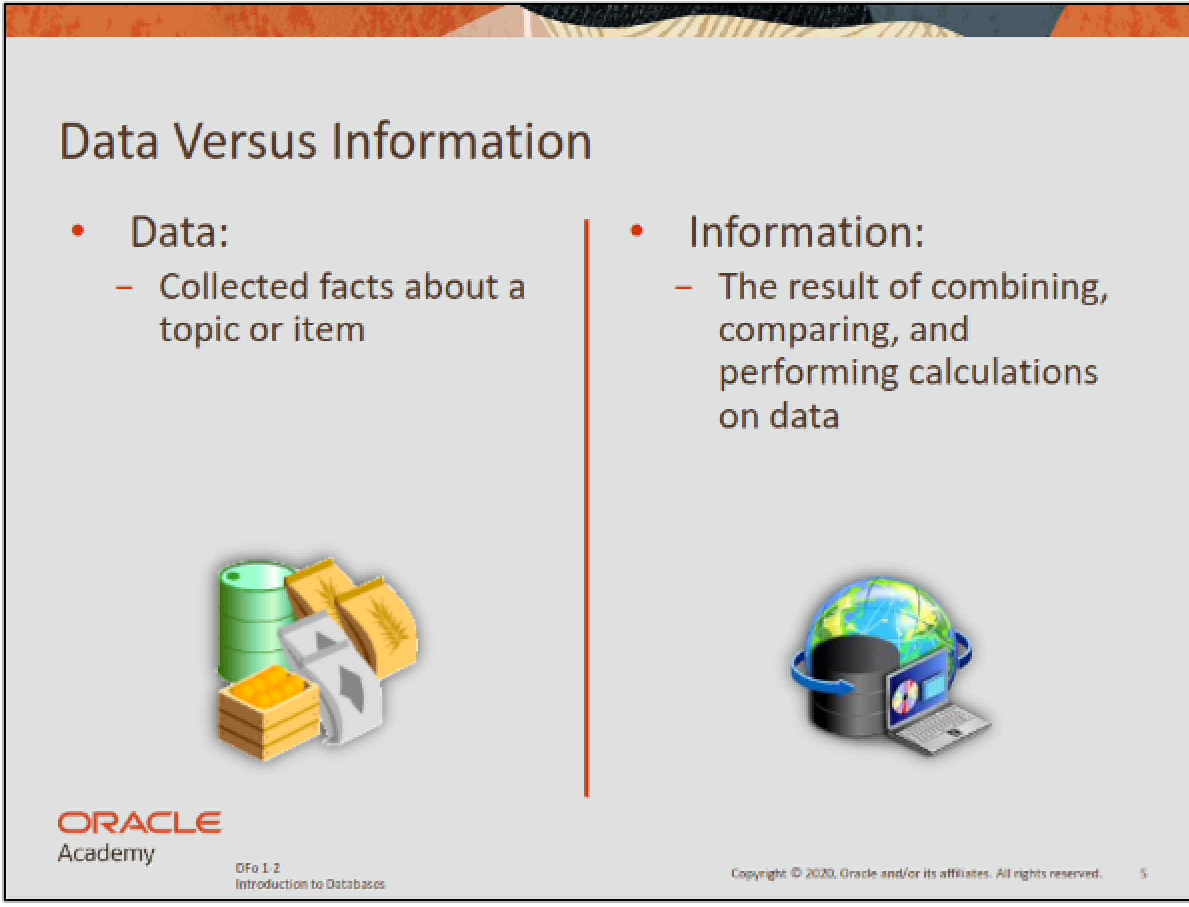
Database Foundations

1-2
Introduction to Databases

ORACLE
Academy



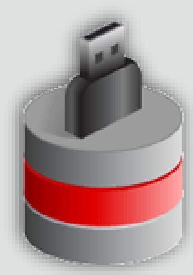
The words "data" and "information" are often used as if they are synonyms. Nevertheless, they have different meanings.

The difference between data and information can be explained by using an example such as test scores. In one class, if every student receives a numbered score, the scores can be calculated to determine a class average. The class averages can be calculated to determine the school average. So in this scenario, how can you differentiate between data and information?

- For data, each student's test score is one piece of data.
- Information is the class's average score or the school's average score.

Database Definition

- A database:
  - Is a centralized and structured set of data stored on a computer system
  - Provides facilities for retrieving, adding, modifying, and deleting the data when required
  - Provides facilities for transforming retrieved data into useful information



ORACLE
Academy

Every organization needs to collect and maintain data to meet its requirements. An information system can be defined as a formal system for storing and processing data. Most organizations today use a database to automate their information systems. A database is an organized collection of data put together as a unit. The rationale of a database is to collect, store, and retrieve related data for use by database applications. A database application is a software program that interacts with a database to access and manipulate data. A database is usually managed by a database administrator (DBA).

