ANSWERS TO REVIEW QUESTIONS 7

1. What is normalization?

Normalization is the process for assigning attributes to entities. Properly executed, the normalization process eliminates uncontrolled data redundancies, thus eliminating the data anomalies and the data integrity problems that are produced by such redundancies.
Normalization does not eliminate data redundancy; instead, it produces the carefully *controlled* redundancy that lets us properly link database tables.

2. When is a table in 1NF?

A table is in 1NF when all the key attributes are defined (no repeating groups in the table) and when all remaining attributes are dependent on the primary key. However, a table in 1NF still may contain partial dependencies, i.e., dependencies based on only part of the primary key.

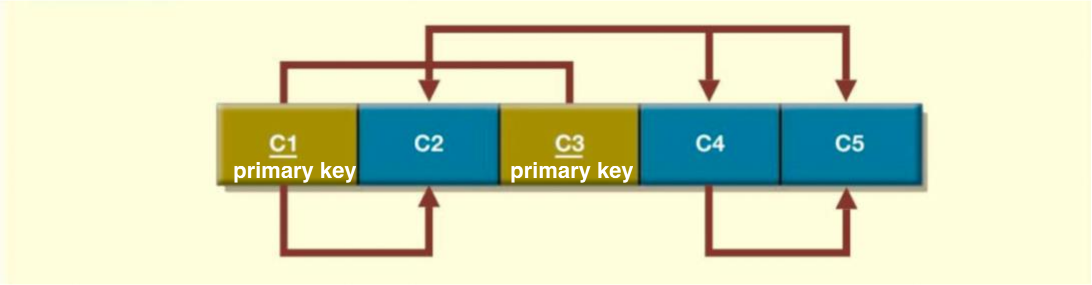
3. When is a table in 2NF?

A table is in 2NF when it is in 1NF and it includes no partial dependencies. However, a table in 2NF may still have transitive dependencies, i.e., dependencies based on attributes that are not part of the primary key.

4. When is a table in 3NF?

A table is in 3NF when it is in 2NF and it contains no transitive dependencies.

5. Given the dependency diagram shown in figure below, answer questions 5a through 5c:



a. Identify and discuss each of the indicated dependencies.

C1 → C2 represents a *partial dependency*, because C2 depends only on C1, rather than on the entire primary key composed of C1 and C3.

C4 → C5 represents a *transitive dependency*, because C5 depends on an attribute (C4) that is not part of a primary key.

C1, C3 → C2, C4, C5 represents a functional dependency, because C2, C4, and C5 depend on the primary key composed of C1 and C3.

b. Create a database whose tables are at least in 2NF, showing the dependency diagrams for each table.

Table 1

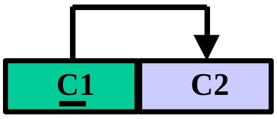


Table 1
Primary key: C1
Foreign key: None
Normal form: 3NF

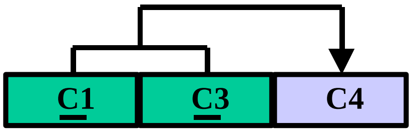


Table 2
Primary key: C1 + C3
Foreign key: C1 (to Table 1)
C4 (to Table 3)
Normal form: 3NF

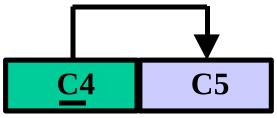


Table 3
Primary key: C4
Foreign key: None
Normal form: 3NF

c. Create a database whose tables are at least in 3NF, showing the dependency diagrams for each table.

6. What is a partial dependency? With what normal form is it associated?

A partial dependency exists when an attribute is dependent on only a portion of the primary key. This type of dependency is associated with 1NF.

7. What three data anomalies are likely to be the result of data redundancy? How can such anomalies be eliminated?

The most common anomalies considered when data redundancy exists are: update anomalies, addition anomalies, and deletion anomalies. All these can easily be avoided through data normalization. Data redundancy produces data integrity problems, caused by the fact that data entry failed to conform to the rule that all copies of redundant data must be identical.

