ISTE-230 Introduction to Database & Data Modeling

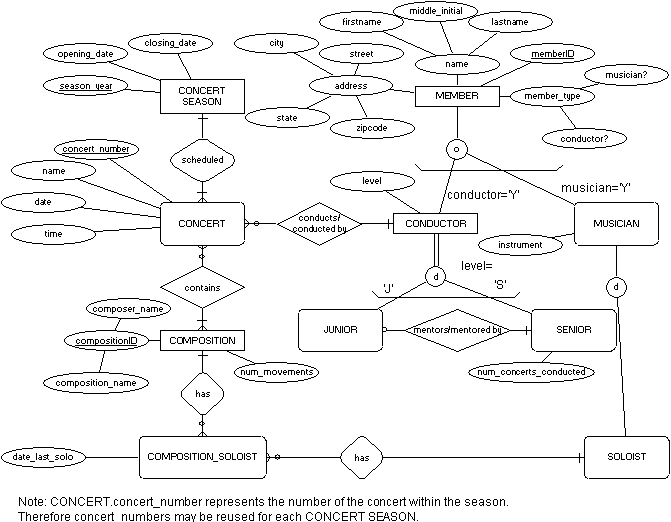
## Practice Exercise # 10 – Transposing II

**Name:** Brian Zarzuela

(It may be helpful to right-click on the  icon and select Hide Spelling Errors and Hide Grammatical Errors.)

**Problem #1**

Using the RITPO diagram that appears below, answer the following questions. **Please account for the note that is at the bottom of the diagram.**



Explain why SOLOIST is weak and what the specific term for that type of entity is.

**REASON:** SOLOIST is weak because an instance can’t exist without an instance of MUSICIAN

**TERM:** subtype entity

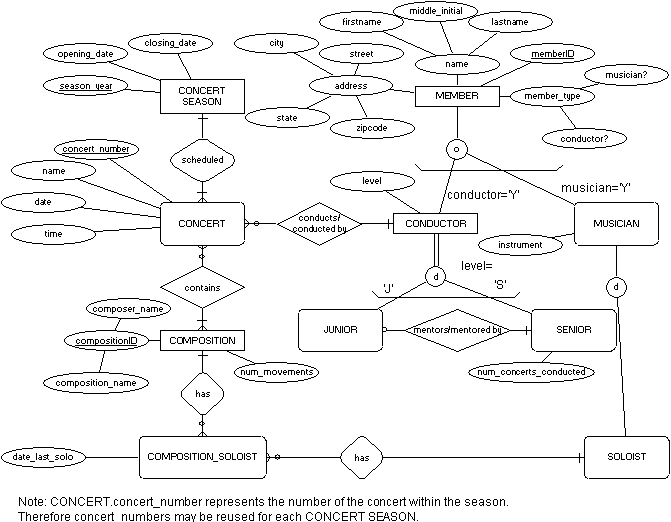
Explain why CONCERT is weak and what the specific term for that type of entity is.

**REASON:** can’t exist on it’s own, must participate in the relationships with CONCERT\_SEASON, CONDUCTOR, and COMPOSITION

**TERM:** ID-dependant weak entity

**Problem #2**

Transpose the E-R diagram below into relations, implementing all relationships. Denote primary keys and foreign keys appropriately. Use proper relation notation. Make sure your relation names are in all caps. **Please account for the note that is at the bottom of the diagram. Rounded relationship diamonds denote identifying relationships.**



**YOUR TRANSPOSED RELATIONS:**

MEMBER(memberID, firstname, middle\_inital, lastname, city, street, state, zipcode, conductor?, musician?)

CONDUCTOR(*memberID*, level)

CONDUCTOR(memberID) mei MEMBER(memberID)

JUNIOR(*memberID*, *mentor\_memberID*)

JUNIOR(memberID) mei CONDUCTOR(memberID)

JUNIOR(mentor\_memberID) mei SENIOR(memberID)

SENIOR(*memberID*, num\_concerts\_conducted)

SENIOR(memberID) mei CONDUCTOR(memberID)

MUSICIAN(*memberID*, instrument)

MUSICIAN(memberID) mei MEMBER(memberID)

SOLOIST(*memberID*)

SOLOIST(memberID) mei MUSICIAN(memberID)

CONCERT\_SEASON(season\_year, operating\_date, closing\_date)

CONCERT(concert\_number, *season\_year*, name, date, time, *memberID*)

CONCERT(season\_year) mei CONCERT\_SEASON(season\_year)

CONCERT(memberID) mei CONDUCTOR(memberID)

CONCERT\_COMPOSITION(*concert\_number, season\_year, composition\_name, composer\_name*)

CONCERT\_COMPOSITION(concert\_number, season\_year) mei CONCERT(concert\_number, season\_year)

CONCERT\_COMPOSITION(composition\_name, composer\_name) mei COMPOSITION(composition\_name, composer\_name)

COMPOSITION(composition\_name, composer\_name, num\_movements)

