**Foundations Practice 1,2,3,4**

Tasks

**Database Foundations  
1-2: Introduction to Database  
Practices  
Exercise 1: Identify Database Design Considerations for Given Case Scenarios**

* ABC School District would like to create an online information and registration system for students to capture student-related information. The system must be designed as an online process to allow all new students to register online. It should also allow existing students to update and review all information. Create a list of important data that would need to be captured and stored in the student registration database.
* A list of important data that would need to be captured includes:

1. Student Personal information including student ID, first name, last name, DOB, gender, address, and phone number.
2. Parent/Guardian information including name, relationship, address, phone number, and occupation.
3. Emergency contact information including emergency contact name, relationship, address( if different from student), phone number, and alternate emergency contact information.
4. Academic information such as current grade level, previous schools attended, school enrollment date, expected graduation date, standardized test scores, and attendance record.
5. Medical information including health conditions, medications, immunization records, primary care physician, consent for treatment, and emergency medical instructions.
6. Enrollment information such as enrollment status, date of enrollment, and enrollment confirmation.
7. Login and security information including username, password, and security questions.

* XYZ community would like to create a library management system. The objective is for the database to handle all transactions for  
  the library. The database needs to store all the data that is relevant to managing the books, managing customers, and the day-to-  
  day activities of the library. Create a list of important data that would need to be captured and stored in the library management  
  database.
* A list of important data that would need to be captured and stored in the library management database includes:

1. Book information including title, author, publisher, ISBN, edition, and publication date.
2. Member information including member ID, name, DOB, DOB, membership date, membership type, library card number, photo ID, and borrowing history.
3. Loan information includes loan ID, book ID, membership ID, loan date, due date, return date, and loan status.
4. Fine management includes fine ID, member ID, Book ID, fine amount, fine reason, payment status, payment date, and payment method.
5. Staff information including staff ID, name, role/position, phone number, email, work schedule, login credentials, and permission/roles.
6. Inventory and acquisition management including acquisition ID, book ID, acquisition date, supplier name, cost, purchase order number, supplier contact information and received by.
7. System security and audit logs including audit log ID, action performed, performed by, date and time of action, changes made, IP address, login address, and security incidents.

**Database Foundations  
1-3: Types of Database Models  
Practices  
Exercise 1: Identify the Database Models**

1. Identify the type of database model that has been represented in the given model snapshots:
2. Hierarchical Database Model.
3. Network Model
4. Object-Oriented Model
5. Relational Model
6. Relational Model

**Database Foundations  
1-4: Business Requirements  
Practices  
Exercise 1: Business Requirements**

* LibBook is a successful digital library that rents CDs and provides Internet access for browsing their articles and magazines repository. With the growing business, LibBook needs to enhance its information system to support proposed changes to the business. LibBook attracts new members easily and the number of members is growing rapidly. However, the membership base is unstable, which is a cause for concern. The main idea is to introduce the concept of membership at LibBook. Members will pay a membership fee and initially, there will be three types of membership (corporate, student, individual) although more may be introduced later. Student membership is free. Corporate and Faculty memberships incur a fee but entitle the member to privileges. The type of membership can be changed only if sufficient justification is provided. **Your task is to identify the business rules and the associated constraints from the case scenario described.**
* Based on the scenario described for LibBook, these are the identified business rules and associated constraints:

**Business Rules**

1. Membership Requirement:

* All users must become members to rent CDs or access the full repository of articles and magazines.

2. Membership Types:

* There are three types of membership: Corporate, Student, and Individual.
* Student membership is free of charge.
* Corporate membership and Individual membership require payment of a membership fee.

3. Membership Fees and Privileges:

* Corporate members and Individual members must pay a membership fee.
* Corporate and Individual members are entitled to additional privileges

4. Membership Changes:

* A member's type of membership can be changed, but only if sufficient justification is provided.
* The system must log and review the justification before approving a change in membership type.

5. Membership Growth and Flexibility:

* The system should allow for the introduction of new membership types in the future.
* Membership rules and associated privileges should be easily configurable to accommodate future changes.

6. Membership Stability Concerns:

* The system should monitor and track membership stability (e.g., track the frequency of members joining and leaving).
* The system may incorporate features to address membership stability, such as automatic reminders for renewals, or special offers for long-term members.

**Associated Constraints**

1. Payment Processing:

* The system must support secure and reliable payment processing for Corporate and Individual memberships.
* Membership fees should be configurable based on the type of membership and may change over time.

1. Eligibility for Student Membership:

* Members applying for student membership must provide proof of student status (ex. student ID, enrollment confirmation).
* The system should validate and store documentation for Student membership eligibility.

1. Justification for Membership Change:

* Justifications for changing the membership type must be documented and reviewed before approval.
* The system must enforce business rules that define what constitutes sufficient justification).

1. Privileges Management:

* Privileges associated with different membership types must be clearly defined and enforced by the system.
* The system should automatically adjust privileges when a membership type is changed.

1. Scalability and Flexibility:

* The system must be scalable to handle the rapid growth in membership numbers.
* The system must be designed to accommodate the introduction of new membership types and rules without requiring significant changes to the underlying code.

1. Security and Privacy:

* The system must ensure the security and privacy of member data, particularly for payment information and personal details.
* The system should comply with relevant data protection regulations

By defining these business rules and constraints, LibBook can enhance its information system to better support its growing and evolving business needs while addressing concerns about membership stability and offering a flexible membership structure.

* Star Care Hospital is a multi-specialty hospital that caters to the needs of different patients. Every doctor registered with this hospital is assigned a unique ID that starts with the letter "DC". The hospital ensures that the doctors associated with them have a minimum of seven years of working experience. Every patient is required to register with the hospital on their first visit. When a patient arrives, a unique patient number starting with the letter "PT" is assigned to him/her. **Your task is to identify the business rules and the associated constraints from the case scenario described.**
* Based on the scenario described for Star Care Hospital, here are the identified business rules and associated constraints:

**Business Rules**

1. Doctor Registration and Identification:

* Every doctor registered with Star Care Hospital must be assigned a unique ID.
* The unique ID for doctors must start with the prefix "DC".
* The hospital only associates with doctors with at least seven years of working experience.

1. Patient Registration and Identification:

* Every patient must register with the hospital on their first visit.
* Upon registration, a unique patient number is assigned to each patient.
* The unique patient number must start with the prefix "PT".

**Associated Constraints**

1. Doctor ID Format:

* The unique ID assigned to each doctor must strictly follow the format "DC" followed by a unique numeric sequence.
* The system must ensure that each doctor's ID is unique and does not conflict with existing IDs in the system.

2. Doctor Experience Requirement:

* The hospital's system must validate that every doctor has a minimum of seven years of working experience before registration or association with the hospital.
* The system should not allow the registration of doctors who do not meet the experience requirement.

1. Patient Number Format:

* The unique patient number must follow the format "PT" followed by a unique numeric sequence.
* The system must ensure that each patient number is unique and does not conflict with existing patient numbers in the system.

1. Patient Registration Requirement:

* The system must enforce that every patient must register on their first visit.
* No medical services should be provided until the patient is registered and assigned a unique patient number.

