DBMS-RDBMS

Fresher Learning Program
January, 2012

People matter, results count.

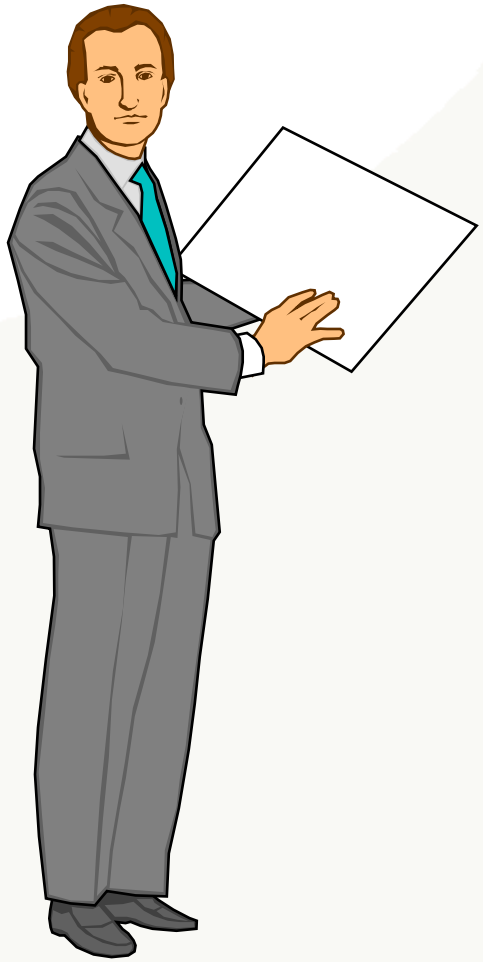
Objectives of DBMS - RDBMS

- Purpose:
  - To cover the Database and Relational database concepts.
- Product:
  - Understand the concepts of DBMS - RDBMS
  - Understand the architecture and various data models
  - To be able to understand the Rational Data Model
  - To be able to design the database at the basic level
- Process:
  - Theory Sessions followed by couple of assignments
  - A review at the end of the session and a Quiz.

Table of Contents

Introduction to database

- **Database**
  - Organized collection of related data which captures the essential properties of objects and records the relationships among them.
- **Database system**
  - It is an integrated collection of related files, along with details of interpretation of data.
- **Database Management System (DBMS)**
  - A set of computer programs for organizing the information in a database.
  - A DBMS supports the structuring of the database in a standard format and provides tools for data input, verification, storage, retrieval, query and manipulation.



Flat file – A Simple Database

Flat files are stored at operating system level. Made up of set of strings in one or more files that can be parsed to get the information they store. They are good for storing simple lists and data. Can get complicated when tried to replicate more complex data structures.

WHAT'S GOOD

- All operating systems come with an inbuilt file system.
- Standardized formats that can be read across platforms.

WHAT'S NOT

- Near absence of access concurrency.
- Possibility of data corruption.
- Needle-in-the-haystack method of finding data.
- Data is not structured and can not be easily related
- Usually necessary to store redundant data.

Why DBMS?

- **DBMS**

Allows to store information as binary strings, a key, which is used to find the associated value. It provides a way for end-users to use the database without having to worry about how the data is stored or how to retrieve the values.

