1. Logical schema for Continental Palms Hotel (CPH).

RoomType (name, description, roomSize)

Primary Key ( name )

Constraint: size BETWEEN 200 AND 900

RoomTypeAmenity (roomType, amenity)

Primary Key ( roomType, amenity )

Foreign Key ( roomType ) References RoomType ( name )

RoomPrice (roomType, pricingDate, price)

Primary Key ( roomType, pricingDate)

Foreign Key ( roomType ) References RoomType ( name )

Constraint: price BETWEEN 50 AND 400

Room (no, floor, wing, roomView, roomType)

Primary Key ( no )

Foreign Key ( roomType ) References RoomType ( name )

Constraint: roomType NOT NULL

Constraint: floor BETWEEN 1 AND 5

Constraint: wing IN ('E', 'W', 'C')

Constraint: roomView IN ('O', 'G', 'C')

RoomAmenity (room, amenity)

Primary Key ( room, amenity )

Foreign Key ( room ) References Room ( no )

Department (name, extension, fax, managerNo, managerCellphone, managerExtension)

Primary Key ( name )

Alternative Key (managerNo)

Foreign Key (managerNo) References Employee ( no )

Constraint: managerNo NOT NULL

Constraint: extension BETWEEN 100 AND 999

Constraint: managerExtension BETWEEN 100 AND 999

Employee (no, firstName, lastName, position, birthdate, gender, hireDate, department, supervisor)

Primary Key (no)

Foreign Key ( department ) References Department ( name )

Foreign Key ( supervisor ) References Employee ( no )

Constraint: department NOT NULL

Constraint: gender IN ('M', 'F')

Constraint: hireDate – birthdate > 18 years

Constraint: “The employee supervision relationship is hierarchical up to three levels.”

Guest (no, firstName, lastName, birthdate, street, city, state, zip, phone)

Primary Key (no)

Member (no, memberLevel, totalPoints, redeemedPoints)

Primary Key (no)

Foreign Key ( no ) References Guest ( no )

Constraint: memberLevel IN ('S', 'G', 'P')

CreditCard (no, type, expiration)

Primary Key (no)

GuestCreditCard (guestNo, creditCardNo, ownership)

Primary Key ( guestNo, creditCardNo )

Foreign Key ( guestNo ) References Guest ( no )

Foreign Key ( creditCardNo ) References CreditCard ( no )

Visit (confirmation, checkIn, checkOut, status, guestNo, creditCardNo, roomType, room)

Primary Key ( confirmation )

Foreign Key ( guestNo ) References Guest ( no )

Foreign Key ( creditCardNo ) References CreditCard ( no )

Foreign Key ( roomType ) References RoomType ( name )

Foreign Key ( room ) References Room ( no )

Constraint: guestNo NOT NULL

Constraint: creditCardNo NOT NULL

Constraint: roomType NOT NULL

Constraint: checkOut > checkIn

Constraint: status IN ('R', 'C', 'I', 'O')

Constraint: (status IN ('R', 'C')) OR (room IS NOT NULL)

Constraint: “A guest cannot make multiple visits on any given day.”

Constraint: “No room can be assigned to multiple guest visits on the same day.”

VisitRequest (confirmation, request)

Primary Key ( confirmation, request )

Foreign Key ( confirmation ) References Visit ( confirmation )

TransactionType (code, description, direction, enteringMethod)

Primary Key ( code )

Constraint: code BETWEEN 100 AND 999

Constraint: direction IN ('C', 'D')

Constraint: enteringMethod IN ('A', 'M')

Transaction (no, transactionDate, amount, memo, transactionType, confirmation, enteringEmployee, voidingEmployee, voidingDate, voidingReason)

Primary Key ( no )

Foreign Key ( transactionType ) References TransactionType ( code )

Foreign Key ( confirmation ) References Visit ( confirmation )

Foreign Key ( enteringEmployee ) References Employee ( no )

Foreign Key ( voidingEmployee ) References Employee ( no )

Constraint: transactionType NOT NULL

Constraint: confirmation NOT NULL

Constraint: no BETWEEN 1000000 AND 9999999

Constraint: amount > 0

Constraint: (voidingEmployee IS NULL) OR ((voidingDate IS NOT NULL) AND (voidingReason IS NOT NULL))

Constraint: “There is at least one room of every room type.”

