

A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the date.

18-5-2022

Database

Design 1 / 2

Several thin, curved lines in dark blue and light grey originate from the bottom left corner and curve upwards and to the right.

JUAN CARLOS HERRERA HERNANDEZ
UNIVERSIDAD POLITECNICA DE AGUASCALIENTES

Contenido

- Section 1 – Introduction2
 - 1-1 Introduction to the Oracle Academy2
 - 1-2 Data vs. Information2
 - 1-3 History of the Database.....2
 - 1-4 Major Transformations in Computing.....3
- Section 2 – Entities and Attributes4
 - 2-2 Entities, Instances, Attributes, and Identifiers4
 - 2-3 Entity Relationship Modeling (ERM) and ERDs5
- SQL (DDL and DML).....7
- Section 3 – Relationship Basics.....8
 - 3-1 Identifying Relationships8
 - 3-2 ER Diagramming Conventions8
 - 3-3 Speaking ERDish and Drawing Relationships9
 - 3-4 Matrix Diagrams9
- Section 4 – Super/Sub Types and Business Rules..... 10
 - 4-1 Supertypes and Subtypes &&..... 10
 - 4-2 Documenting Business Rules..... 11
- Section 5 – Relationship Fundamentals 12
 - 5-1 Relationship Transferability 12
 - 5-2 Relationship Types 12
 - 5-3 Resolving Many-to-Many Relationships..... 13
 - 5-4 Understanding CRUD Requirements..... 14

Section 1 – Introduction

1-1 Introduction to the Oracle Academy

The Oracle Academy Database Design and Programming with SQL Course

- Entity relationship modeling: database design, development, and normalization
- SQL (structured query language) “The language of the database”
- Accessing data with SQL
- Data definition, manipulation, and control languages
- Transaction control

Oracle Certifications:

- Foundations Associate, Database 1Z0 006: Oracle Database Foundations
- Oracle Database SQL Certified Associate 1Z0 071: Oracle Database SQL

1-2 Data vs. Information

Database: A collection of data arranged for ease and speed of search and retrieval.

Data: Raw material, from which you can draw conclusions

Information: Knowledge, intelligence, a particular piece of data with a special meaning or function

1-3 History of the Database

- 1970-72: E. F. Codd proposes the relational model for databases, disconnecting the logical organization from the physical storage.
- 1976: P. Chen proposes the entity relationship model (ERM) for database design
- Mid-1980s: SQL (structured query language) becomes "intergalactic standard"
- Mid-1990s: Kaboom! The usable Internet/World Wide Web (WWW) appears
- 1980s: As personal computers (PCs) became faster and widely available, processing moved from the mainframes to the desktop (workstations).

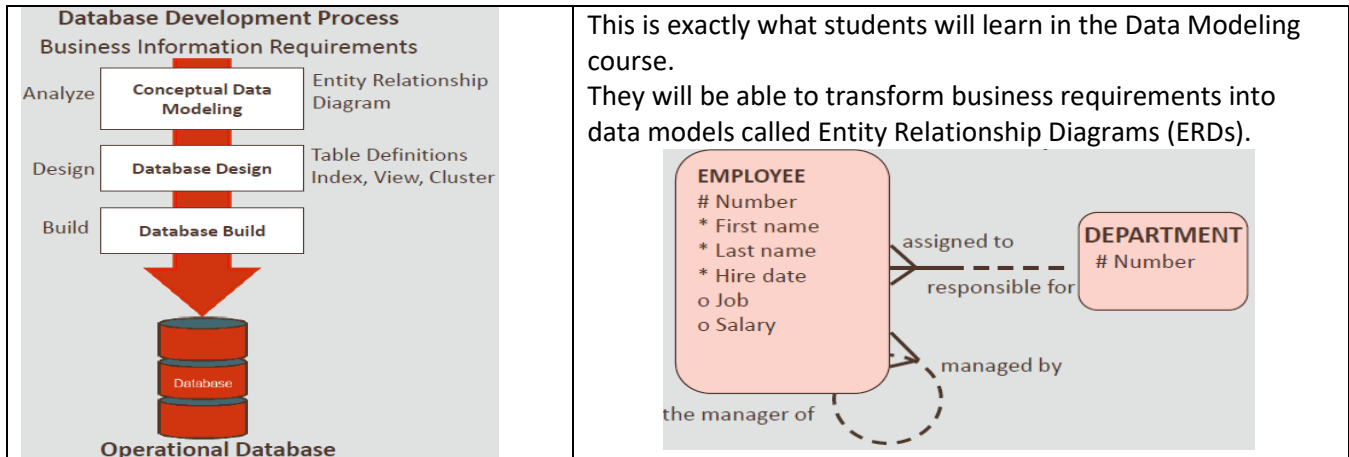


Table instance chart	A relational database chart that is used to map the information from the entity relationship diagram
Table name	Identity
Column names	Attributes
Primary key (PK)	the unique identifier for each row of data
Foreign key (FK)	Links data in one table to the data in a second table by referring to the PK column in the second table
Nulls	Indicates if a column must contain a value
Unique	Indicates if the value in the column is unique within the table
Datatype	Refers to the format and definition of the data in each column

1-4 Major Transformations in Computing

Infrastructure	The basic framework or features of a system
Grid computing	A global effort to develop an environment in which individual users can access computers, databases, and experimental facilities simply and transparently, without having to consider where those facilities are located
Software	The programs, routines, and symbolic languages that control the functioning of the hardware and direct its operation.
Hardware	A computer and the associated physical equipment directly involved in the performance of data-processing or communications functions.
Operating System	Software designed to control the hardware of a specific data-processing system in order to allow users and application programs to make use of it.
Application	A software program which carries out specific tasks on behalf of computer users
Client	A workstation or desktop computer including a screen, keyboard, and mouse; communicates directly with the user
Server	A more powerful computer which accepts work requests from clients, does the work, and sends results back to the client
Cloud Computing	Cloud computing allows businesses to access software and hardware from a cloud provider. These services are located remotely and delivered to users using web technologies.