8. Define and discuss the concept of transitive dependency. You may use the following table structure as an example.

INVOICE (INV_NUM, INV_DATE, INV_AMOUNT, CUS_NUM, CUS_ADDRESS, CUS_PHONE)

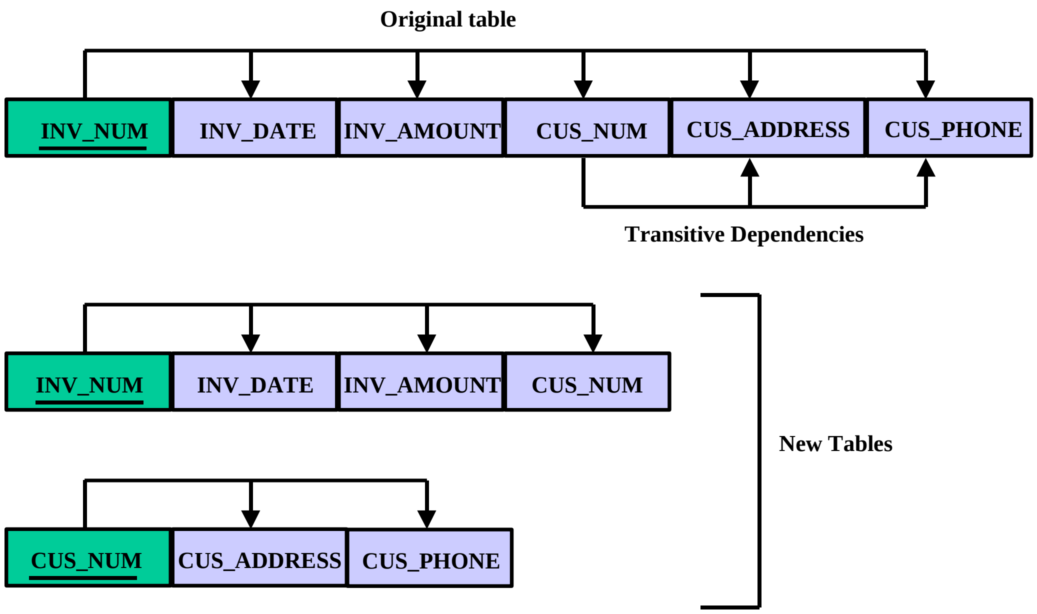
Transitive dependency is a condition in which an attribute is dependent on another attribute that is not part of the primary key. This kind of dependency usually requires the decomposition of the table containing the transitive dependency.

To remove a transitive dependency, the designer must perform the following actions:

- Place the attributes that create the transitive dependency in a separate table.

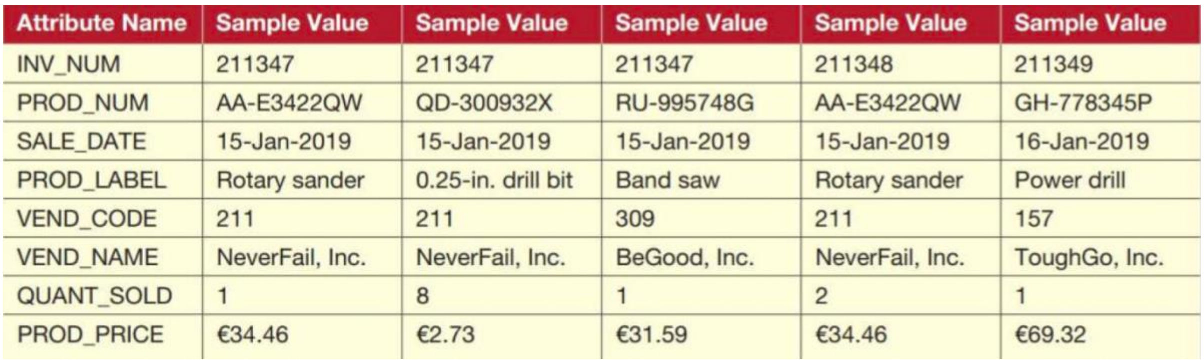
- Make sure that the new table's primary key attribute is the foreign key in the original table.