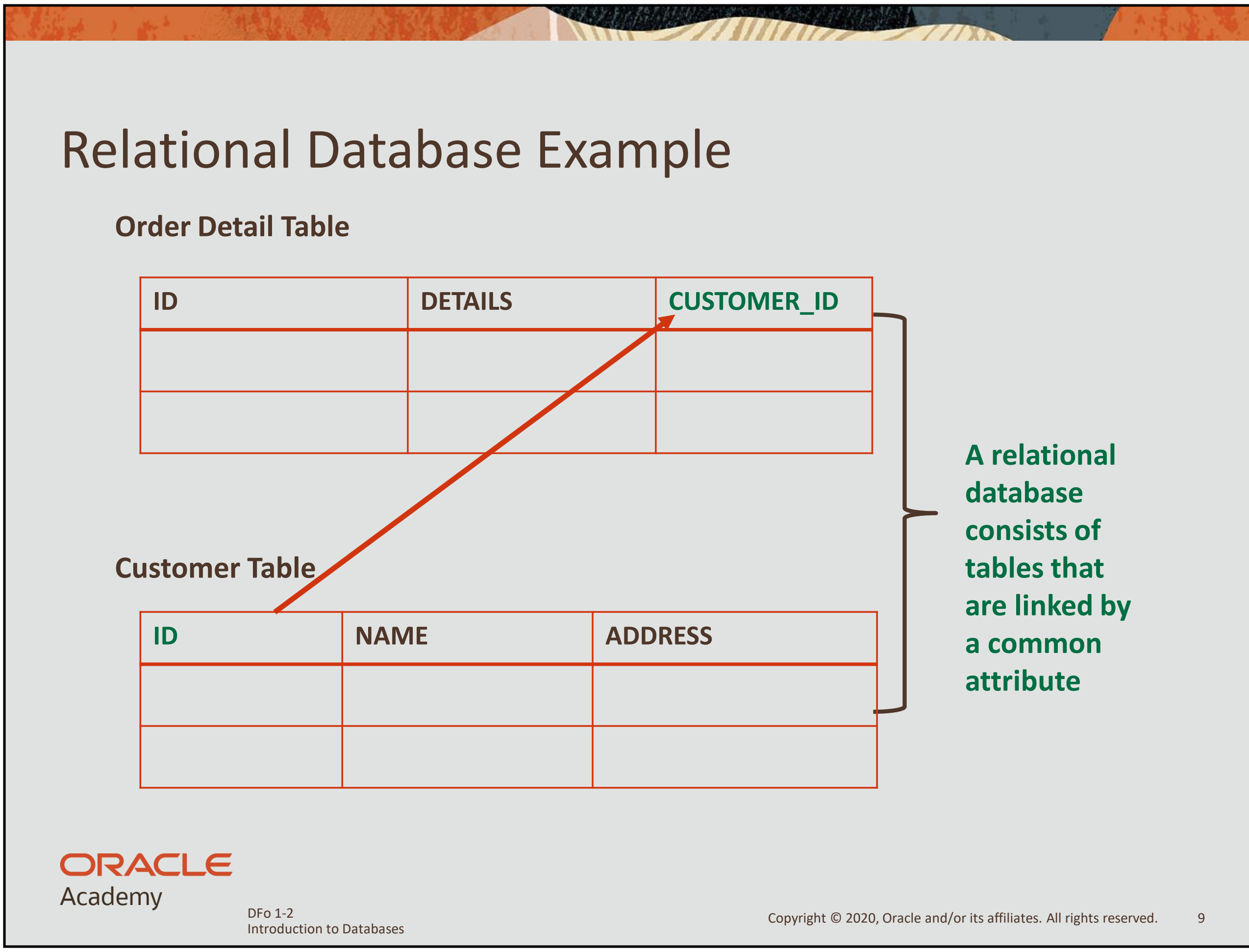
Introduction to Relational Databases

- A relational database stores information in tables with rows and columns
- A table is a collection of records
- A row is called a record (or instance)
- A column is referred to as a field (or attribute)

ORACLE
Academy

A relational database is a collection of records that are stored in tables. Each relational database table contains rows of records and columns containing fields of information about each record. A table in a relational database can also be referred to as an entity. A row in a relational database can also be referred to as an instance.

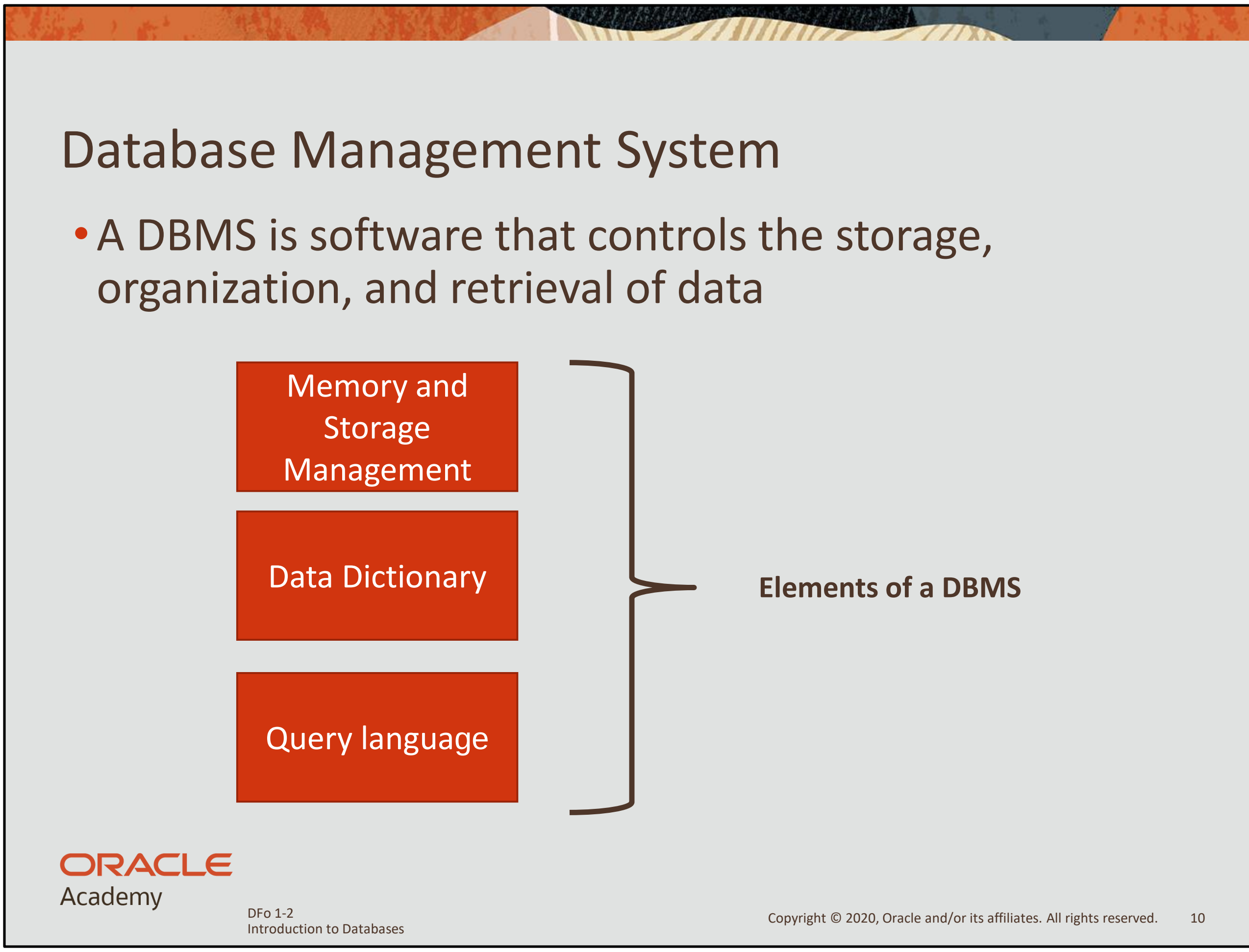
Each table of records will have a relationship to another table of records when the two tables share a field (or column).



The slide depicts two tables: Order Details and Customer. The tables are related to each other by a common attribute, ID and Customer ID.

Imagine a single order placed by a customer. Each order will contain one or more order details. Each detail will be related to one customer.

The data provides information about the details of orders placed by customers. For example, the company could gather information about products that are commonly purchased together. Bundles of products could then be offered to better market the products to customers.



A DBMS has the following elements:

- The kernel code manages memory and storage for the DBMS.
- The repository of metadata is called a data dictionary.
- The query language enables applications to access the data.

Key Computing Terms

- In the field of computing, these are some of the key terms:
  - **Hardware** : physical parts of a computer
  - **Software** : instructions to tell hardware what to do
  - **Operating system** : software that directly controls the hardware
  - **Application** : performs specific task
  - **Client** : workstation used by end users
  - **Server** : accepts work requiring more power from clients

Hardware: The physical "bits and pieces" of a computer; for example, keyboard, screen, mouse, disk drive, memory.

