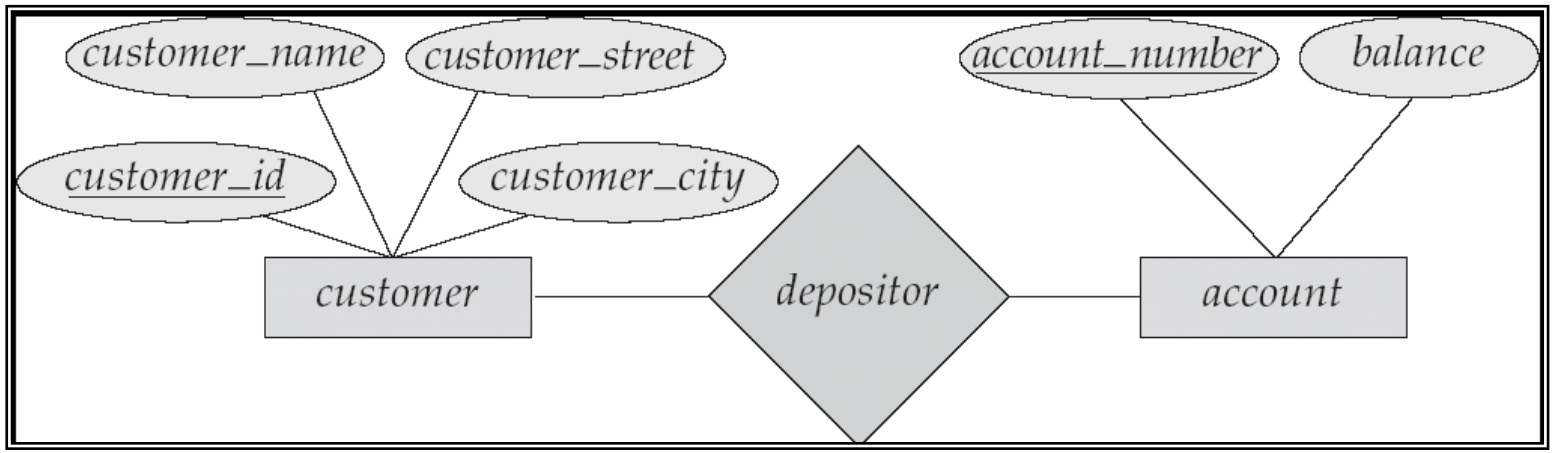
Data Models

- 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 (graphs, XML)
- Other older models:
  - Network model
  - Hierarchical model

Example: Entity-Relationship Model

- Models an application as a collection of *entities* and *relationships*
  - Entity: a "thing" or "object" 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:*

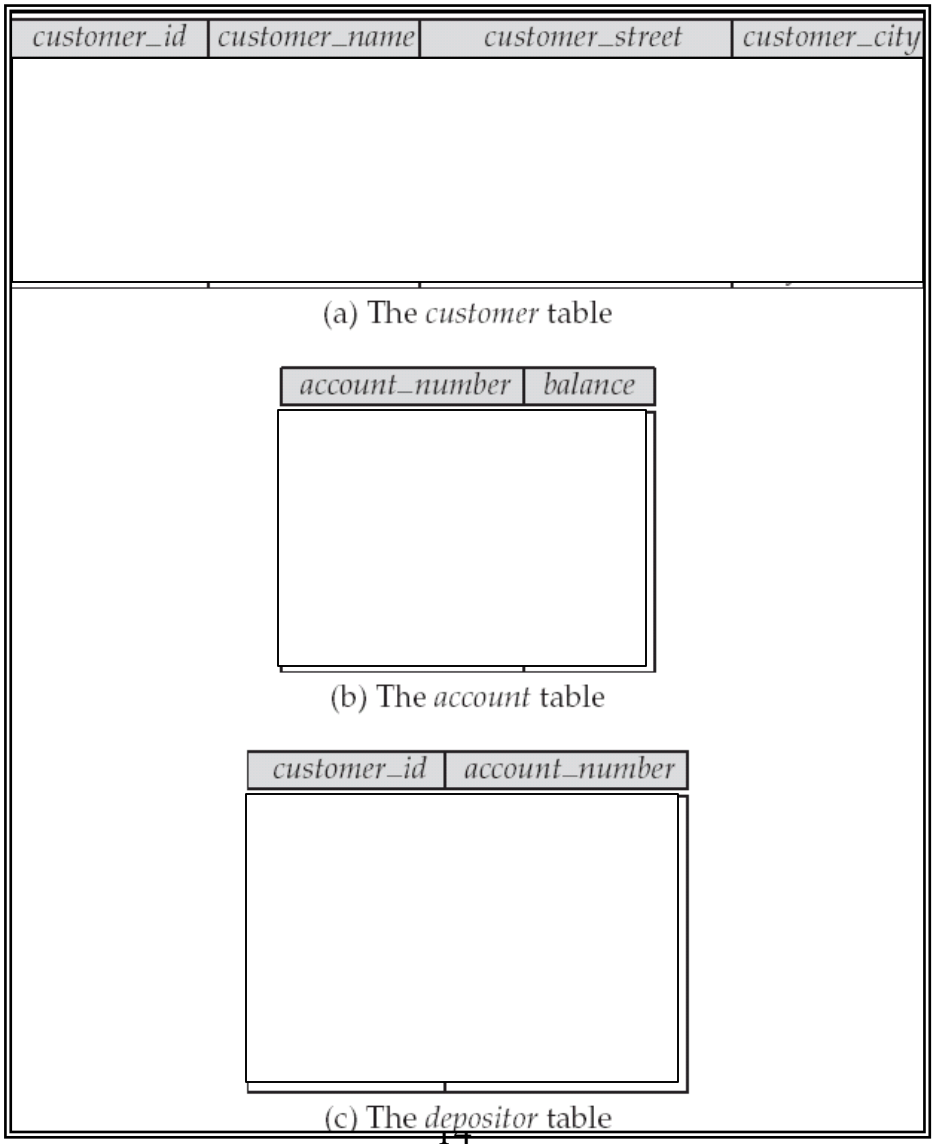


Schemas and Instances

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: Relational Model



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 in SQL:

```
create table account (
    account-number    char(10),
    balance           integer)
```

- DDL compiler generates a set of tables described in a *data dictionary*
- Data dictionary contains metadata (i.e., data about data)
  - Database schema
  - Integrity constraints
  - Authorization information

Data Manipulation Language (DML)

- Language for accessing and modifying data

  DML also known as query/update language
- Two main classes of query languages
  - Declarative (nonprocedural) – user specifies what data is required without specifying how to get it
  - Procedural – user specifies what data is required and how to get that data

- SQL is the most widely used query language

  primarily declarative

This course: core database issues

- The relational model
- Standard relational query language: SQL
- Formal query languages: relational algebra and calculus
- Query processing
- Schema design: normal forms and the ER model
- Concurrency control
- Other topics as time allows

Databases at UCSD

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

Database group Web site: https://dbucsd.github.io/

papers, seminars, bragging….

- Intersections with other CSE groups
  - storage
  - multimedia
  - machine learning
  - networks
- Intersections with HDSI (e.g. Prof. Babak Salimi)