Figure below shows an example of a transitive dependency removal.



ANSWERS TO PROBLEMS 7

1. **Using the following INVOICE table structure, draw its dependency diagram and identify all dependencies (including all partial and transitive dependencies). You can assume that the table does not contain repeating groups and that any invoice number may reference more than one product, and each product is purchased from a single vendor. (*Hint:* This table uses a composite primary key.)**



Attribute Name	Sample Value	Sample Value	Sample Value	Sample Value	Sample Value
INV_NUM	211347	211347	211347	211348	211349
PROD_NUM	AA-E3422QW	QD-300932X	RU-995748G	AA-E3422QW	GH-778345P
SALE_DATE	15-Jan-2019	15-Jan-2019	15-Jan-2019	15-Jan-2019	16-Jan-2019
PROD_LABEL	Rotary sander	0.25-in. drill bit	Band saw	Rotary sander	Power drill
VEND_CODE	211	211	309	211	157
VEND_NAME	NeverFail, Inc.	NeverFail, Inc.	BeGood, Inc.	NeverFail, Inc.	ToughGo, Inc.
QUANT_SOLD	1	8	1	2	1
PROD_PRICE	€34.46	€2.73	€31.59	€34.46	€69.32

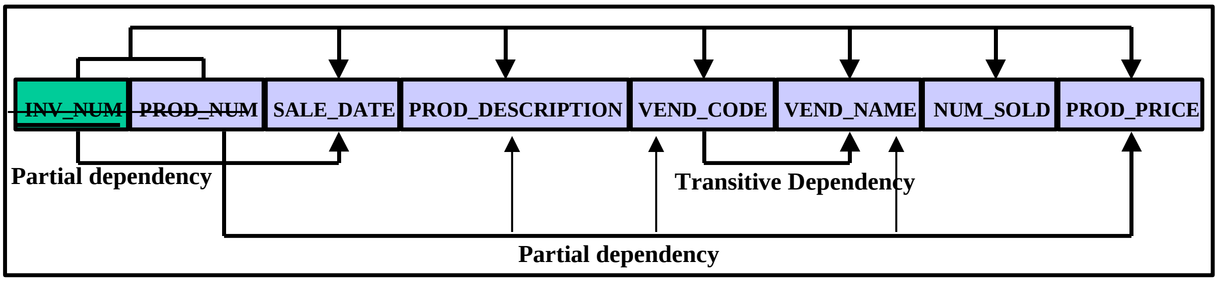
The solution to this problem is combined with the solution to Problem 2 to let you trace the decomposition process in a single slide.

2. **Using the initial dependency diagram drawn in problem 1, remove all partial dependencies, draw the new dependency diagrams, and identify the normal forms for each table structure you created.**

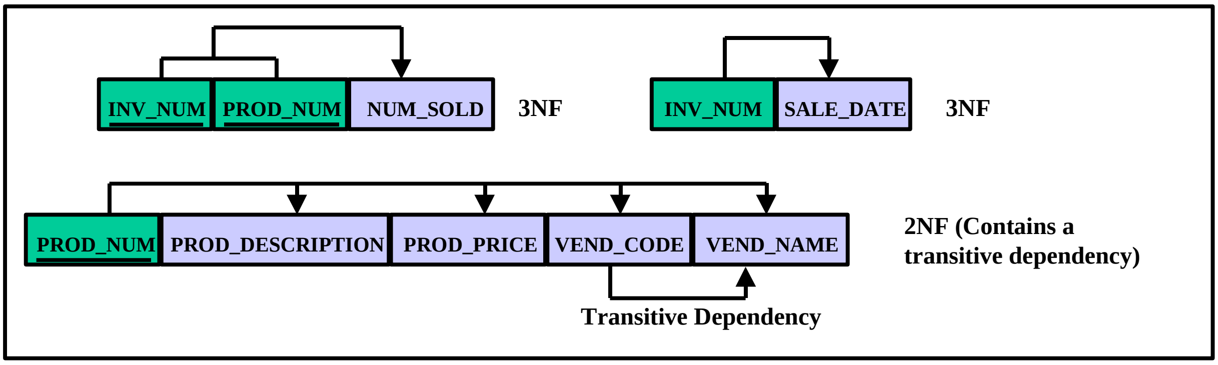
Note: You can assume that any given product is supplied by a single vendor, but a vendor can supply many products. Therefore, it is proper to conclude that the following dependency exists: PROD_NUM → PROD_DESCRIPTION, PROD_PRICE, VEND_CODE, VEND_NAME (*Hint:* Your actions should produce three new dependency diagrams.)

