Milestone 1 & 2
The Semantic-Conceptual Model
And Database Relational Model

COVER PAGE

Title: Hotel Management System Database

Created by: Danish Siddiqui

TABLE OF CONTENTS

Section No. Page

SECTION I: PROJECT DESCRIPTION

This project, Hotel Management Database System, will be going to include most of the functionality areas and technical tools that are necessary to run a good hotel business. The project will consider a chain of hotels such as “Hilton hotel and resorts”, in which a proper hotel management system will be elaborated with its services, sections, and all that makes it a desirable place to stay in during holidays or for business purposes. The project will have a defined booking process for the guests where they could book rooms for their family on their provided check in and check out dates. Along with guests, the tourists can also visit the hotel if they just want to avail the services and have fun in the hotel’s luxurious sections. Lastly, the project will have a payment process section where guests as well as tourists will provide their payment information to submit their payment. Many outsiders for example business people can also visit the hotel and their data will also be saved by the system. The ultimate goal to create this project is to let people book hotel rooms in the most friendliest way by minimizing the technical problems that may arise especially during the booking process. The hotel’s database system will have personal details of all the guests, tourists, and outsiders uniquely separated from each other using a primary key and their user type. Using the guest id, the hotel rooms will be reserved under one name per payment for N number of rooms selected. My motivation to create this project is to create a good database system for a well-organized institution and be able to complete all the tasks that a client wants me to do. I think this project requires so much creativity to design a good database system. Through the project, a very important technical issue during the booking process could be handled through the database system which will assign the reservation order to a unique guest. Another technical issue that I am planning to handle is that after the guest has made his reservation and paid the reservation amount, then if he wants to cancel the reservation, then the hotel employee would be able to delete his reservation information as well as his personal and payment information from the database system. Likewise, if he wants to make any changes to their reservation, then I will update the fields in my database system with new values. Same process will be applied for other users as well. To do such modifications and to handle the technical problems defined earlier in the description, a well-structured database system for hotel management has to be designed in order to avoid any kind of inconvenience and discomfort while saving a huge chunk of very important data that is needed to store in the system.

SECTION II: USE CASES

Use Case Title:	Guest Hotel Reservation
Actors:	User, Room, Employee, Account, Hotel, Account Type, Region, Guest, Family, Room Type, Check in/out, User Address, Sections, Events, Hotel Address, Employee Address
Description:	John decides to reserve a room in a hotel which is in another city. The system asks the user to enter his username and password to create an account first. John chooses the account type as a guest so that he could book some rooms. He provides check in and check out dates as well as his family size (number of adults and kids). He also has an option to bring his family with him, but he decides to stay by himself. Later, he decides that he should bring his family too but he can't go back to start with a new reservation because the system doesn't let him do it until he finishes this one. John calls the hotel's phone number and the receptionist picks up the call and transfers it to her manager to cancel his reservation so that he could reserve rooms for his family as well. This time, John logs in into his account, provides his information, and reserves two double rooms as room type for his family. He could make one reservation order per time as the system wouldn't allow him to make more than one order at the same time. John and his family love the first impression of the hotel and discover many sections including fitness area,pool, games and so on. John reads the list of upcoming events, and he learns that since he's staying in the hotel, he and his family can attend all the events including concerts, birthday parties, and holiday celebrations. John and his family seem excited about it.

Use Case Title:	Tourism and Business in the Hotel
Actors:	Room, Family Size, Room Type, Employee, Account, Account Type, Events, Businessperson, Business Company, Business Meeting, Birthday Party, Concert, Holiday Celebration, Holiday Name, Artist, Sections
Description:	Steven and Clara are colleagues at their work place. Steven tells Clara that he will be having a business meeting in a luxurious hotel where so many tech and business companies are participating as well. Clara seems interested to visit the hotel, but she will only be there as a tourist and wouldn't be able to join the business meeting. Steven doesn't need to create a hotel account, instead he can directly go to the hotel's meeting hall while Clara will have to authenticate herself through a hotel account and needs to choose the account type as a tourist. She calls the receptionist of the hotel and learns that there are going to be lots of events in the hotel. She wants to attend some of the events but since she's a tourist, the system would let her attend only concerts and holiday celebrations but not birthday parties. It's almost new year time therefore many well-known music bands will be performing this week as well. Clara can also go to other hotel's sections including the cuisine area, fitness center, gaming area, and many more. She feels that this is the right time to explore and enjoy as many things as she could since the hotel is crowded with so many tourists.

Use Case Title:	Hotel’s Special Services and Sections
Actors:	User, Room, Services, Cuisine, Category, Sections, Gaming Section, Category, Pool Section, Restaurants, Fitness Section, Fitness Type, CrossFit, Training Gym, Lunch, Breakfast, Dinner, Breakfast Menu, Lunch Menu, Dinner Menu
Description:	Both John and Clara can enjoy maximum hotel services. John is in his room with his family. He and his son decide to explore the food section in the hotel. The hotels are very famous especially for their food services and healthy food menus because they have a variety of restaurant categories including American, Chinese, and Italian. John feels that he should take his family to all restaurants during lunch and dinner; however during breakfast, he decides to eat outside of the hotel. The next day, John decides to visit the gaming section of the hotel upon his son’s request. The gaming section has categories such as casinos as well as digital games to enjoy. Clara decides to check out the hotel’s fitness section since she finds out that the hotel has both cross fit areas as well as training gyms where they could learn some cool exercises. Later in the evening, John takes his family to the pool section where there are a variety of pools to go in and the place looks enjoyable.

Use Case Title:	Payment and Transportation
Actors:	Guest, Employee, Room, Payment Type, Credit Card, Bank, Transportation, Car Company, Transportation Type
Description:	John arrives at the hotel after paying off all the hotel fees through his credit card in the system. Both Clara and John pay for each additional hotel service that they used during their visit. The system asks them for the payment type at first, but Clara only has a bank account so she will provide that, however; John once again uses his credit card to pay that fee. Clara has to catch a flight so she requests for the hotel transportation. The system provides her transportation types and makes her pick the best and the earliest trip to the airport. Clara picks up a very luxurious Mercedes model airport shuttle because she is too tired right now and wants to be comfortable during her trip.

Use Case Title:	Cancel Hotel Reservation
Actors:	Guest, Employee, Room, Payment, Owner, Owner Address
Description:	Roger makes a hotel reservation during summers for a picnic at Lake Tahoe. His family doesn't seem interested in going with him so Roger feels that he should cancel the reservation. The next day, Roger calls the hotel employee to cancel his reservation. The hotel employee cancels his reservation and removes all of his information from the system as long as John notifies for the cancellation before the check in date. If the guest arrives after the check in date to cancel his reservation, then the system does not cancel his reservation.

SECTION III: DATABASE REQUIREMENTS

1. User

A. A general user shall be able to create one account using a unique email.
B. A user shall have at most one family size.
C. A user shall be a businessperson with no account.
D. A register user shall be able to register as only one account type.
E. A register user shall be able to register for one account plan.
F. A registered guess shall have at least one payment type in the system.
G. A registered user with a guest account shall reserve multiple rooms in many hotels.
H. A registered user with a guest account guest shall check in and check out many rooms.
I. A registered user with a guest account shall eat unlimited cuisine.
J. A registered user with a guest account shall eat breakfast many times.
K. A registered user with a guest account shall eat lunch many times.
L. A registered user with a guest account shall eat dinner many times.
M. A registered user with a guest account shall be able to attend many events in all hotels.
N. A registered user with a guest account shall be able to attend many concerts in all hotels.
O. A registered user with a guest account shall be able to attend many holiday celebrations in all hotels.
P. A registered user with a guest account shall be able to attend at least one birthday party in all hotels.
Q. A registered user with a tourist account shall eat unlimited cuisine.
R. A registered user with a tourist account shall eat lunch many times.
S. A registered user with a tourist account shall eat dinner many times.
T. A registered user with a tourist account shall be able to attend many events in all hotels.
U. A registered user with a tourist account shall be able to attend many concerts in all hotels.
V. A registered user with a tourist account shall be able to attend many holiday celebrations in all hotels.
W. A user shall have a name.
X. A user shall have a last name.
Y. A user shall have a unique user id.
Z. A user shall have a unique email.

AA. A user shall be located at many user addresses.
BB. A registered user with a guest account shall be able to visit many section types in all hotels.
CC. A registered user with a tourist account shall be able to visit many section types in all hotels.
DD. A registered user with a guest account shall be able to visit the gaming section many times in all hotels.
EE. A registered user with a guest account shall be able to visit the fitness section many times in all hotels.
FF. A registered user with a tourist account shall be able to visit the gaming section many times in all hotels.
GG. A registered user with a tourist account shall be able to visit the fitness section many times in all hotels.
HH. All users shall take many transportations in all hotels.

2. User Address

A. A user address shall have a unique id.
B. A user address shall have a street.
C. A user address shall have a city.
D. A user address shall have a state.
E. A user address shall have a country.
F. A user address shall have a zip code.
G. A user address shall be located for many users.

3. Account

A. An account shall belong to one and only one user.
B. An account shall have one unique id.
C. An account shall have an account type.
D. An account shall have user id.
E. An account shall have one guest encrypted password.
F. An account shall be located in one region.
G. An account shall have one creation date.
H. An account shall have a renewal.
I. An account has an account type.
J. An account shall have a type id.

4. Account Type

A. An account type shall have one unique account id.

B. An account type shall have one description.
C. An account type shall enjoy supported services depending on the type.

5. Region

A. A region shall have a unique id.
B. A region shall have a description.
C. A region shall belong to many accounts.

6. Services:

A. A service has a unique service id.
B. A service has a description.

7. Family

A. A family shall have a unique family id.
B. A family shall have a guest id.
C. A family shall have at least one adult.
D. A family shall have many numbers of kids.

8. Businessperson

A. A businessperson shall have a unique businessperson id.
B. A businessperson shall have no account.
C. A businessperson shall have a name.
D. A businessperson shall have a last name.
E. A businessperson shall have a phone number.
F. A businessperson shall join many business meetings in many hotels.

9. Business Meeting

A. A business meeting shall have a unique meeting id.
B. A business meeting shall have a businessperson id.
C. A business meeting shall have one place.
D. A business meeting shall be joined by many businesspersons.
E. A business meeting shall be organized in many hotels.
F. A business meeting shall have a hotel id.
G. A business meeting shall be conducted by at most one business company at a time.
H. A business meeting shall have a company id.

10. Business Company

A. A business company shall have a unique company id.
B. A business company shall have a company name.
C. A business company shall conduct many business meetings.

11. Employee

A. An employee shall have a unique employee id.
B. An employee shall have a name.
C. An employee shall have a salary.
D. An employee shall have one role.
E. An employee shall supervise many employees.
F. An employee shall supervise many employees.
G. An employee shall work in many hotels.
H. An employee shall have a phone number.
I. An employee shall be located at many addresses.

12. Employee Address

A. An employee address shall have a unique id
B. An employee address shall be located for many addresses.
C. An employee address shall have a street.
D. An employee address shall have a city.
F. An employee address shall have a state.
G. An employee address shall have a country.
H. An employee address shall have a zip code.

13. Owner

A. An owner shall own many hotels.
B. An owner shall have a unique owner id.
C. An owner shall have a first name.
D. An owner shall have a last name
E. An owner shall have a phone number.
F. An owner shall have many owner addresses.

14. Owner Address

A. An owner address shall have a unique address id.
B. An owner address shall have a zip code.
C. An owner address shall have a country.

D. An owner address shall have a city.
E. An owner address shall have a state.
F. An owner address shall have a street.

15. Hotel

A. A hotel shall have a unique hotel id.
B. A hotel shall have many rooms.
C. A hotel shall be owned by many owners.
D. A hotel shall have a name.
E. A hotel shall have many employees working in it.
F. A hotel shall organize many events to be attended by many guests and tourists.
G. A hotell shall have many cuisines.
H. A hotel shall have many business meetings.
I. A hotel shall have many sections.
J. A hotel shall have many transportation services.
K. A hotel shall have at least one address.

16. Hotel Address

A. A hotel address shall have a unique id.
B. A hotel address shall be located for at least one hotel.
C. A hotel address shall have a street.
D. A hotel address shall have a state.
E. A hotel address shall have a country.
F. A hotel address shall have a zip code.
G. A hotel address shall have a city.

17. Room

A. A room shall have a unique room id.
B. A shall have one guest id.
C. A room shall have one hotel id.
D. A room shall belong to only one hotel.
E. A room shall have a price.
F. A room shall be reserved by at most one guest at a time.
G. A shall be checked in and checked out many times by the guest.
H. A room shall have only one room type.
I. A room shall have one number.

18. Room Type

A. A room type shall have one unique id.
B. A room type shall have one description.
C. A room type shall be of one room.

19. Sections

A. A section shall have a unique section id.
B. A section shall be in many hotels.
C. A section shall have a hotel id.
D. A section is a fitness section, gaming section, pool section, and resort section.
E. A section shall have at least one category.
F. A section shall be visited by many guests.
G. A section shall be visited by many tourists.

20. Category

A. A category shall have a unique category id.
B. A category shall have a description.
C. A category shall belong to many sections in all hotels.

21. Fitness Section

A. A fitness section shall have a section type id.
B. A fitness section shall have one fitness section id.
C. A fitness section shall be visited by many guests.
D. A fitness section shall be visited by many tourists.
E. A fitness section shall have at least one fitness type.