Section 2 – Entities and Attributes

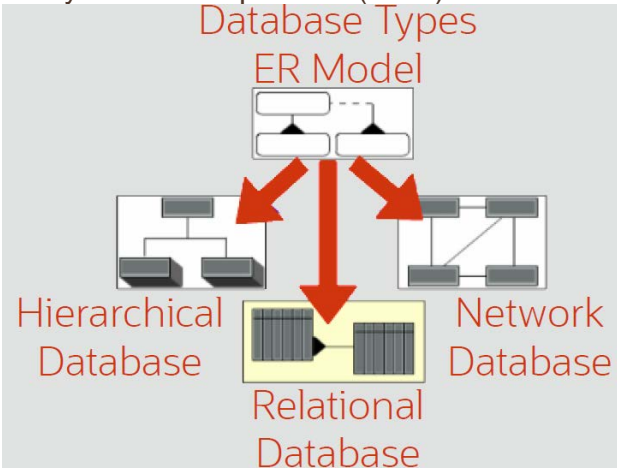
Conceptual model “client’s dream”	<p>A data model, usually represented by an entity relationship diagram (ERD). Identifies :</p> <ul style="list-style-type: none"> • important entities (objects that become tables in database) • relationships among entities <p>Does not specify :</p> <ul style="list-style-type: none"> • attributes (objects that become columns or fields in database) • unique identifiers (attribute that becomes primary key in database) <p>Documents the processes (also known as the “business rules”) of the business</p>
Logical model	<ul style="list-style-type: none"> • Includes all entities and relationships among them • Is called an entity relationship model (ERM) • Is illustrated in an ERD • Specifies all attributes and UIDs for each entity • Determines attribute optionality • Determines relationship optionality and cardinality
Physical model “physical reality”	<p>A design for an object (a car, a house, a database, etc.) which includes implementation details such as size, volume, weight, etc.).</p> <p>Shows all table structures, including columns, primary keys, and foreign keys.</p> <p>Is an extension to a logical data model:</p> <ul style="list-style-type: none"> • Defines table definitions, data types, and precision • Identifies views, indexes, and other database objects
Data	A collection of facts from which conclusions may be drawn
Data modeling	The process of capturing the important concepts and rules that shape a business and depicting them visually on a conceptual model

2-2 Entities, Instances, Attributes, and Identifiers

Mandatory	Required
Nontangible/ Intangible	Incapable of being perceived by the senses
Unique Identifier (UID)	Any combination of attributes and/or relationships that serves, in all cases, to uniquely identify an instance of an entity
Instance	An occurrence or example of an entity
Attribute	A characteristic; something that describes, quantifies, or specifies an entity
Optional	Not required
Entity	A named thing or category of things that is significant to the business and about which data must be known
Null	A value that is unavailable, unassigned, unknown, or empty; however, it is neither a zero nor a space

Volatile	Highly changeable
Tangible	Perceptible to the senses, especially the sense of touch
Datatype	A classification identifying one of various types of data, stating the possible values for that type, the operations that can be done on that type, and the way the values of that type are stored. Example: number, string, date, boolean
Single Valued	Can only have one value at any point for each instance in the entity

2-3 Entity Relationship Modeling (ERM) and ERDs

Implementation-free	Not dependent on the physical model
Entity relationship diagram (ERD)	A drawing that is used to represent a data model. Is a consistent tool can be used to represent data requirements regardless of the type of database used.
<p>Entity Relationship Model (ERM)</p>  <p>The diagram illustrates the relationship between Database Types, ER Model, and three database types: Hierarchical Database, Relational Database, and Network Database. Red arrows point from the ER Model to each of the three database types.</p>	<p>Is a list of all entities and attributes as well as all relationships between the entities that are of importance.</p> <p>Provides background information such as entity descriptions, data types, and constraints.</p> <p>Note: The model does not require a diagram, but the diagram is typically a very useful tool</p>

There are four goals of ER modeling:

- Capture all required data
- Ensure that data appears only once
- Model no data that is derivable from other data already modeled
- Locate data in a predictable, logical place

The ERD is the conversation tool between the consultant and the client, and it is also the blueprint for the DBA who will eventually build the database

```
graph TD
    EMPLOYEE[EMPLOYEE] -- "works in" --> DEPARTMENT[DEPARTMENT]
    EMPLOYEE -- "responsible for" --> DEPARTMENT
    EMPLOYEE -- "manager of" --> EMPLOYEE
    EMPLOYEE -- "assigned" --> JOB[JOB]
    DEPARTMENT -- "located in" --> LOCATION[LOCATION]
    LOCATION -- "address of" --> DEPARTMENT
    LOCATION -- "situated" --> COUNTRY[COUNTRY]
    COUNTRY -- "site of" --> LOCATION
    COUNTRY -- "part of" --> REGION[REGION]
    REGION -- "comprised of" --> COUNTRY
    EMPLOYEE -.- "managed by" --> EMPLOYEE
    EMPLOYEE -.- "assigned to" --> JOB
```

