DATA MODELS

DATABASE MANAGEMENT SYSTEM

Sujan Tamrakar

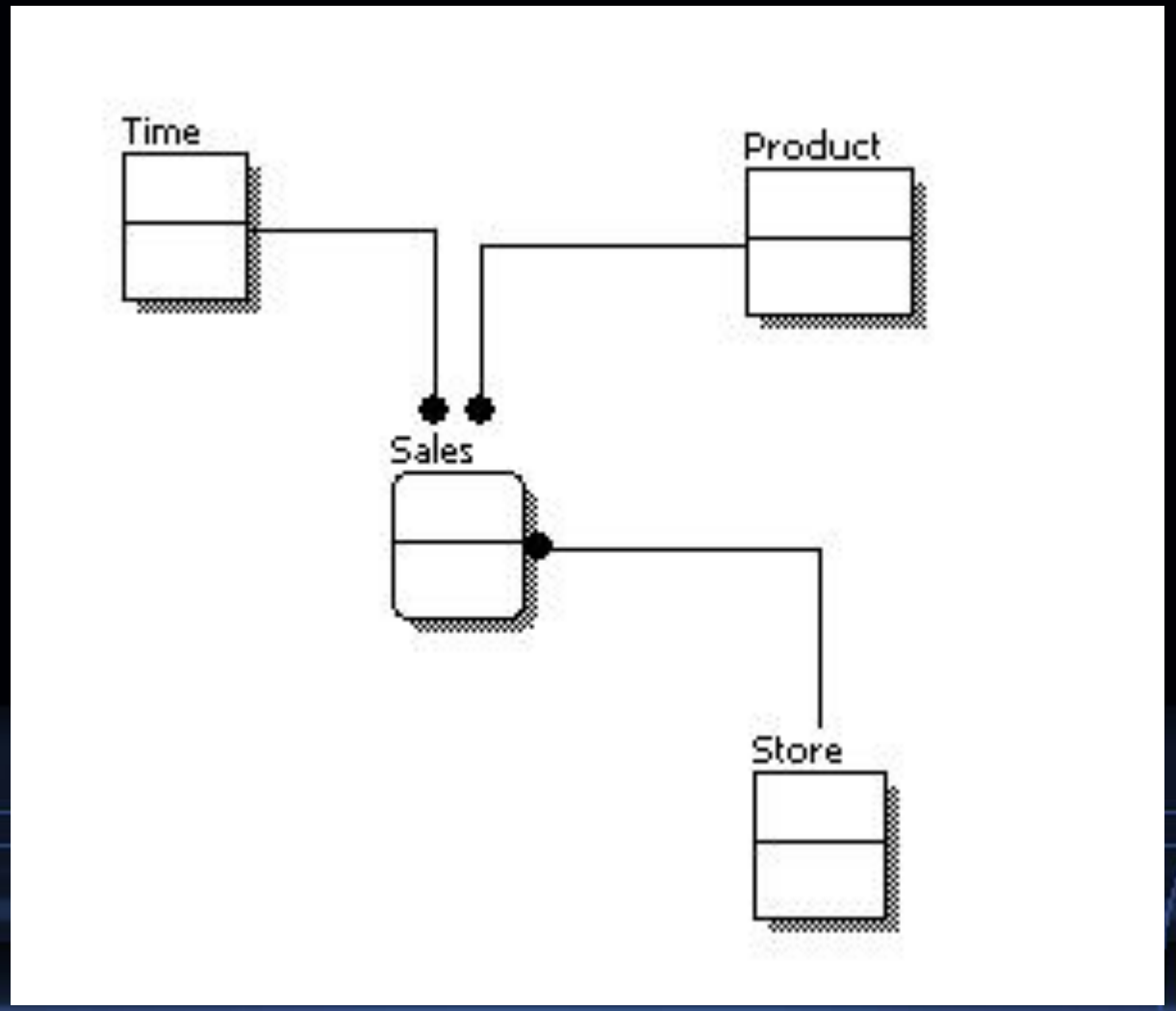
Background

- Data model: collection of concepts that can be used to describe the structure of a database (data types, relationships, constraints of data)
- Provides necessary means to achieve this abstraction
- Describes basic operations for specifying retrievals & updates on db
- Various concepts has been used to describe the structure of db
  1. Conceptual (high level) data model
  2. Logical (representational / implementational) data model
  3. Physical (low level) data model

