Database Design and Implementation
Final Project

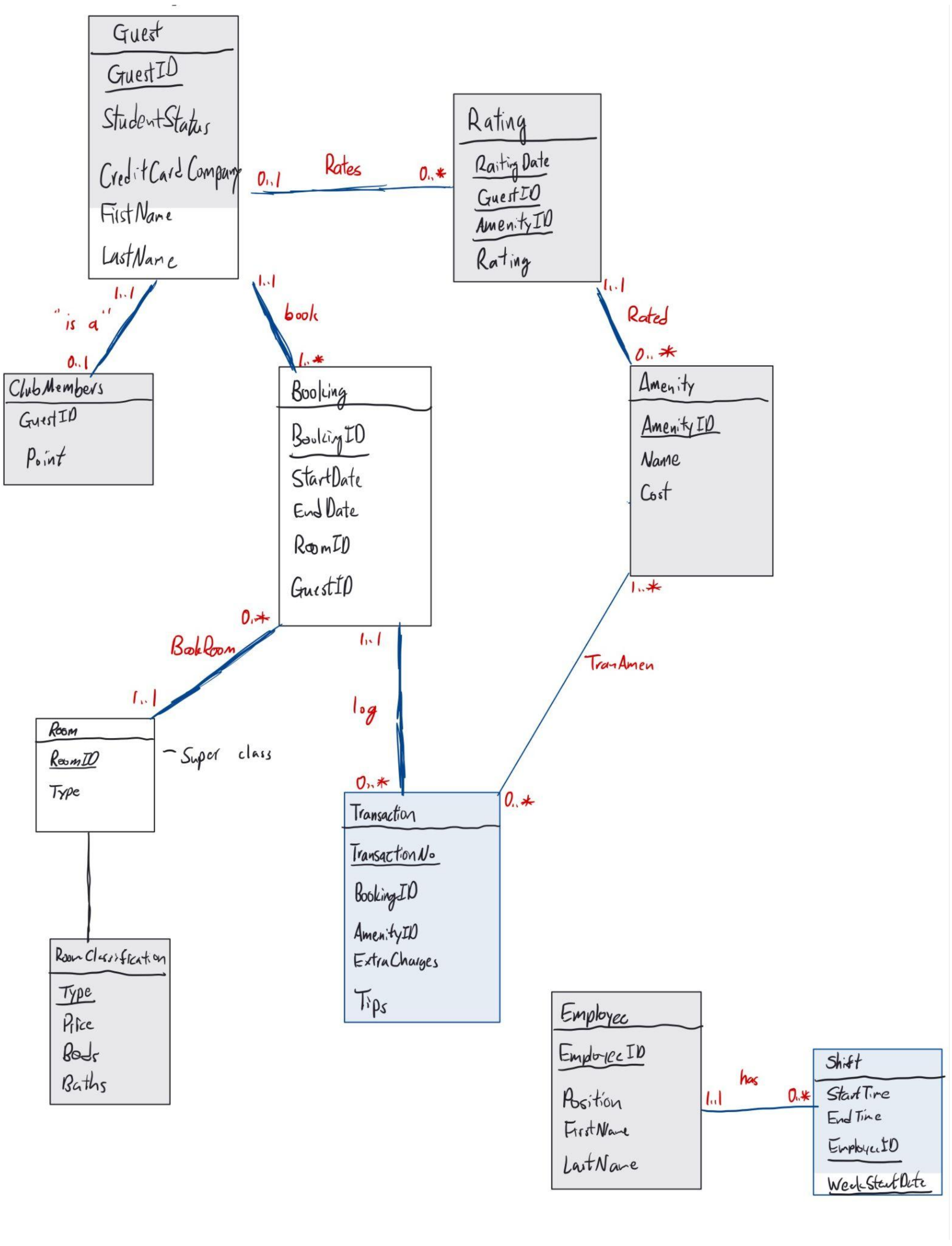
CSC 460: Database Design

Prof. Lester I. McCann, Ph.D.

