# Database Management Systems

## Relational Data Model

Malay Bhattacharyya

Assistant Professor

Machine Intelligence Unit
and
Centre for Artificial Intelligence and Machine Learning
Indian Statistical Institute, Kolkata

February, 2022

# A relational view of data

We can define a *relation* within the data from the mathematical perspective. Given the sets $X_1, X_2, \ldots, X_n$ (not necessarily distinct), $R$ is a relation on these $n$ sets if it is a set of $n$-tuples each of which has its first element from $X_1$, its second element from $X_2$, and so on. We often refer to $X_i$ as the $i$th attribute (taking values from a domain) of $R$.

As defined above, $R$ is said to have degree $n$. Relations of degree 1 are often called unary, degree 2 binary, degree 3 ternary, and degree $n$ $n$-ary.

# A relational view of data

We can define a *relation* within the data from the mathematical perspective. Given the sets $X_1, X_2, \ldots, X_n$ (not necessarily distinct), $R$ is a relation on these $n$ sets if it is a set of $n$-tuples each of which has its first element from $X_1$, its second element from $X_2$, and so on. We often refer to $X_i$ as the $i$th attribute (taking values from a domain) of $R$.

As defined above, $R$ is said to have degree $n$. Relations of degree 1 are often called unary, degree 2 binary, degree 3 ternary, and degree $n$ $n$-ary.

**Note:** There is no practical importance of defining a relation with degree 0.

# Preliminaries

Example of a relation of degree 4:

Table: OSCAR

| Year | Category | Movie | Name |
|---|---|---|---|
| 1982 | Best Costume Design | Gandhi | Bhanu Athaiya |
| 1992 | Lifetime Achievement Award | null | Satyajit Ray |
| 2008 | Best Original Song | Slumdog Millionaire | Gulzar |
| 2008 | Best Original Song | Slumdog Millionaire | A.R. Rahman |
| 2008 | Best Original Score | Slumdog Millionaire | A.R. Rahman |
| 2008 | Best Sound Mixing | Slumdog Millionaire | Resul Pookutty |

# The basic notions

In the Entity-Relationship (E-R) data model, we deal with multiple relations present in the data.

The E-R data model uses three basic notions as listed below:

- Entity sets
- Attributes
- Relationship sets

# The basic notions

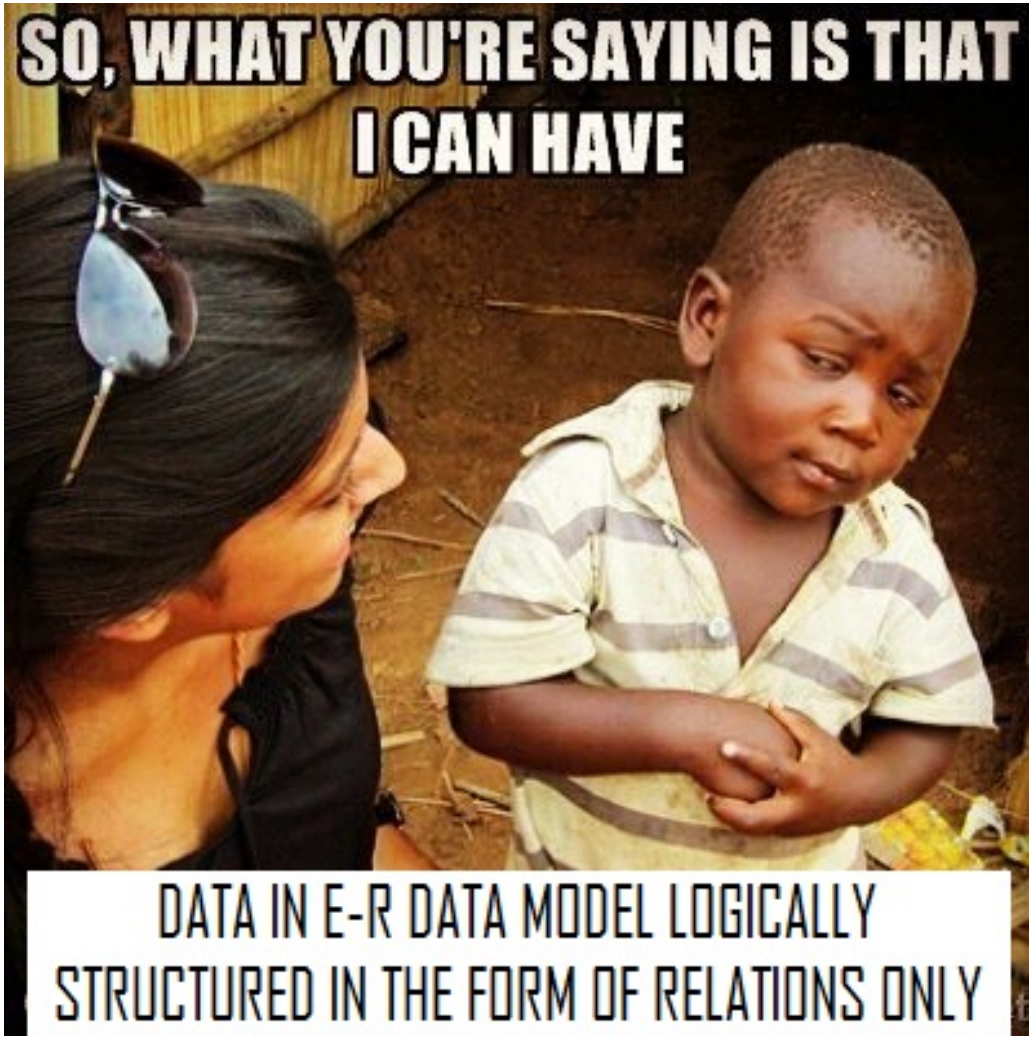
In the Entity-Relationship (E-R) data model, we deal with multiple relations present in the data.

The E-R data model uses three basic notions as listed below:

- Entity sets
- Attributes
- Relationship sets

**Note:** The Entity-Relationship (E-R) data model defines relations between the relations (tables).

# The relations



# Entity sets

An *entity set* is a set of objects (entities) of the same type that share the same attributes. E.g., the set of all musicians who are associated with a particular music company can be defined as the entity set 'Musicians'.

