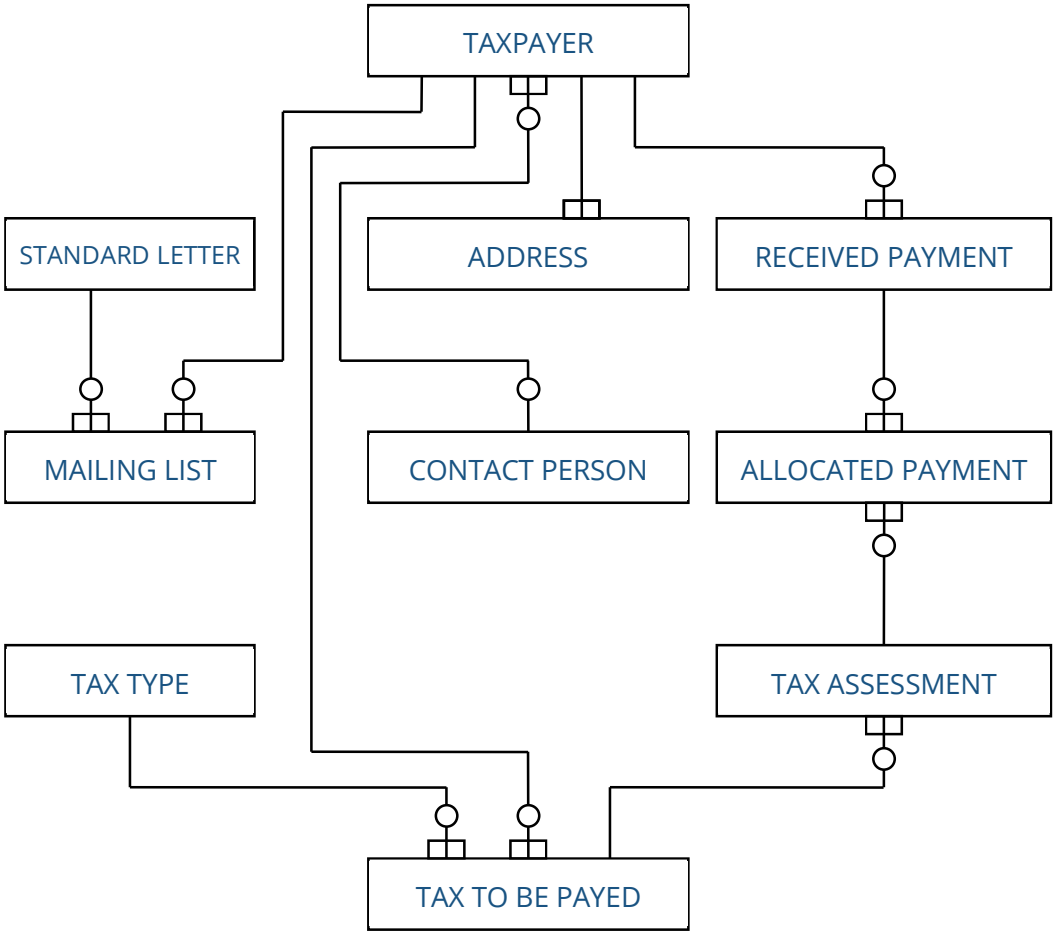
9 COUNTING LOGICAL FILES (DATA FUNCTIONS)

Problem description

Below a part of a normalized data model is illustrated.



This data model was made on the basis of the following user specifications.

All of the entity types named are maintained by the application.

The entity type *Taxpayer* contains the taxpayer identification number, the name, the date of birth, and some personal information about a taxpayer.

A taxpayer can have several addresses. For example, in addition to a home address (the minimum that must be present), an invoice address and/or a post office box number may also be identified.

The entity type *Standard Letter* consists of a unique letter number and of a fixed text belonging to the letter.

The entity type *Mailing List* contains only reference keys and indicates which letter is sent to which taxpayer.

The entity type *Tax Type* contains the different kinds of taxes that can be charged. The composition of *Tax Type* is as follows: code, description, and tax amount per month. (In this particular case, the tax pertains to fixed assessments that are the same for every taxpayer.)

