8.5 Designs the logical schema of a database

Time: 6 periods

Learning Outcomes

- Defines logical schema of a database
- Describes relational schema
- Describes relational instances
- Briefly describes Candidate key, primary key, alternate key and foreign key

8.6 Transforms ER diagrams to logical schema

Time: 6 periods

Learning Outcomes

- Describes the methods of transformation ER diagram to logical schema
- Transforms ER diagrams (entity, attribute, relationships) to logical schema

There are different kinds of database schemas:

- Conceptual schema
- Logical schema
- Physical schema

Logical Database Schema –

This schema defines all the logical constraints that need to be applied on the data stored. It defines tables, views, and integrity constraints.

Relation schema - A set of attributes is called a relation schema (or relation scheme). A relation schema is also known as table schema (or table scheme). A relation schema can be thought of as the basic information describing a table or relation. It is the logical definition of a table. Relation schema defines what the name of the table is. This includes a set of column names, the data types associated with each column.

Relational schema may also refer to as database schema. A database schema is the collection of relation schemas for a whole database. Relational or Database schema is a collection of meta-data. Database schema describes the structure and constraints of data representing in a particular domain. A Relational schema can be described a blueprint of a database that outlines the way data is organized into tables. This blueprint will not contain any type of data. In a relational schema, each tuple is divided into fields called Domains.

Transform ER Diagram into Relational schema

- **Step 1**
  - Each entity type is mapped to a relation,
  - One candidate key of the entity type is chosen as primary key for the relation. The primary key is underlined.
  - Each *simple attribute* is mapped to an attribute in the relation.

  COURSE (<u>code</u>, name, fee, length)

- **Step 2**
  - Each simple component of a composite attribute is mapped to a separate attribute in the corresponding relation.

  STAFF (<u>staff_no</u>, fname, lname)

- **Step 3**

  - Each multi-valued attribute is mapped to a separate relation that includes the primary key of the owner relation.

  - In the example shown below, *branch_no*in the new BRANCH_TEL_NO relation is a foreign key as well as a part of the relation's primary key.

  BRANCH_TEL_NO (branch_no*, tel_no)

  BRANCH (branch_no, fax_no)

- **Step 4**

  - A derived attribute is not included in the set of attributes for the corresponding relation.

  - Derived attributes are not shown in the relational schema.

  STAFF (staff_no, salary, nin)

- **Step 5**

  - Each one-to-many (1:M) relationship is mapped to a foreign key in the relation at the *many-side* (*M-side*) referring to the relation at the *one-side* (*1-side*).

  PURCHASE_ORDER (po_no, po_date, supplier_no*)

  SUPPLIER (supplier_no, supplier_address)

- **Step 6**

  - Each many to many (M: N) relationship is mapped to a separate relation whose set of attributes includes foreign keys referring to the primary keys of both participating relations

  - In the example shown below, prod_code and supplier_no are foreign keys in the new SUPPLIER_PRODUCT relation The primary key of the new relation must also include both these foreign keys, unless a surrogate/artificial key is produced

  SUPPLIER_PRODUCT (prod_code*, supplier_no*, price)

  SUPPLIER (supplier_no, name)

  PRODUCT (prod_code, name)

- **Step 7**

    - Each one to one 1:1 relationship type is mapped to a foreign key from one relation referring to the other relation.

    - The foreign key is placed in the relation corresponding to the entity type that has a **mandatory participation in the relationship type, or the entity type that is created second**.

    - Alternatively, it is possible to map both entity types to a single relation, especially if the participation of both entity types in the 1 to 1 relationship is mandatory.

      DELIVERY (delivery_no, delivery date, po_no*)

      PURCHASE_ORDER (po_no, po_date, supplier_no)

Reference

http://ecomputernotes.com/fundamental/what-is-a-database/relation-and-relational-schema

Lecture notes - Dr. Dilani Wickramaarachchi (University of Kelaniya)

Teachers Guide (2108)