The solution to both problem 1 and 2 is shown in figure below.

Problem 1 Solution



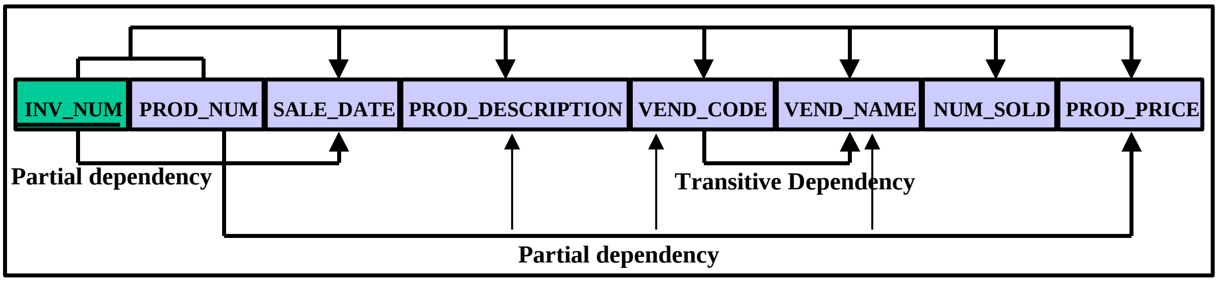
Problem 2 Solution



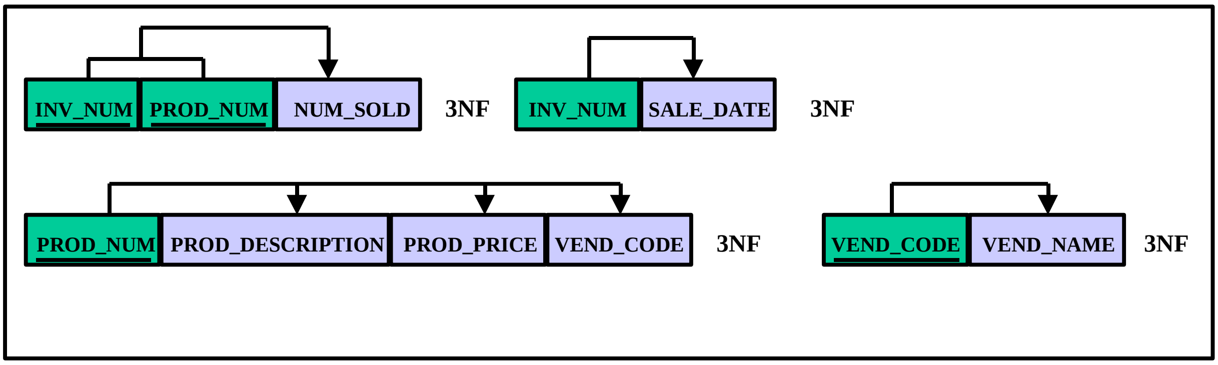
3. Using the table structures you have created in problem 2, remove all transitive dependencies, draw the new dependency diagrams, and identify the normal forms for each table structure you created.

To illustrate the effect of Problem 3's complete decomposition, we have shown Problem 1's dependency diagram again in figure below.