EMPLOYEE
Employee ID
* First name
* Last name
* Hire date
* Salary
o Commission

DEPARTMENT
Department ID
* Name

LOCATION
Location ID
* Street address
* Postal code
* City
* State province

JOB
Job ID
* Title
* Max salary
* Min salary

COUNTRY
Country ID
* Country name

REGION
Region ID
* Region name

Relationships:
- EMPLOYEE works in DEPARTMENT
- EMPLOYEE responsible for DEPARTMENT
- EMPLOYEE manager of EMPLOYEE
- EMPLOYEE assigned to JOB
- DEPARTMENT located in LOCATION
- LOCATION address of DEPARTMENT
- LOCATION situated in COUNTRY
- COUNTRY site of LOCATION
- COUNTRY part of REGION
- REGION comprised of COUNTRY
- EMPLOYEE managed by EMPLOYEE (self-referencing)

```

    erDiagram
        JOB_HISTORY ||--o{ JOB : "Record of"
        JOB_HISTORY ||--o{ DEPARTMENT : "Be listed in"
        JOB_HISTORY ||--o{ EMPLOYEE : "Employment record of"
        JOB ||--o{ EMPLOYEE : "Held by"
        EMPLOYEE ||--o{ JOB : "Holds"
        EMPLOYEE ||--o{ DEPARTMENT : "Be recorded on"
        DEPARTMENT ||--o{ EMPLOYEE : "Responsible for"
        DEPARTMENT ||--o{ EMPLOYEE : "Assigned to"
        DEPARTMENT ||--o{ EMPLOYEE : "Managed by"
        EMPLOYEE ||--o{ DEPARTMENT : "The Manager of"
        DEPARTMENT ||--o{ LOCATION : "Located at"
        LOCATION ||--o{ DEPARTMENT : "Location of"
        LOCATION ||--o{ COUNTRY : "Situating in"
        COUNTRY ||--o{ LOCATION : "The site of"
        COUNTRY ||--o{ REGION : "Part of"
        REGION ||--o{ COUNTRY : "Consists of"

        JOB_HISTORY {
            string START_DATE
            string END_DATE
        }
        JOB {
            string JOB_ID PK
            string JOB_TITLE
            float MIN_SALARY
            float MAX_SALARY
        }
        DEPARTMENT {
            string DEPARTMENT_ID PK
            string DEPARTMENT_NAME
        }
        EMPLOYEE {
            string EMPLOYEE_ID PK
            string FIRST_NAME
            string LAST_NAME
            string EMAIL
            string PHONE_NUMBER
            string HIRE_DATE
            float SALARY
            float COMMISSION_PCT
            string BONUS
        }
        LOCATION {
            string LOCATION_ID PK
            string STREET_ADDRESS
            string POSTAL_CODE
            string CITY
            string STATE_PROVINCE
        }
        COUNTRY {
            string COUNTRY_ID PK
            string COUNTRY_NAME
        }
        REGION {
            string REGION_ID PK
            string REGION_NAME
        }
    
```

```

    erDiagram
        JOBS ||--o{ EMPLOYEES : "works for"
        EMPLOYEES ||--o{ DEPARTMENTS : "works in"
        EMPLOYEES ||--o{ LOCATIONS : "works at"
        EMPLOYEES ||--o{ COUNTRIES : "works in"
        EMPLOYEES ||--o{ REGIONS : "works in"
        EMPLOYEES ||--o{ JOB_GRADES : "has grade"

        JOBS {
            VARCHAR2(10) JOB_ID PK
            VARCHAR2(30) JOB_TITLE
            NUMBER(3) MIN_SAL
            NUMBER(3) MAX_SAL
        }

        EMPLOYEES {
            NUMBER(6) EMPLOYEE_ID PK
            VARCHAR2(30) FIRST_NAME
            VARCHAR2(30) LAST_NAME
            VARCHAR2(30) EMAIL
            VARCHAR2(30) PHONE_NUMBER
            DATE HIRE_DATE
            NUMBER(3) SALARY
            NUMBER(3) COMMISSION_PCT
            VARCHAR2(30) MANAGER_ID
            NUMBER(6) DEPARTMENT_ID
            NUMBER(4) DIVISION
        }

        DEPARTMENTS {
            NUMBER(4) DEPARTMENT_ID PK
            VARCHAR2(30) DEPARTMENT_NAME
            NUMBER(6) MANAGER_ID
            NUMBER(4) LOCATION_ID
        }

        LOCATIONS {
            NUMBER(4) LOCATION_ID PK
            VARCHAR2(30) STREET_ADDRESS
            VARCHAR2(30) POSTAL_CODE
            VARCHAR2(30) CITY
            VARCHAR2(30) STATE_PROVINCE
            CHAR(2) COUNTRY
        }

        COUNTRIES {
            CHAR(2) COUNTRY_ID PK
            VARCHAR2(30) COUNTRY_NAME
            VARCHAR(30) REGION_ID
            NUMBER(6) POPULATION
        }

        REGIONS {
            NUMBER(2) REGION_ID PK
            VARCHAR2(30) REGION_NAME
        }

        JOB_GRADES {
            VARCHAR2(9) GRADE_LEVEL PK
            VARCHAR(30) LOWEST_SAL
            VARCHAR(30) HIGHEST_SAL
        }

        JOBS }|--o{ EMPLOYEES : "has"
        EMPLOYEES }|--o{ DEPARTMENTS : "has"
        EMPLOYEES }|--o{ LOCATIONS : "has"
        EMPLOYEES }|--o{ COUNTRIES : "has"
        EMPLOYEES }|--o{ REGIONS : "has"
        EMPLOYEES }|--o{ JOB_GRADES : "has"
  
