|  |  |
| --- | --- |
|  | **WEEK—2 Lecture-3 hr** |
|  | **Database design** |
| **1** | **Data model; types** |
|  | A **data model**—a collection of concepts that can be used to describe the structure of a database—provides the necessary means to achieve this abstraction  **Categories of Data Models**  Based on the types of concepts they use to describe the database structure data models can be categorized as follows:   * **High-level** or **conceptual data models:** Conceptual data models use concepts such as entities, attributes, and relationships.   An **entity** represents a real-world object or concept, such as an employee or a project from the mini world that is described in the database.  An **attribute** represents some property of interest that further describes an entity, such as the employee’s name or salary.  A **relationship** among two or more entities represents an association among the entities, for example, a works-on relationship between an employee and a project.   * **Low-level** or **physical data models** which provide concepts that describe the details of how data is stored on the computer storage media, typically magnetic disks. An **access path** is a structure that makes the search for particular database records efficient * **Representational** (or **implementation**) **data models**, which provide concepts that, may be easily understood by end users. Representational data models hide many details of data storage on disk but can be implemented on a computer system directly. |
|  |  |
| **2** | **Importance of data modeling** |
|  | Ensuring that the objects are accurately represented.  Allow us to define the relationship between tables, stored procedures and primary and foreign keys.  Help business to communicate within and across organizations  A comprehensive and optimized data model helps create a simplified, logical database that eliminates redundancy, reduces storage requirements, and enables efficient retrieval.   Data models are often used as an aid to communication between the business people defining the requirements for a computer system and the technical people defining the design in response to those requirements.  They are used to show the data needed and created by business processes. |
|  |  |
| **3** | **Overview of database design** |
|  | A good database design is one that:   * Divides your information into subject-based tables to reduce redundant data. * Provides Access with the information it requires to join the information in the tables together as needed. * Helps support and ensure the accuracy and integrity of your information. * Accommodates your data processing and reporting needs. |
|  |  |
| **4** | **Phases of database design;** |
|  | Using high level conceptual data models for database design. The DB design using high level conceptual data models consists of different phases or steps.  **a) Requirement’s collection and analysis:** The database designers interview database users to understand and document their data requirements completely.  **b) Conceptual design:** The conceptual design is a concise description of the data requirements of the users and includes detailed description of the entity types, relationships and constraints.  **c) Logical design**: (Actual implementation of the database) after the conceptual schema, the next step is conceptual schema is transformed from high level data model into the Actual implementation data model using a commercial DBMS.  **d) Physical design**: During this last step the internal storage structure files organization, index, access paths and physical design parameters for the database files are specified.  In parallel, the application programs are designed and implemented. |
|  |  |
| **5** | **Database development life cycle;(from internet)** |
|  | Database Design Tutorial: Learn Data Modeling  The database development life cycle has a number of stages that are followed when developing database systems.  **Requirements analysis**   * **Planning** – This stage of database design concepts are concerned with planning of entire Database Development Life Cycle. It takes into consideration the Information Systems strategy of the organization. * **System definition** – This stage defines the scope and boundaries of the proposed database system.   **Database designing**   * **Logical model** – This stage is concerned with developing a database model based on requirements. The entire design is on paper without any physical implementations or specific DBMS considerations. * **Physical model** – This stage implements the logical model of the database taking into account the DBMS and physical implementation factors.   **Implementation**   * **Data conversion and loading** – this stage of relational databases design is concerned with importing and converting data from the old system into the new database. * **Testing** – this stage is concerned with the identification of errors in the newly implemented system. It checks the database against requirement specifications. |
|  |  |
| **6** | **Conceptual design: ER-Model: entity: types;** |
|  | **Entities:** Entity is a basic unit in ER model. An **entity** is a real world object with an independent existence.  An entity may be an object with a physical existence (example person, car, house, or employee) or it may be an object with a conceptual existence (for instance, a company, a job, or a university course). |
|  |  |
| **7** | **Attribute: types** |
