

Outline of today's session

- Database Design
 - Modelling
 - Gathering business rules
 - ERD's

Data Modelling

- Graphical, simple representation of data
- Must have:
 - Description of data structure
 - Set of enforceable rules to guarantee the integrity of data
 - Data manipulation methodology to support real world transformation

Data Modelling

- Entity – Person, place, thing (noun) that is stored. Object can be tracked in database
- Attribute – Characteristics of entity
- Relationship
 - One to Many (1:M or 1..*)
 - Many to Many (M:N or *.*)
 - One to One (1:1 or 1..1)
- Constraints

Business rules

- Brief
- Precise
- Unambiguous



Business Rule Examples

- Terms
 - A manager is a person to whom 2 or more people report directly
 - An order must have a buyer
- Facts
 - Each order can only have 1 discount
 - Orders can contain multiple line items
 - Customers must have a customer ID
- Constraints
 - Total sale must not exceed \$500
 - Alcohol can only be purchased by people over 21

Gathering business rules

Honest Bob's Discount Warehouse of 23 Dubious Lane Cheapside NT 8023 needs a computerised invoicing system. Because Honest Bob is a cheapskate they are still using a paper-based system.

The organisation has a number of customers, which are all businesses (Honest Bob's does not deal with the general public, which avoids all the problems with GST). Information kept on customers includes their ABN (CustABN), business name and address (which includes the address, suburb, state and postcode). Because Honest Bob's is keen on keeping their customers and having them return time after time, the prices there are suspiciously cheap.

When a customer purchases products they are given an invoice that lists the items purchased. Generally customers purchase a number of different items of varying quantities each time. If a customer decides that they haven't got enough of a particular item, instead of increasing the quantity for that item, Honest Bob's staff just add it again to the bottom of the invoice and give the new, updated copy to the customer.

Details on the invoice include the customer details, an invoice number, a date, an item number, its description, price, quantity ordered, subtotals for each item line and an invoice total.

Gathering business rules

Honest Bob's Discount Warehouse of 23 Dubious Lane Cheapside NT 8023 needs a computerised invoicing system. Because Honest Bob is a cheapskate they are still using a paper-based system.

The organisation has a number of **customers**, which are all businesses (Honest Bob's does not deal with the general public, which avoids all the problems with GST). Information kept on customers includes their **ABN (CustABN)**, **business name and address (which includes the address, suburb, state and postcode)**. Because Honest Bob's is keen on keeping their customers and having them return time after time, the prices there are suspiciously cheap.

When a customer purchases **products** they are given an **invoice** that lists the items purchased. Generally customers purchase a number of different items of varying quantities each time. If a customer decides that they haven't got enough of a particular item, instead of increasing the quantity for that item, Honest Bob's staff just add it again to the bottom of the invoice and give the new, updated copy to the customer.

Details on the invoice include the customer details, an **invoice number**, a **date**, an **item number**, its **description**, **price**, **quantity ordered**, **subtotals** for each item line and an invoice total.

Business Rules – Honest Bob

1. A customer can receive many invoices
2. Each invoice is for one customer only
3. A invoice can contain more than one item
4. An item can appear on many invoices
5. An item can appear on an invoice more than once

Scenarios & Business Rules

A company has a number of employees. The attributes of employee include employee_id (identifier), name, address, and birthdate. The company also has several projects. attributes of project include project_id (identifier), project_name, and start_date. Each employee may be assigned to one or more projects, or may not be assigned to a project. A project must have at least one employee assigned and may have any number of employees assigned. An employee's billing rate may vary by project, and the company wishes to record the applicable billing rate (billing_rate) for each employee when assigned to a particular project.

Scenarios & Business Rules

A company has a number of employees. The attributes of employee include employee_id (identifier), name, address, and birthdate. The company also has several projects. attributes of project include project_id (identifier), project_name, and start_date. Each employee may be assigned to one or more projects, or may not be assigned to a project. A project must have at least one employee assigned and may have any number of employees assigned. An employee's billing rate may vary by project, and the company wishes to record the applicable billing rate (billing_rate) for each employee when assigned to a particular project.

