Intelligent Data Management with SQL Server

Session: 2

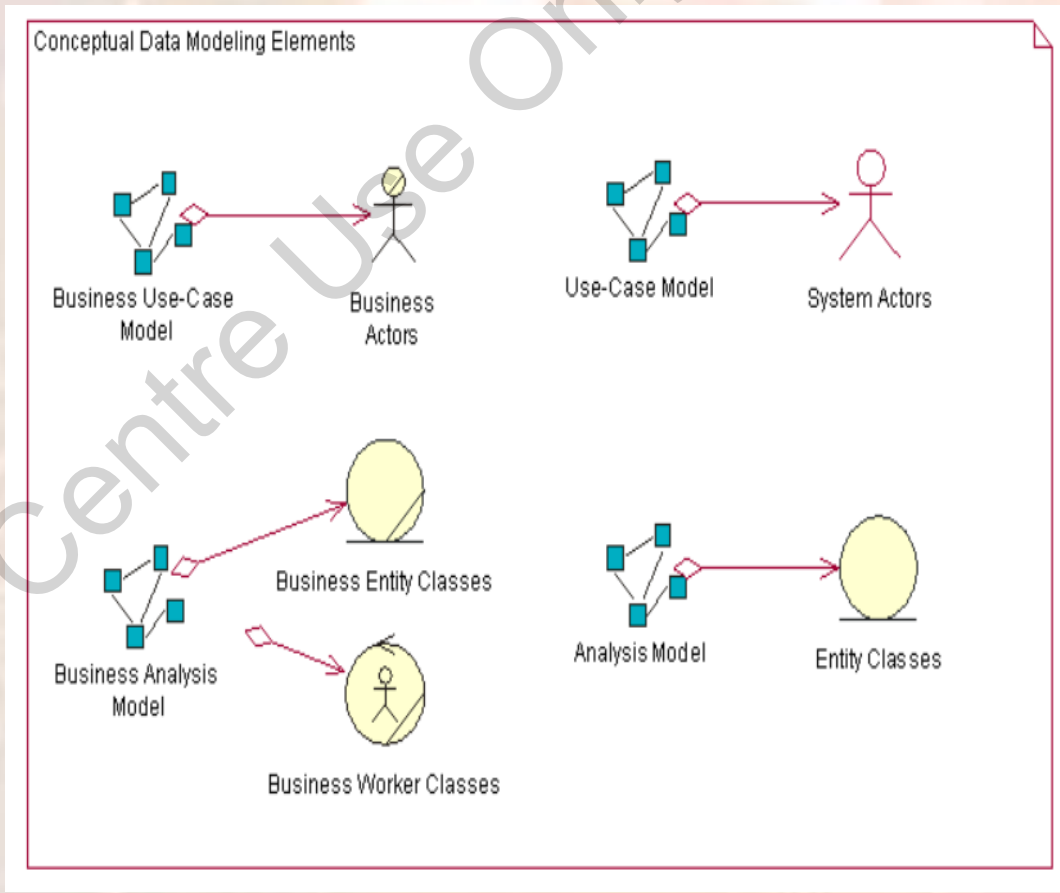
Entity-Relationship (E-R) Model and Normalization

Objectives

- Define and describe data modeling
- Identify and describe the components of the E-R model
- Identify relationships that can be formed between entities
- Explain E-R diagrams and their use
- Describe an E-R diagram, the symbols used for drawing, and show various relationships
- Describe various Normal Forms
- Outline uses of different Relational Operators

Introduction

- A data model is a group of conceptual tools that describes data, its relationships, and semantics.
- Some of the Data model examples are as follows:
  - Entity-Relationship
  - Relational
  - Network
  - Hierarchical models