|  | **Attributes:** An attribute is a properties that describes an entity.  **Ex:** Employee’s name, age, address, salary are the attributes of the entity employee.  Several types of attributes occur in the ER model.  **a) Simple versus composite**  Attributes that are not divisible are called simple or atomic attributes.  **Ex:** Age of employee entity.  Attributes that can be divided into smaller subparts. Which represent more basic attributes with independent meanings.  **Ex:** name of employee.  **b) Single valued versus multivalued attributes**  Attributes that have a single value for a particular entity is called multivalued attributes.  **Ex:** age  Attributes that have a set of values for the same entity is called multivalued attributes.  **Ex:** Qualification  **c) Stored versus Derived Attributes**  For an employee entity, the value of age can be determined from the current date and DOB attributes.  The Age attributes is hence called a derived attribute.  DOB is called a stored attribute.  **d) Null Values or Null attribute**  In some cases, a particular entity may not have an applicable value for an attribute.  **Ex:** phone no.  **e) Complex attribute**  Composite and Multivalued attribute can be nested, such attribute are called Complex attributes.  **Ex:** If a person can have more than one residence and each residence can have multiple phones |
|  |  |
| **8** | **Relationships: types, constraints, Symbols and Notations** |
|  | **Relationship Types, Relationship sets, Roles and Structural constraints**  In the ER model, an attribute of one entity type refers to another entity type (some relationship exists) these references are represented as relationships.  **Ex:** The attribute manager of Dept refers to an employee who manages the department.  ***Relationship types: (Relationship set)***  A relationship is an association among 2 or more entities  A relationship type R among n entity sets E1, E2……En defines a set of associations among entities from there types.    **Relationship Instance:**  An instance of a relationship set is a set of values of relationship.    **Degree of a Relationship type:**  The degree of a relationship type is the number of participating entity types.  **Unary Relationship:** If the no. of participating entity sets is only one.    **Binary Relationship:** If the no. of participating entity sets is only two.    **Ternary Relationship:** If the no. of participating entity sets is only three.    **Quaternary Relationship:** If the no. of participating entity sets is four.    **Nary Relationship:** If the no. of participating entity sets is n.  **Role Names:** The role name signifies the role that a participating entity from the entity type plays in each relationship instance and helps to explain what the relationship means.  **Ex:** In the work\_for relationship type, Employee plays the role of employee or worker and Department plays the role of department or employer.  In such case the same i.e. role name becomes essential for distinguishing the meaning of each participation such relationship types are called **recursive relationships**  **Ex:** The supervises relationship type both supervisee and supervisor are members of the same entity type employee.    ***Constraints on Binary relationship types***  The two main types of binary relationship constraints.  Cardinality ratio  Participation  ***a) Cardinality ratio:***  The Cardinality ratio for a binary relationship specifies the maximum no. of relationship instances that an entity can participate.  i. One to one (1:1)  ii. One to many (1:m) / many to one (m:1)  iii. Many to many (m:n)  **One to one (1:1) Relationships:**  An entity in entity set A is associated with at most one entity in entity set B.  **Ex:** an employee can manage only one department and that department has only one manager.    **One to many / many to one relationship:**  An entity in A is associated with any no. of entities in B (1: N)  An entity in B can associated with at most one entity in A.  **Ex:** an employee can work for at most one dept and dept have many employees.    **Many to many (m: m)**  An entity in A is associated with any no. of entities in B  An entity in B can associated with any no. of entities in A.  **Ex:** an employee can work on many projects and a project can be handled by many employees.    ***b)* *Participating Constraints***  The participation constraint specifies whether the existence of an entity depends on its being related to another entity.  There are 2 types of participation constraints  **i.** Total participation (Existence Dependency)  **ii.** Partial Participation  **Total Participation:** The participation constraint b/w employees and department with work for relationship is total, because every employee should work for one or other department.    If every entity participates in at least one relationship instance in R  **Partial Participation:** If some entities in entity type participate in Relation R in called partial Participation.  **Ex:** Not all employees manage the department.  **c) Structural Constraints:** The cardinality ratio and Participation Constraints together known as structural constraints.  **Attributes of Relationship Types:**  Relationship types can also have attributes, similar to those of entity types.  **Ex:** To record the no. of hrs per week that an employee works on a particular project. We can include an attribute hours for the works\_on relationship type. |
|  |  |
| **9** | **Case study: conceptual design for a set of specifications**  **i. Restaurant**  **ii. Retail shop**  **iii. Recruitment**  **iv. College**  **v. Library** |
|  | **Conceptual design** is the first stage in the database design process.  The goal at this stage is to design a database that is independent of database software and physical details.  The output of this process is a conceptual data model that describes the main data entities, attributes, relationships, and constraints of a given problem domain.  This design is descriptive and narrative in form.  **i. Restaurant**  D:\DBMS\recruitment.jpeg |