Vanessa Chen
HHA 507 - Oracle SQL Training
Dr. Hants Williams

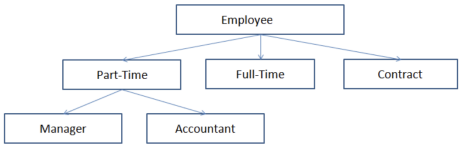
Database Foundations 1-2: Introduction to Database Practices

- *Exercise 1:* Identify Database Design Considerations for Given Case Scenarios
    - Task #1: ABC School District would like to create a student on-line information and registration system to capture student-related information. The system needs to be designed as an on-line process to allow all new students to register on-line. 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.
        - Student personal information
            - Examples: first name, last name, gender, date of birth, home address, grade year
        - Parent/guardian information
            - Examples: first name, last name, phone numbers, emergency contact information
        - Financial information
            - Examples: household income, financial assistance
        - Education information
            - Examples: previous schools attended, course transcripts, academic accommodations
        - Health information
            - Examples: medical history, immunization records
    - Task #2: 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.
        - Book information
            - Examples: library book ID, title, author, language, publisher, edition, genre, ISBN
        - Customer information
            - Examples: library card number, first name, last name, date of birth, home address, phone number, account history
        - Inventory management
            - Examples: book information, condition, acquisition date, availability, cost

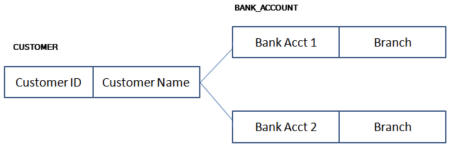
        - Fine management
            - Examples: fine number, library card number, reason for fine, fine amount, payment status
        - Reservation management
            - Examples: reservation number, book information, customer information, reservation date, pick-up time, status

Database Foundations 1-3: Types of Database Models Practices

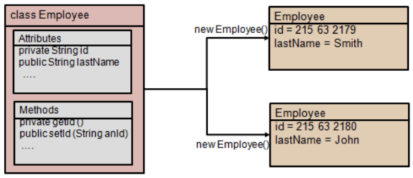
- *Exercise 1:* Identify the Database Models
    - A.



        - Answer: hierarchical model
    - B.



        - Answer: network model
    - C.



        - Answer: object-oriented model
    - D.

CUSTOMER

Account ID	Customer Name	Branch
A0001	Jeff Covey	Burlington Blvd
A0002	William Jake	Sheldon Park
A0003	Mary Schmidt	Notre Dam Street

        - Answer: relational model
    - E.

	Book ID	Author Name	Book Title
Record 1	BK0001	Oscar Wilde	A Vision
Record 2	BK0002	Leo Tolstoy	War and Peace
Record 3	BK0003	Oliver Goldsmith	Citizen of the World
Record 4	BK0003	Oliver Goldsmith	Androcles and the Lion

        - Answer: flat file model

Database Foundations 1-4: Business Requirements Practices

- *Exercise 1:* Business Requirements

- Task #1: LibBook is a successful digital library that rents CDs and provides access to Internet for browsing their repository of articles and magazines. With the growing business, LibBook needs to enhance their information system to support proposed changes to the business. LibBook attracts new members easily and the number of members is growing rapidly. The membership base is not stable, however, 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.
    - Business rule:
        - LibBook will be introducing memberships. Customers must pay a membership fee. They will start with introducing three types of memberships. The choices are corporate, student, and individual. Student memberships will be free. Corporate and faculty memberships will incur a fee, but these members will have privileges.
    - Constraints:
        - Customers can change their type of memberships, but they will need to provide justification for the adjustment.
- Task #2: Star Care hospital is a multi-specialty hospital that caters to 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 letters "PT" is assigned to him/her. Your task is to identify the business rules and the associated constraints from the case scenario described.
    - Business rule:
        - Star Care Hospital will be assigning every doctor with a unique ID starting with “DC”. Patients are required to complete registration on their visit and their unique patient numbers will be starting with “PT”.
    - Constraints:
        - The doctors associated with the unique ID starting with “DC” must have a minimum of seven years of working experience.

