Laboratory 2
ER Modelling and Relational Table Transformation

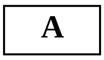
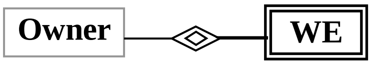
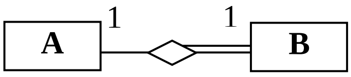
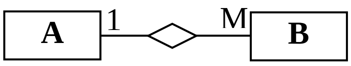
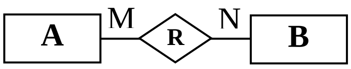
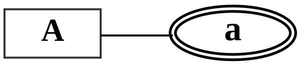
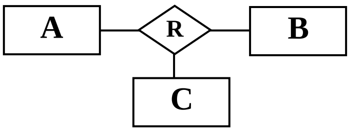
References

Lecture Notes: Topics 2 and 3 – Elmasri and Navathe, 2017: Chapters 3 and 9

In this class you will learn to model given database problems using the Entity-Relationship modelling technique and how to transform an ER-Model into Relational Tables. You are to use the notation shown in the lecture notes for the Rental-Property example.

Helpful Tips:

1. **Relational Table Transformation – 7 Steps**

Step	ER-Model		Relational Model Transformation
1	**Entity**		**1 table A** **+ attributes** **+ PK**
2	**Weak entity**		**1 table WE** **+ attributes** **+ [Owner.PK + WE.key]**
3	**1:1 relationship**		**2 tables A, B** **+ A.PK is FK in B**
4	**1:N relationship**		**2 tables A, B** **+ A.PK is FK in B**
5	**M:N relationship**		**3 tables A, B, R** **+ <A.PK, B.PK> is PK in R**
6	**Multi-valued attribute**		**2 tables A, Aa** **+ <A.PK + a> is PK in Aa**
7	**n-ary relationship**		**(n + 1) tables A, B, C, R** **+ <A.PK, B.PK, C.PK> is PK in R**

2. **Foreign key is not drawn in the ER-Model, only when you transform it into tables, FK will be added into tables based on the ER-Model transformation rules.**

3. **After the transformation, do not forget to list the final tables.**

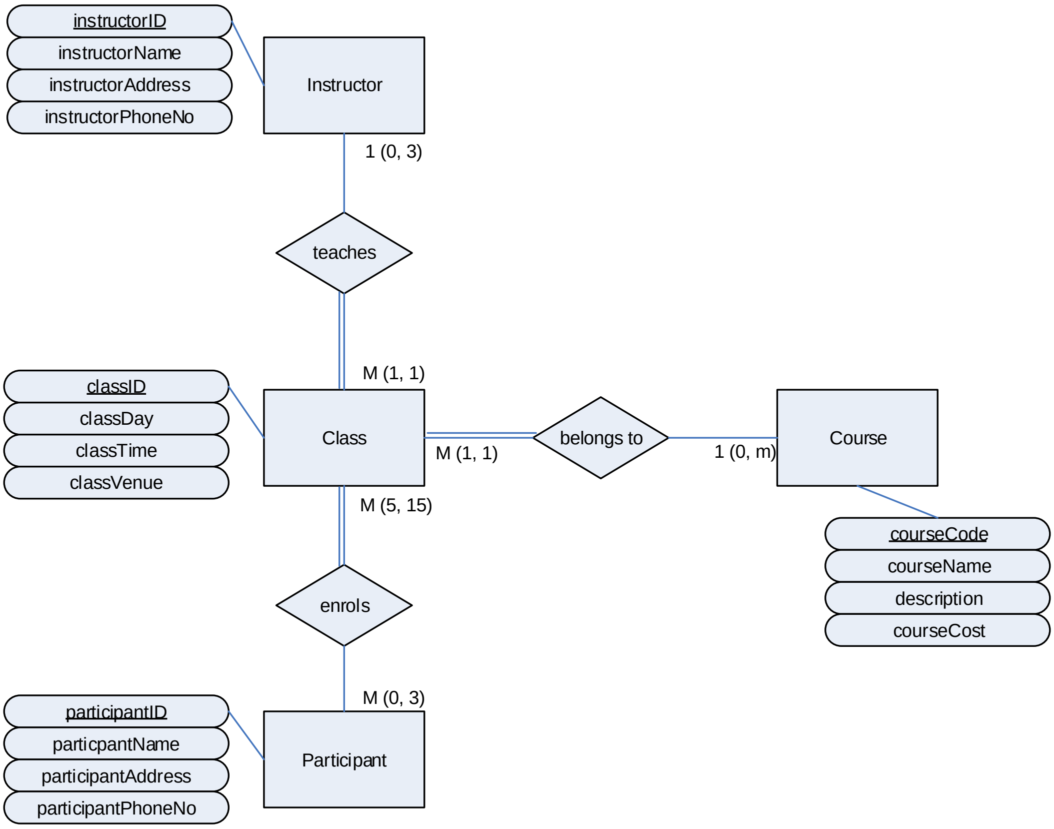
Exercise 1 – Practice with cardinalities, participation and (min, max)

