

DATABASE DESIGN BY EXAMPLE: AN AQUARIUM EXPORTS DATABASE

INTRODUCTION

This article examines how to derive a conceptual or logical model from a sample export form, and how to normalise the model to reduce the risk of inconsistencies and anomalies. This methodology, described in depth in Oppel (2004) and Powell (2005), can be applied for any similar example to model a relational database.

As an example, let's say that several companies in your country are exporting marine products, in particular live reef fish for the aquarium trade. You, as a fisheries manager, decide to monitor that fishery through de-

Franck Magron
Reef Fisheries Information
Manager, SPC PROCFish/C
Project (FranckM@spc.int)

tailed export permits delivered by fisheries department to the exporting companies for each shipment. You want to produce various statistics by commodity, period, country of export, and local company and decide to create a database to store and query the data.

This typical scenario follows the steps explained in the FAO book "Guidelines for routine collection of capture fishery data"

(Anon 1999). The scenario begins with a policy decision (e.g. managing the aquarium fishery) that requires indicators. A data collection strategy is then defined and is followed by designing a data management plan and database. The final stages are planning and implementing the data collection scheme.

DISSECTING A SAMPLE EXPORT PERMIT

To design a database, it is often useful to obtain not only an empty form but to also gather some typical examples of special cases. For the latter, a decision will have to be made: either increase the complexity of the model to cope with these special cases, or transform them so that they fit the standard model, possibly losing information in the process. For this exercise we will assume that the sample form presented in Figure 1 is sufficient to characterise an export permit.

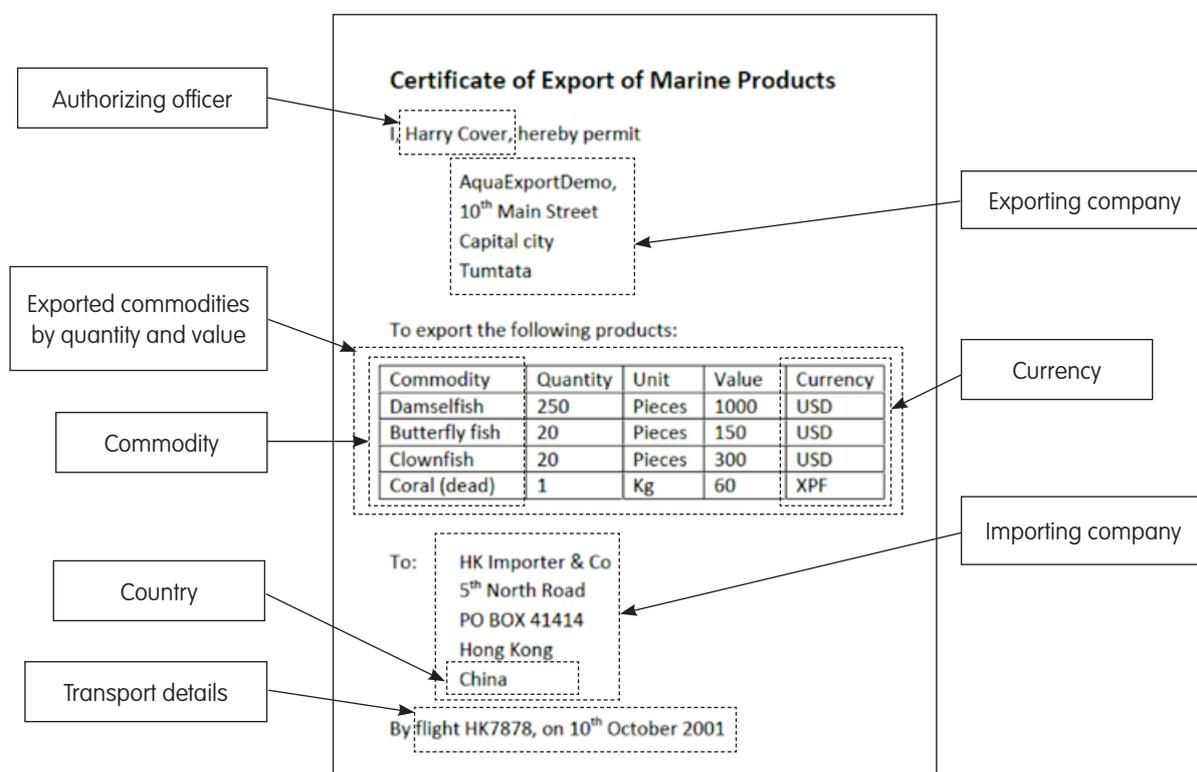


Figure 1. Sample export permit.

The form can be divided into coherent blocks of data: name of the exporting company, name of the importing company (or consignee), general information about the export permit (authorising officer, shipping method, etc.), and finally, details about the commodities being exported.

