CS315: Database Systems
Relational Model

Arnab Bhattacharya

arnabb@cse.iitk.ac.in

Computer Science and Engineering,
Indian Institute of Technology, Kanpur
http://web.cse.iitk.ac.in/~cs315/

2nd semester, 2022-23

Tue 10:30-11:45, Thu 12:00-13:15

Relation

- A relation is a subset of the cross-product of sets
- For sets $D_1, D_2, \ldots, D_n$, a relation r is a set of n-ary tuples of the form $(a_1, a_2, \ldots, a_n)$ where each $a_i \in D_i$
- Example
  - `name` = {A, B, C}
  - `designation` = {L, E, W}
  - `identifier` = $\mathcal{N}$
  - r = {(A, E, 4), (B, E, 9), (C, W, 23)} is a relation over `name × designation × identifier`
- Relations are *unordered*

Relation

- A relation is a subset of the cross-product of sets
- For sets $D_1, D_2, \ldots, D_n$, a relation r is a set of n-ary tuples of the form $(a_1, a_2, \ldots, a_n)$ where each $a_i \in D_i$
- Example
  - `name` = {A, B, C}
  - `designation` = {L, E, W}
  - `identifier` = $\mathcal{N}$
  - r = {(A, E, 4), (B, E, 9), (C, W, 23)} is a relation over `name` × `designation` × `identifier`
- Relations are *unordered*
- Generally depicted as a table

name	designation	identifier
A	E	4
B	E	9
C	W	23

Attribute

- Each attribute of a relation has a name
- There is a domain for each attribute
- Attributes are *generally* atomic
  - Indivisible, not sets
- Domain is atomic if all members are atomic
- Special value null in every domain

Relation Schema and Tuple

- The sets define a relation schema
- Example
  - Schema is `Person = (name, designation, identifier)`
- Relations are defined over a schema
- If schema is R, relation is denoted by $r(R)$
  - Example: `persons(Person)`
- A relation instance is a particular instance from the schema
  - Earlier example
- An element of a relation (instance) is a tuple

Relation Schema and Tuple

- The sets define a relation schema
- Example
  - Schema is `Person = (name, designation, identifier)`
- Relations are defined over a schema
- If schema is R, relation is denoted by $r(R)$
  - Example: `persons(Person)`
- A relation instance is a particular instance from the schema
  - Earlier example
- An element of a relation (instance) is a tuple
- Tuples are rows and attributes are columns

Database

- Consists of multiple *inter-related* relations
- Each relation stores information about a particular relationship

Database

- Consists of multiple *inter-related* relations
- Each relation stores information about a particular relationship
- Alternatively, a single relation can store all data
- Problems
    - Data repetition
    - Need for null values
- Normalization theory deals with how to design relation schemas

Key

- $K \subseteq R$ is a superkey of R if and only if values for K are sufficient to identify a unique tuple in *all* possible relations $r(R)$
    - Possible $r(R)$ signifies a relation that can exist from the data that is being modeled
- Example: $\{$name$\}$ is a superkey if each person has a unique name, otherwise not
- All supersets of superkeys are superkeys
    - $\{$name, designation$\}$ is also a superkey
- In practical situations, an id will be used which is guaranteed to be a superkey

Key

- $K \subseteq R$ is a superkey of R if and only if values for K are sufficient to identify a unique tuple in *all* possible relations $r(R)$
  - Possible $r(R)$ signifies a relation that can exist from the data that is being modeled
- Example: {name} is a superkey if each person has a unique name, otherwise not
- All supersets of superkeys are superkeys
  - {name, designation} is also a superkey
- In practical situations, an id will be used which is guaranteed to be a superkey
- A superkey K is a candidate key if K is minimal, i.e., no proper subset of it is a superkey
  - {name} is a candidate key

Key

- $K \subseteq R$ is a superkey of R if and only if values for K are sufficient to identify a unique tuple in *all* possible relations $r(R)$
    - Possible $r(R)$ signifies a relation that can exist from the data that is being modeled
- Example: {name} is a superkey if each person has a unique name, otherwise not
- All supersets of superkeys are superkeys
    - {name, designation} is also a superkey
- In practical situations, an id will be used which is guaranteed to be a superkey
- A superkey K is a candidate key if K is minimal, i.e., no proper subset of it is a superkey
    - {name} is a candidate key
- There may be multiple candidate keys
    - {name}, {identifier} are candidate keys