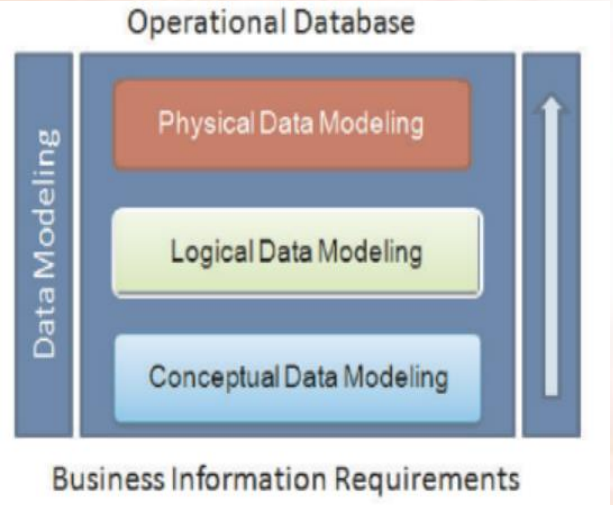


Data Modeling

The process of applying an appropriate data model to the data, in order to organize and structure it.

Its as essential as planning and designing are to any project.

Building a database without a data model is similar to developing a project without its plans and design.



Helps to define:

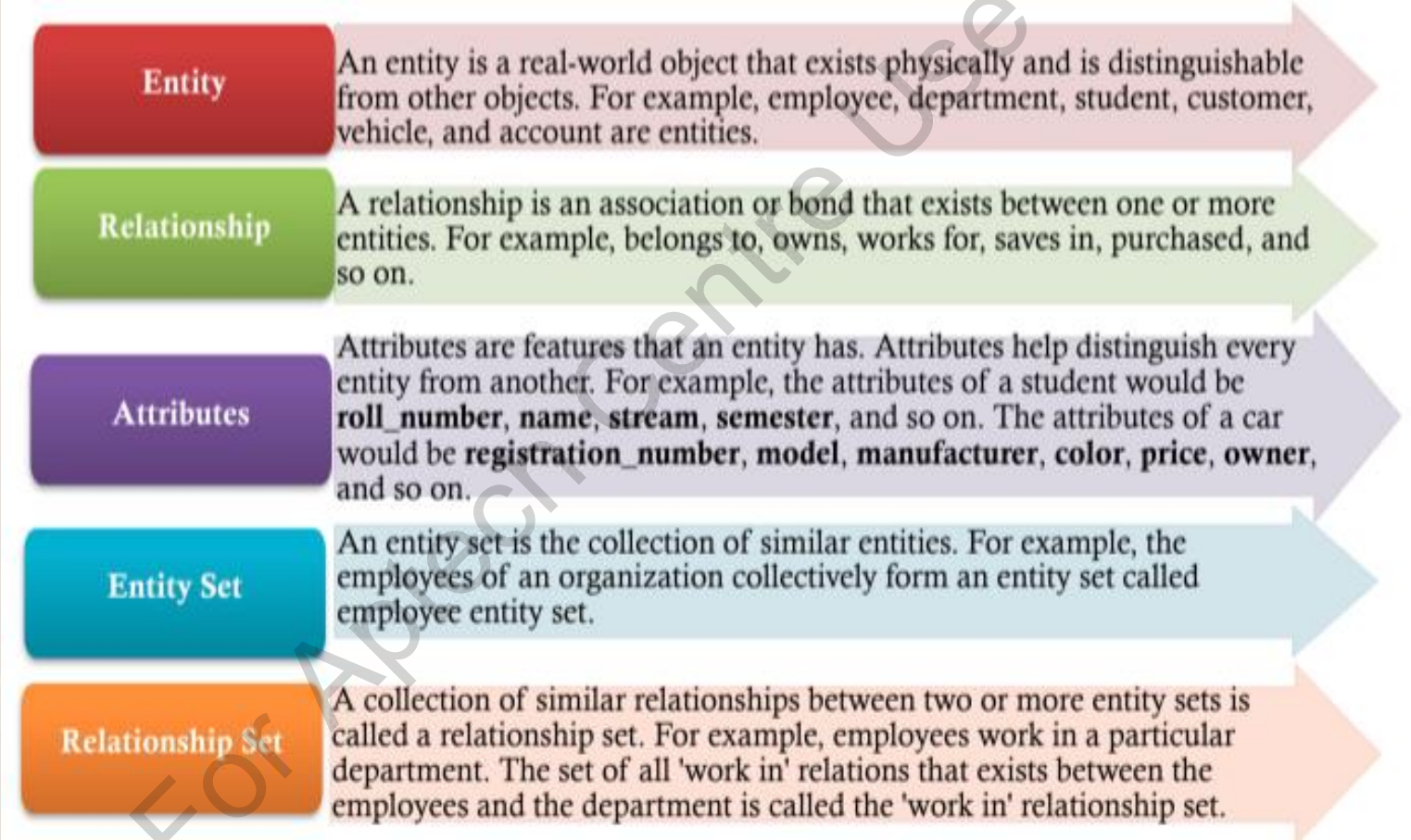
Relational tables, Primary and Foreign keys, Stored procedures and triggers