a) Construct an ER-Model for the following problem description. State any assumptions that you make.(including entity, attribute, pk, relationship, cardinality and participation)

The Municipal Arts Centre (MAC) runs creative art courses for residents of the local community. They require a database to keep track of course and current enrolment details. At present they run five courses: Pottery; Photography; Painting with Watercolours; Oil Painting; Needlework; and Doll Making. Each course is offered three times a year, once in each term. Information to be kept on courses includes: the course code, course title, course description and course cost. The cost of each course is fixed across all terms. Each course may have a number of different class times per week for residents to choose from. A class will only go ahead if it has at least five people enrolled. The maximum number of people a class can cater for is 15. The class venue, time and day of the week need to be stored.

Each class is run by one instructor. Each instructor may teach up to three classes, however if there are insufficient enrolments they may not be required for a particular term. The name, address and telephone number of each instructor need to be stored. Each resident may take up to three courses per term. Once they have enrolled in one course their details are to be kept on the system so they can be sent advertisements of future courses.

Answer:



NOTE: (1) the solution is to model the problem on "one-term" basis; (2) For each course they are taking, each resident needs only to attend one class.

b) Transform your ER-model into relational tables. Make sure you follow the 7-steps and list your final tables at the end.

Answer:

Step 1:
INSTRUCTOR (instructorID, instructorName, instructorAddress, instructorPhoneNo)
CLASS (classID, classDay, classTime, classVenue)
PARTICIPANT (participantID, participantName, participantAddress, participantPhoneNo)
COURSE (courseCode, courseName, courseDescription, courseCost)

Step 2: No weak entities

Step 3: No 1:1 relations

Step 4:
CLASS (classID, classDay, classTime, classVenue, *instructorID*, *courseCode*)

Step 5:
ENROL (*classID*, *participantID*)

Step 6: No multi-valued attributes

Step 7: No n-ary relationships

FINAL TABLES
INSTRUCTOR (instructorID, instructorName, instructorAddress, instructorPhoneNo)
PARTICIPANT (participantID, participantName, participantAddress, participantPhoneNo)
COURSE (courseCode, courseName, courseDescription, courseCost)
CLASS (classID, classDay, classTime, classVenue, *instructorID*, *courseCode*)
ENROL (*classID*, *participantID*)

Exercise 2 – weak entities, multi-valued attributes and different degrees of relationship (unary, binary, etc)

