Module 7 Homework

# DBMS and its components-

Database management system or in short DBMS is a common system followed by enterprises to store, organize and access data in a way that avoids duplication and maintains the relationship between objects. DBMS design is also referred to as relational database or relational model designs. It can also be defined as the collection of various tools that enables end users to manipulate and read and parse and analyze the data that is stored in the databases. This ensures the following

1. Scalability
2. Cost Reduction
3. Enforcing standards
4. Security

Components of DBMS are as follows

1. Interfaces – This is required for users to interact with the databases to perform CRUD (CREATE, READ, UPDATE, DELETE) operations as well as for the database administrator to perform maintenance and support operations.
2. Data manipulation language (DML) this controls the following operations like
   1. Storing
   2. Sorting
   3. Updating
   4. Deleting the data
3. Schema – A schema is defined as a database with all the related fields, description of the fields, the set of tables and the relationship among those tables. A schema can also have a sub schema and also have controls over what user can have what level of access on the tables under that schema.
4. Physical Data Repository- Repository in general means a place to store metadata of an object. This data will be accessed by system or user on request. Physical Data repository is no different. It consists of a data dictionary regarding all the objects like tables, schemas and their relationship. This repository can either be centralized or distributed across multiple locations.

# Keys in a database

1. Primary key –

A field or a combination of fields that could uniquely identify a record is a called a primary key. This key will have no additional information that is not required to identify the row distinctly

Example – A sale record can have a number that is associated with record as the primary key for the sale table.

|  |  |  |
| --- | --- | --- |
| Sale ID | Description | Name |
| 1 | Test sale | Test |
| 2 | Bike sale | Bike |

1. Candidate key – Any field or a combination of fields that can be used as primary keys are called candidate keys

Example – In the following table either Member ID or Person ID can be called candidate keys

|  |  |  |
| --- | --- | --- |
| ID | Member ID | Person ID |
| 1 | 001 | A-001 |
| 2 | 002 | A-002 |

1. Secondary key – This field is a field or combination of columns that are not unique but, help retrieving a set of data

Example – In the following table zip code is considered the secondary key as it can be used to retrieve a group of member records that belong to that area.

|  |  |  |  |
| --- | --- | --- | --- |
| Sequence number | Zip code | Member ID | Name |
| 1 | 51239 | 001 | MemberA |
| 2 | 32458 | 002 | MemberB |

1. Foreign key – This is a column in a table with a different primary key and is a primary key itself in a table of its own. This helps in joining two tables together

Example – The following tables have personID as the foreign key in the first table and the primary key in the second table

|  |  |  |
| --- | --- | --- |
| ID | MemberID | PersonID |
| 001 | M-001 | P-001 |
| 002 | M-002 | P-001 |

|  |  |
| --- | --- |
| PersonID | Name |
| 001 | MemberA |
| 002 | MemberB |

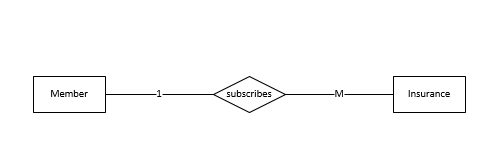
1. Combination key – Combination key is primary key with multiple fields used for identification.

Example - A transaction for a sale can have the code of the transaction and the timestamp it was received as the primary key to help identity any transaction uniquely

|  |  |  |
| --- | --- | --- |
| Transaction Code | Transaction Time | Name |
| CVV | 04-26-2019 | office |
| DTT | 04-12-2019 | district |
| CVV | 04-12-2019 | office |

# Entity-Relationship Diagram –

The logical relationship between entities are represented using an entity-relationship diagram. It is a blue print between the overall system and its entities. There are three types of entity relationship. Entities are usually represented by a rectangle and the relationship is represented by a diamond.



* 1. One to one – One entity instance will have only one instance of the second entity
  2. One to many – One instance will have many instances of the second entity
  3. Many to many – One instance of the first entity can have many instances of the second entity and an instance of the second entity can have multiple instances of the first entity. The entities in many to many are linked by another entity called associative entity. Example a member can enroll in multiple insurances and insurance can have multiple members. The two entities can be linked by an associative entity called registration
  4. 