MIDDLE EAST TECHNICAL UNIVERSITY

NORTHERN CYPRUS CAMPUS

CNG 352: DATABASE MANAGEMENT SYSTEMS

Term Project Step – 4
“Alacritas”

Name of Student#1: Temirlan Bayeshov

ID Number: 2041226

Name of Student #2: Anıl Peker

ID Number: 2016129

Table of Contents

Requirement Analysis

A family getaway, self-adventure, romantic trip, expeditions or weekend run away are all have connection with traveling. Traveling to a new country involves new experience and fun adventures but sometimes it can lead to un-expectable cases like a long flight can you risk of sick, or new food can cause stomach problems and etc. That's why depending on our traveler experience we decided to make "Alacritas" application, which will be helpful for tourist abroad to find a best suitable treatment in foreign Hospitals. Our application will contain hospital information, ratings, available doctors, etc. Hospitals will divided according to theirs type like branches, public, main and research hospitals. Doctors also will be divided according their professions. System will provide comfortability and ease of usability both for clients and for hospitals.

Data Requirements

User

User will be stored in database and will control trip information that will be used for future statistical functions. System will show best suitable hospitals to user depends on his/her given information, also it will show rates and comments about given hospitals, so user will chose the best suitable hospital depends on given information. All previous trips will be stored for a statistical collection. In addition user will be allowed to rate and comment facility that he/she visited.

Necessary Attributes:

User entity will store necessary data about person who will travel. It will contain user ID, gender, number of trips, username and password. User ID will be primary key, because it should be unique for each user. Number of trips will be derived data.

Trip

Depending on users given data, trip will search for best hospital in the address and other information provided by user. Trip will be shown in specific date ranges to check availability of hospitals. Each trip data will be stored for each different user. For each trip to search best suitable facility data will be collected both from user and hospitals.

Necessary Attributes:

The data required here is trip id, starting-ending dates and location to where user will travel. Depends on location user will be allow to see which hospitals he/she can visit during the trip. Trip id will be primary key in the database table.

Country/City (Location)

Before planning a trip user will give necessary information on location he/she will visit. User will provide country and specific city where adventure will start. In case of visiting different city or countries during the trip, user will be allowed to enter multiple addresses. Depending on this data system will list suitable hospitals in the given location.

Necessary Attributes:

Trip entity will consists of countries entity, where country name will be stored. Country name will be unique here.

City entity which connected to country entity will require data like city name and city id. City id will be primary key, while city name will be normal attribute.

Hospital

Hospital will be divided on different types, depends on user choice like price, rating and comments. After visiting a specific hospital, user will rate and comment a visited facility. So, hospital here may change in ranks among another hospitals in the given location. After performing statistical function, they will be stored according their ranks.

Necessary Attributes:

Hospital entity will store data like hospital id, hospital name, and rating and foundation year. Hospital id will be unique and rating will be derived attribute which will be changed depends on user rating options.

Hospital will be divided into Branch, Research Hospital, Public Hospital and Private Hospital. Disjoint will be used here. In addition to Hospital data Public Hospital type will store department and Research Hospital entity will store research topic and number of researches

Rating

Each hospital will be have ranks. According theirs rating and comments which given by a visited customers they will be shown in the system in priority lists. Ranking will be calculated according to customer service, comfortability and another necessary information.

Doctor

System will keep track of different types of doctors depending on users' choice. Frequently visited doctors in specific hospitals will be shown in the system. In addition doctors known languages will be shown to the user, so user will be allowed easily visit specialist without any language barriers. Also user will be allowed to see doctors' years of experience, so before visiting a specific specialist user will have no doubts about it.

Necessary Attributes:

Data required here is employee id, age, first and last name, years of experiences and multivalued attribute languages that doctors know. Employee id will be primary key. Doctor will consists of two types: Practitioner or Surgeon. Disjoint diagram will be used here. In addition to doctor data, surgeon will store data like specialty and number of surgeries surgeon done.

