Database Management System (DBMS)

Lecture-12

Dharmendra Kumar

August 20, 2020

Reduction of an E-R Schema to Tables

Every E-R schema or diagram can be converted into set of tables. In this section, we describe how an E-R schema can be represented by tables. The constraints specified in an E-R diagram, such as primary keys and cardinality constraints, are mapped to constraints on the tables generated from the E-R diagram.

Tabular Representation of Strong Entity Sets

Let E be a strong entity set with descriptive attributes $a_1, a_2, ..., a_n$. We represent this entity by a table called E with n distinct columns, each of which corresponds to one of the attributes of E. Each row in this table corresponds to one entity of the entity set E.

Reduction of an E-R Schema to Tables

Example:

Consider a strong entity set **loan** with attributes loan-number and amount. The table corresponding to loan will be the following

loan-number	amount
L-1	10000
L-2	15000
L-3	20000

Table 1: loan table

Tabular Representation of Weak Entity Sets

Let A be a weak entity set with attributes $a_1, a_2, ..., a_m$. Let B be the strong entity set on which A depends. Let the primary key of B consist of attributes $b_1, b_2, ..., b_n$. We represent the entity set A by a table called A with one column for each attribute of the set:

$$\{a_1, a_2, ..., a_m\} \cup \{b_1, b_2, ..., b_n\}$$

Reduction of an E-R Schema to Tables

Example:

Consider the weak entity set payment with three attributes: payment-number, payment-date, and payment-amount. It depends on strong entity set loan. The primary key of loan is loan-number. Thus, we represent payment entity set by a table with four columns labeled loan-number, payment-number, payment-date, and payment-amount. The table corresponding to payment will be the following

loan-number	payment-number	payment-date	payment-amount
L-1	1	9 May 20018	5000
L-2	3	29 May 20018	6000
L-2	4	20 June 20018	9000

Table 2: payment table

Tabular Representation of Relationship Sets

Let R be a relationship set, let $a_1, a_2, ..., a_m$ be the set of attributes formed by the union of the primary keys of each of the entity sets participating in R, and let the descriptive attributes (if any) of R be $b_1, b_2, ..., b_n$. We represent this relationship set by a table called R with one column for each attribute of the set:

$$\{a_1, a_2, ..., a_m\} \cup \{b_1, b_2, ..., b_n\}$$

Reduction of an E-R Schema to Tables

Example:

Consider the relationship set borrower. This relationship set involves the following two entity sets:

- customer, with the primary key customer-id
- loan, with the primary key loan-number

Assume relationship set borrower has no attributes. Then, the borrower table has two columns, labeled customer-id and loan-number. Borrower table is the following:-

customer-id	loan-number
C-34	L-1
C-45	L-2
C-20	L-3

Table 3: borrower table

Tabular Representation corresponding to Composite Attributes

We handle composite attributes by creating a separate attribute for each of the component attributes; we do not create a separate column for the composite attribute itself.

Example:

Suppose address is a composite attribute of entity set customer, and the components of address are street and city. The table generated from customer would then contain columns address-street and address-city; there is no separate column for address.

Tabular Representation corresponding to Multivalued Attributes

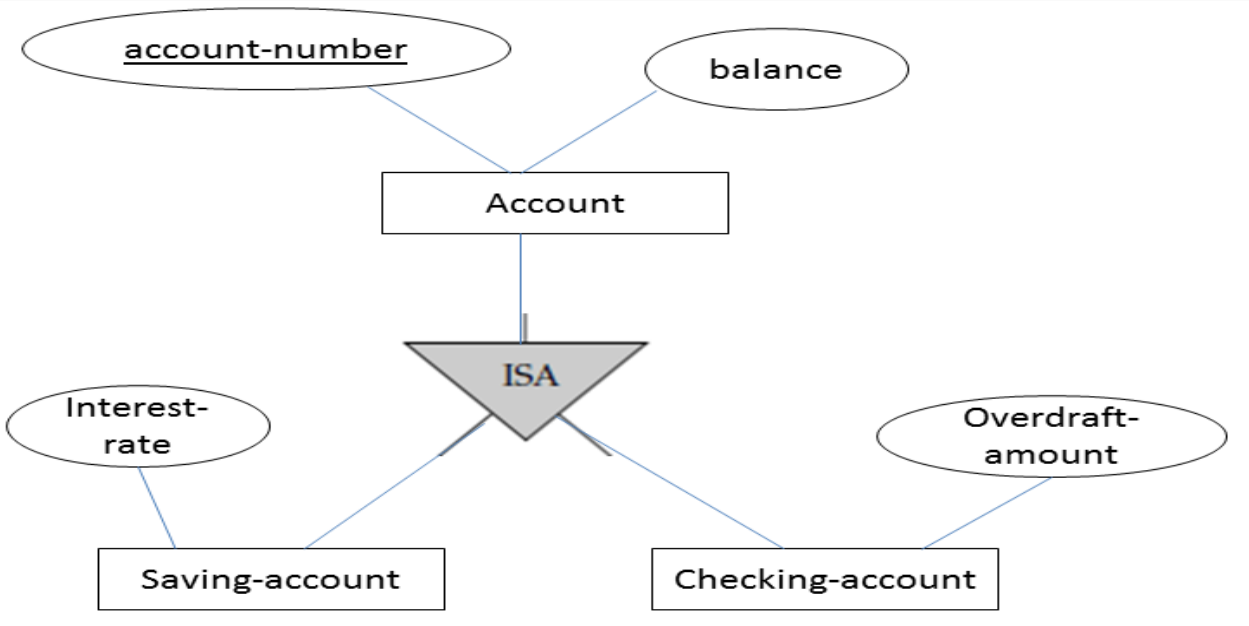
For a multivalued attribute M, we create a table T with a column C that corresponds to M and columns corresponding to the primary key of the entity set or relationship set of which M is an attribute.