Project Business Rules

1. An employee can be assigned to many projects
2. A project can have many employees
3. An employee's billing rate can vary for each project

Scenarios & Business Rules

The firm has a number of sales offices in several states. Attributes of sales office include Office_Number (identifier) and Location.

Each sales office is assigned one or more employees. Attributes of employee include Employee_ID (identifier) and Employee_Name. An employee must be assigned to only one sales office.

For each sales office, there is always one employee assigned to manage that office. An employee may manage only the sales office to which he or she is assigned.

The firm lists property for sale. Attributes of property include Property_ID (identifier) and Location. Components of Location include Address, City, State and Zip_Code.

Each unit of property must be listed with one (and only one) of the sales offices. A sales office may have any number of properties listed or may have no properties listed.

Each unit of property has one or more owners. Attributes of owners are Owner_ID (identifier) and Owner_Name. An owner may own one or more units of property. Owners own a percentage of a property.

Scenarios & Business Rules

The firm has a number of sales offices in several states. Attributes of sales office include Office_Number (identifier) and Location.

Each sales office is assigned one or more employees. Attributes of employee include Employee_ID (identifier) and Employee_Name. An employee must be assigned to only one sales office.

For each sales office, there is always one employee assigned to manage that office. An employee may manage only the sales office to which he or she is assigned.

The firm lists property for sale. Attributes of property include Property_ID (identifier) and Location. Components of Location include Address, City, State and Zip_Code.

Each unit of property must be listed with one (and only one) of the sales offices. A sales office may have any number of properties listed or may have no properties listed.

Each unit of property has one or more owners. Attributes of owners are Owner_ID (identifier) and Owner_Name. An owner may own one or more units of property. Owners own a percentage of a property.

Real Estate Rules

1. A sales office has many employees
2. An employee is assigned to one sales office
3. A sales office has one employee who is a manager
4. A manager must be an employee assigned to a sales office
5. A sales office has many properties listed
6. A property is only listed with one sales office
7. A property can have many owners
8. An owner can own many properties
9. Each owner owns a percentage of a property

Entity-Relationship Diagrams

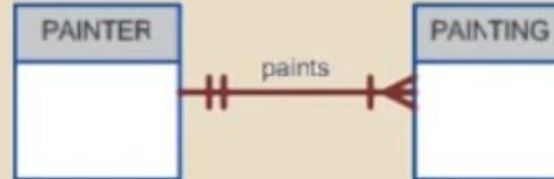
- E-R model conceptually describes relationships between entities
 - Eventually (maybe)- Entities evolve through relations into tables
- Model is represented using ERDs
 - Enables consultation with clients as to accuracy of design

Chen Notation

Crow's Foot Notation

UML Class Diagram Notation

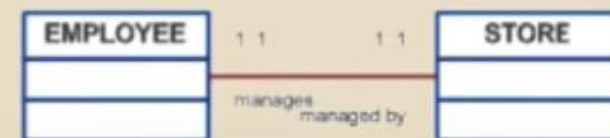
A One-to-Many (1:M) Relationship: a PAINTER can paint many PAINTINGs; each PAINTING is painted by one PAINTER.