Additionally, we can identify information that is likely to be common to different export permits, such as the authorising officer, the commodity type and the currency. These pieces of information can also be linked to other data, such as scientific names for the commodities, currency exchange rates, or authentication for authorising officers.

Once this analysis has been conducted we can start the logical design of the database.

DATABASE DESIGN

On an Excel spreadsheet, data are presented as a single flat table (Table 1) and part of the information is duplicated across rows. This redundancy of data makes the data prone to inconsistencies because any update or creation of a new attribute must be applied to all copies. A flat structure is, therefore, not recommended for data storage.

In a relational database however, data are spread into several tables and the normalisation process ensures that data are not redundant (first normal form¹) and that attributes of each table depend on the primary key and not on other attributes (second and third normal form).

From the sample form, we can derive tables for export permits, export details and companies. Reference tables are also created for countries, commodities, processing levels, currencies and fisheries officers, allowing additional attributes (country code, comments, etc.) to be attached to each reference type (Fig. 2).

Because a company can have both importing and exporting activities, a single table is sufficient and is linked twice to the export permits table: once for the exporting company and once for the importing country. Boolean attributes `COM_Company_Exporting` and `COM_Company_Importing` are used to indicate the activities the company is involved in.

In the `FIS_Export_Details` table, the quantity from the sample form has been split into two attributes depending on the volume or unit (pieces or kg), thus reducing the risk of mixing the two in subsequent queries.

In Figure 2, the symbols (key and infinity sign) at the ends of the table links indicate a “one to many” relationship: an export detail row corresponds to one and only one commodity but the same commodity can be referred to by many export detail rows.

The coherence of the normalised database is maintained through not null, primary key and referential constraints.

A “not null” constraint on an attribute makes it mandatory. In Figure 3, the date of export can’t be null, therefore it is mandatory, whereas the airway bill is optional.

A primary key must be not null and unique. No two export permits will share the same identifier (`FIS_Export_Permit_ID`). The primary key is indicated in SQL Server Management Studio by a gold key icon on the left of the column name. The constraint “PK” is shown in parenthesis.

Finally, referential constraints enforce coherence between tables and ensure, for example, that it is not possible to delete a commodity referenced by `COM_Export_Details` and that when a permit is deleted, all corresponding export details are

deleted also (cascade deletion). In Figure 3, exporting company ID, consignee company ID and user ID are all references to other tables’ primary key and are, therefore, called foreign keys (FK). They are materialised by a silver key icon.

CONCLUSION

The normalisation process (coupled with integrity constraints) considerably reduces the risk of redundancies and inconsistencies in the database, whether the user interacts directly with the database or through data entry forms. The conceptual design is independent of the choice of the database management system such as MS Access, SQL Server or MySQL and they all provide a way of implementing integrity constraints.

This export database is currently implemented by SPC as part of a larger coastal fisheries database that will be made available to fisheries officers in the region.

BIBLIOGRAPHY

- Anon. 1999. Guidelines for the routine collection of capture fishery data. FAO Fisheries Technical Paper No. 382. 113p. Available at: <http://www.fao.org/docrep/003/x2465e/x2465e00.htm>
- Oppel A. 2004. Databases demystified: A self-teaching guide. McGraw-Hill/Osborne Media. 360 p.
- Powell G. 2005. Beginning database design. Wrox. 496 p.



¹ A full chapter is devoted to normalisation and formal definitions of normal forms in “Databases demystified” (Oppel 2004).

Table 1. Tabular data.

| Permit N° | Officer | Export company | Commodity | Qty | Unit | Value | Currency | Consignee | Flight |
|-----------|-------------|----------------|-------------|-----|------|-------|----------|----------------|--------|
| 2001/25 | Harry Cover | AquaExp | Damselfish | 250 | Pcs | 1000 | USD | HK Import | HK7878 |
| 2001/25 | Harry Cover | AquaExp | Clownfish | 20 | Pcs | 300 | USD | HK Import | HK7878 |
| 2001/25 | Harry Cover | AquaExp | Coral(dead) | 1 | Kg | 6000 | XPF | HK Import | HK7878 |
| 2001/26 | Harry Cover | AquaExp | Damselfish | 200 | Pcs | 800 | USD | The Big Import | TW1124 |
| 2001/27 | Paul Smith | FreshExp | Coral | 4 | Kg | 200 | EUR | Europa Import | GR4451 |

