CSc 460 — Database Design
Fall 2025 (McCann)

http://www2.cs.arizona.edu/classes/cs460/fall25/

Program #4: Database Design and Implementation

Due Dates:

Team Members:	November 24th, 2025, at the beginning of class
Draft E–R Diagram:	December 1st, 2025, at the beginning of class
Final Product:	December 8th, 2025, at the beginning of class

Designed by *Utkarsh Upadhyay and Jianwei (James) Shen*

Overview: In this assignment, you will build a database–driven information management system from ground up. We will give you an application domain to work on. Your goal is to design the underlying database and define the application functionalities you will provide with the database, and implement this application using Oracle within a text–based JDBC program.

Assignment: In this assignment you are to implement a two–tier client–server architecture.

1. **Database Back-End**, which runs the Oracle DBMS on `aloe.cs.arizona.edu`. Your job is to design the database relational schema, create tables and populate your tables with some initial data. We are requiring that you create an E–R diagram, analyze the FDs of each table and apply table normalization techniques to your schema to justify that your schema satisfies 3NF, and, if possible, BCNF.

2. **JDBC Front–End**, which is the client's user interface. You need to design a text–based application that appropriately handles all the required functionalities. Your client application will run on lectura.

Application Domain: The problem description for the project is as follows:

> As a pet cafe striving to create a welcoming and enriching environment for both animals and visitors, we have developed a unique space where people can enjoy quality food and beverages while interacting with adoptable and resident pets. Our mission is twofold: to provide a relaxing cafe experience and to facilitate pet adoptions by allowing potential adopters to spend meaningful time with animals in a comfortable setting. To achieve this, we need a comprehensive system that manages customer visits, pet welfare, adoption processes, and cafe operations seamlessly.
>
> The pet cafe houses a variety of animals, including cats, dogs, rabbits, and other small animals, each with unique needs and temperaments. Not all pets are available for adoption—some are permanent residents that serve as cafe ambassadors, while others are rescue animals seeking forever homes. Before any pet can interact with visitors, they must undergo health screenings and behavioral assessments to ensure safety for both the animals and customers.
>
> The cafe is divided into multiple rooms and areas to accommodate different types of animals and experiences. For example, we maintain separate spaces for cats, dogs, and small animals to minimize stress and ensure appropriate interactions. Each room has a maximum capacity to prevent overcrowding and maintain a calm environment. Some areas are designated for adoption consultations, while others are purely for recreational visits.

(Continued...)

Before visiting, customers must register as members of our cafe. For each member, we collect essential information including name, telephone number, email address, date of birth, and emergency contact. We offer different membership tiers — such as day passes, monthly memberships, and annual memberships — each providing different benefits like priority booking, discounts on food and merchandise, and unlimited visits. Once registered, members can make reservations for specific time slots and rooms, ensuring a quality experience without overcrowding.

The reservation system is central to our operations. Customers book visits for specific dates, times, and durations (typically 1-2 hours). Each reservation is linked to a particular room, and the system must enforce capacity limits. Upon arrival, customers check in, and when they leave, they check out. During their visit, customers can order food and beverages from our cafe menu, which includes coffee, tea, pastries, light meals, and specialty drinks. All orders are tracked and linked to the customer's visit for billing and inventory purposes.

Pet welfare is our top priority. We maintain detailed health records for every animal, documenting vaccinations, veterinary checkups, grooming sessions, feeding schedules, and behavioral observations. Our staff includes specialized roles such as pet handlers, veterinary staff, baristas, adoption coordinators, and managers. Veterinary staff must ensure all pets are up-to-date on vaccinations and receive regular health monitoring. Pet handlers manage daily care and supervise interactions between customers and animals.

The adoption process is carefully managed through our system. When a customer expresses interest in adopting a pet, they submit an adoption application. An adoption coordinator reviews each application and conducts follow-up interviews if necessary. Applications can be approved, rejected, or withdrawn. Once an adoption is approved and finalized, the system records the adoption details including the date, adoption fee, and follow-up schedule.

To enhance community engagement and education, the cafe regularly hosts events and classes. These may include pet care workshops, training sessions, adoption events, and birthday parties. Each event has a scheduled date and time, takes place in a designated room, and has a maximum capacity. Customers can register for events in advance, and the system tracks attendance and payment status for each booking.

The cafe's stakeholders want to monitor business performance by tracking revenue from various sources including membership fees, food and beverage sales, event bookings, and adoption fees. They also want to analyze customer behavior, pet popularity, and operational efficiency to make data-driven decisions about staffing, inventory, and facility improvements.