```

The diagram illustrates the Oracle HR database schema with the following tables and their attributes:

- JOBS**
 - JOBS_ID (VARCHAR2(10), PK)
 - JOB_TITLE (VARCHAR2(30))
 - MIN_SAL (NUMBER(3))
 - MAX_SAL (NUMBER(3))
- EMPLOYEES**
 - EMPLOYEE_ID (NUMBER(6), PK)
 - FIRST_NAME (VARCHAR2(30))
 - LAST_NAME (VARCHAR2(30))
 - EMAIL (VARCHAR2(30))
 - PHONE_NUMBER (VARCHAR2(30))
 - HIRE_DATE (DATE)
 - SALARY (NUMBER(3))
 - COMMISSION_PCT (NUMBER(3))
 - MANAGER_ID (VARCHAR2(30))
 - DEPARTMENT_ID (NUMBER(6))
 - DIVISION (NUMBER(4))
- DEPARTMENTS**
 - DEPARTMENT_ID (NUMBER(4), PK)
 - DEPARTMENT_NAME (VARCHAR2(30))
 - MANAGER_ID (NUMBER(6))
 - LOCATION_ID (NUMBER(4))
- LOCATIONS**
 - LOCATION_ID (NUMBER(4), PK)
 - STREET_ADDRESS (VARCHAR2(30))
 - POSTAL_CODE (VARCHAR2(12))
 - CITY (VARCHAR2(30))
 - STATE_PROVINCE (VARCHAR2(30))
 - COUNTRY (CHAR(2))
- COUNTRIES**
 - COUNTRY_ID (CHAR(2), PK)
 - COUNTRY_NAME (VARCHAR2(30))
 - REGION_ID (NUMBER(2))
 - POPULATION (NUMBER(6))
- REGIONS**
 - REGION_ID (NUMBER(2), PK)
 - REGION_NAME (VARCHAR2(30))
- JOB_GRADES**
 - GRADE_LEVEL (VARCHAR2(9), PK)
 - LOWEST_SAL (VARCHAR(30))
 - HIGHEST_SAL (VARCHAR(30))

Relationships (Foreign Keys):

- JOBS** to **EMPLOYEES**: EMPLOYEE_ID references JOBS_ID.
- EMPLOYEES** to **DEPARTMENTS**: DEPARTMENT_ID references DEPARTMENT_ID.
- EMPLOYEES** to **LOCATIONS**: LOCATION_ID references LOCATION_ID.
- EMPLOYEES** to **COUNTRIES**: COUNTRY_ID references COUNTRY_ID.
- EMPLOYEES** to **REGIONS**: REGION_ID references REGION_ID.
- EMPLOYEES** to **JOB_GRADES**: GRADE_LEVEL references GRADE_LEVEL.
- DEPARTMENTS** to **LOCATIONS**: LOCATION_ID references LOCATION_ID.
- DEPARTMENTS** to **EMPLOYEES**: MANAGER_ID references EMPLOYEE_ID.
- COUNTRIES** to **REGIONS**: REGION_ID references REGION_ID.

SQL (DDL and DML)

SQL (DDL) Data Definition Language	
-- MySQL	<pre> CREATE TABLE departments (deptno integer PRIMARY KEY, name VARCHAR(25) NOT NULL, loc VARCHAR(30)); CREATE TABLE employees (empno integer PRIMARY KEY, fname VARCHAR(15) NOT NULL, lname VARCHAR(20) NOT NULL, hiredate DATE NOT NULL, salary decimal(9,2), commission decimal(9,2), mgr integer, deptno integer, CONSTRAINT emps_mgr_FK FOREIGN KEY (mgr) REFERENCES employees(empno), CONSTRAINT emps_deptno_FK foreign key (deptno) REFERENCES departments(deptno)); </pre>
-- Oracle	<pre> CREATE TABLE departments (deptno NUMBER(5) CONSTRAINT depts_deptno_PK PRIMARY KEY, name VARCHAR2(25) CONSTRAINT depts_name_NN NOT NULL, loc VARCHAR2(30) NOT NULL); CREATE TABLE employees (empno NUMBER(9) PRIMARY KEY, fname VARCHAR2(15) NOT NULL, lname VARCHAR2(20) NOT NULL, hiredate DATE NOT NULL, salary NUMBER(9,2), commission NUMBER(9,2), mgr NUMBER(9) CONSTRAINT emps_mgr_FK REFERENCES employees(empno), deptno NUMBER(5), CONSTRAINT emps_deptno_FK foreign key (deptno) REFERENCES departments(deptno)); </pre>

CRUD	ABCD	SQL (DML) Data Manipulation Language
Create	Altas	<pre> INSERT INTO departments(deptno, name) VALUES (123,'Accounts'); INSERT INTO departments VALUES (124, 'Sales','US'); </pre>
Delete	Bajas	Delete from departments where deptno = 124
Update	Cambios	<pre> Update departments set loc = 'US' where deptno = 123; Update departments set name = 'Sales', Loc = 'US' where deptno = 123 </pre>
Read	Despliegue	<pre> Select * from departments; Select deptno, name from departments; </pre>