# Entity sets

An *entity set* is a set of objects (entities) of the same type that share the same attributes. E.g., the set of all musicians who are associated with a particular music company can be defined as the entity set 'Musicians'.

**Note:** Entity sets are not necessarily disjoint.

# Entity sets

An entity signifies something that exists in real-world. It is like an object that is distinguishable from others. An entity is either *tangible* (exists physically) or *intangible* (exists only logically and have no physical existence).

Table: Musicians → Entity Type

| **Name** | **Aadhaar** | |
|---|---|---|
| Zakir Hussain | ???????????? | |
| Pt. Shiv Kumar Sharma | XXXXXXXXXXX | → Entity 1 |
| Shivmani | ********** | |
| A. R. Rahman | @@@@@@@@@@@@ | → Entity 2 |

# Entity sets

An entity signifies something that exists in real-world. It is like an object that is distinguishable from others. An entity is either *tangible* (exists physically) or *intangible* (exists only logically and have no physical existence).

Table: Musicians → Entity Type

| **Name** | **Aadhaar** | |
|---|---|---|
| Zakir Hussain | ???????????? | |
| Pt. Shiv Kumar Sharma | XXXXXXXXXXX | → Entity 1 |
| Shivmani | ********** | |
| A. R. Rahman | @@@@@@@@@@@@ | → Entity 2 |

| Entity Set |
|---|
| Entity 1 |
| Entity 2 |

# Attributes

An entity is represented by a set of attributes. An *attribute* of an entity set is a function that maps from the entity set into a domain.

# Attributes

An entity is represented by a set of attributes. An *attribute* of an entity set is a function that maps from the entity set into a domain.

An attribute can be of different types as given below.

- Simple attribute – indivisible (e.g., age)
- Composite attribute – divisible (e.g., STD code and local code in a phone number)
- Single-valued attribute – takes a single value (e.g., gender)
- Multi-valued attribute – takes multiple values (e.g., playing instruments)
- Derived attribute – value can be derived from other attributes (e.g., age can be derived from DOB)
- Descriptive attribute - takes descriptive value (e.g., reason of breaking a contract)

# Let us brainstorm!!!

Relational data models are so powerful that it can represent any kind of real-life database scenarios, which bear relations therein. Imagine about a real-world environment where an attribute is considered as composite, multi-valued, and also derived in the same relational data model. Cite such an example.

# Let us brainstorm!!!

Let there be a relation with the attribute `PerformerDOB`, a multi-valued attribute representing the date of births of active performers in a theatre group. It is also a composite attribute as one can derive the simple attributes like date, month and year of births from this. It is derived from `PerformerAge`, another attribute in the same relation having exact ages (in years, months and days) as entries.

# Relationship sets

A *relationship set* denotes a set of associations (relationships) among multiple entities. In a formal sense, it is a mathematical relation on no less that two (possibly nondistinct) entity sets.

# Relationship sets

A *relationship set* denotes a set of associations (relationships) among multiple entities. In a formal sense, it is a mathematical relation on no less that two (possibly nondistinct) entity sets.

Consider the following two entity sets 'Musicians' and 'Instruments'. We can define a relationship set 'Plays' to denote the association between musicians and the instruments they play.

Table: Musicians

| Name | Aadhaar |
|---|---|
| Zakir Hussain | ???????????? |
| Pt. Shiv Kumar Sharma | XXXXXXXXXXX |
| Shivmani | ********** |
| A. R. Rahman | @@@@@@@@@@@ |

Table: Instruments

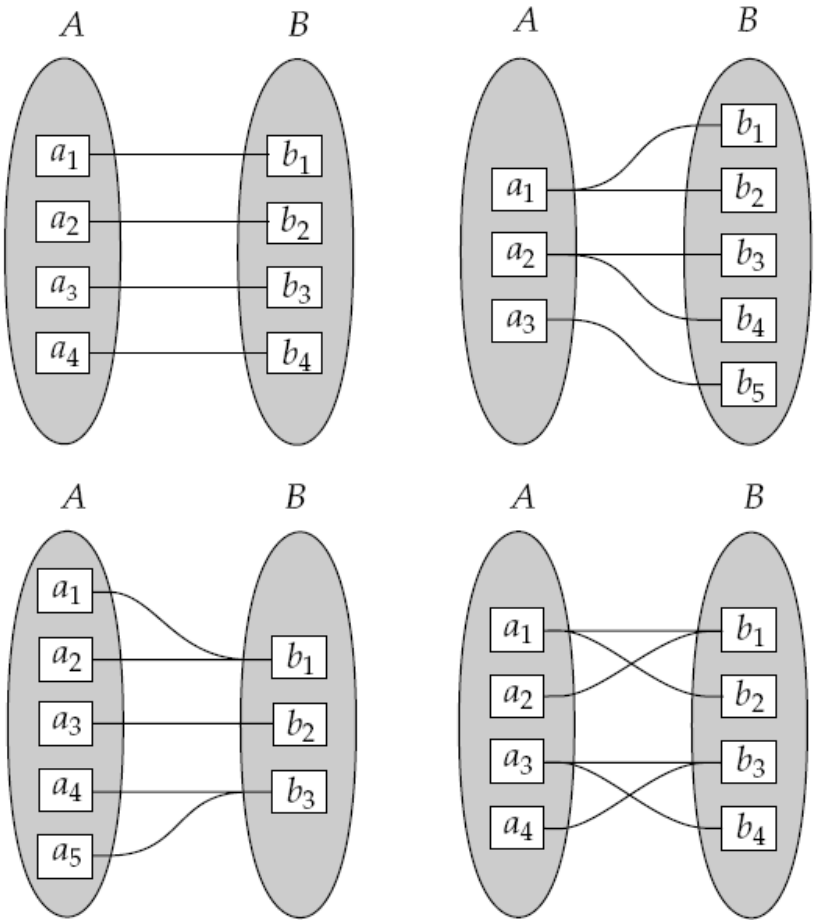
| ID | Name | Key |
|---|---|---|
| 1 | Flute | G-Flat |
| 2 | Flute | B |
| 3 | Flute | B-flat |
| 4 | Guitar | E-flat |

