ST. XAVIER’S COLLEGE

**(Affiliated to Tribhuvan University)**

**Maitighar, Kathmandu**

****

**Database Management System**

**Theory Assignment (#4)**

**SUBMITTED BY**

**Pratik Gautam**

**013BSCCSIT029**

**4th sem/ 2nd year**

**SUBMITTED TO**

**Er. Sanjay K Yadav**

**Lecturer**

**Department of Computer Science**

**St. Xavier’s College**

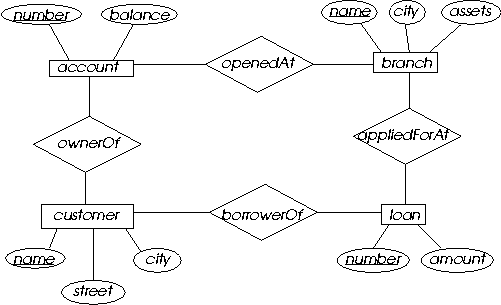
**Date of submission: 2072/ 04/24**

1. **E-R DIAGRAM WITH ONE CASE STUDY**

**DESIGN**

The main components of ER models are entities (things) and the relationships that can exist among them. An ER model is typically implemented as a database. In the case of a relational database, which stores data in tables, every row of each table represents one instance of an entity. Some data fields in these tables point to indexes in other tables; such pointers are the physical implementation of the relationships.

Following figure show a simple E- R diagram of customer at bank.

****

**FUNCTIONAL DESIGN**

1. **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. Logical and physical database designs are perhaps the most straightforward. Conceptual database design is a bit more ambiguous because during this phase there is no direct work on a database model.

**CONCEPTUAL DATABASE DESIGN**

A conceptual data model is a summary-level data model that is most often used on strategic data projects.  It typically describes an entire enterprise.  Due to its highly abstract nature, it may be referred to as a conceptual model.

**LOGICAL DATABASE DESIGN**

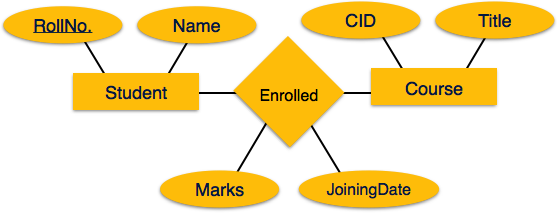
A logical data model is a fully-attributed data model that is independent of DBMS, technology, data storage or organizational constraints.  It typically describes data requirements from the business point of view.  While common data modeling techniques use a relational model notation, there is no requirement that resulting data implementations must be created using relational technologies.

**PHYSICAL DATABASE DESIGN**

A physical data model is a fully-attributed data model that is dependent upon a specific version of a data persistence technology.  The target implementation technology may be a relational DBMS, an XML document, a NoSQL data storage component, a spreadsheet or any other data implementation option.

1. **CHARACTERISTICS OF DATABASE DESIGN**
2. **ER TO RELATIONAL MAPPING ALGORITHM**
3. **MAPPING OF REGULAR ENTITY TYPE**

A relationship is an association among entities.

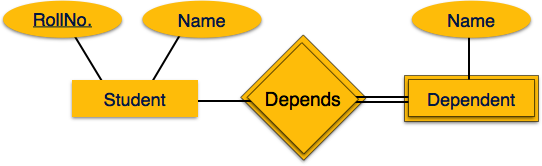


Mapping Process

* Create table for a relationship.
* Add the primary keys of all participating Entities as fields of table with their respective data types.
* If relationship has any attribute, add each attribute as field of table.
* Declare a primary key composing all the primary keys of participating entities.
* Declare all foreign key constraints.

1. **MAPPING OF WEAK ENTITY TYPES**

A weak entity set is one which does not have any primary key associated with it.



Mapping Process

* Create table for weak entity set.
* Add all its attributes to table as field.
* Add the primary key of identifying entity set.
* Declare all foreign key constraints.

1. **MAPPING OF BINARY ONE : ONE RELATION TYPES.**
   * Identify one entity as “parent”
   * Other entity as “child”
   * As general rule,   
     PK of parent is added to child as FK
   * Any attributes of the relationship

-are added to *child* relation

1. **MAPPING OF BINARY ONE: N RELATION TYPE.**
   * Relation at “1” end is parent
   * Relation at “many” end is child
   * include parent's PK in child as foreign key.
   * Any attributes of the relationship are added to child
2. **MAPPING OF BINARY M: N RELATIONSHIP TYPE.**
   * For each binary many-to-many relationship type create a new relation.
   * Add PKs of both “parents” to the new relation (as FKs) and also any attributes of the relationship.
   * PK of the new relation is usually composite: simply combine both FKs. If this is not unique, include additional fields as needed
3. **MAPPING OF MULTI VALUED ATTRIBUTES.**
4. **MAPPING OF AN –ARY RELATIONSHIP TYPE**