22. Fitness Type

A. A fitness type shall have a unique type id.
B. A fitness type is a crossfit and a training gym.
C. A fitness type shall have a title.

23. CrossFit

A. A crossfit shall have a unique crossfit id.
B. A crossfit shall have type id.
C. A crossfit shall have a starting time.
D. A crossfit shall have an ending time.

24. Training Gym

E. A training gym shall have a unique training id.
F. A training gym shall have type id.
G. A training gym shall have a starting time.
H. A training gym shall have an ending time.

25. Gaming Section

A. A gaming section shall have a section id.
B. A gaming section shall have one unique gaming section id.
C. A gaming section shall be visited by many guests.
D. A gaming section shall be visited by many tourists.
E. A gaming section shall have one category name.

26. Category

A. A game category shall have a unique category id.
B. A game category shall have a description.

27. Pool Section

A. A pool section shall have a section id.
B. A pool shall have a unique pool id.
C. A pool section shall be visited by many guests.
D. A pool section shall be visited by many tourists.
E. A pool section a pool number.

28. Payment Type

A. A payment type shall have many guests.
B. A payment type is a credit card or bank account.
C. A payment type shall one unique payment type id.
D. A payment type shall have one unique address.
E. A payment type shall have one city.
F. A payment type shall have one unique state.
G. A payment type shall have one zip code.
H. A payment type shall have one country.

29. Credit Card Payment Type

A. A credit card payment type shall have one unique card number.

B. A credit card payment type shall have one bank assigned number.
C. A credit card payment type shall have one expiration date.
D. A credit card shall have a type id.
E. A credit card payment type shall have one verification value (CVV).

30. Bank Account Payment Type

A. A bank account payment type shall have one unique account number.
B. A bank account payment type shall have one bank code.
C. A bank account payment type shall have a type id.
D. A bank account payment type shall have a routing number.

31. Cuisine

A. The cuisine is breakfast, lunch, and dinner.
B. The cuisine shall have one unique cuisine id.
C. The cuisine shall be eaten by many registered users with a guest account.
D. The cuisine shall be eaten by many registered users with a tourist account.
E. The cuisine shall have one name.
F. A cuisine shall be in many hotels.
G. The cuisine shall be provided by many restaurants.
H. The cuisine shall have many categories.

32. Breakfast

A. A breakfast shall have a cuisine id.
B. A breakfast shall have a unique breakfast id.
C. A breakfast shall be eaten by many registered users with a guest account.
D. A breakfast shall have a serving time.
E. A breakfast shall have a dish name.
F. A breakfast shall have at least one menu.

33. Breakfast Menu

A. A breakfast menu shall have a unique id.
B. A breakfast menu shall have calories.
C. A breakfast menu shall be in many breakfasts.

34. Lunch

A. A lunch shall have a cuisine id.

B. A lunch shall have a unique lunch id.
C. A lunch shall be eaten by many registered users with a guest account.
D. A lunch shall be eaten by many registered users with a tourist account.
E. A lunch shall have a dish name.
F. A breakfast shall have a serving time.
G. A lunch shall have at least one menu.

35. Lunch Menu

A. A lunch menu shall have a unique id.
B. A lunch menu shall have calories.
C. A lunch menu shall be in many lunches.

36. Dinner

A. A dinner shall have one cuisine id.
B. A dinner shall have a unique dinner id.
C. A dinner shall be eaten by many registered users with a guest account.
D. A dinner shall be eaten by many registered users with a tourist account.
E. A dinner shall have a serving time.
F. A dinner shall have a dish name.
G. A dinner shall have at least one menu.

37. Dinner Menu

A. A dinner menu shall have a unique id.
B. A dinner menu shall have calories.
C. A dinner menu shall be in many dinners.

38. Events

A. An event shall have a unique event id.
B. An event shall have a name.
C. An event shall be organized in many hotels.
D. An event is a concert, a party, and a celebration.
E. An event shall be attended by many registered users with a guest account.
F. An event shall be attended by many registered users with a tourist account.

39. Concert

A. A concert shall have a unique id.

B. A concert shall have an event id.
C. A concert shall have an artist id.
D. A concert shall be watched by many registered users with a guest account.
E. A concert shall be watched by many registered users with a tourist account.
F. A concert shall be performed by at most one artist.

40. Artist

A. An artist shall have one unique id.
B. An artist shall have a first name.
C. An artist shall have a last name.
D. An artist shall perform many concerts in many hotels.

41. Birthday Party

A. A birthday party shall have one unique id.
B. A birthday party shall have an event id.
C. A birthday party shall have one name.
D. A birthday party shall be attended by many guests.

42. Holiday Celebration

A. A holiday celebration shall have a unique id.
B. A holiday celebration shall have an event id.
C. A holiday celebration shall be attended by many registered users with a guest account.
D. A holiday celebration shall be attended by many registered users with a tourist account.
E. A holiday celebration shall have a holiday name.

43. Holiday Name

A. A holiday name shall have one id.
B. A holiday name shall have one description.

44. Transportation

A. A transportation shall have a unique transportation id.
B. A transportation shall have a hotel id.
C. A transportation shall be in many hotels.
D. A transportation shall be given to many users.

E. A transportation shall have at least one transportation Type.

45. Transportation Type

A. A transportation type shall have a unique type id.
B. A transportation type shall have a vehicle brand name.
C. A transportation type shall have a capacity.
D. A transportation type can do many transportations.

46. Car Company

A. A car company shall have a unique car id.
B. A car company shall have a car name.
C. A car company shall have a country name.
D. A car company shall manufacture many cars.

SECTION IV: DETAILED LIST OF MAIN ENTITIES, ATTRIBUTES, AND KEYS

1. User (strong)

a. user_id: key, numeric
b. name: multivalue, alphanumeric
c. last_name: multivalue, alphanumeric
d. email: key, alphanumeric
e. age: derived, numeric
f. phone: multivalue, alphanumeric
g. gender: multivalue, alphanumeric

2. Family (strong)

a. family_id: key, numeric
b. Adults: derived, numeric
c. kids: derived, numeric
d. user: weak key, numeric

3. User Address (strong)

a. address_id: key, numeric
b. street: alphanumeric
c. city: alphanumeric
d. state: alphanumeric
e. country: alphanumeric
f. zip code: alphanumeric

4. UseAdd (weak)

a. user: weak key, numeric
b. address: weak key, numeric

5. Account (weak)

a. account_id: key, numeric
b. user: weak key, numeric
c. type: weak key, numeric
d. password: multivalue, alphanumeric
e. created: composite, alphanumeric
f. region: weak key, numeric

g. renewal: composite, multivalue, date

6. Account Type (strong)

a. acct_typeid: key, numeric
b. desc: alphanumeric

7. Session (weak)

a. session_id: key, numeric
b. user_id: weak key, numeric
c. expires: composite, date

8. Region (strong)

a. region _id: key, numeric
b. description: alphanumeric

9. Region_Located (weak)

a. region: weak key, numeric
b. account: weak key, numeric

10. Services (strong)

a. Service_id: key, numeric
b. description: alphanumeric

11. Supported Services (weak)

a. type_id: weak key, numeric
b. service: weak key, numeric

12. Businessperson (strong)

a. bp_id: key, numeric
b. name: multivalue, alphanumeric
c. last_name: multivalue, alphanumeric

13. Business Meeting (weak)

a. meeting _id: key, numeric
b. bp_id: weak key, numeric
c. company_id: weak key, numeric

d. hotel_id: weak key, numeric

14. Conducted (weak)

a. busincompany: weak key, numeric
b. meeting: weak key, numeric

15. Joined (weak)

a. business_person: weak key, numeric
b. business_meeting: weak key, numeric

16. Business Company (strong)

a. company_id: key, numeric
b. name: multivalue, alphanumeric

17. Employee (strong)

a. employee_id: key, numeric
b. hotel: weak key, numeric
c. name: multivalue, alphanumeric
d. last_name: multivalue, alphanumeric
e. phone: multivalue, alphanumeric
f. role: multivalue, alphanumeric
g. managed_by: weak key, numeric
h. salary: numeric

18. Employee Address (strong)

a. emplyeeadd_id: key, numeric
b. street: alphanumeric
c. city: alphanumeric
d. state: alphanumeric
e. country: alphanumeric
f. zip code: alphanumeric

19. Work (weak)

a. employee: weak key, numeric
b. hotel: weak key, numeric

20. Owner (strong)

a. owner_id: key, numeric
b. hotel: weak key, numeric

c. name: multi-value, alphanumeric
d. last_name: multivalue, alphanumeric
e. phone: multivalue, alphanumeric

21. Owner Address (strong)
a. own_add_id: key, numeric
b. street: alphanumeric
c. city: alphanumeric
d. state: alphanumeric
e. country: alphanumeric
f. zip code: alphanumeric

22. OwnAdd (Weak)
a. owner: weak key, numeric
b. address: weak key, numeric

23. Owned (weak)
a. owner: weak key, numeric
b. hotel, weak key, numeric

24. Hotel (strong)
a. hotel_id: key,numeric
b. employee: weak key, numeric
d. name: alphanumeric

25. HotelAdd (weak)
a. hotel: weak key, numeric
b. hoteladd: weak key, numeric

26. Hotel Address (strong)
a. hoteladdress_id: key, numeric
b. street: alphanumeric
c. city: alphanumeric
d. state: alphanumeric
e. country: alphanumeric
f. zip code: alphanumeric

27. HotelHas (weak)

a. hotel: weak key, numeric
b. room: weak key, numeric
c. business: weak key, numeric
d. events: weak key, numeric
e. cuisine: weak key, numeric
f. transportation: weak key, numeric
g. sections: weak key, numeric

28. Room (weak)

a. room_id: key, numeric
b. hotel: weak key, numeric
c. acct_type: weak key, numeric
d. room_number: multi-value, alphanumeric
e. type: weak key, numeric
f. price: multivalue, alphanumeric

29. Reserve (weak)

a. reseving_time: compositie, alphanumeric, time
b. room: weak key, numeric
c. user: weak key, numeric

30. Check in_out (weak)

a. guest: weak key, numeric
b. room: weak key, numeric
c. from: composite, alphanumeric, date
d. to: composite, alphanumeric, date

31. Room Type (strong)

a. type_id: key, numeric
b. desc: alphanumeric

32. Payment Type: (strong)

a. type_id: key, numeric
b. billing_address: alphanumeric
c. billing_city: alphanumeric
d. billing_state: alphanumeric
e. billing_zipcode: alphanumeric
f. billing_country: alphanumeric

33. Billing Info (weak)

a. billing_id: key, numeric
b. user: weak key, numeric
c. payment_type: weak key, numeric
d. amount: alphanumeric

34. Credit Card (weak)

a. card_number: key,numeric
b. bank_code: alphanumeric
c. cvv: numeric
d. type: weak key, numeric
e. expiration_date: composite, date

35. Bank Account (weak)

a. account_number: key, numeric
b. bank_code: alphanumeric
c. type: weak key, numeric
d. routin_number: number

36. Cuisine (strong)

a. cuisine_id: key, numeric
b. dish_name: multi-value, alphanumeric
c. user: weak key, numeric
d. hotel: weak key, numeric

37. Food Category (weak)

a. category: weak key, numeric
b. cuisine: weak key, numeric

38. Category (strong)

a. category_id: key, numeric
b. desc: alphanumeric

39. Breakfast (weak)

a. cuisine: weak key, numeric
b. breakfast_id: key, numeric
c. serving_time: composite, multivalue, time
d. dish_name: multi-value, alphanumeric

40. HasMenu (weak)

a. menu: weak key, numeric
b. breakfast: weak key, numeric

41. Breakfast Menu (strong)

a. breakfastmenu_id: key, numeric
b. calories: alphanumeric

42. Eaten (weak)

a. user: weak key, numeric
b. cuisine: weak key, numeric
c. time: composite, alphanumeric, time

43. Lunch (weak)

a. cuisine: weak key, numeric
b. lunch_id: key, numeric
c. serving_time: composite, multivalue, time
d. dish_name: multi-value, alphanumeric

44. HasMenu (weak)

a. menu: weak key, numeric
b. lunch: weak key, numeric

45. Lunch Menu (strong)

a. lunchmenu_id: key, numeric
b. calories: alphanumeric

46. Dinner (weak)

a. cuisine: weak key, numeric
b. dinner_id: key, numeric
c. serving_time: composite, multivalue, time
d. dish_name: multi-value, alphanumeric

47. HasMenu (weak)

a. menu: weak key, numeric
b. dinner: weak key, numeric

48. Dinner Menu (strong)

a. dinnermeny_id: key, numeric
b. calories: alphanumeric

49. Provided (weak)

a. food: weak key, numeric
b. food_type: weak, numeric
c. restaurant: weak key, numeric

50. Restaurant (strong)

a. restaurant_id: key, numeric
b. name: multi-value, alphanumeric

51. Section (weak)

a. section_id: key,numeric
b. hotel: weak key, numeric
c. user: weak key, numeric

52. Section Category (weak)

a. section: weak key, numeric
b. category: weak key, numeric

53. Category (strong)

a. category_id: key, numeric
b. description: alphanumeric

54. Fitness Section (weak)

a. fitness_id: key, numeric
b. section:weak key, numeric

55. Fitness Type (strong)

a. type_id: key, numeric
b. title: multivalue, alphanumeric

56. CrossFit (weak)

a. crossfit_id: key, numeric
b. type: weak key, numeric
c. starting_time: compositie, multivalue, time
d. ending_time: composite, multivalue, time