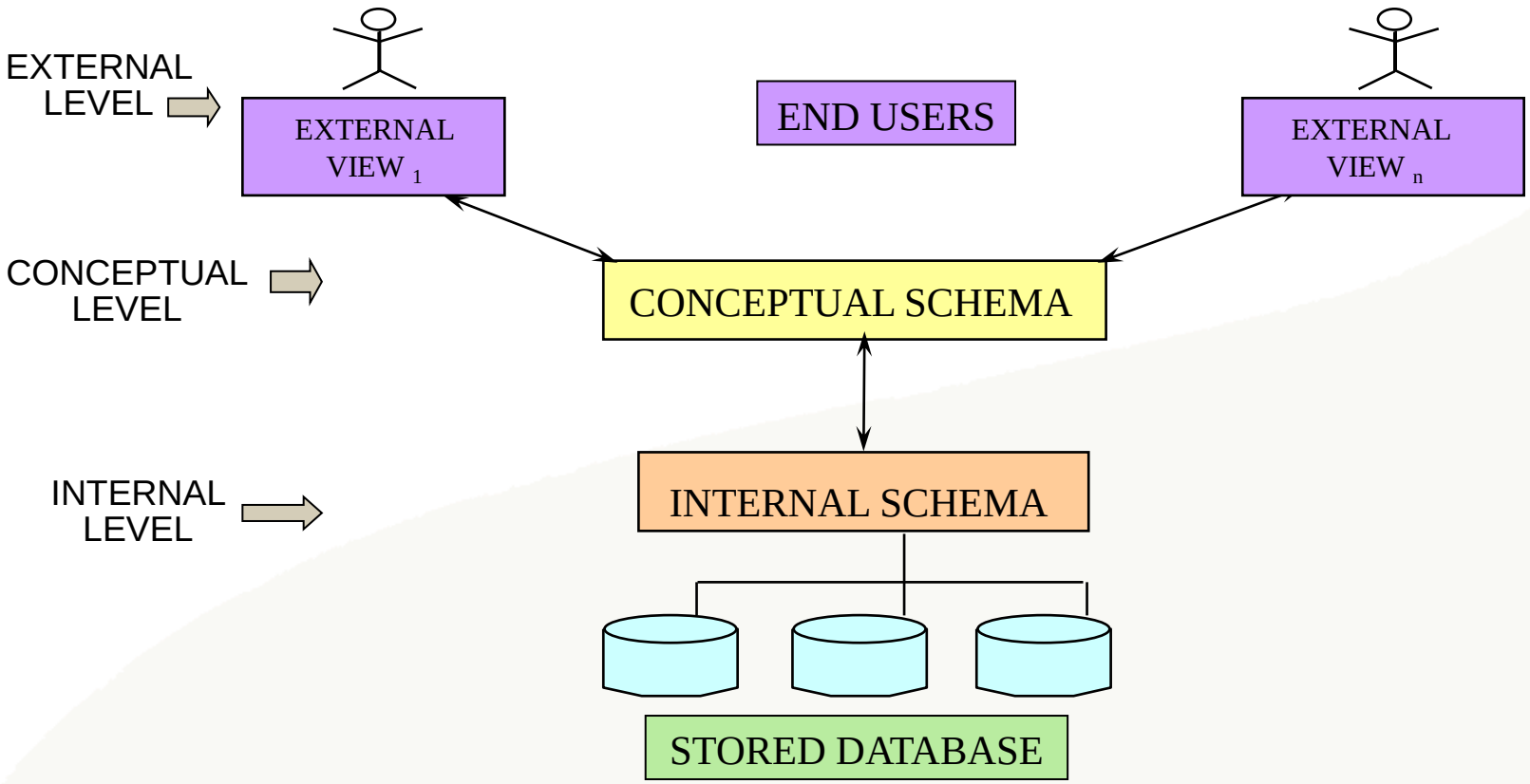
Advantages of DBMS over Files:

- Allows users to define, create, manipulate, store, maintain, retrieve, and process the data in a database in order to produce meaningful information.
- Focus on information representation.
- Data stored as records in various database files that can be combined to produce meaningful information for users.
- DBMS controls all functions of capturing, processing, storing, retrieving data and generates various forms of data output.
- Manages access by multiple users and multiple programs to a common store of data.

DBMS Characteristics

- **Data persistence –**Stored on some hardware and persist after access.
- **Concurrency –**Multi user access
- **Managed Transactions** – Data manipulation and saving of work
- **Query language –** for retrieval of data
- **Data recoverability –** To recover from failure

Three Level Architecture for DBMS



Advantages and Disadvantages of DBMS

- Advantages
  - Reduction of redundancies.
  - Shared data
  - Data integrity
  - Data security
  - Conflict resolution
  - Data independence

- Disadvantages
  - Problems associated with centralisation.
  - Cost of hardware / software and migration
  - Complexity of backup and recovery

Data Model

- Set of concepts used to describe the structure of a Database.
- The structure of a Database means the Data types, Relationships, and constraints that should hold for the data.
- Data model is defined to consist of a combination of 3 components:
  - Collection of data object types, which form the basic building blocks for any database that conforms to the model.
  - A collection of general integrity rules, which constrain the set of occurrences of those object types that can legally appear in any such database.
  - A collection of operators, which can be applied to such object occurrences for retrieval and other purposes.

Components of Data Modeling

- **Entity**:
  - An object with a Physical existence - a person, car, an employee - or it may be an object with a Conceptual existence - a company, a job or a university course.

- **Attributes**:
  - Properties that describe an Entity.
  - For example: An employee entity may be described by the employee's name, age, address etc.

- **Values**:
  - A particular Entity will have a Value for each of its attributes.
  - The attribute values that describe each entity become a major part of the data stored in the database.

Components of Data Modeling – Continued

- **Domains:**
  - Specifies the set of values that may be assigned to that attribute for each individual entity.

- **Keys:**
  - An attribute whose value is unique or distinct for each entity and its values can be used to identify each entity uniquely.
  - For example: For the entity “Person”, the Social Security Number is the Key.

- **Relationships:**
  - Whenever an attribute of one entity type refers to another entity type, a relationship exists.
  - A Relationship among N entities defines a set of associations among entities from these types.