1. Data Integrity and Uniqueness:

* The system must maintain data integrity by ensuring that both doctor IDs and patient numbers are unique across the entire hospital database.
* The system should implement checks to prevent duplication of doctor IDs or patient numbers.

1. Automated ID Assignment:

* The system should automate the assignment of both doctor IDs and patient numbers to ensure consistency and adherence to the required formats.

1. Experience Verification:

* The system should include a mechanism for verifying a doctor’s years of experience, such as requiring documentation or references during the registration process.

These business rules and constraints help ensure that Star Care Hospital can efficiently manage its operations by uniquely identifying its doctors and patients, enforcing necessary experience requirements, and maintaining data integrity across its systems.

**Database Foundations  
2-1: Relational Databases  
Practices  
Exercise 1: Analyze the features of a Relational Database**

* 1. Identify the possible tables and associated fields from the given scenario:  
  Book.com is an online virtual store on the Internet where customers can browse the catalog and select products of interest.  
  a. Every book has a title, ISBN, year, and price. The store also keeps the author and publisher for any book.  
  b. For authors, the database keeps the name, address, and the URL of their homepage.  
  c. For publishers, the database keeps the name, address, phone number, and the URL of their website.  
  d. The store has several warehouses, each of which has a code, address, and phone number.  
  e. The warehouse stocks several books. A book may be stocked at multiple warehouses.  
  f. The database records the number of copies of a book stocked at various warehouses.  
  g. The bookstore keeps the name, address, email, and phone number of its customers.  
  h. A customer owns several shopping carts. A shopping cart is identified by a Shopping\_Cart\_ID and contains several books.  
  i. Some shopping carts may contain more than one copy of the same book. The database records the number of copies of each  
  book in any shopping cart.  
  j. At that time, more information will be needed to complete the transaction. Usually, the customer will be asked to fill or select a  
  billing address, a shipping address, a shipping option, and payment information such as credit card numbers. An email  
  notification is sent to the customer as soon as the order is placed.
* Based on the given scenario for Book.com, we can identify several tables and their associated fields. Below is a breakdown of the possible tables along with their associated fields:

1. **Books**

* Book ID
* Title
* ISBN
* Year
* Price
* Author ID
* Publisher ID

1. **Authors**

* Author ID
* Name
* Address
* URL

1. **Publishers**

* Publisher ID
* Name
* Address
* Phone number
* URL

1. **Warehouses**

* Warehouse ID
* Code
* Adress
* Phone number

1. **Warehouse Stock**

* Warehouse stock ID
* Warehouse ID
* Book ID
* Number of copies

1. **Customers**

* Customer ID
* Name
* Address
* Email ID
* Phone number

1. **Shopping carts**

* Cart Item ID
* Shopping cart ID
* Book ID
* Number of copies

1. **Shopping cart items**

* Cart Item ID
* Shopping cart ID
* Book ID
* Number of copies

1. **Orders**

* Order ID
* Customer ID
* Shopping cart ID
* Order Date
* Billing Address
* Shipping Address
* Shipping option
* Payment info
* Email Notification Status

1. **Payments**

* Payment ID
* Order ID
* Payment Type
* Payment Amount
* Payment Status
* ABC Ltd plans to computerize its sales ordering and stock control system. A feasibility study has strongly suggested that a  
  relational database system be installed. The details of ABC's sales and stock control are as follows:  
  a. Customers send in orders for goods. Each order may contain requests for variable quantities of one or more products from  
  ABC's range. ABC keeps a stock file showing for each product the product details and the preferred supplier, the quantity in  
  stock, the reorder level and other details.  
  b. ABC delivers those products that it has in stock in response to the customer order and an invoice is produced for the  
  dispatched items. Any items that were not in stock are placed on a back-order list and these items are usually re-ordered from  
  the preferred supplier. Occasionally items are ordered from alternative sources.  
  c. In response to the invoices that are sent out to ABC's customers, the customers send in payments. Sometimes a payment will  
  be for one invoice, sometimes for part of an invoice, and sometimes for several invoices and part invoices.  
  d. Identify the tables and associated fields from the above scenario.
* Based on the scenario provided for ABC Ltd, here are the possible tables and associated fields that could be used in a relational database system for their sales ordering and stock control:

1**. Customers**

* Customer ID
* Customer Name
* Address
* Phone Number
* Email
* Credit Limit
* Current Balance

2**. Orders**

* Order ID
* Customer ID
* Order Date
* Order Status
* Total Amount

3**.** Order Items

* Order Item ID
* Order ID
* Product ID
* Quantity Ordered
* Quantity Dispatched
* Unit Price
* Total Price
* Back Order Status

4**. Products**

* Product ID
* Product Name
* Description
* Unit Price
* Preferred Supplier ID
* Quantity In Stock
* Reorder Level
* Reorder Quantity
* Alternative Supplier ID

5**. Suppliers**

* Supplier ID
* Supplier Name
* Address
* Phone Number
* Email
* Website URL

6**. Invoices**

* Invoice ID
* Order ID
* Invoice Date
* Total Amount
* Payment Status
* Due Date

7**. Payments**

* Payment ID
* Customer ID
* Payment Date
* Payment Amount
* Payment Method
* Payment Details

8**. Invoice Payments**

* Invoice Payment ID
* Invoice ID
* Payment ID
* Amount Applied

9**. Back Orders**

* Back Order ID
* Order Item ID
* Product ID
* Quantity
* Back Order Date
* Expected Delivery Date
* Supplier ID
* Alternative Supplier ID
* Reorder Status

**Database Foundations  
2-2: Conceptual and Physical Data Models  
Practices  
Exercise 1: Conceptual and Physical Models**

* Provide five reasons for creating a conceptual data model.

**1. Clarifies Business Requirements**

A conceptual data model helps to convert and clarify business requirements into a visual representation that can be easily understood by both technical and non-technical stakeholders. It focuses on what data needs to be stored and how different entities relate to each other without delving into the technical details of data implementation.

**2. Facilitates Communication**

The conceptual data model functions as a communication tool for business analysts, stakeholders, and the development team. It ensures that all parties have a shared understanding of the primary entities, their relationships, and the extent of the data managed by the system.

**3. Provides a Foundation for Logical and Physical Design**

The conceptual data model is fundamental for developing logical and physical data models. It establishes the high-level structure of the data and its relationships, which can then be transformed into a more detailed logical model (tables, fields, keys), and ultimately into a physical database implementation.

**4. Identifies Key Entities and Relationships**

A conceptual data model assists in recognizing the main entities, like customers, orders, and products, and their relationships, such as customers placing orders or products being ordered. This recognition is needed to understand the essential data elements that the system needs to manage and how they are connected.

**5. Supports Decision-Making and System Design**

By visualizing the data and its structure, a conceptual data model assists in decision-making about the scope and design of the system. It assists in identifying essential data, organizing it effectively, and determining crucial relationships for the system to operate efficiently. This understanding is important during the planning and design phases, reducing the risk of costly changes later in the development process.

