



# Database System Concepts and Architecture

---



# Objectives

---

- Terminology and Basic Concepts +
- DB Architecture & Data Independence +
- DBMS Languages +
- DBMS Interfaces +
- DBMS Component Modules +
- Database System Utilities +
- Additional Tools +
- Classification of DBMS+



## - Terminology and Basic Concepts

---

- Database Schema +
- Database States +
- Data Models +
- Categories of Data Models +



## -- Database Schema

---

- Is the description of a database
- It is specified during the database design and is not expected to change frequently.
- It is represented as a diagram called **schema diagram**.
  - A schema diagram displays the structure of each record type but not the actual instance of a record.
- Each object in a schema is called a **schema construct**.



## --- Example of a Database Schema

---

STUDENT	Name	StudentNumber	Class	Major
---------	------	---------------	-------	-------

COURSE	CourseName	CourseNumber	CreditHours	Department
--------	------------	--------------	-------------	------------

SECTION	SectionIdentifier	CourseNumber	Semester	Year	Instructor
---------	-------------------	--------------	----------	------	------------

GRADE_REPORT	StudentNumber	SectionIdentifier	Grade
--------------	---------------	-------------------	-------

PREREQUISITE	CourseNumber	PrerequisiteNumber
--------------	--------------	--------------------



## -- Database State

---

- A Database state or **instance** is the data in the database at a particular moment of time.
- Every update operation changes the database from one state to another.
- The Schema is sometimes is called the **intension**, and the database state an **extension** of the schema.



## --- Example of a Database State

STUDENT	Name	StudentNumber	Class	Major
	Smith	17	1	CS
	Brown	8	2	CS

COURSE	CourseName	CourseNumber	CreditHours	Department
	Intro to Computer Science	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Database	CS3380	3	CS

SECTION	SectionIdentifier	CourseNumber	Semester	Year	Instructor
	85	MATH2410	Fall	98	King
	92	CS1310	Fall	98	Anderson
	102	CS3320	Spring	99	Knuth
	112	MATH2410	Fall	99	Chang
	119	CS1310	Fall	99	Anderson
	135	CS3380	Fall	99	Stone

GRADE_REPORT	StudentNumber	SectionIdentifier	Grade
	17	112	B
	17	119	C
	8	85	A
	8	92	A
	8	102	B
	8	135	A

PREREQUISITE	CourseNumber	PrerequisiteNumber
	CS3380	CS3320
	CS3380	MATH2410
	CS3320	CS1310



## -- Data Models

---

- A data model is a collection of concepts that can be used to describe the structure of a database.
- By the structure we mean the data types, relationships, and constraints that should hold on the data.
- A data model also includes a set of basic operations for specifying retrievals and updates on the database.
- It is also becoming more common to include in the data model the dynamic aspects of a database application.
  - Example: Object-oriented data models





## -- Categories of Data Models

---

- Many data models have been proposed, and they can be categorized according to the types of concepts they use to describe the database structure.
  - **Conceptual (high-level) Data Models**
    - Provide concepts that are close to the way many users perceive data (Also called entity-based, object-based data models)
  - **Implementation (representational) Data Models**
    - Provide concepts that may be understood by end users but that are not too far removed from the way data is organized within the computer
    - Hide some details of data storage but can be implemented on a computer system in a direct way
    - Implementation data models include, relational, network, hierarchical, or object data models.
  - **Physical (low-level) Data Models**
    - Provide concepts that describe details of how data is stored in the computer
    - Meant for computer specialists, not for typical end users