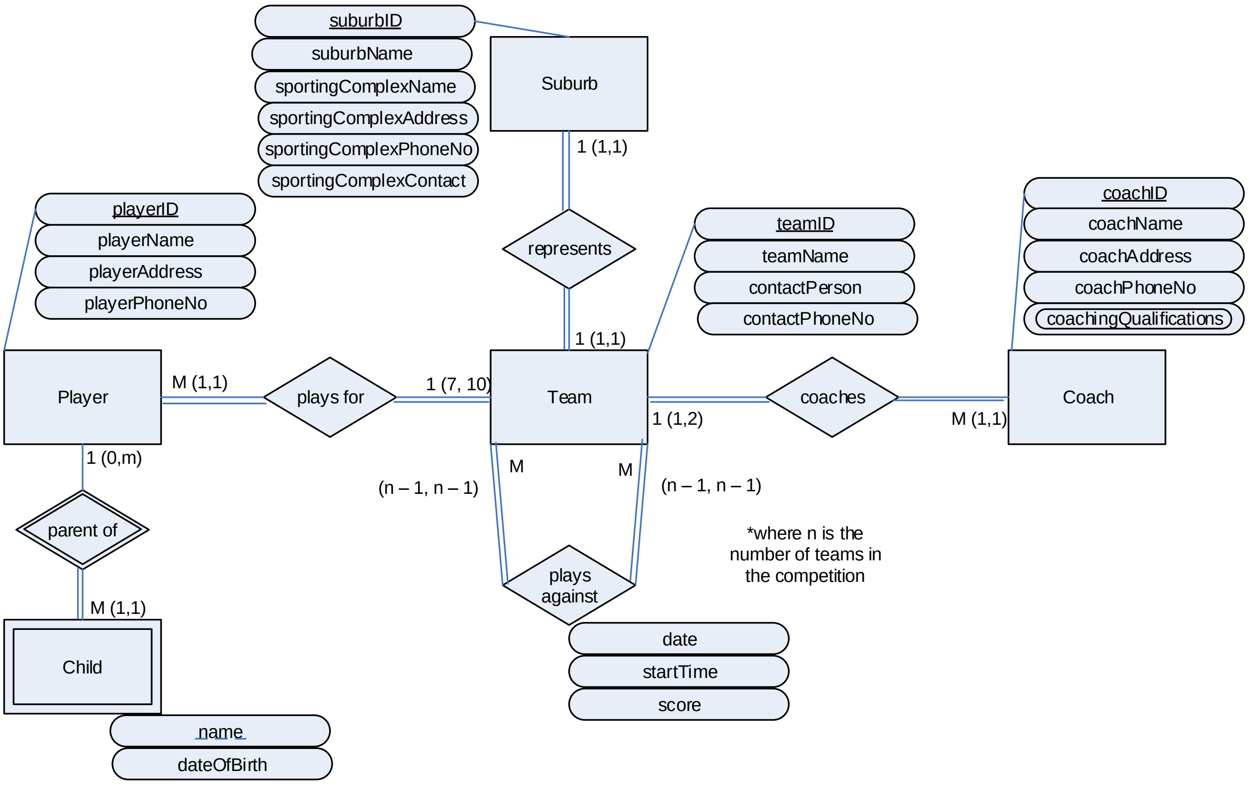
a) Construct an ER-Model for the following problem description. State any assumptions that you make.

DSS (Dig, Set, Spike) Volleyball is an amateur volleyball association. Each team in the association represents a particular suburb. Each suburb has its own sporting complex where the team's home games are played. Teams are identified by their id and also have a name, a contact person and a contact phone number. Each team has a maximum of ten players and a minimum of seven players registered. Players are identified by a unique player id and their name, address and contact phone number need to be recorded. Each team has a coach; some have up to two coaches. Coaches also have unique ids, and their name, address, contact phone number and coaching qualifications are required.

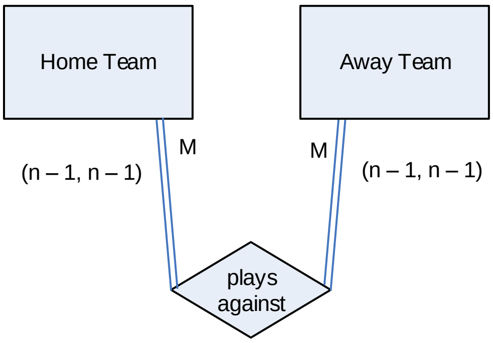
During the season, each team plays two games, one as the home team and one as the visitor, against each of the other teams. For each game the date of the game, start time and final score need to be recorded.

As a special treat at the end of the season the association holds a break-up party for the players. At the break-up party treats are given out to all the players' children who are under 12. The association therefore requires each child's name and date of birth.

Answer:



Try to view the unary relationship like this binary relationship:



b) Transform your ER-model into relational tables. Make sure you follow the 7-steps and list your final tables at the end.

Answer:

Step 1:
SUBURB (suburbID, suburbName, sportingComplexName, sportingComplexAddress, sportingComplexPhoneNo, sportingComplexContact)
TEAM (teamID, teamName, contactPerson, contactPhoneNo)
COACH (coachID, coachName, coachAddress, coachPhoneNo)
PLAYER (playerID, playerName, playerAddress, playerPhoneNo)

Step 2:
CHILD (*playerID*, name, dateOfBirth)

Step 3:
SUBURB (suburbID, suburbName, sportingComplexName, sportingComplexAddress, sportingComplexPhoneNo, sportingComplexContact, *teamID*)
OR
TEAM (teamID, teamName, , contactPerson, contactPhoneNo, *suburbID*) *preferable

Step 4:
COACH (coachID, coachName, coachAddress, coachPhoneNo, accreditationLevel, *teamID*)
PLAYER (playerID, playerName, playerAddress, playerPhoneNo, *teamID*)