Database Foundations 2-1: Relational Databases Practices

- *Exercise 1:* Analyze the features of a Relational Database
    - Task #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
            - Create a books table. The fields would be the book title, ISBN, publication_year, price, author, and publisher. The primary key could be the ISBN or develop a unique book_ID for each book. Author and publisher could also be foreign keys for an authors table and publishers table.
        - b. For authors, the database keeps the name, address and the URL of their homepage.
            - Create an authors table. A unique author_ID could be developed for each author and be used as the primary key. The fields would be the author_ID, first_name, last_name, address, and website_URL.
        - c. For publishers, the database keeps the name, address, phone number and the URL of their website.
            - Create a publishers table. A unique publisher_ID could be developed for each publisher and be used as the primary key. The fields would be the publisher_ID, first_name, last_name, address, phone_number, and website_URL.
        - d. The store has several warehouses, each of which has a code, address and phone number.
            - Create a warehouses table. The primary key is the warehouse_code. The fields would be the warehouse_code, address, and phone_number.
        - e. The warehouse stocks several books. A book may be stocked at multiple warehouses.
            - Create a warehouse book stocks table. The primary key will be book_ID. The fields would be the book_ID and warehouse_code.
        - f. The database records the number of copies of a book stocked at various warehouses.
            - Create a warehouse book stocks table. The primary key will still be book_ID to address the scenario of the number of copies of a book being stocked at various warehouses. The fields would be the book_ID, warehouse_code, and book_quantity.

        - g. The bookstore keeps the name, address, email-id, and phone number of its customers.
            - Create a customers table. A unique customer_ID could be developed for each customer and be used as the primary key. The fields would be the customer_ID, first_name, last_name, address, email, and phone_number.
        - h. A customer owns several shopping carts. A shopping cart is identified by a Shopping_Cart_ID and contains several books.
            - Create a shopping carts table. The primary key is the shopping_cart_ID. The fields are the shopping_cart_ID, customer_ID, and book_ID.
        - 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.
            - Create a shopping carts table. The primary key will be the book_ID to address the scenario of some shopping carts containing more than one copy of the same book. The fields will be book_ID, shopping_cart_ID, and customer_ID.
        - 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.
            - Create an orders table. A unique order_ID could be developed for each order and be used as the primary key. The fields would be the order_ID, customer_ID, date, billing_address, shipping_address, shipping_option, total_amount, payment_information, and estimated_arrival.
    - Task #2: 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. Identify the tables and associated fields from the given scenario. 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.
            - Create a customers table. A unique customer_ID could be developed for each customer and be used as the primary key. The fields would be the customer_ID, first_name, last_name, address, phone_number, email.

        - Create a products table. A unique product_ID could be developed for each product and be used as the primary key. The fields would be product_ID, first_name, last_name, description, preferred_supplier, stock_quantity, and reorder_level.
        - Create an orders table. A unique order_ID could be developed for each order and be used as the primary key. The fields would be the order_ID, customer_ID, billing_address, shipping_address, shipping_option, total_amount, payment_information, email_address, order_date, and estimated_arrival_date.
    - 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.
        - Create an invoices table. A unique invoice_ID could be developed for each order/invoice and be used as the primary key. The fields would be the invoice_ID, date, and order_ID.
        - Create a back order table. A unique back_order_ID could be developed for each back order item and be used as the primary key. The fields would be the back_order_ID, product_ID, reorder_date, supplier, quantity_ordered, and estimated_delivery_date.
    - 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.
        - Create a payments table. A unique payment_ID could be developed for each payment made by a customer. The fields would be the payment_ID, invoice_ID, order_ID, customer_ID, date, paid_amount, and remaining_balance.

Database Foundations 2-2: Conceptual and Physical Data Models Practices

- *Exercise 1:* Conceptual and Physical Models
    - Task #1: Provide five reasons for creating a conceptual data model.
        - 1. Displays requirements
            - Provides an overview of system expectations
        - 2. Identifies entities and relationships
            - Organizes the important components of the system
        - 3. Guides development
            - Beginning step on how the client’s vision can executed
        - 4. Avoid redundancy

        - Allows for the structure of the system to be consistent
    - 5. Facilitates communication
        - Makes sure business needs are addressed and understood
- Task #2: List two examples of conceptual models and physical models.
    - Example #1
        - Conceptual model
            - Entities: customer, product, shopping cart, order
            - Relationships: customer to shopping cart, shopping cart to product, shopping cart to order, order to product, order to customer
        - Physical model
            - Customer table: customer_ID (primary key), first_name, last_name, email, phone_number, address
            - Product table: product_ID (primary key), name, price, stock_quantity
            - Shopping cart table: cart_ID (primary key), customer_ID (foreign key), product_ID (foreign key), [insert product_ID]_quantity
            - Order table: order_ID (primary key), cart_ID (foreign key), customer ID (foreign key), billing_address, shipping_address, shipping_option, total_amount, payment_information, order_date, and estimated_arrival
    - Example #2
        - Conceptual model
            - Entities: student, course, professor
            - Relationships: student to course, course to professor, student to professor
        - Physical model
            - Student table: student_ID (primary key), first_name, last_name, major, email, address, phone_number, course_ID (foreign key)
            - Course table: course_ID (primary key), professor_ID (foreign key), name, credit_amount,
            - Professor table: professor_ID (primary key), course_ID (foreign key), first_name, last_name, department, email, office_number

