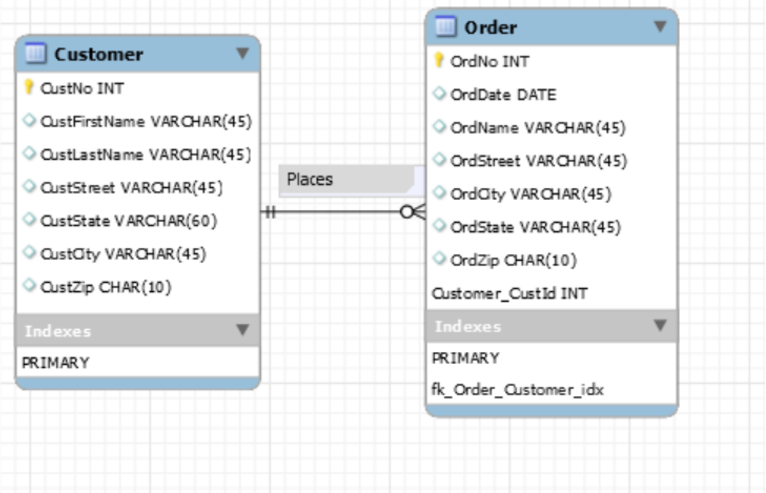
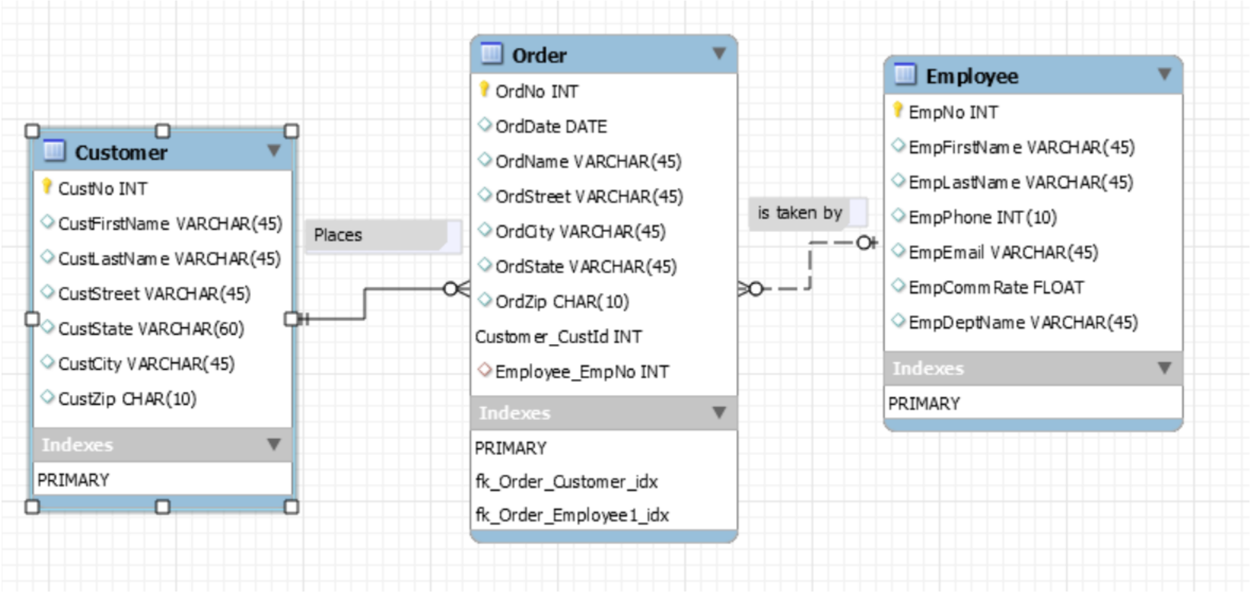
Assignment 2

