UNIT - 2
Introduction of RDBMS and ERM

Introduction of RDBMS

- RDBMS stands for Relational Database Management System.

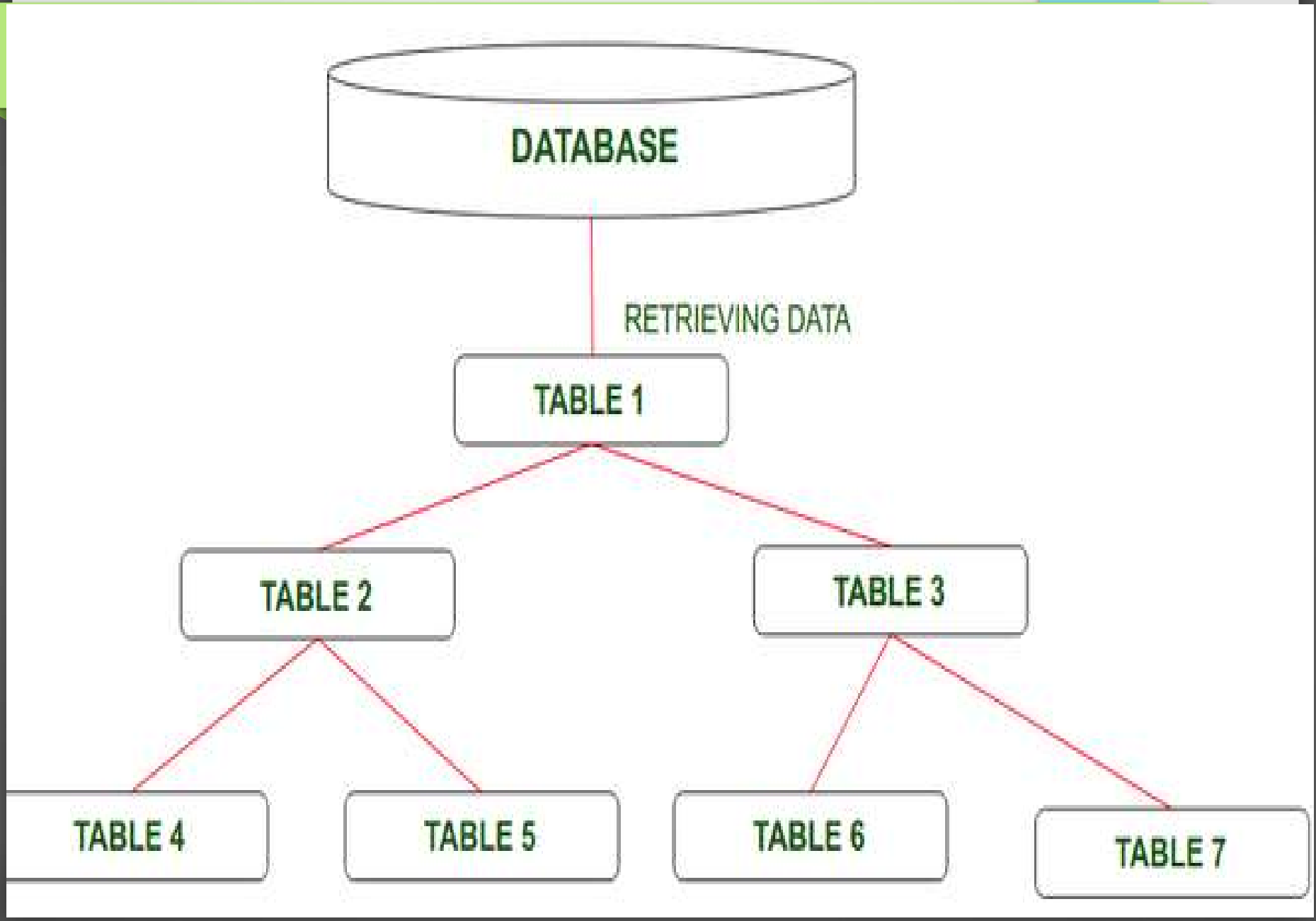
- Most popular database system.
- Simple and sound theoretical basis.
- The model is based on tables, rows and columns and the manipulation of data stored within.
- Relational database is a collection of these tables.
- First commercial system: MULTICS in 1978.
- Has overtaken Hierarchical and Network models.
- Main feature: Single database can be spread across several tables.
- Examples include: Oracle, IBM's DB2, Sybase, MySQL & Microsoft Access.

