STEEL PRODUCTION DATA WAREHOUSE REENGINEERING

François Fouss, Martí Ibarz, Manuel Kolp, Alain Pirotte
IAG –School of Management, ISYS –Information Systems Research Unit, University of Louvain, 1 Place des Doyens, Belgium
Email: {fouss, kolp, pirotte}@isys.ul.ac.be

Abstract: This paper is the second part of the object-oriented study of the Carsid cokerie. The aim of this project was to redesign the data models from the Marchienne plant of CARSID and to integrate all of them in one single data model, while improving its design and showing the possibilities offered by OLAP technologies in the bettering of the data exploitation carried out in this plant.

1 INTRODUCTION

From the middle of the 90s, a great majority of businesses, from smaller to bigger companies, have applied some kind of information technology to their activities. This has caused that a great quantity of businesses have now some sort of system where data are stored. There are a wide range of cases where those systems have been designed to work with technologies which are nowadays obsolete (dBase IV or other sorts of prerelational storing systems). In other cases, the staff who developed the systems was not made up of qualified personnel. However, the data storing systems so developed have been working in those enterprises and have gathered a huge quantity of crucial data. Moreover, these databases have been developed to home data from an information system designed with the subsequent capital investment, something which we cannot afford disregarding.

Nowadays, when the information applications running in these companies must evolve to comply with the needs of a changing socioeconomic and technological context, a new data model must be designed from the existing structures, a new model which takes profit from the valuable data already gathered. Sometimes this is not an optimal solution, but it is the solution taken because the company cannot devote the capital needed for working out a new design. Some other times, a solution based on the existing one *is* the best solution, since the necessary changes can be perfectly integrated in the currently operative structure. Anyway, database reengineering would be the technique used in both cases, a methodology which is becoming stronger.

As a consequence of this phenomenon, a great majority of businesses has been gathering data for 5 to 10 years now, and these data could be highly useful if submitted to a suitable treatment. This leads us to the question whether the information the business has is being optimally analysed and used. Whereas traditional technologies can abstract a great deal of information, with new technologies such as data warehouse data can be treated multidimensionally so that correlations may be established. This multidimensionality presents information in such a way that new relationships between data may come to the light thanks to new visualizing possibilities and not obviously apprehensible conclusions may be reached. Data mining is another technique which can be used in pattern searching and data classification. This technology simplifies processes and minimizes the costs of submitting data to statistical analyses. Thus obtained patterns and classifications will let us draw purchaser's profiles, relate the source of any raw material to the product's final quality, or establish many more relationships which could prove critical for any business. Therefore, these technologies will allow for a more exhaustive and relevant data analysis.

The technical evolution stated above has made us notice the interest lying in doing a study in the application of these technologies to a real case in order to see the way in which a business can be helped to grow evolving its data models and improving the exploitation of the information gathered by its information system. A good way to understand and study these technologies would be working with them in a real case. The business under study should have previously developed some kind of data storing system and have a substantial volume of data to work with. Obviously, it would

also be very important that this company be willing to work along the stated lines.

CARSID (Carolo-Sidérurgie) is a joint venture that has recently arisen from the concentration movement. It was created by the combination of the Duferco and Usinor Belgium S.A companies (capital share 60/40). Duferco is an Italian enterprise that is very active in Belgium (La Louvière and Clabecq) and Usinor is a part of the worldwide leading steel giant, Arcelor, in partnership with Luxembourg's Arbed and the Spanish company Acelaria. As stated above, Duferco has a majority holding in CARSID.

UCL has been working with CARSID for quite a long time now and from this consultancy relationship an MA project was conducted where requirement abstraction and UML modelization were applied to that same plant. As a result of this work, it became evident the necessity of redesigning the databases currently operating in the plant and developing a data warehouse which allowed for a suitable exploitation of the data available. These necessities gave rise to the present study which involves different phases of the tasks being done in the plant, namely material transportation, both input and output, and production process. Our aim is to study and improve the databases which worked with these processes.

This project wants to modify the existing databases with new data models. In the process of doing so, we must take into account any applications and interfaces which are currently feeding those databases. Therefore, we must find a solution which takes advantage from them in order to reduce the cost that an optimal development of the requirement abstraction, posed by the previous project, would imply. In our project, we will also study and present the possibilities that arise with the application of data warehouse and data mining technologies and will also develop some examples of OLAP (On-Line Analytical Processing) cubes so that the company may see the possibilities offered by this technology bearing in mind a prospective future implementation.