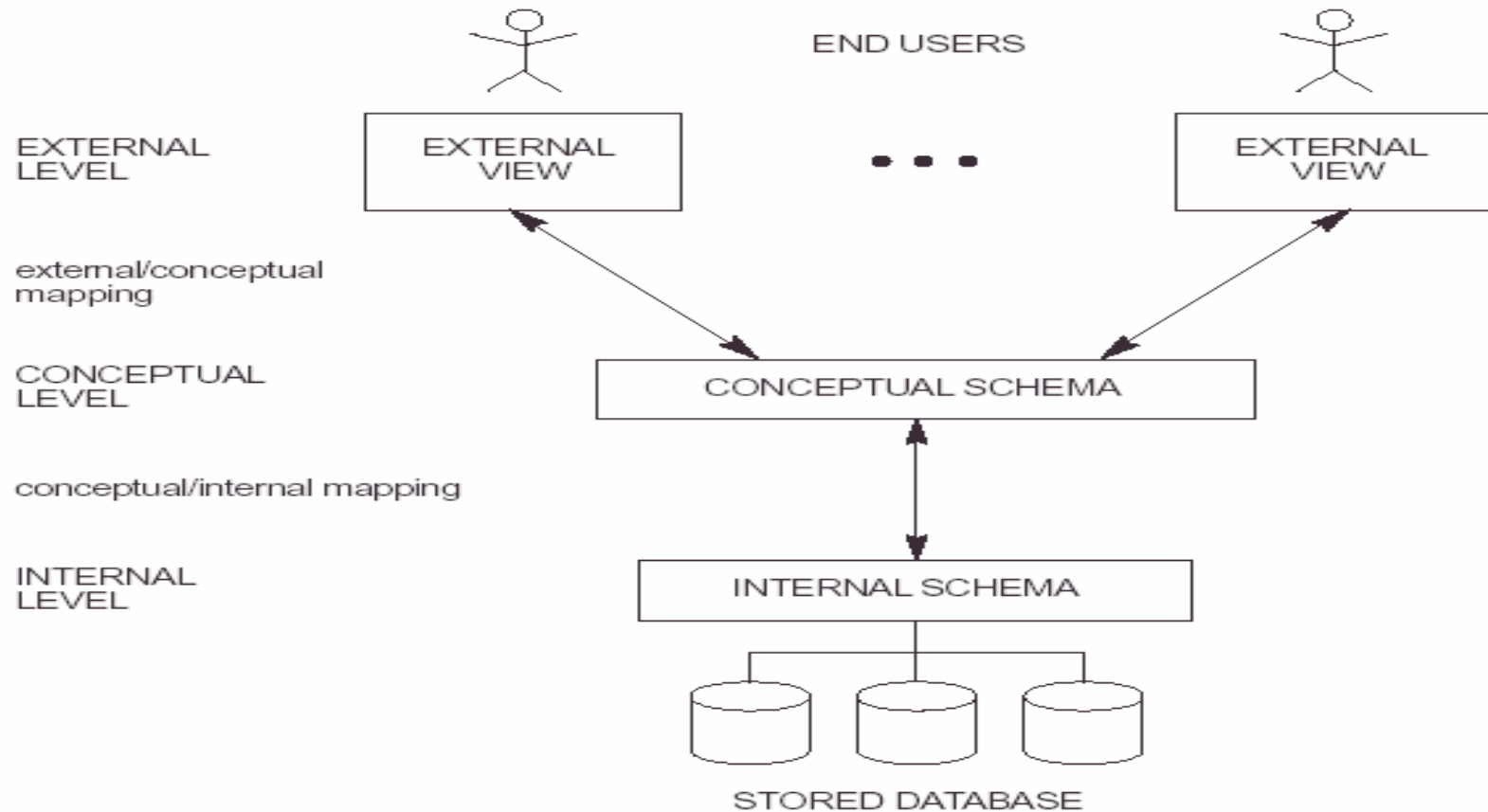
## - DB Architecture & Data Independence ...

---

- Database systems can be developed using three-schema architecture
  - **The external level or external schema**
    - describes the part of the database that a particular user group is interested in and hides the rest of the database from that user group.
    - uses a high-level data mode or implementation data model
  - **The conceptual level or conceptual schema**
    - describes the structure of the whole database for a community of users. It hides the details of the physical storage structures and concentrates on describing entities, data types, relationships, user operations, and constraints.
    - uses a high-level data mode or implementation data model
  - **The internal level or internal schema**
    - describes the physical storage structure of the database. It describes the complete details of data storage and access paths for the database.
    - uses a physical data model.

# -- Illustrating The Three-Schema Architecture

**Figure 2.2** Illustrating the three-schema architecture.





## ... - DB Architecture & Data Independence

---

- Is the ability to change the schema at one level of the database system without having to change the schema at the next higher level. There are two types of data independence. They are:
  - Logical Data Independence
    - The capacity to change the conceptual schema without having to change the external schema or application programs
  - Physical Data Independence
    - The capacity to change the internal schema without having to change the conceptual schema



## - DBMS Languages

---

- **Data Definition Language** (DDL) is a language that is used to define conceptual and internal schemas. The DDL statement is used to identify description of the schema construct and store the schema description in the DBMS catalog.
- **Data Manipulation Language** (DML) is a language that is used to manipulate that is to retrieve, insert, delete, and modify data.
  - A high-level or non-procedural DML can be used on its own to specify complex database operations in a concise manner, such as SQL.
  - A low-level or procedural DML must be embedded in a general purpose programming language. This type of DML typically retrieves individual records or objects from the database and processes each separately, such as PL/SQL.