Step 5:
PLAYS_AGAINST (*homeTeamID*, *awayTeamID*, date, startTime, score)

Step 6:
COACHING_QUALIFICATIONS (*coachID*, qualification)

Step 7: No n-ary relationships

FINAL TABLES

TEAM (teamID, teamName, contactPerson, contactPhoneNo, *suburbID*) *Assuming second option at Step 3
SUBURB (suburbID, suburbName, sportingComplexName, sportingComplexAddress, sportingComplexPhoneNo, sportingComplexContact)

COACH (coachID, coachName, coachAddress, coachPhoneNo, *teamID*)

PLAYER (playerID, playerName, playerAddress, playerPhoneNo, *teamID*)

PLAYS_AGAINST (*homeTeamID*, *awayTeamID*, date, startTime, score)

COACHING_QUALIFICATIONS (*coachID*, qualification)

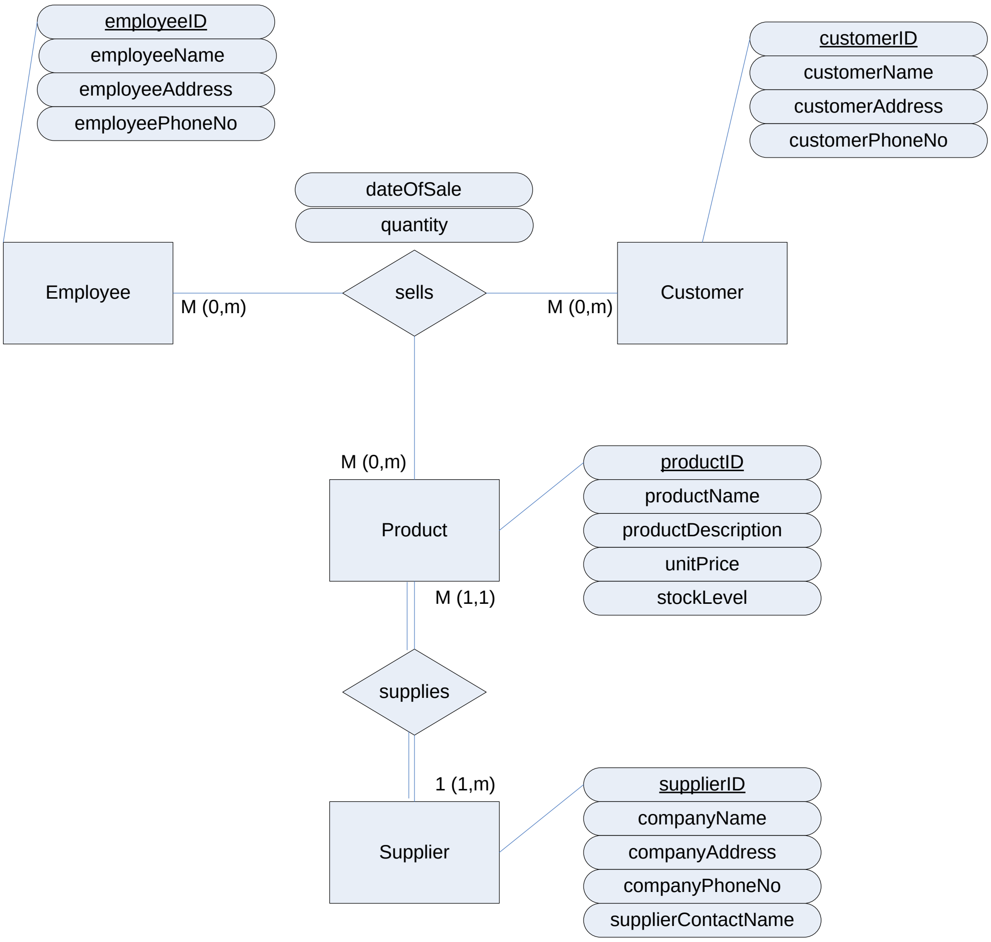
Exercise 3 – *n*-ary Relationships

a) Construct an ER-Model for the following problem description. State any assumptions that you make.

An electrical appliance company needs to keep track of the sales made by its employees. In particular it wants to know which products each employee has sold and who the employee has sold the products to. The company keeps the following personal information about its employees: employee name, address, and contact phone number. For warranty purposes, it also keeps the following information about its customers: customer name, address, contact phone number, products bought and the date on which they were bought.

