Final Solution to Assignment 5

PERSON (PersonID, lastNM, firstNM, homePhone, workPhone, email, streetNumber, streetNM, City, State, Zip)

EMPLOYEE (PersonID, birthDT, startDT, endDT)
FK: EMPLOYEE.personID PERSON.personID

CUSTOMER (PersonID)
FK: CUSTOMER.personID PERSON.personID

CONTACT (PersonID, supplierID)
FK: CONTACT.personID PERSON.personID
FK: CONTACT.supplierID SUPPLIER.supplierID

TRADESPERSON (PersonID)
FK: TRADESPERSON.personIDEMPLOYEE.personID

PRIMARYCONTACT (employeeID, customerID, assignDT)
FK: PRIMARYCONTACT.employeeID EMPLOYEE.personID
FK: PRIMARYCONTACT.customerID CUSTOMER.personID

CORPORATE (PersonID, discountRate)
FK: CORPORATE.personIDCUSTOMER.personID

RESIDENTIAL (PersonID, birthDT, marketSector)
FK: RESIDENTIAL.personIDCUSTOMER.personID

RESOURCE (resourceID, description)

SERVICE (serviceID, serviceShortNM)
FK: SERVICE.serviceIDRESOURCE.resourceID

MATERIAL (materialID, minUnitStock)
FK: MATERIAL.materialIDRESOURCE.resourceID

JOB (employeeID, projectID, serviceID, jobID, startDT, endDT, estimatedCost, actualCost)
FK: JOB.employeeID TRADESPERSON.personID
FK: JOB.projectID PROJECT.projectID
FK: JOB.serviceID SERVICE.serviceID

PROJECT (projectID, createDT, description, isInside, parentProjectID, customerID)
FK: PROJECT.parentProjectID PROJECT.projectID
FK: PROJECT.customerIDCUSTOMER.personID

SKILLEDIN (personID, serviceID, certNM, certDT)
FK: SKILLEDIN.personID EMPLOYEE.personID FK:
SKILLEDIN.serviceID SERVICE.serviceID

PROJECTLOCATION (ProjectID, Location)
FK: PROJECTLOCATION.projectIDPROJECT.projectID

REQUIRES (resourceID, projectID)
FK: REQUIRES.projectIDPROJECT.projectID
FK: REQUIRES.resourceIDRESOURCE.resourceID

SUPPLIER (supplierID, supplierNM, isPreferred)

SOURCEDFROM (materialID, supplierID, restockDT, cost)
FK: SOURCEDFROM.supplierIDSUPPLIER.supplierID FK:
SOURCEDFROM.materialIDMATERIAL.materialID

RETURNS (personID, projectID, resourceID, returnDT, reason)
FK: RETURNS.personID PERSON.personID
FK: RETURNS.projectID PROJECT.projectID
FK: RETURNS.resourceID MATERIAL.materialID

Step-by-Step Solution to Assignment 5

Key: FK Foreign key constraint
* This table is revised in subsequent steps
JOB Indicates an entity, relationship or attribute depicted on the EER diagram

Step 1 – Map out the strong entities.
Leave the super/subclass entities until step 8
SUPPLIER (supplierID, supplierNM, isPreferred)
PROJECT (projectID, createDT, description, isInside) *

Step 2 – Map out weak entities.
JOB is a weak entity, but we can't create the table yet because the *TRADESPERSON* and *SERVICE* subclasses are not yet mapped.

Step 3 – 1 to 1 relationships. There aren't any.

Step 4 – 1 to N relationships.
The *designs* and *worksFor* relationships are 1:N, but one of the entities in each of these relationships is a subclass that we haven't yet mapped. The only 1:N relationship that we can map at this point is the *partOf* recursive relationship.
PROJECT (projectID, createDT, description, isInside, parentProjectID)*
FK: PROJECT.parentProjectID PROJECT.projectID

Step 5 – M:N relationships.
The relationships *skilledIn, primaryContact, requires,* and *returns* are N:M, but one of the entities in each of these is a subclass that we haven't yet mapped.

Step 6 – Multi-valued attributes.