# Constraints in E-R data model

An E-R data model may include the following types of constraints:

- Constraints on mapping cardinalities (also termed as constraints on cardinality ratios) – reflects the number of entities to which another entity can be associated via a relationship set
- Participation constraints – reflects the fraction of entities that can participate in at least one relationship
- Complex constraints – hybrid of other constraints
- Key related constraints

# Constraints on mapping cardinalities



# Constraints on mapping cardinalities

- One-to-one: An entity in $A$ is associated with at most one entity in $B$, and an entity in $B$ is associated with at most one entity in $A$.
- One-to-many: An entity in $A$ is associated with any number (zero or more) of entities in $B$. An entity in $B$, however, can be associated with at most one entity in $A$.
- Many-to-one: An entity in $A$ is associated with at most one entity in $B$. An entity in $B$, however, can be associated with any number (zero or more) of entities in $A$.
- Many-to-many: An entity in $A$ is associated with any number (zero or more) of entities in $B$, and an entity in $B$ is associated with any number (zero or more) of entities in $A$.

# Constraints on mapping cardinalities

- One-to-one: An entity in $A$ is associated with at most one entity in $B$, and an entity in $B$ is associated with at most one entity in $A$.
- One-to-many: An entity in $A$ is associated with any number (zero or more) of entities in $B$. An entity in $B$, however, can be associated with at most one entity in $A$.
- Many-to-one: An entity in $A$ is associated with at most one entity in $B$. An entity in $B$, however, can be associated with any number (zero or more) of entities in $A$.
- Many-to-many: An entity in $A$ is associated with any number (zero or more) of entities in $B$, and an entity in $B$ is associated with any number (zero or more) of entities in $A$.

As for example, the relationship set 'Plays' between the entity sets 'Musicians' and 'Instruments' is many-to-many.

# Participation constraints

- Total participation: The participation of an entity set $E$ in a relationship set $R$ is said to be *total* if every entity in $E$ participates in at least one relationship in $R$.
- Partial participation: The participation of an entity set $E$ in a relationship set $R$ is said to be *partial* if only some entities in $E$ participate in at least one relationship in $R$.

# Participation constraints

- Total participation: The participation of an entity set $E$ in a relationship set $R$ is said to be *total* if every entity in $E$ participates in at least one relationship in $R$.
- Partial participation: The participation of an entity set $E$ in a relationship set $R$ is said to be *partial* if only some entities in $E$ participate in at least one relationship in $R$.

As for example, consider that an entity set 'Album' is associated with another entity set 'Musicians' via a relationship set 'Directs'. Then, the participation of 'Album' in the relationship set 'Directs' is total but the participation of 'Musicians' is partial.

**Note:** The constraints on mapping cardinalities are between the entity pairs but the participation constraints are between an entity and a relationship set.

# Complex constraints

E-R diagrams can also reflect more complex constraints on the number of times each entity participates in relationships in a relationship set.

An edge between an entity set and a binary relationship set can be labelled with minimum and maximum cardinalities, shown in the form $l..h$. Here. $l \in \{0, 1, \ldots, *\}$ and $h \in \{1, 2, \ldots, *\}$ $(l \leq h)$ denote the minimum and maximum cardinality, respectively.

# Complex constraints

E-R diagrams can also reflect more complex constraints on the number of times each entity participates in relationships in a relationship set.

An edge between an entity set and a binary relationship set can be labelled with minimum and maximum cardinalities, shown in the form $l..h$. Here. $l \in \{0, 1, \ldots, *\}$ and $h \in \{1, 2, \ldots, *\}$ $(l \leq h)$ denote the minimum and maximum cardinality, respectively.

- A minimum value of 1 (i.e., $l = 1$) indicates total participation of the entity set in the relationship set.
- A maximum value of 1 (i.e., $h = 1$) indicates that the entity participates in at most one relationship, while a maximum value * indicates no limit.

**Note:** A label 1..* on an edge is equivalent to a double line.

# Concept of keys

Given a relation $R$, a set of attributes $K \subseteq \mathcal{A}(R)$ is a *superkey* if for all distinct pairs of tuples $t1$ and $t2$ ($t1 \neq t2$) in $R$, $t1[K] \neq t2[K]$ hold, i.e. no two tuples in any legal relation $r(R)$ may have the same value on attribute set $K$.

Basically, a *superkey* is a set of one or more attributes that can uniquely identify an entity in the entity set.

# Concept of keys

Given a relation $R$, a set of attributes $K \subseteq \mathcal{A}(R)$ is a *superkey* if for all distinct pairs of tuples $t1$ and $t2$ ($t1 \neq t2$) in $R$, $t1[K] \neq t2[K]$ hold, i.e. no two tuples in any legal relation $r(R)$ may have the same value on attribute set $K$.

Basically, a *superkey* is a set of one or more attributes that can uniquely identify an entity in the entity set.

A *candidate key* is a *superkey* for which no proper subset is a superkey, i.e. a minimal *superkey*.

# Concept of keys

Given a relation $R$, a set of attributes $K \subseteq \mathcal{A}(R)$ is a *superkey* if for all distinct pairs of tuples $t1$ and $t2$ ($t1 \neq t2$) in $R$, $t1[K] \neq t2[K]$ hold, i.e. no two tuples in any legal relation $r(R)$ may have the same value on attribute set $K$.

Basically, a *superkey* is a set of one or more attributes that can uniquely identify an entity in the entity set.

A *candidate key* is a *superkey* for which no proper subset is a superkey, i.e. a minimal *superkey*.

A *primary key* is a *candidate key* that is finally used by the database designer.

# Concept of keys

Given a relation $R$, a set of attributes $K \subseteq \mathcal{A}(R)$ is a *superkey* if for all distinct pairs of tuples $t1$ and $t2$ ($t1 \neq t2$) in $R$, $t1[K] \neq t2[K]$ hold, i.e. no two tuples in any legal relation $r(R)$ may have the same value on attribute set $K$.

Basically, a *superkey* is a set of one or more attributes that can uniquely identify an entity in the entity set.