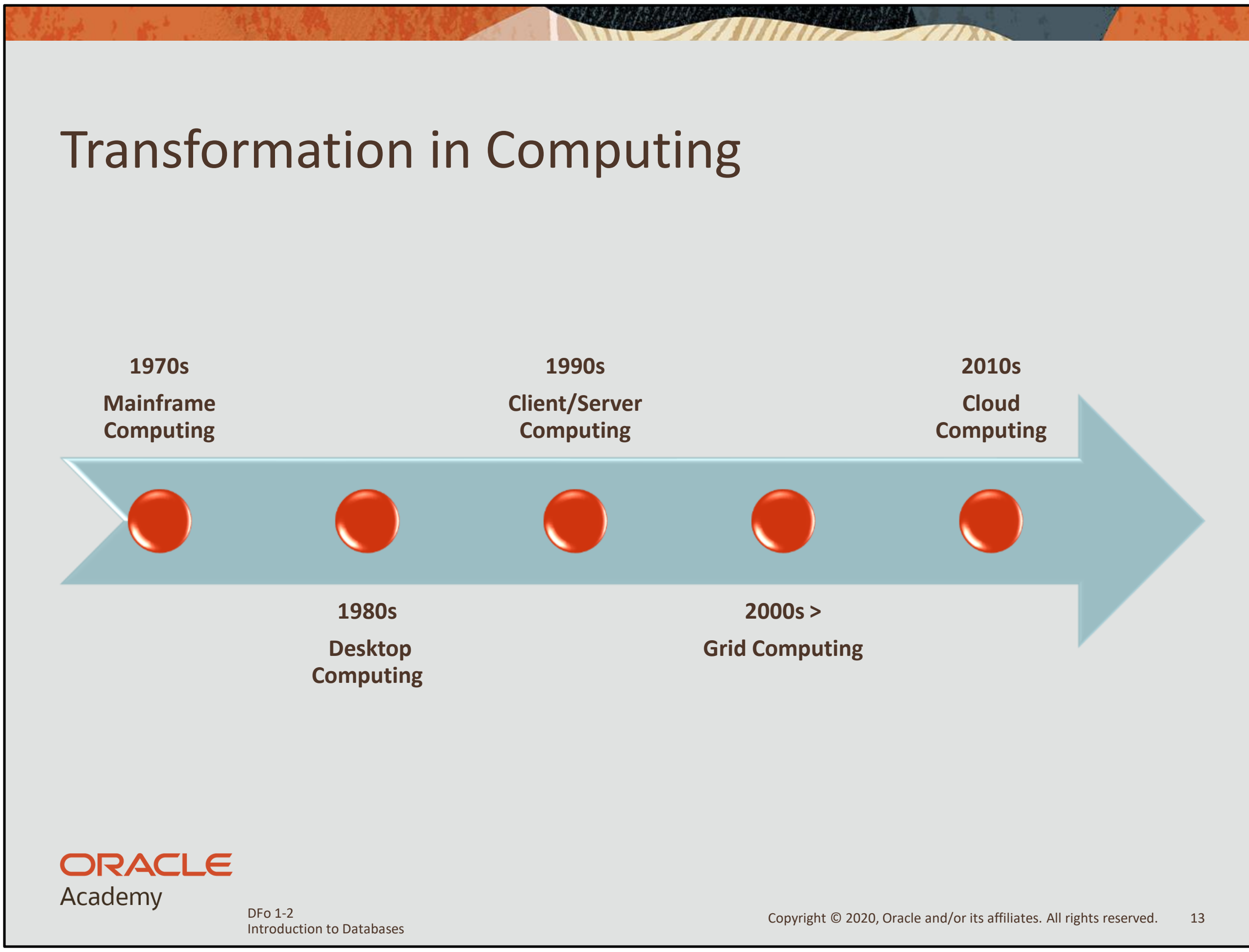
Software: Programs (sets of instructions) that tell the hardware what to do.

Operating system: A software program that directly controls and manages the hardware; for example, Microsoft Windows.

Application: A software program that carries out specific tasks on behalf of computer users – Microsoft Word or Excel for example.

Client: A workstation or desktop computer, including a screen, a keyboard, and a mouse. Clients communicate directly with human computer users.

Server: A more powerful computer that accepts work requests from clients, does the work, and sends results back to the client.

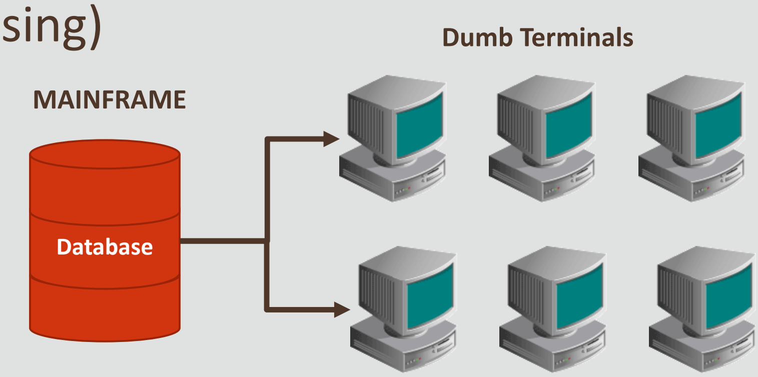


Early computer applications focused on tasks that were clerical in nature; for example, payroll, accounting, and inventory. These applications accessed data stored in computer files, converted the data into meaningful information, and generated reports to fulfill the organization's requirements. These systems were called file-based systems.

The decades-long evolution in computer technology, coupled with the needs and demands of organizations, has resulted in the development of a database technology from the primitive file-based systems to the robust, integrated database systems of today.

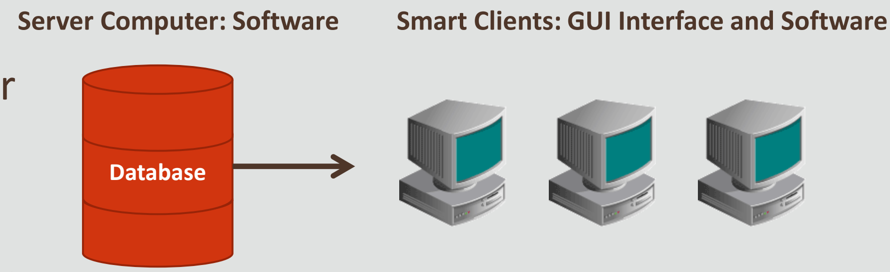
1970s: Mainframe Computing (Centralized Processing)

- In the 1970s, attempts were made to build database systems with integrated hardware and software



- Smaller computers, or "dumb terminals," were used to access the large mainframe and execute commands
- The terminals depended on the mainframe and displayed the results only after the processing was completed in the mainframe
- They were not capable of much processing on their own

1980s: Desktop Computing (Localized Processing)

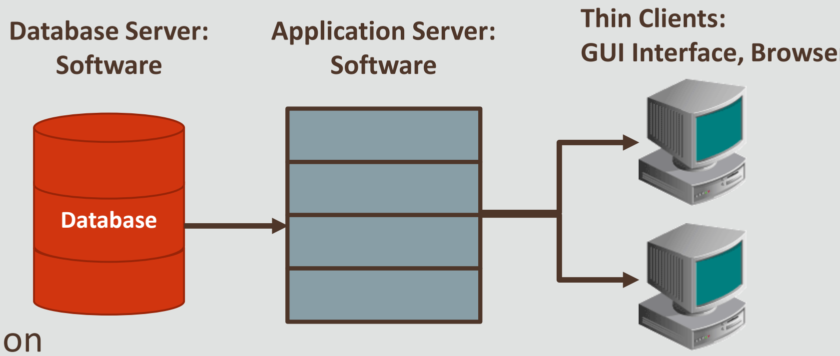


- As PCs became faster and widely available, processing moved from mainframes to clients
- PCs had their own software and were capable of doing some processing on their own, they came to be known as "smart clients" or "workstations.“
- Having the processing power within the client machine ushered in a wave of graphical user interface (GUI) applications. Many of today’s common applications (Word, Excel, PowerPoint) were created during this era

ORACLE Academy

1990s: Client/Server Computing (Centalized and Local Processing)