* List two examples of conceptual models and physical models.
* **Conceptual models**

1. High-Level Business Process Models, which represent the major business processes in an organization, such as Order Processing, Customer Service, and Inventory Management.
2. A mind map visually organizes information around a central concept using branches to illustrate relationships between ideas. This helps to clarify thinking and structure information hierarchically. For example, a project mind map could include main branches like Objectives, Stakeholders, Resources, and Risks.

* **Physical Models**

1. A Database Schema Diagram is a physical model that outlines the actual structure of a database. It includes detailed information about tables, columns, data types, indexes, constraints, and relationships between tables.
2. A Physical Network Diagram represents the physical layout of a computer network. This model focuses on the actual hardware and physical connections, unlike a conceptual model that might only show logical connections or abstract components.

**Database Foundations  
2-3: Entities and Attributes  
Practices  
Exercise 1: Identify and draw entities as the beginning of an ERD**

* With the information provided above, identify and create the entities for the School Management System.

1. School/University

* Represents the overarching entity for the system.

1. Department

* Represents various departments within the school/university that offer courses

1. Course

* Represents the courses offered by different departments

1. Student

* Represents students who enroll in courses

1. Faculty

* Represents the instructors who teach the course

1. Academic session

* Represent different academic sessions in which students enroll and courses are offered

1. Parent information

* Represents the parent or guardian details associated with each student

1. Exam

* Represents exams conducted for the courses in each academic session

**Exercise 2: Identify and add Attributes and corresponding Mandatory and Optional notation to ERD**

* Add the appropriate attributes as well as the optionality (\*, °) to all the entities of the Academic Database

1. Department

* Department\_ID (Primary Key) \*
* Department\_Name \*
* Department\_Head °

1. Course

* Course\_ID (Primary Key) \*
* Course\_Name \*
* Credits \*
* Department\_ID (Foreign Key to Department) \*

1. Student

* Student\_ID (Primary Key) \*
* First\_Name \*
* Last\_Name \*
* DOB \*
* Gender °
* Address °
* Enrollment\_Date \*
* Parent\_ID (Foreign Key to Parent Information) \*
* Academic\_Session\_ID (Foreign Key to Academic Session) \*

1. Faculty

* Faculty\_ID (Primary Key) \*
* First\_Name \*
* Last\_Name \*
* DOB \*
* Gender °
* Address °
* Hire\_Date \*
* Department\_ID (Foreign Key to Department) \*

1. Academic Session

* Academic\_Session\_ID (Primary Key) \*
* Session\_Name \*
* Start\_Date \*
* End\_Date \*

1. Parent Information

* Parent\_ID (Primary Key) \*
* Parent\_Name \*
* Phone\_Number \*
* Email °
* Address °

1. Exam

* Exam\_ID (Primary Key) \*
* Course\_ID (Foreign Key to Course) \*
* Academic\_Session\_ID (Foreign Key to Academic Session) \*
* Exam\_Date \*
* Result\_Date \*
* Result °

**Database Foundations  
2-4: Unique Identifiers  
Practices  
Exercise 1: Identify the Unique Identifier and corresponding Primary keys**

1. How do you find a particular song in the whole collection? What would be a unique identifier for a SONG?

* The unique identifier for a song could be the Song ID or ISRC (International Standard Recording Code), which is a unique identifier assigned to each song. **Unique Identifier**: `Song\_ID` or `ISRC`

1. Think about all the students in the classroom. Each student is described by several traits or attributes. Which attribute or attributes allow you to pick a single student from the rest of the class?

* The unique attribute that identifies a single student would be the Student ID. While names and addresses can be duplicated, the Student ID is unique for each student.

**Unique Identifier**: `Student\_ID`

1. For each entity, select the attribute that could be the unique identifier of each entity

* Entity: STUDENT

Attributes: student ID, first name, last name, address

Unique Identifier: `Student\_ID`

* Entity: MOVIE

Attributes: title, date released, producer, director

Unique Identifier: The combination of title, date released, and director could serve as a unique identifier since movie titles could be reused but the release date would distinguish them. However, a better approach would be to assign a unique Movie\_ID.

Unique Identifier: `Movie\_ID` (or `Title + Date\_Released + Director`)

* Entity: LOCKER

Attributes: size, location, number

Unique Identifier: The Locker Number would be a unique identifier since each locker in a location will have a unique number.

Unique Identifier: `Locker\_Number`

**Exercise 2: Identify the Unique Identifiers and add them to the ERD**

* **Use the Academic Database ERD from the previous exercises to identify the following:**

**a. Unique Identifiers  
b. Candidate Unique Identifiers**

* Department: Primary: Department\_ID, Candidate: Department\_Name
* Course: Primary: Course\_ID, Candidate: Course\_Name
* Student: Primary: Student\_ID, Candidate: First\_Name + Last\_Name + DOB
* Faculty: Primary: Faculty\_ID, Candidate: Email
* Academic Session: Primary: Academic\_Session\_ID, Candidate: Session\_Name + Start\_Date
* Parent Information: Primary: Parent\_ID, Candidate: Phone\_Number
* Exam: Primary: Exam\_ID, Candidate: Course\_ID + Exam\_Date

**Database Foundations  
2-5: Relationships  
Practices  
Exercise 1: Identify relationships from the ERD**

* Read the relationship. Which text corresponds to the diagram?
* Answer: Each EMPLOYEE must be assigned to one and only one DEPARTMENT.  
  Each DEPARTMENT must be responsible for one or more EMPLOYEEs.

Due to the branched line, the department would be responsible for one or more employees and because there is one single line connecting the employee to the department, they would be responsible for one and only one department.

* Read each relationship in the model below. For each relationship, write the ERD statement and your comments. Use your knowledge of normal people and towns in your comments.
* Answer:
* A person can only be born in one and only one town, A town may be the birthplace of many people.
* Many people can be living in this town, but it may or may not be the hometown of all people living in it.
* A town is visited by many people and people may be visitors of a town.
* A town may be governed by a mayor.

**Exercise 2: Analyze and Model Relationships**

* Write the ERDish for each of the relationships in the Academic Database including relationship names, optionality, and cardinality. Draw the ERD including the relationships.

1. **Department-Course Relationship:**

* A department can offer many courses, but each course only belongs to one department.

**Cardinality:** 1 department 🡪 N courses

1. **Course-Student Relationship**

* A student can enroll in multiple courses, and a course can have many students.

**Cardinality:** M Students 🡪 N courses (Many-to-Many)

1. **Faculty-Department Relationships:**

* A faculty member can belong to one department, but a department can have many faculty members.

**Cardinality:** 1 Department 🡪 N Faculty

1. **Faculty-Course Relationship:**

* A faculty member can teach many courses, and a course can be taught by many faculty members.

**Cardinality:** M Faculty 🡪 N courses (Many-to-Many)

1. **Student-Academic Session Relationship:**

* A student belongs to one academic session, but an academic session can have many students.

**Cardinality:** 1 Academic Session 🡪 N students

1. **Student-Parent Information Relationship:**

* A student can have one or more parents and parent information is optional.

**Cardinality:** 1 student 🡪 1 parent information (optional)

1. **Student-Exam relationship:**

* A student can take many exams, and an exam can be taken by many students.

