
# LECTURE 6: Database Security

Dr. Esam A. Alwagait
alwagait@ksu.edu.sa

# Introduction to databases

- A database is a collection of data and a set of rules that organize the data by specifying certain relationships among the data
  - Structured, queryable collection of data (records)
- Each record consists of fields (elements)
- The logical structure of a database is called a schema, and is set by database administrator
- Database management system (DBMS) is a computer software that provides support for queries and management
- Most popular DBMS is based on relational model
- Stores records in one or multiple tables (relations)
- The name of each column is called an attribute of the database, and rows are called tuples
- Individual tables can have relationships between them
- A note!
  - The next generation of post-relational databases in the late 2000s became known as NoSQL databases
  - Used increasingly in Big Data and real-time web applications
  - Examples:
    - Document-based: MongoDB
    - Graph-based: SPARQL

# Advantages of using databases

- shared access
- minimal redundancy
- data consistency
- data integrity
- controlled access

# Database schema

Name-Airport

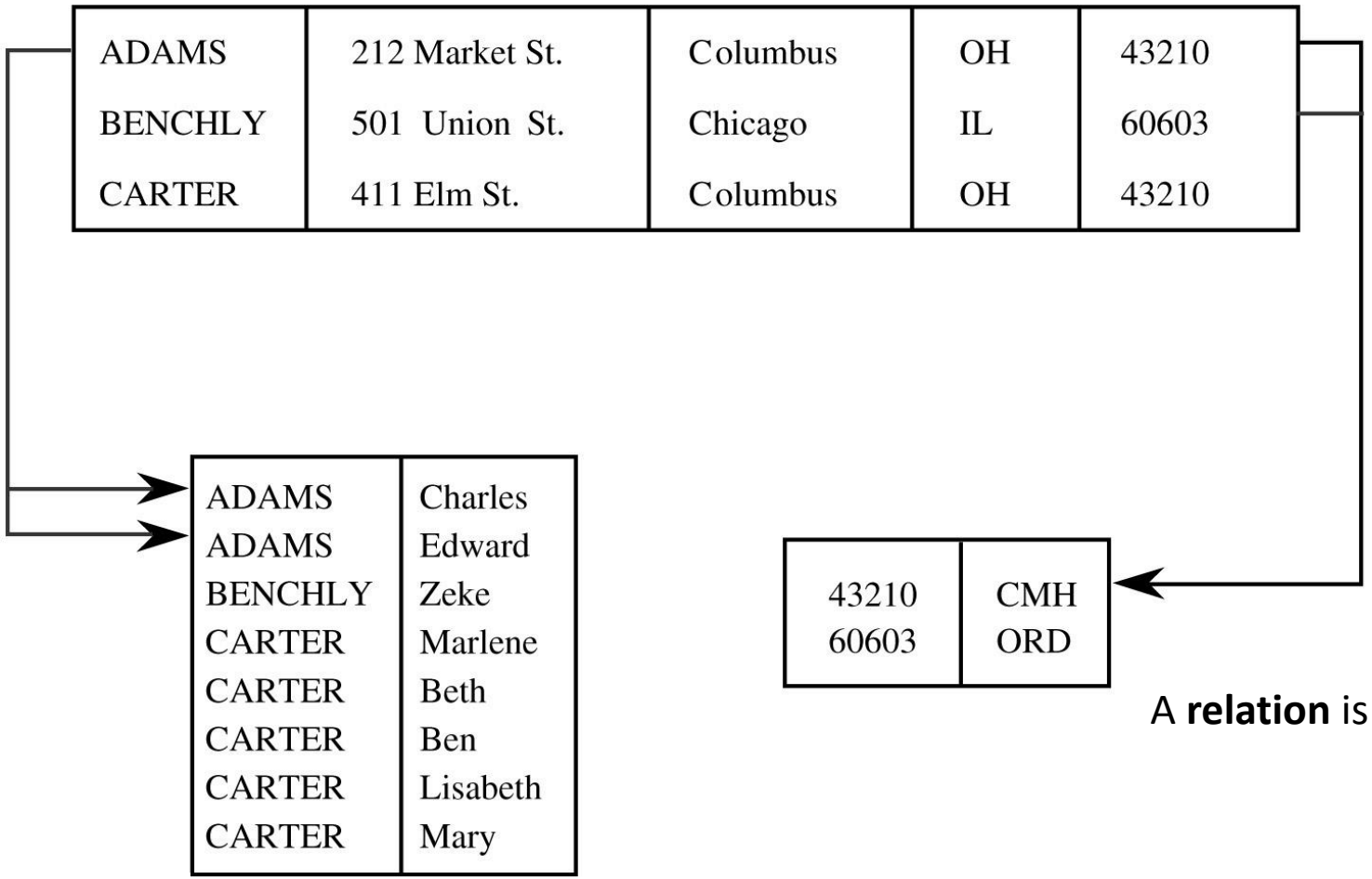
| Name | First | Address | City | State | Zip | Airport |
|---|---|---|---|---|---|---|
| ADAMS | Charles | 212 Market St. | Columbus | OH | 43210 | CMH |
| ADAMS | Edward | 212 Market St. | Columbus | OH | 43210 | CMH |
| BENCHLY | Zeke | 501 Union St. | Chicago | IL | 60603 | ORD |
| CARTER | Marlene | 411 Elm St. | Columbus | OH | 43210 | CMH |
| CARTER | Beth | 411 Elm St. | Columbus | OH | 43210 | CMH |
| CARTER | Ben | 411 Elm St. | Columbus | OH | 43210 | CMH |
| CARTER | Lisabeth | 411 Elm St. | Columbus | OH | 43210 | CMH |
| CARTER | Mary | 411 Elm St. | Columbus | OH | 43210 | CMH |