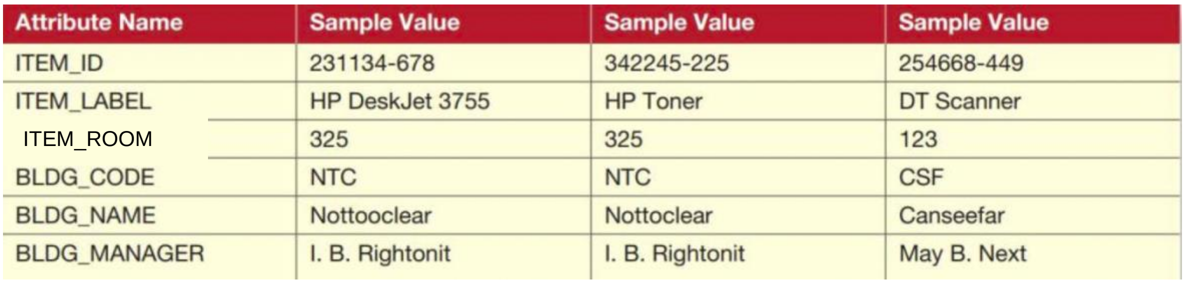
Problem 1 Solution



Problem 3 Solution



4. **To keep track of office furniture, computers, printers, and so on, the FOUNDIT company uses the following table structure:**



Attribute Name	Sample Value	Sample Value	Sample Value
ITEM_ID	231134-678	342245-225	254668-449
ITEM_LABEL	HP DeskJet 3755	HP Toner	DT Scanner
ITEM_ROOM	325	325	123
BLDG_CODE	NTC	NTC	CSF
BLDG_NAME	Nottooclear	Nottoclear	Canseefar
BLDG_MANAGER	I. B. Rightonit	I. B. Rightonit	May B. Next

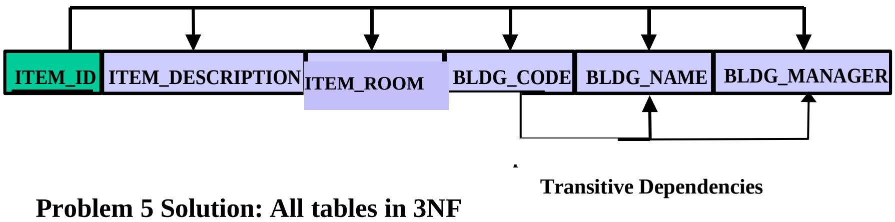
Given this information, draw the dependency diagram. Make sure you label the transitive and/or partial dependencies.

The answer to this problem is shown in combination with Problem 5's answer to show the decomposition process.

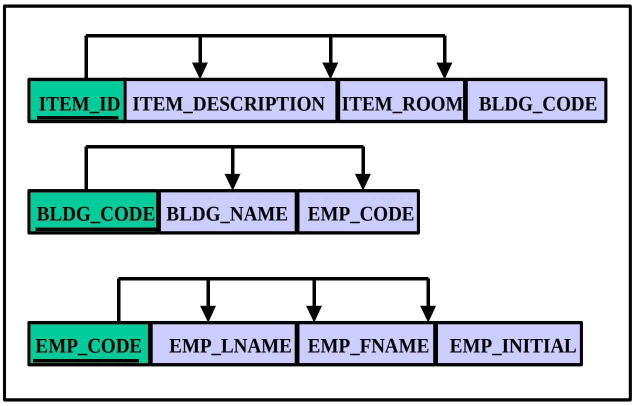
5. **Starting with the dependency diagram drawn for problem 1, create a set of dependency diagrams that meet 3NF requirements. Rename attributes to meet the naming conventions and create new entities and attributes as necessary.**

Note that the dependency diagrams in figure below reflect the notion that each building is managed by one employee.