**Cardinality:** M students 🡪 N exams (Many-to-Many)

1. **Course-Exam Relationships:**

* A course can have many exams, but each exam is associated with one course.

**Cardinality:** 1 course 🡪 N Exams

**ERD Diagram Outline**

A screenshot of a computer

Description automatically generated**Database Foundations  
2-6: Entity Relationship Modeling (ERDs)  
Practices  
Exercise 1: Identify the components in the ERD**

Identify the possible Entities and Attributes from the given scenario.  
A company has several departments. Each department has a supervisor and at least one employee. Employees must be assigned  
to at least one, but possibly more departments. At least one employee is assigned to a project, but an employee may be on vacation and not assigned to any projects. The important data fields are the names of the departments, projects, supervisors, and employees, as well as the supervisor and employee number and a unique project number.

* The entities are departments, projects, supervisors, and employees.
* The attributes include supervisor and employee number, unique project number, and vacation dates.

Read the given business scenario. Draw the entities HAIRSTYLIST and CLIENT. List the attributes associated with each entity and  
specify whether they are mandatory or optional. Identify the UIDs. Follow the diagramming conventions discussed. State the  
ERDish for the relationships.  
“In our salon, we have a number of hairstylists. They are all salaried employees, so we keep a record of their first name, last name,  
address, phone number, social-security number, and salary. During the course of a day, a hairstylist may see several clients. On a  
slow day, a hairstylist may not work on anyone at all. We have several walk-in clients, and they each get assigned to one  
hairstylist. We just ask for their first name. We also have customers who call to make an appointment. When they do this, we ask  
for their first name, last name, and phone number. We also ask if they would like a specific hairstylist. If they have no preference,  
we assign one for them. Of course, they are allowed to switch to another hairstylist for their next visit to the salon. We are  
interested in tracking the daily appointments -- which stylist works on which client during a given day.”

**Entities and Attributes:**

**1. HAIRSTYLIST**

* **Attributes**:
  + **First Name** (Mandatory)
  + **Last Name** (Mandatory)
  + **Address** (Mandatory)
  + **Phone Number** (Mandatory)
  + **Social Security Number** (Mandatory) → Unique Identifier (UID)
  + **Salary** (Mandatory)

**2. CLIENT**

* **Attributes**:
  + **First Name** (Mandatory)
  + **Last Name** (Optional) - Only required for appointment clients, not walk-ins.
  + **Phone Number** (Optional) - Required for appointment clients, not walk-ins.
  + **Preferred Hairstylist** (Optional) - They may or may not have a preferred hairstylist.

**ERD for HAIRSTYLIST and CLIENT:**

1. **Entities:**
   * **HAIRSTYLIST**
     + First Name
     + Last Name
     + Address
     + Phone Number
     + Social Security Number (UID)
     + Salary
   * **CLIENT**
     + First Name
     + Last Name (Optional)
     + Phone Number (Optional)
     + Preferred Hairstylist (Optional)
2. **Relationships:**
   * **Relationship** between HAIRSTYLIST and CLIENT:
     + A hairstylist can work on many clients in a day.
     + A client can be seen by one hairstylist during a visit, but may have a different one for the next visit.
     + ERDish: "A hairstylist works on zero or more clients, and a client may be assigned to one hairstylist."
3. **Cardinality and Optionality**:
   * **HAIRSTYLIST – CLIENT**:

One-to-Many relationship: One hairstylist can see many clients on a given day, but each client is assigned to one hairstylist during the appointment.

Optionality:

* It is optional for a hairstylist to have clients on a slow day (0 clients).
* It is mandatory for a client to be assigned a hairstylist during their visit.

**ERDish for the Relationships:**

1. **A hairstylist may see zero or more clients** during the day (optional relationship for the hairstylist).
2. **A client is always assigned to exactly one hairstylist** for each visit, but may switch hairstylists for future visits.

* Read the given business scenario. Draw the entities TEACHER and COURSE and CLASS. List the attributes underneath each  
  entity. Specify whether they are mandatory or optional. Identify the UIDs. . State the ERDish for the relationships.  
  “We have several teachers at our school. A teacher can be assigned up to three classes per semester. If a teacher is on  
  sabbatical, he doesn’t teach that semester. We keep a record of the teacher’s first name, last name, address, phone number, and  
  email address.  
  Our school offers many courses -- such as Data Modeling, Introduction to SQL, Trigonometry, Physics, and Biology. Each course  
  has a code. For example: Data Modeling would be DM001, Trigonometry would be TR004, etc. During each semester, a course  
  may be taught in several classes -- so there could be two classes of Physics, three classes of Biology, etc. Each class can be  
  taught by only one teacher. We assign a unique ID for each class, and we also keep track of the day it is taught, the time, and the  
  classroom.”

**Entities and Attributes**

**1. TEACHER**

* **Attributes**:
  + **First Name** (Mandatory)
  + **Last Name** (Mandatory)
  + **Address** (Mandatory)
  + **Phone Number** (Mandatory)
  + **Email Address** (Mandatory)
  + **UID**: There’s no natural unique identifier in the scenario, so you can create a system-generated **Teacher ID** (Mandatory, UID).

**2. COURSE**

* **Attributes**:
  + **Course Code** (Mandatory, UID)
  + **Course Name** (Mandatory)

**3. CLASS**

* **Attributes**:
  + **Class ID** (Mandatory, UID)
  + **Day** (Mandatory)
  + **Time** (Mandatory)
  + **Classroom** (Mandatory)

**ERDish for Relationships**

1. **TEACHER – CLASS**:
   * A **teacher** can teach up to 3 **classes** per semester, and a **class** can only be taught by one teacher.
   * **ERDish**: "A teacher teaches zero, one, two, or three classes per semester. A class must be taught by one and only one teacher."
   * **Cardinality**: 1 Teacher → 0..3 Classes (One-to-Many)
2. **COURSE – CLASS**:
   * A **course** can be offered in multiple **classes**, but each **class** is associated with only one **course**.
   * **ERDish**: "A course is taught in one or more classes. A class is linked to exactly one course."
   * **Cardinality**: 1 Course → N Classes (One-to-Many)

**ERD Attributes Summary:**

* **TEACHER**:
  + Teacher ID (UID)
  + First Name (Mandatory)
  + Last Name (Mandatory)
  + Address (Mandatory)
  + Phone Number (Mandatory)
  + Email Address (Mandatory)
* **COURSE**:
  + Course Code (UID)
  + Course Name (Mandatory)
* **CLASS**:
  + Class ID (UID)
  + Day (Mandatory)
  + Time (Mandatory)
  + Classroom (Mandatory)

**Optionality**

* **Teacher-Classes Relationship**: It's **optional** for a teacher to teach classes because a teacher may be on sabbatical (0 classes).
* **Course-Class Relationship**: It’s **mandatory** for a class to be assigned a course.