A *candidate key* is a *superkey* for which no proper subset is a superkey, i.e. a minimal *superkey*.

A *primary key* is a *candidate key* that is finally used by the database designer.

An attribute of a relation $R_1$ is its *foreign key*, referencing another relation $R_2$, if it is a *primary key* for $R_2$.

# Primary keys – A caution

# Concept of keys

Table: OSCAR_DIRECTOR

| Best Director | Awards | Nominations |
|---|---|---|
| John Ford | 4 | 5 |
| William Wyler | 3 | 12 |
| Frank Capra | 3 | 6 |
| Billy Wilder | 2 | 8 |
| David Lean | 2 | 7 |
| Fred Zinnemann | 2 | 7 |
| Steven Spielberg | 2 | 7 |

In the above relational schema, {Best Director}, {Best Director, Awards}, {Best Director, Nominations} and {Best Director, Awards, Nominations} are all *superkeys* and {Best Director} is the only *candidate key*.

# How to choose the primary key?

Although there is no Golden rule, many suggestions on choosing the primary key (from candidate keys) are available in practice.

1. The candidate key having attributes that allow null values is not suitable.
2. The candidate key having attributes with no embedded spaces, special characters, or differential capitalization is better.
3. The candidate key having the lest cardinality is better.
4. The candidate key having numeric (preferably integer) attributes is better.
5. The candidate key which is minimally updated is better.

# How to choose the primary key?

Although there is no Golden rule, many suggestions on choosing the primary key (from candidate keys) are available in practice.

1. The candidate key having attributes that allow null values is not suitable.
2. The candidate key having attributes with no embedded spaces, special characters, or differential capitalization is better.
3. The candidate key having the lest cardinality is better.
4. The candidate key having numeric (preferably integer) attributes is better.
5. The candidate key which is minimally updated is better.

**Note:** Candidate keys that are not chosen as the primary key are often termed as `alternate keys`. We often use an additional `surrogate key` as the primary key.

# Concept of keys

Table: SCORE

| A | B | C | D | E | F |
|---|---|---|---|---|---|
| 4 | 5 | 2 | 3 | 2 | 5 |
| 5 | 4 | 2 | 4 | 2 | 5 |
| 4 | 7 | 3 | 5 | 2 | 5 |
| 6 | 7 | 2 | 3 | 4 | 5 |

In the above relational schema, {A, B} and {C, D, E} are both *candidate keys*.

# Concept of keys

Table: SCORE

| A | B | C | D | E | F |
|---|---|---|---|---|---|
| 4 | 5 | 2 | 3 | 2 | 5 |
| 5 | 4 | 2 | 4 | 2 | 5 |
| 4 | 7 | 3 | 5 | 2 | 5 |
| 6 | 7 | 2 | 3 | 4 | 5 |

In the above relational schema, {A, B} and {C, D, E} are both *candidate keys*. Let us assume {A, B} is chosen as the primary key in SCORE. Then {A, B} is a foreign key in relation RANK.

Table: RANK

| R | A | B |
|---|---|---|
| 1 | 4 | 5 |
| 2 | 5 | 4 |
| 3 | 4 | 7 |
| 4 | 4 | 5 |

# Weak and strong entity sets

If an entity set does not have sufficient attributes to form a primary key then it is termed as a *weak entity set*, otherwise it is termed as a *strong entity set*.

# Special features – Specialization

An entity set may include subgroupings of entities that are distinct in some way from other entities in the set.

For instance, a subset of entities within an entity set may have attributes that are not shared by all the entities in the entity set. The E-R model provides a means for representing these distinctive entity groupings.

# Special features – Generalization

There might exist similarities between two entities in the sense that they have several attributes in common. This commonality can be expressed by generalization, which is a containment relationship that exists between a higher-level entity set and one or more lower-level entity sets.

Higher- and lower-level entity sets also may be designated by the terms superclass and subclass, respectively.

# Special features – Generalization

The *total generalization* demands that every entity in the superclass must belong to some subclass.

The *partial generalization* ensures that the entities in the superclass may not belong to any subclass.

The *disjoint generalization* demands that every subclass must be disjoint.

The *overlapping generalization* ensures that the same entity may belong to more than one subclass within a single generalization.

# Special features – Attribute inheritance

A crucial property of the higher- and lower-level entities created by specialization and generalization is attribute inheritance. The attributes of the higher-level entity sets are said to be inherited by the lower-level entity sets.

# Disadvantages

The limitations of entity-relationship data model are as follows:

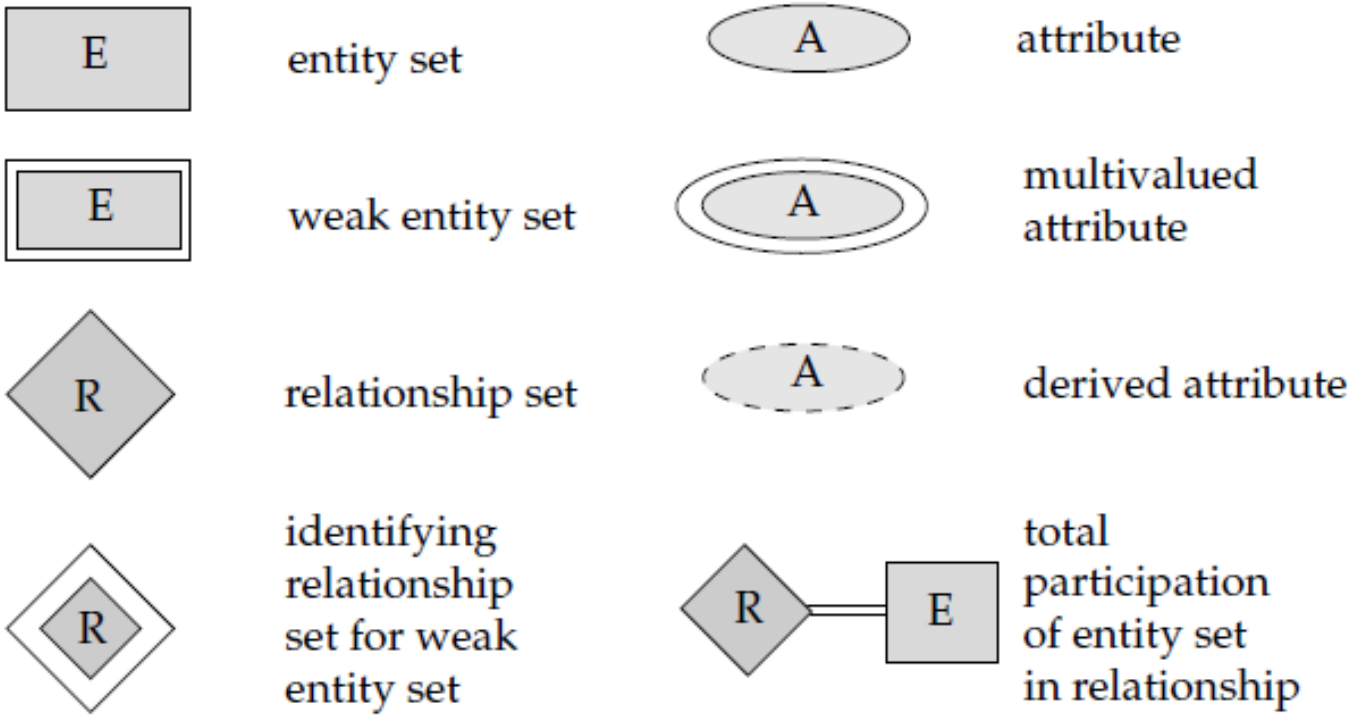
- **Hardware overheads:** It hides the implementation complexities and the physical data storage details from the users, thereby increasing the overhead on the hardware.
- **Ease of design:** As it is easy to design and use, it may lead to bad design.
- **'Information island' phenomenon:** It creates a situation where too many people will come up with their own databases and applications.

# The components in Entity-Relation (E-R) diagram

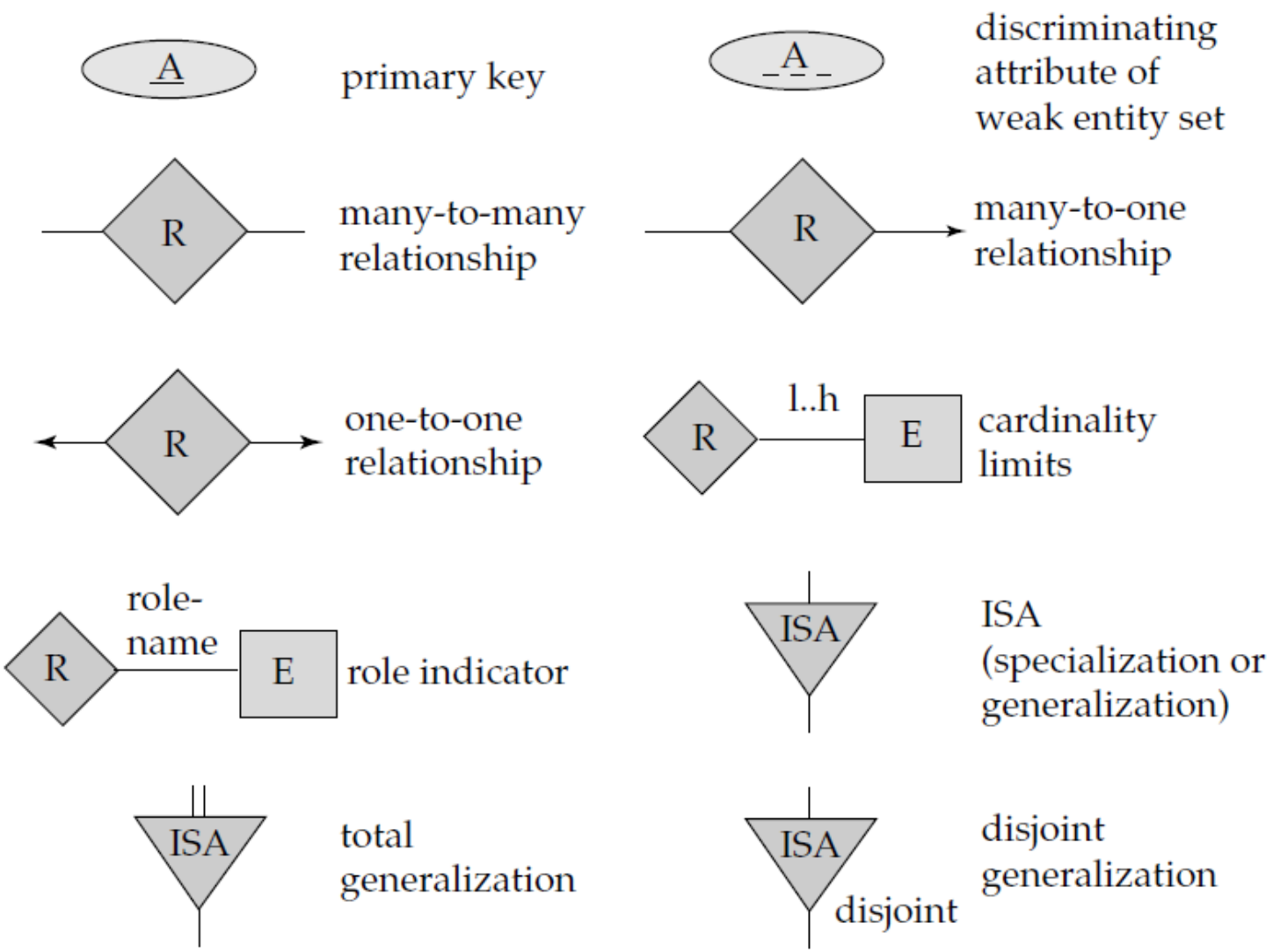
An E-R diagram graphically represents the entire logical structure of a database. It comprises the following components.

