



systems

INTRODUCTION TO RELATIONS & TABLES

DATABASE DESIGN

202 – Introduction to Database Systems
Week 1 / Day 1

LEARNING OBJECTIVES & AGENDA

Learning Objectives:

- Describe the need for databases.
- Understand the need for database systems.
- Describe tables and relations.

Agenda / Sub-Topics:

- Introduction to databases.
- Introduction to relational models.

INTRODUCTION TO DATABASES

FILE SYSTEMS – METHODS OF STORING / RETRIEVING DATA

- File system allows users to store / retrieve data using directories created on storage media.
- Supports basic functions such as file naming and managing file access rules.
- Data is stored on files e.g. *.xlsx, *.csv, *.json, etc.
- Examples (file systems): NTFS, FAT, Gluster, etc.

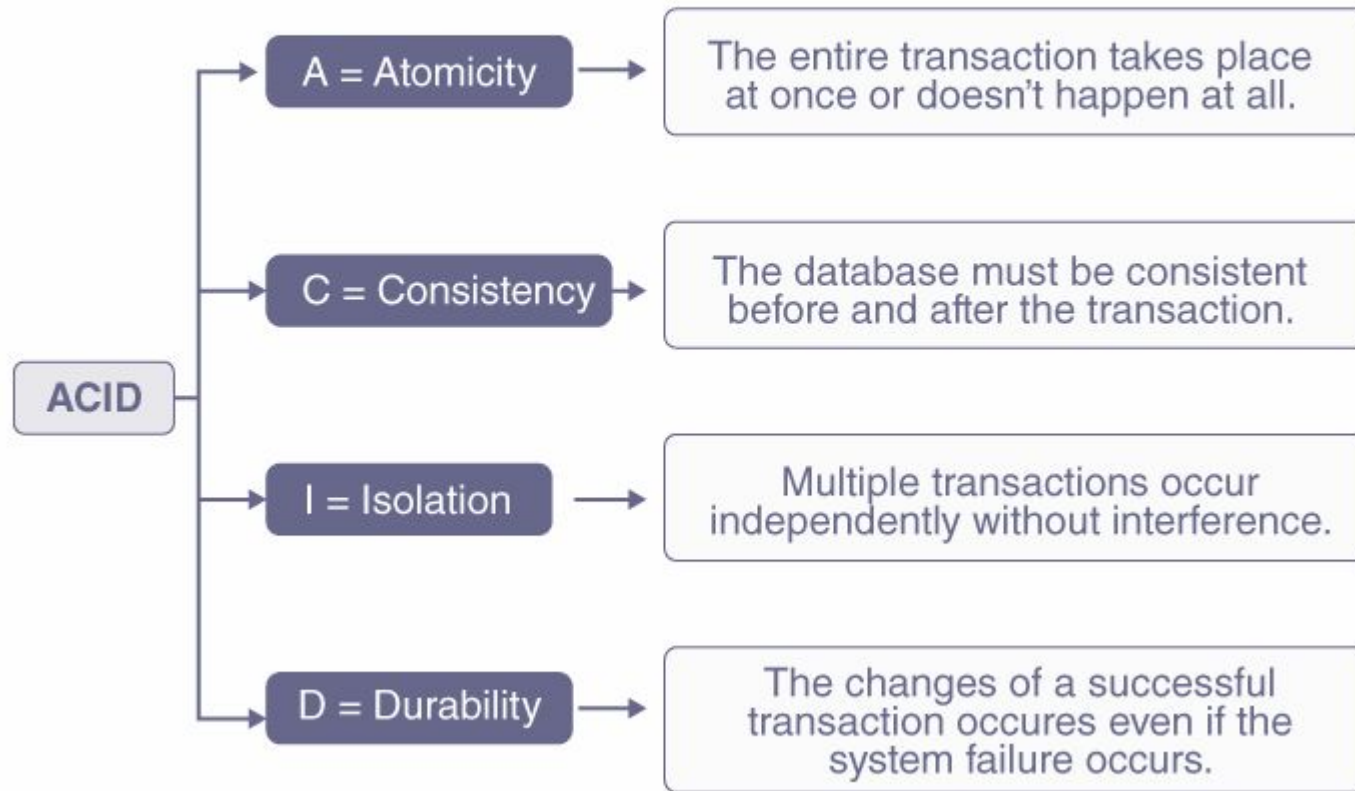
DATABASE SYSTEMS – METHODS OF STORING / RETRIEVING DATA

- Databases used to store collections of (usually related) data in an **organized** manner that allows **fast** and **efficient** access.
- Focuses on ensuring **reliability** of data.
- Provides **security** / **authorization** needs using rights.
- Allows centralized data access.
- Normally implemented using specialized software called database management systems (DBMS).
- Examples (DBMS): MySQL, MS SQL Server, Oracle.

COMPARING FILE SYSTEMS & DATABASE SYSTEMS

Feature	File System	Database System
Data Redundancy	May be present.	Designed absence.
Backup & Recovery	Not supported.	Supported.
Querying	Not supported.	Supported, efficient.
Constraints	Not supported.	Supported.
Concurrency	Not supported.	Supported.
User-Rights	May be enforced.	Always enforced.

ACID PROPERTIES OF DATABASES



RELATIONAL MODELS

INTRODUCING RELATIONAL DATABASE MODELS

- A model comprised of entities, attributes and their relations.
- Data stored in tables (may be multiple).
- Tuples (i.e. data records / rows) uniquely identified by keys.
- Tables may be governed by rules, e.g. whether to allow tables to be deleted while data is present in other tables.

KEY TERMS

Table also called Relation

The diagram shows a table with three columns: CustomerID, CustomerName, and Status. The first column is labeled 'Primary Key' and the second column is labeled 'Domain' with the example 'Ex: NOT NULL'. The table contains three rows of data. To the right of the table, the text 'Tuple OR Row' is shown with three arrows pointing to the three rows. Below the table, the text 'Column OR Attributes' is shown with three arrows pointing to the three columns. Below that, the text 'Total # of rows is Cardinality' is shown. At the bottom, the text 'Total # of column is Degree' is shown.

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive

Primary Key

Domain
Ex: NOT NULL

Tuple OR Row
Total # of rows is Cardinality

Column OR Attributes
Total # of column is Degree

STRUCTURED QUERY LANGUAGE

INTRODUCING THE STRUCTURED QUERY LANGUAGE (SQL)?

- Language used to command the DBMS to take action / retrieve data.
- Commands are known as **queries**.
- Querying syntax generally common across DBMSs.

INTRODUCING THE STRUCTURED QUERY LANGUAGE (SQL)?

- Declarative in nature; Provides abstraction from how data is accessed unlike in C++.
- User simply commands, DBMS applies the best algorithms to achieve the task.
- Used to manage / access tables and data.
- Can be divided into:
 - Data Definition Language – Manages structure.
 - Data Manipulation Language – Accesses data.