Key

- $K \subseteq R$ is a superkey of R if and only if values for K are sufficient to identify a unique tuple in *all* possible relations $r(R)$
  - Possible $r(R)$ signifies a relation that can exist from the data that is being modeled
- Example: {name} is a superkey if each person has a unique name, otherwise not
- All supersets of superkeys are superkeys
  - {name, designation} is also a superkey
- In practical situations, an id will be used which is guaranteed to be a superkey
- A superkey K is a candidate key if K is minimal, i.e., no proper subset of it is a superkey
  - {name} is a candidate key
- There may be multiple candidate keys
  - {name}, {identifier} are candidate keys
- Primary key is a candidate key chosen to serve as the primary means of identifying tuples
  - Choice is arbitrary as it depends on the database designer
  - Other candidate keys are called secondary keys

Foreign Key

- A relation schema may have an attribute that is unique (e.g., a primary key) in another schema
- This attribute is then called a foreign key
- Example
    - depositor = (name, number)
    - customer = (name, street, city)
    - account = (number, balance)
    - name and number in depositor are foreign keys
- Values in the foreign key attribute of the referencing relation may only come from those in the primary key of the referenced relation

Entity-Relationship Model

- Entity-Relationship Model: A database is a collection of *entities* and *relationships* among them

Entity-Relationship Model

- Entity-Relationship Model: A database is a collection of *entities* and *relationships* among them
- An entity is an object that exists and is distinguishable from objects, e.g., person “A”
- Entities have attributes, e.g., persons have names
- An entity set is a set of entities of the same type that share the same properties, e.g., a set of persons

Entity-Relationship Model

- Entity-Relationship Model: A database is a collection of *entities* and *relationships* among them
- An entity is an object that exists and is distinguishable from objects, e.g., person “A”
- Entities have attributes, e.g., persons have names
- An entity set is a set of entities of the same type that share the same properties, e.g., a set of persons
- A relationship is an association among the entities
- A relationship set is a relation among two or more entities, each taken from an entity set, e.g., person is a depositor for an account

Entity-Relationship Model

- Entity-Relationship Model: A database is a collection of *entities* and *relationships* among them
- An entity is an object that exists and is distinguishable from objects, e.g., person “A”
- Entities have attributes, e.g., persons have names
- An entity set is a set of entities of the same type that share the same properties, e.g., a set of persons
- A relationship is an association among the entities
- A relationship set is a relation among two or more entities, each taken from an entity set, e.g., person is a depositor for an account
- Relationship sets can also have attributes, e.g., access date for depositor
- Primary keys of entity sets form a superkey of the relationship set

Model Parameters

- Relationships between two entity sets are *binary*
    - It is rare to have more than degree two
- Cardinality constraints for binary relationships can be one-to-one, one-to-many, many-to-one or many-to-many

Model Parameters

- Relationships between two entity sets are *binary*
  - It is rare to have more than degree two
- Cardinality constraints for binary relationships can be one-to-one, one-to-many, many-to-one or many-to-many
- Attributes may be
  - Simple or composite
  - Single-valued or multi-valued (e.g., phone number)
  - Derived (e.g., age)

Model Parameters

- Relationships between two entity sets are *binary*
    - It is rare to have more than degree two
- Cardinality constraints for binary relationships can be one-to-one, one-to-many, many-to-one or many-to-many
- Attributes may be
    - Simple or composite
    - Single-valued or multi-valued (e.g., phone number)
    - Derived (e.g., age)
- Participation can be total (every entity participates in the relationship) or partial (not all participate)

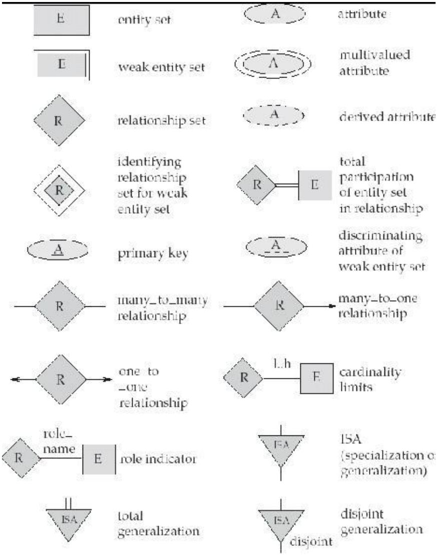
Model Parameters

- Relationships between two entity sets are *binary*
    - It is rare to have more than degree two
- Cardinality constraints for binary relationships can be one-to-one, one-to-many, many-to-one or many-to-many
- Attributes may be
    - Simple or composite
    - Single-valued or multi-valued (e.g., phone number)
    - Derived (e.g., age)
- Participation can be total (every entity participates in the relationship) or partial (not all participate)
- Specialization, generalization, aggregation are features of extended ER model