Entity-Relationship (E-R) Model 1-5

Data models can be classified into three different groups:

Entity-Relationship (E-R) Model 2-5

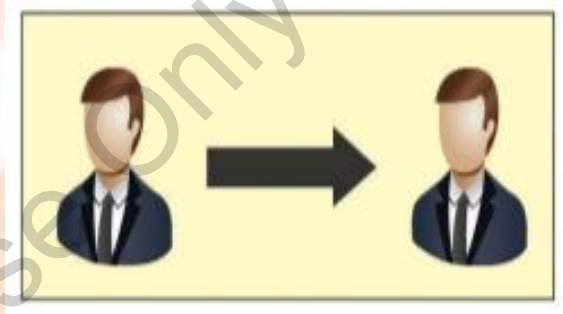
- Components of an E-R model:



Entity-Relationship (E-R) Model 3-5

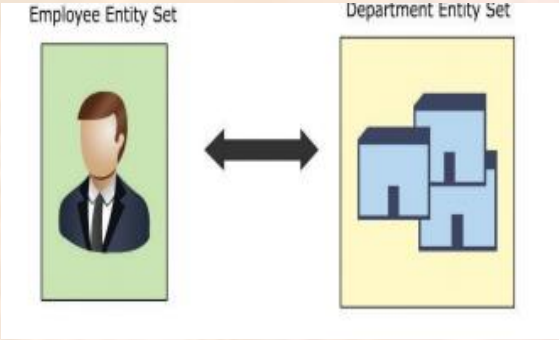
➢ **Self-relationships:**

- Relationships between entities of the same entity set are called self-relationships.
- For example, a manager and his team member, both belong to the employee entity set.



➢ **Binary relationships:**

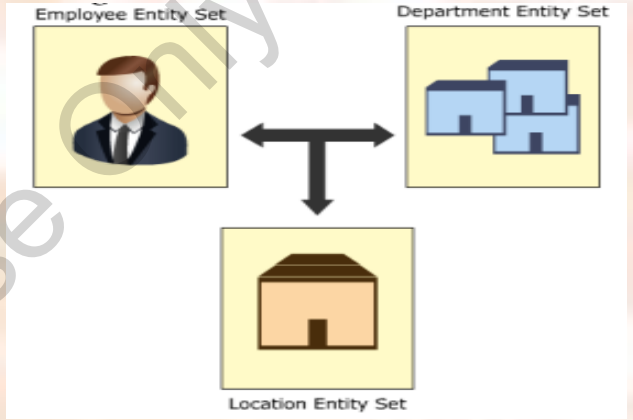
- Relationships that exist between entities of two different entity sets are called binary relationships.
- For example, an employee belongs to a department.



Entity-Relationship (E-R) Model 4-5

➢ **Ternary relationships:**

- Relationships that exist between three entities of different entity sets are called ternary relationships.
- For example, an employee works in the accounts department at the regional branch.



Relationships can also be mapped as:

- ➢ One-to-One
- ➢ One-to-Many
- ➢ Many-to-One
- ➢ Many-to-Many

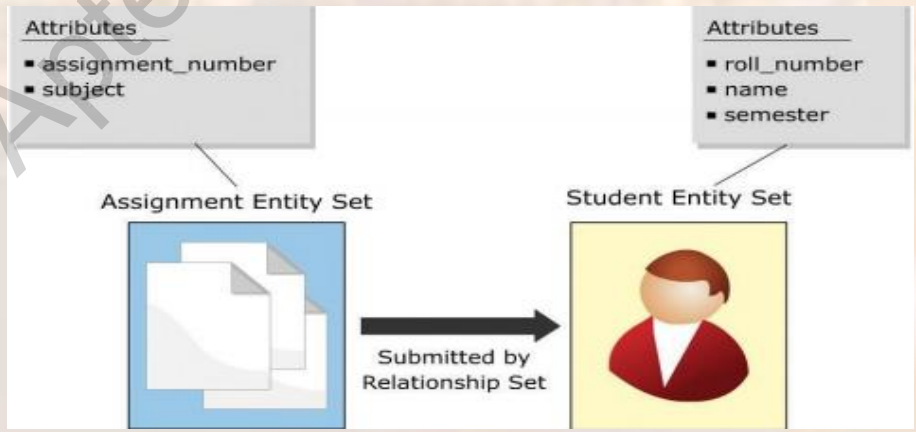
Entity-Relationship (E-R) Model 5-5

Some additional concepts in the E-R model are as follows:

- **Primary Keys:** A primary key is an attribute that can uniquely define an entity in an entity set.

Enrollment_Number	Name	Grade	Division
786	Ashley	Seven	B
957	Joseph	Five	A
1011	Kelly	One	A

- **Weak entity sets:** Entity sets that do not have enough attributes to establish a primary key are called weak entity sets.
- **Strong entity sets:** Entity sets that have enough attributes to establish a primary key are called strong entity sets.



Entity-Relationship Diagrams 1-3

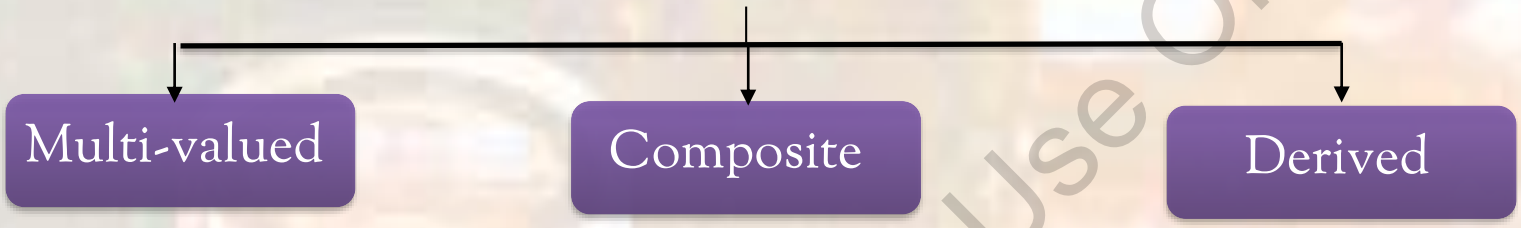
The E-R diagram is a graphical representation of the E-R model. The E-R diagram, with the help of various symbols, effectively represents various components of the E-R model.

Component	Symbol	Example
Entity	Entity	Student
Weak Entity	Weak Entity	Assignments
Attribute	Attribute	Roll_num
Relationship	Relationship	Saves in
Key Attribute	Attribute	Acct_num

E-R Diagram Symbols

Entity-Relationship Diagrams 2-3

Attributes in the E-R model can be further classified as follows:

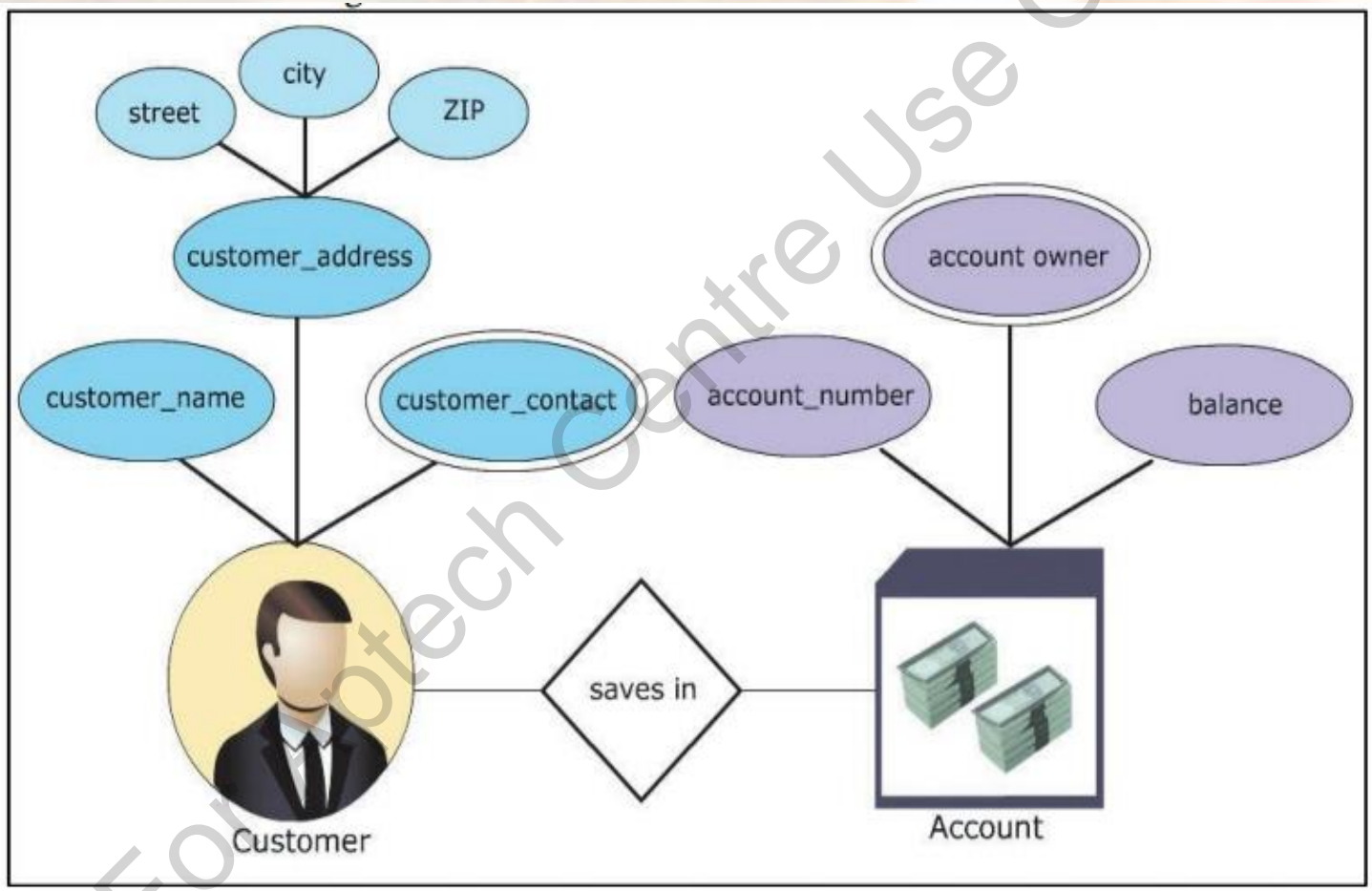


Steps to construct an E-R diagram are as follows:

1. Gather all the data that must be modeled.
2. Identify data that can be modeled as real-world entities.
3. Identify the attributes for each entity.
4. Sort entity sets as weak or strong entity sets.
5. Sort entity attributes as key attributes, multi-valued attributes, composite attributes, derived attributes, and so on.
6. Identify the relations between the different entities.
7. Using different symbols draw the entities, their attributes, and their relationships. Use appropriate symbols while drawing attributes.

