The Relational Model

A relational table stores a collection of related entities. In this respect, the relational database table resembles a file, but there is a critical difference between a table and a file. A table yields complete data and structural independence because it is purely a logical structure. How the data are physically stored in the database is of no concern to the user or the designer; the perception is what counts.

- It was this factor that led to the relational data model revolutionizing the database world.
- Another reason for the relational data model's rise to dominance is its powerful and flexible query language.
- Most relational database software uses SQL, which allows the user to specify what must be done without having to specify how it is done.

The Relational Model

From an end-user perspective, and SQL-based relational database application involves three parts: a user interface, a set of tables stored in the database, and the SQL "engine."

✓ The end-user interfaces. Basically, the interface allows the end user to interact with the data (by automatically generating SQL code). Each interface is a product of the software vendor's ides of meaningful interaction with the data.

✓ A collection of tables stored in the database. In a relational database, all data are perceived to be stored in tables. The tables simply present the data to the end user in a way that is easy to understand. Each table is independent. Rows in different tables are related by common values in common attributes.

✓ SQL engine. Largely hidden from the end user, the SQL engine executes all queries or data requests. The SQL engine is part of the DBMS software.

The Entity-Relationship Model

Because it is easier to examine structures graphically than to describe them in text, database designers prefer to use a graphical tool in which entities and their relationships are pictured.

• As a result, the entity-relationship (ER) model, or ERM, has become the most widely accepted standard for database modeling.

• Peter Chen first introduced the ER data model in 1976 and it quickly became popular because it complemented the relational data model concepts.

• The relational data model (a logical model) and the ER model (a conceptual model) are combined to provide the foundation for tightly structured database design.

The Entity-Relationship Model

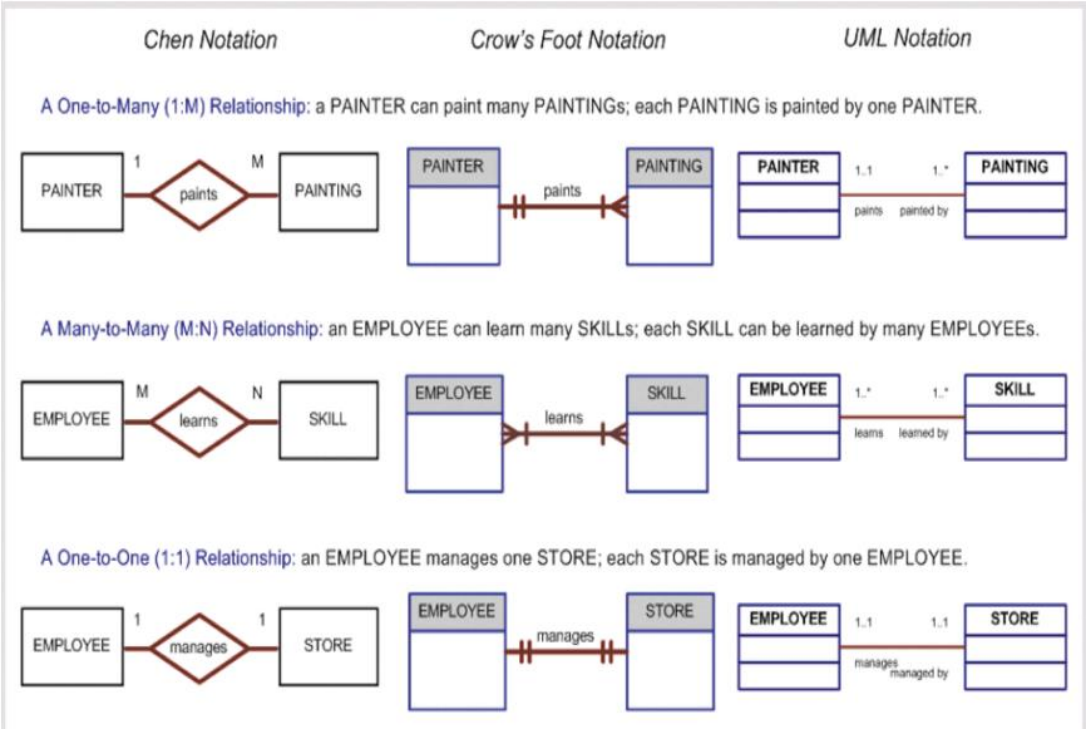
The ER model is based on the following components:

Entity. An entity is represented in the ERD by a rectangle, also known as an entity box. The name of the entity, a noun, is written in the center of the rectangle.