- Rectangles
- Double rectangles
- Diamonds
- Double diamonds
- Ellipses
- Double ellipses
- Dashed ellipses
- Lines
- Double lines
- and so on

# The components in Entity-Relation (E-R) diagram

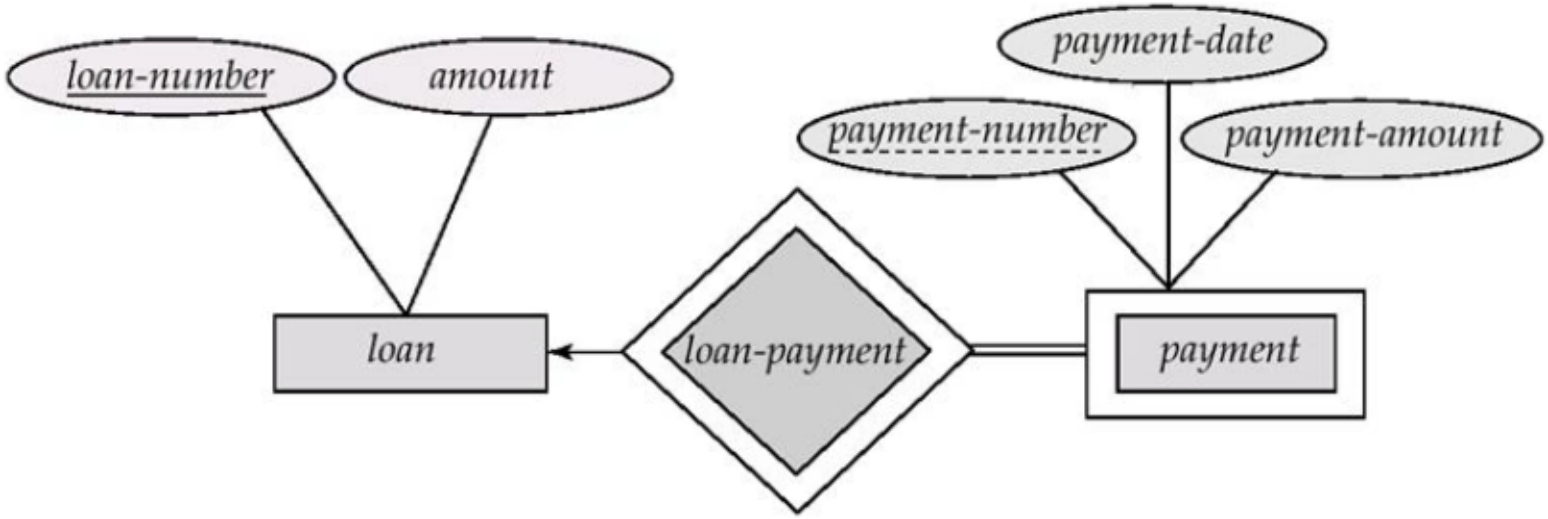


# The components in Entity-Relation (E-R) diagram



# Discriminating attribute of weak entity set

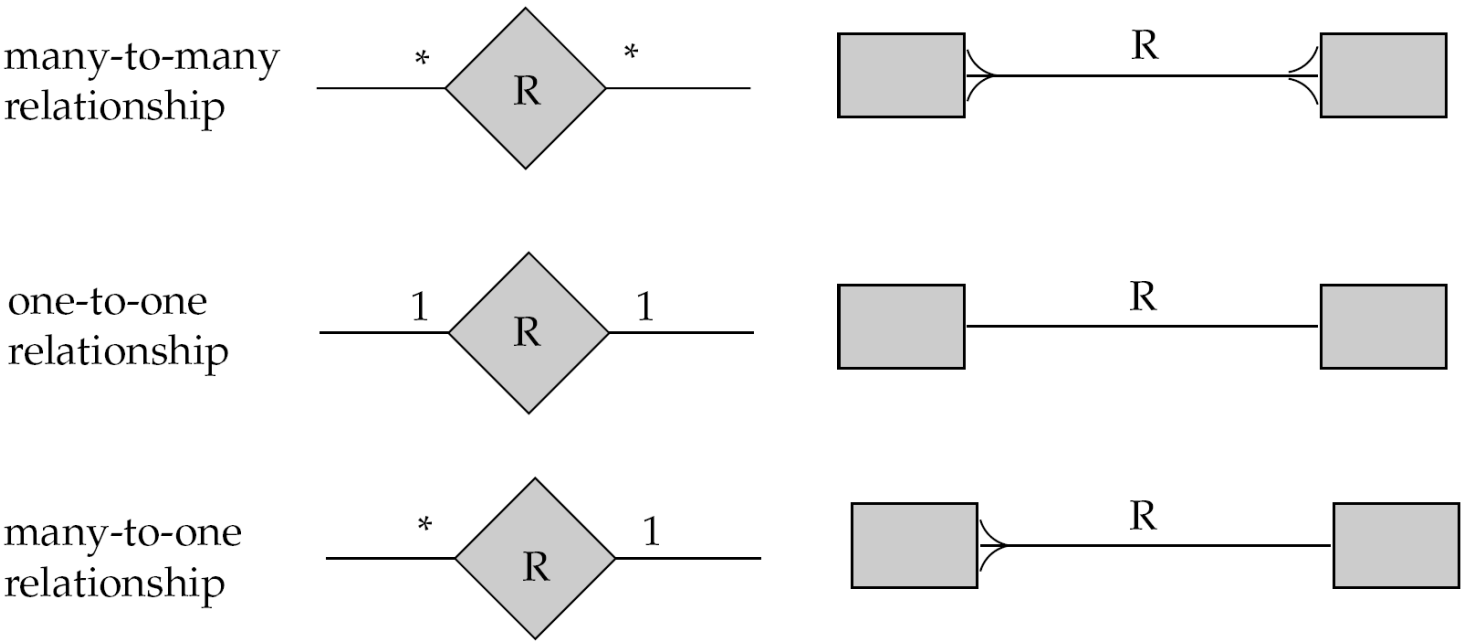
A primary key does not exist for a weak entity set. However, it contains a partial key termed as *discriminating attribute* that can identify a group of entities from the entity set.




The combination of *discriminating attribute* and primary key of the strong entity set makes it possible to uniquely identify all entities of the weak entity set.