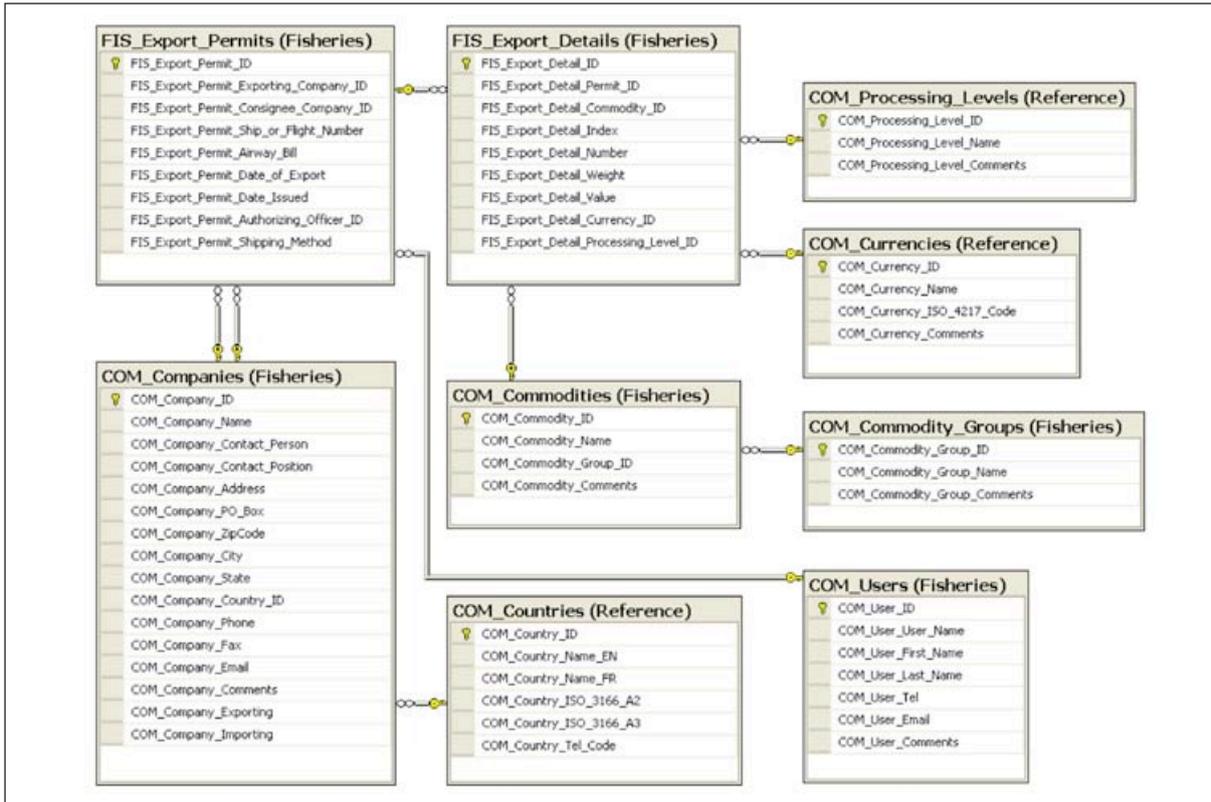


Figure 2. Database schematic.

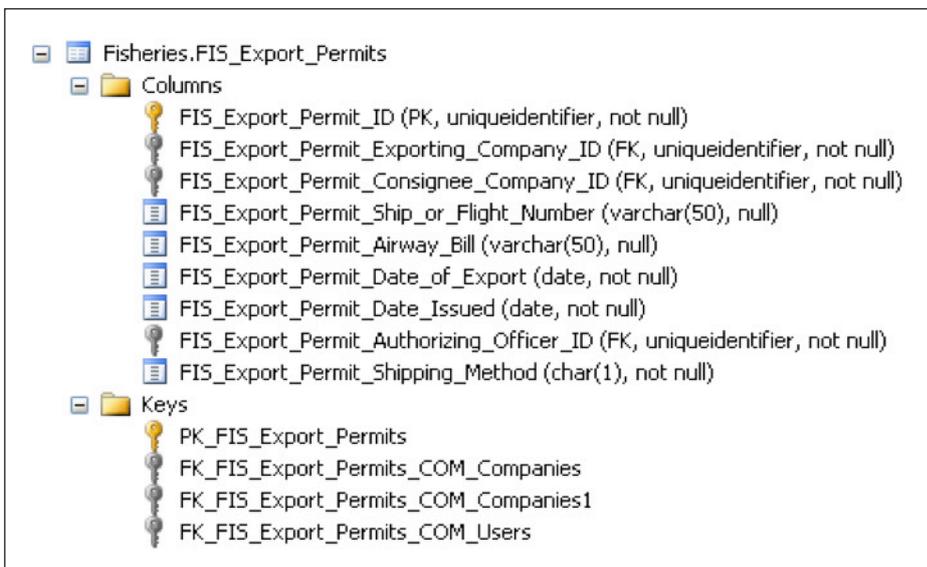


Figure 3. Columns and keys of FIS_Export_Permits table.