- The entity name is generally written in all capital letters and singular in form: PAINTER rather than PAINTERS and EMPLOYEE rather than EMPLOYEES. When applying the ERD to the relational model, an entity is mapped to a relational table.
- Each row in the relational table is known as an entity instance or entity occurrence in the ER model. Each entity consists of a set of attributes that describes particular characteristics of the entity. We'll see how to include these shortly.

Relationships. Relationships describe associations among data. Most relationships describe associations between two entities. The ER model uses the term connectivity or cardinality to label the relationship types.

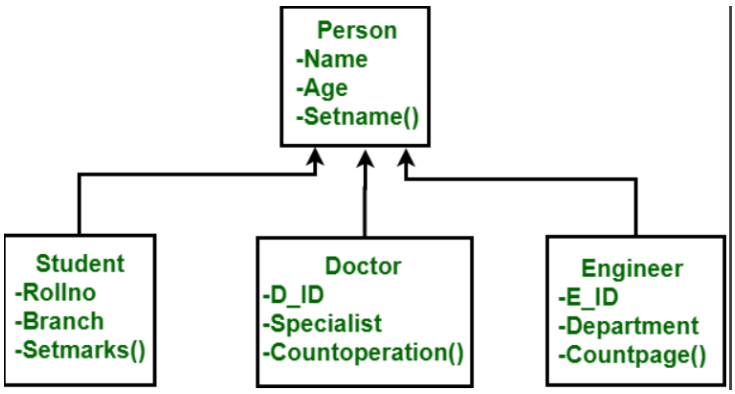
• The next slide illustrates the different types of relationships using three different ER notations.



The Object-Oriented (OO) Model

In Object Oriented Data Model, data and their relationships are contained in a single structure which is referred as object in this data model. In this, real world problems are represented as objects with different attributes. All objects have multiple relationships between them. Basically, it is combination of Object Oriented programming and Relational Database Model as it is clear from the following figure :

Components of Object Oriented Data Model :



Objects

An object is an abstraction of a real-world entity or we can say it is an instance of class. Objects encapsulates data and code into a single unit which provide data abstraction by hiding the implementation details from the user. For example: Instances of student, doctor, engineer in above figure.

Attribute

An attribute describes the properties of object. For example: Object is STUDENT and its attribute are Roll no, Branch, Setmarks() in the Student class.

Methods

Method represents the behaviour of an object. Basically, it represents the real-world action. For example: Finding a STUDENT marks in above figure as Setmarks().

Class

A class is a collection of similar objects with shared structure i.e. attributes and behavior i.e. methods. An object is an instance of class. For example: Person, Student, Doctor, Engineer in above figure.

Object/Relational and XML Models

- Facing the demand to support more complex data representations, the relational model's main vendors evolved the model further and created the extended relational data model (ERDM).
- The ERDM adds many of the OO model's features within the inherently simpler relational database structure. The ERDM gave rise to a new generation of relational databases that support OO features such as objects, extensible data types based on classes and inheritance.
- DBMS based on the ERDM are often described as an object/relational database management system (O/R DBMS).
- Today, most relational database products can be classified as object/relational, and they represent the dominate market share of OLTP(online transaction processing) and OLAP(online analytical processing) database applications.
- The success of the O/R DBMS can be attributed to the model's conceptual simplicity, data integrity, easy-to-use query language, high transaction

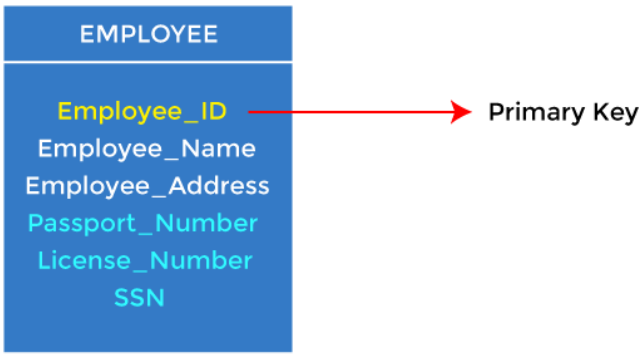
performance, high availability, security, scalability, and expandability.