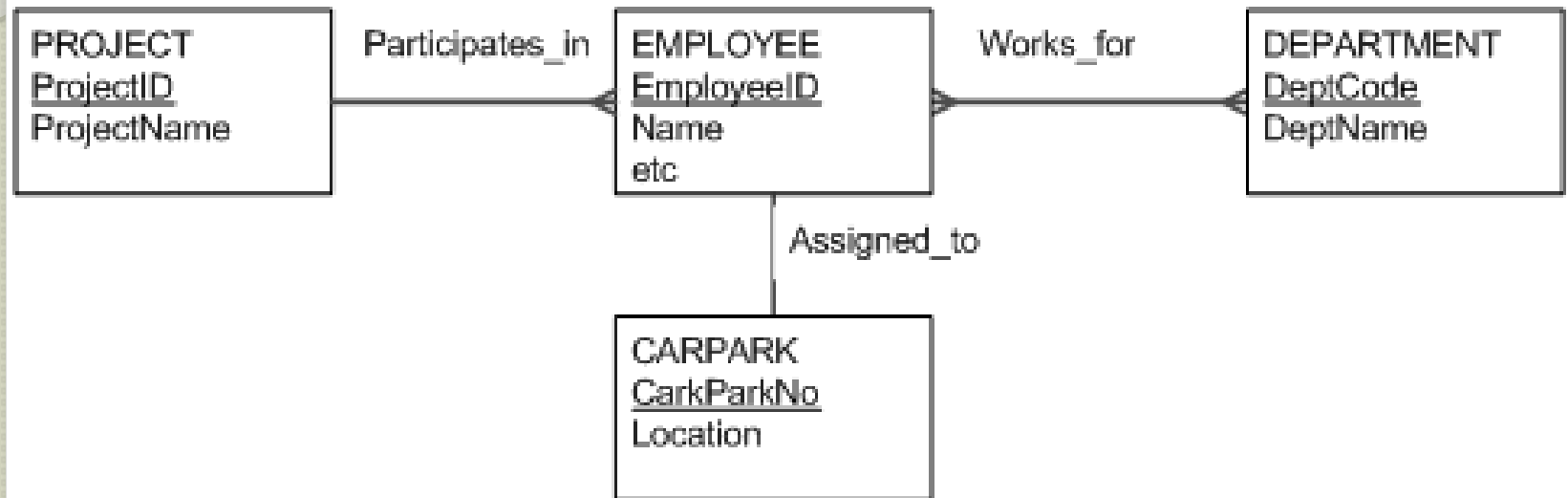
A Many-to-Many (M:N) Relationship: an EMPLOYEE can learn many SKILLs; each SKILL can be learned by many EMPLOYEEs.



A One-to-One (1:1) Relationship: an EMPLOYEE manages one STORE; each STORE is managed by one EMPLOYEE.



A Simple ERD



E-R constructs - entities

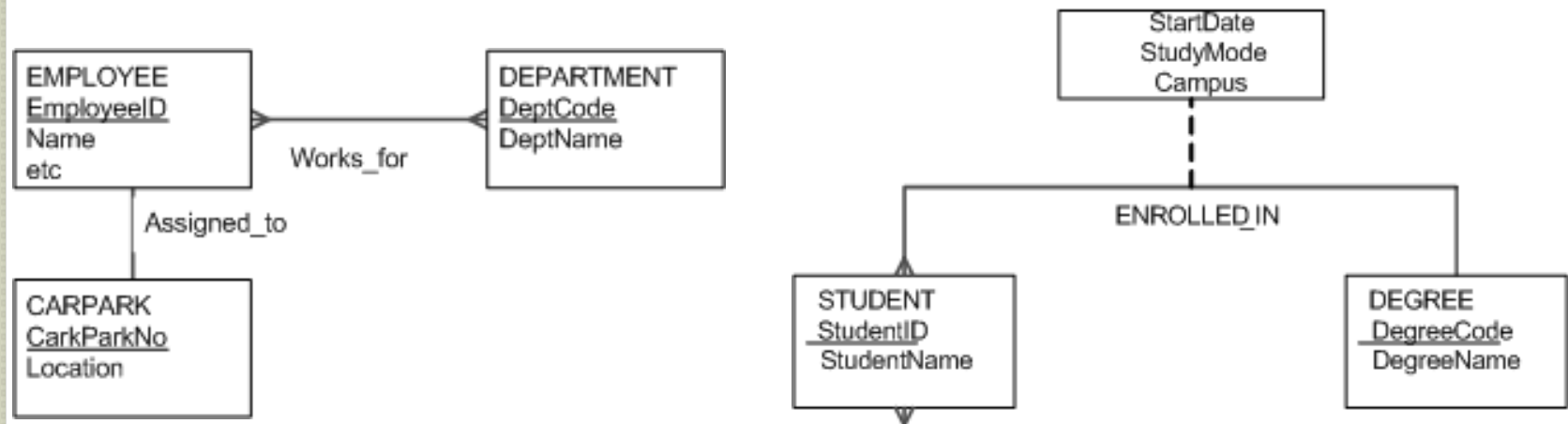
- Naming entities
 - Singular
 - Noun
 - In upper-case