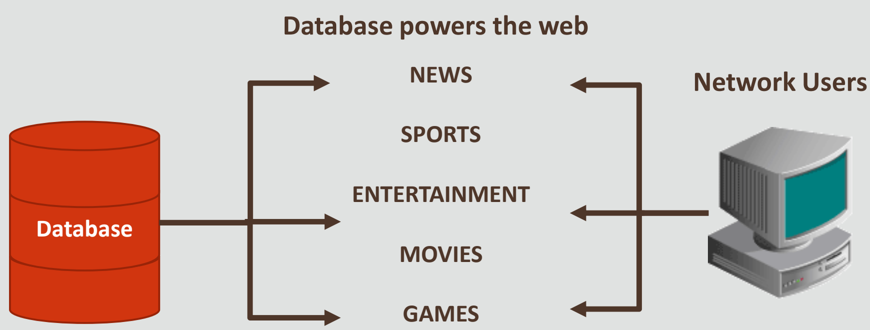
- Client/Server computing uses the Internet and fast processing servers to meet the needs of organizations in storing data and producing information



- The software that manages the data is on the database server, it performs processing for storage and retrieval
- Applications for business operations sit on the application server, it performs processing for document creation, developing, interacting, or manipulating the data
- Clients can have applications of their own, but the essential business applications are accessed from the clients by using an Internet browser

Upgradation was one of the issues with multiple applications on multiple client workstations. An upgrade made to a software application warranted that each and every server plus each and every client had to be upgraded this paved the path to grid computing.

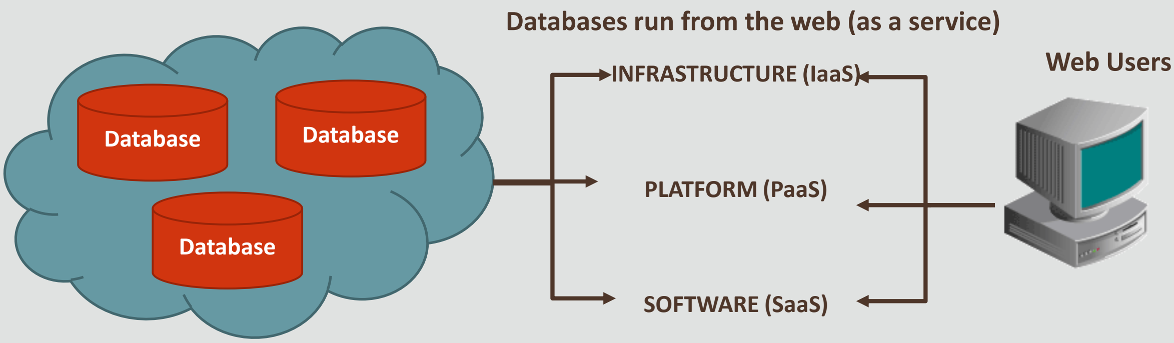
2000s: Grid Computing (Shared Processing)



- In the grid-computing model, all of an organization’s computers in different locations can be utilized just like a pool of computing resources
- Grid computing builds a software infrastructure that can run on a large number of networked servers
- A user makes a request for information or computation from his or her workstation and that request is processed somewhere in the grid as efficiently as possible

Grid computing treats computing as a utility, like the electric company. You don't know where the generator is or how the electric grid is wired. You just ask for electricity and you get it.

2010s: Cloud Computing (Internet Based Processing)



- Cloud computing allows the delivery of computing services over the Internet
- The three main categories of cloud services are:
  - IaaS – Allows you to rent cloud based servers, storage, operating systems etc
  - PaaS – Gives access to an online environment for developing and testing software without any setup or management costs
  - SaaS – Delivers software direct from the Internet. Users normally access it through a web browser

Most people use cloud (web based) services all the time without even realising it. Storing files online such as photographs, using movie subscription services or online game playing are all examples of cloud computing.

What is Cloud Computing?

On-demand self-service

No human intervention needed to get resources

Broad network access

Access from anywhere

Resource pooling

Provider shares resources to customers

Rapid elasticity

Get more resources quickly as needed

Measured service

Pay only for what you consume

History of the Database Timeline

Year	Description
1960s	Computers become cost-effective for private companies along with increased storage capability
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
Early 1980s	The first commercially available relational database systems start to appear at the beginning of the 1980s with Oracle Version 2

History of the Database Timeline

Year	Description
Mid-1980s	SQL (structured query language) becomes widely used
1990s	The large investment in Internet companies helps create a tools market boom for web/internet/DB connectors
2000s	Solid growth of DB applications continues. Examples: commercial websites (yahoo.com, amazon.com), government systems (Bureau of Citizenship and Immigration Services, Bureau of the Census), art museums, hospitals, schools
2010s	Cloud based services from companies such as Oracle, Apple and Microsoft as well as Amazon's AWS have turned Cloud Computing into a multi billion dollar industry

Examples

- Schools and colleges use databases to maintain details about courses, students, and faculty
- Banks use databases for storing information on customers, accounts, loans, and transactions
- Airlines and railways use online databases for reservations and for displaying information on the schedule

Examples

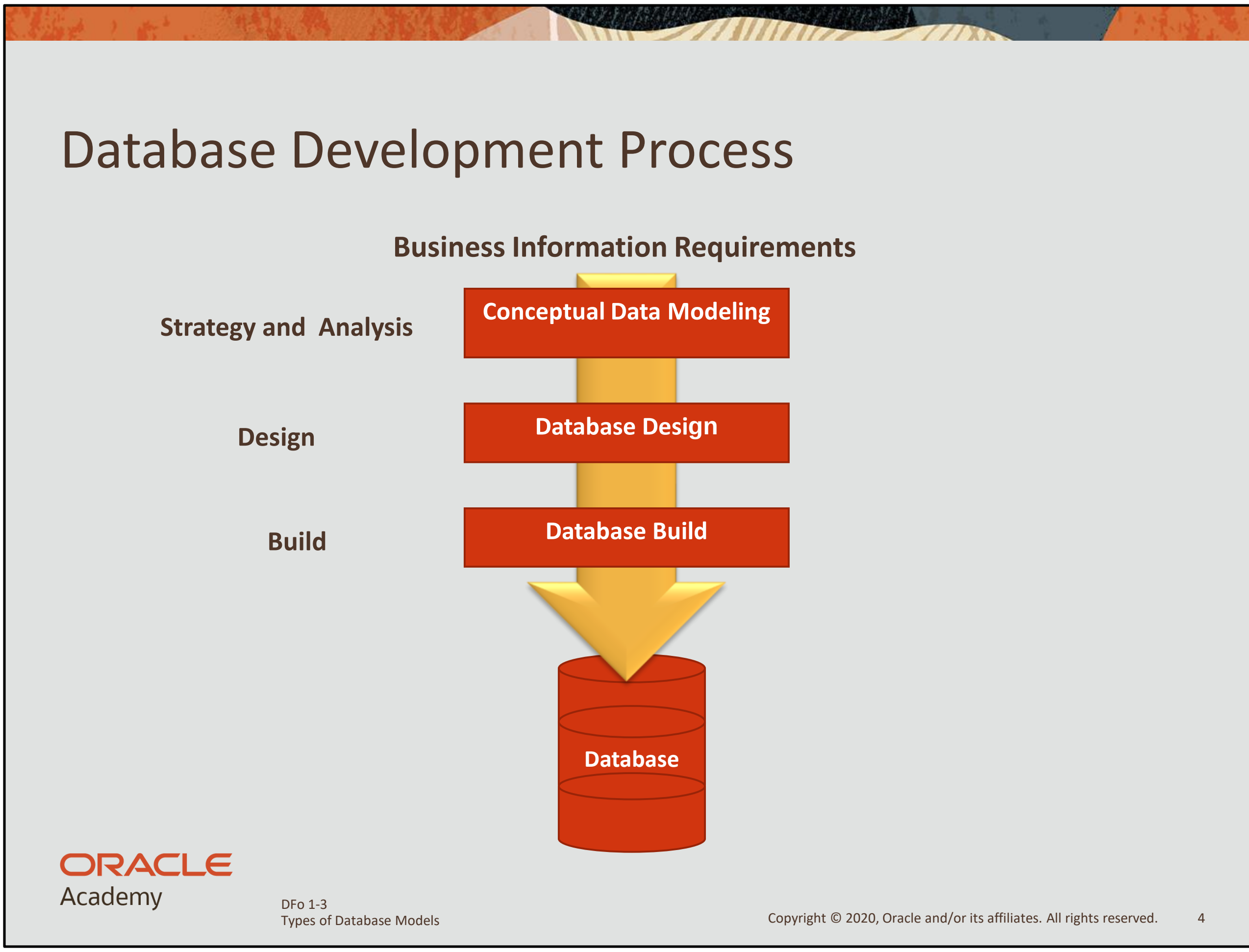
- Telecommunication departments store information about the communication network, telephone numbers, call details, and monthly bills in databases
- In finance and trading, databases are used for storing information pertaining to sales, purchases of stocks and bonds, or online trading
- Organizations use databases for storing information about their employees, salaries, benefits, taxes, and for generating paychecks
- Can you think of more uses for databases?

