Department of Electrical and Computer Engineering
University of Puerto Rico
Mayagüez Campus

CIIC 4060/ICOM 5016 – Introduction to Database Systems
Spring 2024

Term Project – Hotel Analytics Systems
Phase I – Database and ETL Due
Date: March 15, 2024

Objectives

1. Understand the design, implementation and use of an application backed by a database system.
2. Understand the use of table diagram for database application design.
3. Gain experience by implementing applications using layers of increasing complexity and complex data structures.
4. Understand the design, implementation and use of a data pipeline using the ETL concept.
5. Gain overall knowledge with OLAP systems.

Overview

You will design, implement, and test the backend of an application used to manage hotels reservations. The data in the application is managed by a relational database system and exposed to client applications through a REST API. You will build the database application and REST API using **Flask**, which form the backend of the system. Your database engine must be **PostgreSQL**, and you must implement the code in Python. The backend site will provide the user with the features specified in this document. In addition, your solution will provide a Web-based dashboard using the tool provided on a later phase indicating relevant statistics that are also specified below.

Your solution MUST follow the Model-View-Controller Design Pattern. In this scheme, your solution will be organized as follows:

1) View – application pages will handle all interaction with the users and will show results from operations performed on the database. This is the client code for the application. The client **MUST NOT** interact directly with the database. They must talk through the REST API.

2) Controller – **Python** objects will act as controllers. Each object will get a request, create a business service object to handle the request, collect the results from the methods in this business service object and forward the results to the client using JavaScript Object Notation (JSON).

3) Model – a set of business service objects that implement all tasks and access to the database system. **You cannot use ORM APIs for this layer**. **If your team uses ORM you will get an automatic 0 in the project.**

You will host the application and the database in **Heroku**. This database will be used as your production database; thus, you cannot upload random or dummy data into the system. **Points can be deducted if the data does not match up**. To prevent this, **you will use a Docker container for any testing or data manipulation**.

Phase I:

For this phase you will be given raw data from different sources that will be used for the project database. This process is called a data pipeline, you **extract** data from different areas, **transform** them, and later **load** them to the destination (ETL). The three important elements in ETL are:

Extract – Getting data from the source(s) (e.g. CSV, DBMS, XML, JSONs, others).

Transform – Modify and change the extracted data.

Load – Save the data.

You will create the tables in the database hosted in Heroku of the given Table Diagram. Keep in mind that attributes name, datatypes, relationships, and primary and foreign keys must match with those in the Table Diagram**. It is of utmost importance that you do not change the primary and foreign keys, if any of these values are changed you will lose points for this phase**.

If a row has an empty or null value, or the row is fully empty, you must discard it. Only full records need to be uploaded into the database. Keep in mind the following details about the given data and how the application is expected to work:

Details:

1. The room price is per day.
2. There are three positions for employees: regular, supervisor, and administrators:
    a. Regular: Works for a chain, has no access to any information in the dashboard.
        i. Can make a reservation for the clients. ii.
        Can view local statistics of their hotel. iii.
        Salary: $18,000-$49,999
    b. Supervisor:
        i. Can view local statistics of other hotels of the same chain. ii. Only
        supervisors can make a room unavailable when it is not reserved. iii.
        Salary: $50,000-$79,999
    c. Administrator:
        i. Has access to all local and global statistics.
        ii. The hotel and chain they worked for is a generic dummy value; do not remove. iii.
        Salary: $80,000-$120,000
3. The rooms' descriptions are divided as follow:

a. Standard:
    i. Capacity: 1
    ii. Types: Basic or Premium.
b. Standard Queen:
    i. Capacities: 1 or 2
    ii. Types: Basic, Premium, or Deluxe.
c. Standard King:
    i. Capacity: 2
    ii. Types: Basic, Premium, or Deluxe.
d. Double Queen:
    i. Capacity: 4
    ii. Types: Basic, Premium, or Deluxe.
e. Double King:
    i. Capacity: 4 or 6
    ii. Types: Basic, Premium, Deluxe, or Suite.
f. Triple King:
    i. Capacity: 6
    ii. Types: Deluxe or Suite.
g. Executive Family:
    i. Capacity: 4, 6, or 8
    ii. Types: Deluxe or Suite.
h. Presidential:
i. Capacity: 4, 6, or 8 ii. Type: Suite.

4. Clients can only reserve a room with equal or less guests than the room's capacity.
    a. E.g. 1-4 guests for an executive family suite with a capacity of 4.
5. The seasons markups are divided by the reservation start date's months:
    a. Spring: March-May
    b. Summer: June-August
    c. Fall: September-November
    d. Winter: December-February
6. The membership discount is based on the years a client has been member:
    a. 1-4 years: 2%
    b. 5-9 years: 5%
    c. 10-14 years: 8%
    d. Over 15 years: 12%
7. The reservation cost is calculated by using the room priced times the number of days reserved times the season markup of the hotel's chain less the membership discount.
8. There are 5 different payment methods: cash, check, credit card, debit card, and pear pay.

Local Statistics

9. Top 5 handicap rooms that were reserved the most.

10. Top 3 rooms that were the least time unavailable.
11. Top 5 clients under 30 years old that made the most reservation with a credit card.
12. Top 3 highest paid regular employees.
13. Top 5 clients that received the most discounts.
14. Total reservation percentage by room type.
15. Top 3 rooms that were reserved that had the least guest-to-capacity ratio.
    a. E.g. Room holds 8 guests, but the average amount of guests was 4, the ratio is 50%.