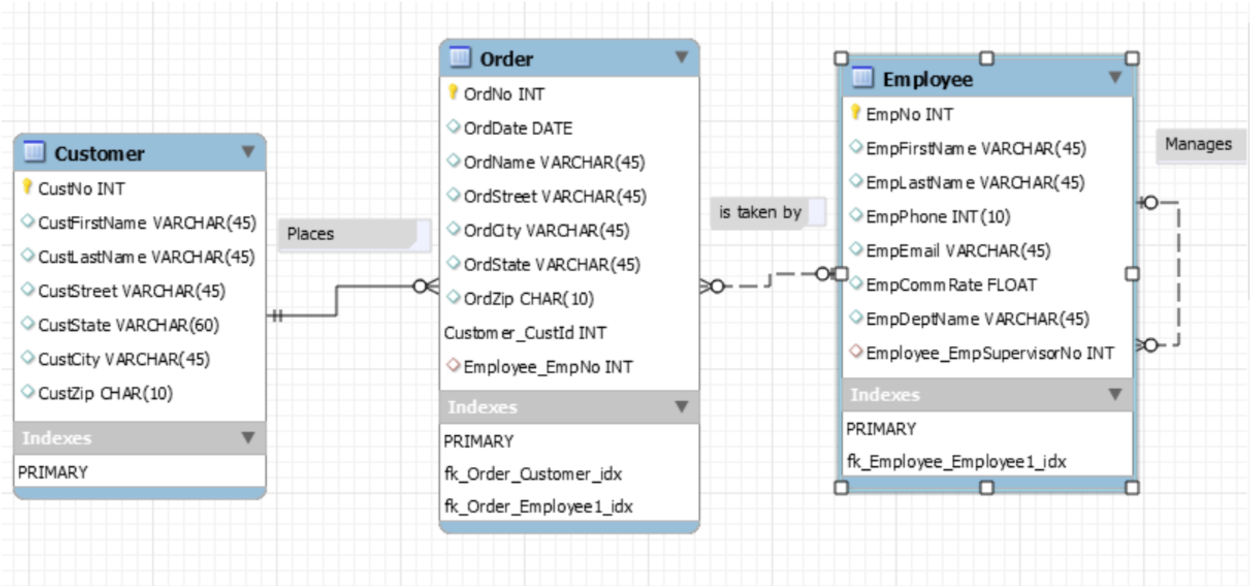
1. Draw an ERD containing the Order and Customer entity types connected by a 1-M relationship from Customer to Order. Choose an appropriate relationship name using your common knowledge of interactions between customers and orders. Define minimum cardinalities so that an order is optional for a customer and a customer is mandatory for an order. For the Customer entity type, add attributes CustNo (primary key), CustFirstName, CustLastName, CustStreet, CustCity, CustState, CustZip, and CustBal (balance). For the Order entity type, add attributes for the OrdNo (primary key), OrdDate, OrdNamee, OrdStreet, OrdCity, OrdState, and OrdZip. If you are using a data modeling tool that supports data type specification, choose appropriate data types for the attributes based on your common knowledge.