| Name | Zip |
|---|---|
| ADAMS | 43210 |
| BENCHLY | 60603 |
| CARTER | 43210 |

schema

subschema

# Database relations




A **relation** is a set of columns

# Queries

- Commands used to interact with databases
- Facilitated through commands to the DBMS that retrieve, modify, add, or delete fields and records of the database
- Database management systems have precise rules of syntax for queries
- Most popular query language is SQL
- Examples
  - SELECT NAME = 'ADAMS‘ FROM Name-Airport
  - SELECT (ZIP='43210') ^ (NAME='ADAMS‘) FROM Name-Airport
  - SELECT COUNT(Name) FROM Name-Airport WHERE City = 'Chicago'

# Security requirements

- Physical database integrity
- Logical database integrity
- Element integrity
- Referential integrity
- Auditability
- Access control
- User authentication
- Availability

# Database integrity

- Logical and physical integrity: data must be protected from corruption
- Protect against database corruption
  - Allow only authorized individuals to perform updates
- Recover from physical problems (power failures, disk crashes, etc.)
  - Perform periodic backups
  - Keep log of transactions to replay transactions since last backup

# Element integrity

- Ensure correctness and accuracy of database elements
- This corrective action can be taken in three ways
  - Element checks to validate correctness
    - Element must be numeric, within a particular range,. . .
    - Not more than one employee can be president
    - Helps against mistakes by authorized users
    - Typically enforced by triggers (procedures that are automatically executed after events such as INSERT, DELETE, etc.)
  - Access control to limit who can update element
  - Maintaining a change log for the database to undo erroneous changes
    - A change log lists every change made to the database with both original and modified values

# Referential integrity

- Each table has a primary key:
  - Minimal set of attributes that uniquely identifies each tuple
  - Examples: national ID, student #, employee #, etc.
- A table might also have a foreign key, or multiple foreign keys, which are primary keys in some other table
- Referential integrity requires every value of one attribute (column) of a relation (table) to exist as a value of another attribute in a different (or the same) relation (table)
  - i.e., when a foreign key value is used it must reference a valid, existing primary key in the parent table
- Referential integrity ensures that there are no dangling foreign keys

# Auditability

- Keep an audit log (records) of all database accesses
  - Both read and write
  - Operations, such as SELECT, INSERT, etc.
- Audit log allows to identify users who accessed forbidden data (after the fact)
  - Police accessing somebody's criminal record as a favor to a friend
  - unauthorized access to patients medical personnel
- Sometimes a combination of database accesses resulted in disclosure
  - Not necessary from a single database access
  - How?