Problem 4 Solution



Problem 5 Solution: All tables in 3NF

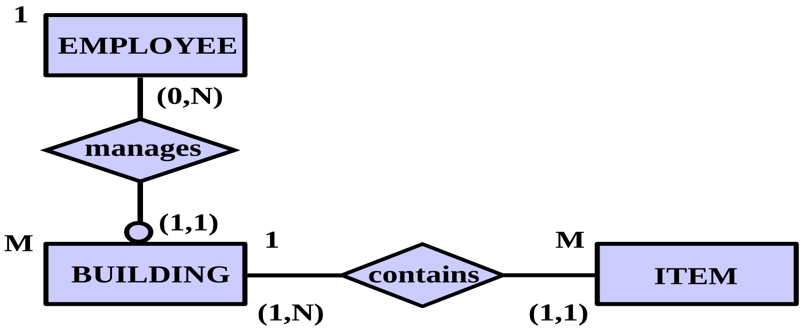


6. **Using the results of problem 5, draw the E-R diagram.**

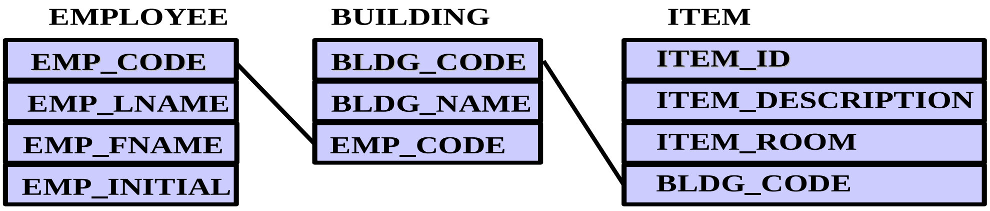
Use figure below to show that, in this case, the E-R diagram reflects the business rule that one employee can manage many (or at least more than one) building. Because all employees are not *required* to manage buildings, BUILDING is optional to EMPLOYEE in the *manages* relationship. Once again, the nature of this relationship is not and cannot be reflected in the dependency diagram.

Note: We also assume here that each item has a unique item code and that, therefore, an item can be located in only one place at a time.

Problem 6 Solution - ERD



Relational Schema (Do not forget to add 1 and ∞ symbols to the appropriate side of the relationship

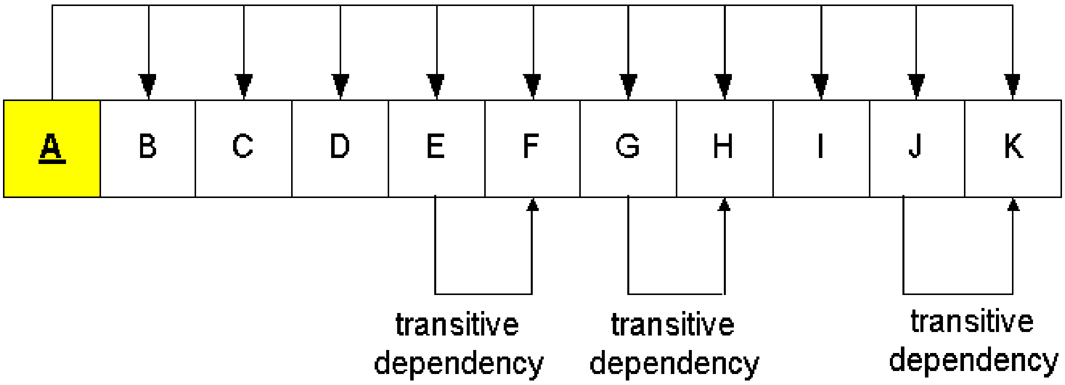

EMPLOYEE
EMP_CODE
EMP_LNAME
EMP_FNAME
EMP_INITIAL

BUILDING
BLDG_CODE
BLDG_NAME
EMP_CODE

ITEM
ITEM_ID
ITEM_DESCRIPTION
ITEM_ROOM
BLDG_CODE

7. **Given the following five sample records in the CHARTER table, draw the dependency diagram for this table structure. Make sure you label all dependencies. (The record contents are written vertically to save space. The EMP_NUM entry refers to the pilot who flew the charter trip). To save space for the dependency diagram, code the attributes A, B, C, D,.... For example, CHAR_TRIP will be coded A, CHAR_DATE will be coded B, and so on.**