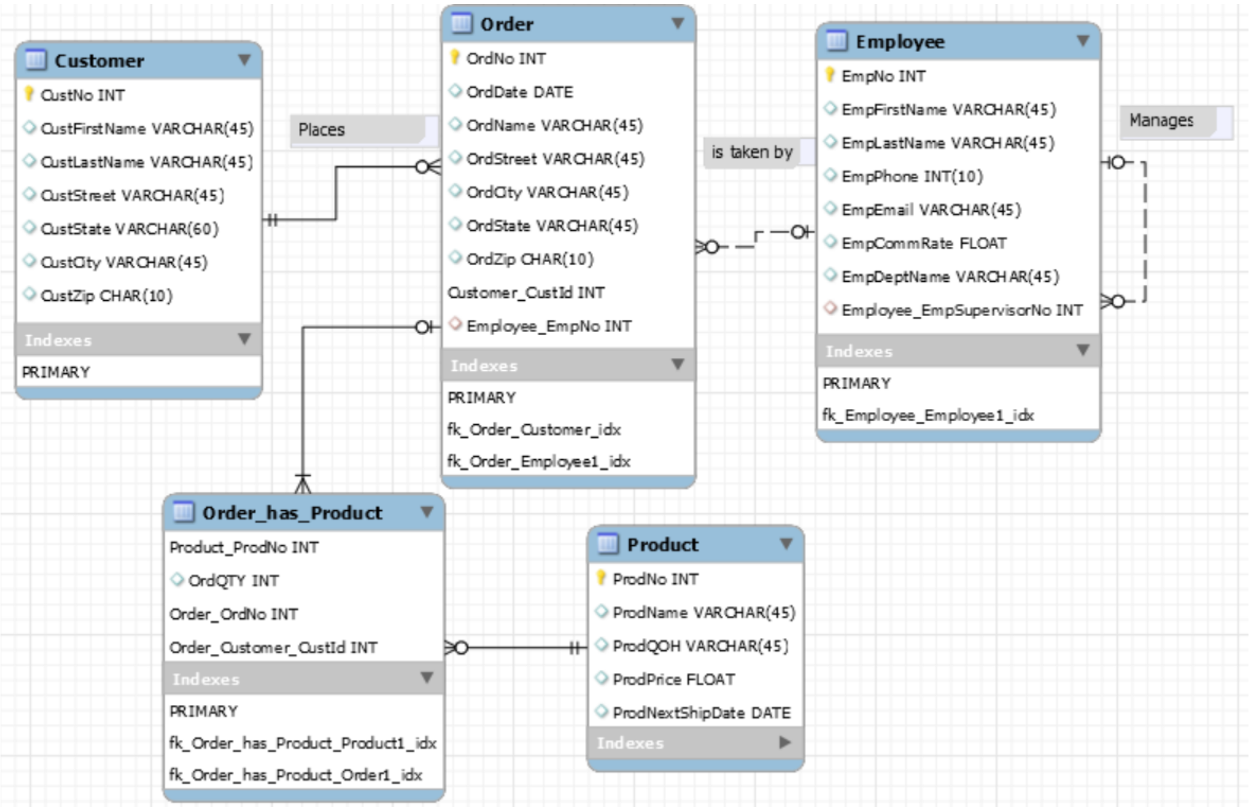
2. Extend the ERD from problem 1 with the Employee entity type and a 1-M relationship from Employee to Order. Choose an appropriate relationship name using your common knowledge of interactions between employees and orders. Define minimum cardinalities so that an employee is optional to an order and an order is optional to an employee. For the Employee entity type, add attributes EmpNo (primary key), EmpFirstName, EmpLastName, EmpPhone, EmpEmail, EmpCommRate (commission rate), and EmpDeptName. If you are using a data modeling tool that supports data type specification, choose appropriate data types for the attributes based on your common knowledge.



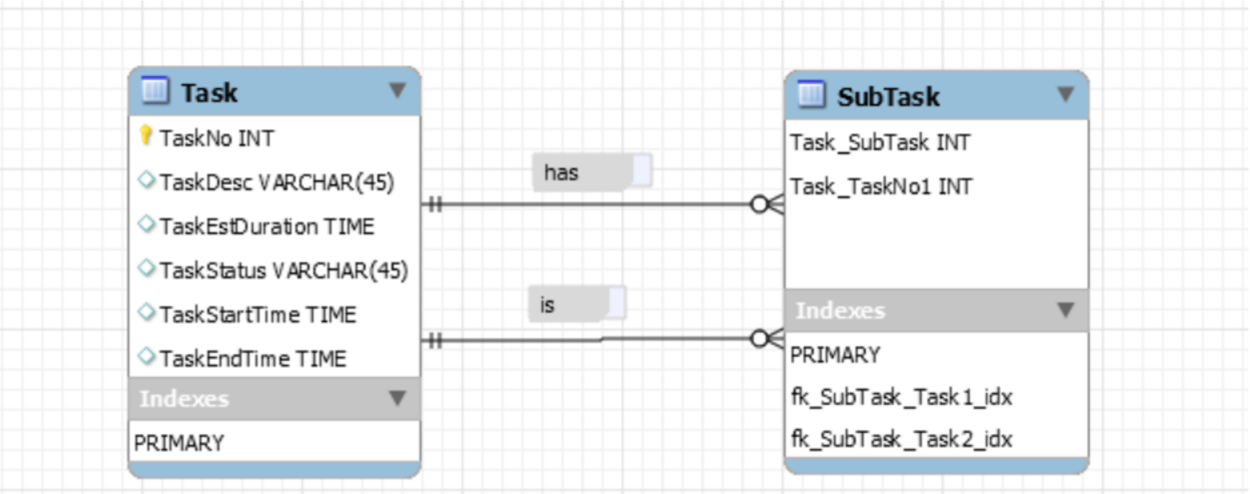
3. Extend the ERD from problem 2 with a self-referencing 1-M relationship involving the Employee entity type. Choose an appropriate relationship name using your common knowledge of organizational relationships among employees. Define minimum cardinalities so that the relationship is optional in both directions.



