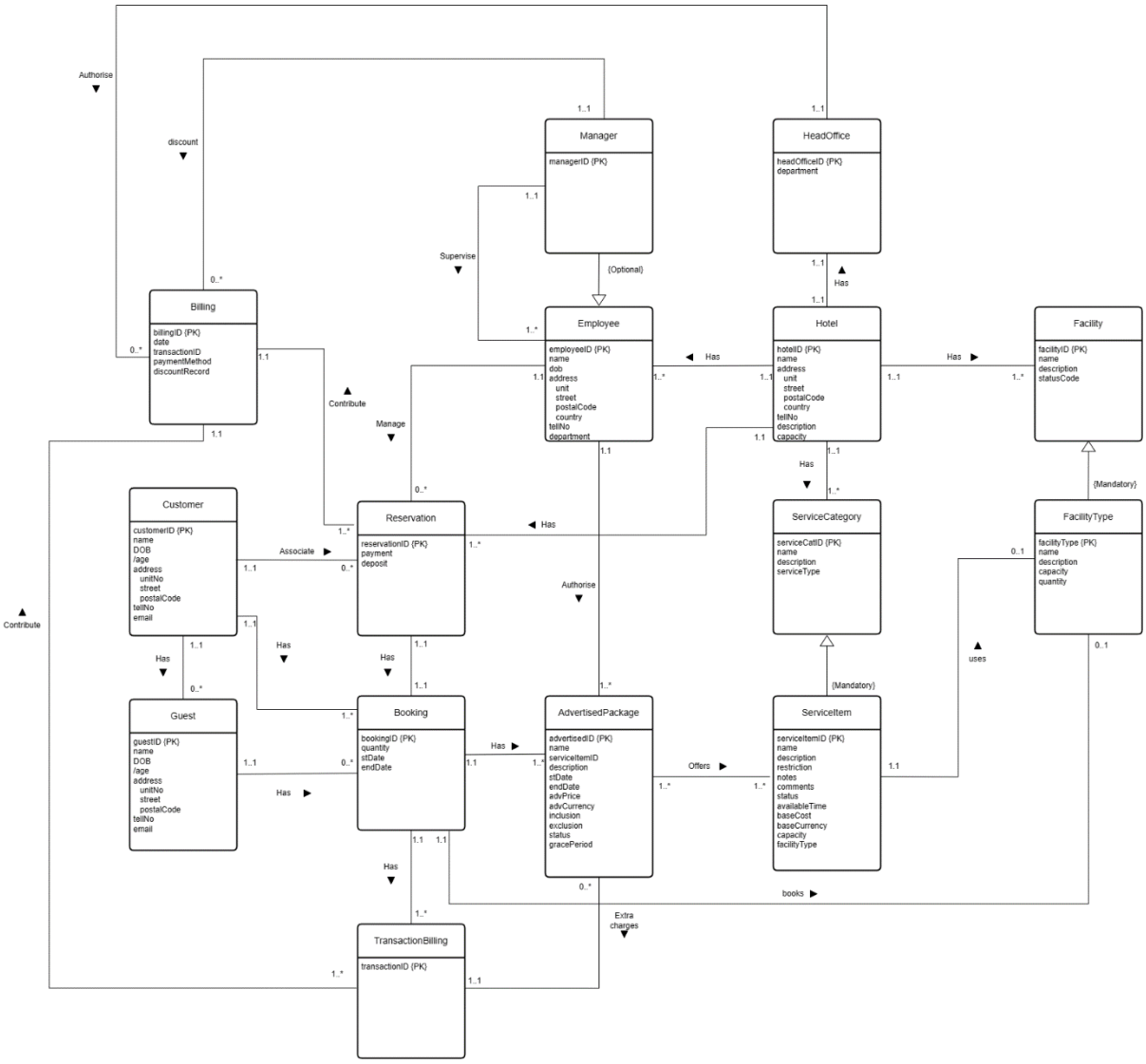
COMP3350 group 12 assignment 1 section 1-2

Group Members:

Victor Chua Jia Zhi Student ID: c3418248

Sander Fabian Student ID: c3418982

EER MODEL



Question 2: Normalised BCNF Relational schema

Justification:

We have decided to place the constraints of the FK majorly as NO UPDATE NO ACTION and NO DELETE NO ACTION due to countless testing which SQL do not accept many foreign key constraints as it might cost error which will be set as default. We have consulted with the lecturer, and he acknowledged.