Section 3 – Relationship Basics

3-1 Identifying Relationships

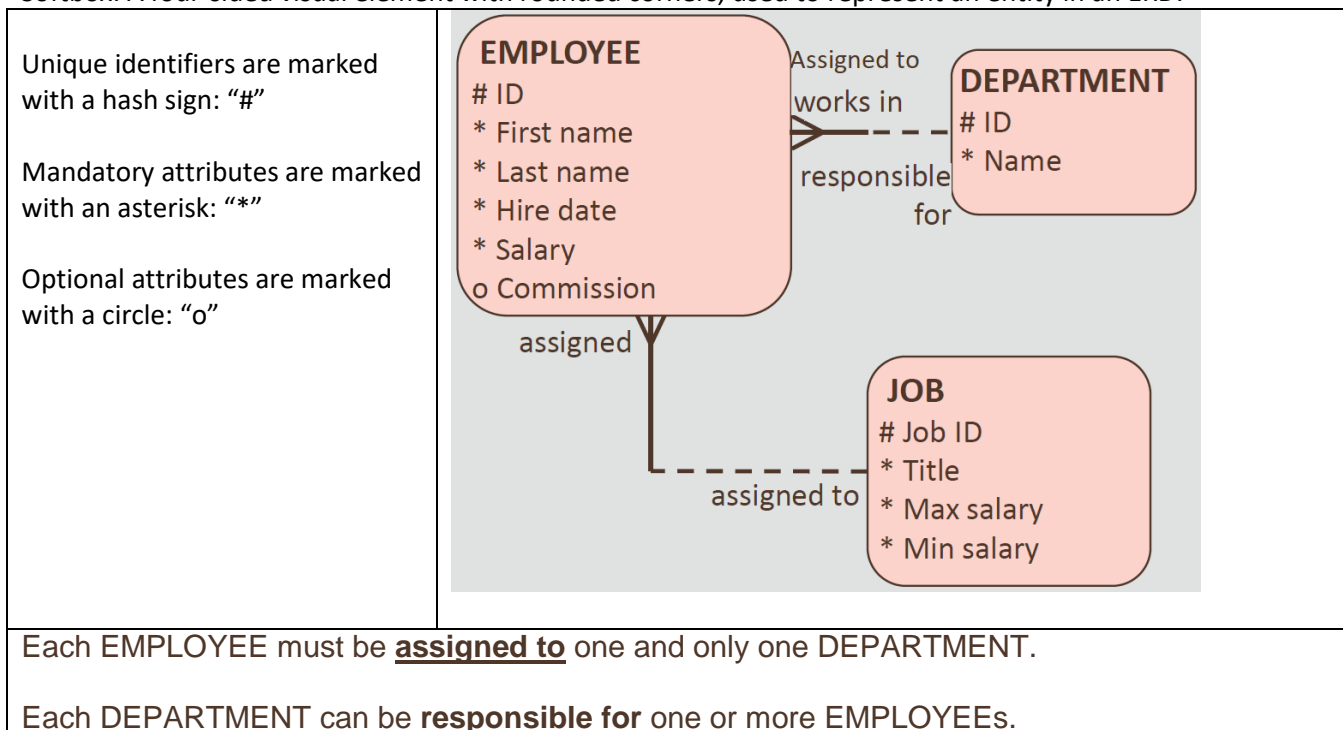
Cardinality	A property of an end of a relationship between X and Y that describes how many of X is related to Y. [1 to 1, 1 to N, N to M] Note: The cardinality of a relationship only answers whether the number is singular or plural; it does not answer with a specific plural number
Relationship	A connection or association between objects (entities).
Optionality	A property of an end of a relationship between X and Y that describes whether X must be or may be related to Y. [must , can o]

Relationships:

- Exist only between entities (or one entity and itself)
- Are bi-directional
- Are named at both ends
- Have optionality: Are either mandatory or optional
- Have cardinality

3-2 ER Diagramming Conventions

Softbox: A four-sided visual element with rounded corners, used to represent an entity in an ERD.



3-3 Speaking ERDish and Drawing Relationships

State relationships between entities in precise words (ERDish)

ERDish: The language or statements used to describe relationships between entities in an entity-relationship diagram.