## - DBMS Interfaces

---

- Menu-based Interfaces for Browsing
- Forms-based Interfaces
- Graphical User Interfaces
- Natural Language Interfaces
- Interfaces for Parametric Users
- Interfaces for The DBA



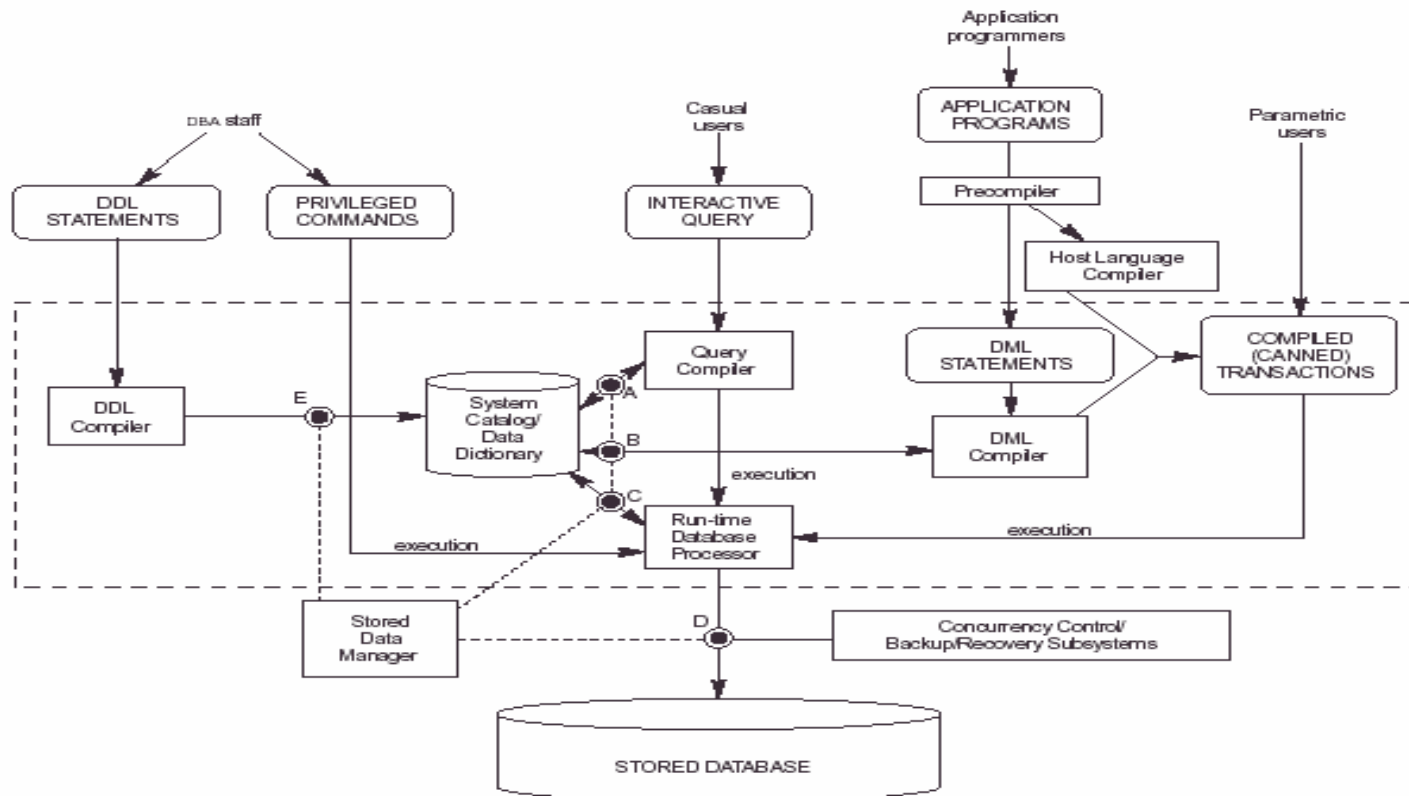
## - DBMS Component Modules

---

- A typical DBMS consists of the following components:
  - **DDL compiler**: process schema definitions, specified in the DDL statements, and stores descriptions of the schemas in the system catalog.
  - **DML compiler**: compiles the DML commands into object code for database access.
  - **Run-time database processor**: handles database access at run time. It receives retrieval and update operations and carries them out on the database.
  - **Query compiler**: handles high-level queries that are entered interactively.
  - **Data manager**: controls access to DBMS information that is stored on disk through interaction with operating system.

# -- DBMS Component Modules

**Figure 2.3** Typical component modules of a DBMS. Dotted lines show accesses that are under the control of the stored data manager.