Database Foundations 2-3: entities and Attributes Practices

- Scenario Summary
    - Summary of how the Academic Database (School Management System) works:

        - a. A School/University has many Departments which offer courses to students in a given academic session.
        - b. Each of these courses is taught by a faculty.
        - c. Students enroll for different courses in an academic session.
        - d. Besides the registration details, the parent information of the student also needs to be maintained by the University/School.
        - e. The Department maintains the student's attendance details which would decide the eligibility of the student to take up the exams for that academic session.
        - f. For each academic session, exams are conducted and the results are shared with the student within a stipulated period of time.
        - g. The Department also maintains a log of the Faculty login and logout time for their reporting needs.
- *Exercise 1:* Identify and draw entities as a beginning of an ERD
    - With the information provided above, identify and create the entities for the School Management System.
        - Entities: department, course, student, faculty, enrollment, academic session, registration, parent information, attendance, exam, exam result, faculty log
- *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.
        - **Course**
            - *Course_ID
            - *Department_ID
            - *Faculty_ID
            - *Credit_amount
            - *Schedule
            - *Location
            - °Description
            - °Prerequisites
        - **Department**
            - *Department_ID
            - *Name
            - *Location
            - *Email
            - *Phone_number
        - **Student**
            - *Student_ID

        - *First_name
        - *Last_name
        - *DOB
        - *Major
        - °Minor
        - *Enrollment_date
        - *Graduation_date
        - *Email
        - *Phone_number
        - °Address
- **Faculty**
        - *Faculty_ID
        - *First_name
        - *Last_name
        - *Department_ID
        - *Email
        - *Phone_number
        - °Office_location
- **Academic Session**
        - *Session_ID
        - *Name
        - *Start_date
        - *End_date
        - °Description
- **Parent Information**
        - *Parent_ID
        - *First_name
        - *Last_name
        - *Student_ID
        - *Email
        - *Phone_number
        - °Address
        - °Occupation
- **Exam**
        - *Exam_ID
        - *Course_ID
        - *Session_ID
        - *Faculty_ID
        - *Date
        - *Start_time

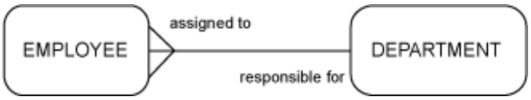
- *End_time
- *Location
- °Description

Database Foundations 2-4: Unique Identifiers Practices

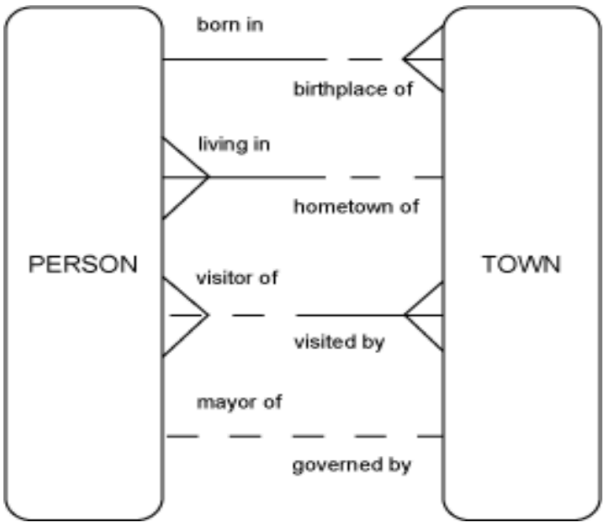
- *Exercise 1:* Identify the Unique Identifier and corresponding Primary keys
  - Task #1: How do you find a particular song in the whole collection? What would be a unique identifier for SONG?
    - In a collection, you can find a particular song through searching its title. For a unique identifier for a song, it would be beneficial to have a composite unique identifier. Since multiple songs can have the same title, a composite unique identifier could consist of the song title and song artist to better distinguish a particular song.
  - Task #2: 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 attributes could be first name, last name, student ID, and/or date of birth.
  - Task #3: 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
    - Entity: MOVIE
      Attributes: title, date released, producer, director
    - Entity: LOCKER
      Attributes: size, location, number
- *Exercise 2:* Identify the Unique Identifier and add to the ERD
  - 1. Use the Academic Database ERD from the previous exercises to identify the following:
    - a. Unique Identifiers: Course_ID, Department_ID, Faculty_ID, Student_ID, Exam_ID, Course_ID, Parent_ID, Session_ID
    - b. Candidate Unique Identifiers: the email attribute is the only candidate unique identifier from the previous exercise