- Databases are used:
  - For keeping track of purchases on credit and debit cards, which helps generate monthly statements.
  - For integrating heterogeneous information sources for business-related activities, such as online shopping, booking of holiday packages, and doctor consultations.
  - In the health-care industry to maintain and track patient health care details.
  - In the area of digital publishing and digital libraries to manage and deliver textual and multimedia data.

Database Foundations

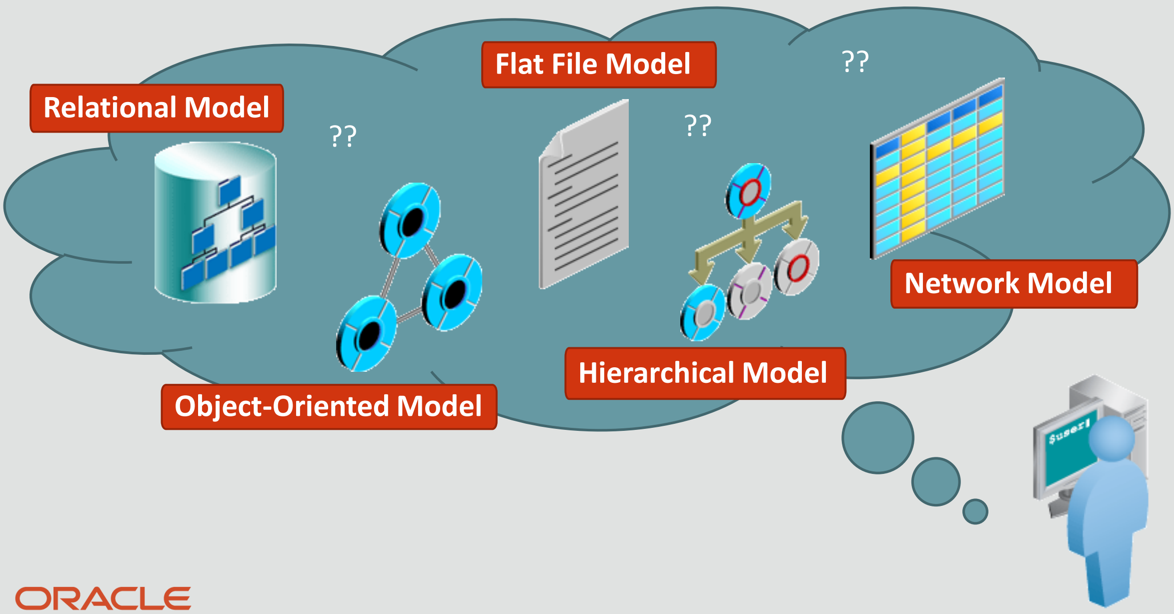
1-3
Types of Database Models

ORACLE
Academy

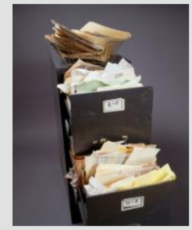


Data modeling is the first part of the database development process. Conceptual data modeling is the examination of a business and business data to determine the structure of business information and the rules that govern it. This structure forms the basis for database design. A conceptual model is relatively stable over long periods of time. Physical data modeling (or database building) is concerned with implementation in a given technical software and hardware environment. The physical implementation is highly dependent on the current state of technology and is subject to change as available technologies rapidly change.

Case Scenario: Types of Database Models



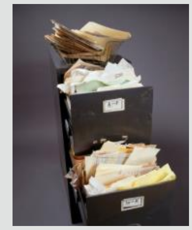
Flat File Model



- A flat file database is a database designed around a single table
- Flat file databases are generally in plain-text form, where each line holds only one record.
- The fields in the record are separated with delimiters, such as tabs and commas.

Flat files serve as a solution for simple database tasks. The flat file design puts all database information in one table, or list, with fields to represent all parameters. A flat file may contain many fields, often with duplicated data that is prone to data corruption.

Example of a Flat File Model



- Books as well as Authors are stored in this single table, causing repetition of data values

	AUTHOR_ID	AUTHOR_NAME	TITLE
Record 1	AD0001	Oscar Wilde	A Vision
Record 2	AD0002	Leo Tolstoy	War and Peace
Record 3	AD0003	Oliver Goldsmith	Citizen of the World
Record 4	AD0003	Oliver Goldsmith	The Deserted Village

Hierarchical Model

- In a hierarchical database model, the data is organized in a tree-like structure
- The data is stored as records that are connected to one another through links.
- A record is a collection of fields
- A record in the hierarchical database model corresponds to a row in the relational database model

ORACLE
Academy

DFo 1-3
Types of Database Models

 8

Each field contains only one value. The entity type of a record defines which fields the record contains.

A record in the hierarchical database model corresponds to a row in the relational database model. An entity type corresponds to a table.

In a hierarchical database model:

- Each child record has only one parent.
- A parent record can have one or more child records.

To retrieve data from a hierarchical database, the whole tree needs to be traversed, starting from the root node.

Advantages:

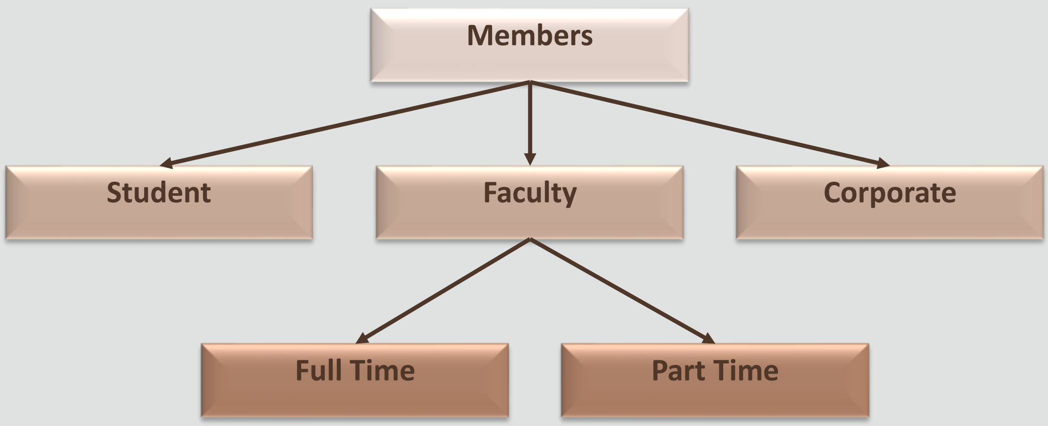
- Easy addition and deletion of new information
- Faster access to data at the top of the hierarchy