PROJECTLOCATION (ProjectID, Location)
FK: PROJECTLOCATION.projectIDPROJECT.projectID

Step 7 – N-Ary relationships. Leave *performs* until we have mapped out the super/subclass

Step 8 – Superclass/Subclass Entities.

Superclass/Subclasses	Choices Available	Choice Made and why
PERSON / EMPLOYEE, CUSTOMER, CONTACT	A, D	A because the *designs* and *worksFor* relationships apply to only one of the subclasses. The relationship *primary contact* is much easier to implement if using A rather than D.
EMPLOYEE / TRADESPERSON	A, C, D	A because only a trades-person can perform a service for the project.
CUSTOMER/CORPORATE, RESIDENTIAL	A, B, C, D	Only option A and B enable us to make market sector required. Option B would make it difficult to ensure that either type of customer can design a project, thus option A is best.
RESOURCE/SERVICE, MATERIAL	A, C, D	A because the *performs*, *returns* and *sourcedFrom* relationships apply to only one subclass.

The super/subclass mappings are:

PERSON (PersonID, lastNM, firstNM, homePhone, workPhone, email, streetNumber, streetNM, City, State, Zip)

EMPLOYEE (PersonID, birthDT, startDT, endDT)
FK: EMPLOYEE.personIDPERSON.personID

CUSTOMER (PersonID)
FK: CUSTOMER.personIDPERSON.personID

CONTACT (PersonID)
FK: CONTACT.personIDPERSON.personID

TRADESPERSON (PersonID)
FK: TRADESPERSON.personIDEMPLOYEE.personID

CORPORATE (PersonID, discountRate)
FK: CORPORATE.personIDCUSTOMER.personID

RESIDENTIAL (PersonID, birthDT, marketSector)
FK: RESIDENTIAL.personIDCUSTOMER.personID

RESOURCE (resourceID, description)

SERVICE (serviceID, serviceShortNM)
FK: SERVICE.serviceIDRESOURCE.resourceID

MATERIAL (materialID, minUnitStock)
FK: MATERIAL.materialIDRESOURCE.resourceID

With the super-class entities mapped out we can now step through each of the steps to make sure that we have all the relationships mapped

Step 1 –Strong entities - Done

Step 2 –Weak entities
JOB (employeeID, projectID, serviceID, jobID, startDT, endDT, estimatedCost, actualCost) FK: JOB.employeeID TRADESPERSON.personID
FK: JOB.projectID PROJECT.projectID
FK: JOB.serviceID SERVICE.serviceID

Step 3 –1:1 relationships - Done

Step 4 – 1 to N relationships
To map the *designs* relationship we add a foreign key to the *PROJECT* table
PROJECT (projectID, createDT, description, isInside, parentProjectID, customerID)
FK: PROJECT.parentProjectID PROJECT.projectID FK:
PROJECT.customerIDCUSTOMER.personID

To map the *worksFor* relationship we add a foreign key to the *CONTACT* table
CONTACT (PersonID, supplierID)
FK: CONTACT.supplierID SUPPLIER.supplierID

Step 5 – M:N relationships
SKILLEDIN (personID, serviceID, certNM, certDT)
Note that we need a compound primary key so that the same tradesperson can get multiple certifications for the same service.
FK: SKILLEDIN.personID EMPLOYEE.personID FK:
SKILLEDIN.serviceID SERVICE.serviceID

REQUIRES (resourceID, projectID)
FK: REQUIRES.projectIDPROJECT.projectID

FK: REQUIRES.resourceIDRESOURCE.resourceID

PRIMARYCONTACT (employeeID, customerID, assignDT)
FK: PRIMARYCONTACT.employeeID EMPLOYEE.personID
FK: PRIMARYCONTACT.customerID CUSTOMER.personID

SOURCEDFROM (materialID, supplierID)
FK: SOURCEDFROM.supplierIDSUPPLIER.supplierID FK:
SOURCEDFROM.materialIDMATERIAL.materialID

RETURNS (personID, projectID, resourceID, returnDT, reason)
FK: RETURNS.personID PERSON.personID
FK: RETURNS.projectID PROJECT.projectID
FK: RETURNS.resourceID MATERIAL.materialID