Team Members: Gavin Pogson, Huy Le,
Noel Poothokaran, Mike Yu

Conceptual Database Design

E-R Diagram



High-level clarification

Students get a 10% discount. Another discount would be if you are a club member and have accumulated points. Each point would equal .01% off with a cap of 1000 points (10% off). These discounts are not able to be stacked and the higher one will always be chosen.

There are three different types of rooms. One is a Single and the other room is a Suite and the last one is a Deluxe Suite.

Amenities are not paid until the customers check out and get their whole bill back. This includes tips.

Logical database design

Table Schemas

Foreign Keys Italicized

Primary Keys Underlined

Guest

GuestID (integer) (cannot be null) (cannot be updated) (auto increment)

StudentStatus (number of length 1) (0-1) (String)

CreditCardCompany (String 10 char) (can be null)

FirstName (String 10 char) (cannot be null)

LastName (String 10 char) (cannot be null)

Constraint: STUDENTSTATUS HAS TO BE 0,1.

Rating

GuestID (Foreign key)

AmenityID (Foreign key)

RatingDate (partial key)

Rating (integer from 1 to 5)

ClubMember

GuestID (Foreign key)

Points (Integer) (Default is 0)

Booking

BookingID (Integer) (auto increment)

StartDate (DATE)

EndDate (DATE)

RoomID (Integer)

GuestID

Transaction

TransactionNo (auto increment)

BookingID

AmenityID

ExtraCharge

Tips

Room (Superclass)

RoomID
Type (Single, Suite, DeluxeSuite)

RoomClassification

Type
Price
Beds
Baths

Single (“Is a” Room)

RoomID
Price

Suite (“Is a” Room)

RoomID
Price
Beds
Baths

Employee

EmployeeID (integer) (auto increment)
Position
First Name
Last Name

Shift (Weak Entity Set depending on Employee)

EmployeeID
StartTime (MILITARY TIME, 4 integer)
EndTime (MILITARY TIME, 4 integer)
WeekStartDate

Constraint: TIME <= 0
TIME < 2400

Amenity

AmenityID (auto increment)
Name (String 10 char)
Price

Relationships:

1. Guest → Rating (0..1:0..M). Guest “rates” a rating.
2. Amenity → Rating (1..1:0..M). Amenity is “rated” by a rating.
3. Guest → ClubMember (1..1:0..1) A guest “is a” club member.
4. Guest -> Booking (1..1:1..M) A guest “books” one booking or more.

5. Booking → Room (0..M:1..1) Booking “BookRoom” Room
6. Employee -> Shift (1...1:0..M) Employee “EmployeeShift” a shift.
7. Booking -> Transaction (1..1:0..M). Booking “log” transactions.
8. Transaction -> Amenity (0..M:1..M). Transaction “TranAmen” Amenity.