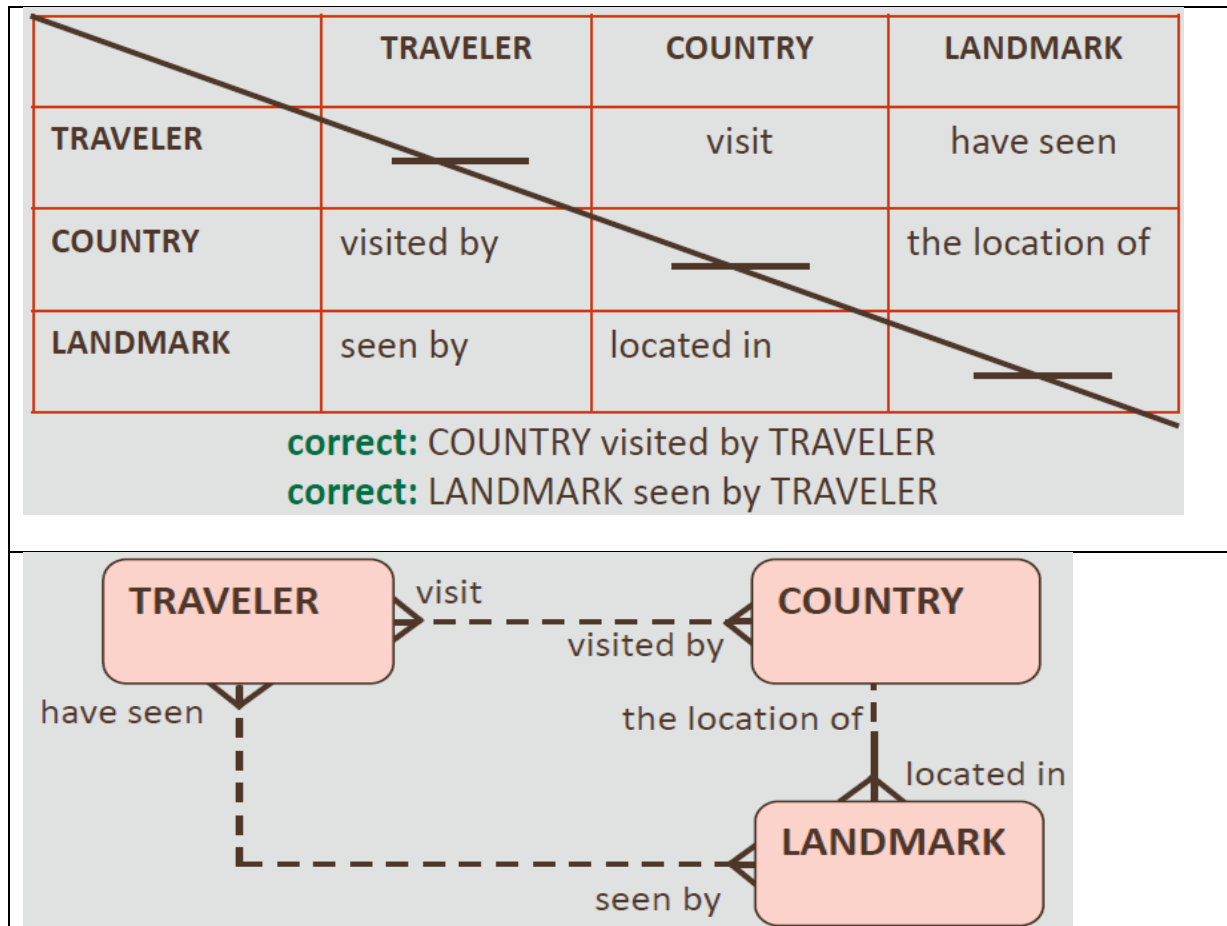
The Components of ERDish:

1. EACH
2. Entity A
3. OPTIONALITY
(must be/may be)
4. RELATIONSHIP NAME
5. CARDINALITY
(one and only one/one or more)
6. Entity B



- | | |
|--|---|
| <ol style="list-style-type: none"> 1. EACH 2. EMPLOYEE (entity A) 3. MUST (optionality, solid line) 4. WORK IN (relationship name) 5. ONE AND ONLY ONE (cardinality, single toe) 6. DEPARTMENT (entity B) | <ol style="list-style-type: none"> 1. EACH 2. DEPARTMENT (entity B) 3. MAY BE (optionality, dotted line) 4. RESPONSIBLE FOR (relationship name) 5. ONE OR MORE (cardinality, crow's foot) 6. EMPLOYEE (entity B) |
|--|---|

3-4 Matrix Diagrams



Practices

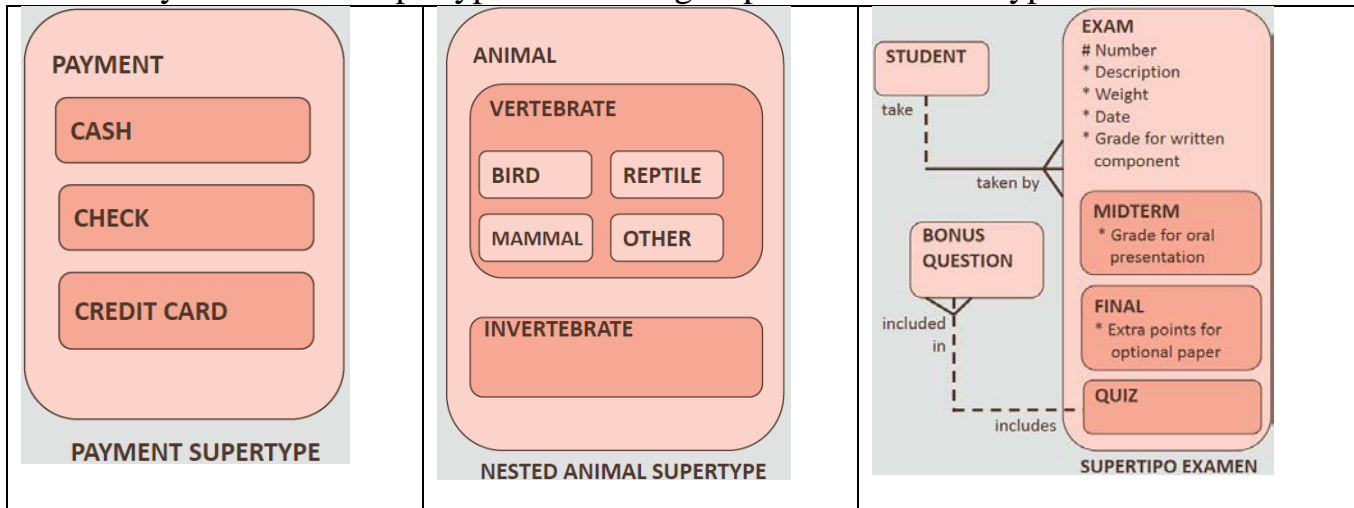
- | | | | |
|-----|---|-----------------------|---|
| 3.2 | 3 | | |
| 3.3 | 1 | 4 | 5 |
| 3.4 | 2 | Analizar solo gráfica | |

Section 4 – Super/Sub Types and Business Rules

4-1 Supertypes and Subtypes &&

Exhaustive	All subtypes are listed without omission.
Supertype	A means of classifying an entity that has subtypes.
Subtype/Subentity	Something an entity may be split into based on common attributes and/or relationships.
Mutually Exclusive	Each instance of a supertype is an instance of only one possible subtype.