Capacity(capacityID, name, size)

Primary key capacityID

Justification: We have separated the capacity attribute into a class so it does not violate the 3NF where they shouldn't be any transitive dependency.

Location(locationID, city, country)

Primary Key locationID

Justification: After many testings, we decided to separate location into its own class as it is more efficient and comply with the normalisation.

HotelAddress(hotelAddressID, unitNo, street, postalCode, locationID)

Primary key hotelAddressID

Foreign key locationID references Location(locationID)

ON UPDATE NO ACTION, ON DELETE NO ACTION

Justification: We have separated Hotel address as finding data for hotel address would be more efficient and would not have many foreign key constraint issues as originally, we put employee, hotel, guest, and customer into one address table which can cost errors when populating data.

Hotel (hotelID, name, description, tellNo, capacityID, hotelAddressID)

Primary Key hotelID

Foreign Key hotelAddressID references HotelAddress(hotelAddressID)

ON UPDATE NO ACTION, ON DELETE NO ACTION

Foreign Key capacityID reference Capacity(capacityID)

ON UPDATE CASCADE, ON DELETE NO ACTION

Department(departmentID, name, area)

Primary Key DepartmentID

Justification: We have decided to separate the department table as it would be more efficient and would not violate any normalisation.

EmployeeAddress(employeeAddressID, unitNo, street, postalCode, locationID)

Foreign key locationID references Location(locationID)

ON UPDATE NO ACTION, ON DELETE NO ACTION

Justification: as previously mentioned, we decided to separate the address tables.

Employee (employeeID, name, DOB, tellNo, employeeAddressID, hotelID, departmentID)

Primary Key employeeID

Foreign Key employeeAddressID references to EmployeeAddress(employeeAddressID)

ON UPDATE NO ACTION, ON DELETE NO ACTION

Foreign Key departmentID references Department(departmentID)

ON UPDATE CASCADE, ON DELETE NO ACTION

Foreign Key hotelID references Hotel (hotelID)

ON UPDATE CASCADE, ON DELETE NO ACTION

Manager (managerID, employeeID)

Primary Key managerID

Foreign Key employeeID references Employee(employeeID)

ON UPDATE CASCADE, ON DELETE CASCADE

HeadOffice(headOfficeID, departmentID)

Primary Key headOfficeID

Foreign Key departmentID references Department(departmentID)

ON UPDATE CASCADE, ON DELETE NO ACTION

Facility(facilityID, name, description, statusCode, hotelID)

Primary Key facilityID

Foreign Key hotelID references Hotel(hotelID)

ON UPDATE CASCADE, ON DELETE NO ACTION

FacilityType(facilityTypeID, name, description, quantity, capacity, facilityID)

Primary Key facilityTypeID

Foreign Key capacity references Capacity(capacityID)

ON UPDATE NO ACTION, ON DELETE NO ACTION

Foreign Key facilityID references Facility(facilityID)

ON UPDATE CASCADE, ON DELETE CASCADE

ServiceCategory(serviceCatID, name, description, serviceType, hotelID)

Primary Key serviceCatID

Foreign key hotelID references Hotel(hotelID)

ON UPDATE CASCADE, ON DELETE CASCADE

baseCurrency(currencySymbol, name)

Primary Key currencySymbol

Justification: We have decided to separate base currency as it should be a individual table that can be used to references to other tables for data integrity sake.

ServiceItem(serviceItemID, name, description, restriction, notes, comments, availableTime, quantity, baseCost, status, capacityID, baseCurrency, facilityTypeID, serviceCatID)

Primary Key serviceItemID

Foreign Key capacity references to Capacity(capacityID)

ON UPDATE NO ACTION, ON DELETE NO ACTION

Foreign Key baseCurrency references BaseCurrency(currencySymbol)

ON UPDATE NO ACTION, ON DELETE NO ACTION

Foreign Key facilityTypeID references FacilityType(facilityTypeID)

ON UPDATE CASCADE, ON DELETE CASCADE

Foreign Key serviceCatID references ServiceCategory(serviceCatID)

ON UPDATE NO ACTION, ON DELETE NO ACTION

AdvertisedPackage(adPackID, name, description, price , inclusion, exclusion, , startDate, endDate, statusCode, employeeID, baseCurrency)

Primary Key adPackID

Foreign Key employeeID references Employee(employeeID)

ON UPDATE NO ACTION, ON DELETE NO ACTION

Foreign Key baseCurrency references BaseCurrency(currencySymbol)

ON UPDATE NO ACTION, ON DELETE NO ACTION

AdvertisedService(adPackID, serviceItemID)

Primary Key (adPackID, serviceItemID)

Foreign Key adPackID references AdvertisedPackage(adPackID)

ON UPDATE CASCADE, ON DELETE CASCADE

Foreign Key serviceItemID references ServiceItem(serviceItemID)

ON UPDATE CASCADE, ON DELETE CASCADE

Justification: Since, we assumed that advertised packages can have many service items, we decided to put them into a composite table where the user is able to see what kind of advertised packages are associated with service items.