In our database reengineering we have made use of CASE tools. Namely, we have used Rational Rose, a case tool which has made it easier for us to carry out the adequate processes of engineering and reverse engineering. The database management system (DBMS) has been Microsoft SQL Server 7 and 2000, and the data warehouse has been developed using Microsoft Analysis Services 2000.

The rest of the paper is organized as follows. Section 2 describes the coking process used by Carsid. Section 3 explains the process of database reengineering. Section 4 describes the data warehousing and the data mining techniques. The section 3 and 4 are illustrated with the Carsid example. Finally, section 5 summarizes the contributions and points to further work.

2. DESCRIPTION OF THE COKING PROCESS USED BY CARSID

The job of preparing coke from a suitable coal involves a long, hard process in which the coal is submitted to a series of successive transformations (Fig. 1). The aim is to produce high quality coke while treating all the by-products generated during the different stages of the process. The coking plant is divided into three sections.

Preparation section. The admittance and primary treatment of the coal to be used to make the coke is the beginning of the chain in the coking plant.

Until the 80s the coal came from Belgian mines. However, the closure of these mines drove Cockerill to import the coal from other sources. It is shipped from the United States, Poland, South Africa and Australia to the ports of Antwerp and Rotterdam. The properties of each type of coal differ according to where it comes from. Three means of transport are used to transfer the coal from the ports to the plants: by road, by barges and by train, this last method being the most commonly used at the CARSID coking plant.

The coals that arrive are stored either in a pit or in a bunker that serves as a security stock to provide protection against the difficulties that may arise from problems in supplies or deliveries. Use of this stock guarantees production for a maximum of about 15 days. Prolonged storage could in fact have a detrimental effect on the quality of the coal used, which would inevitably also have repercussions on the quality of the coke produced.

A series of transport instruments (buckets, conveyor belts, etc.) are used to carry the coal from the bunker to the storage bins. This transport is supervised by a machine that scans and detects the metals that might be present so as to prevent tears from occurring in the belt. The bins are divided into two categories according to whether they are storage or mixing bins. The latter can hold 250 tonnes and there are 10

of them. These allow the coals from different origins to be blended in order to obtain the required chemical characteristics. This blending is carried out under the bins: the desired amounts of coal are poured from each bin into buckets and from there onto a single conveyor belt which carries the mixture for milling. The storage bins have a capacity ten times larger than those used for mixing and can be used to fill the latter if needed.

The first treatment acts on the physical nature of the coal since it involves the crushing, or milling, of the load that has been carried from the mixing bins. After travelling along a series of conveyor belts, a mill crushes the different coals present in the blend in order to produce a homogeneous granulometry, which differs according to the origin of the coals. A metal detector identical to the one used when the raw coal was being transported is used to monitor the operations in order to protect the machinery. After milling, the product that is obtained is known as the "coal charge" and is carried on a series of belts from the milling tower to the coal tower. This tower is located above the ovens and has a capacity of 3 600 tonnes, that is, the amount needed for a day's production.

Oven Section. Before beginning the process of actually distilling the coal charge, the charge must be transferred to the ovens. The team in charge of the ovens puts a certain amount of charge into a machine called a charging machine. This mechanism, which is situated above the oven batteries, then fills the ovens through openings in the top—the charge holes. These openings are in the shape of bottlenecks and each oven has four of them. During loading a "leveller bar", which is part of the pusher car, flattens the coal charge evenly throughout the oven to ensure a better distribution of the charge and a more homogeneous heating process. The loadings, or cokings, follow a very precise schedule that attempts to maximise the number of charges performed per day.

Once charging is finished, the openings at the top and in the sides are closed and the process of firing the coal charge begins. The operation lasts between 16 and 19 hours, depending on the battery the oven is located in. It is, however, difficult to estimate the exact number of hours required for firing because of the numerous uncertainties linked with the ovens. Charges are performed in steps of five, which means that the procedure involves charging oven (n+5) after having charged oven (n). This method prevents large variations in temperature from taking place within the ovens.

The ovens are assembled in series called batteries. There are four of these and are made up of between 20 and 50 ovens, which gives a total assembly of 122 ovens at the CARSID coking plant. The properties of the ovens as regards theoretical firing time and the amounts that can be charged differ according to the battery the ovens belong to. The four batteries are lined up in a row with the coal tower situated in the middle. This layout allows machinery to circulate more easily from one oven to the next.