PROJECT <u>ProjectID</u> ProjectName

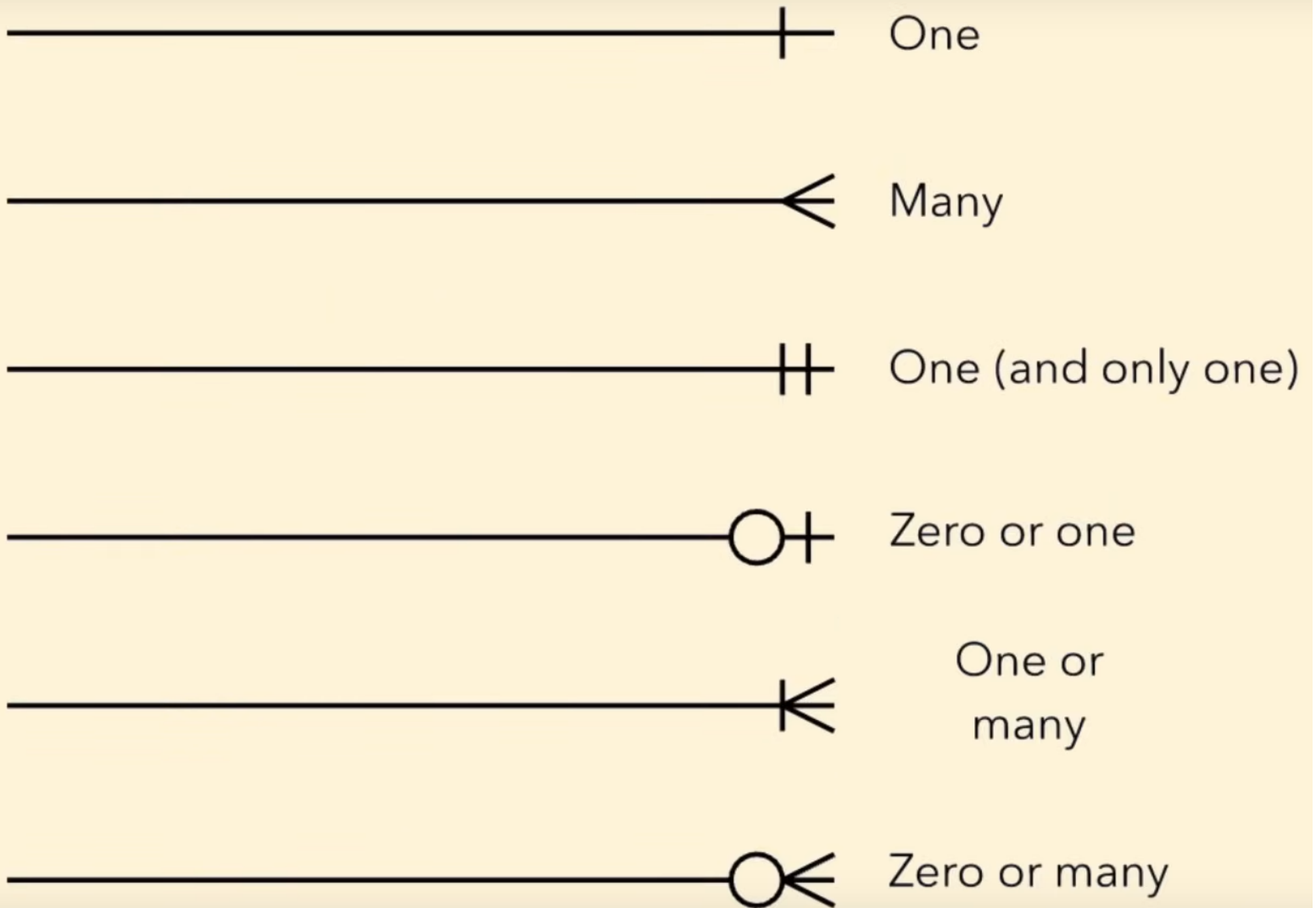
CARPARK <u>CarkParkNo</u> Location

E-R constructs - relationships

- Naming relationships
 - Verb-phrase
- Remember
 - Relationships can contain attributes



E-R constructs - relationships



E-R constructs - attributes

- Naming attributes
 - Noun that can have multiple words
 - Can use aliases
- Simple or atomic
- Compound (, ,)
 - Address (Street, City, State, ZipCode)
- { multivalued }
 - {Skill}
- [derived]
 - [TotalPrice]

