Week 1: Introduction, Review of Database Concepts

- Database Management Systems – DBMS
  - What is it?
    - Collection of programs that manage the data and controls access.
    - Intermediary between the user/program/system (mostly referenced as Application) and the data
  - Benefits
    - Improved data sharing. More users have access to more & well managed data.
    - Improved data security. Users can have controlled access to data.
    - Minimized data inconsistency. Due to enforced rules on how to store the data.
    - Improved data access. Data can be related together to answer queries.
    - Improved decision making. Due all of the above.
    - Increased end-user productivity. Less time to get tasks done.
  - Different types of databases:
    - Relational – RDBMS. Data structured in tables, columns, and rows.
      - Deviations from/interpretations of standards by vendors.
    - Hierarchical
    - Network
    - Object Oriented
- Keys
  - Key - A column or group of columns used to identify/retrieve records from a table
    - Not necessarily to identify entire row, e.g. can use state GPA as a key to select Rank from a table of many columns.
  - Compound/composite key - a key consisting of more than one column.
    - In some literatures, a compound key may be a key consisting of sub-keys
  - Unique / Non unique key
    - Uniquely identifies certain attributes on the row but not entire row, using it to query returns multiple records, e.g. (Last Name, Phone Number) would return every member of the household.
  - Superkey – a key that identifies uniquely the entire row in a table
  - Candidate key – A minimal superkey, i.e. using the least number of columns
  - Primary Key – a super/candidate key that is designated as the Primary Key
  - Secondary/Alternative Key – may be a non-unique key, e.g. (Last Name & Phone)
  - Foreign Key – a candidate key of another table, used for joining tables
  - Surrogate key. A column added by the designer to a table to uniquely identify each row, has no added value semantically to the data in the table, e.g. Customer Number, Column ID / Table ID on metadata tables.
- Entity Relationship (ER) Modeling
  - ERD - Entity, Attribute, Primary Key, Relationship, Cardinality/Multiplicity
  - Entity – Person, place, thing = Table

        - Entity Instance = Row
    - Attribute – Various characteristics of an Entity = Column
    - Primary Key- Attribute that uniquely identify an Entity Instance = row
    - Relationships (Associations) –How Entities relate to each other
        - 1 to 1
        - 1 to Many
        - Many to Many
    - Cardinality/ Multiplicity – Count of instances allowed/necessary between entity relationships, minimum needed & maximum allowed.
    - ERD notations – Crow's foot, UML
- Normalization
    - Benefit:
        - Avoid data integrity issues (anomalies) caused by multiple copies of same data.
        - Reduce space usage resulting from duplicated data
        - Provides for less complex update process of the data, one copy to update.
        - Speeds up updates to the data (allows better concurrency)
    - Various levels: 1NF…5NF
        - 1NF
            1. No repeating groups, e.g. CSIT1810,CSIT2320,CSIT2510 in one field; Repeating columns on a table, e.g. class1, class2, class3.
            2. Every row should be unique, therefore you must identify the primary key.
        - 2NF
            - Must be 1NF
            - Attributes must be dependent on the entire key. Applicable only if the primary key is multi-column. For example, a table with a key consisting of (BookID, GenreID) is not in 2NF if it has a column such as GenreName because GenreName depends on GenreID only.
        - 3NF
            - Must be 2NF
            - Attributes must be dependent solely on the key. Typically you will find attributes that are dependent on other attributes which are dependent on the key. For example, a table with an EmployeeID as a primary key is not in 3NF if it contains the columns DeptID, DeptName because DeptName is dependendt on DeptId and DeptID is dependen on EmployeeID.
- SQL- Structured Query Language
    - Variations across different vendors, e.g. handling of date & time, PL-SQL, T-SQL.
- Terminology
    - Schema, Table, View, Attribute/Column, Row/Tuple, Normalization, Modeling (External/Conceptual/Internal/Physical, Trigger, Stored Procedure, constraints, SQL, PL-

SQL, T-SQL, DDL, DML, Metadata, single user v multi-user, concurrency, Transaction, OLTP/Operational database, Data Warehouse/Analytical database, structured/unstructured data, ANSI