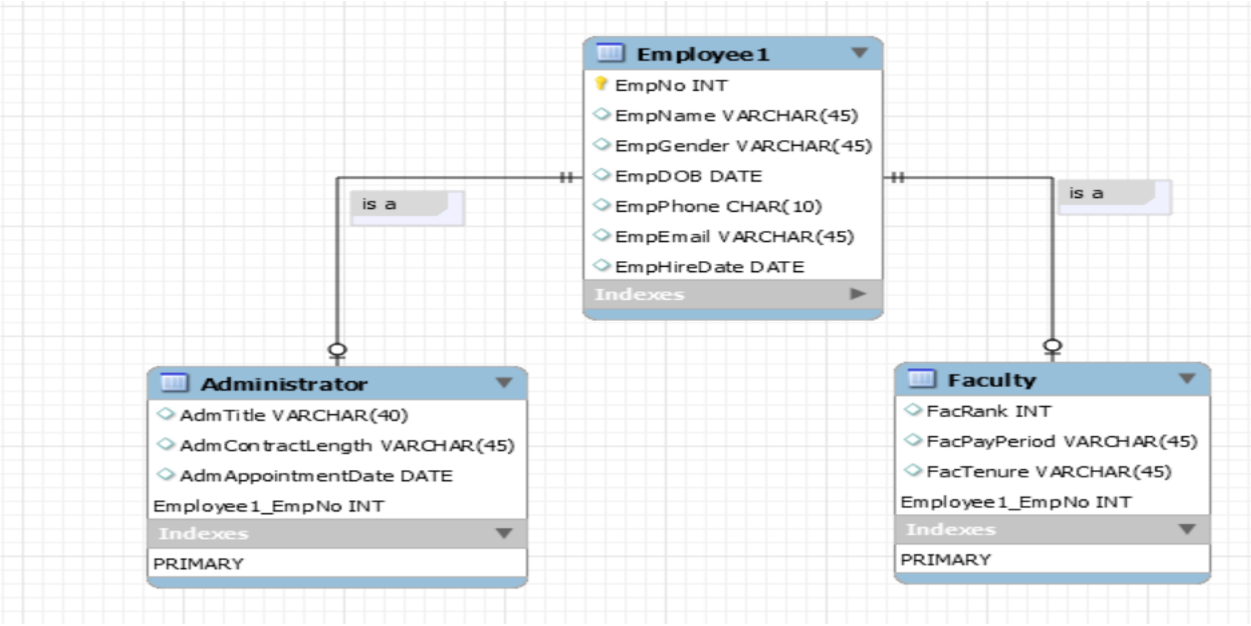
4. Extend the ERD from problem 3 with the Product entity type and an M-N relationship between Product and Order. Choose an appropriate relationship name using your common knowledge of connections between products and orders. Define minimum cardinalities so that an order is optional to a product, and a product is mandatory to an order. For the Product entity type, add attributes ProdNo (primary key), ProdName, ProdQOH, ProdPrice, and ProdNextShipDate. For the M-N relationship, add an attribute for the order quantity. If you are using a data modeling tool that supports data type specification, choose appropriate data types for the attributes based on your common knowledge. Revise ERD by transforming the M-N relationship into an associative entity type and two identifying, 1-M relationships.