Characterstics Of RDBMS

- RDBMS is design to handle frequently changing data.
- It often used in transaction oriented application.
- RDBMS can store vast amount of historical data,which can later analysed.
- The RDBMS is configured as per application basisand a unique schema exist to support each application.
- RDBMS features ensures data intigrity.

A Logical View of Data

- Relational database model's structural and data independence enables us to view data *logically* rather than *physically*.
- The logical view allows a simpler file concept of data storage.
- The use of logically independent tables is easier to understand.
- Logical simplicity yields simpler and more effective database design methodologies.


DATABASE
RETRIEVING DATA
TABLE 1
TABLE 2
TABLE 3
TABLE 4
TABLE 5
TABLE 6
TABLE 7

Keys

- Each row in a table must be uniquely identifiable
- **Key** is one or more attributes that determine other attributes

Keys

TABLE 3.3 Relational Database Keys

KEY TYPE	DEFINITION
Superkey	An attribute (or combination of attributes) that uniquely identifies each row in a table.
Candidate key	A minimal (irreducible) superkey. A superkey that does not contain a subset of attributes that is itself a superkey.
Primary key	A candidate key selected to uniquely identify all other attribute values in any given row. Cannot contain null entries.
Secondary key	An attribute (or combination of attributes) used strictly for data retrieval purposes.
Foreign key	An attribute (or combination of attributes) in one table whose values must either match the primary key in another table or be null.

Tables and Their Characteristics

- Logical view of relational database based on relation
  Relation thought of as a table

- Table: two-dimensional structure composed of rows and columns ,Persistent representation of logical relation

- Contains group of related entities = an entity set

TABLE 3.1 Characteristics of a Relational Table

1	A table is perceived as a two-dimensional structure composed of rows and columns.
2	Each table row (**tuple**) represents a single entity occurrence within the entity set.
3	Each table column represents an attribute, and each column has a distinct name.
4	Each row/column intersection represents a single data value.
5	All values in a column must conform to the same data format.
6	Each column has a specific range of values known as the **attribute domain**.
7	The order of the rows and columns is immaterial to the DBMS.
8	Each table must have an attribute or a combination of attributes that uniquely identifies each row.