Conceptual (high level) data model

- Identifies highest level relationships between different entities
- Features of conceptual data model include:
  - Important entities & relationships among them
  - No attribute is specified
  - No primary key is defined
  - Only information shown via conceptual model is entities that describe data & relationships between those entities. No other information is shown through conceptual model

Conceptual (high level) data model



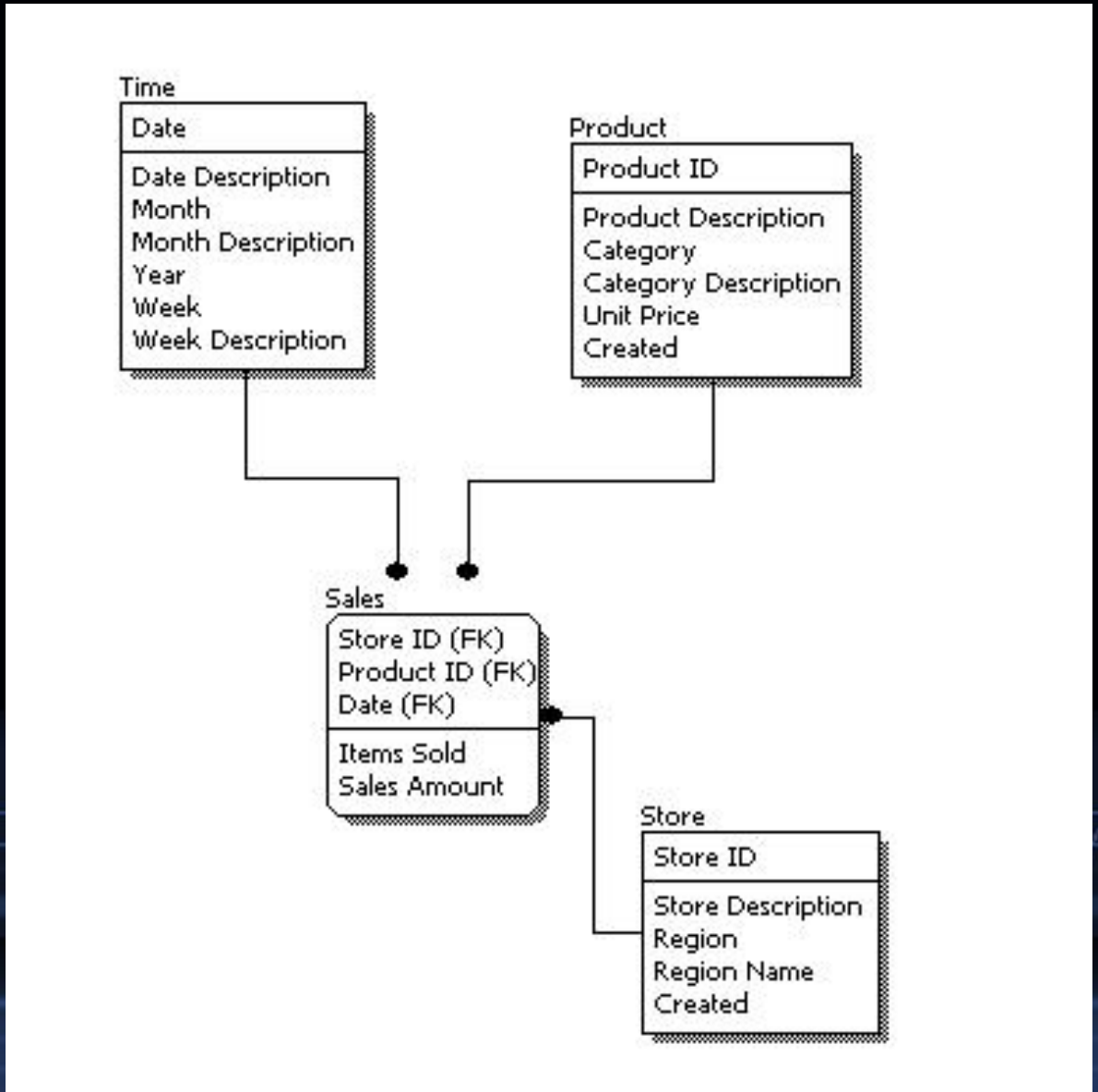
Logical (representational / implementational) data model