The entity type *Tax To Be Paid* records which taxes must be paid by which taxpayer. In addition to reference keys, it also contains the date on which the tax obligation becomes effective and the date on which this obligation ends (expiration date). (The latter is usually not known when the obligation becomes effective, but is recorded later.)

The entity type *Tax Assessment* contains an amount, the final payment date, and the applicable tax period. An assessment always covers a fixed period: a year, half-year, quarter, or month.

The entity type *Received Payment* contains the amount received, the date on which the payment is received, and the amount that has still not been allocated to a *Tax Assessment*.

The entity type *Allocated Payment* contains reference keys to *Tax Assessment* and *Received Payment*, but also contains the part of the *Received Payment* that has been allocated for payment of the linked *Tax Assessment*.

The entity type *Contact Person* contains the names and some supplementary data about the employees of the tax department who can act as a contact person for a taxpayer. A particular contact person is assigned only when a taxpayer asks for advice. From that moment on, the taxpayer is always spoken to by the same person.

In principle, a taxpayer is entered into the system only when he is required to pay one or more kinds of tax. The taxpayer can be deleted as soon as he is no longer registered for a *Tax Type* (i.e., all the expiration dates in the linked entities of *Tax To Be Paid* have elapsed or, in other words, the taxpayer is no longer obliged to pay the tax) and no *Received Payments* are linked to the taxpayer anymore. When deleting the *Taxpayer* the linked occurrences in *Mailing List* will be automatically deleted. *Taxes To Be Paid* will also be deleted automatically when deleting the taxpayer, provided that no *Tax Assessments* are linked to it still.

A *Tax Assessment* is archived via a batch function one year after it has been paid in full. The archive file created contains the taxpayer identification number, the type of tax involved, the period the tax covers, the amount of the tax, the date on which the assessment was sent, and the date on which the assessment was paid in full. When the data is recorded in the archive, the *Tax Assessment* is deleted immediately together with the *Allocated Payments* linked to it.

A *Received Payment* can be deleted only if the full amount has been allocated and *Allocated Payments* are no longer linked to it.

Finally, a *Tax Type* may be deleted only if it does not have any *Tax To Be Paid* still linked to it.

How many logical files are present in this normalized data model? Are there any historical files?

Discussion

To analyze this data model, you should assume the denormalization rules given in section 4.21. The first question that must then be posed is whether any FPA tables are present. The description of the entity types shows that only the entity type Standard Letter meets the criteria for an FPA table. The only entity type whose status is ambiguous and can be discussed in this regard is *Tax Type* because it contains an amount, in addition to a code and a description. This means that it contains dissimilar kinds of data; i.e., it is not just meant for the translation of the code.

In keeping with the denormalization rules the next question that should be asked is "which entity types contain only key data"? These entity types do not count according to the guideline; however referral attributes are being counted as data element type in both logical data files associated with the present key-key entity.

At first glance, in this data model, it appears that it is only the *Mailing List* entity type. However, the taxpayer's key should then be counted on the *Standard Letter* and thus the entity type *Standard Letter* (which was referred to as the FPA table) would lose the character of the FPA table. In fact, *Mailing List* is no longer a key-key entity but the result of normalizing a recurring attribute in *Taxpayer*. Therefore, the conclusion is that *Mailing List* is not covered by the concept of "key-key entity" and still "simply" is counted.

The *Allocated Payment* entity type contains, in addition to the referring keys, the amount paid to a *Tax Assessment* and does not meet the requirements at this point. The entity type *Tax To Be Paid* also contains more data than just key data.

The other nine entity types must be examined as to how many internal logical files they represent. This is done on the basis of cardinality, optionality, and entity independence. Each pair of entity types linked via a relationship is looked at to see whether they should be included in one logical file.

The relationship between *Taxpayer* and *Contact Person* is bilaterally optional. Within the context of the guidelines, then, they are independent logical files. Additionally, *Contact Person* does not have any relationships with other entity types and is therefore one internal logical file with one record type.