57. Training Gym (weak)
e. training_id: key, numeric
f. type: weak key, numeric
g. starting_time: compositie, multivalue, time
h. ending_time: composite, multivalue, time

58. Gaming Section (weak)
a. gaming_id: key, numeric
b. section: key, numeric
c. name: multivalue, alphanumeric

59. Gaming Category (strong)
a. category: weak key, numeric
b. gaming: weak key, numeric

60. Category (strong)
a. category_id: key, numeric
b. description: alphanumeric

61. Pool Section (weak)
a. section: key, numeric
b. pool_id: key, numeric
c. pool_number: key, numeric

62. Events (strong)
a. event_id: key, numeric
b. name: multivalue, alphanumeric

63. Attended (weak)
a. user: weak key, numeric
b. event: weak key, numeric
c. time: composite, alphanumeric, time

64. Concerts (weak)
a. concert_id: key, numeric
b. event: weak key, numeric
c. artist: weak key, numeric

65. Artists (strong)

a. artist_id: key, numeric
b. name: multivalue, alphanumeric
c. last_name: multivalue, alphanumeric

66. Birthday Party (weak)

a. party_id: key, numeric
b. event: weak key, numeric
c. name: multivalue, alphanumeric

67. Holiday Celebrations (weak)

a. holiday_id: key, numeric
b. event: weak key, numeric
c. type: weak key, numeric

68. Holiday Type (strong)

a. desc: alphanumeric
b. type_id: key, numeric

69. Transportation (weak)

a. transportation_id: key, numeric
b. hotel: weak key, numeric
c. time: composite, multivalue, time

70. Take (weak)

a. transportation: weak key, numeric
b. user: weak key, numeric

71. Transportation Info (weak)

a. type: weak key, numeric
b. transportation: weak key, numeric
c. pick_up: multivalue, alphanumeric
d. drop_off: multivalue, alphanumeric

72. Transportation Type (strong)

a. type_id: key, numeric
b. model_year: multivalue, numeric
c. capacity: multivalue, numeric
d. vehicle_name: multivalue, alphanumeric

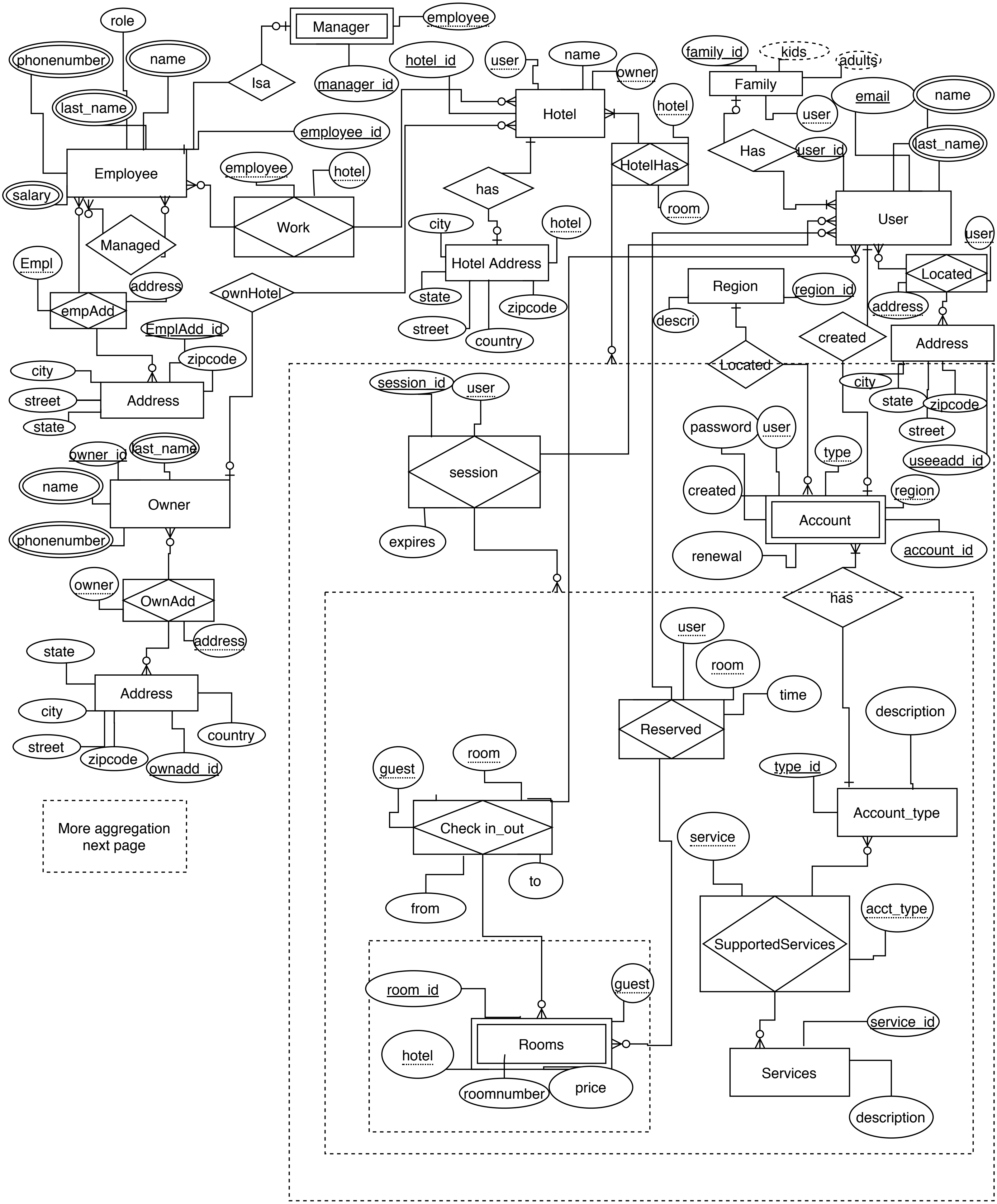
73. Manufactured (weak)

a. car: weak key, numeric
b. type: weak key, numeric

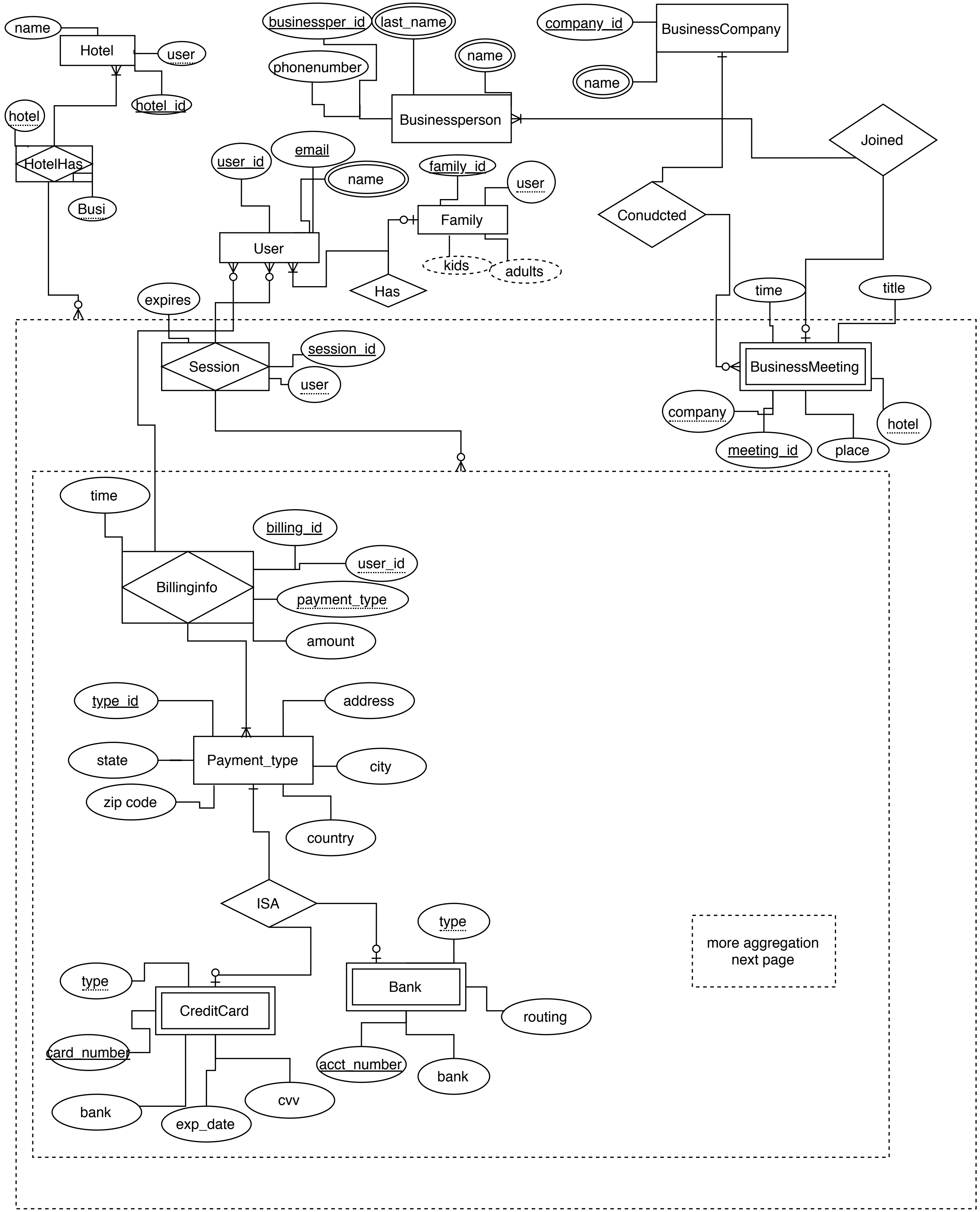
74. Car Company (strong)

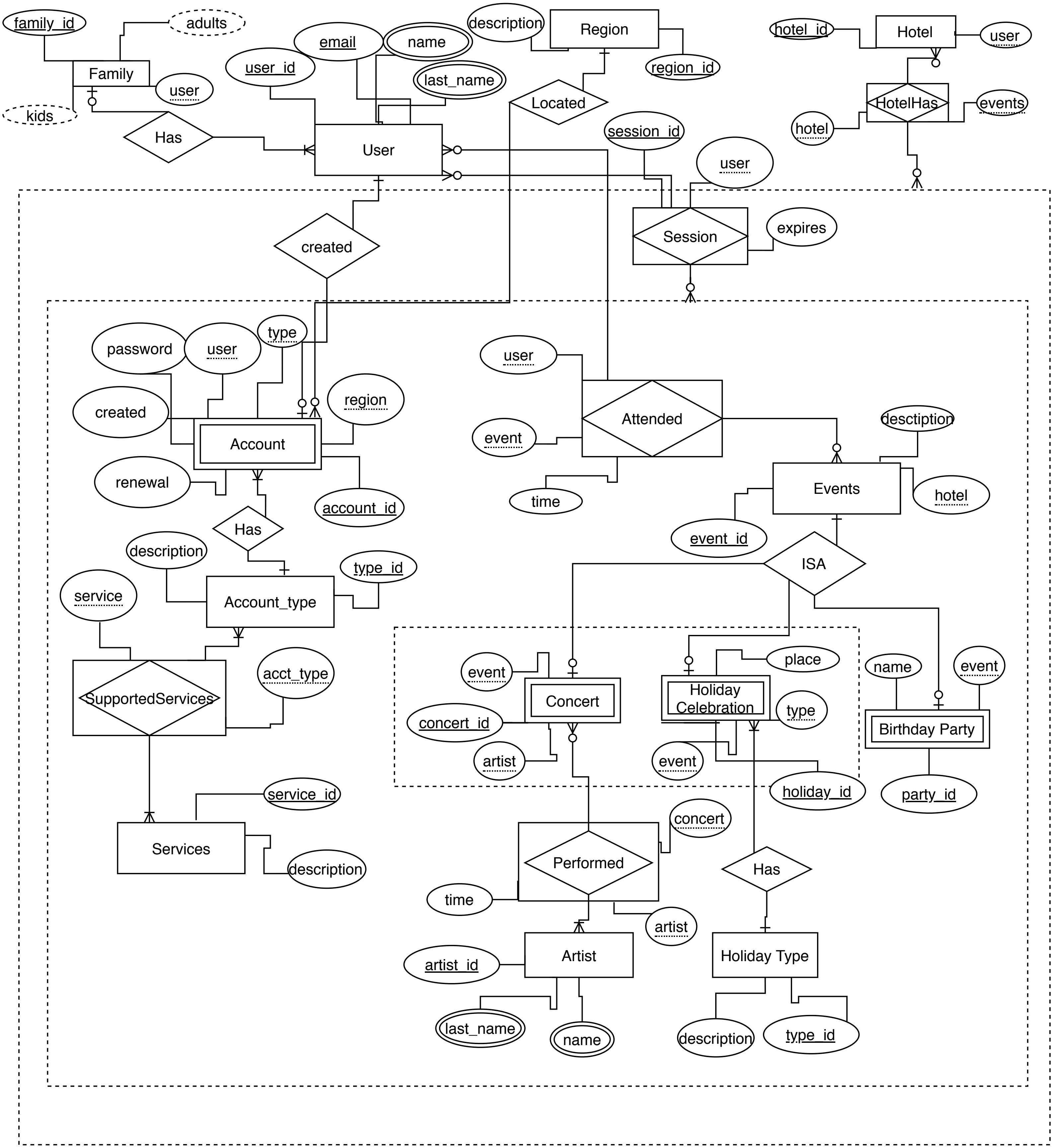
a. car_id: key, numeric
b. car_name: multivalue, alphanumeric
c. country: multivalue, numeric