- Describes the data in as much detail as possible without regard to how they will be physically implemented in db
- Features of logical data model include:
  - All entities & relationships among them
  - All attributes for each entity are specified
  - Primary key for each entity is specified
  - Foreign keys are specified
  - Normalization occurs at this level

Logical (representational / implementational) data model

- Steps for designing:
  - Find all attributes for each entity
  - Specify primary keys for all entities
  - Find relationships between different entities
  - Normalization procedure
  - Resolve relationships (one-many, many-many, one-one)

Logical (representational / implementational) data model



Physical (low-level) data model

- Describes the details of how data are stored in the computer
- Concepts provided by low level data model are generally for computer specialists not for typical users
- Represents how the data model will be built in the db
- Shows all table structure including column name, data type, constraints, primary key, foreign key, relationship between tables

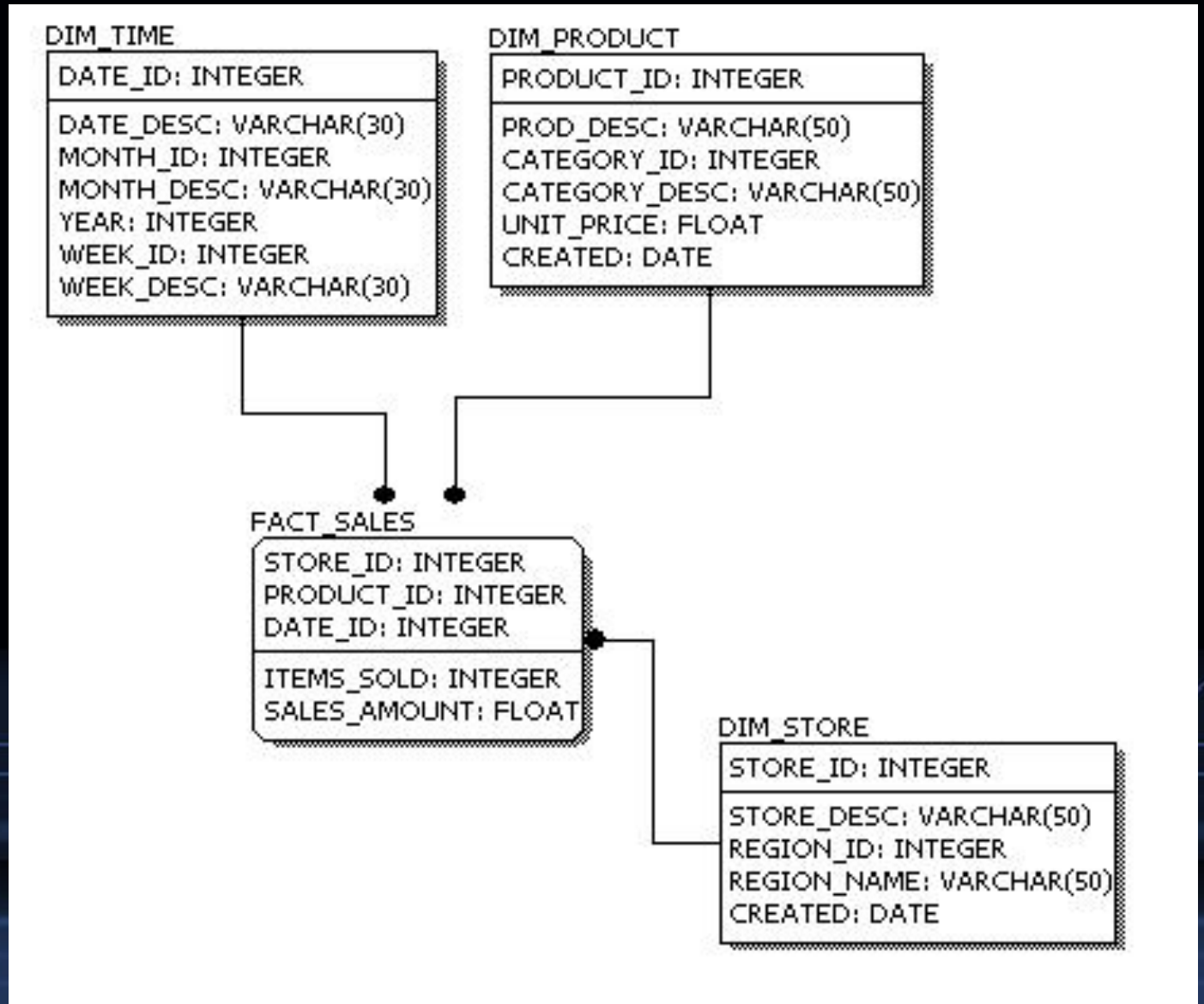
Physical (low-level) data model

- Features:
  - Specification of all tables & its columns
  - Foreign keys are used to identify relationships between tables
  - De-normalization may occur based on user requirement
  - Physical considerations may cause the physical data model to be quite different from logical data model
  - Physical data model will be different for different RDBMS.

Physical (low-level) data model

- Steps:
  - Convert entities into tables
  - Convert relationships into foreign keys
  - Convert attributes into columns
  - Modify physical data model based on physical requirement

Physical (low-level) data model



Briefing…

- Complexity increases from

  Conceptual [what are different entities in our data & how are they related]

  Logical [understand details of data without worrying about how they will be implemented]

  Physical [how to implement data in database]