Types of Relationship

Relationships within the Relational Database

- 1:M relationship
- 1:1 relationship

M:M relationship

The 1:M Relationship

- Relational database norm
- Found in any database environment

FIGURE 3.18 The 1:M relationship between PAINTER and PAINTING

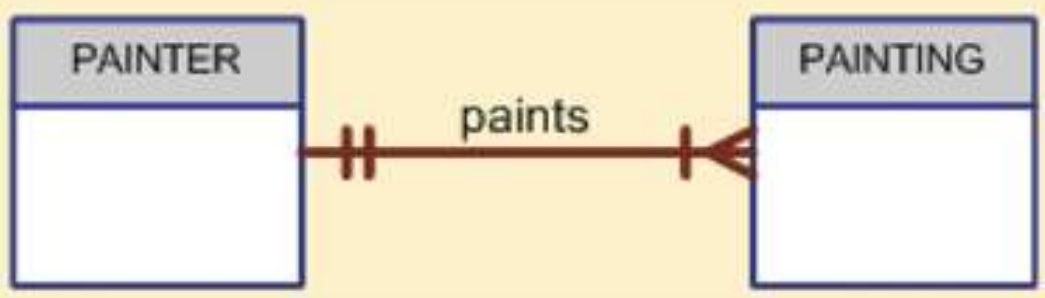


FIGURE 3.19 The implemented 1:M relationship between PAINTER and PAINTING

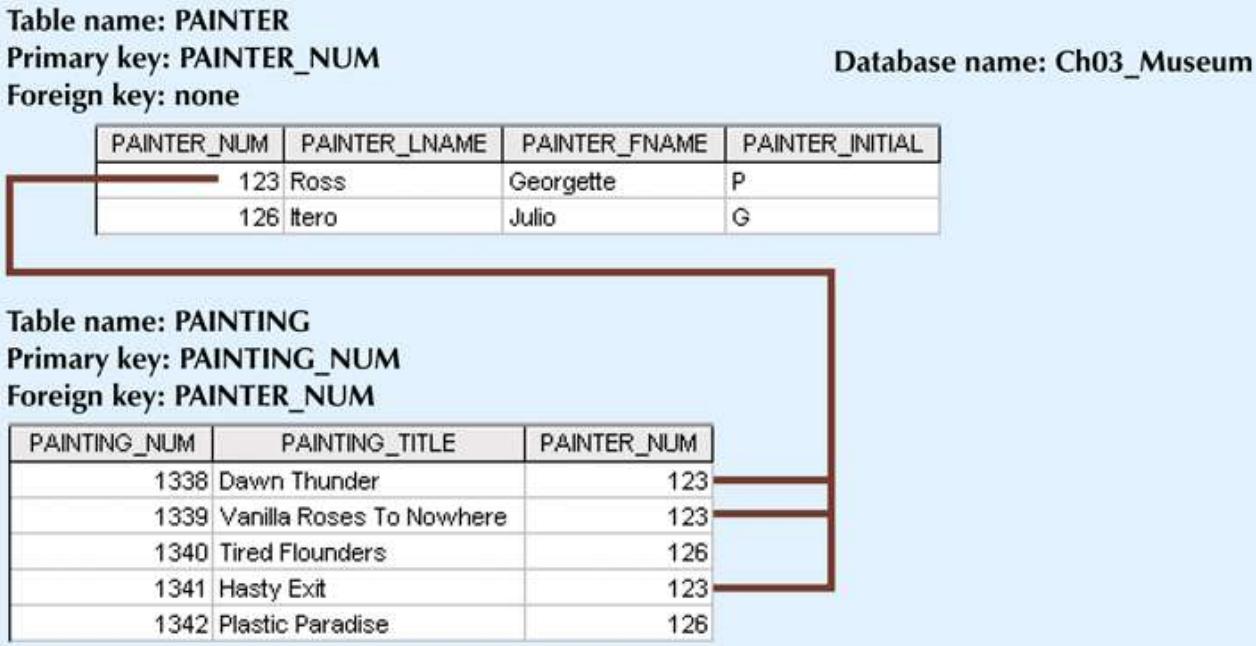


Table name: PAINTER
Primary key: PAINTER_NUM
Foreign key: none

Database name: Ch03_Museum

PAINTER_NUM	PAINTER_LNAME	PAINTER_FNAME	PAINTER_INITIAL
123	Ross	Georgette	P
126	Itero	Julio	G

Table name: PAINTING
Primary key: PAINTING_NUM
Foreign key: PAINTER_NUM

PAINTING_NUM	PAINTING_TITLE	PAINTER_NUM
1338	Dawn Thunder	123
1339	Vanilla Roses To Nowhere	123
1340	Tired Flounders	126
1341	Hasty Exit	123
1342	Plastic Paradise	126

Eg:

- Each painting is painted by one and only one painter, but each painter could have painted many paintings.

- A mother can have **many** children and each child has only **one** mother.

- A car and its parts. Each part belongs to one car and one car has multiple parts.

- A movie theater and screens. One theatre usually has multiple screens and each screen belongs to one theatre.

- Houses in a street. One street had multiple houses and a house belongs to one street.