SECTION V: ENTITY RELATIONSHIP DIAGRAM

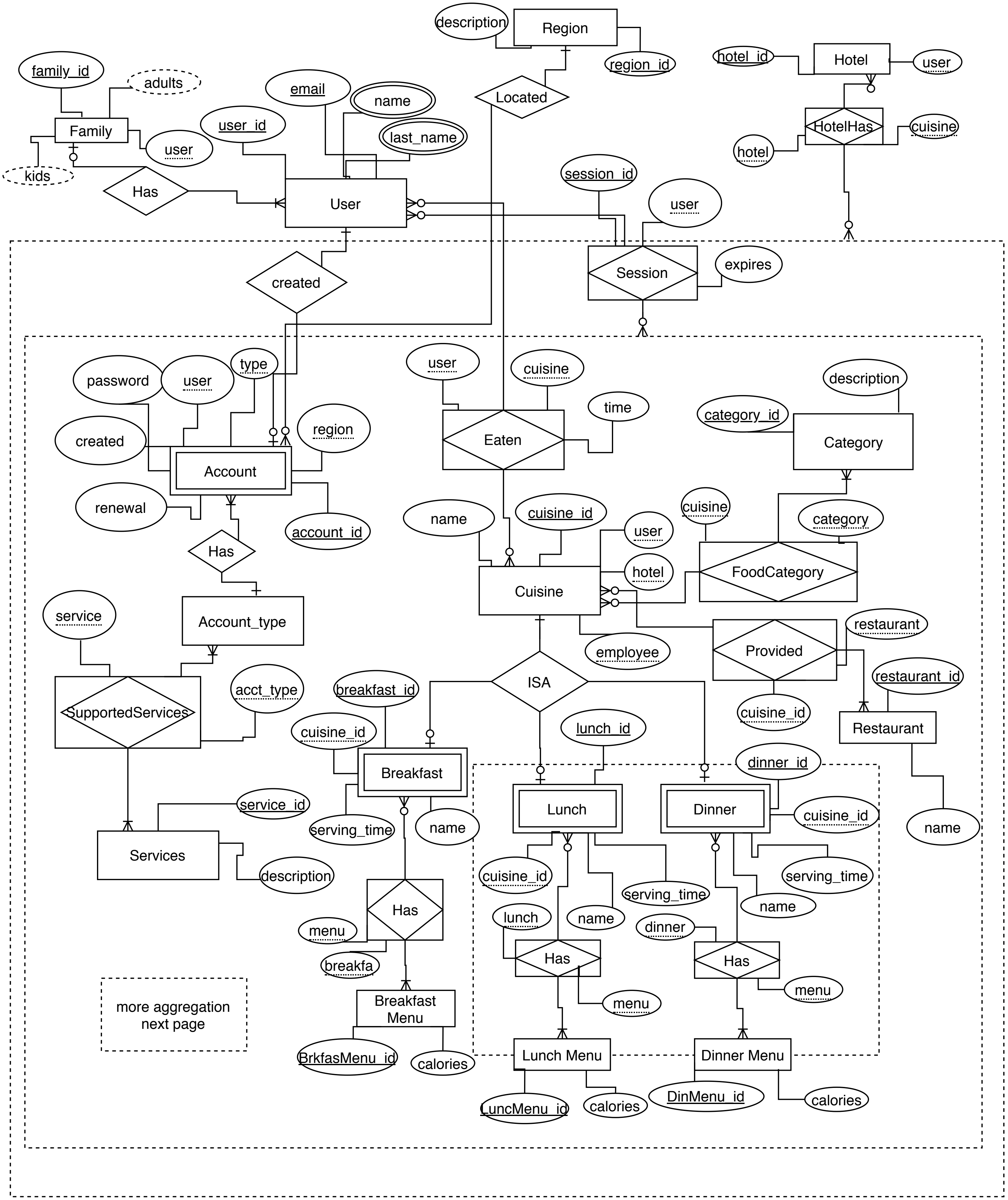


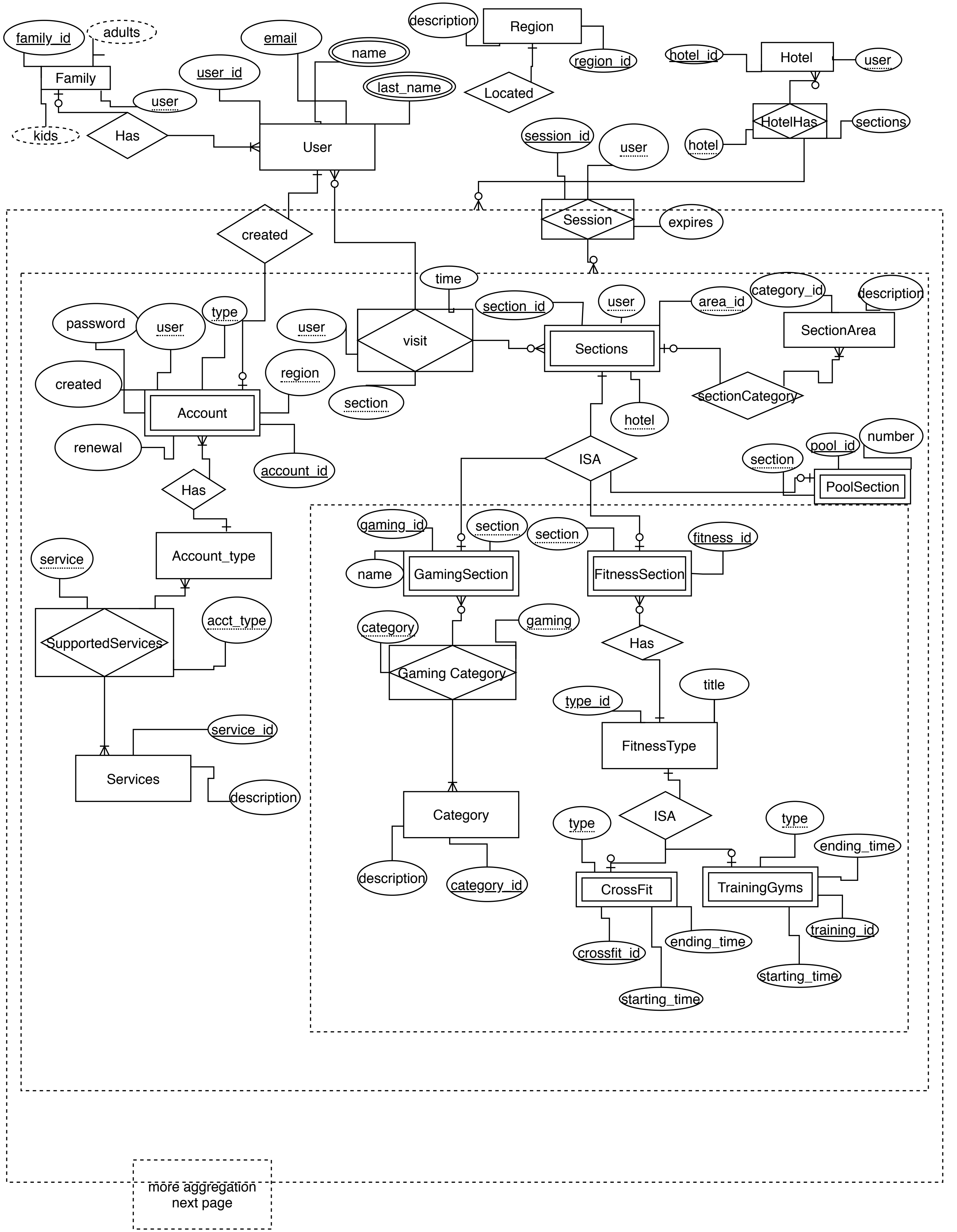
More aggregation next page

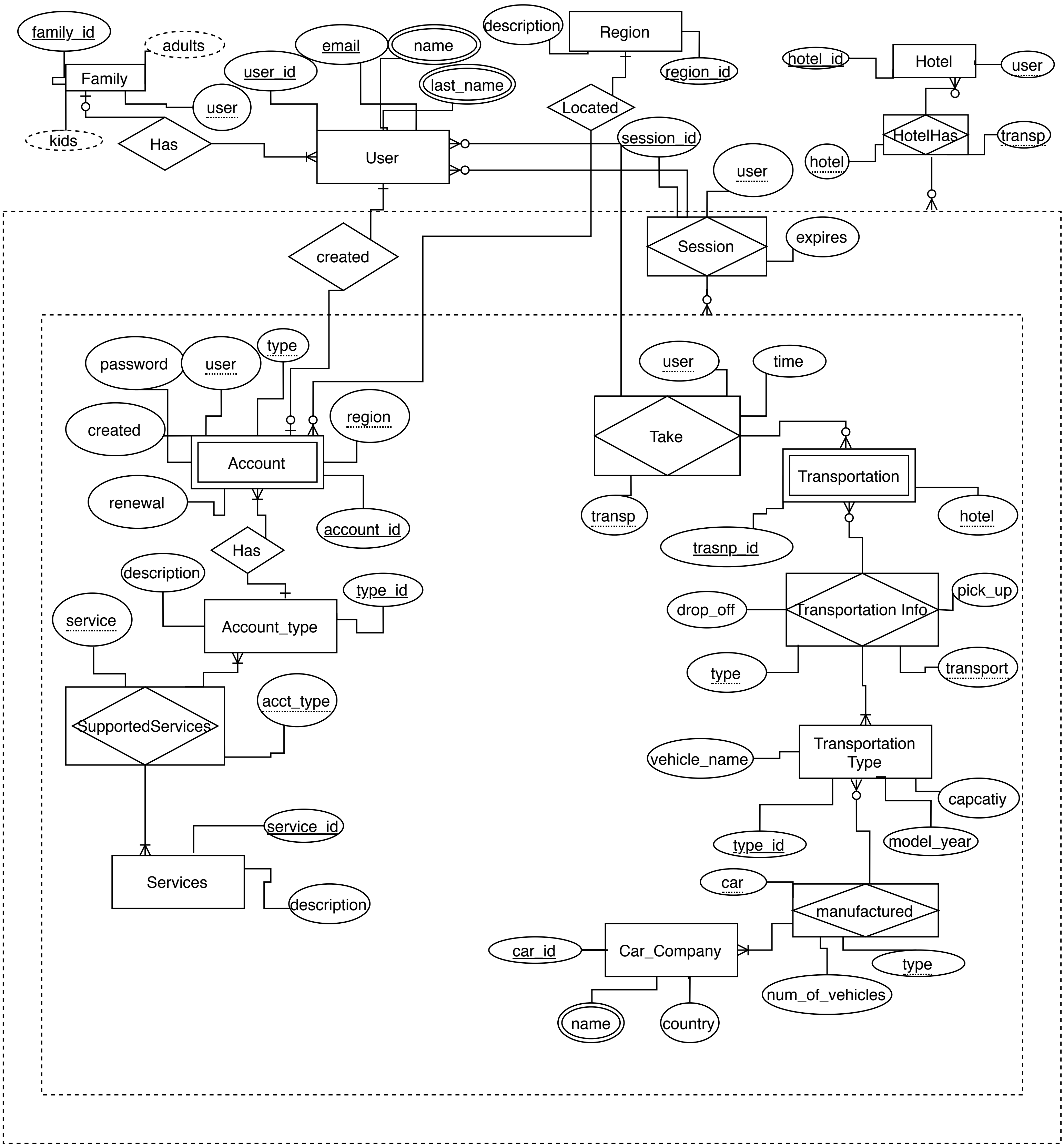

name
Hotel
user
hotel_id
hotel
HotelHas
Busi
businessper_id
last_name
phonenumber
name
Businessperson
company_id
BusinessCompany
name
Joined
Conudcted
user_id
email
name
User
family_id
user
Family
kids
adults
Has
expires
Session
session_id
user
time
title
BusinessMeeting
company
meeting_id
place
hotel
time
billing_id
user_id
Billinginfo
payment_type
amount
type_id
address
state
Payment_type
city
zip code
country
ISA
type
Bank
routing
acct_number
bank
type
CreditCard
card_number
bank
exp_date
cvv
more aggregation
next page



more aggregation next page


family_id
adults
Family
user
kids
Has
email
name
user_id
last_name
User
description
Region
region_id
Located
hotel_id
Hotel
user
HotelHas
cuisine
hotel
session_id
user
Session
expires
created
password
user
type
region
created
Account
renewal
account_id
Has
Account_type
service
SupportedServices
acct_type
Services
service_id
description
user
cuisine
time
Eaten
description
category_id
Category
cuisine
category
name
cuisine_id
user
hotel
FoodCategory
Cuisine
employee
Provided
restaurant
restaurant_id
cuisine_id
Restaurant
name
ISA
breakfast_id
cuisine_id
Breakfast
serving_time
name
lunch_id
Lunch
dinner_id
Dinner
cuisine_id
cuisine_id
serving_time
name
serving_time
name
Has
menu
breakfa
Breakfast Menu
BrkfasMenu_id
calories
lunch
Has
menu
dinner
Has
menu
Lunch Menu
LuncMenu_id
calories
Dinner Menu
DinMenu_id
calories
more aggregation next page