Database Foundations 2-5: Relationships Practices

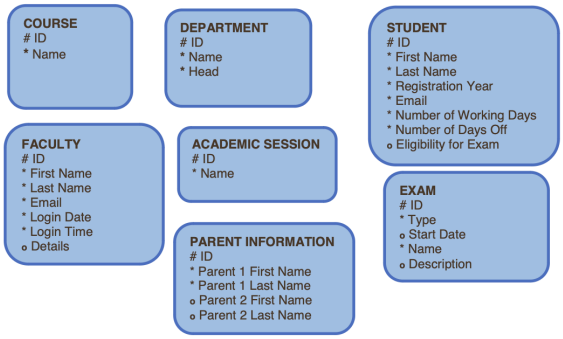
- *Exercise 1:* Identify relationships from the ERD
  - Task #1: Read the relationship. Which text corresponds to the diagram?



        - Answer: Each EMPLOYEE may be assigned to one or more DEPARTMENTs. Each DEPARTMENT must be responsible for one or more EMPLOYEEs.
    - Task #2: 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.

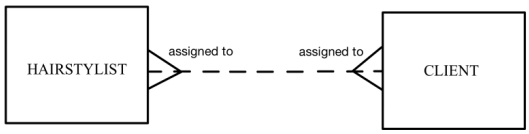


        - Each PERSON must be born in one and only one TOWN. Each TOWN may be the birthplace of one or more PERSONS.
        - Each PERSON must be living in one or more TOWNS. Each TOWN may be the hometown of and only one PERSONS.
        - Each PERSON may be the visitor of one or more TOWNS. Each TOWN must be visited by one or more PERSONS.
        - Each PERSON may be the mayor of one and only one TOWN. Each TOWN may be governed by one and only one PERSON.
- *Exercise 2:* Analyze and Model Relationships
    - Task #1:Write the ERDish for each of the relationships in the Academic Database including relationship names, optionality and cardinality. Draw the ERD including the relationships.



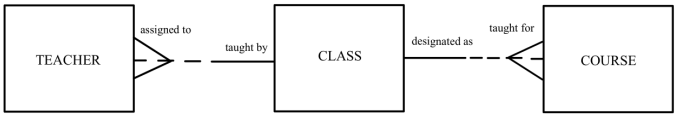
Database Foundations 2-6:

- *Exercise 1:* Identify the components in the ERD
    - Task #1: 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.
        - Entities: department, employee, project
        - Attributes
            - Department
                - Department_ID
                - Name
                - Supervisor_ID
                - Employee_ID
            - Employee
                - Employee_ID
                - First_name
                - Last_name
                - Department_ID
                - Project_ID
            - Project
                - Project_ID
                - Name
    - Task #2: 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."



- ERDish
  - Each HAIRSTYLIST may be assigned to zero or more CLIENTS. Each CLIENT may be assigned to one or more HAIRSTYLISTS.
- Attributes
  - Hair Stylist
    - *Stylist ID (UID)
    - *First_name
    - *Last_name
    - *Address
    - *Phone_number
    - *SSN
    - *Salary
  - Client
    - *Client_ID (UID)
    - *First_name
    - °Last_name
    - °Phone_number

- Task #3: 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.”



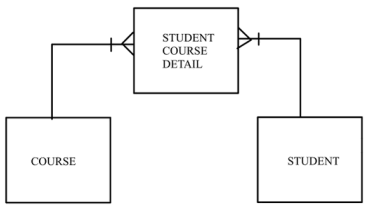
- ERDish

- Each TEACHER may be assigned to one or more CLASSES. Each CLASS must be taught by one TEACHER.
- Each CLASS must be designated as one and only one COURSE. Each COURSE may be taught for one or more CLASSES.

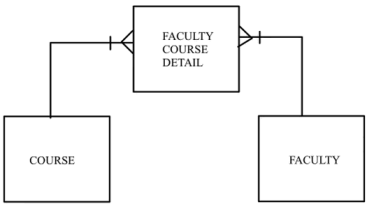
- Attributes
  - Teacher
    - *Teacher_ID (UID)
    - *First_name
    - *Last_name
    - *Phone_number
    - *Email
    - °Sabbtical_status
  - Course
    - *Course_code (UID)
    - *Name
  - Class
    - *Class_ID (UID)
    - *Course_code
    - *Date
    - *Time
    - *Location

Database Foundations 3-1: More with relationships Practices

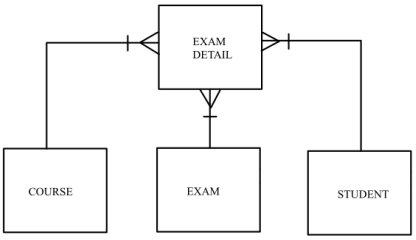
- *Exercise 1:* Resolve M:M Relationships
  - Task #1: Resolve M: M relationships between STUDENT and the COURSE using a barred relationship.