Entity-Relationship Diagrams 3-3

An example of the E-R Diagram



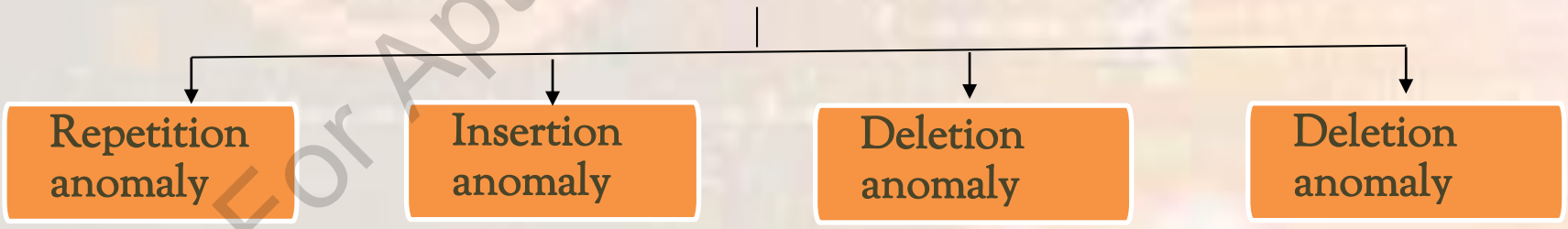
Normalization

Databases are characterized by large number of columns and records

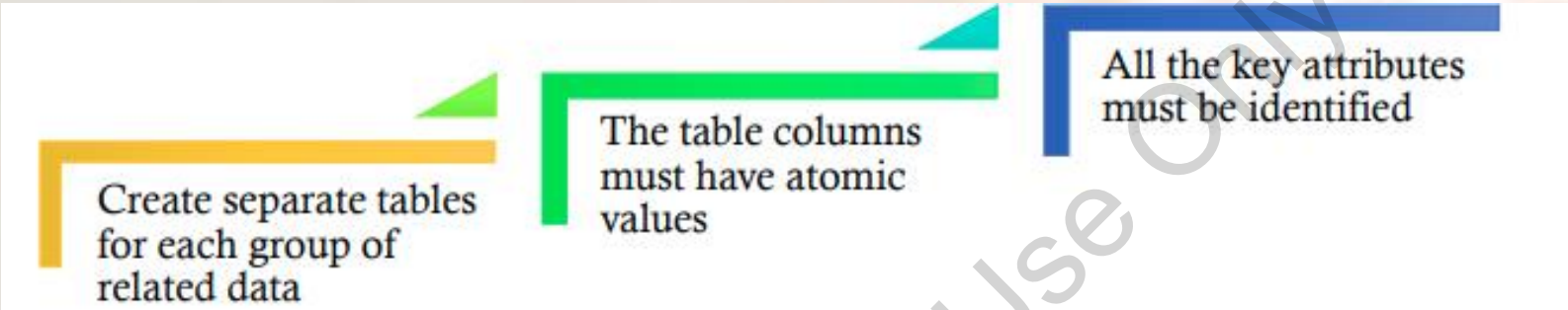
Emp_No	Project_Id	Project_Name	Emp_Name	Grade	Salary
142	113, 124	BLUE STAR, MAGNUM	John	A	20,000
168	113	BLUE STAR	James	B	15,000
263	113	BLUE STAR	Andrew	C	10,000
109	124	MAGNUM	Bob	C	10,000

Department Employee Details

It consists of the following anomalies:



First Normal Form



The table has data related to projects and employees. The table must be split into two tables, that is, a Project Details table and an Employee Details table.

Project_Id	Project_Name
113	BLUE STAR
124	MAGNUM

Table 2.8: Project Details

Emp_No	Emp_Name	Grade	Salary
142	John	A	20,000
168	James	B	15,000
263	Andrew	C	10,000
109	Bob	C	10,000

Second Normal Form

Tables are said to be in second normal form if:

- They meet the requirements of the first normal form
- There are no partial dependencies in the tables
- The tables are related through foreign keys

Project_Id	Project_Name
113	BLUE STAR
124	MAGNUM

Project Details After Conversion to Second Normal Form

Emp_No	Project_Id
142	113
142	124
168	113
263	113

Employee Project Details After Conversion to Second Normal Form

Emp_No	Emp_Name	Grade	Salary
142	John	A	20,000
168	James	B	15,000
263	Andrew	C	10,000

Employee Details After Conversion to Second Normal Form

Third Normal Form

To achieve the third normal form:

- ➢ The tables should meet the requirements of the second normal form
- ➢ The tables should not have transitive dependencies in them

Project_Id	Project_Name
113	BLUE STAR
124	MAGNUM

Project Details After Conversion to Third Normal Form

Emp_No	Emp_Name	Grade
142	John	A
168	James	B
263	Andrew	C
109	Bob	C

Employee Details After Conversion to Third Normal Form

Emp_No	Project_Id
142	113
142	124
168	113
263	113
109	124

Employee Project Details After Conversion to Third Normal Form

Grade	Salary
A	20,000
B	15,000
C	10,000

Grade Salary Details After Conversion to Third Normal Form

Denormalization

By normalizing a database, redundancy is reduced

This, in turn, reduces the storage requirements for the database and ensures data integrity.