	Record 1	Record 2	Record 3	Record 4	Record 5
CHAR_TRIP	10232	10233	10234	10235	10236
CHAR_DATE	27-Jan-2002	27-Jan-2002	28-Jan-2002	28-Jan-2002	28-Jan-2002
CHAR_DESTINATION	STL	ATL	ATL	GNV	MEM
CHAR_MILES	580	470	510	1024	280
CUST_NUMBER	784	784	546	567	546
CUST_LNAME	Brown	Brown	Alero	Green	Alero
EMP_NUM	32	32	18	41	18
EMP_LNAME	Yantil	Yantil	Smith	Chen	Smith
AC_NUM	2098W	2098W	6711Y	6711Y	3124R
MOD_CODE	PA31-350	PA31-350	C-90	C-90	C-421
MOD_CHG_MILE	$2.12	$2.12	$3.85	$3.85	$2.37



8. **Decompose the dependency diagram in problem 4 to create table structures that are all in 3NF. Make sure you label all dependencies.**

dependency dependency dependency

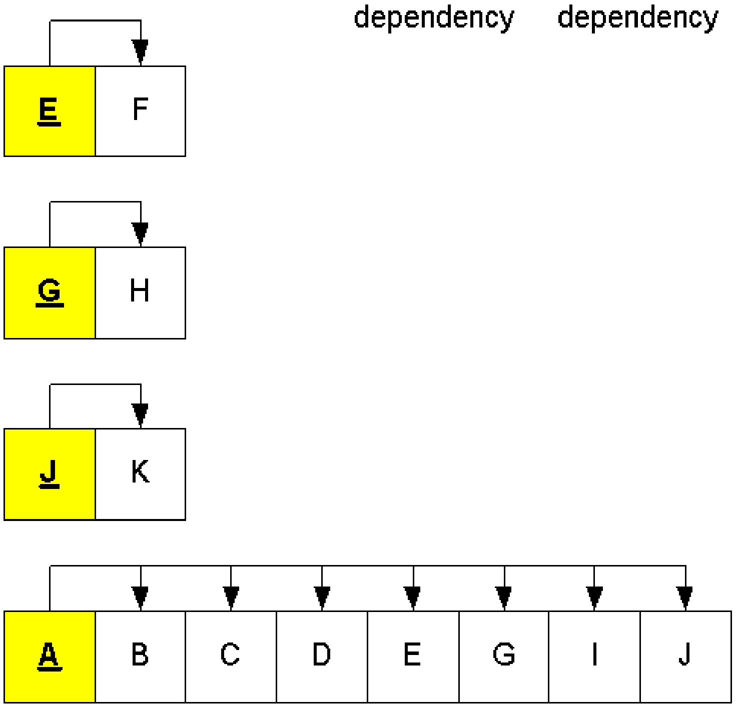