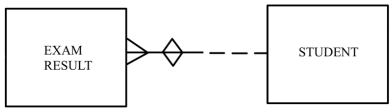
  - Task #2: Resolve M: M relationships between FACULTY and the COURSE.



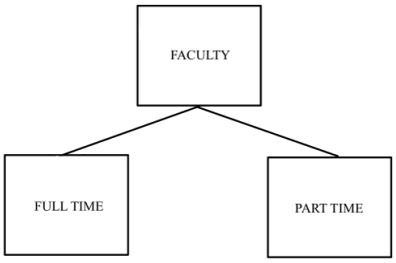
  - Task #3: Resolve M: M relationships between STUDENT,COURSE and EXAM.



- *Exercise 2:* Adding nontransferability option to an ERD
    - Task #1: 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.

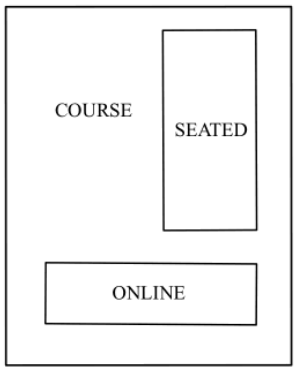


- *Exercise 3:* Identify and draw supertype and subtype entities
    - Task #1: Faculty can be either full time or part time. Full time faculty receive a salary and are entitled to an insurance plan. Part time faculty are paid on an hourly basis and receive no benefits. Redraw the following entity as a supertype with subtype entities reflecting the new information.



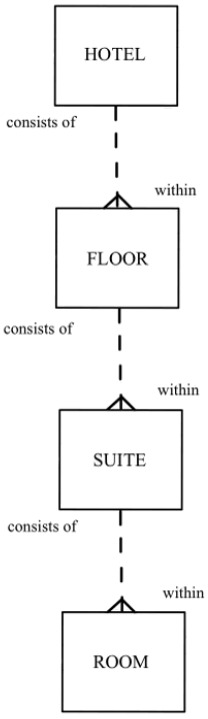
        - Faculty
            - Faculty_ID
            - First_name
            - Last_name
            - Email
            - Login_date
            - Login_time
        - Full time
            - Salary_amount
            - Insurance_plan
        - Part time
            - Wage _amount
- *Exercise 4:* Examine Exclusive Relationships (Arcs)

- Task #1: 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.



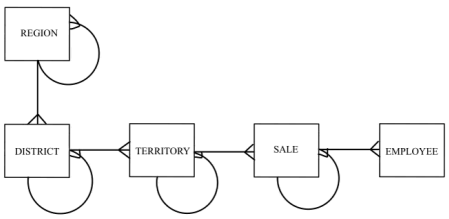
  - Course
    - Course_ID
    - Department_ID
    - Faculty_ID
    - Credit_amount
    - Schedule
    - Location
  - Seated
    - Building_name
    - Room_numver
    - Date
    - Time
  - Online
    - Logon_ID
    - Password

- *Exercise 5:* Model Hierarchical Data
  - Task #1: 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.



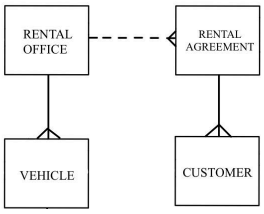
    - Hotel
        - Hotel_ID (UID)
        - Hotel_name
        - Address
        - Phone number
    - Floor
        - Floor_number (UID)
        - Hotel_ID (FK)
    - Suite
        - Suite_number (UID)
        - Floor_number (FK)
    - Room
        - Room_number (UID)
        - Suite_number (FK)
        - Room_type
        - Bed_amount
- *Exercise 6:* Model Hierarchical Data and Recursive Relationships
    - Task #1: 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.



- *Exercise 7:* Developing a complete ERD using Supertype/Subtypes and Arcs
    - Task #1: 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 is 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.