The company manager requires the new system to also keep track of product stock levels so she/he knows when to contact the company's suppliers to order more stock. Product information includes: a unique product id, product name, product description, unit price, current stock level and the name of the company who supplies the product. In order to easily contact the suppliers the system should also store their relevant contact details.

Answer:



b) Transform your ER-model into relational tables. Make sure you follow the 7-steps and list your final tables at the end.

Answer:

Step 1:
EMPLOYEE (employeeID, employeeName, employeeAddress, employeePhoneNo)
CUSTOMER (customerID, customerName, customerAddress, customerPhoneNo)
PRODUCT (productID, productName, productDescription, unitPrice, stockLevel)
SUPPLIER (supplierID, companyName, companyAddress, companyPhoneNo, supplierContactName)

Step 2: - no weak entities

Step 3: - no 1:1 relationships

Step 4:
PRODUCT (productID, productName, productDescription, unitPrice, stockLevel, *supplierID*)

Step 5: - no binary M:M relationships

Step 6: - no multi-valued attributes

Step 7:
SELLS (*employeeID*, *customerID*, *productID*, quantity, dateOfSale)

FINAL TABLES
EMPLOYEE (employeeID, employeeName, employeeAddress, employeePhoneNo)
CUSTOMER (customerID, customerName, customerAddress, customerPhoneNo)
SUPPLIER (supplierID, companyName, companyAddress, companyPhoneNo, supplierContactName)
PRODUCT (productID, productName, productDescription, unitPrice, stockLevel, *supplierID*)
SELLS (*employeeID*, *customerID*, *productID*, quantity, dateOfSale)

Exercise 4

a) Construct an ER-Model for the following problem description. State any assumptions that you make.

The Melbourne Employment Corporation (MEC) places temporary workers in companies during peak periods. MEC's manager gives you the following description of the business:

MEC has a file of candidates who are willing to work. They record the candidate's number, name, contact address, contact phone number and maximum hours the candidate is available per week. Each candidate may have several qualifications. MEC uses special codes to record the candidate's qualification for a position. As well as recording the code, MEC also records the experience of the candidate in each qualification area, expressed as the number of months of experience. The list of qualification codes includes:

Code	**Description**
SW-60	Secretarial Work, 60 words per minute
CLWRK	Clerical Work
PRGC++	Programmer: C++
PRGJAV	Programmer: JAVA
……	……

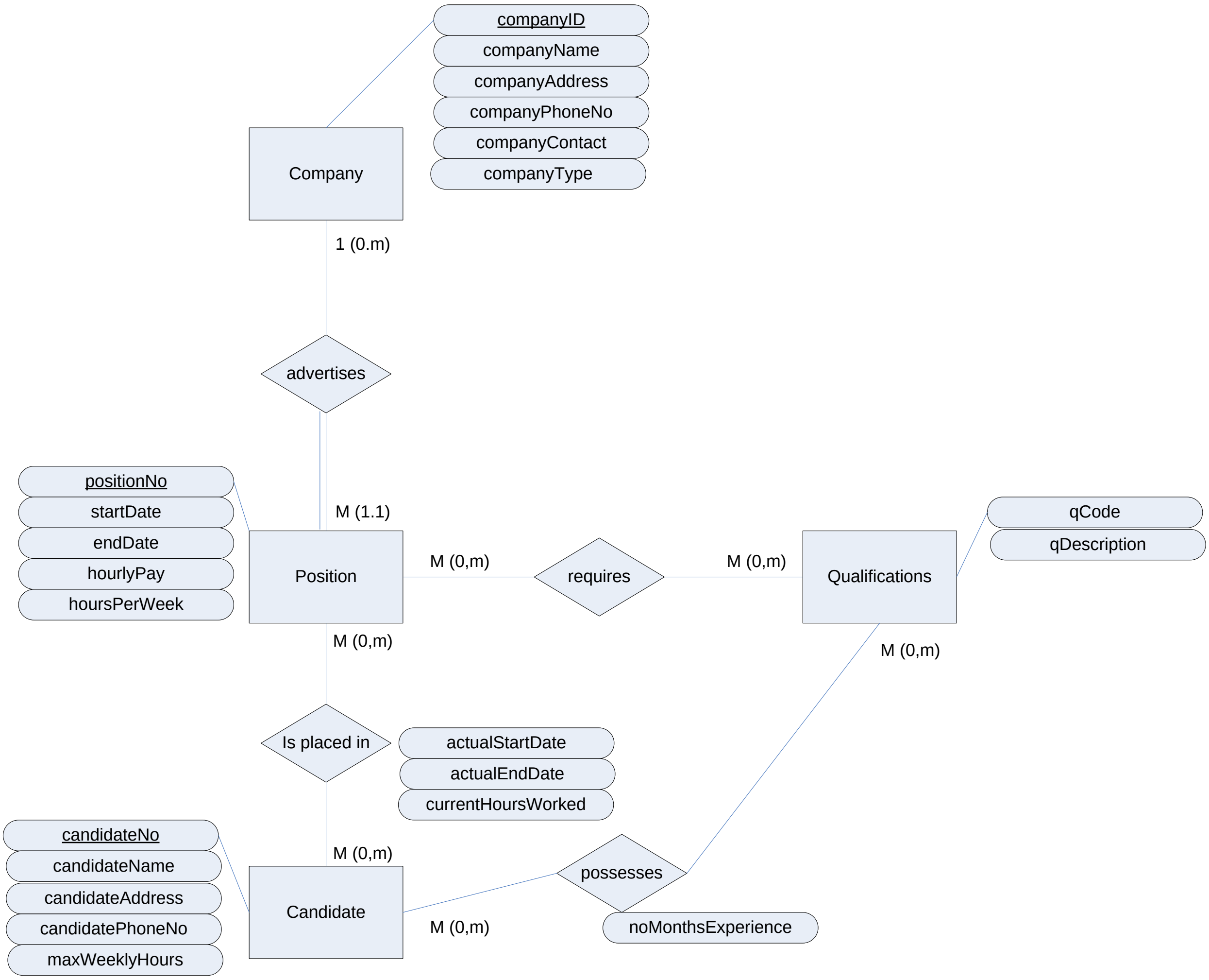
MEC also has a list of companies that request temporaries. Each company is assigned a company number as an identifier. The company name, address, phone number, type of business and principal contact for employment placements is also recorded.