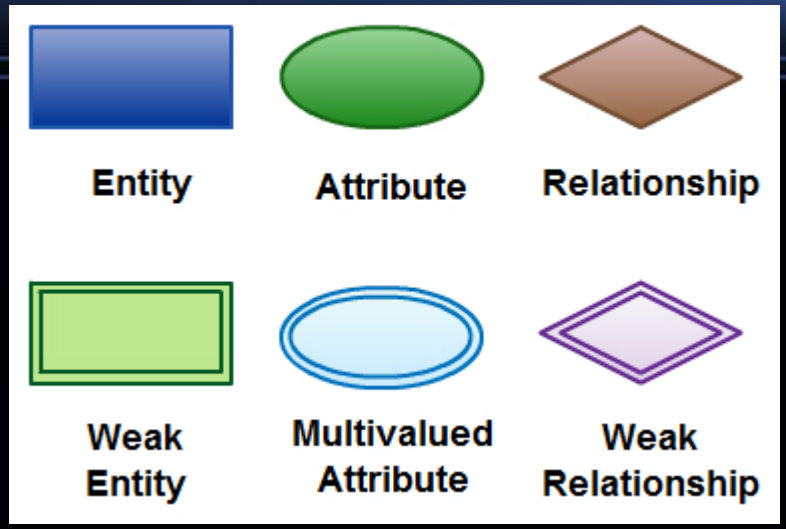
Briefing…

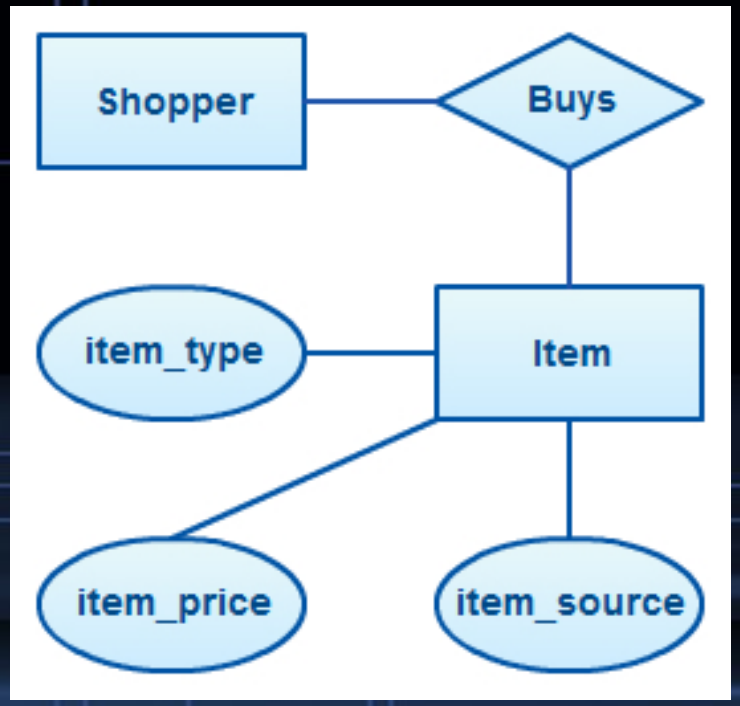
- Entities: set of attributes

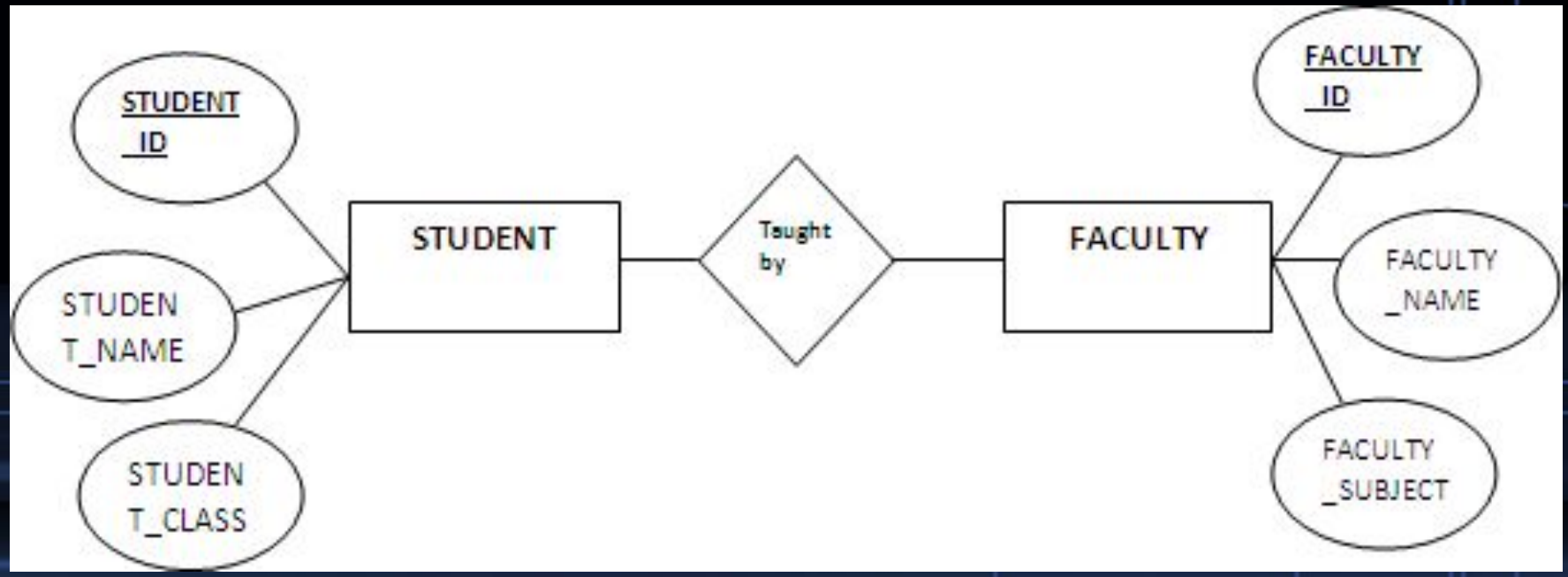
  Ex: Customer = entity

  customer_name, customer_address = attributes
- Relationship: an association among several entities
- Primary key: a candidate key that is chosen by db designer as principal means of identifying tuples within a relation. It is unique in entire table.