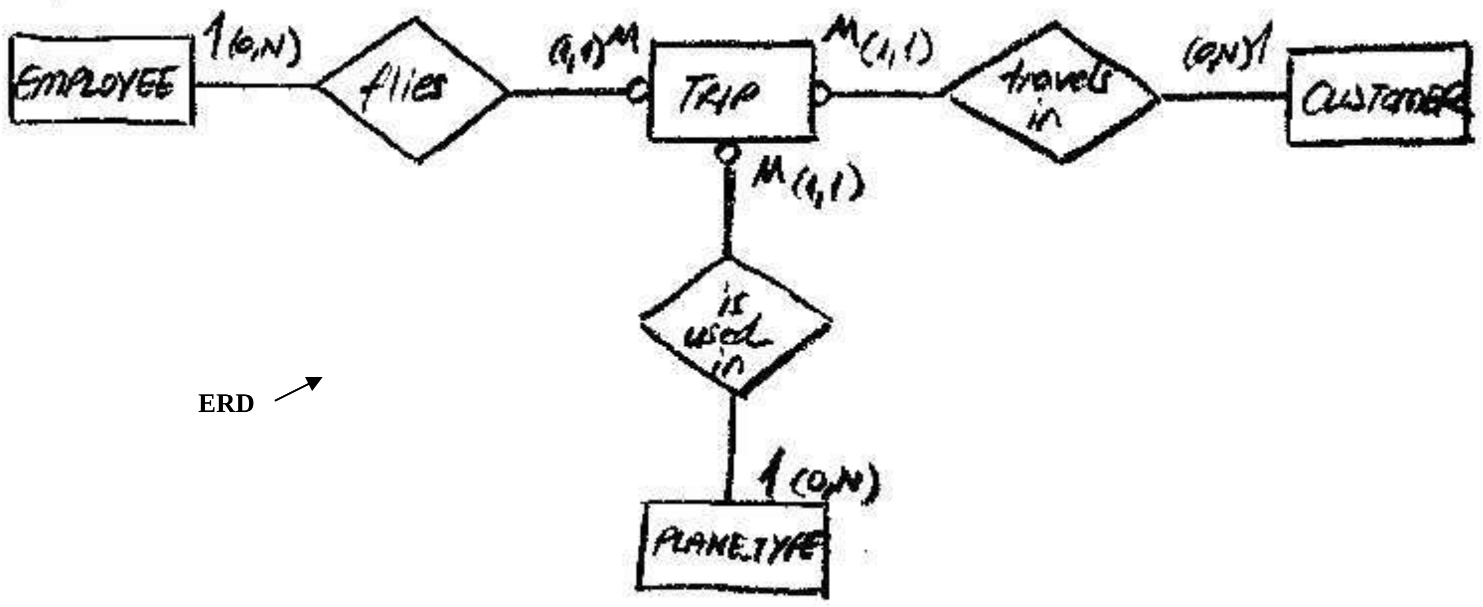
Table name: Customers

Table name: Employees

Table name: PlaneTypes

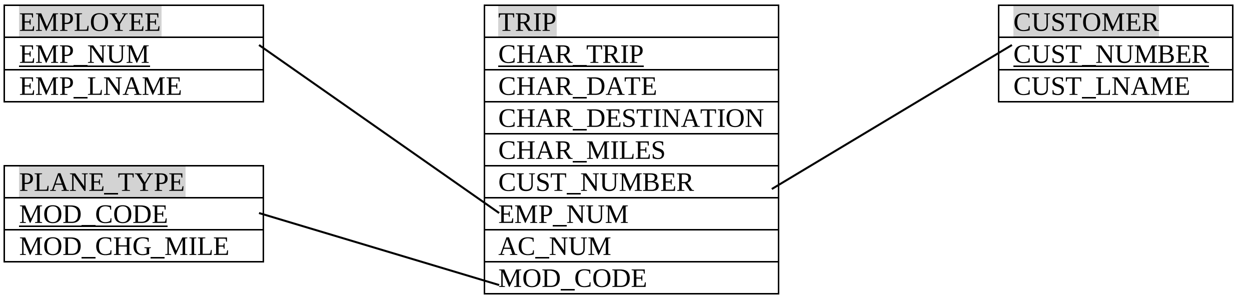
Table name: Trips

9. **Draw the ERD to reflect the properly decomposed dependency diagrams you created in problem 8. Make sure that the ERD yields a database that can all the data shown in problem 7. Show all entities, relationships, connectivities, optionalities, and cardinalities.**



ERD →

Relational Schema (Do not forget to add 1 and ∞ symbols to the appropriate side of the relationship



EMPLOYEE
EMP_NUM
EMP_LNAME

PLANE_TYPE
MOD_CODE
MOD_CHG_MILE

TRIP
CHAR_TRIP
CHAR_DATE
CHAR_DESTINATION
CHAR_MILES
CUST_NUMBER
EMP_NUM
AC_NUM
MOD_CODE

CUSTOMER
CUST_NUMBER
CUST_LNAME