**Database Foundations  
3-1: More with Relationships  
Practices  
Exercise 1: Resolve M:M Relationships**

* Resolve M: M relationships between STUDENT and the COURSE using a barred relationship
* To resolve this, a new associative entity needs to be introduced, which can be called enrollment, this entity will have foreign keys referencing both student and course and will capture any additional attributes related to the enrollment, such as enrollment date.
* Resolve M: M relationships between FACULTY and the COURSE.
* To resolve this a new associative entity called teaching assignment would be introduced, which will have foreign keys referencing both faculty and course and it can have details such as the semester, year, or faculty ID.
* Resolve M: M relationships between STUDENT, COURSE and EXAM
* To resolve this an entity can be used to link the three entities, this entity can be called exam registration, which would store information about which students are registered for exams related to specific courses.

**Exercise 2: Adding non-transferability option to an ERD**

* A STUDENT will be assigned an EXAM RESULT after taking an exam. Once an EXAM RESULT has been issued, it cannot be transferred to another STUDENT.
* For this scenario we would introduce a new entity called exam result, which will have a relationship with both the student and exam. The exam result entity will capture the results of students for each exam they take. The non-transferability of the results can be enforced by ensuring the exam results entity has a foreign key reference to a student (student ID), making sure the result is bound to the student who took the exam.

**Exercise 3: Identify and draw supertype and subtype entities**

Faculty can be either full-time or part-time. Full-time faculty receive a salary and are entitled to an insurance plan. Part-time faculty  
are paid on an hourly basis and receive no benefits. Redraw the following entity as a supertype with subtype entities reflecting the new information.

**Supertype: FACULTY**

* ID
* First Name
* Last Name
* Email
* Login Date
* Login Time

**Subtypes:**

* **Full-Time Faculty:**
  + Salary
  + Insurance Plan
* **Part-Time Faculty:**
  + Hourly Rate
  + No Benefits

**Exercise 4: Examine Exclusive Relationships (Arcs)**

Determine how exclusive relationships should be modeled in the following scenario.  
Each COURSE instance in the Academic Database can either be held ONLINE or in a SEATED location. Each SEATED location has a building name, room number and a date/time when the COURSE is offered. The ONLINE classes have a logon id and a password required to enter the COURSE. Model this new information as an Arc in the Academic Database

The **COURSE** entity can either be held **ONLINE** or in a **SEATED** location. This represents a mutually exclusive relationship.

The **ARC** will help ensure that each course is either associated with an **ONLINE** location or a **SEATED** location, but not both.

For **ONLINE** courses, attributes might include:

* **Logon ID**
* **Password**

For **SEATED** courses, attributes might include:

* **Building Name**
* **Room Number**
* **Date/Time** (when the course is offered)

To model this:

* A **supertype** called **COURSE** will be created with common attributes like **Course ID**, and **Name**.
* Two subtypes, **ONLINE\_COURSE** and **SEATED\_COURSE** will be defined, with their specific attributes. Ensuring that one course can only belong to either subtype through the exclusive arc.

**Exercise 5: Model Hierarchical Data**

In this practice, model the entities, relationships, attributes, and unique identifiers for the hierarchy of a hotel. The hotel has many floors, many suites on each floor, and many rooms within each suite.

**Entities:**

1. **Hotel**:
   * Attributes: Hotel ID, Name, Address, etc.
2. **Floor**:
   * Attributes: Floor ID, Floor Number, Hotel ID
3. **Suite**:
   * Attributes: Suite ID, Suite Number, Floor ID
4. **Room**:
   * Attributes: Room ID, Room Number, Suite ID, Room Type,

**Relationships:**

* **Hotel** has a (One-to-Many) relationship with **Floor**. A hotel can have multiple floors.
* **Floor** has a relationship with **Suite**. Each floor can have multiple suites.
* **Suite** has a relationship with **Room**. Each suite can have multiple rooms.

**ERD Structure:**

1. Hotel → Floor
2. Floor → Suite
3. Suite → Room

**Exercise 6: Model Hierarchical Data and Recursive Relationships**

Develop two ERDs to represent the following situation. Develop one as a hierarchical structure and one as a recursive structure.

* Curves Dynamics sells products throughout the United States. They are divided into four major sales regions: the Northern, Eastern, Southern, and Western regions. Each sales region has a unique region code. Each sales region is then divided into sales  
  districts. For example, the Western Region is divided into the Rocky Mountain, Northwest, Pacific Coast, and Pacific districts. Each district has a unique district code.  
  Each district is made up of sales territories. The Rocky Mountain district is composed of three territories: Wyoming-Montana, Colorado, and Utah-New Mexico. The Northwest district is made up of two territories: the Washington and Oregon-Idaho territories.  
  The Pacific Coast district is composed of two territories: the California and Nevada territories. The Pacific district includes the Hawaii territory and the Alaska territory. Each territory has a unique territory code. Each sales territory is broken down into sales areas. For example, Colorado is made up of two sales areas: the Front Range and  
  the Western Slope sales areas. Each sales area has a unique sales-area code.  
  Each salesperson is responsible for one or more sales areas, and has a specific sales quota. Each sales manager is responsible for one or more sales districts and sales directors who are responsible for one or more sales regions. Each sales manager is  
  responsible for the territories within his districts. Employees’ responsibilities do not overlap. A sales area is always the responsibility of a single salesperson, and managers and directors’ responsibilities do not overlap. Sometimes salespersons,  
  managers, and directors will be on leave or special assignments and will not have sales area responsibilities. All sales personnel are identified by their employee IDs.

**Hierarchical ERD Structure**

1. **Region:**
   * Each region (e.g., Northern, Eastern, Southern, Western) has a unique code and is divided into districts.
2. **District:**
   * Each district (e.g., Rocky Mountain, Northwest, Pacific Coast) belongs to a region and has a unique code.
3. **Territory:**
   * Territories within each district (e.g., Wyoming-Montana, Utah-New Mexico) are responsible for multiple sales areas.
4. **Sales Area:**
   * Each territory is divided into sales areas (e.g., Front Range, Western Slope), and each area has a unique code.
5. **Salesperson:**
   * Each salesperson is responsible for one or more sales areas and is assigned a quota.

**Entities and Relationships:**

* Region → District
* District → Territory
* Territory → Sales Area
* Sales Area → Salesperson

**Exercise 7: Developing a complete ERD using Supertype/Subtypes and Arcs**

Develop an ERD for the following information requirements:

* The Right-Way Rental Truck Company rents small moving trucks and trailers for local and one-way usage. There are 347 rental offices across the western United States. The rental inventory includes a total of 5,750 vehicles, including various types of trucks  
  and trailers. The data that needs to be tracked are rental agreements and vehicle assignments. Each rental office rents vehicles that they have in inventory, to customers ready to take possession of the vehicle. Reservations are not taken, and speculation on when the customer will return the rented vehicles is not tracked. The central office oversees the vehicle distribution, and directs transfers of vehicles from one rental office to another. Each rental office has an office name like “Madison Right-Way” and address. Each office also has a unique three-digit office number. Each office is a home office for some vehicles, and each vehicle is based out of a single home office.  
  Each vehicle has a vehicle ID, state of registration, and a license plate registration number. There are five different types of vehicles: 36-foot trucks, 24-foot trucks, 10-foot trucks, 8-foot covered trailers, and 6-foot open trailers, each with a type code. For  
  all vehicles, a last maintenance date and expiration date of its registration needs to be tracked. In addition, for trucks, the current odometer reading, the gas tank capacity, and whether or not it has a working radio needs to be stored. For long moves, customers  
  really prefer a radio. The current mileage is logged before the truck is rented, and then again when it returns. Additionally for trailers the maximum weight capacity must be logged. Most rental agreements are for individual customers, but a rental agreement can be for either an individual or a company. A small percentage of trucks are rented to companies. Each company is assigned a company number and the name and address of the company are tracked. The corporate sales group handles all the information separately. For each individual customer, the following information is tracked: name, home phone, address, and driver’s license state, number  
  and expiration date. If a customer damaged a vehicle, abandoned it, or did not fully pay the bill, the customer is tagged as a poor risk, and the customer may not rent again.  
  Only a single individual or company can obtain a rental agreement, and a separate rental agreement is written for each vehicle. Customers can rent two or more vehicles at the same time. Each rental agreement is identified by the originating rental office  
  number and a rental agreement number. In addition, the rental date, anticipated duration of the rental, the originating rental office, the drop-off rental office, the amount of the deposit paid, the quoted daily rental rate, and the quoted rate per mile are tracked. For  
  trailers, there is no mileage charge.

**ERD Design**

**Entities:**

1. **Rental Office**:
   * **Rental Office ID** (Primary Key)
   * Office Name
   * Address
   * Three Digit Office Code
2. **Vehicle** (Supertype for different types of vehicles):
   * **Vehicle ID** (Primary Key)
   * Vehicle Type
   * Registration Number
   * Last Maintenance Date
   * Odometer Reading
   * Is Radio Working (For trucks)
   * Vehicle Capacity (For trailers)
   * **Subtypes**:
     + **Truck** (Subtype): Specific attributes for trucks like **Vehicle Capacity** and radio **Availability**.
     + **Trailer** (Subtype): Specific attributes like **Trailer Capacity**.
3. **Customer** (Supertype for individual and company customers):
   * **Customer ID** (Primary Key)
   * Name
   * Contact Number
   * Address
   * License State
   * License Number
   * **Subtypes**:
     + **Individual Customer**: Specific attributes for individuals like **FirstName**, **LastName**, **Home Phone**.
     + **Company Customer**: Specific attributes for companies like **Company Name**, **Contact Person**.
4. **Rental Agreement**:
   * **Agreement ID** (Primary Key)
   * Agreement Number
   * Rental Office ID (Foreign Key to **Rental Office**)
   * Customer ID (Foreign Key to **Customer**)
   * Rental Date
   * Expected Return Date
   * Deposit Amount
   * Daily Rate
   * Rate Per Mile
5. **Vehicle Assignment**:
   * **Assignment ID** (Primary Key)
   * Agreement ID (Foreign Key to **Rental Agreement**)
   * Vehicle ID (Foreign Key to **Vehicle**)
   * Odometer Reading at the Rental
   * Return Odometer Reading

**Relationships:**

1. A **Rental Office** has **Vehicles** (1 relationship). Each office manages multiple vehicles.
2. A **Vehicle** can be assigned to **Rental Agreements** (M relationship via **Vehicle Assignment**). Each rental agreement may involve multiple vehicles, and each vehicle can be rented multiple times.
3. A **Customer** (either individual or company) can make one or more **Rental Agreements** (1 relationship). Each agreement is for one customer.
4. **Vehicle Assignment** associates a **Vehicle** with a **Rental Agreement**.

**Database Foundations  
3-2: Tracking Data Changes  
Practices  
Exercise 1: Track Data Change over Time**

* Construct the ERD for the given scenario. In the Academic Database, a Grade is issued to each STUDENT for each COURSE taken and stored in the STUDENT COURSE  
  DETAIL entity. A STUDENT may decide to re-take a COURSE to better their Grade. The administration would like to keep a record of the old/previous Grade as well as the new Grade. Show how the ERD would be modified to include historical Grades if the  
  Students should have them. \*\* We will not make this actual change to the ERD.

The ERD can be modified by adding a Grade History entity that will store the historical grades for each course taken by a student, linked with the Student ID and Course ID. This way both the current and past grades are maintained.

* Examine the ERD that represents classroom assignments for different exams.  
  a. Why is start time part of the UID of ASSIGNMENT?  
  b. Name at least three time-related constraints. For example: End time must be later than start time. Indicate if the constraint represents conditional non-transferability.
  1. The start time UID helps to uniquely identify a specific exam occurrence since many exams can happen on the same day. It would be important to distinguish between exams held at different times in the same room.
  2. Name three time-related constraints:
* End time must be later than start time to ensure exams won’t end before they start
* Not overlapping exams: ensuring that no two exams can be scheduled in the same room at overlapping times.
* Transferability constraint: The start time is non-transferable, meaning it cannot be modified once det for a specific exam session.

**Database Foundations  
3-3: Normalization and Business Rules  
Practices  
Exercise 1: Relational Databases**

* 1. Analyze the given table which is not normalized. The table holds information specific to items such as the Item ID, the Color of the item, and the Unit price of each item. Some of the rows in the table have repeating groups of information. Evaluate the data  
  in the table and bring the table to its first normal form:

The original table shown is not in 1NF because it contains repeating groups (multiple colors for an item in a single row). To bring the table to 1NF, each color should be separated into different rows.

**Transformed Table (1NF):**

| **Item ID** | **Color** | **Unit Price** |
| --- | --- | --- |
| IT001 | Red | $16.56 |
| IT001 | Blue | $16.56 |
| IT002 | Yellow | $17.48 |
| IT003 | Green | $19.76 |
| IT004 | Blue | $20.00 |
| IT004 | Yellow | $20.00 |

* Analyze the given table. The table is in the first normal form and has composite primary key made up of the Suppler ID and Store Id. The non-key attribute location is only dependent on the Store ID. Evaluate the data stored in the table and bring the table to second normal form:

The given table is already in 1NF but not in 2NF because "Location" is dependent on "Store ID" rather than the composite primary key ("Supplier ID" and "Store ID"). To bring it to 2NF, we need to remove partial dependencies by separating the store details into another table.

**Supplier-Store Table (after 2NF):**

| **Supplier ID** | **Store ID** |
| --- | --- |
| SPD001 | S1 |
| SPD001 | S3 |
| SPD002 | S1 |
| SPD003 | S2 |
| SPD004 | S3 |

**Store Table (after 2NF):**

| **Store ID** | **Location** |
| --- | --- |
| S1 | New York |
| S2 | Rhode Island |
| S3 | Vermont |
| S3 | Illinois |

* Analyze the given table and the data stored. In the table the Book ID is the primary key and the Category Description is dependent on the Category ID. Evaluate the data stored in the table and eliminate the transitive dependency to bring the table to  
  the third normal form:

The third table shows a transitive dependency because "Category Desc" is dependent on "Category ID," which is not the primary key. We need to split this table to eliminate this transitive dependency.