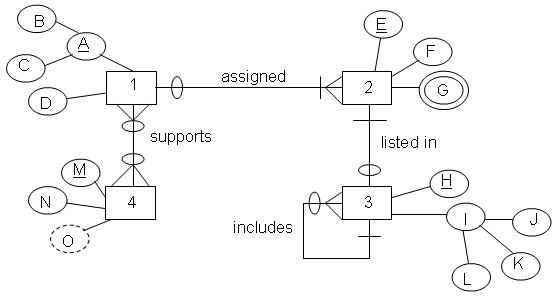
COMPOSITION\_SOLOIST(*composition\_name, composer\_name, memberID*, date\_last\_solo)

COMPOSITION\_SOLOIST(composition\_name, composer\_name) mei COMPOSITION(composition\_name, composer\_name)

COMPOSITION\_SOLOIST(memberID) mei SOLOIST(memberID)

**Problem #3**

Given the E-R diagram, transposed relations (from PE9 – does not distinguish between identifying and non-identifying relationships) and the functional dependencies shown below, normalize all relations, and any resulting relations through BCNF, using proper relational notation. Note: The lines in the E-R diagram do not distinguish identifying and non-identifying relationships.



**Functional Dependencies:**

B, C -> D

C->D

D->C

E->F, (B, C)

H-> J, K, L, E, Hb

K-> L

M->N

**The Transposed Relations:**

1(B, C, D)

2(E, F, G, *B, C*)

2(B, C) mei 1(B, C)

3(H, J, K, L, *E, Hb*)

3(E) mei 2(E)

3(Hb) mei 3(H)

4(M, N)

1\_4(*B, C, M*)

1\_4(B, C) mei 1(B, C)

1\_4(M) mei 4(M)

**YOUR FINAL SET OF NORMALIZED RELATIONS:**

1(B, *C*)

1(C) mei 5(C)

2(E, F, *B, C*)

2(B, C) mei 1(B, C)

3(H, J, K, *E, Hb*)

3(E) mei 2(E)

3(Hb) mei 3(H)

3(K) mei 7(K)

4(M, N)

1\_4(*B, C, M*)

1\_4(B, C) mei 1(B, C)

1\_4(M) mei 4(M)

5(C, D)

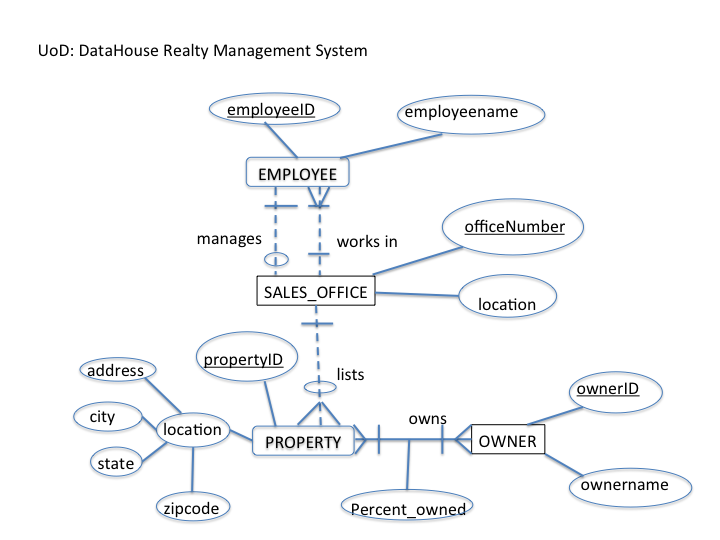
6(*E, G*)

6(E) mei 2(E)

7(K, L)

**Problem #4**

Transpose the E-R diagram below into relations, implementing all relationships. Denote primary keys and foreign keys appropriately. Use proper relation notation. Make sure your relation names are in all caps.



**YOUR TRANSPOSED RELATIONS:**

EMPLOYEE(employeeID, employeename *officeNumber*)

EMPLOYEE(officeNumber) mei SALES\_OFFICE(officeNumber)

SALES\_OFFICE(officeNumber, location, *manager\_employeeID*)

SALES\_OFFICE(manager\_employeeID) mei EMPLOYEE(employeeID)

PROPERTY(propertyID, address, city, state, zipcode, *officeNumber*)

PROPERTY(officeNumber) mei SALES\_OFFICE(officeNumber)

OWNER(ownerID, ownername)

PROPERTY\_OWNER(*propertyID, ownerID*, percent\_owned)

PROPERTY\_OWNER(propertyID) mei PROPERTY(propertyID)

PROPERTY\_OWNER(ownerID) mei OWNER(ownerID)

**Problem #5**

Using the DataHouse Realty diagram from Problem #4, please explain what relationship makes PROPERTY weak and what about that relationship causes PROPERTY to be weak.

**YOUR REASON:**

PROPERTY is a weak entity because the lists relationship requires total participation from PROPERTY