Admin Transaction Requirements

Data Integration

- Enter data of new country
- Enter data of new city
- Enter data of new practitioner
- Enter data of new surgeon
- Enter data of new branch
- Enter data of new research hospital
- Enter data of new public hospital
- Enter data of new private hospital

Data Update/Delete

- Update/delete data of integrated trip
- Update/delete data of integrated country
- Update/delete data of integrated city
- Update/delete data of integrated practitioner
- Update/delete data of integrated surgeon
- Update/delete data of integrated branch
- Update/delete data of integrated research hospital
- Update/delete data of integrated public hospital
- Update/delete data of integrated private hospital

Admin View Queries

- List all of the trips' info according to trip dates that all user added to system.
- List all of the travelers' info according to their IDs.
- List all of the hospitals' info according to their rates.
- List all of the countries' info on the alphabetically.
- List all of the cities' info on the alphabetically.
- List all of the doctors' info on the alphabetically.
- Filter all the hospitals' info which are below than 1 point ranking which is out of 5.
    - Delete after 50 ranking which are comes from users, if hospital's rank is still below than 1 out of 5.
- List all countries which have more than 50 hospital and average rank of these hospitals have greater than 3.

User Transaction Requirements

Data Integration

- Enter data of new user.
- Enter data of new trip.

Data Update

- Update data of integrated rate of a hospital.
- Update data of own user info.

User View Queries

- List all of the trips' info according to trip dates that he/she added to system. List all of the hospitals' info according to their rates.
- Filter hospitals according to their rates or departments. (Example).
    - Filter all of the hospitals according to the rate which is greater than 4.
    - Filter all of the hospitals according to department which is cardiology.
- List all of the countries' info on the alphabetically.
- List all of the cities' info on the alphabetically.
- List all of the doctors' info on the alphabetically.
- Filter all of the surgeons' info according to their specialty. (Example).
    - Filter all the surgeons according to their specialty which is cardiologist.