- Foreign key: a key which specifies a referential integrity constraint between the two relational schemas.

E-R diagram


Entity
Attribute
Relationship
Weak Entity
Multivalued Attribute
Weak Relationship


Shopper
Buys
item_type
Item
item_price
item_source


STUDENT _ID
STUDEN T_NAME
STUDEN T_CLASS
STUDENT
Taught by
FACULTY
FACULTY _ID
FACULTY _NAME
FACULTY _SUBJECT

ER Model

- A widely used data model for db design
- Provides a convenient graphical representation to view data, relationship & constraints.
- Developed to facilitate db design by allowing the specification of an enterprise schema.
- Such schema represents overall logical structure of db. This overall structure can be expressed graphically by an E-R diagram.
- Useful in mapping the meanings and interactions of real world enterprises onto a conceptual schema.

ER Model

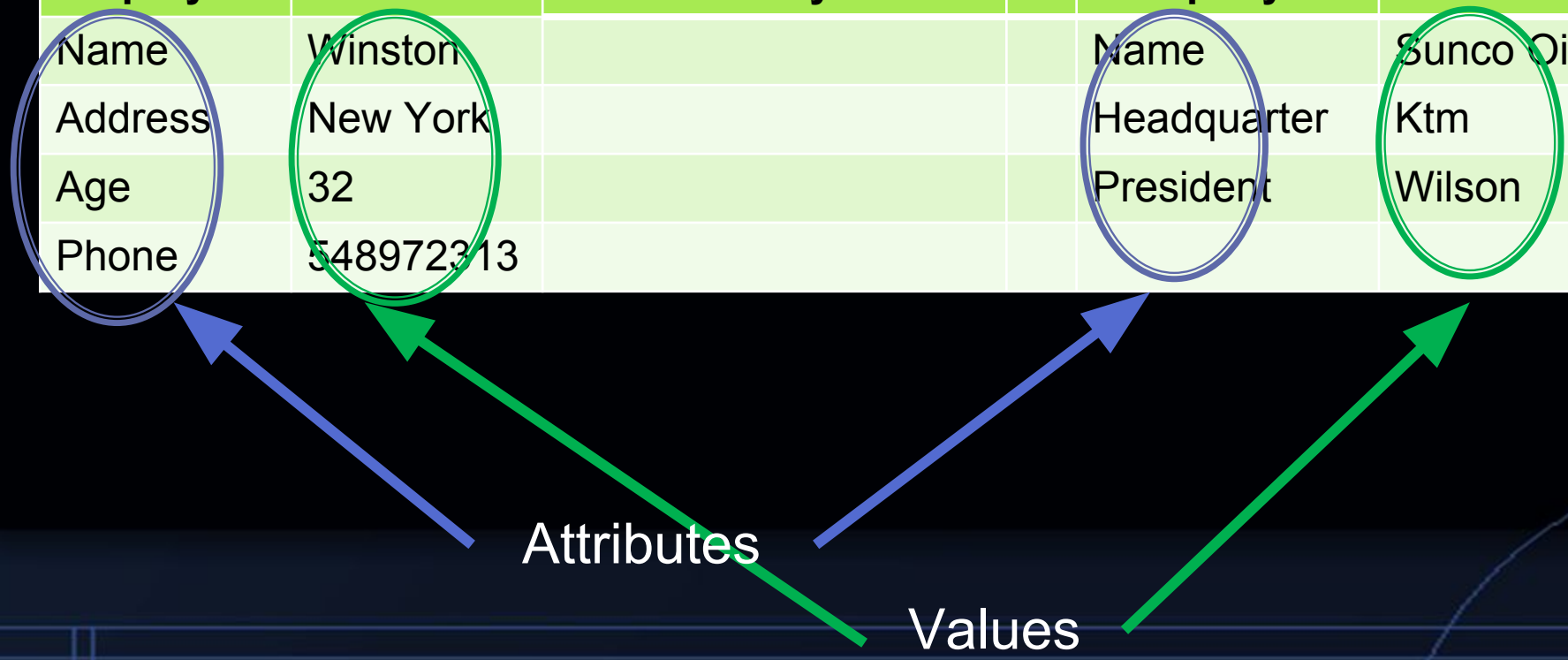
- ER model employs 3 basic notions:
  1. Entity sets
  2. Relationship sets
  3. Attributes

