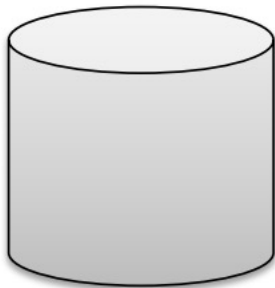# Database Technologies

MS-218
Semester 1
Session 2021-22

History of the Database Timeline

- 1960s: Computers become cost effective for private companies, and storage capacity increases.

- 1970-72: E. F. Codd proposes the relational model for databases, disconnecting the logical organization from the physical storage.

- 1976: P. Chen proposes the entity relationship model (ERM) for database design.

History of the Database Timeline

- Early 1980s: The first commercially-available relational database systems start to appear at the beginning of the 1980s with Oracle Version 2.

- Mid-1980s: SQL (structured query language) becomes "intergalactic standard."

- Early 1990s: An industry shakeout begins with fewer surviving companies. Oracle survives.

History of the Database Timeline

- Mid-1990s: Kaboom! The usable Internet/World Wide Web (WWW) appears. A mad scramble ensues to allow remote access to computer systems with legacy data.

- Late 1990s: The large investment in Internet companies helps create a tools-market boom for Web/Internet/DB connectors.

- Early 21st century: Solid growth of DB applications continues. Examples: commercial websites (yahoo.com, amazon.com, google.com), government systems (Bureau of Citizenship and Immigration Services, Bureau of the Census), art museums, hospitals, schools, etc.
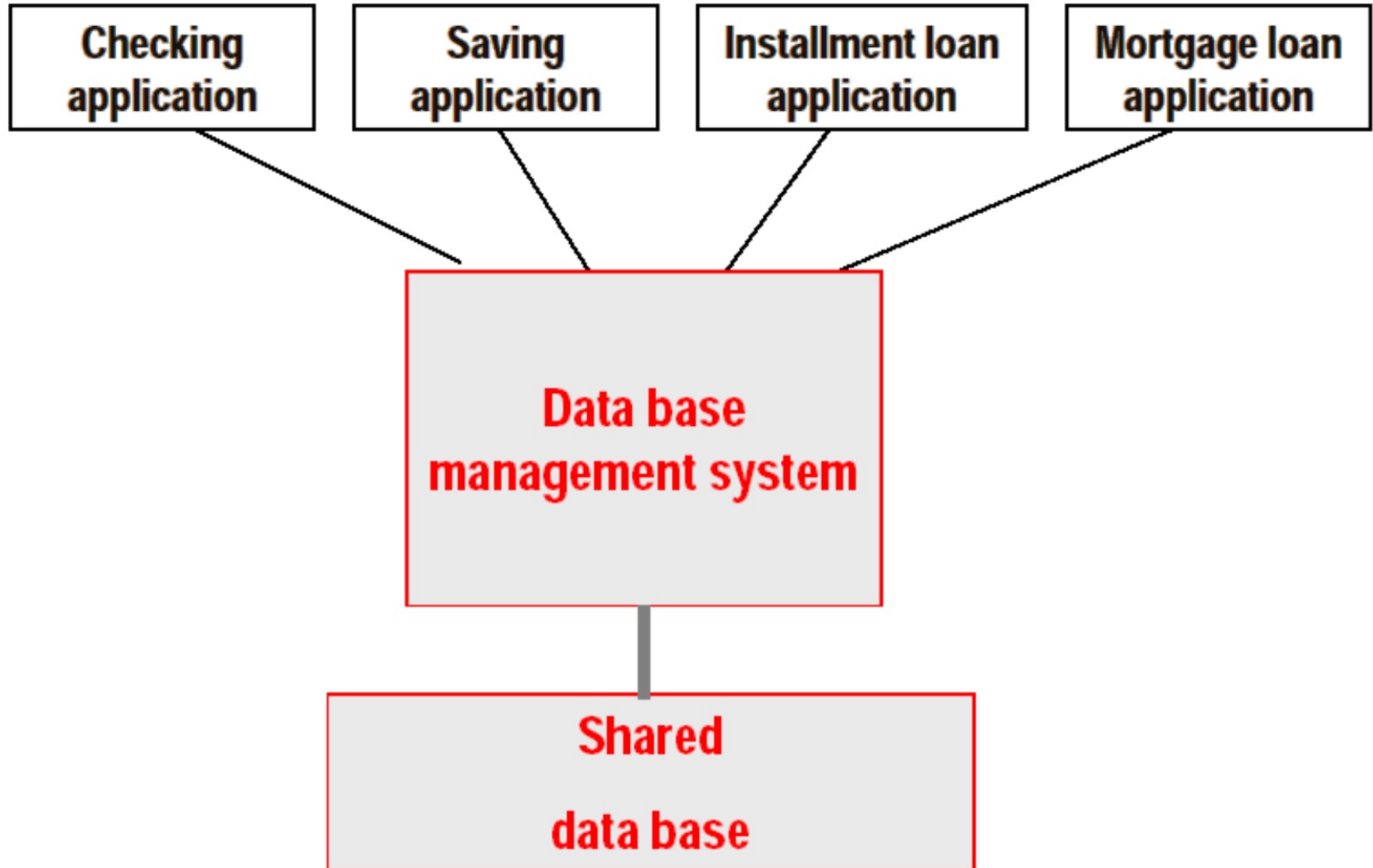
The Birth of DBMS-1
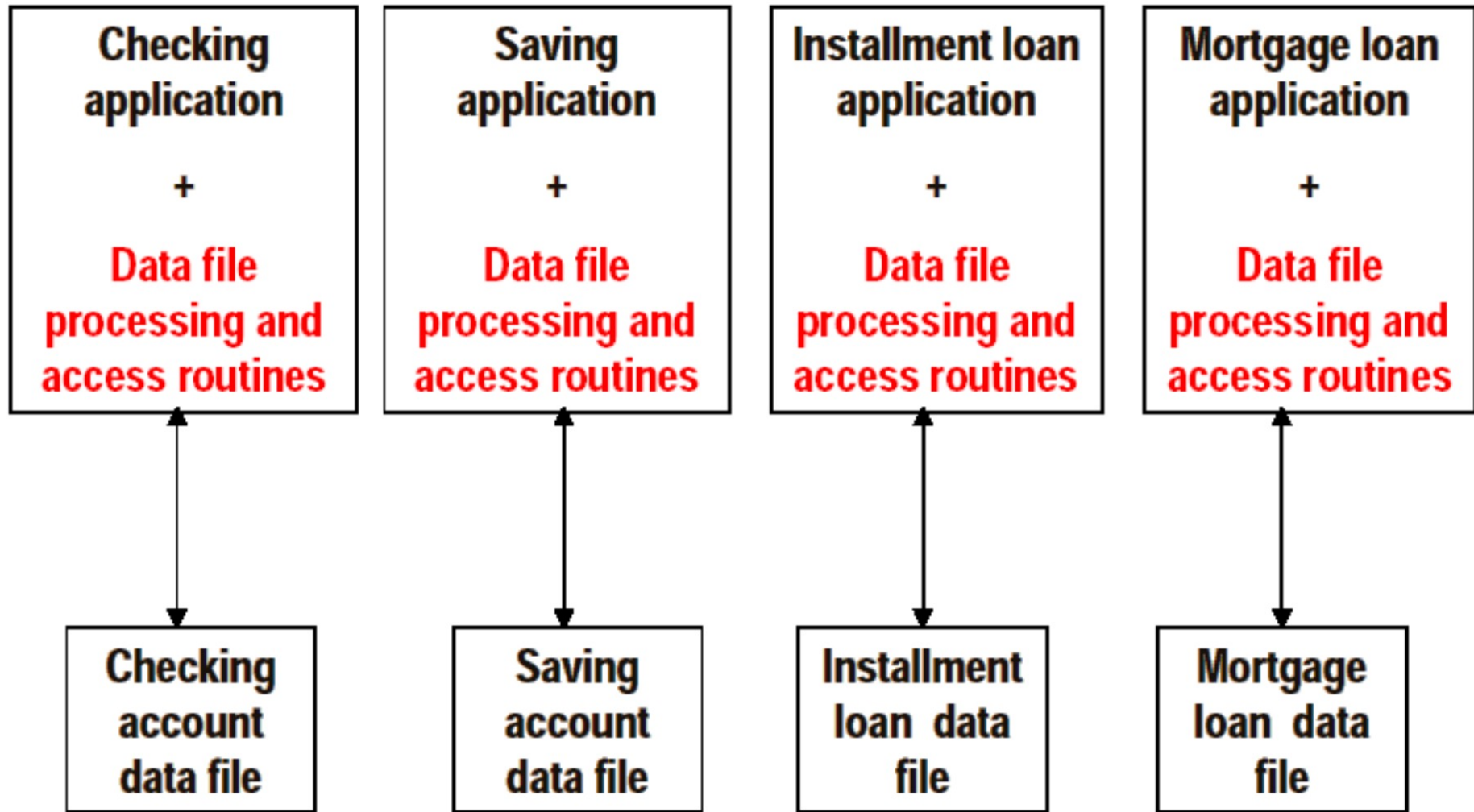
The Birth of DBMS-2
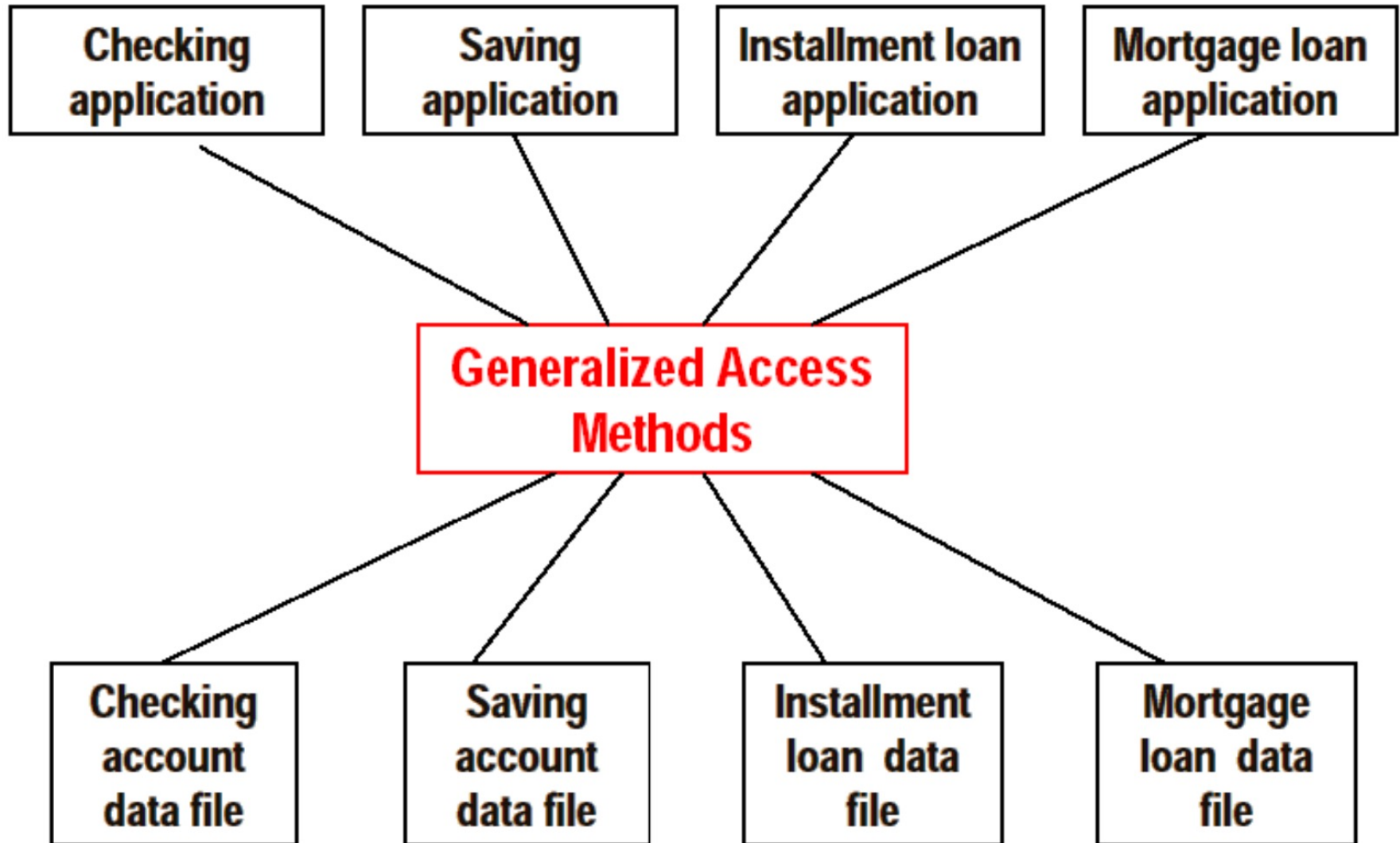
The Birth of DBMS-3