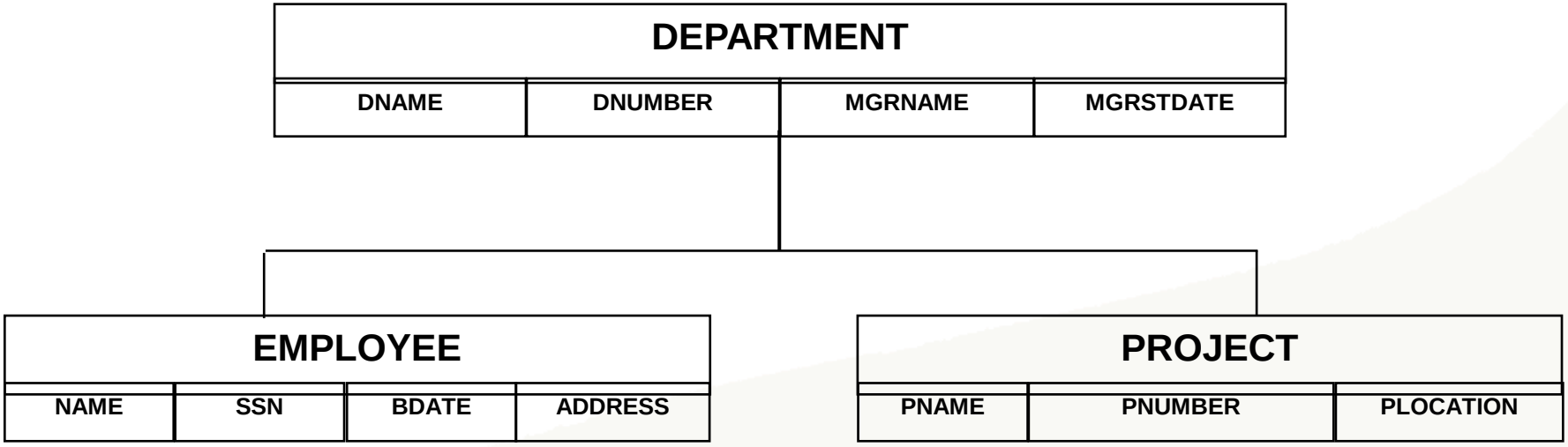
Types of Database Models

- Hierarchical database model
- Network database model
- Object Oriented database model
- Relational database model

Hierarchical model

- The hierarchical model uses the tree concept to represent the data and the relationship among data
- A Hierarchical Data Model makes use of 2 main data structuring concepts :
  - RECORDS : A Record is a collection of field values that provide information on an Entity. Records of the same type are grouped in Record Types.
  - PARENT CHILD RELATIONSHIPS **:** is a **1:N** relationship between two Records Types. The record type on 1side is the **Parent** and the record on the N side is the **Child.**
- Deleting a parent record occurrence requires deleting all its children record occurrences.

Hierarchical model – continued



PARENT : DEPARTMENT
CHILD : EMPLOYEE

PARENT : DEPARTMENT
CHILD : PROJECT

Network model

- Extends the hierarchical model by allowing a record to participate in multiple parent/child relationship. It depicts data logically as many-to-many relationships.
- These relations are known as Sets.
- User can access data starting from any tables in the structure.
- Data is accessed very quickly.

Object Oriented Database Models

- **Object Oriented DBMS :**Stores data and procedures as objects that can be retrieved and shared automatically**.**