The relationship between *Taxpayer* and *Address* is a bilateral-mandatory 1:N relationship. In keeping with the denormalization rules, these two entity types belong to the same internal logical file. In order to determine whether any other entity types should be included in this internal logical file, the remaining relationships of the entity type *Taxpayer* must be investigated.

The relationship between the entity type *Taxpayer* and *Mailing List* is a 1: (N) relationship. The problem description shows that the occurrences of *Mailing List* associated with a *Taxpayer* to be removed are automatically deleted. *Mailing List* is thus dependent and is therefore included in the same internal logical file.

The relationship between the entity type *Taxpayer* and *Received Payment* is a 1:(N) relationship in which *Taxpayer* may not be deleted as long as a *Received Payment* is still linked to it. This means that *Received Payment* is entity independent in relation to *Taxpayer* and does not belong to the same internal logical file as *Taxpayer* and *Address*.

The next relationship of *Taxpayer* that must be examined is its 1:(N) relationship to *Tax To Be Paid*. Here when a *Taxpayer* is deleted, the entities *Tax To Be Paid* that are linked are deleted automatically. Consequently, *Tax To Be Paid* is entity dependent on *Taxpayer* and, therefore, belongs to the same internal logical file as *Taxpayer* and *Address*. Whether any more entity types should be included in this internal logical file now also depends on the relationships of *Tax To Be Paid*.

The relationship between *Tax To Be Paid* and *Tax Assessment* is a 1:(N) relationship. The problem description above shows that an entity *Tax To Be Paid* may be deleted only if no *Tax Assessment* entities are linked to it anymore. Therefore, *Tax Assessment* has an autonomous meaning to this application and should consequently be considered entity independent in relation to *Tax To Be Paid*.

The relationship between *Tax Type* and *Tax To Be Paid* is also a 1:(N) relationship. A *Tax Type* may be deleted only if it does not have any *Tax To Be Paid* entities attached to it. *Tax To Be Paid* is therefore entity independent from *Tax Type*.

Now that all the relationships of *Taxpayer*, *Address*, and *Tax To Be Paid* have been analyzed, we can conclude that *Taxpayer*, *Address*, and *Tax To Be Paid*, together, make up one internal logical file with three record types.

As we have seen, *Received Payment* is entity independent with regard to *Taxpayer*. In order to determine whether this entity type is an internal logical file in and of itself, we must investigate its 1:(N) relationship with *Allocated Payment*. The problem description above shows that a *Received Payment* can be deleted only if there are no *Allocated Payments* attached to it anymore. This means that *Allocated Payment* is entity independent with regard to *Received Payment*. *Received Payment* is therefore an internal logical file with one record type.

Earlier we indicated that *Tax Assessment* is entity independent with regard to *Tax To Be Paid*. Additionally, *Tax Assessment* still has a 1:(N) relationship with *Allocated Payment*. According to the problem description above, any *Allocated Payments* linked to a *Tax Assessment* are deleted automatically when the *Tax Assessment* is archived and deleted. This means that an *Allocated Payment* is entity dependent on *Tax Assessment*. *Tax Assessment* and *Allocated Payment* together, therefore, make up one internal logical file with two record types.

As indicated above, *Tax To Be Paid* is entity independent in relation to *Tax Type*. Additionally, *Tax Type* does not have any relationships with other entity types and is not an FPA table. It is therefore an independent internal logical file with one record type.

The problem description above shows that a file with historical data does exist. This file is not included as an entity type in the data model. It is, however, required by the user. The composition of this file is different than the composition of the other internal logical files, so that a separate internal logical file with one record type must be counted for it.

Solution

Count internal logical files as indicated below.

Entity types:	Count as:	Number of record types:
Taxpayer + Address + Tax To Be Paid + Mailing List	1 ILF	4
Tax Type	1 ILF	1
Received Payment	1 ILF	1
Tax Assessment + Allocated Payment	1 ILF	2
Contact person	1 ILF	1
Standard letter	Count as part of the FPA tables ILF	1
Historical Tax Assessment	1 ILF	1

References to the standard

4.20, 4.21, 5.2.a, 5.2.b, 5.2.i and 5.2.k.