Disadvantages:

- Increased storage space
- Slower access to data at the bottom of the hierarchy

Example of a Hierarchical Model

- Data is organized in a tree-like structure and stored as records that are connected to one another through links



Hierarchical Model

- Data organized in a **tree-like structure**

- Data stored as **records** connected to one another through **links**
- A **record** corresponds to a **row** in the relational model

- A **record** is a **collection of fields**
- Each **field** contains only one **value**

- **Entity** type of **record** defines which **fields** the **record** contains
- An **entity** type corresponds to a **table** in the relational model

Hierarchical Model

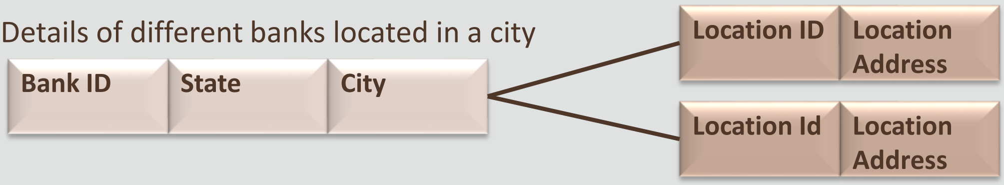
- Very efficient in many real world cases:
  - **Windows Registry** in Windows OS
  - **File systems**
  - Table of contents
- Simple but inflexible (only **one-to-many** relationship)
- sometimes: recognized as the first database model

Hierarchical Model - Database Example

- **`IBM Information Management System (IMS)`**
  - designed in **1966** for the **Apollo** program (`early mainframe DBMS`)
  - one of the most widely used commercial **hierarchical databases**
  - last stable release: **2017**

Network Model

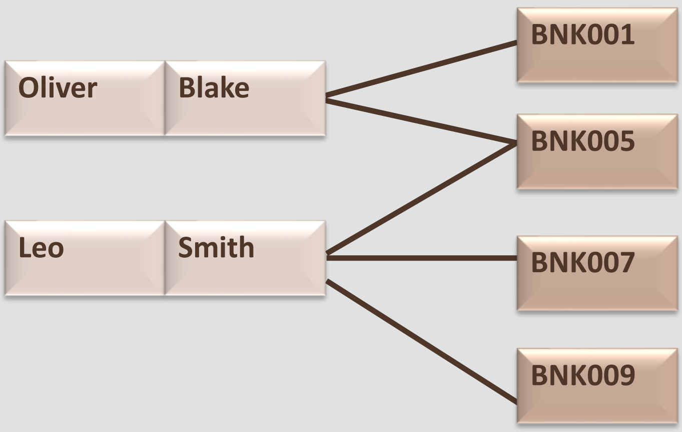
Details of different banks located in a city



- The network model is a database model that can be regarded as a flexible way of representing objects and their relationships
- A network database comprised of a collection of records connected to one another through links. (Boxes = Fields, Lines = Links)
- Each record is a collection of fields, each of which contains only one data value
- A link is an association between two records

In a network database model each record can have multiple parent and child records, forming a generalized graph structure. The network model enables a more natural way of modeling the relationship between records.

Example of a Network Model



- Oliver Blake holds accounts in two banks, BNK001 and BNK005
- Leo Smith holds accounts in three banks, BNK005, BNK007, BNK009

ORACLE Academy

The slide depicts an example of a network model that stores information about bank account details of different people. In the example, The records are connected to each other through links, represented by lines.

Network Model

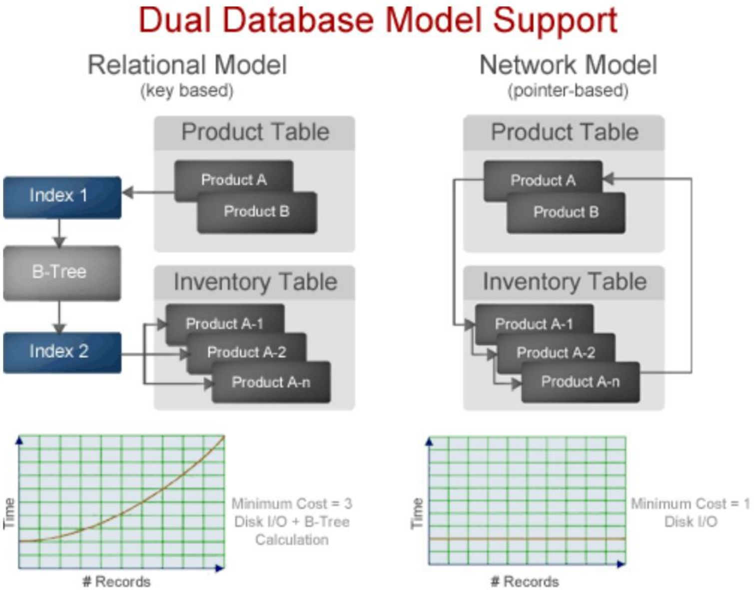
- Expands hierarchical model
  - allowing **many-to-many** relationships
  - tree-like structure but allowing **multiple parents**

- most popular before **relational model**
- defined by the **CODASYL** specs

- **Example: Raima Database Manager**
  - https://raima.com/
  - https://raima.com/network-model-vs-relational-model

Raima Performance Example

- Selecting appropriate data model or combining models
    - produce a far more efficient result



Disadvantages of Network Model

- Complex array of pointers
- System complexity limits efficiency
- Navigation needs complex implementation
- Structural changes require changes in all app programs
- Heavy pressure on developers due to the complex structure

Relational Model

- Data is represented as a collection of tables
- Each column represents attributes that belong to the table
- Each row represents an instance of the table
- Each table is the visual representation of columns and rows
- Every table has a field or a set of fields that uniquely identifies the row

Relational Model

- The order of the rows and columns is not important
- Every row is unique
- Each field can contain only one value
- Values within a column or field are from the same domain (datatype)
- Table names must be unique
- Column names within each table must be unique

ORACLE
Academy

Example of a Relational Model

EMPLOYEE

Foreign Key

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
100	Steven	King	90
101	Neena	Kochhar	90
102	Lex	De Haan	90
200	Jennifer	Whalen	10
205	Shelley	Higgins	110

refers to

DEPARTMENT

DEPARTMENT_ID	DEPARTMENT_NAME
10	Administration
20	Marketing
50	Shipping

Primary Key

In this example a relationship is created between the two tables using the common field of DEPARTMENT_ID