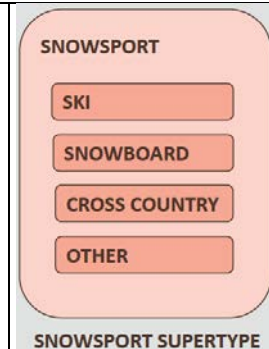
The entity is called a “supertype” and each group is called a “subtype”



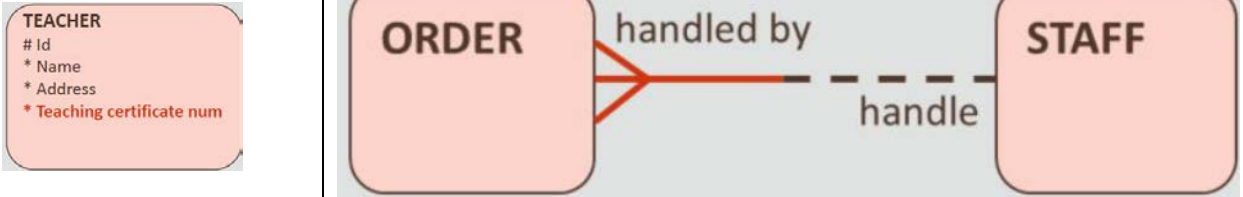
Correctly Identifying Subtypes:

- Is this subtype a kind of supertype?
- Have I covered all possible cases? (exhaustive)
- Does each instance fit into one and only one subtype? (mutually exclusive)

Instancia vertebrados o invertebrados



4-2 Documenting Business Rules

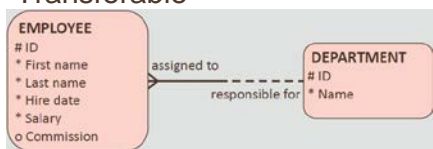
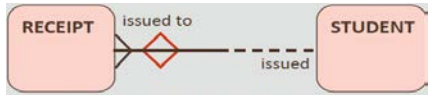
Business rule	A formalized statement of the usual, customary, or generalized course of action or behavior for a business.
Structural business rule	<p>A type of business rule that indicates the types of information to be stored and how the information elements interrelate.</p> <p>Can nearly always be diagrammed in the ERD.</p> <div>  </div>
Procedural business rule	<p>A type of business rule that is workflow or business process related. (e.g., A has to happen before B, and then C has to happen at the same time as D.) This is also called a process business rule.</p> <p>Cannot be diagrammed, but must still be documented so that they can be programmed later. Ej. Any employee whose overtime exceeds 10 hours per week must be paid 1.5 times the hourly rate.</p>