Data Models

❖ A data model is a collection of concepts for describing the data in a database.

Relational ← This Course
Key/Value
Graph
Document
Column-family
Array / Matrix
Hierarchical
Network
Multi-Value

What is a DBMS?

❖ A large, integrated collection of data

❖ Models a real-world *enterprise*
  ▪ *Entities* (e.g., Students, Courses)
  ▪ *Relationships* (e.g., Sara is enrolled in MS-218)

A **Database Management System (DBMS)** is a piece of software designed to store and manage databases

A Motivating, Running Example

❖ Consider building a course management system (**CMS**):

- Students
- Courses   } *Entities*
- Professors

- Who takes what
- Who teaches what   } *Relationships*

Attributes

❖ Each entity would have its own attributes/features/characteristics

- Students
- Courses
- Professors

Entities

Attributes of the student's entity:

- Student Name
- Registration#

Attributes of the Students entity

Relationships

- Who takes what
- Who teaches what

Relationships

Relational Model

❖ A relation is an unordered set that contain the relationship of attributes that represent entities.

Artist(name, year, country)

name	year	country
Wu-Tang Clan	1992	USA
Notorious BIG	1992	USA
Ice Cube	1989	USA

Relational Model

❖ A tuple is a set of attribute values in the relation.

Artist(name, year, country)

name	year	country
Wu-Tang Clan	1992	USA
Notorious BIG	1992	USA
Ice Cube	1989	USA

Relational Model (Primary key)

❖ A relation's primary key uniquely identifies a single tuple.

Artist(<u>id</u>, name, year, country)

id	name	year	country
123	Wu-Tang Clan	1992	USA
456	Notorious BIG	1992	USA
789	Ice Cube	1989	USA

Relational Model (Foreign key)

❖ A foreign key specifies that an attribute from one relation **must** map to a tuple in another relation.

Artist(<u>id</u>, name, year, country)

id	name	year	country
123	Wu-Tang Clan	1992	USA
456	Notorious BIG	1992	USA
789	Ice Cube	1989	USA

Album(<u>id</u>, name, artists, year)

id	name	artists	year
11	Enter the Wu-Tang	123	1993
22	St.Ides Mix Tape	???	1994
33	AmeriKKKa's Most Wanted	789	1990

Relational Model (Foreign key)