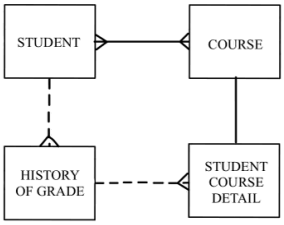


- Rental office
  - Office_number (UID)
  - Name
  - Address
- Vehicle
  - Vehicle_ID (UID)
  - State_registration
  - License_plate
  - Vehicle_type_code
  - Maintenance_date
  - Registration_expiration
  - Weight_capacity
  - Odoteter_reading

        - Gas_capacity
        - Radio_status
    - Rental agreement
        - Agreement_number (UID)
        - Office_number_orginiation
        - Rental_duration
        - Rental_date
        - Office_number_dropoff
        - Deposit_amount
        - Rental_rate
        - Rate_per_mile
        - Customer_ID
    - Customer
        - Cusotmer_ID (UID)
        - First_name
        - Last_name
        - Phone_home
        - Address
        - License_state
        - License_number
        - License_expiration
        - Poor_risk_status

Database Foundations 3-2: Tracking Data Changes Practice

- *Exercise 1:* Track Data Change over Time
    - Task #1: 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 STUDENT should have them. ** We will not make this actual change to the ERD.



    - Student
        - Student_ID (UID)

            - First_name
            - Last_name
            - DOB
            - Major
            - Minor
            - Enrollment_date
            - Graduation_date
            - Email
            - Phone_number
            - Address
        - Course
            - Course_ID (UID)
            - Department_ID
            - Faculty_ID
            - Credit_amount
            - Schedule
            - Location
            - Description
            - Prerequisites
        - Student course detail
            - Record_ID (UID)
            - Student_ID
            - Session_ID
            - Final_grade
        - History of grade
            - Retake_record_ID (UID)
            - Previous_grade
            - New_grade
    - Task #2: Examine the ERD that represents classroom assignments for different exams.
        - a. Why is start time part of the UID of ASSIGNMENT?
            - The start time attribute adds more context to the entity. There could be many assignments with the same date, but they may begin at different times.
        - 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 end time of the assignment must be later than the start time to avoid any scheduling errors

- 2. It needs to be ensured that there are no overlapping assignments in date and time for the same class room.
- 3. The assignment duration cannot exceed the start time and end time to make sure that the scheduling needs of other assignments are met

Database Foundations 3-3: Normalization and Business Rules Practices

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

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

    - The table is not in its first normal form because there are instances with multiple values in the color attribute. To bring the table to the first normal form, another table should be created and titled item color with item ID as a foreign key.
  - Task #2: Analyze the given table. The table is in the first normal form and has composite primary key made up of the Supplier 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:

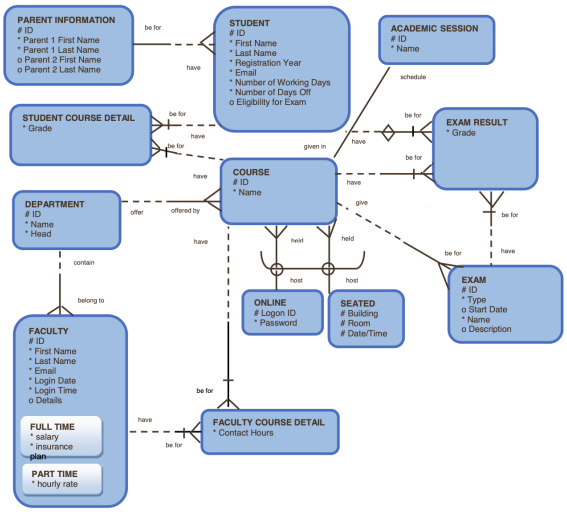
Supplier ID	Store ID	Location
SP001	S1	New York
SP001	S3	Vermont
SP002	S1	New Hampshire
SP003	S2	Rhode Island
SP004	S3	Illinois

    - The table is not in its second normal form because location is a non-UID attribute that does not depend on the entire UID. To bring the table to the second normal form, two tables should be created. One table would have supplied ID and store ID, whereas the other table would have store ID and location.
  - Task #3: 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:

Book ID	Category ID	Category Desc	Price
1	1	Cooking	$27.99
2	2	Travel	$17.99
3	1	Cooking	$20.99
4	3	Computers	$40.99
5	2	Travel	$19.99

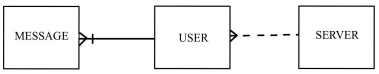
- The table is not in its third normal form because category desc is not dependent on the UID of book ID, instead it is dependent on the category ID. To bring the table to the third normal form, two tables should be created. One table would have the book ID, category ID, and price. Whereas the other table would have category ID and category desc.

- *Exercise 2:* Normalize Academic Database ERD
  - Task #1: 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.