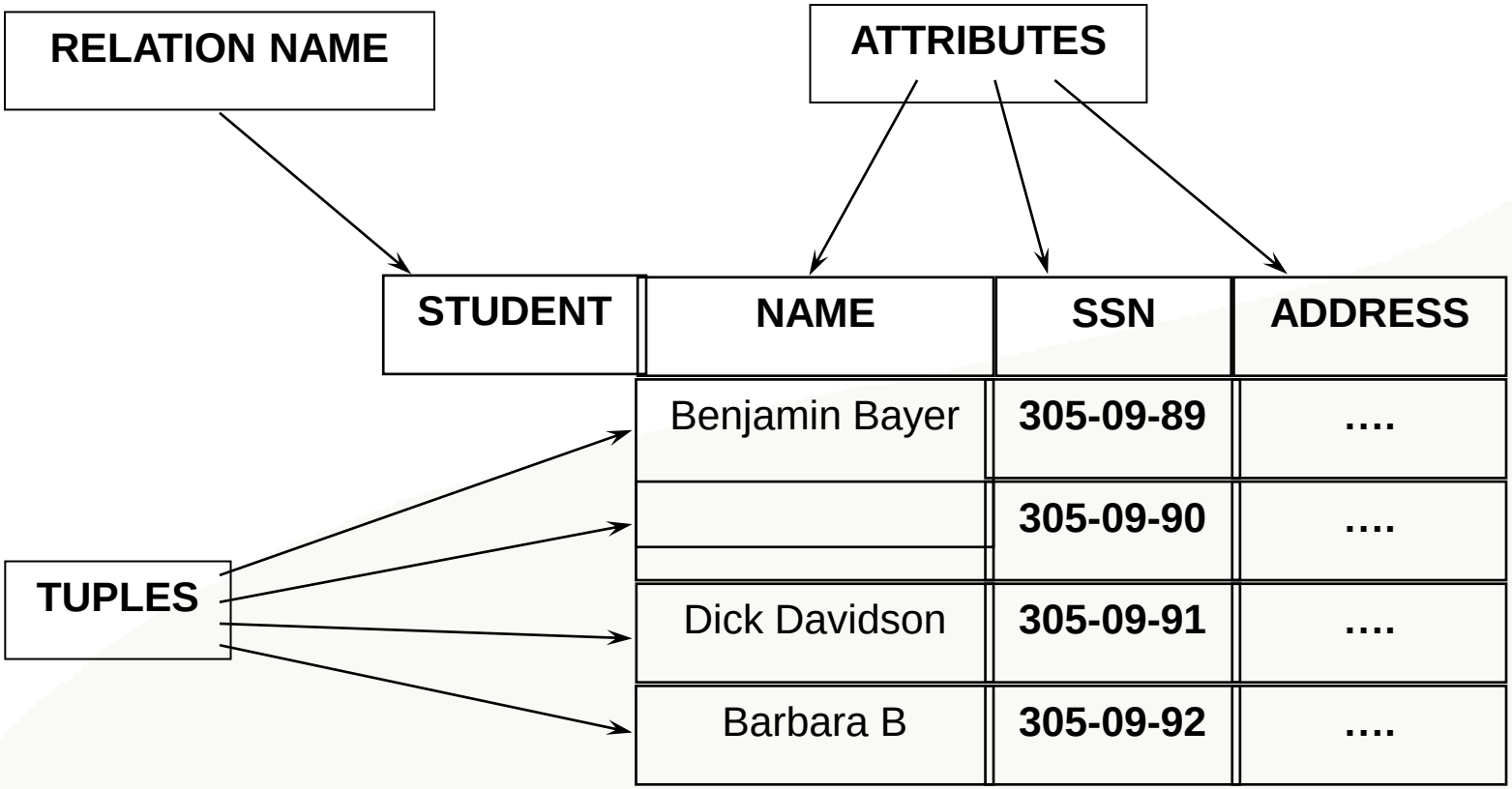
- **Object-relational DBMS :**Provides capabilities of both object-oriented and relational DBMS**.**

Relational Database model

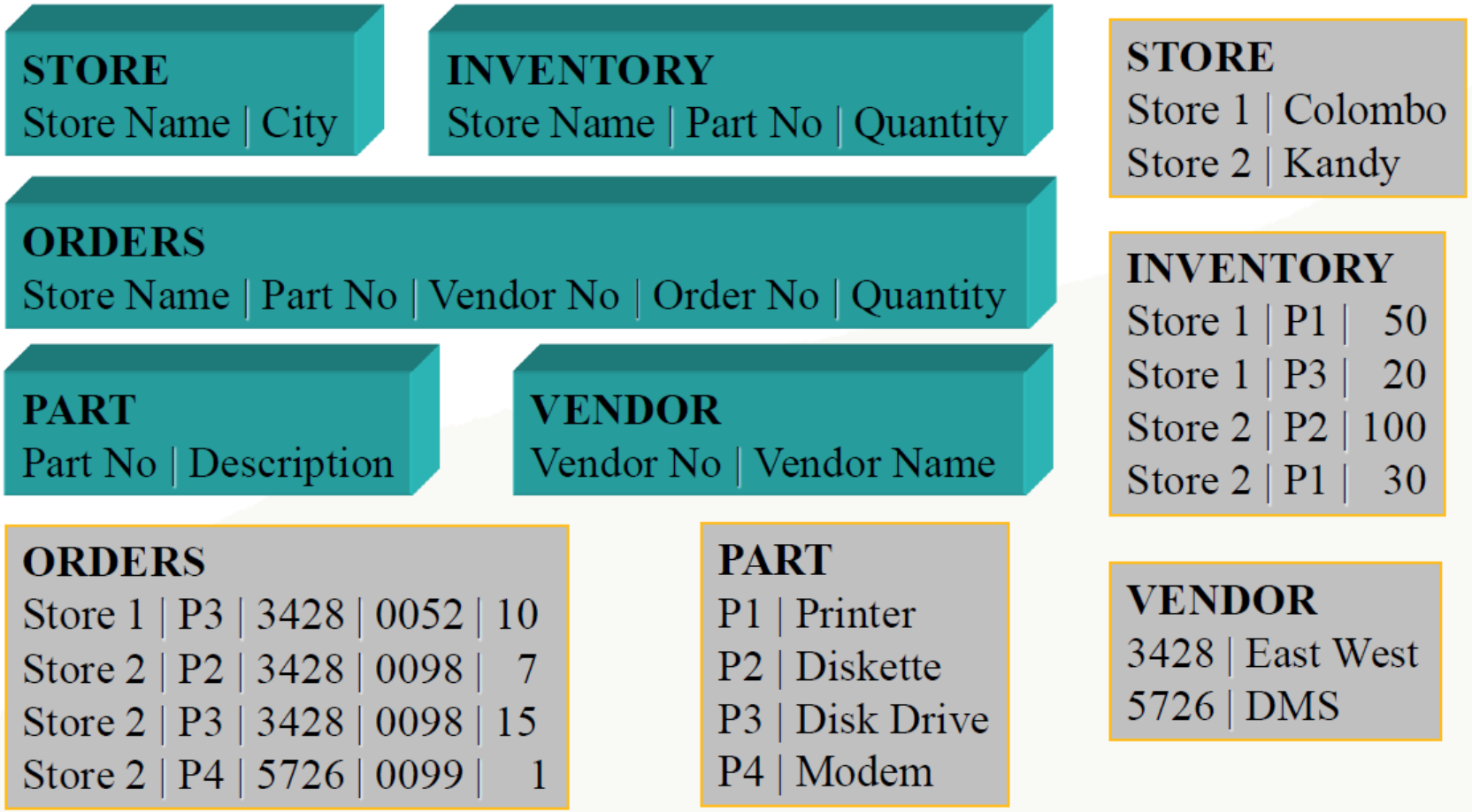
- **A DBMS is said to be a Relational Data Base Management System (RDBMS) if it follows the following main criteria:**