- In contrast, the OO DBMS is popular in niche markets such as CAD/CAM systems, geographic information systems (GIS), telecommunications, and multimedia, which require support for more complex objects.
- From the start, the OO and relational models were developed in response to different problems. The OO model was created to address very specific engineering needs, not the wide-ranging needs of general data management tasks
- Given its focus on a smaller set of problem areas, it is not surprising that the OO market has not grown as rapidly as the relational data model market.
- Although relational and object/relational databases address most current data processing needs, a new generation of databases has emerged to address some very specific challenges found in some Internet-era organizations.
- XML becomes the standard for the representation of structured and semi-structured data on the Web. Relational and object-relational database systems are a well understood technique for managing and querying such large sets of structured data.
- These structures would sometimes result in large database schemas and sparsely populated databases. As a consequence, such XML document fragments should be mapped onto database attributes of type XML and kept as is.
- The XML datatype implementation should support evaluating path expressions and fulltext operations. We present an algorithm that finds a type of optimal mapping based on the XML Document Type Definition (DTD) and statistics. The statistics are derived from sample XML document sets and some knowledge about queries on XML document collections.

Keys in DBMS

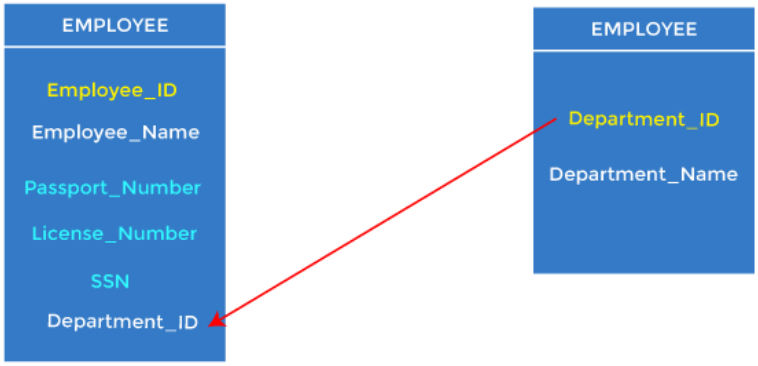
1. Primary key

- It is the first key used to identify one and only one instance of an entity uniquely. An entity can contain multiple keys, as we saw in the PERSON table. The key which is most suitable from those lists becomes a primary key.
- For each entity, the primary key selection is based on requirements and developers.



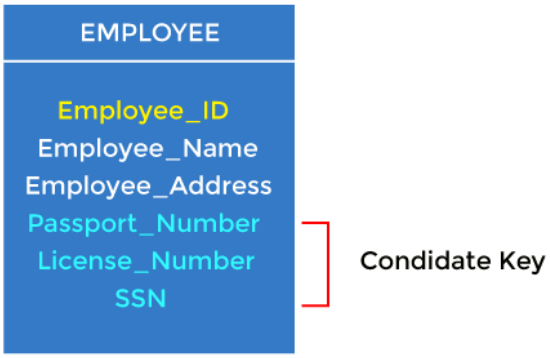
2. Foreign key

- Foreign keys are the column of the table used to point to the primary key of another table.
- Every employee works in a specific department in a company, and employee and department are two different entities. So we can't store the department's information in the employee table. That's why we link these two tables through the primary key of one table.
- We add the primary key of the DEPARTMENT table, Department_Id, as a new attribute in the EMPLOYEE table.
- In the EMPLOYEE table, Department_Id is the foreign key, and both the tables are related.



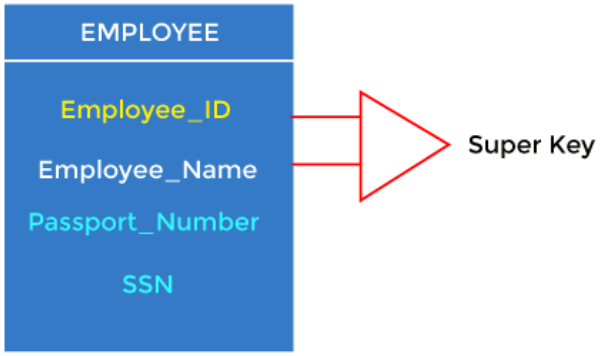
3. Candidate key

- A candidate key is an attribute or set of attributes that can uniquely identify a tuple.
- Except for the primary key, the remaining attributes are considered a candidate key. The candidate keys are as strong as the primary key.
- **For example:** In the EMPLOYEE table, id is best suited for the primary key. The rest of the attributes, like SSN, Passport_Number, License_Number, etc., are considered a candidate key.



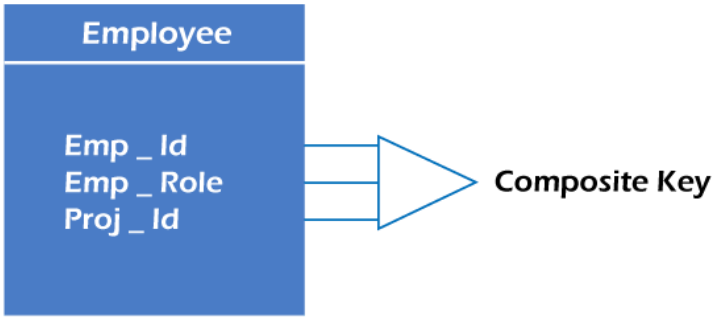
4. Super Key

- Super key is an attribute set that can uniquely identify a tuple. A super key is a superset of a candidate key.