family_id
adults
Family
user
kids
Has
email
name
user_id
last_name
User
description
Region
region_id
Located
hotel_id
Hotel
user
HotelHas
sections
hotel
session_id
user
Session
expires
created
visit
time
user
section
password
user
type
created
region
Account
renewal
account_id
Has
Account_type
service
SupportedServices
acct_type
service_id
Services
description
section_id
user
Sections
area_id
category_id
description
SectionArea
sectionCategory
hotel
ISA
section
pool_id
number
PoolSection
gaming_id
section
section
fitness_id
name
GamingSection
FitnessSection
category
gaming
Gaming Category
Has
type_id
title
FitnessType
Category
description
category_id
ISA
type
type
ending_time
CrossFit
TrainingGyms
training_id
ending_time
crossfit_id
starting_time
starting_time
more aggregation next page



more aggregation next page

SECTION VI: TESTING TABLE

Rule	Entity A	Relation	Entity B	Cardinality	Pass/Fail	Error Description
1	User	Created	Account	1-to-1	Pass	None
2	User	Has	Family	1-to-M	Pass	None
3	User	Session	System	Aggregation / M-to-N	Pass	None
4	User (Guest)	Reserved	Rooms	M-to-N	Pass	None
5	User (Guest)	Check in/out	Rooms	M-to-N	Pass	None
6	Hotel	Has	Rooms	Aggregation / M-to-N	Pass	None
7	Employee	Works	Hotel	M-to-N	Fail	A Hotel shouldn't have at least one employee but it should have 0 or many employees.
8	Employee	Supervise	Employee	Recursion / M-to-1	Fail	An Employee shouldn't supervise at most one employee but he/she should supervise many employees.
9	Owner	Owns	Hotel	M-to-N	Fail	A hotel shouldn't be owned by at least one owner but it could be owned by many owners.
10	Account	Has	Account Type	1-to-M	Fail	An account type shouldn't have many accounts but it should have at least one account.
11	Account Type	Supported Services	Services	M-to-N	Pass	None
12	User	Isa	Business person	ISA / 1-to-1	Pass	None
13	Hotel	Has	Business Meeting	Aggregation / M-to-N	Pass	None
14	User	BillingInfo	Payment Type	M-to-N	Pass	None

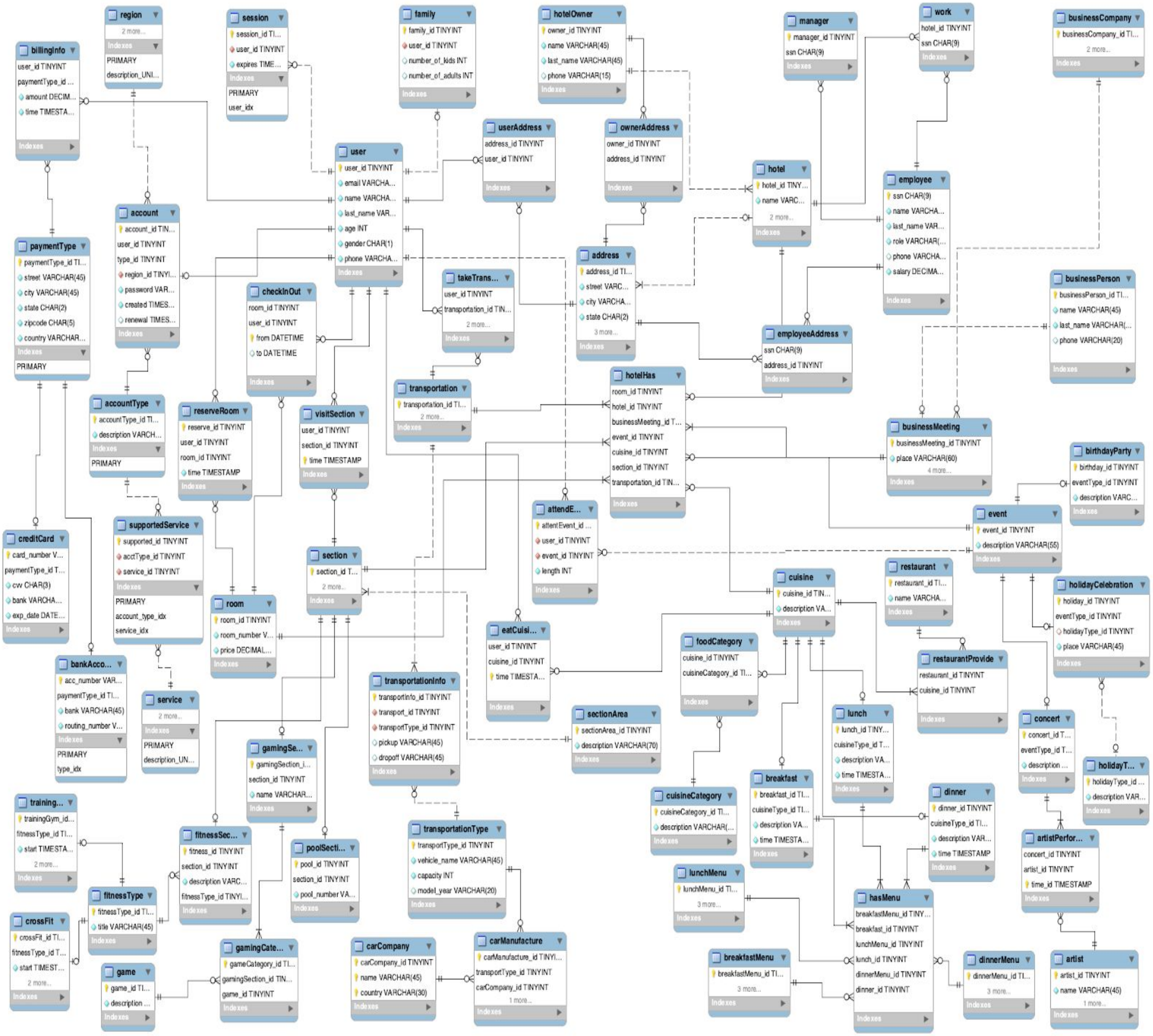
15	Payment Type	Isa	Credit Card	ISA / 1-to-1	Pass	None
16	Payment Type	Isa	Bank	ISA / 1-to-1	Pass	None
17	Business Person	Join	Business Meeting	M-to-N	Fail	A meeting shouldn’t be joined by at least one business person but it should be joined by 0 or many business persons.
18	Business Company	Conduct	Business Meeting	M-to-1	Fail	A business meeting shouldn’t be conducted by only one business company but it should be conducted by at least one business company.
19	User	Attend	Events	M-to-N	Pass	None
20	Hotel	Has	Events	Aggregation / M-to-N	Pass	None
21	Event	Isa	Concert	ISA / 1-to-1	Pass	None
22	Event	Isa	Birthday Party	ISA / 1-to-1	Pass	None
23	Event	Isa	Holiday Celebration	ISA / 1-to-1	Pass	None
24	Guest	Attend	Concert	M-to-N	Pass	None
25	Guest	Attend	Birthday Party	1-to-M	Fail	A guest shouldn’t attend at most one birthday party but he/she should attend 0 or many birthday parties.
26	Guest	Attend	Holiday Celebration	M-to-N	Pass	None
27	Tourist	Attend	Concert	M-to-N	Pass	None
28	Tourist	Attend	Holiday Celebration	M-to-N	Pass	None
29	Holiday	Has	Holiday	M-to-M	Fail	A holiday celebration should have

	Celebration		Type			only one holiday type.
30	Artist	Perform	Concert	M-to-1	Fail	A concert shouldn’t be performed by only one artist but it should be performed by at least one artist.
31	Hotel	Has	Cuisine	Aggregation / M-to-N	Pass	None
32	User	Eats	Cuisine	M-to-M	Pass	None
33	Cuisine	Isa	Breakfast	ISA / 1-to-1	Pass	None
34	Cuisine	Isa	Lunch	ISA / 1-to-1	Pass	None
35	Cuisine	Isa	Dinner	ISA / 1-to-1	Pass	None
36	Guest	Eats	Breakfast	M-to-N	Pass	None
37	Guest	Eats	Lunch	M-to-N	Pass	None
38	Guest	Eats	Dinner	M-to-N	Pass	None
39	Tourist	Eats	Lunch	M-to-N	Pass	None
40	Tourist	Eats	Dinner	M-to-N	Pass	None
41	Restaurant	Provides	Cuisine	M-to-N	Fail	A restaurant shouldn’t provide at least one cuisine but it should provide 0 or many cuisines.
42	Cuisine	Food Category	Category	M-to-N	Pass	A cuisine shouldn’t have 0 or many categories but it should have at least one category.
43	Hotel	Has	Sections	Aggregation / M-to-N	Pass	None
44	User	Visits	Sections	1-to-M	Fail	A user shouldn’t visit sections at most once but he/she should visit sections 0 or many times in all hotels.
45	Section	Isa	Gaming Section	ISA / 1-to-1	Pass	None

46	Section	Isa	Pool Section	ISA / 1-to-1	Pass	None
47	Section	Isa	Fitness Section	ISA / 1-to-1	Pass	None
48	Guest	Visits	Gaming Section	M-to-N	Pass	None
49	Guest	Visits	Pool Section	M-to-N	Pass	None
49	Guest	Visits	Fitness Section	M-to-N	Pass	None
51	Tourist	Visits	Gaming Section	M-to-N	Pass	None
52	Tourist	Visits	Fitness Section	M-to-N	Pass	None
53	Gaming Section	Gaming Category	Category	M-to-N	Fail	Gaming Section shouldn’t have 0 or many categories but it should have at least one gaming category.
54	Account	Located	Region	1-to-M	Pass	None
55	Fitness Section	Has	Fitness Type	M-to-1	Pass	None
56	Sections	Section Category	Category	M-to-N	Pass	None
57	Hotel	Has	Transportation	M-to-N	Pass	None
58	User	Takes	Transportation	M-to-1	Pass	A transportation shouldn’t be taken at most one user but it should taken by 0 or many users
59	Transportation	Transportation Category	Transportation Type	M-to-N	Pass	None
60	Car Company	Manufacture	Transportation Type	M-to-N	Pass	None
61	User	UseAdd	User Address	M-to-N	Pass	None

62	Owner	OwnAdd	Owner Address	M-to-N	Pass	None
63	Breakfast	HasMenu	Breakfast Menu	M-to-1	Fail	A breakfast shouldn’t have only one menu but it should have at least one menu.
64	Lunch	HasMenu	Lunch Menu	M-to-1	Fail	A lunch shouldn’t have only one menu but it should have at least one menu.
65	Dinner	HasMenu	Lunch Menu	M-to-1	Fail	A dinner shouldn’t have only one menu but it should have at least one menu.
66	Hotel	HotelAdd	Hotel Address	M-to-N	Pass	A hotel shouldn’t have 0 or many addresses but it should have at least one address
67	Employee	EmplAdd	Employee Address	M-to-N	Pass	None

SECTION VII: DATABASE MODEL



ON DELETE ON UPDATE ITEMIZED DESCRIPTION

1. Account table is implemented DELETE CASCADE RESTRICT UPDATE for the weak key user, RESTRICT UPDATE, DELETE CASCADE for accountType, and RESTRICT DELETE, RESTRICT UPDATE for region because if user is deleted then all account records related to it must be delete, if accountType is deleted then account records are also deleted, if region is deleted then account records related to it have no effect.

2. ArtistPerformance table is implemented DELETE CASCADE UPDATE CASCADE for the weak keys artist and concert because it can have more than one artist and more than one concert and if we delete and update artist or concert then it should delete and update all records related to them.

3. AttendEvent table is implemented DELETE CASCADE UPDATE CASCADE for the weak keys user and event because it can have more than one user and more than one event and if we delete and update user or event then it should delete and update all records related to them.

4. BankAccount table is implemented ON DELETE ON UPDATE for the weak key paymentType because a bank account is a child entity of the parent entity payment type so if we delete and update paymentType then it must delete and update all bank account records related to it.

5. BillingInfo table is implemented DELETE CASCADE UPDATE CASCADE for the weak keys user and paymentType because it can have more than one user and more than one payment type and if we delete and update user or paymentType then it should delete and update all records related to them.

.

6. BirthdayParty table is implemented DELETE CASCADE UPDATE CASCADE for the weak key event because a birthday party is a child entity of the parent entity event so if we delete and update an event then it must delete and update all bank birthday party records related to it.

7. Breakfast table is implemented DELETE CASCADE UPDATE CASCADE for the weak key cuisine because breakfast is a child entity of the parent entity cuisine so if we delete and update a cuisine then it must delete and update all bank breakfast records related to it.

8. CarManufacture table is implemented DELETE CASCADE UPDATE CASCADE for the weak keys carCompany and transportationType because car manufacture can have more than one car company and more than one transportation type and if we delete and update carCompany or transportationType then it should delete and update all records related to them.

9. CheckInOut table is implemented DELETE CASCADE UPDATE CASCADE for the weak keys room and user because it can have more than one room and more than one user and if we delete and update user or room then it should delete and update all records related to them..

10. Concert tablet is implemented DELETE CASCADE UPDATE CASCADE for the weak key event because a concert is a child entity of the parent entity event so if we delete and update an event then it must delete and update all concert records related to it.

11. CreditCard table is implemented DELETE CASCADE UPDATE CASCADE for the weak key paymentType because a credit card is a child entity of the parent entity paymentType so if we delete and update a paymentType then it must delete and update all credit card records related to it.