- Represents data as two dimensional tables called relations .Data is organized as tables of data values.
- Primary unit of storage is called table which contains rows and columns.
- Relationships are maintained by common data values stored in the related tables. (No explicit pointers are stored like in Hierarchical/network model).
- Tables can be related to one another through common column values called keys.
- Should be able to retrieve the data stored in such tables using SELECT, JOIN and PROJECT operators.

Relational Database Model – continued



Example : Relational Database



CODD's Rules

Codd's twelve rules are a set of thirteen rules (numbered zero to twelve) proposed by Edgar F. Codd, a pioneer of the relational model for databases, designed to define what is required from a database management system in order for it to be considered relational, i.e., a relational database management system (RDBMS).

- ***Rule (0):***

  The system must qualify as relational, as a database, and as a management system. For a system to qualify as a relational database management system (RDBMS), that system must use its relational facilities (exclusively) to manage the database.

- ***Rule 1: The Information Rule***
  - All data should be presented to the user in table form.

- ***Rule 2: Guaranteed Access Rule***
  - All data should be accessible without ambiguity. This can be accomplished through a combination of the table name, primary key, and column name.

CODD's Rules – continued

- ***Rule 3: Systematic Treatment of Null Values***
  - A field should be allowed to remain empty.
  - This involves the support of a null value, which is distinct from an empty string or a number with a value of zero.
  - This rule should not apply to primary keys.
  - In addition, most database implementations support the concept of a nun- null field constraint that prevents null values in a specific table column.

- ***Rule 4: Dynamic On-Line Catalog Based on the Relational Model***
  - A relational database must provide access to its structure through the same tools that are used to access the data.
  - This is usually accomplished by storing the structure definition within special system tables.

- ***Rule 5: Comprehensive Data Sublanguage Rule***
  - The database must support at least one clearly defined language that includes functionality for data definition, data manipulation, data integrity, and database transaction control.
  - All commercial relational databases use forms of the standard SQL (Structured Query Language) as their supported comprehensive language.

CODD's Rules – continued

- ***Rule 6: View Updating Rule***
  - Data can be presented to the user in different logical combinations, called views.
  - Each view should support the same full range of data manipulation that direct-access to a table has available.
  - In practice, providing update and delete access to logical views is difficult and is not fully supported by any current database

- ***Rule 7: High-level Insert, Update, and Delete***
  - Data can be retrieved from a relational database in sets made from multiple rows and/or multiple tables.
  - This rule states that insert, update, and delete operations should be supported for any retrievable set rather than just for a single row in a single table.

- ***Rule 8: Physical Data Independence***
  - The user is isolated from the physical method of storing and retrieving information from the database.
  - Changes can be made to the underlying architecture (hardware, disk storage methods) without affecting how the user accesses it.

CODD's Rules – continued

- ***Rule 9: Logical Data Independence***
  - How a user views data should not change when the logical structure (tables structure) of the database changes.
  - Most databases rely on strong ties between the user view of the data and the actual structure of the underlying tables

- ***Rule 10: Integrity Independence***
  - The database language (like SQL) should support constraints on user input that maintain database integrity. This rule is not fully implemented by most major vendors. At a minimum, all databases do preserve two constraints through SQL.
  - No component of a primary key can have a null value. (see rule 3)
  - If a foreign key is defined in one table, any value in it must exist as a primary key in another table.

CODD's Rules – continued

- ***Rule 11: Distribution Independence***
  - A user should be totally unaware of whether or not the database is distributed (whether parts of the database exist in multiple locations).

