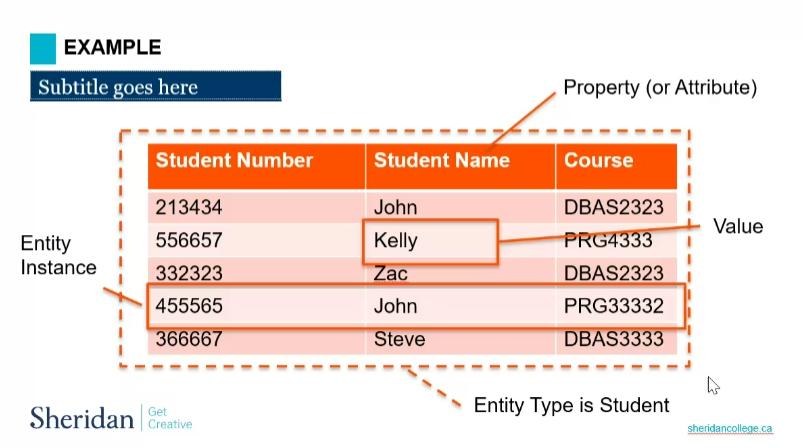
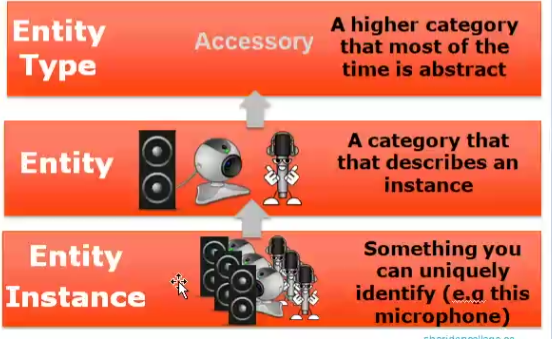
**Databases Week 1 Notes**

* What is Data
  + Facts and figures, which can convey information
  + Data with some basis
  + Data validation
  + Data can be structured or non structured
  + Can come in different formats from different sources
  + Data refers to factual information and single data element is a single fact
* What is Database?
  + Data with some basis
  + Organizes, adds/removes, secures, and validates data
  + Allows data in
  + Structures and relates logically related data or objects
  + Allows information out
  + If data cannot fit in predefined data type of database, it is unstructured
  + Data has to be processed in a way that increases knowledge to become information
* Advantages
  + Program-data independence
  + Planned Data redundancy
  + Improved Data consistency
    - If transferring funds between 2 accounts, the total amounts before and after transfer should be equal. IF this can be guaranteed in all transfers, it is consistent
  + Improved Data sharing
  + Increased Application Development productivity
  + Enforcement of Standards
  + Improved data quality
  + Improved data accessibility and responsiveness
  + Reduced program maintenance
  + Improved decision support
* Metadata
  + Data about data
  + Metadata describes the properties of the data and hence provides the information based on which it can be organized and retrieved
  + Data that provides context for data
* Evolution
  + Databases started to emerge in a file processing dominating environment
  + Database systems were mostly used for large projects like the Apollo Moon Landing
  + In late 1960’s, Database Task Group was established as first effort for database standardization
  + The commercial spread of relational data model
  + Relational models store data in tables and use SQL to manipulate data
  + Object oriented databases were introduced to accommodate the storage of non-structured data types
  + Relational databases are most widely used
  + Object-relational databases to accommodate both structured and unstructured data
  + Emergence of NoSQL (Not only SQL) databases to efficiently address the storage and retrieval of big data
* Range of Database apps
  + PErsonal databases
* Development stages
  + Conceptual model
    - It is a model, nothing but a diagram made at the conceptual level, what the database will look like, and identify the entities.
    - Define detailed inventory of data attributes
    - Lists all the data categories
    - Establishes relations between data categories
    - Results in conceptual schema
  + Logical Model
    - After conceptual, we now start thinking about the technology we will use for the database. How will we write different queries? What is the data, the structure of data going into the database?
    - Transform conceptual schema to logical schema
    - Describe data in terms of data management technology that will be used
    - Normalize the data
    - translating the conceptual data to the specific model we r using
  + Physical Model
    - Focuses on the memory, how the data is physically stored
    - Hardware procedures
    - Describe how data will be stored and managed by the technology
    - Design index for database
* Metadata stored in Repository
* Conceptual Data Modeling
  + Process of abstracting the reality, as our business sees it, to well defined and concrete entities
  + Capture the nature and relation among the data
  + Do I need to store that data? Is it relevant?