This description does not describe every detail. These are the essentials; we expect that your team will create logical and conceptual designs that incorporate all of these features, at minimum. You are free to add additional details that you feel are appropriate.

For each table you create, you need to populate a reasonable number of tuples in order to test your queries adequately. Some data basics are provided in the application domain description; the rest are left for you to determine, based on your needs. (What is 'reasonable' is difficult to define; a few dozen tuples per relation certainly would be; just a handful per relation may not provide sufficient variety.)

We realize that you are not an expert in this domain, but you have dealt with similar organizations in your life. Hopefully, you have enough experience that this problem description makes sense. If you have questions, please ask, and the TAs will help you clear things up.

(Continued...)

Required functionalities: Within the framework provided above, your system is expected to perform examples of the following operations:

1. *Record insertion*: Your application should support inserting a new data record via a JDBC interface.
2. *Record deletion*: Your application should support deleting an existing data record via a JDBC interface.
3. *Record update*: Your application should support updating an existing data record via a JDBC interface.
4. *Queries*: Your application should support querying your database via a JDBC interface for the problem description given above. You are required to implement the three provided queries as well as at least one query of your own design. Details are provided below.

Specifically, the JDBC application's interface should enable users to:

1. **Add, update, or delete a member:** The member table contains essential personal details, including name, phone number, email address, date of birth, and emergency contact. Each member must be assigned a unique member ID upon registration.

   The system must support updating a member's contact details and membership tier at any time.

   Prior to deletion, the system must verify that the member does not hold any active reservations, pending adoption applications, or unpaid food orders. Once all checks pass, the system must remove all related records associated with the member.

2. **Add, update, or delete a pet:** The pet table stores essential information including name, species, breed, age, date of arrival, temperament, special needs, and current status (available for interaction, in care, available for adoption, or adopted). Each pet is assigned a unique pet ID upon intake.

   The system must support updating pet information as circumstances change—such as age updates, temperament adjustments, or status changes.

   Pets cannot be deleted arbitrarily. Deletion is only allowed if the pet has been permanently adopted and all follow-ups are complete, or in the case of a pet passing away. Before deletion, the system must verify that the pet has no pending adoption applications or active health records requiring attention.

3. **Add, update, or delete a food & beverage order:**

   The food and beverage order table tracks all cafe purchases, including order ID, customer ID, reservation ID (if applicable), order time, items ordered, total price, and payment status.

   Updates are necessary when order modifications occur or when payment status changes. The system should support applying membership discounts automatically.

   Orders may only be deleted if created in error and no items have been prepared or delivered.

4. **Add, update, or delete a reservation:**

   The reservation table manages customer bookings, including reservation ID, customer ID, room ID, reservation date and time, duration, check-in status, and check-out status.

   When a customer makes a reservation, the system must verify room availability and enforce capacity limits. Updates may occur for rescheduling or extending visit duration. The system must also update check-in and check-out status as customers arrive and leave.

   Reservations may only be deleted if canceled before the scheduled time and no food orders have been placed for that visit.

(Continued...)

5. **Add, update, or delete a health record entry:**

   The health record table logs all medical and care information for each pet, including record ID, pet ID, employee ID (vet staff), record date, record type (vaccination, checkup, feeding schedule, grooming, behavioral note), description, and next due date.

   Updates may occur to correct information or add follow-up notes. The system should maintain a complete audit trail of all changes.

   Health records cannot be deleted as they form a legal and medical history for each animal. If an entry was made in error, it should be marked as "void" or "corrected" with an explanatory note.

6. **Add, update, or delete an adoption application:**

   The adoption application table manages customer requests to adopt pets, including application ID, customer ID, pet ID, application date, status (pending, approved, rejected, withdrawn), and assigned employee ID (adoption coordinator).

   Updates occur as the application moves through the review process. Coordinators can update the status as they evaluate the applicant.

   Applications may be deleted only if submitted in error and no review has begun. Otherwise, they should be marked as "withdrawn" to maintain application history.

7. **Add, update, or delete an event booking:**

   The event booking table tracks customer attendance at cafe-hosted events, including booking ID, customer ID, event ID, booking date, attendance status (registered, attended, no-show, canceled), and payment status.

   When a customer registers, the system must verify event capacity. Updates occur when customers check in or cancel their registration.

   Bookings may be deleted only if canceled well in advance and payment has been refunded. Otherwise, cancellations should be marked to maintain accurate attendance records.