12. CrossFit table is implemented DELETE CASCADE UPDATE CASCADE for the weak key fitnessType because cross fit is a child entity of the parent entity fitnessType so if we delete and update a fitnessType then it must delete and update all crossFit records related to it.
13. Dinner table is implemented DELETE CASCADE UPDATE CASCADE for the weak key cuisine because dinner is a child entity of the parent entity cuisine so if we delete and update a cuisine then it must delete and update all dinner records related to it.
14. EatCuisine table is implemented DELETE CASCADE UPDATE CASCADE for the weak keys user and cuisine because it can have more than one user and more than one cuisine and if we delete and update a user or cuisine then it should delete and update all records related to them..
15. EmployeeAddress table is implemented DELETE CASCADE UPDATE CASCADE for the weak keys employee and address because it can have more than one employee and more than one address and if we delete and update employee or address then it should delete and update all records related to them.
16. Family table is implemented ON DELETE RESTRICT ON UPDATE CASCADE for the weak key user because it can have more than one user and if we update a user then it should update all the family records related to it but they shouldn’t be deleted.
17. FitnessSection table is implemented DELETE CASCADE UPDATE CASCADE for the weak keys section because it is a child entity of the parent entity section so if we delete or update section then it should delete or update all fitnessSection records, and it is set ON DELETE CASCADE UPDATE CASCADE for weak key fitnessType so if we delete or update section then it should delete or update all fitnessSection records.

18. FoodCategory table is implemented DELETE CASCADE UPDATE CASCADE for the weak keys cuisine and cuisineCategory because it can have more than one cuisine and more than one cuisine category and if we delete and update cuisine or cuisineCategory then it should delete and update all records related to them.
19. GamingCategory table is implemented DELETE CASCADE UPDATE CASCADE for the weak keys gamingSection and game because it can have more than one gaming section and more than one game and if we delete and update gamingSection or game then it should delete and update all records related to them..
20. GamingSection table is implemented DELETE CASCADE UPDATE CASCADE for the weak key section because a gaming section is a child entity of the parent table section so if we delete and update a section then it must delete and update all gamingSection records related to it.
21. HasMenu table is implemented DELETE CASCADE UPDATE CASCADE for the weak keys breakfast, breakfastMenu, lunch, lunchMenu, dinner, and dinnerMenu because it can have more than one breakfast, breakfastMenu, lunch, lunchMenu, dinner, and dinnerMenu and if we delete or update from any of those tables then their related records in hasMenu will also be deleted and updated.
22. HolidayCelebration table is implemented DELETE CASCADE UPDATE CASCADE for the weak keys event because it is a child entity of the parent entity event so if we delete or update event then all related records in holidayCelebration table must also be deleted or updated and it is also set DELETE CASCADE UPDATE CASCADE for weak key holidayType because if we delete or update event then all related records in holidayCelebration table must also be deleted or updated.

23. HotelHas table is implemented DELETE CASCADE UPDATE CASCADE for the weak keys room, hotel, businessMeeting, event, cuisine, section, and transportation because it can have more than one room, hotel, business meeting, event, cuisine, section, and transportation so if delete or update any of these entities then all of their related records will also be deleted and updated..

24. Lunch table is implemented DELETE CASCADE UPDATE CASCADEfor the weak key cuisine because it is a child entity of the parent entity cuisine so if we delete and update a cuisine then it must delete and update all lunch records related to it.

25. Manager table is implemented ON DELETE CASCADE ON UPDATE CASCADE for the weak key employee because it can have more than one employee and it has a recursive relationship with the employee table so if we delete or update an employee then it should delete and update all the related records in the manager table.

26. PoolSection table is implemented DELETE CASCADE UPDATE CASCADE for the weak key section because it is a child entity of the parent entity section so if we delete and update a section then it must delete and update all poolSection records related to it.

27. ReserveRoom table is implemented DELETE CASCADE UPDATE CASCADE for the weak keys user and room because it can have more than one user and more than one room and if we delete and update a user or room then it should delete and update all records related to them.

28. RestaurantProvide table is implemented DELETE CASCADE UPDATE CASCADE for the weak keys restaurant and cuisine because it can have more than one restaurant and more than cuisine and if we delete and update restaurant or cuisine then it should delete and update all records related to them.

29. Session table is implemented DELETE CASCADE UPDATE CASCADE for the weak key user because it can have more than one user so if we delete or update a user then all of its related records in session should also be deleted and updated..

30. SupportedService table is implemented DELETE CASCADE UPDATE CASCADE for the weak keys accountType and service because it can have more than one account type and more than service and if we delete and update accountType or service then it should delete and update all records related to them.

31. TakeTransportation table is implemented DELETE CASCADE UPDATE CASCADE for the weak keys transportation and user because it can have more than one transportation and more than one user and if we delete and update transportation or user then it should delete and update all records related to them.

32. TrainingGym table is implemented DELETE CASCADE UPDATE CASCADE for the weak key fitnessType because training gym is the child entity of the parent entity fitness type so if we delete and update a fitnessType then it must delete and update all trainingGym records related to it..

33. TransportationInfo table is implemented DELETE CASCADE UPDATE CASCADE for the weak keys transportation and transportationType because it can have more than one transportation and more than one transportation type and if we delete and update transportation or transportationType then it should delete and update all records related to them.

34. UserAddress table is implemented DELETE CASCADE UPDATE CASCADE for the weak keys user and address because it can have more than one user and more than one address and if we delete and update address or user then it should delete and update all records related to them.

35. VisitSection table is implemented DELETE CASCADE UPDATE CASCADE for the weak keys user and section because it can have more than one user and more than one section and if we delete and update section or user then it should delete and update all records related to them.

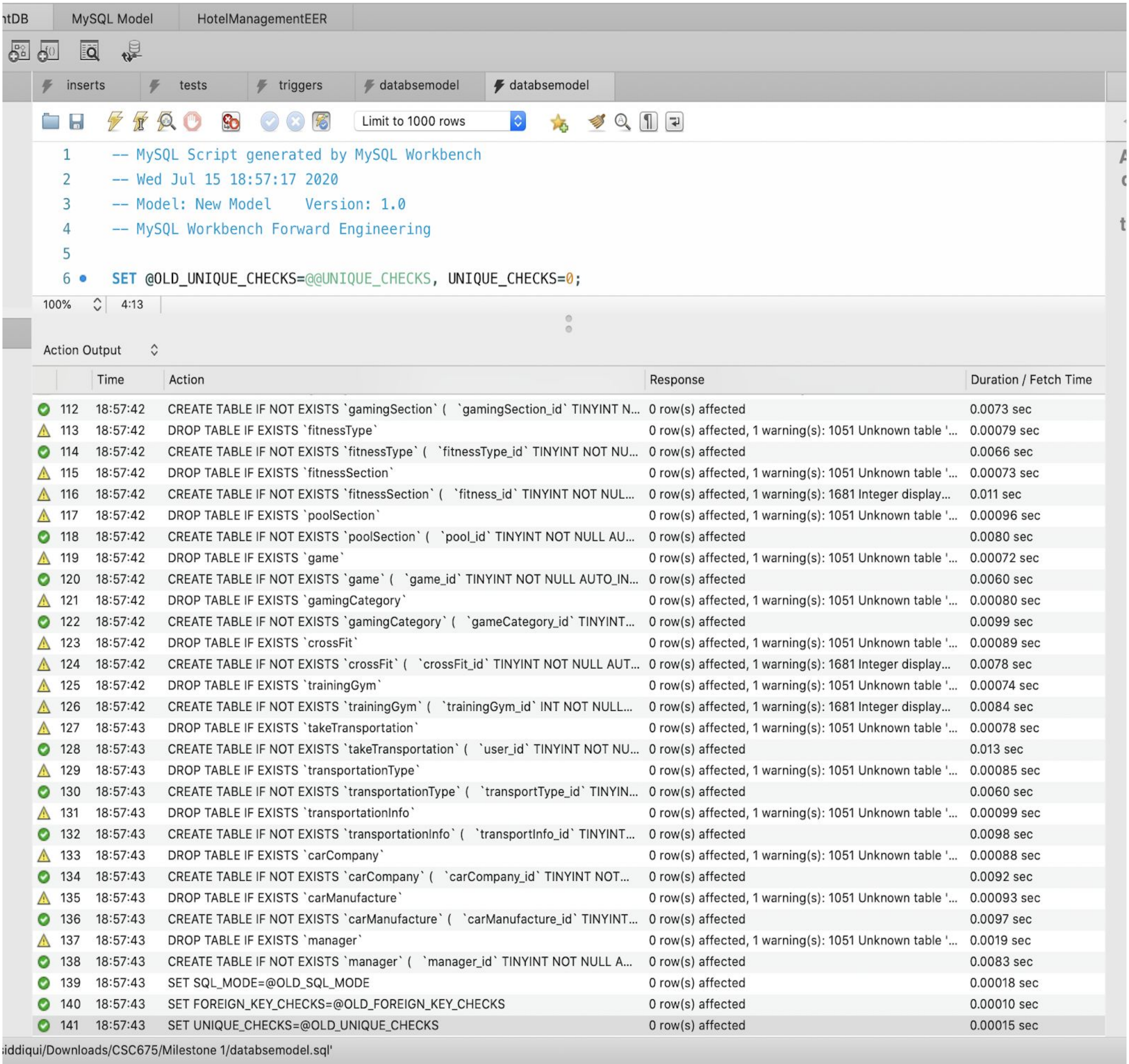
36. Work table is implemented DELETE CASCADE UPDATE CASCADE for the weak keys hotel and employee because it can have more than one hotel and more than one employee and if we delete and update hotel or employee then it should delete and update all records related to them.

37. Hotel table is implement UPDATE CASCADE, DELETE RESTRICT for the weak key owner because it can have one owner and if we delete an owner then it shouldn't delete the hotel records related to that owner because a hotel is strong, however; we can always update owner records in hotel table.

38. Address table is implemented UPDATE RESTRICT, DELETE RESTRICT for the weak key hotel because the address can exist by itself so if we delete a hotel or update it then its records shouldn't be deleted or updated in the address table..

39. Business Meeting table is implemented ON UPDATE CASCADE, ON DELETE RESTRICT for the weak key business person because if it can have many business person and if we delete a business person then its records shouldn't be deleted in business Meeting table and it is set ON UPDATE RESTRICT ON DELETE CASCADE for the weak key business company because it can have many business companies and if we delete the business company then its related records must be deleted; however we can't update them.