- ***Rule 12: Nonsubversion Rule***
  - There should be no way to modify the database structure other than through the multiple row database language (like SQL).
  - Most databases today support administrative tools that allow some direct manipulation of the data structure.

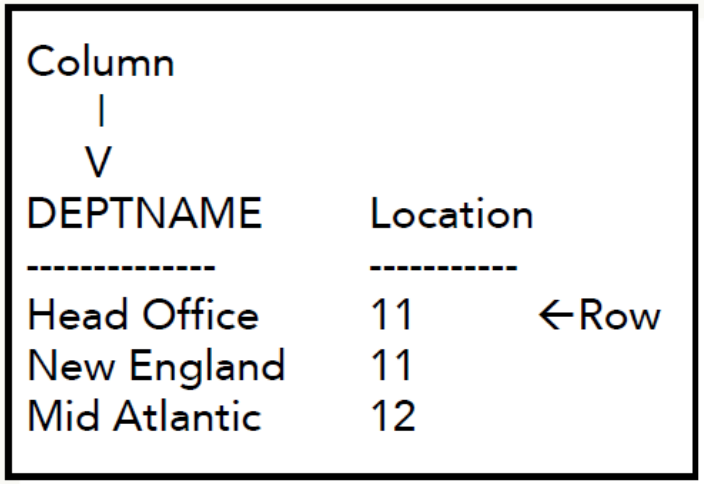
RDBMS Concepts

- What is a relation?

  Describes how elements of two sets relate/correspond to each other.
- Table –Primary unit of storage. Has set of named attributes called columns and data in rows
- Each column in a table has a column name, which is usually written as a heading at the top of the column
- The columns of a table have a left-to-right order, which is defined when the table is first created
- Unlike the columns, the rows in a table do not have any particular order

Characteristics of Relational Database

- Data stored under column is of same data type.
- Each column in a table has a unique name.
- A table will have at least one column
- Table can have any number of rows
- A typical table:



The 3 basic operations in an RDBMS

- **Select :** Creates subset of rows that meet specific criteria.
- **Join :** Combines relational tables to provide users with information.
- **Project :** Enables users to create new tables or retrieve rows containing only relevant information.

Advantages of RDBMS

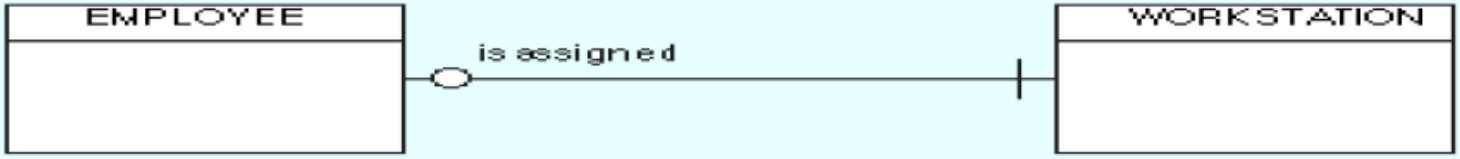
- Data is accessed quickly.
- Database structure is easy to change.
- Data is represented logically. Simple view of the data that represents the data used in business.
- Easy to implement data integrity.
- Easy to develop and modify application programs.
- A standard language SQL is used to create the structure of database, manipulate data and retrieve data.

Entity Relationships

- **Entity** –A business element like Product,Customer
- **Relationship** –Association between entities
- **Attribute** –Characteristic of a entity eg. Employee Name
- **One-to-one relation** – Every value in one entity is related to one and only one value in related entity.
- **One-to-many** –One occurrence of data in one entity might have one or more occurrences in related entity.
- **Many-to-many** – Multiple occurrences of related data.

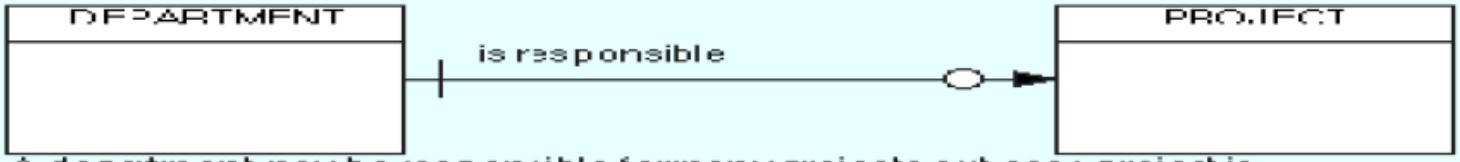
Entity Relationships

A. ONE -TO- ONE



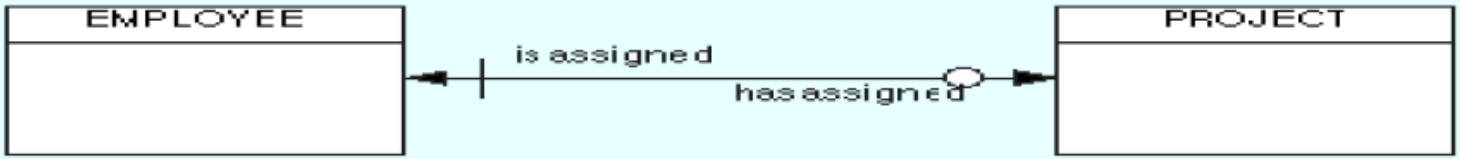
Every employee is assigned one workstation; not all workstations are assigned to employees.

