**ST. XAVIER’S COLLEGE**

**(Affiliated to Tribhuvan University)**

Maitighar, Kathmandu



**Database Management System**

**Lab Assignment #4**

**Submitted by:**

Arun Sanjel

013BSCCSIT010

**Submitted to:**

|  |  |
| --- | --- |
| **Er. Sanjay Kumar Yadav**  Lecturer  St. Xavier’s College |  |

**Date of Submission: August 10, 2015**

# E-R diagram with case study

# Design

## Functional Design

Functional design is the kind of design that is agreed between software architects and business customers. It describes what the system has to do in a way that both parties can understand and agree. It might describe user interface requirements, it might discuss database access, it might discuss web services or other forms of communication, it might discuss service level agreements or clustering and redundancy. The point is that it should be detailled enough to become the agreed definition of what the system will do.The process of functional design begins with the goal of the product: a clear statement of what it is supposed to do. This does not mean that what the client wants it to do is the only thing that the user will, in fact do with it. It does need to do well what it was made to do.

## Database Design

Database design involves identifying the existing relationships between separate pieces of data and mapping out those relationships in an organized way that makes sense. There are several types of database design: conceptual database design, logical database design, and physical database design.

**Conceptual Database design**

Once all the requirements have been collected and analyzed, the next step is to create a conceptual shema for the database, using a high level conceptual data model. This phase is called conceptual design.The result of this phase is an Entity-Relationship (ER) diagram or UML class diagram. It is a high-level data model of the specific application area. It describes how different entities (objects, items) are related to each other. It also describes what attributes (features) each entity has. It includes the definitions of all the concepts (entities, attributes) of the application area.During or after the conceptual shema design, the basic data model operations can be used to specify the high-level user operations identified during the functional analysis. This also serves to confirm that the conceptual schema meets all the indenfied functional requirements.

**Logical Database design**

The process of logical design involves arranging data into a series of logical relationships called entities and attributes. An entity represents a chunk of information. In relational databases, an entity often maps to a table. An attribute is a component of an entity and helps define the uniqueness of the entity.

**Physical Database Design**

Physical database design translates the logical data model into a set of SQL statements that define the database. For relational database systems, it is relatively easy to translate from a logical data model into a physical database. Rules for translation: Entities become tables in the physical database. The goal of the last phase of database design, physical design, is to implement the database. At this phase one must know which database management system (DBMS) is used. For example, different DBMS's have different names for datatypes and have different datatypes.

# Characteristic of relations

# ER to Relational mapping algorithm

## Mapping of regular entity types

* For each regular (strong) entity type E in the ER schema, create a relation R that includes all the simple attributes.
* Include only the simple component attributes of a composite attribute.
* Choose one of key attributes of E as primary key of R.
* If the chosen key of E is composite, the set of simple attributes that form it will together form the primary key of R.

## Mapping of weak entity types

* For each weak entity type W in the ER schema with owner entity type E,
  + create a relation R, and
  + include all simple attributes (simple component of composite attributes) of W as attributes of R
* n addition, include as *foreign key* attributes of R the primary key attribute(s) of the owner entity type(s);
* This takes care of the identifying relationship type of W
* The *primary key* of R is the combination of the primary key(s) of the owner(s) and the partial key of the weak entity type W, if any.
* It is common to choose the propagate (CASCADE) option for the referential triggered action on the foreign key in the identifying entity relation of the weak entity type, since a weak entity has an existence dependency on its owner entity.
* This can be used for both ON UPDATE and ON DELETE.

## Mapping of binary 1:1 relationship types

* For each binary 1:1 relationship type R in the ER schema, and S and T, the relations that participate in R
  + Choose one relation, for example S, and
  + include as foreign key in S the primary key of T.
* It is better to choose an entity type with *total participating* in R as the role of S.
* Include all simple attributes ( or simple component of composite attributes) of the 1:1 relationship type R as attributes of S.

## Mapping of binary 1:M relationship types

## Mapping of binary m:n relationship types

## Mapping of multivalued attribute

## Mapping of N ary relationship types.