## - Database System Utilities

---

- There are some functions that are not provided through the normal DBMS components rather they are provided through additional programs called utilities. Some of these are:
  - **Loading or import utility**: used to load or import existing data files into the database.
  - **Backup utility**: used to create backup copies of the database, usually by dumping the entire database onto tape.
  - **File reorganization utility**: is used to reorganize a database file into a different file organization to improve performance.
  - **Performance monitoring utility**: is used to monitor database usage and provides statistics to the DBA.



## - Additional Tools

---

- Some common additional tools used by workers involved in the database system are:
  - **Case tools**: are used in the design phase of the database system.
  - **Data repository** system: stores information such as design decisions, usage standards, application program descriptions, and user information.
  - **Application development environment**: provides facilities for developing database applications including database design, graphical user interface development, querying and updating, and application program development.
  - **Communications software**: allows users at locations remote from the database system site to access the database through computer terminals, work stations, or their local personal computers.

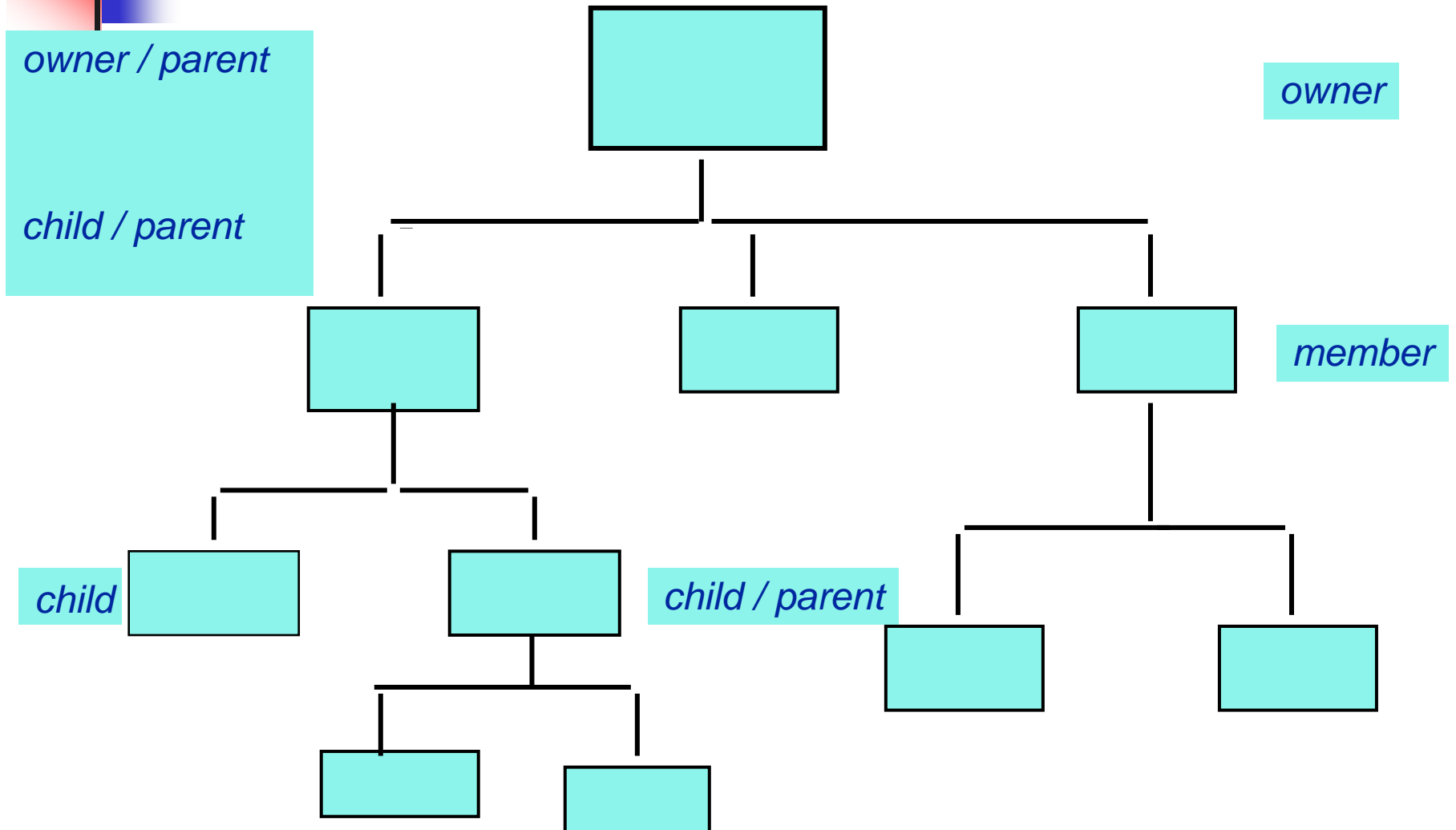


## - Classification of DBMSs

---

- Several criteria are normally used to classify DBMSs
  - Data Model:
    - Relational data model
    - Object data model
    - Hierarchical data model
    - Network data model
  - Number of Users:
    - Single-user system
    - Multi-user system
  - Number of Sites
    - Distributed
    - Centralized
  - Cost:
  - Purpose:
    - General
    - special