Complex join queries may have to be written often to combine the data in multiple tables.

Joins may practically involve more than three tables depending on the need for information.

Relational Operators 1-6

- The relational model is based on the solid foundation of Relational Algebra.
- Relational Algebra consists of a collection of operators that operate on relations.
- Each operator takes one or two relations as its input and produces a new relation as its output.

Branch	Branch_Id	Reserve (Billion €)
London	BS-01	9.2
London	BS-02	10
Paris	BS-03	15
Los Angeles	BS-04	50
Washington	BS-05	30

Relational Operators 2-6

It consists of a set of operators

SELECT Operator: The SELECT operator is used to extract data that satisfies a given condition. The lowercase Greek letter sigma, 'σ', is used to denote selection

Branch	Branch_Id	Reserve (Billion €)
London	BS-01	9.2
London	BS-02	10

PROJECT Operator: The PROJECT operator is used to project certain details of a relational table. The PROJECT operator only displays the required details leaving out certain columns. The PROJECT operator is denoted by the Greek letter pi.

Branch_Id	Reserve (Billion €)
BS-01	9.2
BS-02	10
BS-03	15
BS-04	50
BS-05	30

Relational Operators 3-6

PRODUCT Operator: The PRODUCT operator, denoted by 'x' helps combine information from two relational tables.

Branch_Id	Loan Amount (Billion €)
BS-01	0.56
BS-02	0.84

The product operation on the Branch Reserve Details and Branch Loan Details tables would result in table:

Branch	Branch_Id	Reserve (Billion €)	Loan Amount (Billion €)
London	BS-01	9.2	0.56
London	BS-01	9.2	0.84
London	BS-02	10	0.56
London	BS-02	10	0.84
Paris	BS-03	15	0.56
Paris	BS-03	15	0.84
Los Angeles	BS-04	50	0.56
Los Angeles	BS-04	50	0.84

The product operation combines each record from the first table with all the records in the second table, somewhat generating all possible combinations between the table records.

Relational Operators 4-6

UNION Operator: It collects data from different tables and presents a unified version of the complete data. The union operator is represented by the symbol, 'U'.

Branch	Branch_Id
London	BS-01
London	BS-02
Paris	BS-03

INTERSECT Operator: The INTERSECT operator generates data that holds true in all the tables it is applied on. It is based on the intersection set theory and is represented by the '∩' symbol

Branch	Branch_Id
London	BS-01
London	BS-02

Relational Operators 5-6

DIFFERENCE Operator: The DIFFERENCE operator, symbolized as '-', generates data from different tables too, but it generates data that holds true in one table and not the other.

Branch	Branch_Id
Paris	BS-03

JOIN Operator: The JOIN operation is an enhancement to the product operation. It allows a selection to be performed on the product of tables.

Branch	Branch_Id	Reserve (Billion €)	Loan Amount (Billion €)
London	BS-01	9.2	0.56
London	BS-02	10	0.84

Relational Operators 6-6

DIVIDE Operator: Suppose an official wanted to see the branch names and reserves of all the branches that had loans. This process can be made very easy by using the DIVIDE operator.

Branch	Reserve (Billion €)
London	9.2
London	10

Summary

- Data modeling is the process of applying an appropriate data model to the data at hand.
- E-R model views the real-world as a set of basic objects and relationships among them.
- Entity, attributes, entity set, relationships, and relationship sets form the five basic components of E-R model.
- Mapping cardinalities express the number of entities that an entity is associated with.
- The process of removing redundant data from the tables of a relational database is called normalization.
- Relational Algebra consists of a collection of operators that help retrieve data from the relational databases.
- SELECT, PRODUCT, UNION, and DIVIDE are some of the relational algebra operators.