5. Composite key

- Whenever a primary key consists of more than one attribute, it is known as a composite key. This key is also known as Concatenated Key.



- **For example,** in employee relations, we assume that an employee may be assigned multiple roles, and an employee may work on multiple projects simultaneously. So the primary key will be composed of all three attributes, namely Emp_ID, Emp_role, and Proj_ID in combination. So these attributes act as a composite key since the primary key comprises more than one attribute.

6. Secondary key

- In other words, **Secondary key** in DBMS is a column or a set of columns in a table that uniquely identify each row in a table that is not the primary

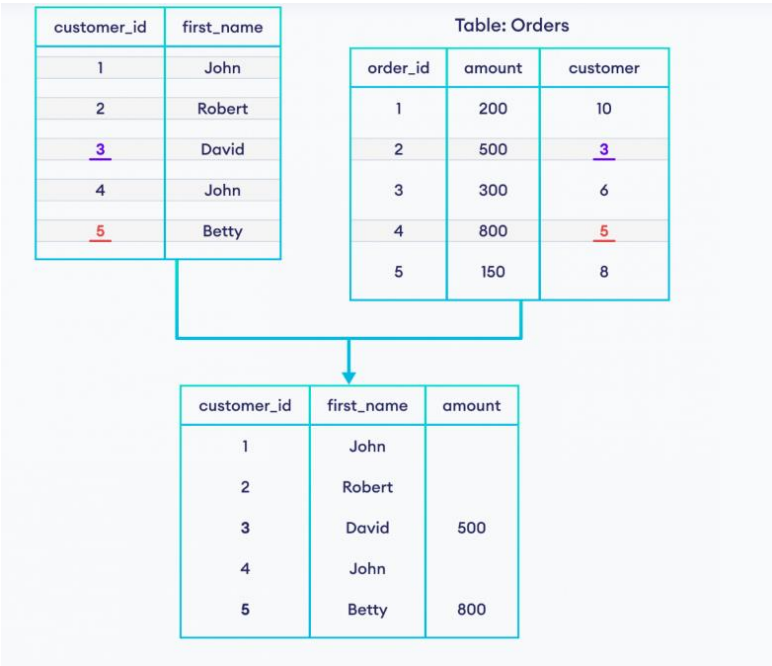
key. Secondary keys are also called **alternate keys** as they can also be used alternatively in place of the primary key.

Basis	Primary Key	Secondary Key
Definition	A key that is unique, not null, and is selected by the database administrator to uniquely identify tuples is called the primary key.	A key that uniquely identifies rows but is not selected as the primary key is known as a secondary key or alternate key.
NULL values	It cannot be NULL.	It can be NULL.
Number of keys	There must be one and only one primary key.	There can be zero or more secondary keys.

Left outer join

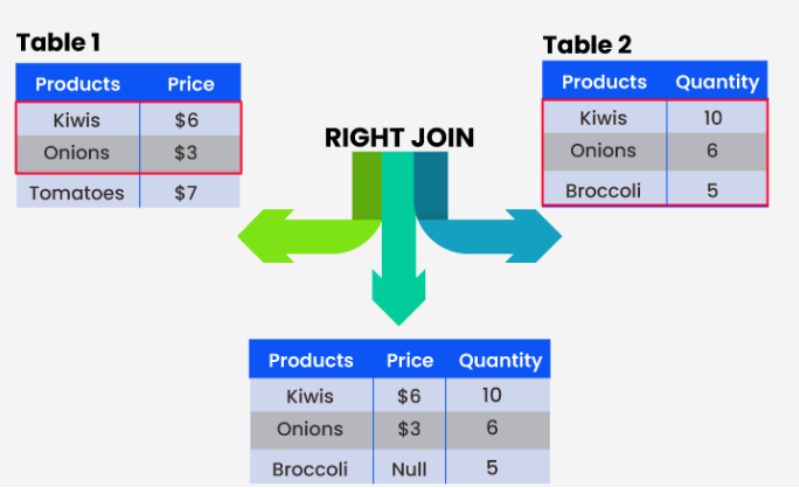
Example 1

Example :2



Right outer join

Example :1



Example 2