Normalization analysis

Functional Dependencies

Guest

GuestID Determines StudentStatus, CreditCardCompany, FirstName, LastName

Rating

GuestID, AmenityID, RatingDate Determines Rating

ClubMembers

GuestID Determines Points

Booking

BookingID Determines StartDate, EndDate, *RoomID, GuestID*

Transaction

TransactionNo Determines: *BookingID, AmenityID,* ExtraCharge, Tips

Room

RoomID Determines Type

RoomClassification

Type Determines Price, Beds, Baths

Employee

EmployeeID Determines Position, First Name, Last Name

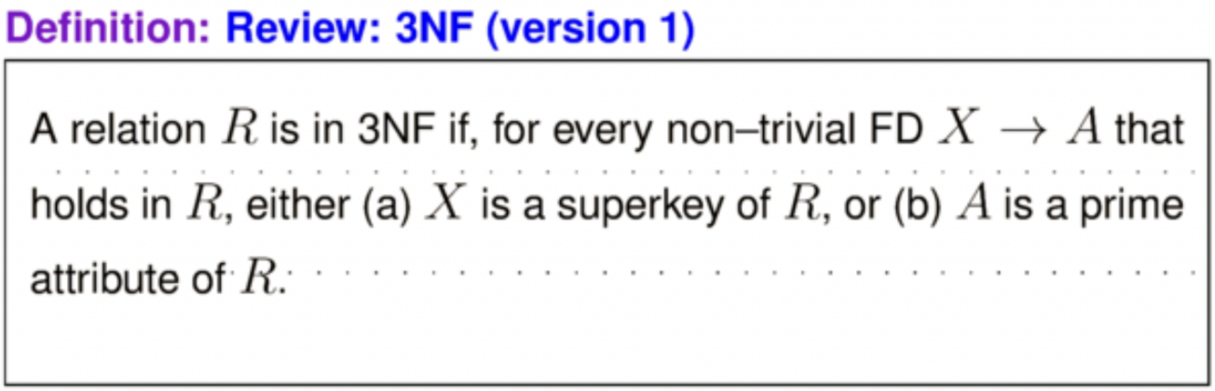
Shift

EmployeeID Determines StartTime, EndTime, WeekStartDate

Amenity

AmenityID Determines Name, Price

3rd Normal Form



Definition: Review: 3NF (version 1)

A relation R is in 3NF if, for every non–trivial FD $X \rightarrow A$ that holds in R, either (a) X is a superkey of R, or (b) A is a prime attribute of R.

With this definition, we can apply it to our Functional Dependencies

Guest

GuestID -> StudentStatus

GuestID -> CreditCardCompany

GuestID -> FirstName

GuestID -> LastName

For each of these FDs GuestId is a Super Key of Guest

It also is in 2NF since every non-prime Attr is Dependent on the CK

It also is in 1NF since the attr are not Set Valued

Rating

GuestID, AmenityID, RatingDate -> Rating

For each of these FDs *GuestID, AmenityID,* RatingDate is a Super Key of Rating

It also is in 2NF since ever non-prime Attr is Dependent on the CK

It also is in 1NF since the attr are not Set Valued

ClubMembers

GuestID Determines Points

For each of these FDs GuestId is a Super Key of ClubMembers

It also is in 2NF since every non-prime Attr is Dependent on the CK

It also is in 1NF since the attr are not Set Valued

Booking

BookingID -> StartDate

BookingID ->EndDate

BookingID -> *RoomID*

BookingID ->*GuestID*

For each of these FDs BookingId is a Super Key of Booking

It also is in 2NF since every non-prime Attr is Dependent on the CK

It also is in 1NF since the attr are not Set Valued

Transaction

TransactionNo -> *BookingID*

TransactionNo -> *AmenityID*

TransactionNo -> ExtraCharge

TransactionNo -> Tips

For each of these FDs TransactionNo is a Super Key of Transaction

It also is in 2NF since every non-prime Attr is Dependent on the CK

It also is in 1NF since the attr are not Set Valued

Room

RoomID -> Type

For each of these FDs RoomID is a Super Key of Room

It also is in 2NF since every non-prime Attr is Dependent on the CK

It also is in 1NF since the attr are not Set Valued

RoomClassification

Type -> Price

Type -> Beds

Type -> Baths

For each of these FDs Type is a Super Key of RoomClassification

It also is in 2NF since every non-prime Attr is Dependent on the CK

It also is in 1NF since the attr are not Set Valued

Employee

EmployeeID -> Position

EmployeeID -> First Name

EmployeeID -> Last Name

For each of these FDs EmployeID is a Super Key of Employee

It also is in 2NF since eveyr non-prime Attr is Dependent on the CK

It also is in 1NF since the attr are not Set Valued

Shift

EmployeeID -> StartTime

EmployeeID -> EndTime

EmployeeID -> WeekStartDate

For each of these FDs EmployeID is a Super Key of Shift

It also is in 2NF since every non-prime Attr is Dependent on the CK

It also is in 1NF since the attr are not Set Valued

Amenity

AmenityID -> Name

AmenityID -> Price

For each of these FDs AmenityID is a Super Key of Amenity

It also is in 2NF since every non-prime Attr is Dependent on the CK

It also is in 1NF since the attr are not Set Valued

Therefore all the tables adhere to 3NF

Query description

The question answered by the query: Who are the top X guests with the highest number of membership points?

(X is taken from the user, uses Guest and ClubMember tables)

This is useful since we can have competitions to see who has earned the most points and encourage guests to stay with Motel460 as much as possible. If the motel wants to reward their most loyal guests, say with a car for its 3 most loyal guests, this would be a useful query.