Weak Entity Sets

- An entity set that does *not* have a primary key is called a weak entity set
- Its existence depends on the existence of another entity set called the identifying entity set or owner entity set
- The identifying relationship set that exists between the two must be *total* and *many-to-one* from the weak entity set
- A weak entity set has a *discriminator* or partial key instead of a primary key
- The discriminator distinguishes weak entities that are related to the same entity of the identifying entity set
- The *primary key* is formed by the primary key of the identifying set and the discriminator

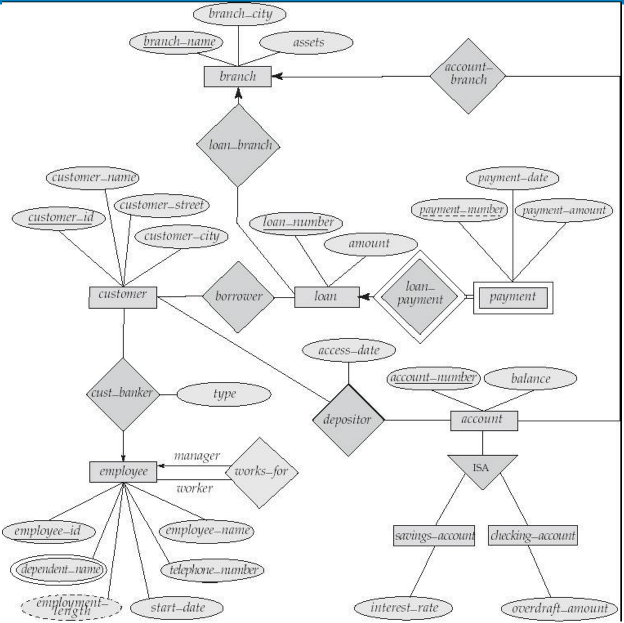
ER Diagram: Summary



ER Diagram: Description

- Entity sets: rectangles
- Relationship sets: diamonds
- Attributes: ellipses
  - Multivalued attributes: double ellipses
  - Derived attributes: dashed ellipses
- Primary keys: underlines
- Roles: on links
- Cardinality constraints
  - One: directed
  - Many: undirected
  - One-to-many: directed-diamond-undirected
- Participation
  - Total: double line
  - Partial: single line
- Cardinality limits: on lines
- Weak entity sets: double rectangle
- Weak relationship set: double ellipse
- Discriminator of a weak entity sets: underline with dashed lines

Example: Banking Schema



Reduction to Relational Model

- Entity sets and relationships sets are reduced uniformly to relations
- A weak entity set is reduced to a relation by including the primary key attributes of the identifying set
    - A foreign key relationship is established
- Many-to-one and one-to-many relationships that are total on the many side may not be reduced to a relation
    - Primary key of entity on "one" side is added to relation of entity on "many" side
- For one-to-one relationships,

Reduction to Relational Model

- Entity sets and relationships sets are reduced uniformly to relations
- A weak entity set is reduced to a relation by including the primary key attributes of the identifying set
  - A foreign key relationship is established
- Many-to-one and one-to-many relationships that are total on the many side may not be reduced to a relation
  - Primary key of entity on “one” side is added to relation of entity on “many” side
- For one-to-one relationships, any side can be chosen as “many”
- If participation is partial,

Reduction to Relational Model

- Entity sets and relationships sets are reduced uniformly to relations
- A weak entity set is reduced to a relation by including the primary key attributes of the identifying set
    - A foreign key relationship is established
- Many-to-one and one-to-many relationships that are total on the many side may not be reduced to a relation
    - Primary key of entity on “one” side is added to relation of entity on “many” side
- For one-to-one relationships, any side can be chosen as “many”
- If participation is partial, null values are used
- Many-to-many relationships

Reduction to Relational Model

- Entity sets and relationships sets are reduced uniformly to relations
- A weak entity set is reduced to a relation by including the primary key attributes of the identifying set
  - A foreign key relationship is established
- Many-to-one and one-to-many relationships that are total on the many side may not be reduced to a relation
  - Primary key of entity on “one” side is added to relation of entity on “many” side
- For one-to-one relationships, any side can be chosen as “many”
- If participation is partial, null values are used
- Many-to-many relationships must be reduced to relations
- Each component of a composite attribute is modeled separately
- Multivalued attributes

Reduction to Relational Model

- Entity sets and relationships sets are reduced uniformly to relations
- A weak entity set is reduced to a relation by including the primary key attributes of the identifying set
    - A foreign key relationship is established
- Many-to-one and one-to-many relationships that are total on the many side may not be reduced to a relation
    - Primary key of entity on “one” side is added to relation of entity on “many” side
- For one-to-one relationships, any side can be chosen as “many”
- If participation is partial, null values are used
- Many-to-many relationships must be reduced to relations
- Each component of a composite attribute is modeled separately
- Multivalued attributes are reduced to relations that include the primary key of the entity set