GracePeriod (gracePeriodID, durationDays ,adPackID)

Primary Key GracePeriodID

Foreign Key adPackID references AdvertisedPackage(adPackID)

ON UPDATE CASCADE, ON DELETE CASCADE

Justification: We decided to remove grace period from AdvertisedPackage as it can be dependent on the dates and this way, it would be more efficient to use.

CustomerGuesAddress(customerAddressID, unitNo, street, postalCode, locationID)

Foreign key locationID references Location(locationID)

ON UPDATE NO ACTION, ON DELETE NO ACTION

Justification: As previously mentioned, we have separated the address table and for this table it accepts the guest and customer addresses into one as their guest might be family members staying with them.

Guest(guestID, name, DOB, tellNo, email, customerID, customerAddressID)

Primary Key guestID

Foreign Key customerAddressID references CustomerGuestAddress(customerAddressID)

ON UPDATE NO ACTION, ON DELETE NO ACTION

Foreign Key customerID references Customer(customerID)

ON UPDATE CASCADE, ON DELETE NO ACTION

Justification: To check which guest is associated with which customer, we have included a Customer ID where it shows the relationship of the guest to customer

Customer(customerID, name, DOB, email, tellNo, customerGuestAddressID)

Primary Key customerID

Foreign Key customerGuestAddressID reference CustomerGuestAddress (customerGuestAddressID)

ON UPDATE CASCADE, ON DELETE NO ACTION

Reservation(reservationID, payment, deposit, customerID, employeeID, hotelID)

Primary Key reservationID

Foreign Key customerID references Customer(customerID)

ON UPDATE CASCADE, ON DELETE NO ACTION

Foreign Key employeeID references Employee(employeeID)

ON UPDATE CASCADE, ON DELETE NO ACTION

Foreign key hotelID references Hotel(hotelID)

ON UPDATE NO ACTION, ON DELETE NO ACTION

Booking(bookingID, quantity, startDate, endDate, reservationID, adPackID, customerID, facilityTypeID)

Primary Key bookingID

Foreign Key reservationID references Reservation(reservationID)

ON UPDATE NO ACTION, ON DELETE NO ACTION

Foreign Key adPackID references AdvertisedPackage(adPackID)

ON UPDATE NO ACTION, ON DELETE NO ACTION

Foreign Key customerID references Customer(customerID)

ON UPDATE CASCADE, ON DELETE NO ACTION

Foreign Key facilityTypeID references facilityType(facilityTypeID)

ON UPDATE CASCADE, ON DELETE NO ACTION

TransactionBilling (transactionID, bookingID, adPackID)

Primary Key transactionID

Foreign Key bookingID references Booking(bookingID)

ON UPDATE NO ACTION, ON DELETE NO ACTION

Foreign Key adPackID references AdvertisedPackage(adPackID)

ON UPDATE NO ACTION, ON DELETE NO ACTION

Discount(discountID, managerID, headOfficeID, discountRecord)

Primary Key: discountID

Foreign Key managerID references Manager(managerID)

ON UPDATE CASCADE, ON DELETE NO ACTION

Foreign Key headOfficeID references HeadOffice(headOfficeID)

ON UPDATE NO ACTION, ON DELETE NO ACTION

Billing(billingID, paymentDate, paymentMethod, reservationID, customerID, discountID, bookingID, adPackID)

Primary Key: billingID

Foreign Key reservationID references Reservation(reservationID)

ON UPDATE NO ACTION, ON DELETE NO ACTION

Foreign Key customerID references Customer(customerID)

ON UPDATE NO ACTION, ON DELETE NO ACTION

Foreign Key discountID references Discount(discountID)

ON UPDATE CASCADE, ON DELETE NO ACTION

Foreign Key (bookingID, adPackID) references TransactionBilling(bookingID, adPackID)

ON UPDATE CASCADE, ON DELETE NO ACTION