ST. XAVIER’S COLLEGE

**(Affiliated to Tribhuvan University)**

**Maitighar, Kathmandu**

****

**DATABASE MANAGEMENT SYSTEM**

**THEORY ASSIGNMENT#4**

**Submitted by:**

**Shreesha Pokharel**

013BSCCSIT038

**Submitted to:**

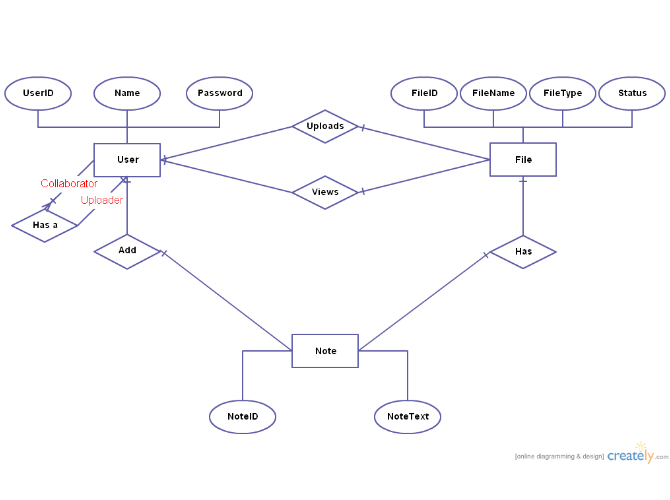
|  |  |
| --- | --- |
| **Er. Sanjay Kumar Yadav** |  |

Lecturer

Department of Computer Science

Date of submission: 10th August, 2015

1. **ER DIAGRAM WITH ONE CASE STUDY.**



1. **DESIGN**

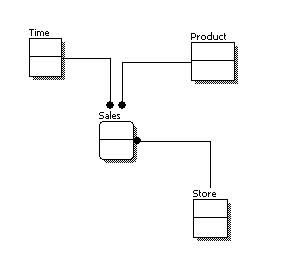
The design process includes:

* **Determine the purpose of the database** - This helps prepare for the remaining steps.
* **Find and organize the information required** - Gather all of the types of information to record in the database, such as product name and order number.
* **Divide the information into tables** - Divide information items into major entities or subjects, such as Products or Orders. Each subject then becomes a table.
* **Turn information items into columns** - Decide what information needs to be stored in each table. Each item becomes a field, and is displayed as a column in the table. For example, an Employees table might include fields such as Last Name and Hire Date.
* **Specify primary keys** - Choose each table’s primary key. The primary key is a column, or a set of columns, that is used to uniquely identify each row. An example might be Product ID or Order ID.
* **Set up the table relationships** - Look at each table and decide how the data in one table is related to the data in other tables. Add fields to tables or create new tables to clarify the relationships, as necessary.
* **Refine the design** - Analyze the design for errors. Create tables and add a few records of sample data. Check if results come from the tables as expected. Make adjustments to the design, as needed.
* **Apply the normalization rules** - Apply the data normalization rules to see if tables are structured correctly. Make adjustments to the tables.
  1. **Functional Design**
  2. **Database Design**

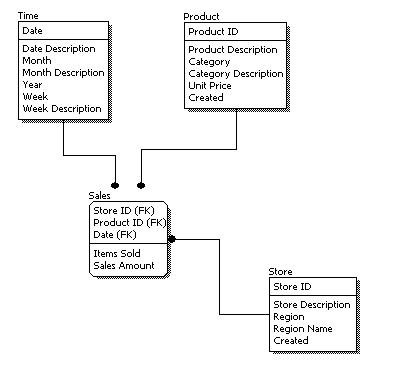
**Database design** is the process of producing a detailed [data model](https://en.wikipedia.org/wiki/Data_model) of a [database](https://en.wikipedia.org/wiki/Database). This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a [data definition language](https://en.wikipedia.org/wiki/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:

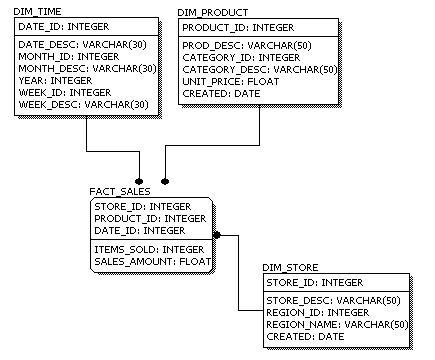
* Determine the relationships between the different data elements.
* Superimpose a logical structure upon the data on the basis of these relationships.
  + 1. **Conceptual Database Design**



* + 1. **Logical Database Design**

****

* + 1. **Physical Database Design**

****

1. **CHARACTERISTICS OF RELATION**
   * Data in the relational database must be represented in tables, with values in columns within rows.
   * Data within a column must be accessible by specifying the table name, the column name, and the value of the primary key of the row.
   * The DBMS must support missing and inapplicable information in a systematic way, distinct from regular values and independent of data type.
   * The DBMS must support an active on-line catalogue.
   * The DBMS must support at least one language that can be used independently and from within programs, and supports data definition operations, data manipulation, constraints, and transaction management.
   * Views must be updatable by the system.
   * The DBMS must support insert, update, and delete operations on sets.
   * The DBMS must support logical data independence.
   * The DBMS must support physical data independence.
   * Integrity constraints must be stored within the catalogue, separate from the application.
   * The DBMS must support distribution independence.  The existing application should run when the existing data is redistributed or when the DBMS is redistributed.
   * If the DBMS provides a low level interface (row at a time), that interface cannot bypass the integrity constraints.
2. **ER TO RELATIONAL MAPPING ALGORITHM**

**Step1**: mapping of regular entity type

**Step 2:** mapping of weak entity type

**Step 3:** mapping of binary 1:1 relation type

**Step 4**: mapping of binary 1:M relation type

**Step 5:** mapping of binary M:M relation type

**Step 6**: mapping of multivalued attributes

**Step 7:** mapping of N-array relationship type