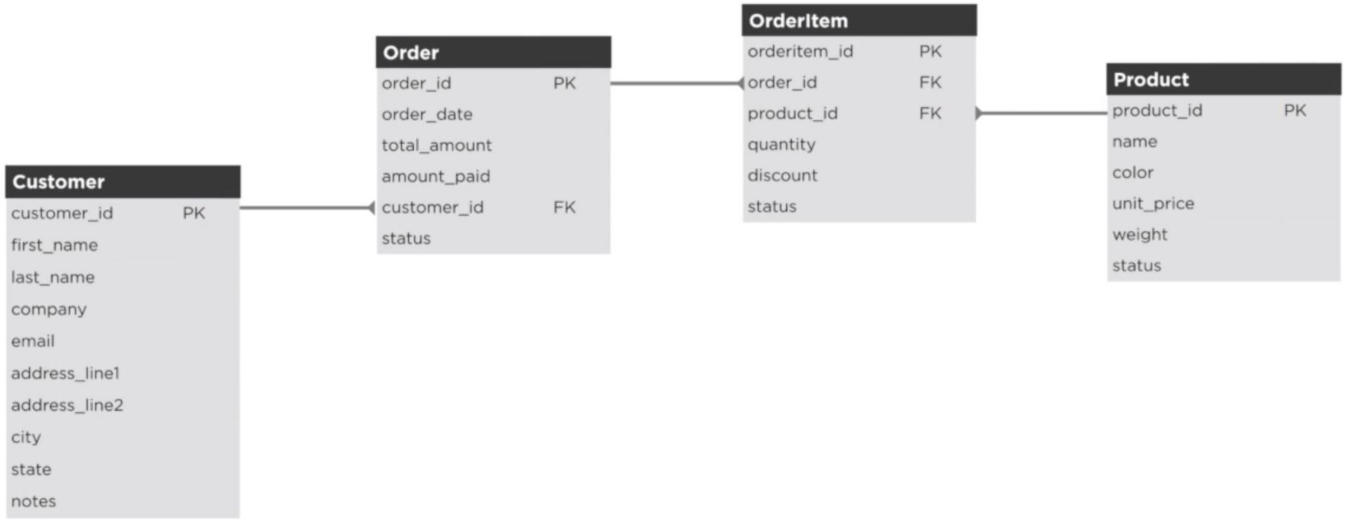
ORACLE Academy

DFo 1-3
Types of Database Models

 16

In the slide example, the Employees table contains a column that relates to the Department_ID from the Departments table. The inclusion of the Department_ID defines the relationship in the relational database model.

Database Schema



ACID Transactions

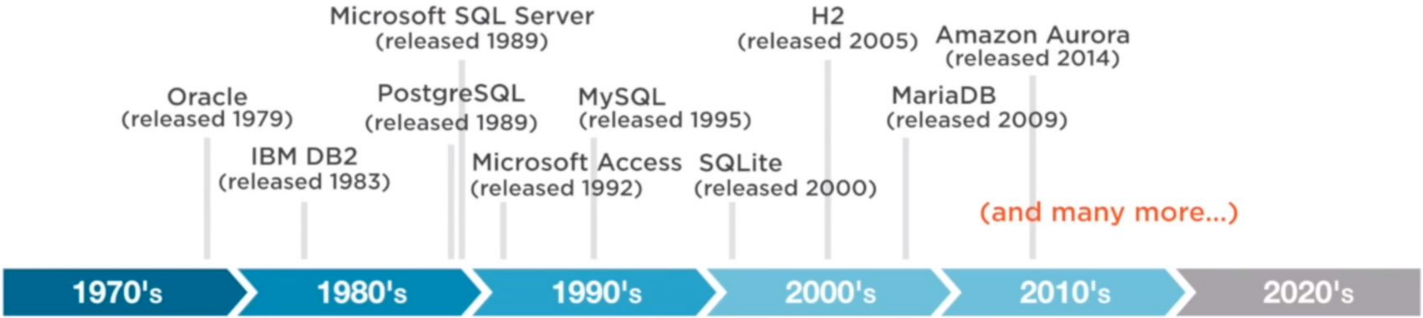
A | **Atomic**
We either do all of it, or none of it

C | **Consistent**
The DB was valid before, it's valid after

I | **Isolated**
No-one sees "half completed" results

D | **Durable**
When it's done, it stays done

Relational BDMS Product Timeline



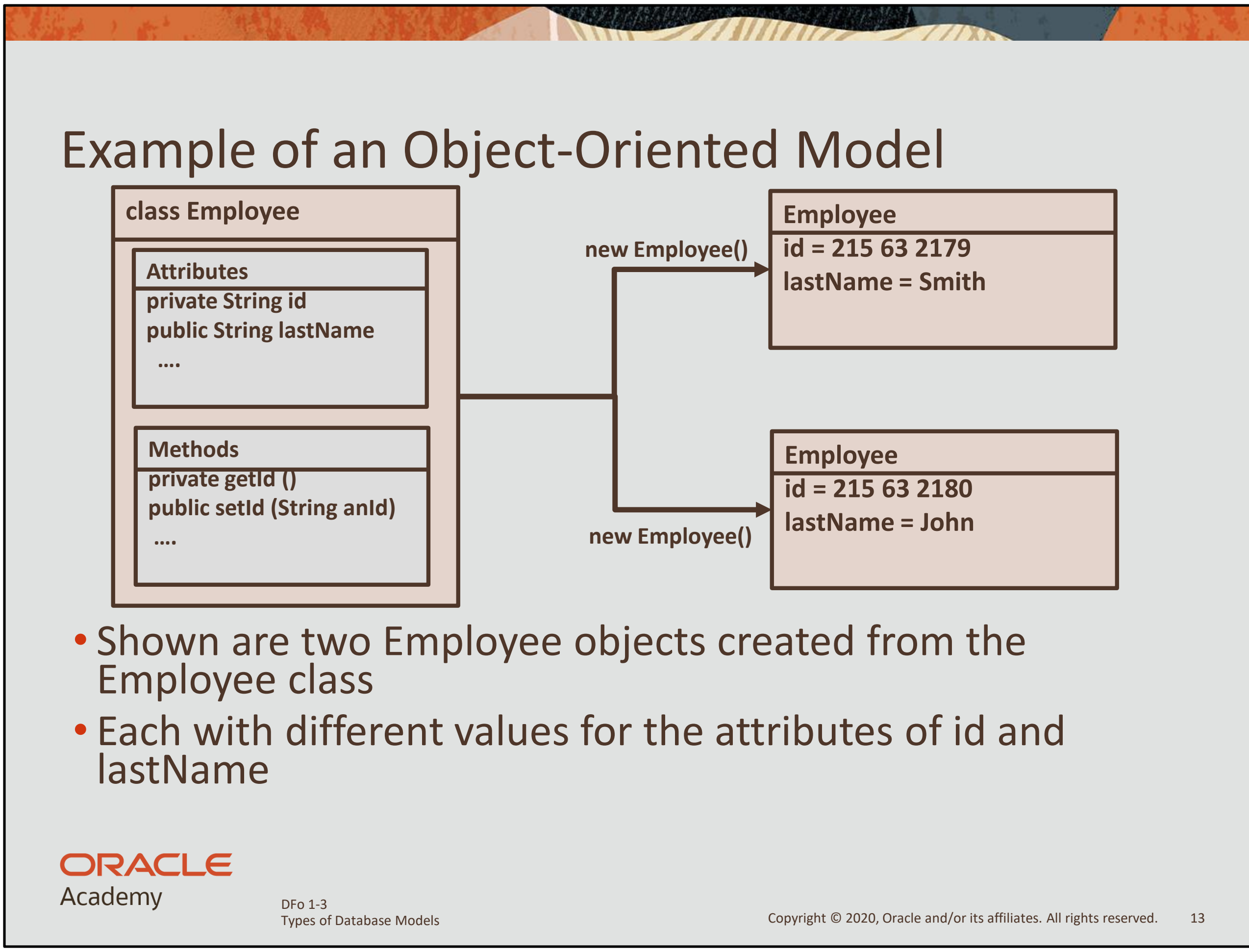
Object-Oriented Model

- A entity is modeled as an object
- Every object has a state (the set of values for the attributes of the object) and a behavior (the set of methods that operate on the state of the object)
- The relationship between the objects is through sharing of access
- An object must belong to only one class as an instance of that class
- You can derive a new class (subclass) from an existing class (superclass)

Advantages

- Reduced maintenance
- Real-world modeling
- High code reusability

Disadvantages: Many information application systems do not benefit from object-oriented modeling because it is best suited for dynamic, interactive environments.