Enhanced Entity Relationship Diagram

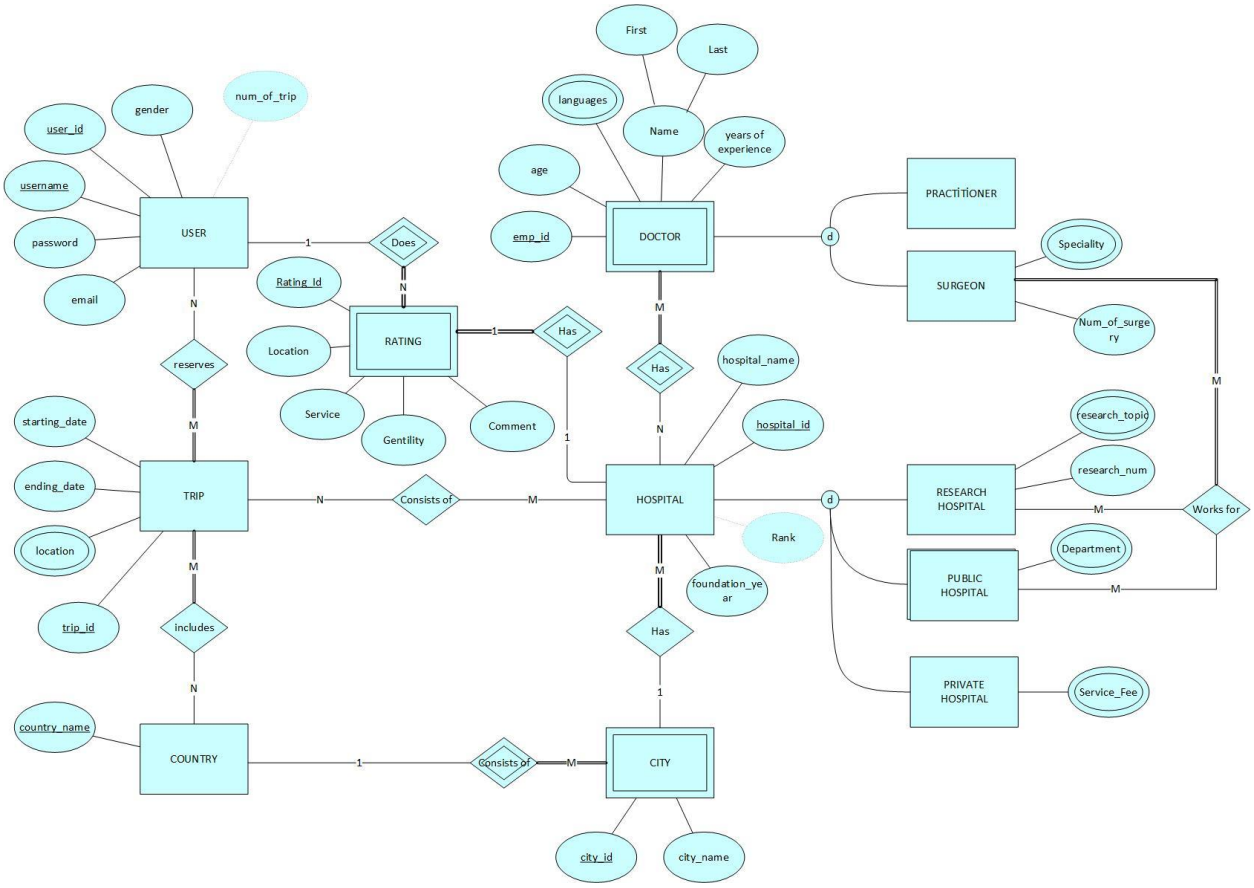


Figure 1:EER Diagram

Assumptions:

1) Number of trips attribute in User entity is derived data. All new trips will added automatically.
2) Country entity has name which is primary key. No another countries with same name allowed.
3) City entity has city id which is primary key. Name of cities in different countries can be same.
4) Hospital will have rank which is derived attribute. Rank will be counted depends on rating attributes.
5) Doctor entity will have language multivalued attribute. Customer will be allowed to know if doctor knows his/her language.
6) Rating is weak entity, cannot exists without user and hospital
7) Private Hospitals will have own payment methods
8) Practitioner doctors' difference with surgeon, in that practitioner don't have any special area of specialty.

Mapping

User Table

user_id	username	password	gender	num_of_trip
U1000	apeker	199523	male	3
U1001	tbayeshov	199454	male	8
U1002	tzaferler	taicho2	female	5
U1003	ccbalci	malatya44	male	6

Trip Table

trip_id	location	starting_date	ending_date
T1000	Moscow	12.06.2019	20.06.2019
T1001	Milano	09.07.2019	02.08.2019
T1002	Munich	13.07.2019	20.07.2019
T1003	London	16.07.2019	15.08.2019

Country Table

country_name
Afghanistan
Albania
Algeria
Andorra
Angola
…
Argentina

Assumption: All the country information will be initialized by admin.

City Table

city_id	city_name
C1000	Kabul
C1001	Mazar-I Sharif
C1002	Kandahar
…	…
C1004	Shurugwi

Assumption: All the city information will be initialized by admin.

Rating Table

Location	Service	Gentility	Comment
3	4	5	
2	5	4	Location is bad
4	2	1	They are so rude
5	5	5	Perfect hospital

Assumption: User doesn't have to write a comment to do evaluation.

Hospital Table

Hospital_id	Hospital_name	Rank	Foundation_year
H1000	Hacettepe University	4	1967
H1001	Howard University	3	1862
H1002	The Private Clinic Manchester	2	1986
H1003	Dr. Burhan Nalbantoğlu Devlet Hastanesi	3	1978
H1004	Cengiz Topel Hatanesi	3	1975
H1005	Dikmen Policlinic	2	1985
…	…	…	…
H1999	Central Manchester University Hospitals	4	1793

Assumption:

- All the hospital information except "rank" will be initialized by admin.
- Rank will be derived from some other table.
- This is an example table and the data of "rank" of hospitals taken from google evaluation.