Here are the queries that your application is to be able to answer:

1. For a given pet, list all adoption applications submitted for it, including applicant name, application date, current status, and the assigned adoption coordinator's name.
2. For a given customer, show their complete visit history including reservation dates, rooms visited, food orders placed during each visit, total amount spent, and membership tier at the time of visit.
3. List all upcoming events that have available capacity, including event name, date/time, room location, current number of registered attendees, maximum capacity, and the employee coordinating the event.
4. One additional non–trivial query of your own design, with these restrictions:
   - The question must use more than two relations.
   - It must be constructed using at least one piece of information gathered from the user.

Working in Groups: In industry, such a project is usually the work of multiple developers, because it involves several different components. Good communication is key to the success of the project. This assignment provides an opportunity to practice just this sort of teamwork. Therefore, we are accepting team sizes of two, three, or four members. Working solo is not permitted.

Early on, you will need to agree on a reasonable workload distribution plan for your team, with well–defined responsibilities, deliverables, and expected completion dates. Such a plan will minimize conflicts and debugging effort in the actual implementation.

Don't have a lot of experience working on group projects and would like some pointers? The following page has a short list of useful suggestions:

https://www.saintleo.edu/about/stories/blog/7-tips-to-more-effectively-work-on-group-projects

(Continued...)

Late days: Late days can be used on this assignment, but only on the third due date. How many a team has to use is determined as follows: Team members total their remaining late days, and divide by the number of members in the team (integer division), producing the number of late days the team has available, **to a max of two days late**. (Why? The TAs need to get grading done soon after the due date, you need time to study for your final exams, and the department has a rule about assignments needing to be due before the start of finals.)

For example, a team whose three members have 1, 1, and 3 late days remaining have $\lfloor \frac{1+1+3}{3} \rfloor = 1$ late day to use, if needed.

Hand In: Here are the 'deliverables' for each of the assignment's three due dates:

1. *Team Composition*: By the first due date (see the top of the front page of this handout), one member of your team must add a new row to the shared Google Sheet with the names and email addresses of all team members. The link to the Google Sheet has already been posted to Piazza. Failure to do so by the start of class on this date will cost your team the corresponding points listed in the Grading Criteria section (below).

2. *E–R Diagram*: As stated in the Assignment section, your team will need to create an E–R diagram that describes your database design. Before the second due date, your team will need to prepare a draft of your E–R diagram **and** a member of your team will need to submit it through `turnin` to the `cs460p4` folder. The purpose of this requirement is to allow the TAs to review your schema and make suggestions for improvement. The sooner you create your design and discuss it with the TAs, the more time you will have to refine your final E–R diagram. If TAs need further explanation of your E–R Diagram, they'll send out an email to make an appointment to have an additional meeting.

3. *Final Product*: On or before the third due date, a member of your team must submit a `.tar` file of your well–documented application program file(s) via turnin to the folder `cs460p4`. The tar file should contain all of the following:
   (a) The source code for your application.
   (b) A PDF file called "design.pdf" containing the following sections in this order:
      i. *Conceptual database design*: Your final E–R diagram along with your design rationale and any necessary high–level text description of the data model (e.g., constraints, or anything you were not able to show in the E–R diagram but that is necessary to help people understand your database design).
      ii. *Logical database design*: The conversion of your E–R schema into a relational database schema. Provide the schemas of the tables resulting from this step.
      iii. *Normalization analysis*: For each of your entity sets (tables), provide all of the FDs of the table and justify why your the table adheres to 3NF / BCNF.
      iv. *Query description*: Describe your self–designed query. Specifically, what question is it answering, and what is the utility of including such a query in the system?
   (c) A `ReadMe.txt` describing:
      i. Compilation and execution instructions, to enable the TAs to execute your application and exercise the required functionalities.
      ii. The workload distribution among team members (that is, which people were responsible for which parts of the project).

   In addition, each team must schedule a time slot (∼15 – 20 minutes) to meet with a TA, demonstrate your system, and answer some questions about it. Closer to the final due date, we will let you know how to sign up.

(Continued...)

Grading Criteria: Total: 100 points

1. Team Composition (1st due date): 5
2. Complete E–R Diagram Draft (2nd due date): 20
3. Final Submission (3rd due date): 75
   (a) Coding / Implementation: 55
      - Documentation 15
      - Style and organization 10
      - Record insertion: 5
      - Record deletion: 5
      - Record update: 10
      - Record query: 10

   (b) Database design: 20
      - Final E–R diagram: 10
      - Normalization analysis: 10

Grading Notes:

1. Unless we receive verifiable complaints about inadequate contributions, each member of a team will receive the same score on this assignment.
2. We won't put much weight at all on the appearance of the text application; concern yourselves with the application's design instead. The main point of the assignment is the DB design and how well it supports the use cases (the DML operations and queries, in particular).