Entity set

- Entity is a thing in the real world with an independent existence
- Entity can have physical existence (house, person, car, book, etc.) or conceptual existence (company, account, job, loan, etc.)
- Entity is represented by a set of attributes
- Each entity has a value for each of its attributes
- Collection of entities of a particular entity type in a db at any point of time is called Entity Set.

Entity set

Employee		Entity		Company	
Name	Winston			Name	Sunco Oil
Address	New York			Headquarter	Ktm
Age	32			President	Wilson
Phone	548972313				



Relationship set

- Relation is an association among several entities
- Relationship set is a collection of relationships of same type
- Relationship instance represents an association between the named entities in the real world enterprise that is being modelled.
- Function that an entity plays in a relationship is called that entity's role. It is necessary when meaning of relationship needs clarification.
- Relationship instance in a given relationship set must be uniquely identifiable from its participating entities without using the descriptive attributes.

Relationship set

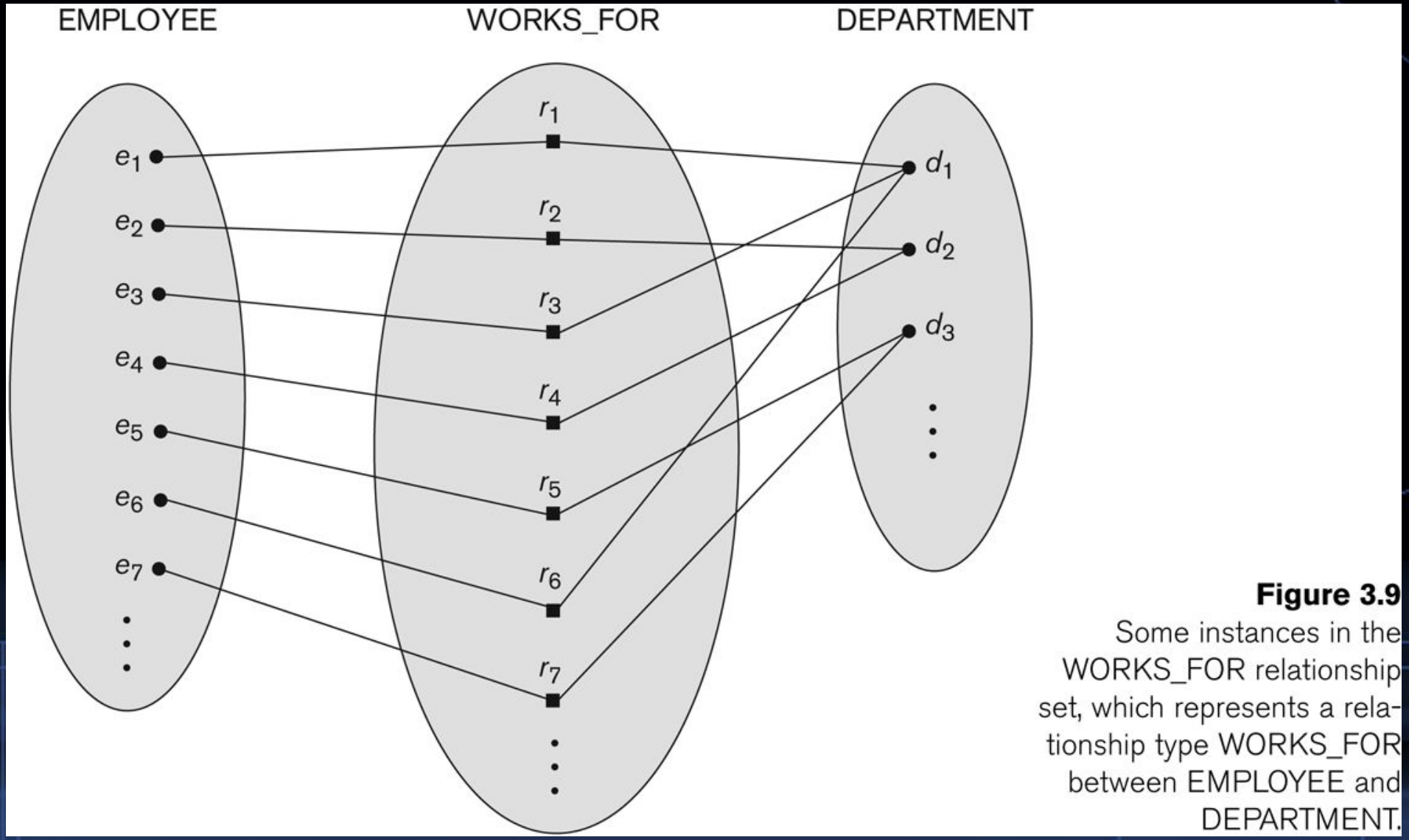


Figure 3.9
Some instances in the WORKS_FOR relationship set, which represents a relationship type WORKS_FOR between EMPLOYEE and DEPARTMENT.

Attributes