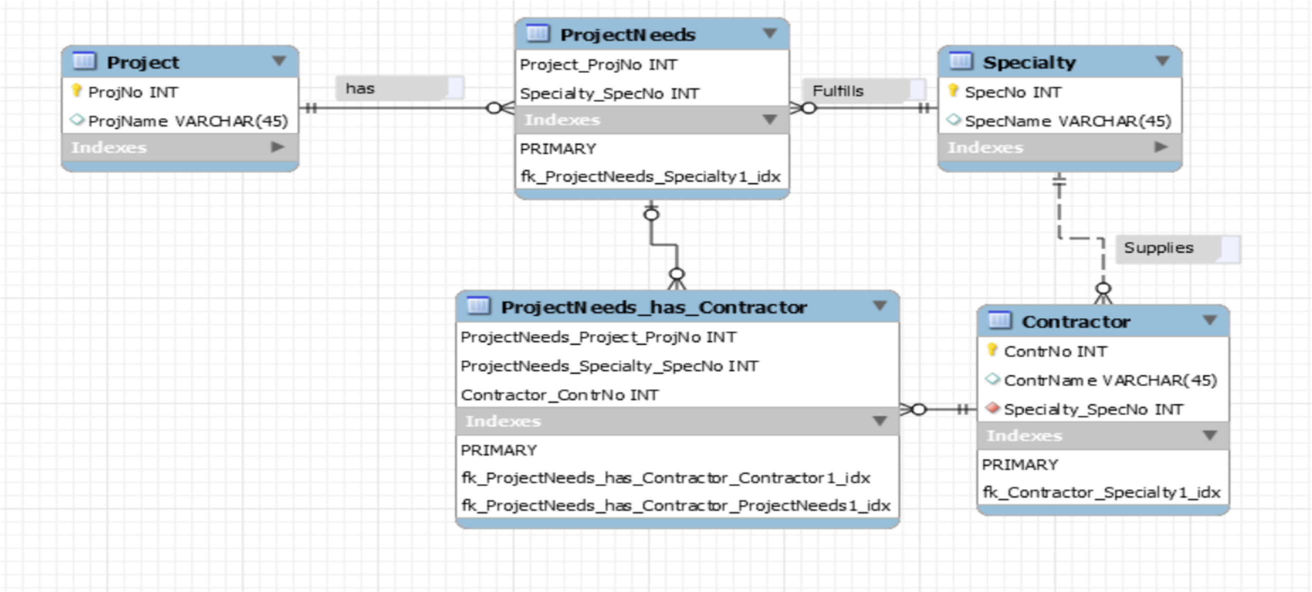
5. Design an ERD for the Task entity type and an M-N self-referencing relationship. For the Task entity type, add attributes TaskNo (primary key), TaskDesc, TaskEstDuration, TaskStatus, TaskStartTilne, and TaskEndTin1e. Choose an appropriate relationship name using your common knowledge of precedence connections among tasks. Define minimum cardinalities so that the relationship is optional in both directions.



6. Define a generalization hierarchy containing the Employee entity type, the Faculty entity type, and the Administrator entity type. The Employee entity type is the supertype and Faculty and Administrator are subtypes. The Employee entity type has attributes EmpNo (primary key), EmpName, EmpGender, EmpDOB (date of birth), EmpPhone, EmpEmail, and EmpHireDate. The Faculty entity type has attributes FacRank, FacPayPeriods, and FacTenure. The Administrator entity type has attributes AdmTitle, AdmContractLength, and AdmAppointmentDate.



7. Figure 6.CP8 (ERD for Conversion Problem 8 in Textbook Page 230) is incomplete in that foreign keys are missing and there is a many-to-many relationship. Draw a complete ERD for it by adding foreign keys and converting the many-to-many relationship into an entity with relationships. Be sure to designate the primary key of every entity and add all foreign keys.



8. Convert the previous question's ERD into a relational database. The result of conversion is a collection of "CREATE TABLE" commands or simplified commands

Project (ProjNo, ProjName)

Specialty (SpecNo, SpecName)

Contractor (ContrNo, SpecNo, ContrName)

Contractor.SpecNo references Specialty.SpecNo.

ProjectNeeds (ProjectNo, SpecNo)

ProjectNeeds.ProjNo references Project.ProjNo.

ProjectNeeds.SpecNo references Specialty.SpecNo.