The graphic in the slide shows an Employee class defined with two attributes:

- The id attribute is the employee identifier.
- The lastName attribute is the last name of the employee.

The Employee class has two methods:

- getId()
- setId(String anId)

The id attribute and the getId() method are private and therefore can be accessed only within the class. The lastName attribute and the setId(String anId) method are public and can be accessed by other classes, too.

When you create an instance, the attributes store individual and private information relevant only to the employee. The information contained within an employee instance is known only to that particular employee. Each instance of Employee holds its own state. You can access that state only if the creator of the class defines it in a way that gives you access.

Examples of Object-Oriented Model

- **ObjectDB**
  - https://www.objectdb.com
- **Objectivity/DB**
  - https://objectivity.com/objectivity-db
- **ObjectDatabase++**
  - http://tdbe.ekkysoftware.com/ODBPP

Disadvantages of Object-Oriented Model

- Poor adoption among the community
- Few database management systems have taken to object databases
  - lack of standardization
- Most developers prefer to use
  - more widely spread databases
  - well-documented databases
- In some cases, high complexity can cause performance problems

Object–Relational Mapping (ORM)

- library translates between OOP objects & relational databases
- RDBMS developers reply to Object-Oriented Database Model

The Marketplace of Database Solutions

Relational

Oracle Database

Microsoft SQL Server

IBM DB2

PostgreSQL

MySQL

(and SQLIte, Microsoft Access, FileMaker, H2, MariaDB, Amazon Aurora, etc.)

Non-Relational

MongoDB

Redis

Cassandra

Neo4J

(and CouchDB, Bigtable, HBase, Couchbase, Amazon Dynamo, etc.)

> Why choose a NoSQL solution instead of a relational database?

or "in addition to"

Common considerations:

Speed	Scaling	Cost

NewSQL

Adoptfrom RDBMS solutions:

the **relational model,**

ACID transactions and **SQL queries**

Take from NoSQL solutions:

the **performance**, **scaling** and **distribution** options

Some Notes

- **Dimensional Database Model**
  - type of relational model optimized to quickly retrieve data from **data warehouse**
  - used in **Business Intelligence (BI)**

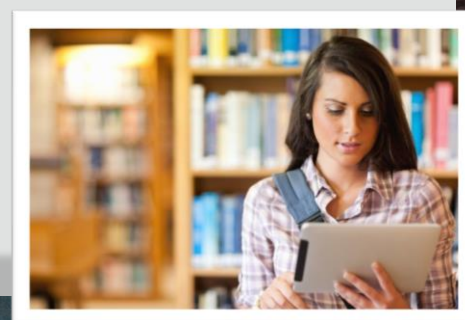
- **NoSQL Databases**
  - Not Only SQL

- **NewSQL Databases**
  - Example: **CockroachDB** https://www.cockroachlabs.com/

Database Foundations

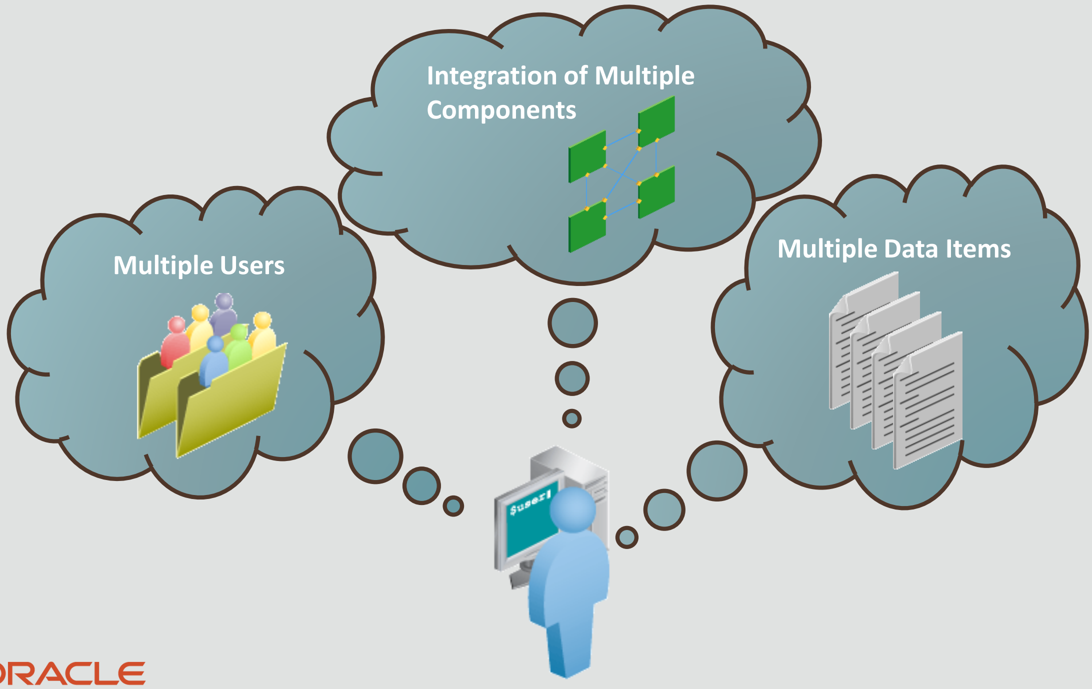
1-4

Business Requirements



ORACLE Academy

Why Do I Need a Database Solution?



ORACLE Academy

DFo 1-4
Business Requirements

 4

- The basic component of a file in a file system is a data item. Examples of data items in the real world are last name, first name, street address, and employee ID.
- A database is a more complex object. It is a collection of interrelated stored data that must meet the needs of many users. A database must also adhere to the business rules and processes of the organization.
- Advantages of using a database rather than a simple file system:
  - Availability of data to a diverse group of users
  - Integration of data for easier access and modification when performing complex transactions
  - Data integrity and reduced data redundancy

Case Scenario: Need a Database Solution

	STUDENT_ID	SPORT_1	PRICE_1	SPORT_2	PRICE_2
Record 1	ST0001	Tennis	$100	Badminton	$150
Record 2	ST0002	Soccer	$175	Tennis	$100
Record 3	ST0003	Cycling	$200	Badminton	$150
…….	…….	…….	…….	…….	…….

This is a flat file that stores information about students, the sports that they selected, and the price for each sport that they selected. This scenario warrants the need for a **relational** database

ORACLE
Academy

The flat file depicted in the slide shows redundant information about each student. Modifications needed include identifying each student uniquely and eliminating recurring columns/fields.

Case Scenario: Possible Database Solution

Student Details Table

ID	FIRST_NAME	LAST_NAME
ST0001	Sean	Smith

Sport Details Table

ID	NAME	PRICE
TN001	Tennis	$100

Participant Details Table

STUDENT_ID	SPORT_ID	SEMESTER_ DETAILS
ST0001	TN001	Fall2017

Flat file was split into three tables eliminating issues related to:

- Redundancy
- Data entry anomalies
- Inconsistency