The reason why the ovens are arranged in batteries is that it helps to reduce heat losses in each of them. The space between the ovens, called a side wall, is made up of a series of vertical stacks which the gases pass through, allowing firing to take place inside the ovens. Thus, these stacks, called flues, help firing in two adjacent ovens. Heat exchange takes place by conduction without direct contact with the coal charge. The gases used come either from the coking plant or from the blast furnace, which enables the coking plant to remain independent on the energetic level. The combustion fumes are recovered by passing them through piles of refractory bricks during a first cycle. In a second cycle, the combustion gases are blown through them in order to recover their heat. The passage from one cycle to the other is called "inversion" and takes place every half an hour.

The oven reaches a temperature above 1 200°C during firing, which allows the coal charge to be transformed into red-hot coke, which already possesses the required final characteristics. The gases released from the distillation process are recovered so that they can be evaluated.

Cooling Section. When firing finishes, the pushing operation, that is to say the emptying of the oven, begins. To perform this process, the side doors of the oven are opened and the pusher car pushes the oven load out. Then the coke guide, a mobile troughed passageway, receives the charge and transfers it to a bogie, the quenching car, and the cooling operation begins. The quenching car is led towards the quenching tower and 25m³ of water is poured onto it. This must cool and quench the incandescent coke. Most of this water evaporates directly on contact with the coke. The waste waters are treated in a settling tank for later reuse. The coke is sprayed by hand should it be only partially quenched.

In the next stage the coke is "put on standby", which involves discharging it onto an inclined "coke wharf" and leaving it to lose some of its moisture. After a waiting period of about thirty minutes, the

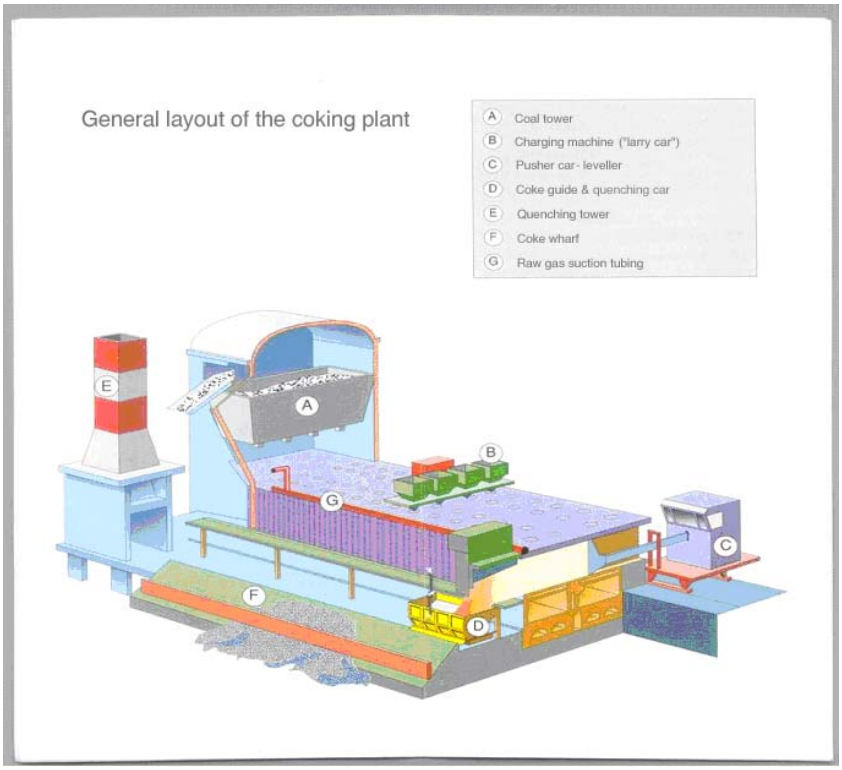


Fig 1: General layout of the coking plant: coking, pushing and cooling

coke car transfers the coke obtained to the coarse coke tower. There, the screening process starts. This involves separating out the fragments considered to be "metallurgical coke", i.e. those that can be used in the manufacture of steel, from others that will be used for other purposes. The selection is performed according the size of the pieces, which must be big enough to provide the charge with good permeability. The fragments that are considered to be suitable for metallurgical use are sent to the blast furnace, while the "small coke" is carried off to the sinter.