# Access control

- More difficult than OS access control
- Might have to control access at the relation, record, or even element level
- Many types of operations, not just read/write
  - SELECT, INSERT, UPDATE, CREATE, DROP, etc.
- Relationships between database objects make it possible to learn sensitive information without directly accessing it
  - Inference problem
- Example (how to protect disclosure of salary?)
  - SELECT lastname, salary FROM staff WHERE salary > 50000

  might be forbidden, but how about:
  - SELECT lastname FROM staff WHERE salary > 50000

# User authentication / availability

- Database can perform its own, rigorous authentication
  - Password
  - Time-of-day check
- This authentication supplements OS authentication
- Availability is central to DBMS
- DBMS facilitates sharing, but availability can suffer if multiple users want to access the same record
  - Block access until other user completes updating record

# Data disclosure and inference

- Sensitive data
  - Data that should not be made public
  - Depends on database and underlying meaning and importance of data (context)
  - Example: public library versus defense-related DB?
- Controls over sensitive data becomes more challenging when DB has varying degrees of sensitivity
  - Example: university databases
- DB administrator decides what data should be in the database and who should have access to it
  - Decisions of the database administrator are based on an access policy

# Types of data disclosure

- Exact data
- Bounds
  - Sensitive value is smaller than H, but bigger than L
  - Might iteratively decrease range (binary search):
    - the user may first determine that $L \leq y \leq H$
    - Then $L \leq y \leq H/2$, and so forth
    - permitting the user to determine $y$ to any desired precision
- Negative result
  - Knowing that a person does not have zero felony convictions is sensitive, even if actual number is hidden
  - The distinction between 1 and 2 or 46 and 47 felonies is not as sensitive as the distinction between 0 and 1
- Existence
  - Knowing of existence of some data can be sensitive
  - an employer may not want employees to know that their use of long distance telephone lines is being monitored,
  - thus, discovering a LONG DISTANCE field in a personnel file would reveal sensitive data
- Probable value
  - Sensitive data has value x with probability y
  - Example: Given a person z lives in Al-Ra'ed Quarter, and want to find whether he registered in a fitness center
    - How many people live in Al-Ra'ed Quarter? (say 5000)
    - How many people live in Al-Ra'ed Quarter and have Yes for fitness subscription? (1000)
    - There is a 20% likelihood that z registered for fitness club

# Data inference

- It is a way to infer or derive sensitive data from nonsensitive data
- The inference problem is a subtle vulnerability in database security
- Direct attack
  - Attacker issues query that directly yields sensitive data
    - SELECT name FROM staff WHERE DRUGS = 1
  - Might make it less obvious!
    - SELECT name FROM staff WHERE (DRUGS = 1) OR (SEX≠M AND SEX≠F)
  - Indirect attack  (statistical inference attack)
    - Infer sensitive data from statistical results
      - The indirect attack seeks to infer a final result based on one or more intermediate statistical results
      - But this approach requires work outside the database itself
      - Data released by governments

# Data inference

Statistical inference attacks:

- Sum
  - Leaks sensitive data if sum covers only one record or if attacker can control set of covered records
  - SELECT SUM(salary)
  - SELECT SUM (salary) WHERE lastname != ‘Khalid’

# Controls for statistical inference attacks

- Apply control to query or to data items
  - Effective primarily against direct attacks
    - As it is difficult to determine whether a given query discloses sensitive data
- Suppression and concealing are two controls applied to data items
  - Suppression
    - Suppress sensitive data from result
      - the query is rejected without response
  - Concealing
    - Answer is close to actual value, but not exactly
- Suppression and concealing reflect the contrast between security and precision