  + Identify entities, their description and behaviour, and their relation with other entities
* Business rules
  + The basis for data model
  + Business rules define what data your business needs and how to govern the data
  + Each org or business has its own business rules
  + Requirements (like OneCard at sheridan)
* ENtity relational (E-R) Model
  + Identify entities
  + Properties of entities and relations between entities
  + Entity is something you can describe with attributes (for a person, name, age, race)
  + Relationships are not common attributes, they are connections, such as a person owning a laptop, a person can use the laptop
  + Entity relational Diagram is a method of presenting the E-R Model with graphics
  + Different notations used by different groups
  + Entities are represented in a rectangular box
  + Relationships are lines connecting boxes
  + Naming logically
* Entities
  + Entity is the unit of things in your business that you want to describe
  + Defining entities with attributes
* Entity Type
  + Higher Category that has all the entities that share similar attributes
  + Abstraction of entities to something that is meaningful and usable to your business
  + Entity type is a singular name
  + When we draw diagram we don't think about entity type, only in terms of entity, attributes, and relationships
* Entity Instance
  + Is a single occurrence of the entity class, and it is the single unit that we want to store data for
  + Object of a class
  + Entity instance ,ust have a unique value that identifies it from all other instances ( if 2 cars have the same make, color, age, they have a unique value to separate them, could be serial number or license plates)
  + If the class is Students in class, the instances would be each different student
  + Entity becomes table inside database, every row of table is instance
* Guidelines to recognizing entities
* A true data entity will have many possible instances
* Each of the instances will have something different that will differentiate it from other instances
* Each instance will have at least one descriptive attribute
* IF you have two entity types in your model that have the exact same characteristics, then they are most likely entities that should be grouped in an entity types
* Strong entity
  + Does not depend on any other entity for its existence
  + All its instances exist independently of other instances from other entity types
* Weak Entities
  + Entity instances depend on instances from other entity types
  + Like onecard, onecard cannot work unless its assigned to a student (instance of another type). Onecard only exists if student uses it



* Attributes
  + Characteristics that describes entity type
  + Each entity instcance within an entity type will have the same attributes, which are attributes of its entity type
  + Attribute is a holder to which we associate values
  + Attribute is a noun name
  + Name starts with a capital
  + Name should be unique for an entity type
  + Attributes should follow a certain format
    - Attributes notation in ERD
    - Add attributes inside their entity type rectangle
  + Required Attributes
    - Must have attributes
    - Every instance of the entity must have a value for these attributes
    - Example: Student #
    - Will be in bold in ERD
  + Optional Attributes
    - May not have to be associated data for some of the entity instances
  + Simple (atomic) attribute
    - Can not be broken down to smaller meaningful components
    - Student #, email
  + Composite Attributes
    - Can be broken down to smaller meaningful components
    - Address, date, name
    - Components named under composite attributes in parenthesis
  + Single Value Attributes
    - Has only one value (data) associated to the attribute
  + Multivalued
    - Can have more than one value associated with he attribute
    - Put in parentheses in ERD
  + Stored Attribute
    - Is the origin of the data. Can not recreate its associated data
  + Derived Attribute
    - Can be recreated from other attribute values
    - Square brackets in ERD
  + Identifier attribute
    - Attribute or combination of them whose value will uniquely identify each entity instance within entity type
    - To identify instances from one another
    - They are underlined in ERD
  + Relationships
    - Represent the interaction between entities
    - Commonly called “Association Relationships”
    - Relationships are representations of the business rules of the org
    - The relation between teacher and student in the classroom is “Teaches”
  + Relationship Type
    - General definition of relationship between entity types. Defines the possible interactions between the instances
  + Relationship Instance
    - IS the actual relation between the entity instances of 2 entity types. There is always only one relationship instance between 2 entity instances
  + Degree of relationship
    - Number of entities participating in that relationship
    - Unary (degree 1), binary (degree 2), Ternary (degree 3) are the most common degrees
  + Binary relationship
    - Relationship between instances of 2 entity types
  + Ternary
    - Relationship among the instances of 3 entity types. Something that happens when the instance of 3 entity types join together
  + Unary
    - Recursive relationship, is the relation between entity instances of the same entity type
    - Manager manages employee, but manager is also an employee, meaning it’s a relationship between 2 instances of the same entity type
  + Naming relationships
    - Is a verb phrase
    - Avoid vague names such as has or is related
  + Relationship Representaiton in ERD
    - Single line between strong entity types that it associates
    - Double line between weak entity types it associates
    - Relation name typed on top of line
  + Cardinality Constraints
    - Specifies number of instances of an entity type that may or must be associated to a single instance from another entity type
    - Cardinality is defined by minimum number of instances that may or must relate (Minimum Cardinality) and the maximum instances that may or must relate (Max Cardinality)
    - Onecard can belong to 1 student, student can have 1 onecard. 1:1 relationship
    - Customer can place many orders, 1 order will be placed by 1 customer 1:M (one to many)