Each time a company requests a temporary employee MEC makes an entry in the open positions file. This file contains a position number, the company requesting an employee, the required qualifications, starting date, anticipated ending date, the number of hours per week required and hourly pay. MEC may be able to fill the opening from the staff on its books; however, in some circumstances it cannot fill the request.

When a candidate has the relevant qualifications, and is available, he/she is given the job, and an entry is made in the placement record folder. This folder contains the position number, candidate number, actual start date, total hours worked to date, and end date if the placement is completed. The placement record folder is used by MEC as a source of placement histories for its various temporaries.[1]

[1] Adapted from Rob P. and Coronel C. *Database Systems – Design, Implementation, & Management*, Thomson Learning, 2000.

Answer:



NOTE:
Some students might want to have a PLACEMENT entity between POSITION and CANDIDATE

b) Transform your ER-model into relational tables. Make sure you follow the 7-steps and list your final tables at the end.

Answer:

Step 1:
COMPANY (companyID, companyName, companyAddress, companyPhone, companyContact, companyType)
POSITION (positionNo, startDate, endDate, hourlyPay, hoursPerWeek)
QUALIFICATION (qCode, qDescription)
CANDIDATE (candidateNo, candidateName, candidateAddress, candidatePhoneNo, maxWeeklyHours)

Step 2: No weak entities

Step 3: No 1:1 relationships

Step 4:
POSITION (positionNo, startDate, endDate, hourlyPay, hoursPerWeek, *companyID*)

Step 5:
HAS_QUALIFICATION (*candidateNo*, *qCode*, noOfMonths)
REQUIRES (*positionNo*, *qCode*)
PLACEMENT (*positionNo*, *candidateNo*, actualStartDate, actualEndDate, currentHoursWorked)

Step 6: No multi-valued attributes

Step 7: No n-ary relationships

FINAL TABLES
COMPANY (companyID, companyName, companyAddress, companyPhone, companyContact, companyType)
CANDIDATE (candidateNo, candidateName, candidateAddress, candidatePhoneNo, maxWeeklyHours)
QUALIFICATION (qCode, qDescription)
POSITION (positionNo, startDate, endDate, hourlyPay, hoursPerWeek, *companyID*)
HAS_QUALIFICATION (*candidateNo*, *qCode*, noOfMonths)
REQUIRES (*positionNo*, *qCode*)
PLACEMENT (*positionNo*, *candidateNo*, actualStartDate, actualEndDate, currentHoursWorked)