ST. XAVIER’S COLLEGE

**Maitighar,Kathmandu**

**(Affiliated to Tribhuvan University)**



**Database Management System**

**Theory Assignment #4**

**Submitted By**

Ajita Khatiwada

B.Sc. CSIT

Year II/IV Semester

013BSCIT004

**Submitted To:**

Er. Sanjay Kumar Yadav

Lecturer,

Department of Computer Science

St. Xavier’s College

Maitighar, Kathmandu

**Submitted On:**

10 August , 2015

# Design

1. **Functional Design:**

* Functional Design is a paradigm used to simplify the design of hardware and software devices such as computer software and increasingly, 3D models.
* A functional design assures that each modular part of a device has only one responsibility and performs that responsibility with the minimum of side effects on other parts.
* The advantage for implementation is that if a software module has a single purpose, it will be simpler, and therefore easier and less expensive, to design and implement.
* Systems with functionally designed parts are easier to modify because each part does only what it claims to do.
* Since maintenance is more than 3/4 of a successful system's life,this feature is a crucial advantage. It also makes the system easier to understand and document, which simplifies training. The result is that the practical lifetime of a functional system is longer.
* In a system of programs, a functional module will be easier to reuse because it is less likely to have side effects that appear in other parts of the system.

1. **Database Design:**

* Database design is the process of producing a detailed data model of a database.
* This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database.
* A fully attributed data model contains detailed attributes for each entity.
* The process of doing database design generally consists of a number of steps which will be carried out by the database designer. Usually, the designer must:
  + Determine the relationships between the different data elements.
  + Superimpose a logical structure upon the data on the basis of these relationships.

1. **Conceptual Database Design:**

* Once a database designer is aware of the data which is to be stored within the database, they must then determine where dependency is within the data.

Sometimes when data is changed you can be changing other data that is not visible.

* For example, in a list of names and addresses, assuming a situation where multiple people can have the same address, but one person cannot have more than one address, the address is dependent upon the name. When provided a name and the list the address can be uniquely determined; however, the inverse does not hold - when given an address and the list, a name cannot be uniquely determined because multiple people can reside at an address. Because an address is determined by a name, an address is considered dependent on a name.
  + 1. Logical Database Design:
* Once the relationships and dependencies amongst the various pieces of information have been determined, it is possible to arrange the data into a logical structure which can then be mapped into the storage objects supported by the database management system.
* In the case of relational databases the storage objects are tables which store data in rows and columns.
  + 1. Physical Database Design:
* The physical design of the database specifies the physical configuration of the database on the storage media.
* This includes detailed specification of data elements, data types, indexing options and other parameters residing in the DBMS data dictionary.
* It is the detailed design of a system that includes modules & the database's hardware & software specifications of the system.

.

**Mapping Algorithm**

**Mapping of Regular Entity Types.**

◦ For each regular (strong) entity type E, create a relation R that includes all the simple attributes of E.

◦ Choose one of the keys of E as the primary key for R.

◦ If the chosen key of E is composite, the set of simple

attributes together form the primary key of R.

Example: Create the relations EMPLOYEE,

DEPARTMENT, a OJ C co espo g to nd PROJECT corresponding to

the regular entities in the ER diagram.

◦ SSN, DNUMBER, and PNUMBER are the primary keys

for the relations EMPLOYEE, DEPARTMENT, and

PROJECT as shown.

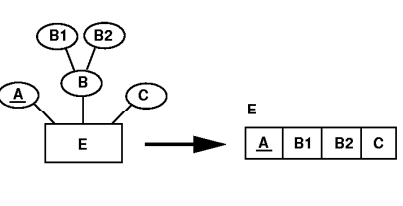


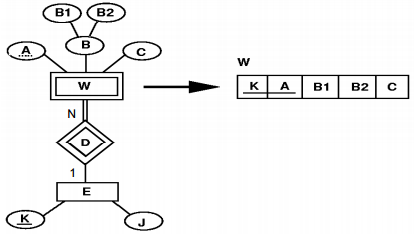
Fig :E-R diagram for regular entity types.

**Mapping of Weak Entity Types**

◦ For each weak entity type W in the ER schema with owner entity create a relation R & include all simple attributes (or simple components of composite attributes) of W as attributes of R.

◦ Also, include as foreign key attributes of R the primary key attribute(s) of the owner(s).