E-R constructs - attributes

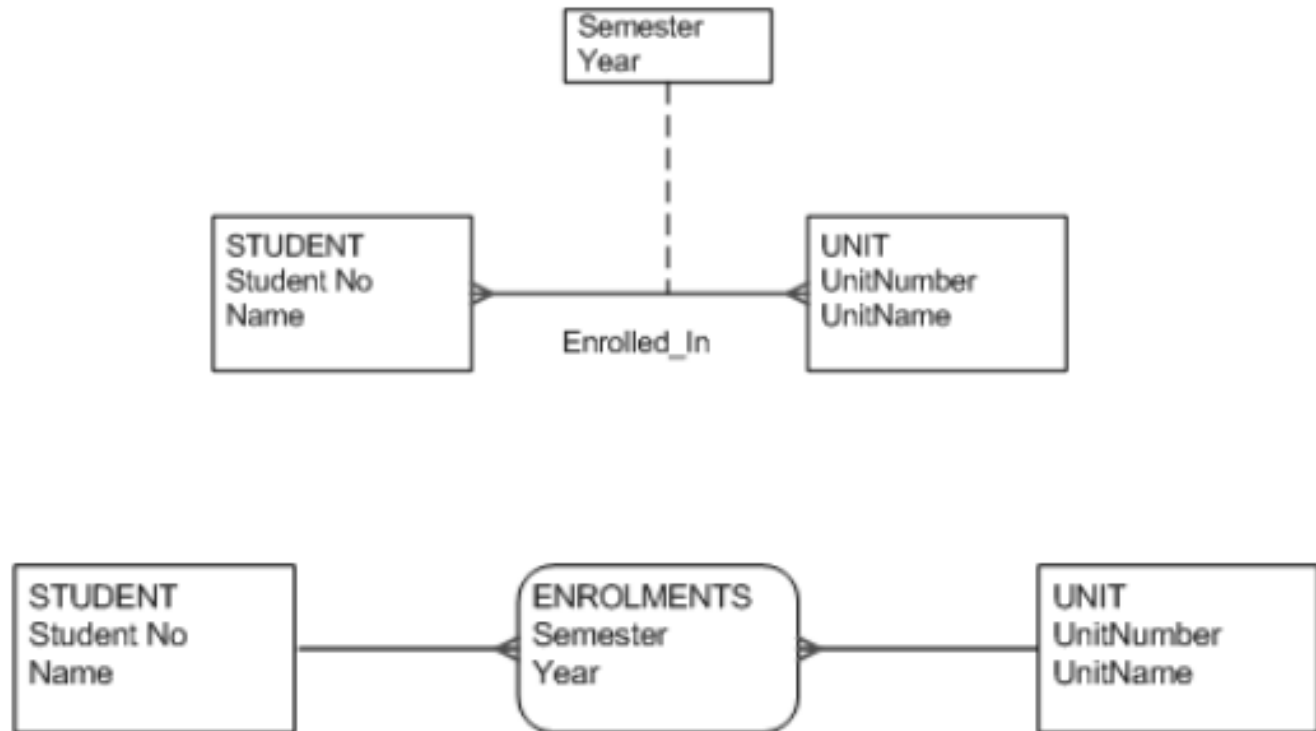
- **Unique identifier**

- Distinguishes individual instances of an entity type (aka primary key)
- May be composite
- May not always be possible to determine at this point

E-R Associative Entities

- Can use in place of Many-to-many relationships
- As a general rule convert a many to many relationship to an associative entity when
 - There are attributes on the relationship
 - The relationship has some independent meaning to the client

E-R Associative Entities



Steps to Create an ERD

Step 1: Identify *all* of the entities involved

Step 2: Determine the relationships between each entity

Step 3: Decide if any of the relationships are candidates to be associative entities

Step 4: Identify the attributes for each entity in turn

Step 5: Identify the attribute that is the *unique identifier* for each individual instance of an entity (if possible)

Step 6: Include any assumptions made (this can also contain data definitions).