  - Another entity should be created as student attendance for number of working days, number of days off, and eligibility for exam because it breaks the third normal form.
  - Another entity should be created as faculty log for login date and login time because it breaks the third normal form.
  - Another entity should be created as exam type for exam name and description because it breaks the third normal form.

- *Exercise 3:* Validate an ERD for Normalization
  - Task #1: Evaluate the following unnormalized data in the USER entity and develop an entity relationship diagram that is normalized to the third normal form.



  - User
    - User_ID
    - User_name
  - Message
    - Message_ID

            - Recorded_date
            - Subject
            - Text
        - Server
            - Server_ID
            - Server_name
    - Task #2: 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.

        - To bring the model to the third normal form, another entity titled car color could be created. The attributes color scheme, paint color, interior color, and care number (foreign key) could be added to this new entity.
- *Exercise 4:* Validate an ERD for Normalization
    - Task #1: 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.
            - Each book in the system must have a title, isbn, publication year, and price as well as a record of their author and publisher associated with them
        - b. For authors, the database keeps the name, address and the url of their homepage.
            - Each author in the system must have a name, address, and URL of their website homepage associated with them
        - c. For publishers, the database keeps the name, address, phone number and the url of their website.
            - Each publisher in the system must have a name, address, phone number, and URL of their website homepage associated with them.
        - d. The store has several warehouses, each of which has a code, address and phone number.

            - Each warehouse in the system must have its own unique code, address, and phone number associated with them.
        - e. The warehouse stocks several books. A book may be stocked at multiple warehouses.
            - A book can be stocked at several warehouses. The system needs to track which books are stocked at which warehouses.
        - f. The database records the number of copies of a book stocked at various warehouses.
            - A book can be stocked at several warehouses. The system needs to track the number of copies a warehouse has of a book.
        - g. The bookstore keeps the name, address, email-id, and phone number of its customers.
            - Each customer in the system must have a name, address, email-id, and phone number associated with them.
        - h. A customer owns several shopping carts. A shopping cart is identified by a Shopping_Cart_ID and contains several books.
            - Each customer can have several shopping carts. Each shopping cart must be identified by a Shopping_Cart_ID and have the ability to contain multiple 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.
            - Each shopping cart can contain more than one copy of the same book. The system needs to record the number of copies of each book for 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.
            - For each transaction to be completed, a customer needs to fill out and provide their information. The system needs to record their billing address, shopping address, shipping option, and payment information. After they complete the transaction, the system needs to send the customers an email notification about their placed order.
    - Task #2: Identify if the given description can be categorized as a Structural Business rule, Procedural Business rule or Programmatic Business rule.
        - a. All teachers in our school must possess a valid teaching certificate
            - Answer: structural business rule

        - b. Each Department must offer a Course
            - Answer: structural business rule
        - c. Approval of travel requests to an event must be signed by the project manager of the event
            - Answer: procedural business rule
        - d. A customer may make numerous payments on account
            - Answer: programmatic business rule
        - e. A machine operator may not work more than 10 hours in a day
            - Answer: procedural business rule
        - f. The Rental amount in RENTAL is calculated from the Rental rate multiplied by the number of days
            - Answer: programmatic business rule
        - g. A Customer can have zero, one or many ORDERS
            - Answer: structural business rule
        - h. The Total cost of the RENTAL is calculated from the sum of Insurance amount, Rental amount, and Late charge
            - Answer: programmatic business rule
        - i. A customer's debt must not exceed the customer's credit limit
            - Answer: programmatic business rule

Database Foundations 3-4: Data Modeling Terminology and Mapping Practices

- *Exercise 1:* Identify entities, attributes, instances and their corresponding tables, rows and columns
    - Task #1: Match the ERD elements to their corresponding database elements.

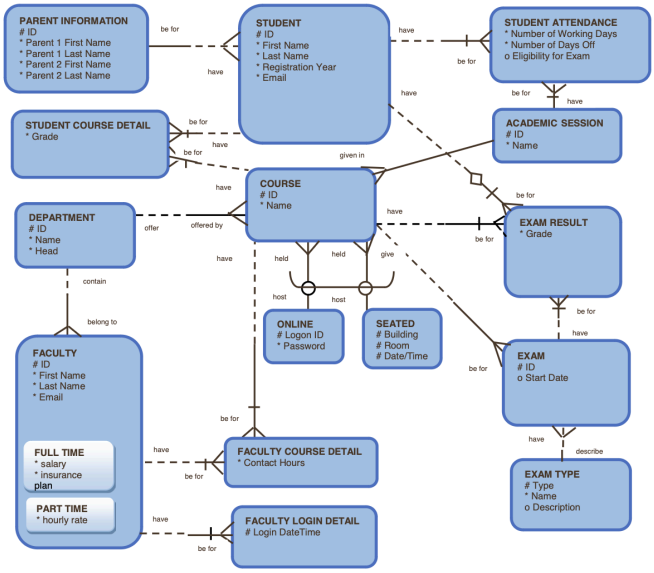
Analysis		Design	
1.	Attribute	a.	Column
2.	Entity	b.	Foreign key
3.	ER Model	c.	Physical design
4.	Instance	d.	Primary key
5.	Primary UID	e.	Row
6.	Relationship	f.	Table
7.	Secondary UID	g.	Unique key