B. ONE-TO-MANY



A department may be responsible for many projects but each project is the responsibility of one department

C. MANY-TO-MANY



Employees may be assigned to many projects; every project has assigned at least one employee

KEYS

- **Primary Key –**A column or combination of columns which uniquely identify a row in a table. The primary key has a different unique value for each row in a table, so no two rows of a table with a primary key are exact duplicates of one another
- **Composite Key –** Primary key made of multiple columns

Employee

E-No	E-Name	D-No
179	Hari	7
857	Jack	4
342	Cat	7

Primary Key

Salary

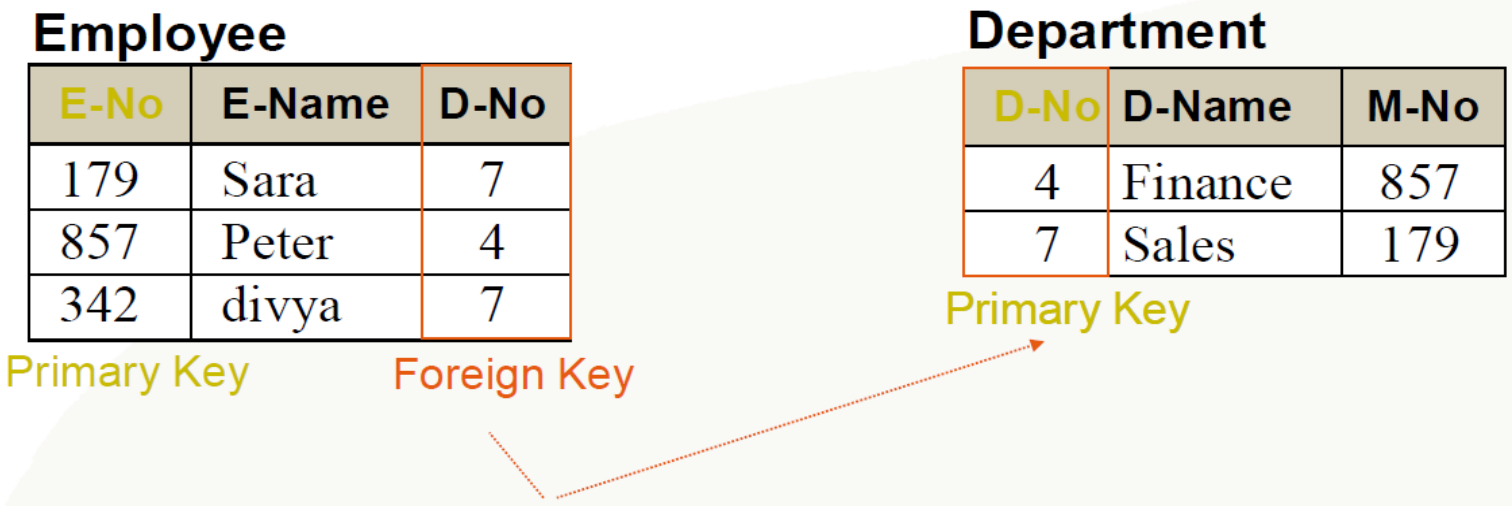
E-No	Eff-Date	Amt
179	1/1/98	8000
857	3/7/94	9000
179	1/6/97	7000
342	28/1/97	7500

←Composite→
Primary Key

KEYS Continued

- **Foreign Key** – A column in one table whose value matches the primary key in some other table . It can be a combination of columns also.

  A table can contain more than one foreign key if it is related to more than one other table.



Integrity

- **Entity Integrity –**Primary key cannot be NULL or empty.(a null value is one that has no value and is not the same as a blank).
- **Data Integrity –**Consistency, accuracy, and correctness of data stored in a database
- **Referential Integrity –**Foreign key must be a null or match value of primary key it references.
- **Self referential integrity –** A foreign key references the primary key of the same table.

Database designing –How to go about it?

- **Conceptual design :** Abstract model of the database from the business perspective.
- **Physical design :** Detailed description of business information needs.
- **Entity-Relationship diagram :** Methodology for documenting databases illustrating relationships between the database entities.
- **Normalization :** Process of creating small, stable data structures from complex groups of data.

Normalization

- Normalization – It refers to the set of rules that help designers to reduce redundancy and avoid data update anomalies.
- Advantages of Normalization:
  - Redundant data is reduced: Multiple copies of data not maintained.
  - Data integrity is maintained
  - Easy to understand –Look at a big picture and make it simple by grouping similar things together.