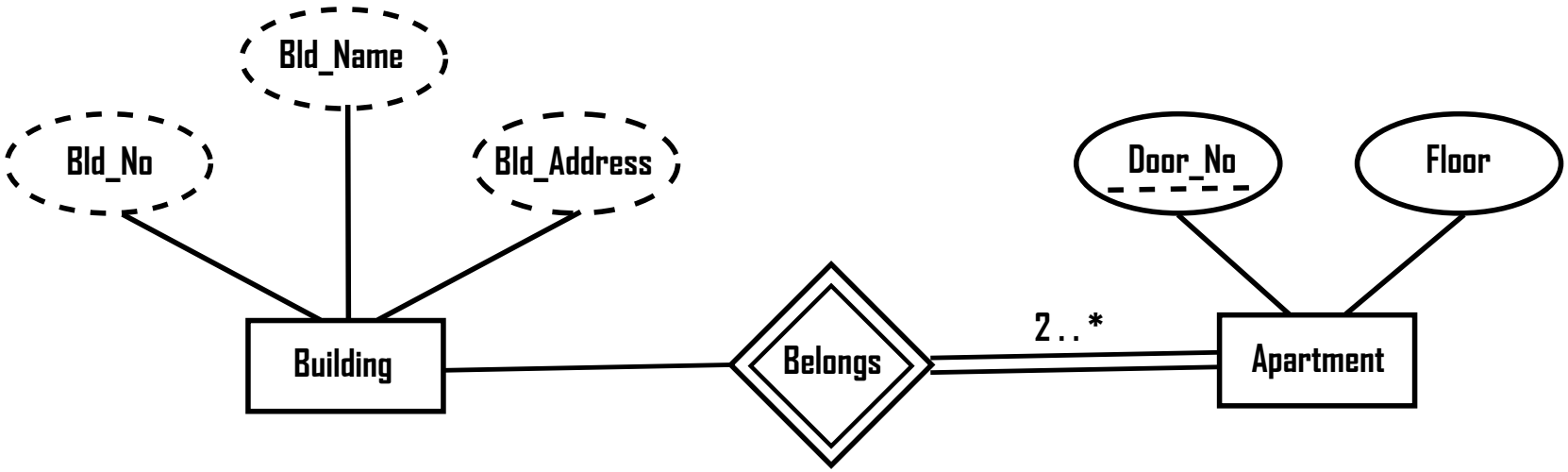
# The components in Entity-Relation (E-R) diagram

The mapping constraints can be represented in multiple alternative ways as shown below.



# Let us brainstorm!!!

Consider the following E-R diagram. It represents the details about real-estate properties of a company. The diagram has been prepared by a novice developer without much understanding of Relational Data Model. Can you list up the inconsistencies present in the diagram?



# Let us brainstorm!!!

Here follow the problems in the given E-R diagram:

(i) All the attributes of the entity set `Building` cannot be of derived type (shown with dashed ellipses). There is no other attribute from which they will be derived.

(ii) If `Belongs` is weak relation (shown with double diamond), either of the `Building` or `Apartment` has to be a weak entity set (shown with double rectangle).

(iii) The mapping cardinality constraints (shown with 2..*) and participation constraints (shown with double line) are conflicting for the entity set `Apartment`. If the minimum cardinality constraint is 2 it cannot be a total participation.

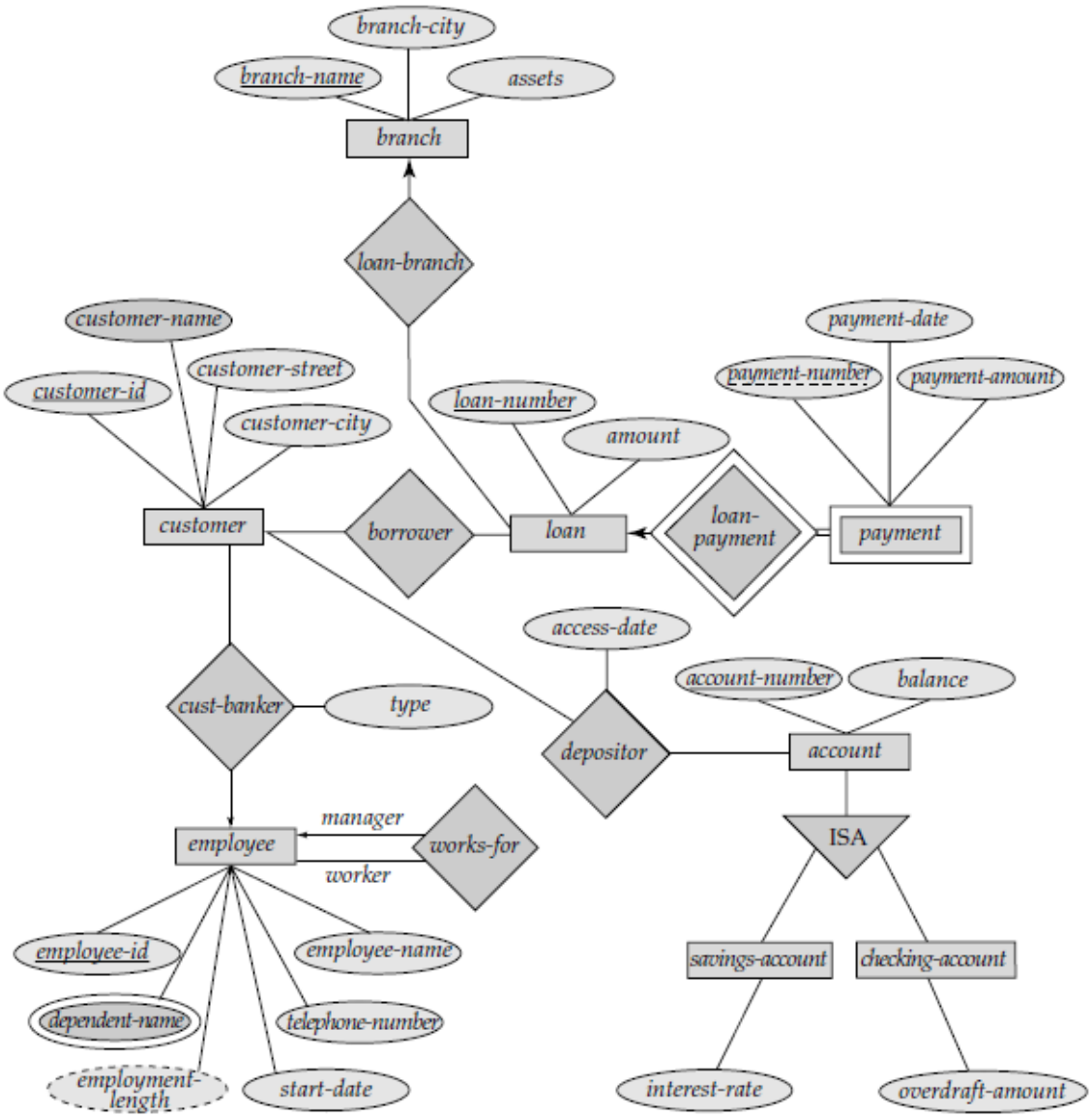
(iv) If there is a discriminating attribute (shown with dashed underline) for the entity set `Apartment`, then there has to be a set of attributes that can serve as the primary key (shown with underline) for the related entity set `Building`.