Example:

Consider the above E-R diagram for banking enterprise. The diagram includes the multivalued attribute dependent-name. For this multivalued attribute, we create a table dependent-name, with columns dname, referring to the dependent-name attribute of employee, and employee-id, representing the primary key of the entity set employee. Each dependent of an employee is represented as a unique row in the table.

Tabular Representation of Generalization

There are two different methods for transforming to a tabular form an E-R diagram that includes generalization. Consider the following E-R diagram:-



Reduction of an E-R Schema to Tables

1. Create a table for the higher-level entity set. For each lower-level entity set, create a table that includes a column for each of the attributes of that entity set plus a column for each attribute of the primary key of the higher-level entity set.
   For the above E-R diagram, we have three tables:
   - account, with attributes account-number and balance
   - savings-account, with attributes account-number and interest-rate
   - checking-account, with attributes account-number and overdraft-amount

Reduction of an E-R Schema to Tables

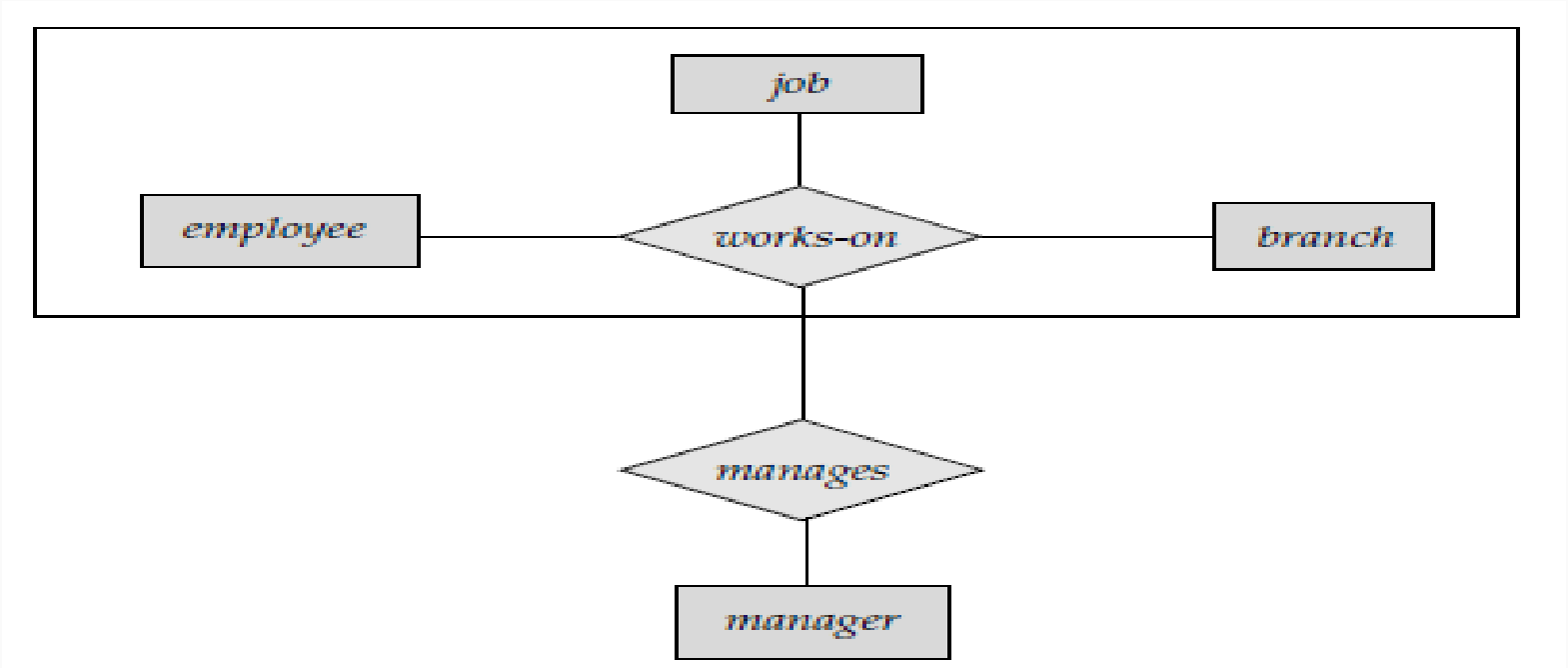
2. If the generalization is disjoint and complete, that is, if no entity is a member of two lower-level entity sets directly below a higher-level entity set, and if every entity in the higher level entity set is also a member of one of the lower-level entity sets. Here, do not create a table for the higher-level entity set. Instead, for each lower-level entity set, create a table that includes a column for each of the attributes of that entity set plus a column for each attribute of the higher-level entity set.

For the above E-R diagram, we have two tables:

- savings-account, with attributes account-number, balance, and interest-rate
- checking-account, with attributes account-number, balance, and overdraft-amount

Tabular Representation of Aggregation

Transforming an E-R diagram containing aggregation to a tabular form is straightforward. Consider the diagram in the following figure:-



Reduction of an E-R Schema to Tables

The table for the relationship set manages between the aggregation of works-on and the entity set manager includes a column for each attribute in the primary keys of the entity set manager and the relationship set works-on. It would also include a column for any descriptive attributes, if they exist, of the relationship set manages. We then transform the relationship sets and entity sets within the aggregated entity.