# DBA 372: Database Management Systems

## Assignment 1

1. **Explain the difference between logical and physical data independence**.

Logical data independence means that users are shielded from changes in the logical structure of the data, while physical data independence insulates users from changes in the physical storage of the data.

1. **Which of the following plays an important role in *representing* information about the real world in a database? Explain briefly.**

**1. The data definition language.**

**2. The data manipulation language.**

**3. The data model.**

Let us discuss the choices in turn.

The data definition language is important in representing information because it is used to describe external and logical schemas.

The data manipulation language is used to access and update data; it is not important for representing the data. (Of course, the data manipulation language must be aware of how data is represented, and reflects this in the constructs that it supports.)

The data model is fundamental to representing information. The data model determines what data representation mechanisms are supported by the DBMS. The data definition language is just the specific set of language constructs available to describe an actual application’s data in terms of the *data model*.

1. **Explain the following terms briefly: *attribute, domain, entity, relationship,***

***entity set, relationship set, one-to-many relationship, many-to-many relationship*.**

Term explanations:

*Attribute* - a property or description of an entity. A toy department employee entity could have attributes describing the employee’s name, salary, and years of service.

*Domain* - a set of possible values for an attribute.

*Entity* - an object in the real world that is distinguishable from other objects such as the green dragon toy.

*Relationship* - an association among two or more entities.

*Entity set* - a collection of similar entities such as all of the toys in the toy department.

*Relationship set* - a collection of similar relationships

*One-to-many relationship* - a key constraint that indicates that one entity can be associated with many of another entity. An example of a one-to-many relationship is when an employee can work for only one department, and a department can have many employees.

*Many-to-many relationship* - a key constraint that indicates that many of one entity can be associated with many of another entity. An example of a many to- many relationship is employees and their hobbies: a person can have many different hobbies, and many people can have the same hobby.

1. **Define the following terms: *relation schema, relational database schema, domain, attribute, attribute domain, relation instance, relation cardinality*, and *relation degree*.**

A *relation schema* can be thought of as the basic information describing a table or *relation*. This includes a set of column names, the data types associated with each column, and the name associated with the entire table. For example, a relation schema for the relation called Students could be expressed using the following representation:

Students(*sid:* string, *name:* string, *login:* string,

*age:* integer, *gpa:* real) There are five fields or columns, with names and types as shown above.

A *relational database schema* is a collection of relation schemas, describing one or more

relations.

*Domain* is synonymous with *data type*. *Attributes* can be thought of as columns in a table. Therefore, an *attribute domain* refers to the data type associated with a column.

A *relation instance* is a set of tuples (also known as *rows* or *records*) that each conform to the schema of the relation.

The *relation cardinality* is the number of tuples in the relation.

The *relation degree* is the number of fields (or columns) in the relation.

1. **What is a foreign key constraint? Why are such constraints important? What is referential integrity?**

Lecture 2 slides 21-25