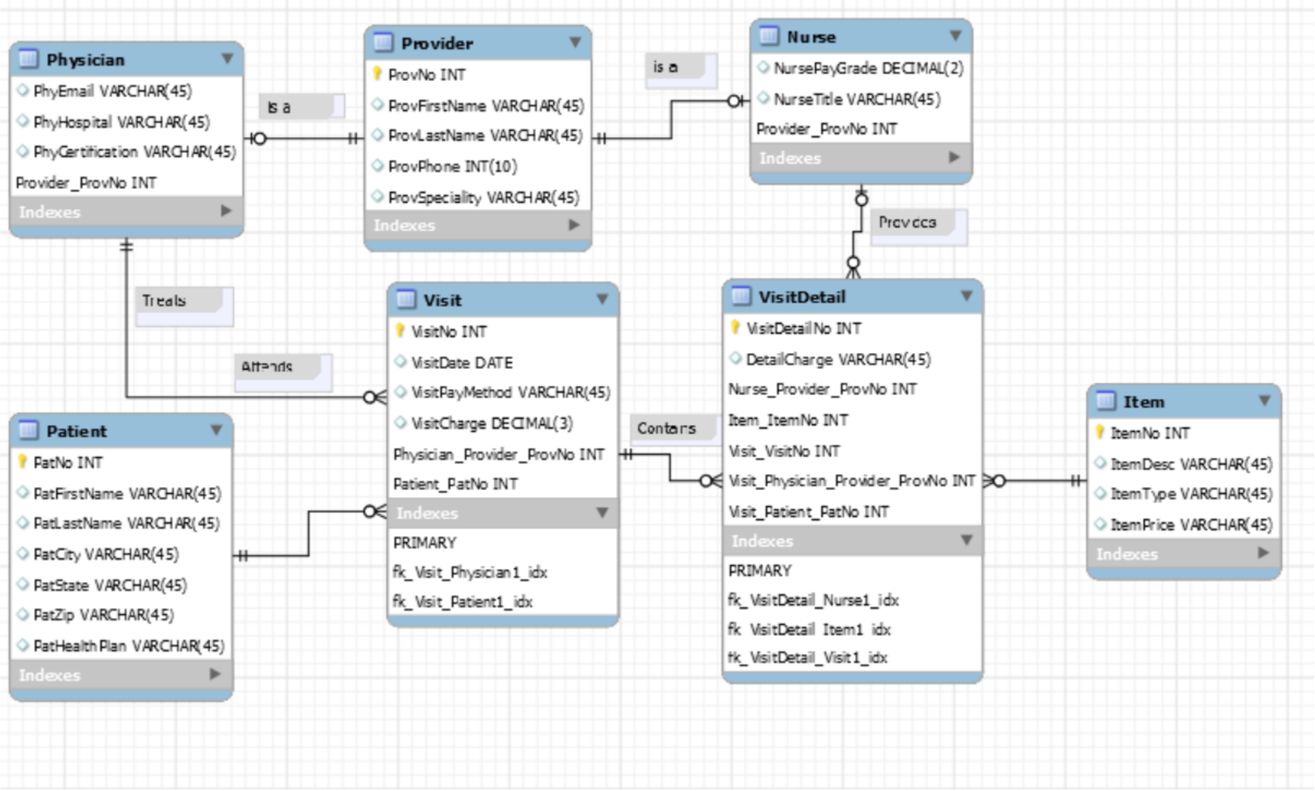
ProjectNeeds_has_Contractor (ProjNo, SpecNo, ContrNo)

ProjectNeeds_has_Contractor.ProjNo references Project.ProjNo.

ProjectNeeds_has_Contractor.SpecNo references Specialty.SpecNo.

ProjectNeeds_has_Contractor.ContrNo references Contractor.ContrNo

9. Figure 6.C10 (ERD for Conversion Problem 10 in Textbook Page 231) is incomplete in that foreign keys are missing. Draw a complete ERD for it by adding foreign keys. Be sure to designate the primary key of every entity and add all foreign keys.



10. Convert the previous question's ERD into a relational database. The result of conversion is a collection of "CREATE TABLE" commands or simplified commands.

Provider (ProvNo, ProvFirstName, ProvLastName, ProvPhone, ProvSpecialty)

Physician (ProvNo, PhyEmail, PhyHospital, PhyCertification)

Nurse (ProvNo, NursePayGrade, NurseTitle)

Physician.ProvNo references Provider.ProvNo

Nurse.ProvNo references Provider.ProvNo

Patient (PatNo, PatFirstName, PatLastName, PatCity, PatState, PatZip, PatHealthPlan)

Visit (VisitNo, ProvNo, PatNo, VisitDate, VisitPayMethod, VisitCharge)

Visit.ProvNo references Provider.ProvNo

Visit.PatNo references Patient.PatNo

Item (ItemNo, ItemDesc, ItemType, ItemPrice)

VisitDetail (VisitDetailNo, ProvNo, ItemNo, VisitNo, DetailCharge)

VisitDetail.ProvNo references Provider.ProvNo

VisitDetail.ItemNo references Item.ItemNo

VisitDetail.VisitNo references Visit.VisitNo