**Book Table (after 3NF):**

| **Book ID** | **Category ID** | **Price** |
| --- | --- | --- |
| 1 | 1 | $27.99 |
| 2 | 2 | $17.99 |
| 3 | 1 | $20.99 |
| 4 | 3 | $40.99 |
| 5 | 2 | $19.99 |

**Category Table (after 3NF):**

| **Category ID** | **Category Desc** |
| --- | --- |
| 1 | Cooking |
| 2 | Travel |
| 3 | Computers |

**Exercise 2: Normalize Academic Database ERD**

For the Academic Database ERD, evaluate each entity against the rules of normalization, identify the misplaced attributes, and explain which rule of normalization each misplaced attribute violates.

**Parent Information Entity:**

* **Attributes:**
  + ID
  + Parent 1 First Name
  + Parent 1 Last Name
  + Parent 2 First Name
  + Parent 2 Last Name

**Potential Violations:**

* **1NF Violation (Atomicity):** "Parent 1 First Name" and "Parent 2 First Name" could indicate that multiple values are stored within a single entity. This structure suggests that the entity is not fully normalized to 1NF. The ideal structure would be separating the parents into distinct entities (or rows), each representing one parent, to avoid redundancy and allow for easier future extension (e.g., handling more than two parents).

**Student Entity:**

* **Attributes:**
  + ID
  + First Name
  + Last Name
  + Registration Year
  + Email
  + Number of Working Days
  + Number of Days Off
  + Eligibility for Exam

**Potential Violations:**

* **2NF Violation (Partial Dependency):** "Number of Working Days," "Number of Days Off," and "Eligibility for Exam" seem to depend on more specific data that could be tracked elsewhere (e.g., in a separate entity like "Attendance" or "Exam Eligibility"). These attributes don’t depend solely on the primary key ("ID") but rather on specific attendance or performance, suggesting that the entity may not be fully normalized to 2NF.

**Academic Session Entity:**

* **Attributes:**
  + ID
  + Name

No obvious normalization violations here, as the attributes seem atomic and no partial or transitive dependencies are present.

**Department Entity:**

* **Attributes:**
  + ID
  + Name
  + Head

No obvious normalization violations in this entity. The attributes are atomic, and the entity doesn’t seem to violate 1NF, 2NF, or 3NF.

**Faculty Entity:**

* **Attributes:**
  + ID
  + First Name
  + Last Name
  + Email
  + Login Date
  + Login Time
  + Details
  + Full-Time (salary, insurance plan)
  + Part-Time (hourly rate)

**Potential Violations:**

* **2NF Violation (Partial Dependency):** The "Full-Time" and "Part-Time" details seem to represent different employment categories. These details could be extracted into separate entities, like "Employment Type" or "Contract Details," which would eliminate redundancy and better normalize the data to 2NF.

**Course Entity:**

* **Attributes:**
  + ID
  + Name

No obvious normalization violations here.

**Faculty Course Detail Entity:**

* **Attributes:**
  + Contact Hours

No obvious normalization violations here.

**Student Course Detail Entity:**

* **Attributes:**
  + Grade

No obvious normalization violations.

**Exam Entity:**

* **Attributes:**
  + ID
  + Type
  + Start Date
  + Name
  + Description

No obvious normalization violations.

**Exam Result Entity:**

* **Attributes:**
  + Grade

No obvious normalization violations.

**Exercise 3: Validate an ERD for Normalization**

* Evaluate the following unnormalized data in the USER entity and develop an entity relationship diagram that is normalized to the third normal form.

**Step 1: First Normal Form (1NF)**

The first step is to eliminate repeating groups and ensure atomicity. In this case, each row is already atomic (no multi-valued attributes), so it complies with 1NF. However, there is redundancy in the "User Name" and "Server ID" columns.

**Step 2: Second Normal Form (2NF**

The table is in 1NF, but it's not in 2NF because non-key attributes (e.g., "User Name," "Server Name") are partially dependent on "User ID" and not entirely on the composite key ("User ID" + "Message ID").

We can break this table into two entities to address this issue:

* **User Entity**:
  + User ID (Primary Key)
  + User Name
* **Message Entity**:
  + Message ID (Primary Key)
  + User ID (Foreign Key)
  + Recorded Date
  + Subject
  + Text
  + Server ID

**Step 3: Third Normal Form (3NF)**

In 2NF, transitive dependencies still exist. For example, "Server Name" depends on "Server ID," not directly on "Message ID." To remove this, we will create a separate table for servers:

* **Server Entity**:
  + Server ID (Primary Key)
  + Server Name

**Final Tables in 3NF**

**1. User Table:**

| **User ID** | **User Name** |
| --- | --- |
| 2301 | Smith |
| 5607 | Jones |
| 7773 | Walsh |
| 0022 | Patel |

**2. Message Table:**

| **Message ID** | **User ID** | **Recorded Date** | **Subject** | **Text** | **Server ID** |
| --- | --- | --- | --- | --- | --- |
| 54101 | 2301 | 05/07 | Meeting Today | There is... | 3786 |
| 54098 | 2301 | 07/12 | Promotions | I like to... | 3786 |
| 54445 | 2301 | 10/06 | Next Assignment | Your next... | 3786 |
| 54512 | 5607 | 06/07 | Lunch? | Can you... | 6001 |
| 54101 | 5607 | 05/07 | Meeting Today | There is... | 6001 |
| 54660 | 5607 | 12/01 | Jogging Today? | Can you... | 6001 |
| 54101 | 7773 | 05/07 | Meeting Today | There is... | 9988 |
| 54554 | 7773 | 03/17 | Stock Quote | The latest... | 9988 |
| 54101 | 0022 | 05/07 | Meeting Today | There is... | 9988 |
| 54512 | 0022 | 06/07 | Lunch? | Can you... | 9988 |

**3. Server Table:**

| **Server ID** | **Server Name** |
| --- | --- |
| 3786 | IMAP05 |
| 6001 | IMAP08 |
| 9988 | EMEA01 |

**ERD Design:**

In the ERD:

* **User** is connected to **Message** through **User ID**.
* **Message** is connected to **Server** through **Server ID**.
* A color scheme for a car includes specifications for paint color for the body and the interior colors and materials. For example: The “Desert” color scheme includes silver paint and gray leather interior; the “Sunburst” color scheme includes gold paint and  
  cream leather interior. Does the model below follow the rules of Third Normal Form? If you spot a violation, correct it.
* The Color Scheme likely determines the values of both Paint Color and Interior Color. This means that Paint Color and Interior Color are not directly dependent on the Car Number (the primary key), but on the Color Scheme. This violates the 3NF rule because non-key attributes should only depend on the primary key, not on other non-key attributes.
* To bring this table into **Third Normal Form**, we can break it into two related tables:

**Car Table:**

| **Number (Primary Key)** | **Make** | **Model** | **Color Scheme** |
| --- | --- | --- | --- |

**Color Scheme Table:**

| **Color Scheme (Primary Key)** | **Paint Color** | **Interior Color** |
| --- | --- | --- |

**Exercise 4: Gather database requirements and Business Rules**

* Book.com is an online store on the Internet where customers can browse the catalog and select products of interest.  
  a. Every book has a title, isbn, year and price. The store also keeps the author and publisher for any book.  
  b. For authors, the database keeps the name, address and the url of their homepage.  
  c. For publishers, the database keeps the name, address, phone number and the url of their website.  
  d. The store has several warehouses, each of which has a code, address and phone number.  
  e. The warehouse stocks several books. A book may be stocked at multiple warehouses.  
  f. The database records the number of copies of a book stocked at various warehouses.  
  g. The bookstore keeps the name, address, email-id, and phone number of its customers.  
  h. A customer owns several shopping carts. A shopping cart is identified by a Shopping\_Cart\_ID and contains several  
  books.  
  i. Some shopping carts may contain more than one copy of same book. The database records the number of copies of each  
  book in any shopping cart.  
  j. At that time, more information will be needed to complete the transaction. Usually, the customer will be asked to fill or  
  select a billing address, a shipping address, a shipping option, and payment information such as credit card number. An email notification is sent to the customer as soon as the order is placed.  
  **Your task is to identify the business rules**.