SECTION VIII: FORWARD ENGINEERING



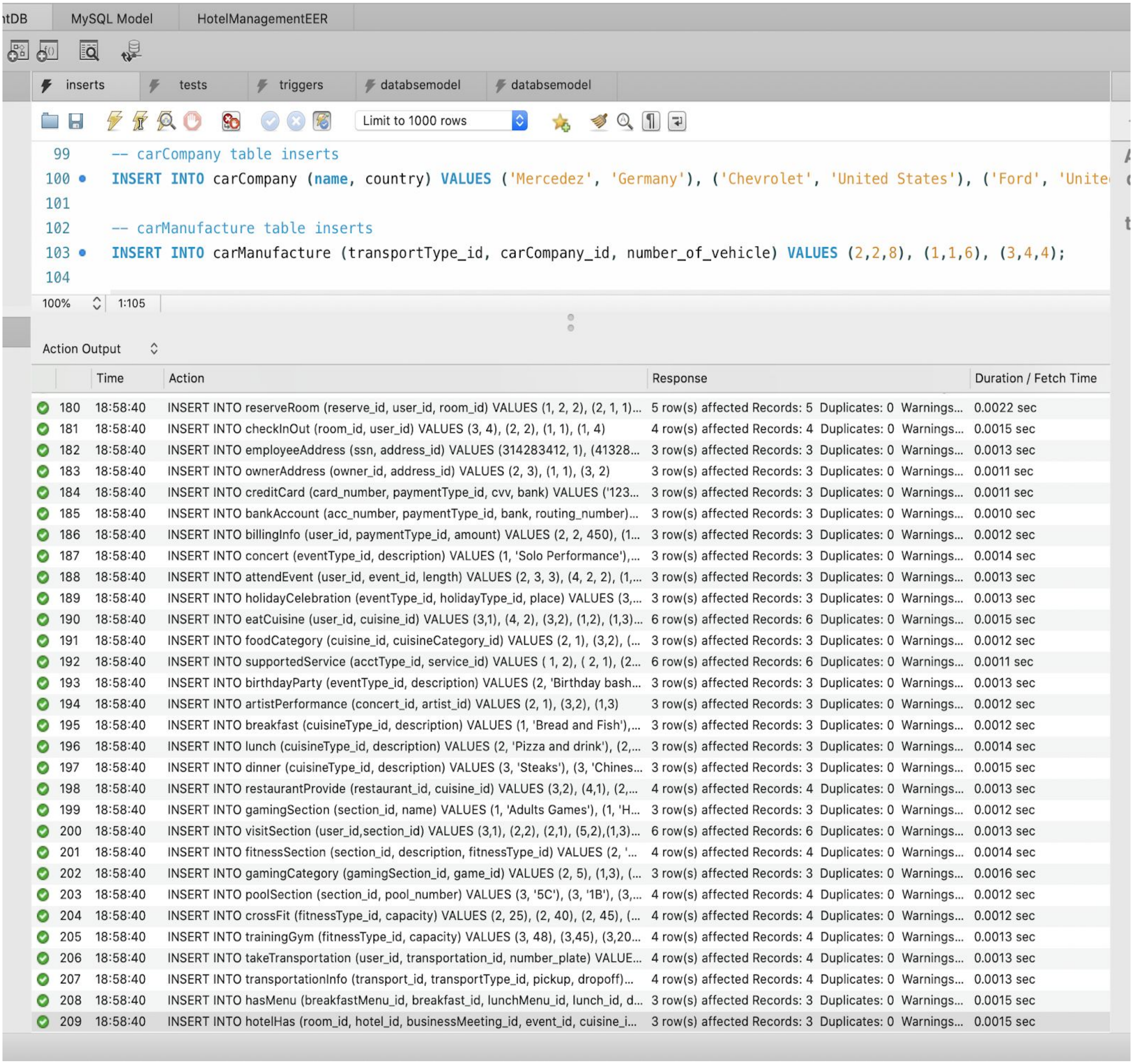
```
-- MySQL Script generated by MySQL Workbench
-- Wed Jul 15 18:57:17 2020
-- Model: New Model    Version: 1.0
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
```

```
     Time      Action                                                                      Response                                                          Duration / Fetch Time
✓ 112  18:57:42  CREATE TABLE IF NOT EXISTS `gamingSection` (  `gamingSection_id` TINYINT N...  0 row(s) affected                                                 0.0073 sec
⚠ 113  18:57:42  DROP TABLE IF EXISTS `fitnessType`                                          0 row(s) affected, 1 warning(s): 1051 Unknown table '...          0.00079 sec
✓ 114  18:57:42  CREATE TABLE IF NOT EXISTS `fitnessType` (  `fitnessType_id` TINYINT NOT NU...  0 row(s) affected                                                 0.0066 sec
⚠ 115  18:57:42  DROP TABLE IF EXISTS `fitnessSection`                                       0 row(s) affected, 1 warning(s): 1051 Unknown table '...          0.00073 sec
⚠ 116  18:57:42  CREATE TABLE IF NOT EXISTS `fitnessSection` (  `fitness_id` TINYINT NOT NUL...  0 row(s) affected, 1 warning(s): 1681 Integer display...          0.011 sec
⚠ 117  18:57:42  DROP TABLE IF EXISTS `poolSection`                                          0 row(s) affected, 1 warning(s): 1051 Unknown table '...          0.00096 sec
✓ 118  18:57:42  CREATE TABLE IF NOT EXISTS `poolSection` (  `pool_id` TINYINT NOT NULL AU...   0 row(s) affected                                                 0.0080 sec
⚠ 119  18:57:42  DROP TABLE IF EXISTS `game`                                                 0 row(s) affected, 1 warning(s): 1051 Unknown table '...          0.00072 sec
✓ 120  18:57:42  CREATE TABLE IF NOT EXISTS `game` (  `game_id` TINYINT NOT NULL AUTO_IN...    0 row(s) affected                                                 0.0060 sec
⚠ 121  18:57:42  DROP TABLE IF EXISTS `gamingCategory`                                       0 row(s) affected, 1 warning(s): 1051 Unknown table '...          0.00080 sec
✓ 122  18:57:42  CREATE TABLE IF NOT EXISTS `gamingCategory` (  `gameCategory_id` TINYINT...   0 row(s) affected                                                 0.0099 sec
⚠ 123  18:57:42  DROP TABLE IF EXISTS `crossFit`                                             0 row(s) affected, 1 warning(s): 1051 Unknown table '...          0.00089 sec
⚠ 124  18:57:42  CREATE TABLE IF NOT EXISTS `crossFit` (  `crossFit_id` TINYINT NOT NULL AUT... 0 row(s) affected, 1 warning(s): 1681 Integer display...          0.0078 sec
⚠ 125  18:57:42  DROP TABLE IF EXISTS `trainingGym`                                          0 row(s) affected, 1 warning(s): 1051 Unknown table '...          0.00074 sec
⚠ 126  18:57:42  CREATE TABLE IF NOT EXISTS `trainingGym` (  `trainingGym_id` INT NOT NULL...  0 row(s) affected, 1 warning(s): 1681 Integer display...          0.0084 sec
⚠ 127  18:57:43  DROP TABLE IF EXISTS `takeTransportation`                                   0 row(s) affected, 1 warning(s): 1051 Unknown table '...          0.00078 sec
✓ 128  18:57:43  CREATE TABLE IF NOT EXISTS `takeTransportation` (  `user_id` TINYINT NOT NU... 0 row(s) affected                                                 0.013 sec
⚠ 129  18:57:43  DROP TABLE IF EXISTS `transportationType`                                   0 row(s) affected, 1 warning(s): 1051 Unknown table '...          0.00085 sec
✓ 130  18:57:43  CREATE TABLE IF NOT EXISTS `transportationType` (  `transportType_id` TINYIN... 0 row(s) affected                                                 0.0060 sec
⚠ 131  18:57:43  DROP TABLE IF EXISTS `transportationInfo`                                   0 row(s) affected, 1 warning(s): 1051 Unknown table '...          0.00099 sec
✓ 132  18:57:43  CREATE TABLE IF NOT EXISTS `transportationInfo` (  `transportInfo_id` TINYINT... 0 row(s) affected                                                 0.0098 sec
⚠ 133  18:57:43  DROP TABLE IF EXISTS `carCompany`                                           0 row(s) affected, 1 warning(s): 1051 Unknown table '...          0.00088 sec
✓ 134  18:57:43  CREATE TABLE IF NOT EXISTS `carCompany` (  `carCompany_id` TINYINT NOT...     0 row(s) affected                                                 0.0092 sec
⚠ 135  18:57:43  DROP TABLE IF EXISTS `carManufacture`                                       0 row(s) affected, 1 warning(s): 1051 Unknown table '...          0.00093 sec
✓ 136  18:57:43  CREATE TABLE IF NOT EXISTS `carManufacture` (  `carManufacture_id` TINYINT... 0 row(s) affected                                                 0.0097 sec
⚠ 137  18:57:43  DROP TABLE IF EXISTS `manager`                                              0 row(s) affected, 1 warning(s): 1051 Unknown table '...          0.0019 sec
✓ 138  18:57:43  CREATE TABLE IF NOT EXISTS `manager` (  `manager_id` TINYINT NOT NULL A...    0 row(s) affected                                                 0.0083 sec
✓ 139  18:57:43  SET SQL_MODE=@OLD_SQL_MODE                                                  0 row(s) affected                                                 0.00018 sec
✓ 140  18:57:43  SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS                              0 row(s) affected                                                 0.00010 sec
✓ 141  18:57:43  SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS                                        0 row(s) affected                                                 0.00015 sec
```

iddiqui/Downloads/CSC675/Milestone 1/databsemodel.sql'

SECTION IX: INSERTING DATA

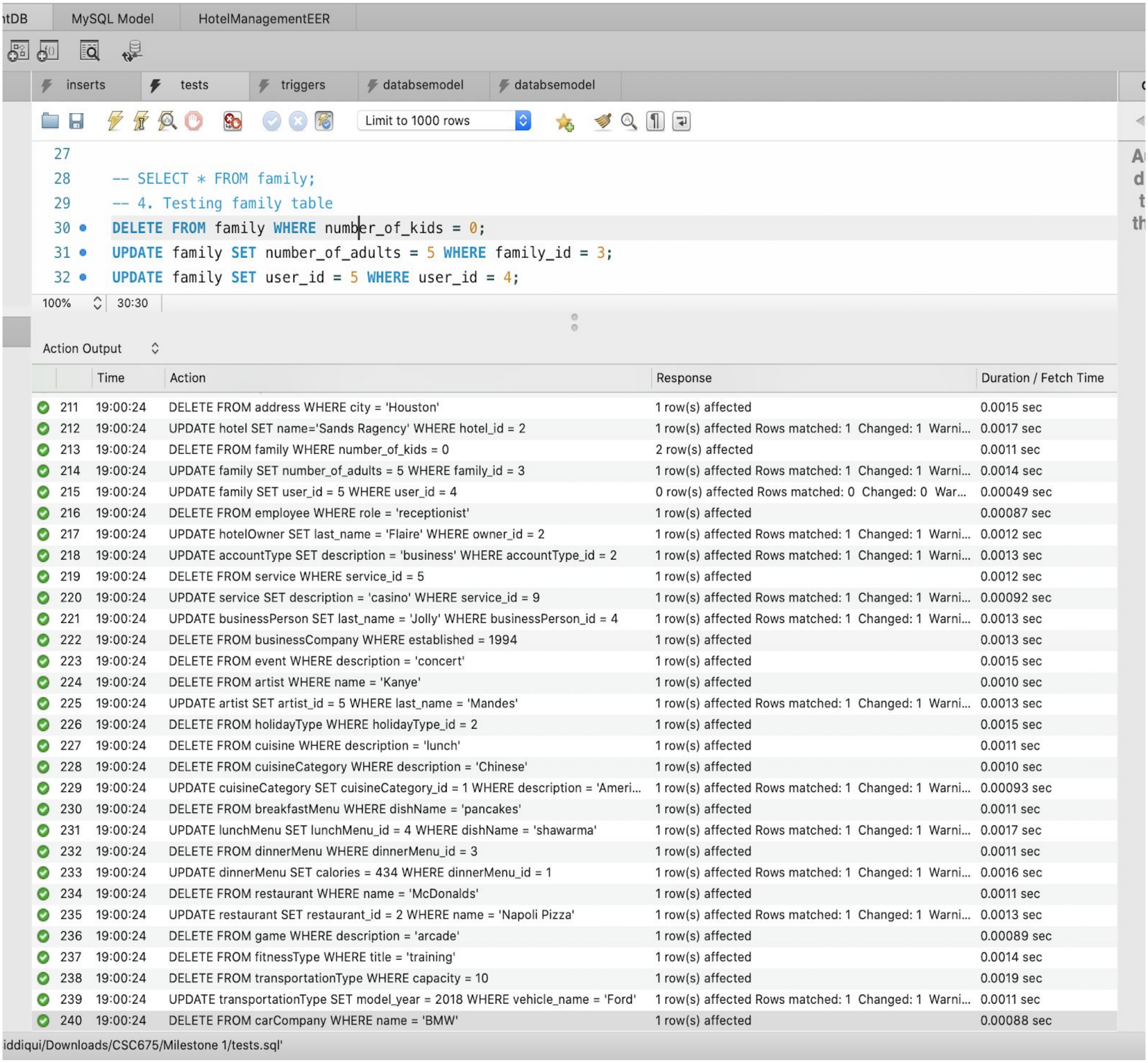


```
-- carCompany table inserts
INSERT INTO carCompany (name, country) VALUES ('Mercedez', 'Germany'), ('Chevrolet', 'United States'), ('Ford', 'Unite

-- carManufacture table inserts
INSERT INTO carManufacture (transportType_id, carCompany_id, number_of_vehicle) VALUES (2,2,8), (1,1,6), (3,4,4);
```