The 1:1 Relationship

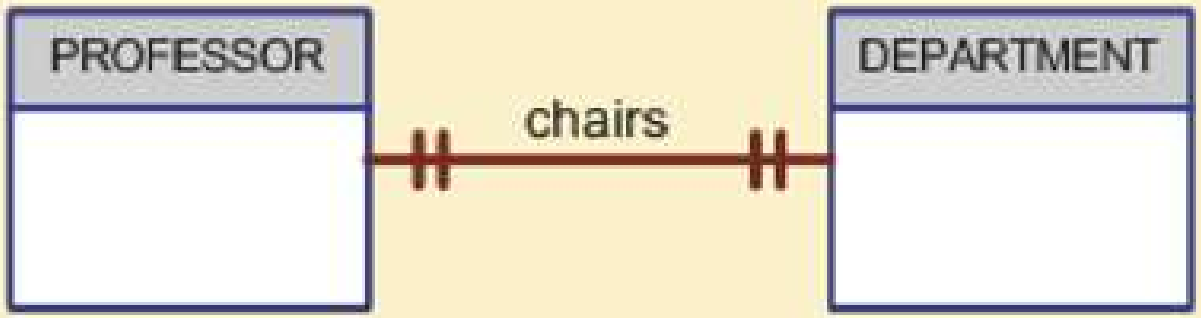
- One entity related to only one other entity, and vice versa

Eg:

- The entities PROFESSOR and DEPARTMENT thus exhibit a 1:1 relationship.
- One person has one passport.
- A car model is made by one company.
- A house building prototype belongs to one company.
- A pair of jeans has one brand name.
- One employee belongs to one organization.

FIGURE 3.22

The 1:1 relationship between PROFESSOR and DEPARTMENT



The M:M Relationship

- Implemented by breaking it up to produce a set of 1:M relationships
- Avoid problems inherent to M:M relationship by creating a **composite entity**
  - Includes as foreign keys the primary keys of tables to be linked

FIGURE 3.24 The ERM's M:N relationship between STUDENT and CLASS

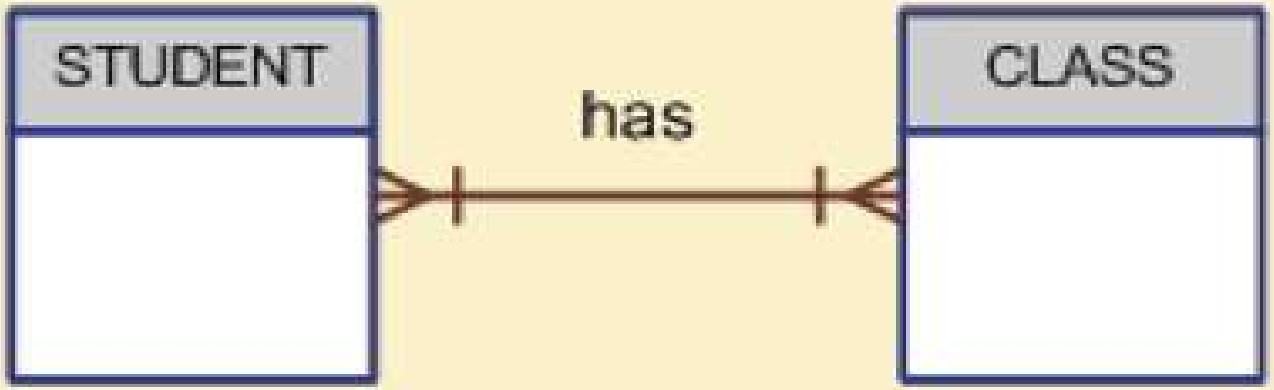


FIGURE 3.26 Converting the M:N relationship into two 1:M relationships

Table name: STUDENT
Primary key: STU_NUM
Foreign key: none

Database name: Ch03_CollegeTry2

STU_NUM	STU_LNAME
321452	Bowser
324257	Smithson

Table name: ENROLL
Primary key: CLASS_CODE + STU_NUM
Foreign key: CLASS_CODE, STU_NUM

CLASS_CODE	STU_NUM	ENROLL_GRADE
10014	321452	C
10014	324257	B
10018	321452	A
10018	324257	B
10021	321452	C
10021	324257	C

Table name: CLASS
Primary key: CLASS_CODE
Foreign key: CRS_CODE

CLASS_CODE	CRS_CODE	CLASS_SECTION	CLASS_TIME	CLASS_ROOM	PROF_NUM
10014	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10018	CIS-220	2	MWF 9:00-9:50 a.m.	KLR211	114
10021	QM-261	1	MWF 8:00-8:50 a.m.	KLR200	114

FIGURE 3.27 Changing the M:N relationship to two 1:M relationships