Global Statistics

16. Top 5 hotel chain with the highest total revenue.
17. Total reservation percentage by payment method.
18. Bottom 3 hotel's chain with the most rooms.
19. Top 5 hotels with the most client capacity.
20. Top 10% of the hotels that had the most reservations.
21. Month with the most reservation.

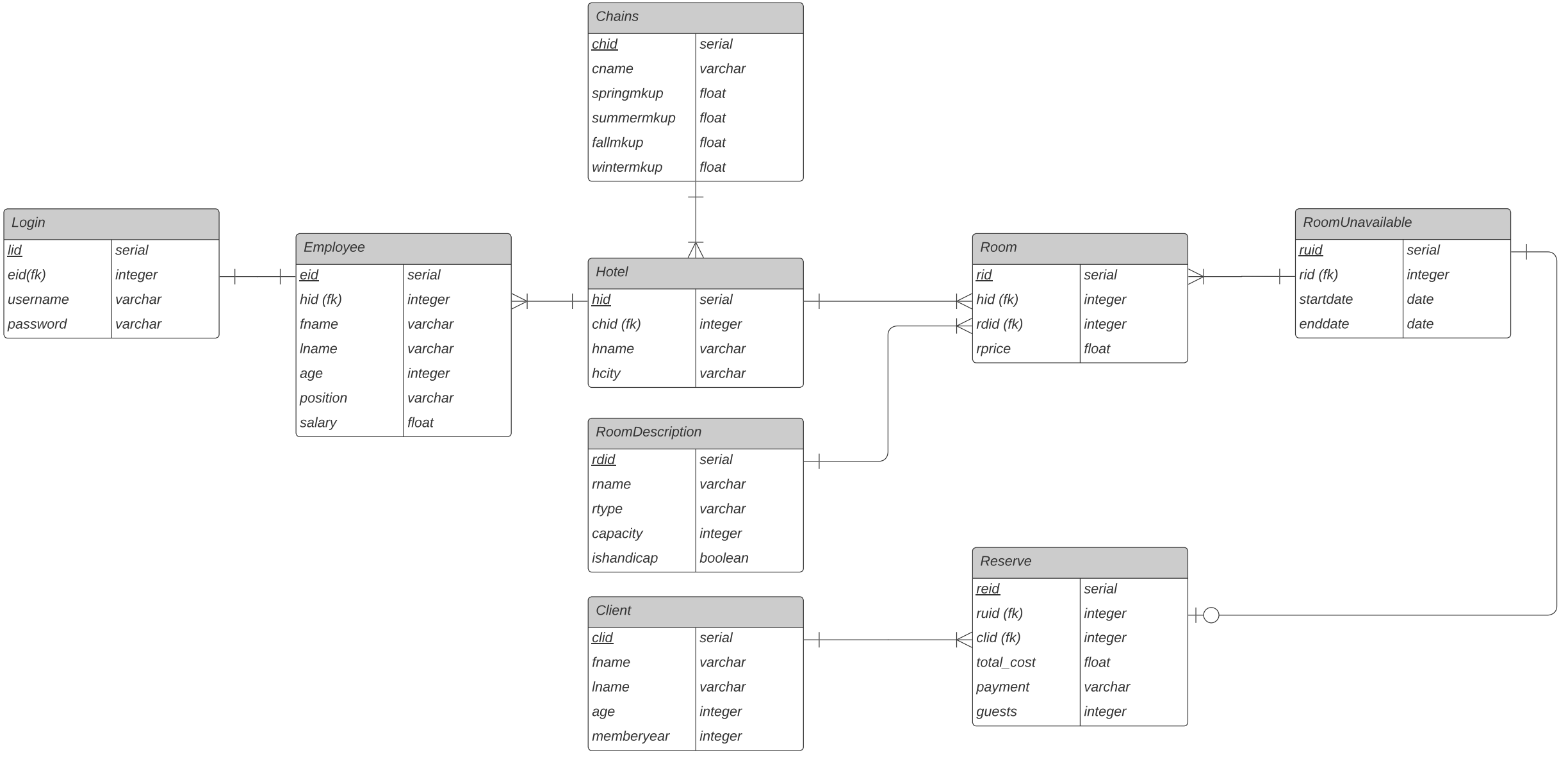
*More Statistics could be added in later phases.

Note: Error handling is required for the entire project. Some error handlings include, user cannot access specific statistics, reserving a room that is unavailable, negative values in revenue, etc.

You are required to use GitHub to manage and submit all phases' documents and code. You will be given access to a GitHub classroom link for this purpose.

Deliverables for Phase I

1) Postgres database hosted in Heroku.
2) Database with schemas and data implemented.
3) You will use the repo provided by classroom to submit the following:
    a. Code used for the ETL, inside a folder named 'ETL'.
    b. ReadMe with the Database credentials.


Chains
chid serial
cname varchar
springmkup float
summermkup float
fallmkup float
wintermkup float
Login
lid serial
eid(fk) integer
username varchar
password varchar
Employee
eid serial
hid (fk) integer
fname varchar
lname varchar
age integer
position varchar
salary float
Hotel
hid serial
chid (fk) integer
hname varchar
hcity varchar
Room
rid serial
hid (fk) integer
rdid (fk) integer
rprice float
RoomUnavailable
ruid serial
rid (fk) integer
startdate date
enddate date
RoomDescription
rdid serial
rname varchar
rtype varchar
capacity integer
ishandicap boolean
Client
clid serial
fname varchar
lname varchar
age integer
memberyear integer
Reserve
reid serial
ruid (fk) integer
clid (fk) integer
total_cost float
payment varchar
guests integer