Practices

4.1 4

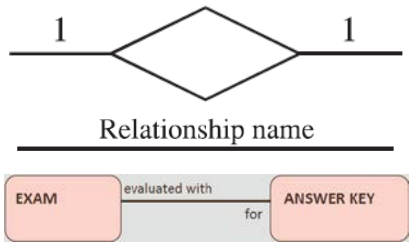

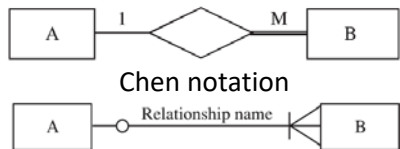


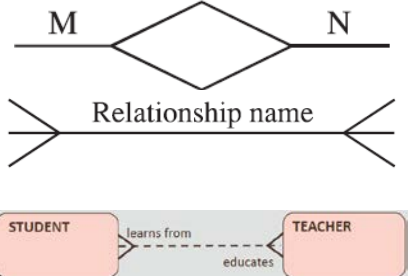
Section 5 – Relationship Fundamentals

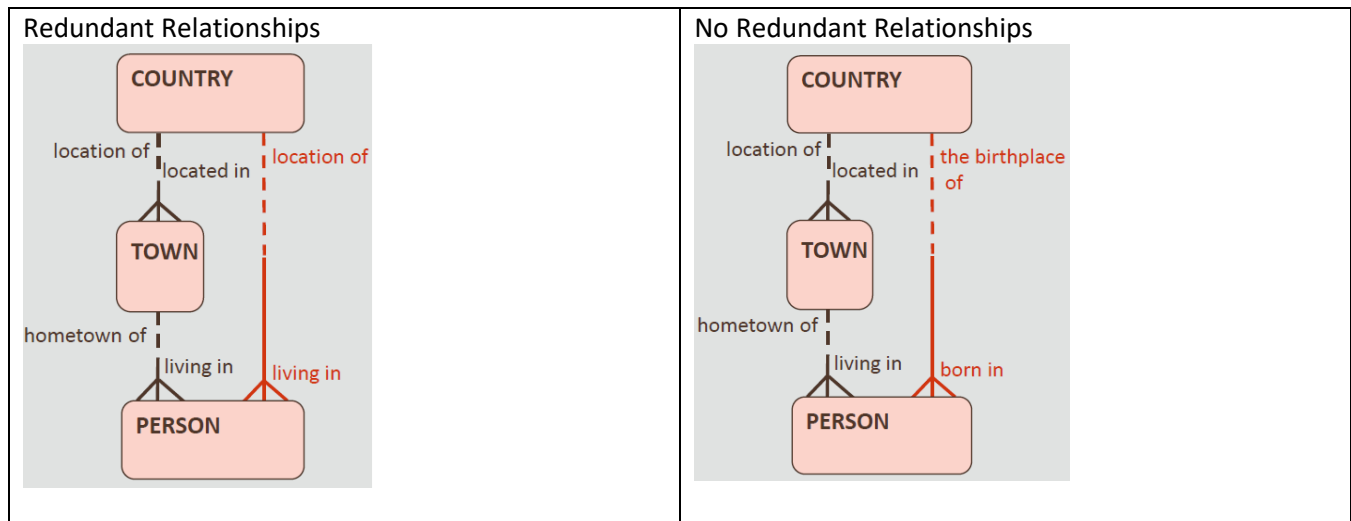
5-1 Relationship Transferability

<p>Transferable</p>  <p>Can you have an EMPLOYEE not assigned to a DEPARTMENT? ANSWER: NO Can you have a DEPARTMENT not responsible for any EMPLOYEEs? YES</p>	<p>Description of a relationship where an instance of A is related to an instance of B, and the association can be moved to another instance of B</p>
<p>Nontransferable</p>  <p>Nontransferable Relationship</p> <p>A nontransferable relationship is represented with the diamond on the relationship</p>	<p>Description of a relationship where an instance of A is related to an instance of B, and the association cannot be moved to another instance of B</p>

5-2 Relationship Types

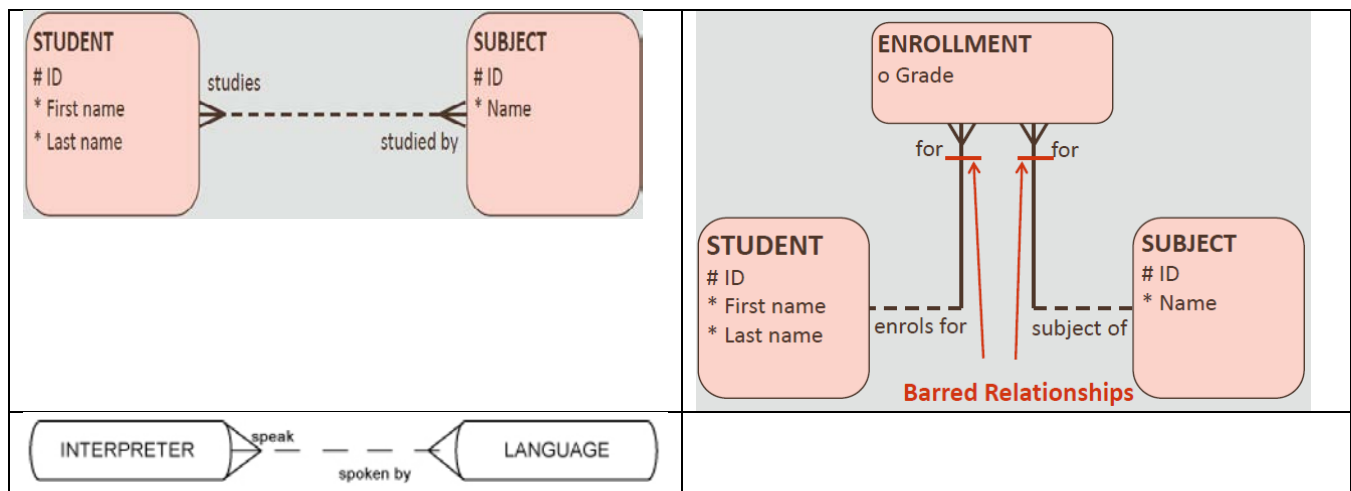
One-to-one (1:1)	A relationship where each record in Table A can be related to one, and only one, record in Table B, and each record in Table B relates to one, and only one, record in Table A.
One-to-many (1:M)	A relationship where a single record in Table A can be related to one or more records in Table B, but a single record in Table B can only be related to one record in Table A.
Many-to-many (M:M)	A relationship in which many records in one table match many records in another table
Redundant	Unnecessarily repetitive

Relationship Types 1:1	Relationship Types 1:M	Relationship Types M:M
	 <p>With optional participation for A Mandatory participation for B</p>  <p>Chen notation</p>  <p>Crow's Feet notation</p> 	



5-3 Resolving Many-to-Many Relationships

Barred relationship	A relationship that participates in an entity's unique identifier.
Intersection entity	The product of the resolution of a many to many relationship.



5-4 Understanding CRUD Requirements

Consultant	One who gives expert or professional advice
CRUD analysis	The practice of checking a data model for create, retrieve, update, and delete functions that the business requires
Functions	Used to perform calculations on data, modify individual data items, manipulate output for groups of rows, format dates and numbers for display, and convert column datatypes

Synonyms

CRUD Analysis—Create Function: INPUT, ENTER, LOAD, IMPORT, RECORD, & CREATE

CRUD Analysis—Retrieve Function: VIEW, REPORT, BRING UP, PRINT, FIND, READ, & LOOK UP

CRUD Analysis—Update Function: CHANGE, MODIFY, ALTER, & UPDATE

CRUD Analysis—Delete Function: DISCARD, REMOVE, TRASH, PURGE, & DELETE

Practices

5.1 m n

5.2 3

5.3 1