# -- Data Base Models - Hierarchical

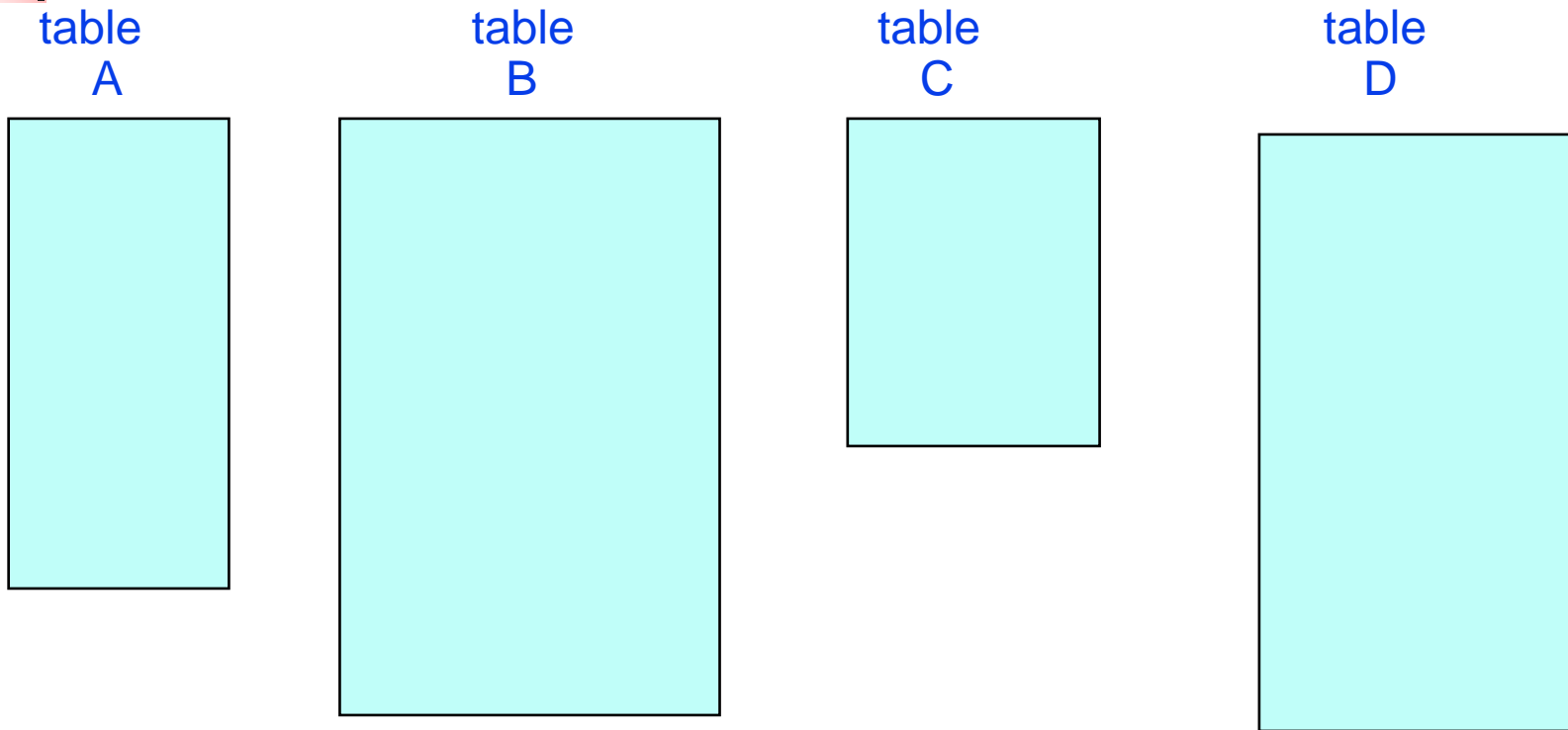






## -- Database Model - Relational

---



Any table(s) can be joined to any other table(s), *provided* there is a means of effecting the join

Primary key / Foreign key concept. Data redundancy

No fixed linkages