Constraint: “There is at least one price for every room type.”

Constraint: “The manager of a department must be an employee in the department.”

Constraint: “The manager of a department does not have a supervisor.”

Constraint: “Every credit card belongs to at least one guest.”

Constraint: “Every guest has at least one credit card.”

Constraint: “The room number of a visit, if there is one, must be consistent with the room type.”

Constraint: “The credit card used for a visit must be one of the credit cards of the guest making the visit.”

Constraint: “The number of reservations for any room type on any day cannot exceed the number of rooms of that room type.”

Constraint: “If a transaction is of a type that is manually entered, the transaction must be entered by an employee.”

Constraint: “The date of a transaction must be between the check-in and check-out dates of the visit associated with the transaction.”

Constraint: “The voiding date of a voided transaction must be between the transaction date and the check-out date of the visit associated with the transaction.”

Constraint: “The transaction balance of every checked-out visit must be zero.”

1. Queries in relational algebra and tuple relational calculus.

* Get the guest number, first name, last name, and birthdate of every guest.

RA:

PROJECT firstName, lastName, birthDate (Guest)

TRC:

{G.firstName, G.lastName, G. birthDate | Guest(G)}

* Get the first name, last name, and birthdate of every guest from Wisconsin.

RA:

PROJECT fistName, lastName, birthDate (SELECT state = 'WI' (Guest))

TRC:

{G.firstName, G.lastName, G. birthDate | Guest(G) /\ G.state= 'WI'}

* Get the first name, last name, and birthdate of every member from Wisconsin.

RA:

PROJECT fistName, lastName, birthDate (SELECT state = 'WI' (Guest) JOIN Member)

TRC:

{G.firstName, G.lastName, G. birthDate | Guest(G) /\ G.state= 'WI' /\ (EXISTS M) (Member(M) /\ G.no=M.no)}

* Get the first name, last name, and birthdate of every member from Wisconsin who has stayed (including reserved visits but excluding canceled visits) in a luxury suite.

RA:

PROJECT fistName, lastName, birthDate (SELECT state='WI' (Guest) JOIN Member JOIN Member.no=Visit.guestNo (SELECT roomType='Luxury Suite' /\ status<>'C' (Visit)))

TRC:

{G.firstName, G.lastName, G. birthDate | Guest(G) /\ G.state= 'WI' /\ (EXISTS M) (EXISTS V) (Member(M) /\ Visit(V) /\ G.no=M.no /\ G.no=V.guestNo /\ V.roomType = 'Luxury Suite' /\ status <> 'C')}

1. CPH also hired another consulting group to design the database. Now you need to help them do the following:

* Give examples of data redundancies and data anomalies in their big table.

There are many data redundancies. For example, data about room types are repeated whenever there is a guest visit. This causes the following anomalies:

Insertion anomaly: Data about a room type cannot be entered until there is a guest visit of that type.

Update anomaly: Change of data about a room type must be consistently applied to all guest visits of that type.

Deletion anomaly: If all guest visits of a room type are deleted, data about that room type will be lost too.

* Determine the primary key of their big table.

{confirmation}

* Determine what normal form their big table is in.

The table is in 2NF, but not in 3NF. There are transitive dependencies.

* Rescue their design by normalizing their big table into multiple 3NF tables.

RoomType (roomType, roomTypeDescription, roomSize)

Primary Key (roomType)

Guest (guestNo, firstName, lastName, birthdate, street, city, state, zip, phone)

Primary Key (guestNo)

CreditCard (creditCardNo, creditCardType, expiration)

Primary Key (creditCardNo)

Visit (confirmation, checkIn, checkOut, status, guestNo, creditCardNo, roomType, room)

Primary Key ( confirmation )

Foreign Key (guestNo) References Guest (guestNo)

Foreign Key (creditCardNo) References CreditCard (creditCardNo)

Foreign Key ( roomType ) References RoomType (roomType)