	Time	Action	Response	Duration / Fetch Time
180	18:58:40	INSERT INTO reserveRoom (reserve_id, user_id, room_id) VALUES (1, 2, 2), (2, 1, 1)...	5 row(s) affected Records: 5 Duplicates: 0 Warnings...	0.0022 sec
181	18:58:40	INSERT INTO checkInOut (room_id, user_id) VALUES (3, 4), (2, 2), (1, 1), (1, 4)	4 row(s) affected Records: 4 Duplicates: 0 Warnings...	0.0015 sec
182	18:58:40	INSERT INTO employeeAddress (ssn, address_id) VALUES (314283412, 1), (41328...	3 row(s) affected Records: 3 Duplicates: 0 Warnings...	0.0013 sec
183	18:58:40	INSERT INTO ownerAddress (owner_id, address_id) VALUES (2, 3), (1, 1), (3, 2)	3 row(s) affected Records: 3 Duplicates: 0 Warnings...	0.0011 sec
184	18:58:40	INSERT INTO creditCard (card_number, paymentType_id, cvv, bank) VALUES ('123...	3 row(s) affected Records: 3 Duplicates: 0 Warnings...	0.0011 sec
185	18:58:40	INSERT INTO bankAccount (acc_number, paymentType_id, bank, routing_number)...	3 row(s) affected Records: 3 Duplicates: 0 Warnings...	0.0010 sec
186	18:58:40	INSERT INTO billingInfo (user_id, paymentType_id, amount) VALUES (2, 2, 450), (1...	3 row(s) affected Records: 3 Duplicates: 0 Warnings...	0.0012 sec
187	18:58:40	INSERT INTO concert (eventType_id, description) VALUES (1, 'Solo Performance'),...	3 row(s) affected Records: 3 Duplicates: 0 Warnings...	0.0014 sec
188	18:58:40	INSERT INTO attendEvent (user_id, event_id, length) VALUES (2, 3, 3), (4, 2, 2), (1,...	3 row(s) affected Records: 3 Duplicates: 0 Warnings...	0.0013 sec
189	18:58:40	INSERT INTO holidayCelebration (eventType_id, holidayType_id, place) VALUES (3,...	3 row(s) affected Records: 3 Duplicates: 0 Warnings...	0.0013 sec
190	18:58:40	INSERT INTO eatCuisine (user_id, cuisine_id) VALUES (3,1), (4, 2), (3,2), (1,2), (1,3)...	6 row(s) affected Records: 6 Duplicates: 0 Warnings...	0.0015 sec
191	18:58:40	INSERT INTO foodCategory (cuisine_id, cuisineCategory_id) VALUES (2, 1), (3,2), (...	3 row(s) affected Records: 3 Duplicates: 0 Warnings...	0.0012 sec
192	18:58:40	INSERT INTO supportedService (acctType_id, service_id) VALUES (1, 2), (2, 1), (2...	6 row(s) affected Records: 6 Duplicates: 0 Warnings...	0.0011 sec
193	18:58:40	INSERT INTO birthdayParty (eventType_id, description) VALUES (2, 'Birthday bash...	3 row(s) affected Records: 3 Duplicates: 0 Warnings...	0.0013 sec
194	18:58:40	INSERT INTO artistPerformance (concert_id, artist_id) VALUES (2, 1), (3,2), (1,3)	3 row(s) affected Records: 3 Duplicates: 0 Warnings...	0.0012 sec
195	18:58:40	INSERT INTO breakfast (cuisineType_id, description) VALUES (1, 'Bread and Fish'),...	3 row(s) affected Records: 3 Duplicates: 0 Warnings...	0.0012 sec
196	18:58:40	INSERT INTO lunch (cuisineType_id, description) VALUES (2, 'Pizza and drink'), (2,...	3 row(s) affected Records: 3 Duplicates: 0 Warnings...	0.0014 sec
197	18:58:40	INSERT INTO dinner (cuisineType_id, description) VALUES (3, 'Steaks'), (3, 'Chines...	3 row(s) affected Records: 3 Duplicates: 0 Warnings...	0.0015 sec
198	18:58:40	INSERT INTO restaurantProvide (restaurant_id, cuisine_id) VALUES (3,2), (4,1), (2,...	4 row(s) affected Records: 4 Duplicates: 0 Warnings...	0.0013 sec
199	18:58:40	INSERT INTO gamingSection (section_id, name) VALUES (1, 'Adults Games'), (1, 'H...	3 row(s) affected Records: 3 Duplicates: 0 Warnings...	0.0012 sec
200	18:58:40	INSERT INTO visitSection (user_id,section_id) VALUES (3,1), (2,2), (2,1), (5,2),(1,3)...	6 row(s) affected Records: 6 Duplicates: 0 Warnings...	0.0013 sec
201	18:58:40	INSERT INTO fitnessSection (section_id, description, fitnessType_id) VALUES (2, '...	4 row(s) affected Records: 4 Duplicates: 0 Warnings...	0.0014 sec
202	18:58:40	INSERT INTO gamingCategory (gamingSection_id, game_id) VALUES (2, 5), (1,3), (...	3 row(s) affected Records: 3 Duplicates: 0 Warnings...	0.0016 sec
203	18:58:40	INSERT INTO poolSection (section_id, pool_number) VALUES (3, '5C'), (3, '1B'), (3,...	4 row(s) affected Records: 4 Duplicates: 0 Warnings...	0.0012 sec
204	18:58:40	INSERT INTO crossFit (fitnessType_id, capacity) VALUES (2, 25), (2, 40), (2, 45), (...	4 row(s) affected Records: 4 Duplicates: 0 Warnings...	0.0012 sec
205	18:58:40	INSERT INTO trainingGym (fitnessType_id, capacity) VALUES (3, 48), (3,45), (3,20...	4 row(s) affected Records: 4 Duplicates: 0 Warnings...	0.0013 sec
206	18:58:40	INSERT INTO takeTransportation (user_id, transportation_id, number_plate) VALUE...	4 row(s) affected Records: 4 Duplicates: 0 Warnings...	0.0013 sec
207	18:58:40	INSERT INTO transportationInfo (transport_id, transportType_id, pickup, dropoff)...	4 row(s) affected Records: 4 Duplicates: 0 Warnings...	0.0013 sec
208	18:58:40	INSERT INTO hasMenu (breakfastMenu_id, breakfast_id, lunchMenu_id, lunch_id, d...	3 row(s) affected Records: 3 Duplicates: 0 Warnings...	0.0015 sec
209	18:58:40	INSERT INTO hotelHas (room_id, hotel_id, businessMeeting_id, event_id, cuisine_i...	3 row(s) affected Records: 3 Duplicates: 0 Warnings...	0.0015 sec

SECTION X: TESTING



```
-- SELECT * FROM family;
-- 4. Testing family table
DELETE FROM family WHERE number_of_kids = 0;
UPDATE family SET number_of_adults = 5 WHERE family_id = 3;
UPDATE family SET user_id = 5 WHERE user_id = 4;
```

	Time	Action	Response	Duration / Fetch Time
211	19:00:24	DELETE FROM address WHERE city = 'Houston'	1 row(s) affected	0.0015 sec
212	19:00:24	UPDATE hotel SET name='Sands Ragency' WHERE hotel_id = 2	1 row(s) affected Rows matched: 1 Changed: 1 Warni...	0.0017 sec
213	19:00:24	DELETE FROM family WHERE number_of_kids = 0	2 row(s) affected	0.0011 sec
214	19:00:24	UPDATE family SET number_of_adults = 5 WHERE family_id = 3	1 row(s) affected Rows matched: 1 Changed: 1 Warni...	0.0014 sec
215	19:00:24	UPDATE family SET user_id = 5 WHERE user_id = 4	0 row(s) affected Rows matched: 0 Changed: 0 War...	0.00049 sec
216	19:00:24	DELETE FROM employee WHERE role = 'receptionist'	1 row(s) affected	0.00087 sec
217	19:00:24	UPDATE hotelOwner SET last_name = 'Flaire' WHERE owner_id = 2	1 row(s) affected Rows matched: 1 Changed: 1 Warni...	0.0012 sec
218	19:00:24	UPDATE accountType SET description = 'business' WHERE accountType_id = 2	1 row(s) affected Rows matched: 1 Changed: 1 Warni...	0.0013 sec
219	19:00:24	DELETE FROM service WHERE service_id = 5	1 row(s) affected	0.0012 sec
220	19:00:24	UPDATE service SET description = 'casino' WHERE service_id = 9	1 row(s) affected Rows matched: 1 Changed: 1 Warni...	0.00092 sec
221	19:00:24	UPDATE businessPerson SET last_name = 'Jolly' WHERE businessPerson_id = 4	1 row(s) affected Rows matched: 1 Changed: 1 Warni...	0.0013 sec
222	19:00:24	DELETE FROM businessCompany WHERE established = 1994	1 row(s) affected	0.0013 sec
223	19:00:24	DELETE FROM event WHERE description = 'concert'	1 row(s) affected	0.0015 sec
224	19:00:24	DELETE FROM artist WHERE name = 'Kanye'	1 row(s) affected	0.0010 sec
225	19:00:24	UPDATE artist SET artist_id = 5 WHERE last_name = 'Mandes'	1 row(s) affected Rows matched: 1 Changed: 1 Warni...	0.0013 sec
226	19:00:24	DELETE FROM holidayType WHERE holidayType_id = 2	1 row(s) affected	0.0015 sec
227	19:00:24	DELETE FROM cuisine WHERE description = 'lunch'	1 row(s) affected	0.0011 sec
228	19:00:24	DELETE FROM cuisineCategory WHERE description = 'Chinese'	1 row(s) affected	0.0010 sec
229	19:00:24	UPDATE cuisineCategory SET cuisineCategory_id = 1 WHERE description = 'Ameri...	1 row(s) affected Rows matched: 1 Changed: 1 Warni...	0.00093 sec
230	19:00:24	DELETE FROM breakfastMenu WHERE dishName = 'pancakes'	1 row(s) affected	0.0011 sec
231	19:00:24	UPDATE lunchMenu SET lunchMenu_id = 4 WHERE dishName = 'shawarma'	1 row(s) affected Rows matched: 1 Changed: 1 Warni...	0.0017 sec
232	19:00:24	DELETE FROM dinnerMenu WHERE dinnerMenu_id = 3	1 row(s) affected	0.0011 sec
233	19:00:24	UPDATE dinnerMenu SET calories = 434 WHERE dinnerMenu_id = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warni...	0.0016 sec
234	19:00:24	DELETE FROM restaurant WHERE name = 'McDonalds'	1 row(s) affected	0.0011 sec
235	19:00:24	UPDATE restaurant SET restaurant_id = 2 WHERE name = 'Napoli Pizza'	1 row(s) affected Rows matched: 1 Changed: 1 Warni...	0.0013 sec
236	19:00:24	DELETE FROM game WHERE description = 'arcade'	1 row(s) affected	0.00089 sec
237	19:00:24	DELETE FROM fitnessType WHERE title = 'training'	1 row(s) affected	0.0014 sec
238	19:00:24	DELETE FROM transportationType WHERE capacity = 10	1 row(s) affected	0.0019 sec
239	19:00:24	UPDATE transportationType SET model_year = 2018 WHERE vehicle_name = 'Ford'	1 row(s) affected Rows matched: 1 Changed: 1 Warni...	0.0011 sec
240	19:00:24	DELETE FROM carCompany WHERE name = 'BMW'	1 row(s) affected	0.00088 sec

iddiqui/Downloads/CSC675/Milestone 1/tests.sql'

SECTION XI: TESTING TABLE

Entity	SQLQuery	OK/Failed	Error Description	Possible Solution
user	DELETE	Failed	Couldn’t restrict delete due to running tests against an older version of database model. Cannot delete or update a parent row	Restart the connection to run tests against a newer version of model. Delete on restrict in family table, change it to cascade
user	UPDATE	Failed	Couldn’t restrict update due to running tests against an older version of database model. Cannot delete or update a parent row	Restart the connection to run tests against a newer version of model. Update on restrict in account table, change it to cascade
address	DELETE	OK	None	None
address	UPDATE	Failed	Duplicate entry '2' for key 'address.PRIMARY'	Don’t allow address id to be primary key
hotel	DELETE	Failed	Cannot delete or update a parent row	Delete on restrict in address table, change it to cascade
hotel	UPDATE	OK	None	None
family	DELETE	OK	None	None
family	UPDATE	OK	None	None
employee	DELETE	OK	None	None
employee	UPDATE	Failed	Cannot delete or update a parent row	Need to delete from the tables where ssn is used as fk
hotelOwner	DELETE	Failed	Cannot delete or update a parent row	Delete on restrict in hotel table, need to first delete from the hotel
hotelOwner	UPDATE	OK	None	None
region	DELETE	Failed	Cannot delete or update a parent row	Delete on restrict, need to change it to cascade or delete it from account table

region	UPDATE	FAILED	Duplicate entry 'New Zealand' for key 'region.description_UNIQUE'	Make region description not unique
accountType	DELETE	Failed	Cannot delete or update a parent row	Delete on restrict, need to delete from account table
accountType	UPDATE	OK	None	None
service	DELETE	OK	None	None
service	UPDATE	OK	None	None

businessPerson	DELETE	Failed	Cannot delete or update a parent row	Delete on restrict, change it to cascade
businessPerson	UPDATE	OK	None	None
businessCompany	DELETE	OK	None	None
businessCompany	UPDATE	Failed	Cannot delete or update a parent row	Delete on restrict, need to delete from the meeting table
paymentType	DELETE	Failed	Cannot delete or update a parent row	Delete on restrict, need to delete from payment table
paymentType	UPDATE	Failed	Duplicate entry '6' for key 'paymenttype.PRIMARY'	Should set a paymentType id that is not already in the table
event	DELETE	OK	None	None
event	UPDATE	Failed	Duplicate entry 'holiday celebration' for key 'event.description_UNIQUE'	Insert a different event name which isn’t already there
artist	DELETE	OK	None	None
artist	UPDATE	OK	None	None
holidayType	DELETE	OK	None	None

holidayType	UPDATE	Failed	Duplicate entry 'New Year' for key 'holidaytype.description_UNIQUE'	Insert a different holiday type which isn’t already there
cuisine	DELETE	OK	None	None
cuisine	UPDATE	Failed	Unknown column ‘name’ in ‘field list’	Use SELECT * to use the cuisine table columns
cuisineCategory	DELETE	OK	None	None
cuisineCategory	UPDATE	OK	None	None
breakfastMenu	DELETE	OK	None	None
breakfastMenu	UPDATE	Failed	Duplicate entry 'eggs' for key 'breakfastmenu.dish_UNIUE'	dishName is unique so we have to insert a different dish name every time
lunchMenu	DELETE	Failed	Unknown column 'soda' in 'where clause'	Should look for a column name that exists in the lunchMenu table
lunchMenu	UPDATE	OK	None	None

dinnerMenu	DELETE	OK	None	None
dinnerMenu	UPDATE	OK	None	None
restaurant	DELETE	OK	None	None
restaurant	UPDATE	OK	None	None
sectionArea	DELETE	Failed	Truncated incorrect DOUBLE value: 'Game for fun'	Delete with correct data type
sectionArea	UPDATE	Failed	Cannot delete or update a parent row	
game	DELETE	OK	None	None
game	UPDATE	Failed	Duplicate entry 'action' for	Need to put a different game

			key 'game.description_UNIQUE'	description
fitnessType	DELETE	OK	None	None
fitnessType	UPDATE	Failed	Unknown column 'fitness_id' in 'field list'	Need to update with column name that exists in the table or use SELECT before updating
transportationType	DELETE	Failed	0 rows returned / 0 rows affected	Need to delete with value that exist in the table
transportationType	UPDATE	OK	None	None
carCompany	DELETE	OK	None	None
carCompany	UPDATE	Failed	Duplicate entry 'Ford' for key 'carcompany.name_UNIQUE'	Car name is unique therefore update or add a different car name
businessPerson	DELETE	Fixed	businessPerson was required to join a meeting	If no businessPerson or if delete then SET NULL
businessCompany	UPDATE	Fixed	businessCompany could be updated while existing in businessMeeting table	If updated then RESTRICT