        - Attribute = column
        - Entity = table
        - ER Model = physical design
        - Instance = row
        - Primary UID = primary key
        - Relationship = foreign key
        - Secondary UID = unique key
    - Task #2: Identify the table diagram notations listed below.
        - Identify the table diagram notations listed below.
            - a. pk: primary key
            - b. fk: foreign key

        - c. uk: unique key
        - d. *: not null (mandatory)
        - e. o: optional (nulls allowed)
    - Task #3: Create short names for the terms below based on the naming conventions rules.
        - a. Authors: ATR
        - b. Publishers: PLR
        - c. Customers: CTR
    - Task #4: The goal of this practice is to recognize attributes for an entity. 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	
			Title
			Description
			Venue
			First Name
			Phone Number
			Release date
			Last Name
			Type
			Email address

        - Song: title, release date, type
        - Event: title, description, venue, type
        - Customer: first name, phone number, last name, email address
- *Exercise 2:* Mapping the Academic Database
    - Task #1: With the ERD provided below, map the entities, attributes and UIDs to tables, rows, and keys using a table diagram as shown:



- PARENT_INFORMATION

Key Type	Optionality	Column name
pk	*	parent_ID
	*	first_name _parent_1
	*	last_name _parent_1
	*	first_name _parent_2
	*	last_name _parent_2
fk	*	student_ID

- STUDENT

Key Type	Optionality	Column name
pk	*	student_ID
	*	first_name
	*	last_name
	*	registration_year
	*	email

- STUDENT_ATTENDANCE

Key Type	Optionality	Column name
	*	nmbr_working_days
	*	nmbr_days_off
	o	exam_elgibility
pk, fk	*	student_ID
pk, fk	*	session_ID

- STUDENT_COURSE_DETAIL

Key Type	Optionality	Column name
	*	grade
pk, fk	*	student_ID
pk, fk	*	course_ID

- COURSE

Key Type	Optionality	Column name

pk	*	course_ID
	*	name
fk	*	department_ID
fk	*	session_ID
fk	o	online_ID
fk	o	seated_ID

- ONLINE

Key Type	Optionality	Column name
pk	*	logon_ID
	*	password
fk	*	course_ID

- SEATED

Key Type	Optionality	Column name
pk	*	building
pk	*	room
pk	*	date_time

- ACADEMIC_SESSION

Key Type	Optionality	Column name
pk	*	session_ID
	*	name
fk	*	student_ID

- DEPARTMENT

Key Type	Optionality	Column name
pk	*	department_ID
	*	name
	*	head

- FACULTY

Key Type	Optionality	Column name
pk	*	faculty_ID

	*	first_name
	*	last_name
	*	Email
	o	full_time_ID
	o	part_time_ID

- FACULTY_FULL_TIME

Key Type	Optionality	Column name
pk	*	full_time_ID
	*	salary
	*	insurance_plan
pk, fk	*	faculty_ID

- FACULTY_PART_TIME

Key Type	Optionality	Column name
pk	*	part_time_ID
	*	hourly_wage
pk, fk	*	faculty_ID

- FACULTY_COURSE_DETAIL

Key Type	Optionality	Column name
pk	*	faculty_course_dtl_ID
	*	contact_hours
fk	*	faculty_ID
fk	*	course_ID

- FACULTY_LOGIN_DETAIL

Key Type	Optionality	Column name
pk	*	login_date_time
fk	*	faculty_ID

- EXAM

Key Type	Optionality	Column name
pk	*	exam_ID

	o	start_date
fk	*	course_ID

- EXAM_RESULT

Key Type	Optionality	Column name
	*	grade
pk, fk	*	student_id
pk, fk	*	exam_id
pk, fk	*	course_id

- EXAM_TYPE

Key Type	Optionality	Column name
pk	*	exam_type
	*	name
	o	description
pk, fk	*	exam_ID