- Descriptive properties possessed by each member of an entity set
- Each entity may have its own value for each attribute
- Ex: ‘customer’ entity = customer_id
  customer_name
  customer_street
  customer_city

  } Attributes
- Ex: ‘loan’ entity = loan_no
  amount

  } Attributes

Attributes types:

- Simple and Composite attributes
  - Simple attributes can’t be divided into subparts.
  - A.k.a. Atomic attributes
  - Ex: Regd_no, age, fname, employee_no

  - Composite attributes can be divided into subparts.
  - Helps to group together related attributes
  - Ex: Name = fname, mname, lname
  - Ex: Address = a_street, a_city, a_state, a_zipcode

Attributes types:

- Single valued & Multi-valued attribute
  - Single valued attributes have a single value for a particular entity.
  - Ex: loan_no have single load id number

  - Multi-valued attributes have set of values for a specific entity.
  - Ex: Phone_no may hold more than one number, multiple email
  - Upper & lower bounds may be placed on the number of values. (minimum and maximum allowed)

Attributes types:

- Derived Attribute
  - Value for such attribute can be derived from values of other related attributes
  - Value of a derived attribute is not stored but is computed when required
  - Ex:
    - Customer: cus_id, dob, age
    - Customer: cus_id, name, loans_held
    - Loans: id, cus_id, loan_type

Thank you!