# Entity-Relation (E-R) diagram – Specialization and generalization

In E-R diagrams, specialization (and also generalization) is depicted by a triangle component labeled ISA (standing for "is a") which denotes that an entity (say a musician) "is a" part of another entity (say a person).

The ISA relationship may also be referred to as a superclass-subclass relationship. Higher- and lower-level entity sets are depicted as regular entity sets—that is, as rectangles containing the name of the entity set.

# The E-R diagram for a banking system



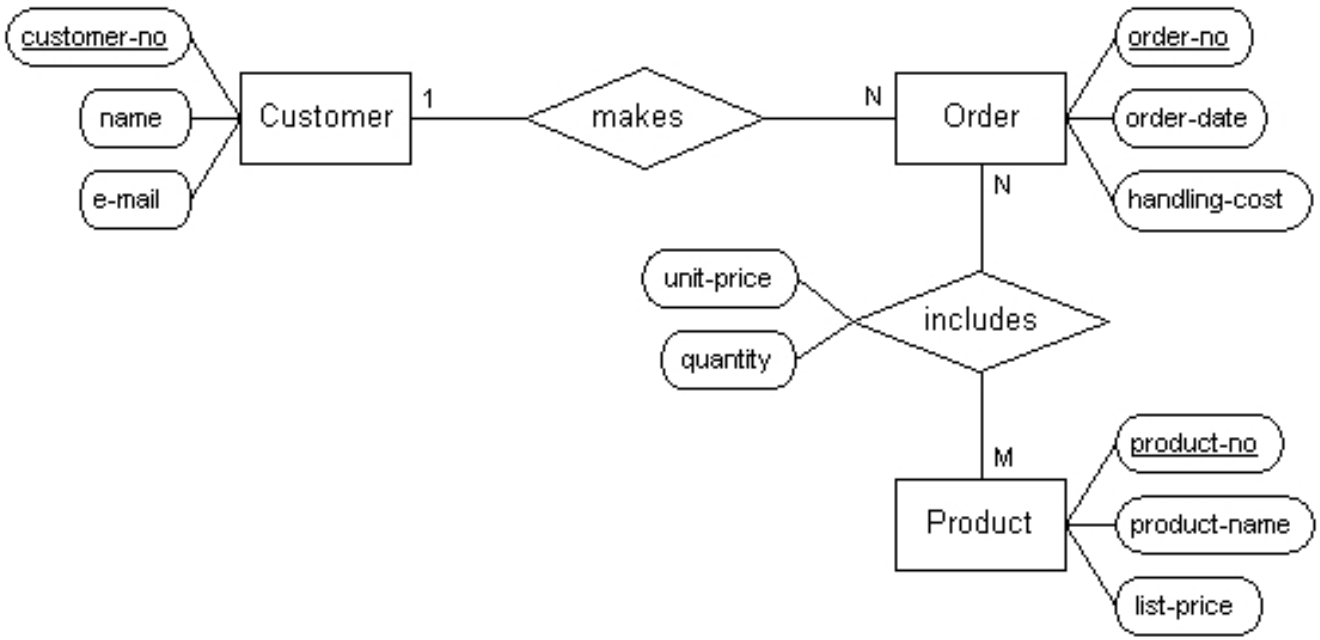
# The E-R diagram for Flipkart Republic Day Sale

# The E-R diagram for Flipkart Republic Day Sale

Flipkart celebrated the 72nd Republic Day of India through sale during January 17-22, 2022. Aiming this, Flipkart stored the details about their products, customers, and the orders that are placed by the customers in a relational database. Prepare an E-R diagram based on the following requirements.

1. Each product has a unique identifier, a name, and a listed price.
2. Each customer has a unique identifier, a name, and an e-mail ID.
3. Every order placed under the Freedom Sale includes a unique identifier, a date, and a handling cost.
4. There is a unit price and quantity associated with each order for a particular product.

# The E-R diagram for Flipkart Republic Day Sale



# The E-R diagram for a music company

Suppose Saregama India Ltd wishes to store the information about their musicians (and other company details) in a database. Prepare an E-R diagram considering the following features.

1. Each musician that records at Saregama has an Aadhaar, a name, an address, and a mobile number. Poorly paid musicians often share the same address, and no address has more than one mobile.
2. Each instrument used in songs recorded at Saregama has a unique ID, a name (e.g., sitar, piano, tabla) and a musical key (e.g., B, B-flat, G-flat).
3. Each album recorded under the Saregama label has a unique ID, a title, a copyright date, and a format type (e.g., CD or VCD). Note that, an album is directed by a musician.
4. Each song recorded at Saregama has a title and an author (lyricist).
5. Each musician may play several instruments, and a given instrument may be played by several musicians.
6. Each album has a number of songs on it, but no song may appear on more than one album.
7. Each song is performed by one or more musicians, and a musician may perform a number of songs.
8. Each album has exactly one musician who acts as its producer. A musician may although produce several albums.

# The E-R diagram for a music company