❖ A foreign key specifies that an attribute from one relation **must** map to a tuple in another relation.
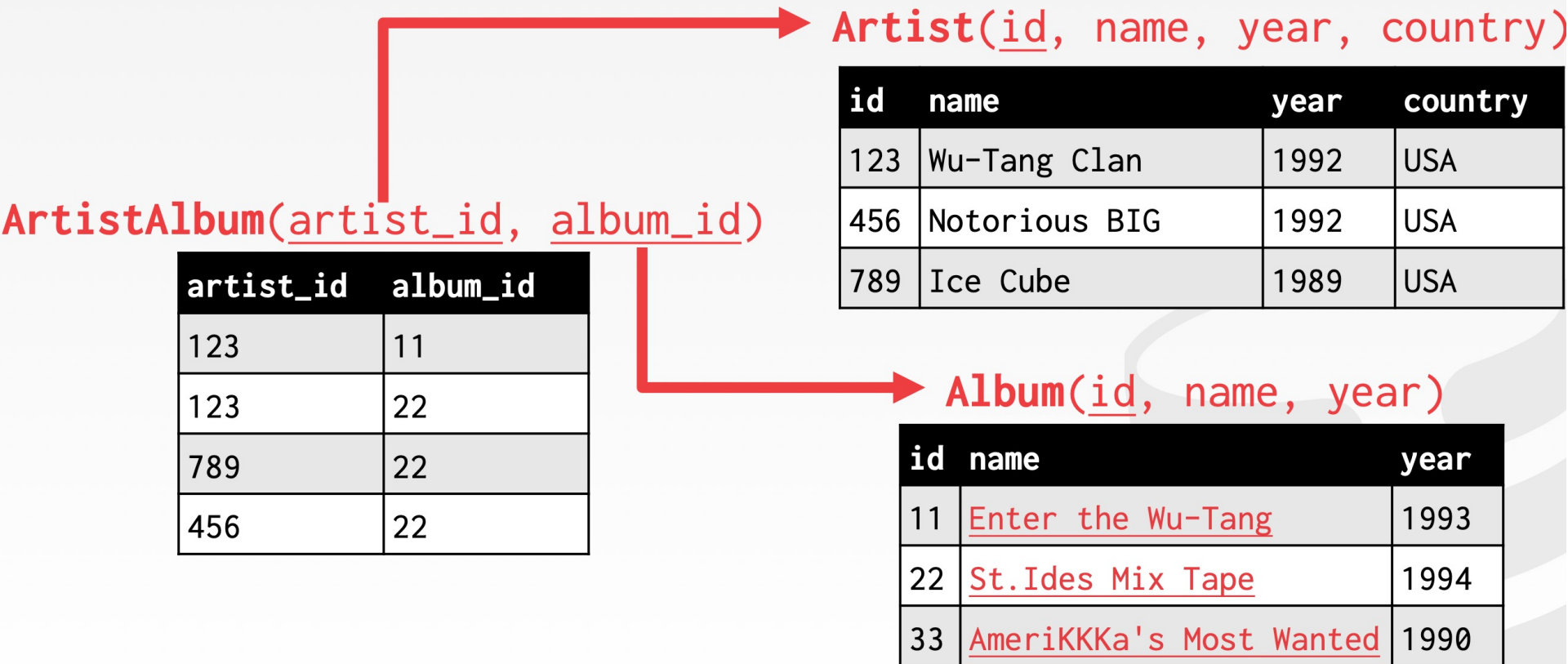
Artist(id, name, year, country)

id	name	year	country
123	Wu-Tang Clan	1992	USA
456	Notorious BIG	1992	USA
789	Ice Cube	1989	USA

ArtistAlbum(artist_id, album_id)

artist_id	album_id
123	11
123	22
789	22
456	22

Album(id, name, year)

id	name	year
11	Enter the Wu-Tang	1993
22	St.Ides Mix Tape	1994
33	AmeriKKKa's Most Wanted	1990

Foreign Keys

Physical data independence

❖ Applications should not need to worry about how data is physically structured and stored

❖ Applications should work with a logical data model and declarative query language

❖ Leave the implementation details and optimization to DBMS

❖ The single most important reason behind the success of DBMS today

Database Development Process