Honest Bob Scenario ERDs

The entities are:

1. CUSTOMER (The organisation has a number of **customers...**)
2. INVOICE (When a customer purchases products they are given an **invoice...**)
3. ITEM (... that lists the **items** purchased)

Honest Bob Scenario ERDs

Determine the relationships between each entity

Review the business rules you identified for the scenario:

Rule 1. A customer can **receive** many invoices

Rule 2. Each invoice is for one customer only

Rule 3. An invoice can **contain** more than one item

Rule 4. An item can appear on an invoice more than once

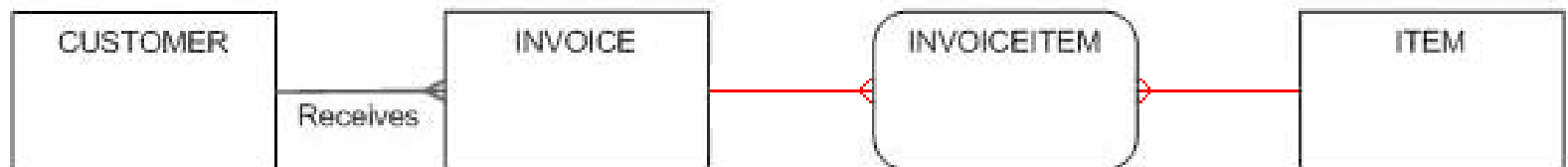
Rule 5. An item can appear on many invoices



Honest Bob Scenario ERDs

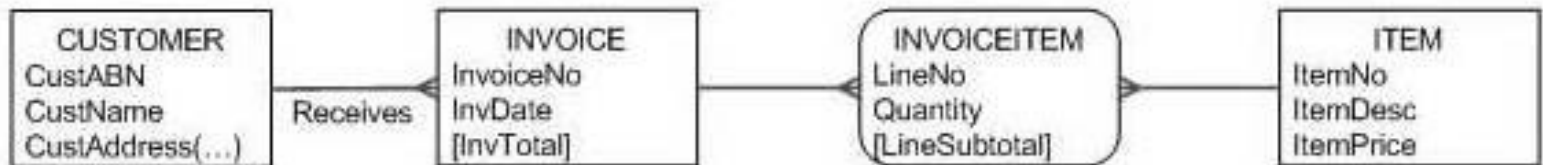
Decide if any of the relationships are candidates to be associative entities

Rule 4: An item can appear on an invoice more than once.



Honest Bob Scenario ERDs

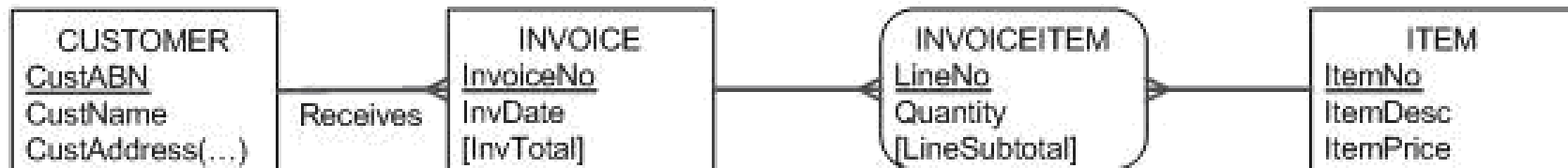
Identify the attributes for each entity



Honest Bob Scenario ERDs

Identify the primary key for each individual instance of an entity

Primary keys are represented with an underline



Honest Bob Scenario ERDs

Include any assumptions made (this can also contain data definitions)

Assumptions:

1. Business name and address have been prefixed with cust in CUSTOMER to keep the attribute names unique
2. The same applies for invoice number and date in INVOICE and item number, description and price in ITEM
3. To follow the business rules LineNo is placed in the associative entity INVOICEITEM as a primary key
4. Quantity is placed in INVOICEITEM to allow a separate quantity for each item on the invoice.
5. LineSubtotal in INVOICEITEM is calculated by: $\text{Quantity} * \text{ItemPrice}$.
6. InvTotal in INVOICE is calculated by: sum of LineSubtotal

Putting it all together

- Use the business rules from
 - Real estate firm
 - Courtsto create ERDs for these systems

Source: Unit website – database design: modelling the system - get your hands dirty – Real Estate Firm and Court Scenarios