**Business Rules:**

* **Book Details:**
  + Each book has a **title**, **ISBN**, **year of publication**, and **price**.
* **Author Information:**
  + For each book, the store keeps the **name**, **address**, and **URL** of the **author's homepage**.
* **Publisher Information:**
  + The store keeps the **name**, **address**, **phone number**, and **URL** for each **publisher** of a book.
* **Warehouse Management:**
  + The store has **several warehouses**, and each warehouse has a **code**, **address**, and **phone number**.
  + A **book may be stocked** in **multiple warehouses**.
  + The database records the **number of copies of a book** stored in each warehouse.
* **Customer Information:**
  + The bookstore keeps the **name**, **address**, **email**, and **phone number** of its **customers**.
* **Shopping Carts:**
  + A **customer can own multiple shopping carts**. Each shopping cart is identified by a **Shopping Cart ID** and contains several books.
  + A shopping cart can have **multiple copies** of the **same book**.
  + The database records the **number of copies** of each book in the shopping cart.
* **Transaction Process:**
  + During the transaction process, the customer will select a **billing address**, **shipping address**, and **shipping option**.
  + The customer provides **payment information** (such as credit card details) during the transaction.
  + An **email notification** is sent to the customer once the order is placed.
* Identify if the given description can be categorized as a Structural Business rule, Procedural Business rule, or Programmatic Business rule.
* **All teachers in our school must possess a valid teaching certificate**
* This is a **Structural Business Rule** because it defines a requirement related to the structure of the organization (teachers must meet a specific qualification).
* **Each Department must offer a Course**
* This is a **Structural Business Rule** as it defines a structural relationship between departments and courses.
* **Approval of travel requests to an event must be signed by the project manager of the event**
* This is a **Procedural Business Rule** because it defines a process or action that must be followed for travel requests.
* **A customer may make numerous payments on account**
* This is a **Procedural Business Rule** as it specifies how customers can interact with the system (allowing multiple payments).
* **A machine operator may not work more than 10 hours in a day**
* This is a **Procedural Business Rule** since it governs a limitation or condition that affects an employee’s working hours.
* **The Rental amount in RENTAL is calculated from the Rental rate multiplied by the number of days**
* This is a **Programmatic Business Rule** because it defines a specific calculation that would be implemented in the system's logic.
* **A Customer can have zero, one, or many ORDERS**
* This is a **Structural Business Rule** as it defines the relationship between customers and orders.
* **The Total cost of the RENTAL is calculated from the sum of Insurance amount, Rental amount, and Late charge**
* This is a **Programmatic Business Rule** as it defines a complex calculation that the system must perform.
* **A customer’s debt must not exceed the customer’s credit limit**
* This is a **Programmatic Business Rule** since it involves a constraint that would be enforced by the system.

**Database Foundations  
3-4: Data Modeling Terminology and Mapping  
Practices  
Exercise 1: Identify entities, attributes, instances and their corresponding tables, rows and columns**

1. Match the ERD elements to their corresponding database elements.

Attribute 🡪 a. Column

Entity 🡪 f. Table

ER Model 🡪 c. Physical Design

Instance 🡪 e. Row

Primary UID (Unique Identifier) 🡪 d. Primary Key

Relationship 🡪 b. Foreign Key

Secondary UID 🡪 g. Unique Key

1. Identify the table diagram notations listed below

**pk 🡪** Primary Key  
**fk 🡪** Foreign Key **uk 🡪** Unique Key  
**e 🡪** Entity  
**o 🡪** Optional Attribute

1. Create short names for the terms below based on the naming convention rules

Authors 🡪 AUTH  
 Publishers 🡪 PUB  
 Customers 🡪 CUST

1. These three entities—SONG, EVENT, and CUSTOMER—play a role in a DJ business and are listed as the first three column headings in the table below. The fourth column contains a list of attributes. Use an X or a check mark to indicate that the attribute could  
   belong to one or more of the entities listed. For example, could Title be an attribute for Song, for Event, and/or for Customer?

|  |  |  |  |
| --- | --- | --- | --- |
| **SONG** | **EVENT** | **CUSTOMER** |  |
| X | X |  | Title |
| X | X |  | Description |
|  |  |  | Venue |
|  |  | X | First Name |
| X | X |  | Phone Number |
|  |  |  | Release Date |
|  |  | X | Last Name |
| X | X |  | Type |
|  |  | X | Email address |

**Exercise 2: Mapping the Academic Database**

* With the ERD provided below, map the entities, attributes, and UIDs to tables, rows, and keys using a table diagram as shown

|  |
| --- |
| **PARENT** |
| First Name |
| Last Name |
| Phone Number\* |
| Address |

|  |
| --- |
| **STUDENT** |
| Student ID\* |
| First Name |
| Phone Number |
| Address |
| Email |
| **STUDENT ATTENDANCE** |
| Number of Working Days |
| Number of Days Off |
| Eligibility for Examº |
| Student ID\* |

|  |
| --- |
| **STUDENT COURSE DETAIL** |
| Student ID\* |
| Course ID\* |
| Grade |

|  |
| --- |
| **DEPARTMENT** |
| ID\* |
| Name |
| Head |

|  |
| --- |
| **COURSE** |
| ID\* |
| Name |

|  |
| --- |
| **ONLINE** |
| Login ID\* |
| Password |

|  |
| --- |
| **SEATED** |
| Course ID\* |
| Building |
| Room |
| Date/Time |

|  |
| --- |
| **ACADEMIC SESSION** |
| ID\* |
| Name |

|  |
| --- |
| **EXAM** |
| ID\* |
| Start Date |

|  |
| --- |
| **EXAM TYPE** |
| Type |
| Name\* |
| Descriptionº |

|  |
| --- |
| **FACULTY** |
| ID\* |
| First Name |
| Last Name |
| Email |

|  |
| --- |
| **FULL TIME** |
| ID\* |
| Salary |
| Insurance plan |

|  |
| --- |
| **PART-TIME** |
| ID\* |
| Hourly rate |

|  |
| --- |
| **FACULTY COURSE DETAIL** |
| Contact Hours |
| Faculty ID\* |
| Course ID\* |

|  |
| --- |
| **FACULTY LOGIN DETAIL** |
| Login Date/Time\* |