Research Hospital

Hospital_id	Research_topic	Research_num
H1000	Cardiology, Cloning, General Surgery	3
H1001	Surgery with 5G technologies	1
…	…	…
H1999	heart transplant, face transplant	2

Assumption:

- All the hospital information except “rank” will be initialized by admin.
- Rank will be derived from some other table.
- This is an example table and the data of “rank” of hospitals taken from google evaluation.

Public Hospital Table

Hospital_id	Deparment
H1003	General Surgery,Eye
H1004	internal medicine

Assumption:

- All the hospital information except “rank” will be initialized by admin.
- Rank will be derived from some other table.
- This is an example table and the data of “rank” of hospitals taken from google evaluation.

Private Hospital Table

Hospital_id	Service Fee
H1002	200$, 50$, 10$

Assumption:

- All the hospital information except “rank” will be initialized by admin.
- Rank will be derived from some other table.
- This is an example table and the data of “rank” of hospitals taken from google evaluation.

Branch Table

Hospital_id
H1005

Doctor Table

Emp_id	age	Languages	Firstname	Lastname	Year of experience
D1000	**43**	**Turkish,English**	**Ahmet**	**Türkmen**	**19**
D1001	**26**	**Turkish,English**	**Hasan**	**Ay**	**2**

Surgeon Table

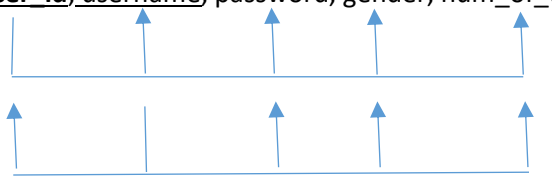
Emp_id	Speciality
D1000	**Cardiologist, Urologist**

Practitioner Table

Emp_id
D1001

Functional Dependency

User(**user_id**, username, password, gender, num_of_trip)



Trip(**trip_id,** location, starting date, ending_date)

Country(**country_name**)

City(**city_id**, city_name, country_name(FK:Country))

Hospital(**hospital_id(PK),**hospital_name, rank, foundation_year, city_id(FK:city))

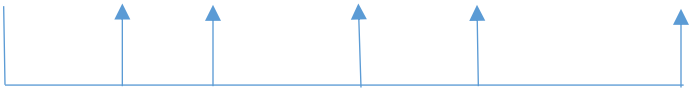
Research_Hospital(**hospital_id(FK: Hospital)**, research_topic, research_num)

Public_Hospital(**hospital_id(FK: Hospital),** department)

Public_Hospital_Dept(**hospital_id(FK: Hospital),** Service Fee)

Branch(**hospital_id(FK: Hospital)**)

Doctor(**emp_id(PK),** age, languages, firstname, lastname, years_of_experience)



Surgeon(**Emp_id**(FK:Doctor), Speciality, Year of experience)

Practitioner(**Emp_id(FK:Doctor)**)

Reserves(**user_id(FK:User), trip_id(FK:Trip)**)

Consits_of(**trip_id(FK:Trip)**, **hospital_id(FK:Hospital)**)

Includes(**trip_id(FK:Trip)**, **country_name(FK:Country)**)

Has(**hospital_id(FK: Hospital)**, **emp_id(FK:Doctor)**)

WorksFor(**hospital_id(FK: Hospital)**, **emp_id(FK:Doctor)**)

Normalization

User

1NF:

User{user_id(PK),username(PK),email,password,gender,num_of_trip}

2NF:

User{user_id(PK),username(PK),email,password,gender,num_of_trip}

3NF:

User{user_id(PK),username(PK),email,password,gender,num_of_trip}

4NF:

User{user_id(PK),username(PK),email,password,gender,num_of_trip}

Trip

1NF:

Trip{trip_id(PK),starting_date,ending_date }

Trip_location{trip_id(FK:Trip),location(PK)}

2NF:

Trip{trip_id(PK),starting_date,ending_date }

Trip_location{trip_id(FK:Trip),location(PK)}

3NF:

Trip{trip_id(PK),starting_date,ending_date }

Trip_location{trip_id(FK:Trip),location(PK)}

BCNF:

Trip{trip_id(PK),starting_date,ending_date }

Trip_location{trip_id(FK:Trip),location(PK)}

Rating

1NF:

Rating{rating_id(PK),location,service,gentility,comment ,user_id(FK:User)}

2NF:

Rating{rating_id(PK),location,service,gentility,comment ,user_id(FK:User)}

3NF:

Rating{rating_id(PK),location,service,gentility,comment ,user_id(FK:User)}

BCNF:

Rating{rating_id(PK),location,service,gentility,comment ,user_id(FK:User)}

Country

1NF:

Country{country_name(PK)}

2NF:

Country{country_name(PK)}

3NF:

Country{country_name(PK)}

BCNF:

Country{country_name(PK)}

City

1NF:

City{city_id(PK),city_name, country_name(FK:Country)}

2NF:

City{city_id(PK),city_name, country_name(FK:Country)}

3NF:

City{city_id(PK),city_name, country_name(FK:Country)}

BCNF:

City{city_id(PK),city_name, country_name(FK:Country)}

Hospital

1NF:

Hospital{hospital_id(PK),hospital_name,rank,foundation_year,city_id(FK:city)}

2NF:

Hospital{hospital_id(PK),hospital_name,rank,foundation_year,city_id(FK:city)}

3NF:

Hospital{hospital_id(PK),hospital_name,rank,foundation_year,city_id(FK:city)}

BCNF:

Hospital{hospital_id(PK),hospital_name,rank,foundation_year,city_id(FK:city)}

Doctor

1NF:

Doctor{emp_id(PK),age,years_of_experience}

Doctor_Language {emp_id(FK:Doctor),language(PK)}

Doctor_Name {emp_id(FK:Doctor),first(PK),last(PK)}

2NF:

Doctor{emp_id(PK),age,years_of_experience}

Doctor_Language {emp_id(FK:Doctor),language(PK)}

Doctor_Name {emp_id(FK:Doctor),first(PK),last(PK)}

3NF:

Doctor{emp_id(PK),age,years_of_experience}

Doctor_Language {emp_id(FK:Doctor),language(PK)}

Doctor_Name {emp_id(FK:Doctor),first(PK),last(PK)}

BCNF:

Doctor{emp_id(PK),age,years_of_experience}

Doctor_Language {emp_id(FK:Doctor),language(PK)}

Doctor_Name {emp_id(FK:Doctor),first(PK),last(PK)}

Surgeon

1NF:

Surgeon{emp_id(FK:Doctor),num_of_suregeries}

Surgeon_speciality{emp_id(FK:Doctor),speciality(PK)}

2NF:

Surgeon{emp_id(FK:Doctor),num_of_suregeries}

Surgeon_speciality{emp_id(FK:Doctor),speciality(PK)}

3NF:

Surgeon{emp_id(FK:Doctor),num_of_suregeries}

Surgeon_speciality{emp_id(FK:Doctor),speciality(PK)}

BCNF:

Surgeon{emp_id(FK:Doctor),num_of_suregeries}

Surgeon_speciality{emp_id(FK:Doctor),speciality(PK)}

Practitioner

1NF:

Practitioner{emp_id(FK:Doctor)}

2NF:

Practitioner{emp_id(FK:Doctor)}

3NF:

Practitioner{emp_id(FK:Doctor)}

BCNF:

Practitioner{emp_id(FK:Doctor)}

Research Hospital

1NF:

Research_Hospital{ hospital_id(FK: Hospital) ,research_num}

Research_Hospital_ResearchTopic{ hospital_id(FK: Hospital),research_topic(PK)}

2NF:

Research_Hospital{ hospital_id(FK: Hospital),research_num}

Research_Hospital_ResearchTopic{ hospital_id(FK: Hospital),research_topic(PK)}

3NF:

Research_Hospital{ hospital_id(FK: Hospital), research_num}

Research_Hospital_ResearchTopic{ hospital_id(FK: Hospital),research_topic(PK)}

BCNF:

Research_Hospital{ hospital_id(FK: Hospital), research_num}

Research_Hospital_ResearchTopic{ hospital_id(FK: Hospital),research_topic(PK)}

Public Hospital

1NF:

Public_Hospital{ hospital_id(FK: Hospital)}

Public_Hospital_Dept{ hospital_id(FK: Hospital),department (PK)}

2NF:

Public_Hospital{ hospital_id(FK: Hospital)}

Public_Hospital_Dept{ hospital_id(FK: Hospital),department (PK)}

3NF:

Public_Hospital{ hospital_id(FK: Hospital)}

Public_Hospital_Dept{ hospital_id(FK: Hospital),department (PK)}

BCNF:

Public_Hospital{ hospital_id(FK: Hospital)}

Public_Hospital_Dept{ hospital_id(FK: Hospital),department (PK)}

Private Hospital

1NF:

Private_Hospital{ hospital_id(FK: Hospital)}

Private_Hospital_Fee{ hospital_id(FK: Hospital),service_fee(PK)}

2NF:

Private_Hospital{ hospital_id(FK: Hospital)}

Private_Hospital_Fee{ hospital_id(FK: Hospital),service_fee(PK)}

3NF:

Private_Hospital{ hospital_id(FK: Hospital)}

Private_Hospital_Fee{ hospital_id(FK: Hospital),service_fee(PK)}

BCNF:

Private_Hospital{ hospital_id(FK: Hospital)}

Private_Hospital_Fee{ hospital_id(FK: Hospital),service_fee(PK)}

Reserves

1NF:

Reserves {user_id(FK:User),trip_id(FK:Trip)}

2NF:

Reserves {user_id(FK:User),trip_id(FK:Trip)}

3NF:

Reserves {user_id(FK:User),trip_id(FK:Trip)}

BCNF:

Reserves {user_id(FK:User),trip_id(FK:Trip)}

Consists of

1NF:

Consits_of{ trip_id(FK:Trip),hospital_id(FK:Hospital)}

2NF:

Consits_of{ trip_id(FK:Trip),hospital_id(FK:Hospital)}

3NF:

Consits_of{ trip_id(FK:Trip),hospital_id(FK:Hospital)}

BCNF:

Consits_of{ trip_id(FK:Trip),hospital_id(FK:Hospital)}

Includes

1NF:

Includes{ trip_id(FK:Trip),country_name(FK:Country)}

2NF:

Includes{ trip_id(FK:Trip),country_name(FK:Country)}

3NF:

Includes{ trip_id(FK:Trip),country_name(FK:Country)}

BCNF:

Includes{ trip_id(FK:Trip),country_name(FK:Country)}

Has

1NF:

Has{ hospital_id(FK: Hospital), emp_id(FK:Doctor)}

2NF:

Has{ hospital_id(FK: Hospital), emp_id(FK:Doctor)}

3NF:

Has{ hospital_id(FK: Hospital), emp_id(FK:Doctor)}

BCNF:

Has{ hospital_id(FK: Hospital), emp_id(FK:Doctor)}

WorksFor

1NF:

WorksFor{ hospital_id(FK: Hospital), emp_id(FK:Doctor)}

2NF:

WorksFor{ hospital_id(FK: Hospital), emp_id(FK:Doctor)}

3NF:

WorksFor{ hospital_id(FK: Hospital), emp_id(FK:Doctor)}

BCNF:

WorksFor{ hospital_id(FK: Hospital), emp_id(FK:Doctor)}