Normal Forms

- Normal forms give the measure of the level or depth to which a database can be normalized.
- Normal Forms –
  - First Normal Form
  - Second Normal Form
  - Third Normal Form
  - Fourth Normal Form
  - Fifth Normal Form

The most commonly used normal forms are up to Third Normal Form

First Normal Form

- Divide base data into logical units of tables.
- Values in the columns should be atomic - there are no repeating groups, all the key attributes are defined, all attributes are dependent on the primary key.

Supp_No	Supp_Name	Supp_City	Supp_Pincd	Part_No	Part_Name	Part_Colour	Qty
S1	Supplier A	PUN	001	P1	Part A	Red	10
S1	Supplier A	PUN	001	P2	Part B	Blue	20
S1	Supplier A	PUN	001	P3	Part C	Green	10
S1	Supplier A	PUN	001	P4	Part D	Red	10
S1	Supplier A	PUN	001	P5	Part E	Green	30
S1	Supplier A	PUN	001	P6	Part F	Blue	10
S1	Supplier A	PUN	001	P7	Part G	Red	20
S1	Supplier A	PUN	001	P8	Part H	Red	10
S1	Supplier A	PUN	001	P9	Part A	Blue	20
S2	Supplier B	PUN	001	P2	Part B	Blue	10
S2	Supplier B	PUN	001	P4	Part D	Red	20
S2	Supplier B	PUN	001	P6	Part F	Blue	30
S2	Supplier B	PUN	001	P8	Part H	Red	10
S3	Supplier C	DEL	002	P1	Part A	Red	25
S4	Supplier D	DEL	002	P3	Part C	Green	15
S4	Supplier D	DEL	002	P5	Part E	Green	15

Second Normal form

- Eliminate all columns that depend only on part of the key i.e, take data that is partly dependent on primary key and enter into another table.(Eliminating partial dependencies)

Supp_No	Part_No	Qty
S1	P1	10
S1	P2	20
S1	P3	10
S1	P4	10
S1	P5	30
S1	P6	10
S1	P7	20
S1	P8	10
S1	P9	20
S2	P2	10
S2	P4	20
S2	P6	30
S2	P8	10
S3	P1	25
S4	P3	15
S4	P5	15

Supp_No	Supp_Name	Supp_City	Supp_Pincd
S1	Supplier A	PUN	001
S2	Supplier B	PUN	001
S3	Supplier C	DEL	002
S4	Supplier D	DEL	002

Part_No	Part_Name	Part_Colour
P1	Part A	Red
P2	Part B	Blue
P3	Part C	Green
P4	Part D	Red
P5	Part E	Green
P6	Part F	Blue
P7	Part G	Red
P8	Part H	Red
P9	Part A	Blue

Third Normal Form

- No transitive dependencies –Transitive dependency is where a non key attribute is dependent on another non key attribute
- Eliminate columns that do not depend on the key at all (transitive dependency)

Supp_No	Supp_Name	Supp_City
S1	Supplier A	PUN
S2	Supplier B	PUN
S3	Supplier C	DEL
S4	Supplier D	DEL

Supp_City	Supp_Pincd
PUN	001
DEL	002

Denormalization

- Denormalization –The process of decomposing the normalized data to put one fact into numerous places
- Need for Denormalization –
  - Performance gains
  - Example-If a application needs to join some tables on regular basis and join cost is prohibitive, then these tables can be denormalized to a single table of pre joined data for query efficiency.
- Note –Denormalization is not normally done unless there are compelling reasons in terms of performance

Transactions

- Transaction:
  - A Transaction is a logical Unit of Work (LUW)
  - A set of SQL statements that either successfully executes to completion, or does not execute at all, leaving the database in a consistent state, is called as a transaction.
  - Play an important role in the data consistency
  - Should be atomic – Complete success or failure
  - Multiple transactions should not access the same data at the same time – Isolation levels
  - Updates made by transactions are permanent

All transactions should satisfy the ACID properties. (Next slide)

ACID Properties

- The concept of **A C I D**:
  - **A**tomicity: Results of a transaction's execution are either all committed or all rolled back. All changes take effect, or none do.
  - **C**onsistency: The database is transformed from one valid state to another valid state
  - **I**solation: The results of a transaction are invisible to other transactions until the transaction is complete.
  - **D**urability: Once committed (completed), the results of a transaction are permanent and survive future system and media failures.

RDMS in the Market

Following are some of the leading RDBMS available in the market

- Oracle (by Oracle Corp)
- DB2, UDB, Informix (IBM)
- SQL Server (Microsoft).

Apart from these there are a few open source RDBMS like Postgre, Green plum etc.

Recap

Database Management System (DBMS)

Data Model – Entity , Attribute , Keys

Integrity

Relational Database Model (RDBMS)

CODD's Rules

Entity Relationships

NOrmalization

ACID Properties

Thank You For Your Time

People matter, results count.