◦ 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.



**Mapping of Binary 1:1 Relation**

Types ◦ For each binary 1:1 relationship type R in the ER schema, identify the relations S and T that correspond to the entity types participating in R.

Three possible approaches:

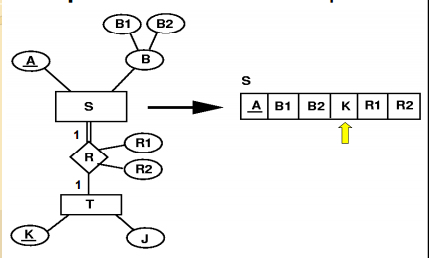
1. Foreign Key approach: Choose one of the relations-say Sand include a foreign key in S the primary key of T. It is better to choose an entity type with total participation in R in the role of S role of S. Example: 1:1 relation MANAGES is mapped by choosing the participating entity type DEPARTMENT to serve in the role of S, because its participation in the MANAGES relationship type is total.

2. Merged relation option:

An alternate mapping of a 1:1 relationship type is possible by merging the two entity yp p t es and the relationship into a single relation. This may be appropriate when both participations are total.

3. Cross-reference or relationship relation option:

The third alternative is to set up a third relation R for the purpose of cross-referencing the primary keys of the two relations S and T representing the entity types.

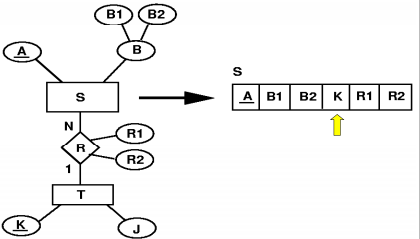


**Mapping of Binary 1:N Relationship Types.**

◦ For each regular binary 1:N relationship type R For each regular binary 1:N relationship type R, identify the relation S that represent the participating entity type at the N-side of the relationship type.

◦ Include as foreign key in S the primary key of the relation T that rep y yp resents the other entity type participating in R.

◦ Include any simple attributes of the 1:N relation type as attributes of S

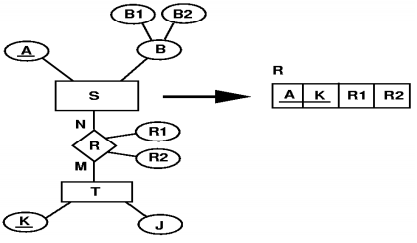


**Mapping of Binary M:N Relationship Types.**

For each regular binary M:N relationship type R, create a new relation S to represent R.

◦ Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types; their combination will form the primary key of S.

◦ Also include any simple attributes of the M:N relationship type (or simple components of composite attributes) as attributes of S.

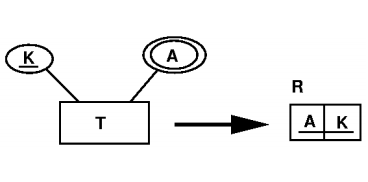


**Mapping of Multivalued attributes.**

◦ For each multivalued attribute A, create a new relation R.

◦ This relation R will include an attribute corresponding to A, plus the primary key attribute K-as a foreign key in R-of the relation that represents the entity type of relationship type that has A as an attribute.

◦ The primary key of R is the combination of A and K. If the multivalued attribute is composite, we include its simple components.

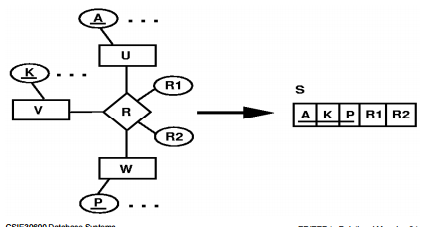


**Mapping of N-ary Relationship Types.**

◦ For each n-ary relationship type R where n>2 ary relationship type R, where n>2, create a new relation S to represent R.

◦ Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types.

◦ Also include any simple attributes of the n-ary relationship type (or simple components of composite attributes) as attributes of S.[1]



References :

[]“https://courses.cs.washington.edu/courses/cse444/12sp/lectures/lecture02-relational-model.pdf”

[1]”[http://www.csie.ndhu.edu.tw/~showyang/DB2010f/09ERtoRelational.pdf, August](http://www.csie.ndhu.edu.tw/~showyang/DB2010f/09ERtoRelational.pdf,%20August)-10-2015.