Certain reparation tasks must be carried out on the ovens. Teams of builders are responsible for filling cracks and mending other problems that may arise concerning the ovens. These workers use them and know the problems that can come about in the bricks that line them. Repairing an oven is done at a temperature of 700°C. It is impossible to go below this temperature as the bricks would change their state and become as brittle as glass. Repair work is therefore a serious, difficult job, bearing in mind the high temperatures that must be dealt with.

Finally, there is a measurement and adjustment team that is in charge of supervising the proper functioning of the coke production machinery. The temperature of the ovens is controlled by measuring the heat present in the flues of the ovens, and can therefore be adjusted. This team is also in charge of inverting the heating cycle of the battery, which takes place about every half an hour.

3. DATABASE REENGINEERING

3.1 Introduction

Reengineering is a very wide concept which can be applied to a large number of fields. We should however start by this somehow ambiguous point in order to try a more focused approach to the object of our work, being *database reengineering*. A possible definition for reengineering would be the following:

Reengineering is any activity that:
1. Improves the understanding of the software.
2. Prepares or improves one's own software, usually so as to increase its maintenance, reusing or evolving easiness.

Arnold (1993)

We may apply Arnold's proposal, especially its second point, in order to obtain a working definition of *database reengineering*. Thus, database reengineering would be the activity that aims at improving an existing database either improving the internal structure of the data model, making this model evolve or widen its semantics, or integrating databases. These activities will

help us reuse or make the database evolve.

3.2 Steps for database reengineering

Database reengineering is not one specific method or system. There are however some important steps that can be established which are closely related to the steps followed in software development (Fig. 2).

But when carrying out the reengineering of a database, different steps are also needed, and this is so because we already have a database to start from. In order to understand how we can carry out the reengineering of a previous database or how we can redesign an existing database, it is necessary to introduce the concept of *reverse engineering*. This

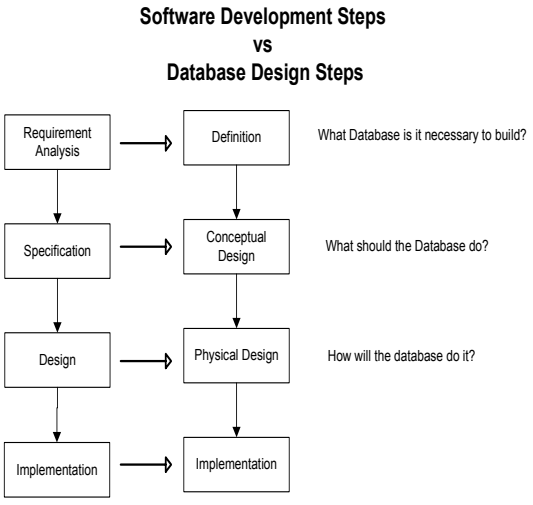


Fig 2: Development Steps: Software vs Database

concept may be defined as follows:

The process of building abstract formal specifications from the source code of a legacy system, so that they can be used to build a new implementation of the system using forward engineering.

Arnold (1993)

Therefore, in order to apply reengineering processes to any database we will have to work with direct engineering, reverse engineering, and, obviously, reengineering. Figure 3 depicts the way how a database development cycle may be understood, from the original creation of the database to the moment where a reengineering process is conducted. Steps 1 to 3 in the figure show the necessary steps to be taken in creating a data base with direct engineering and steps 4 to 9 would be those to be taken in reengineering.

These steps do not constitute one restrictive method (Castellanos 2001), and there will be some times where some of the steps cannot or would not be applied, and some other times where the sequence of steps (Fig. 3) will have to be modified according to the necessities of the case at hand.

3.3 Processes for database reengineering

The principal idea underlying database reengineering is trying to design a new database model from an existing one, making the present design evolve. And this can be done following several steps in different orders, as already stated. The process to be adopted will depend on several factors, among which we want to emphasize the reason why we are redesigning the database. Mainly, there can be three reasons for doing so:

1. The database is not normalized (the design does not meet the Boyce-Codd Normal Form (BCNF) criteria).
2. There is a need for extending the present model in order to include new semantics.
3. There is a need for integrating some previous models from existing databases.

In the following headings from this section we will focus on the principal processes for redesigning databases when one or some of the reasons stated are involved.

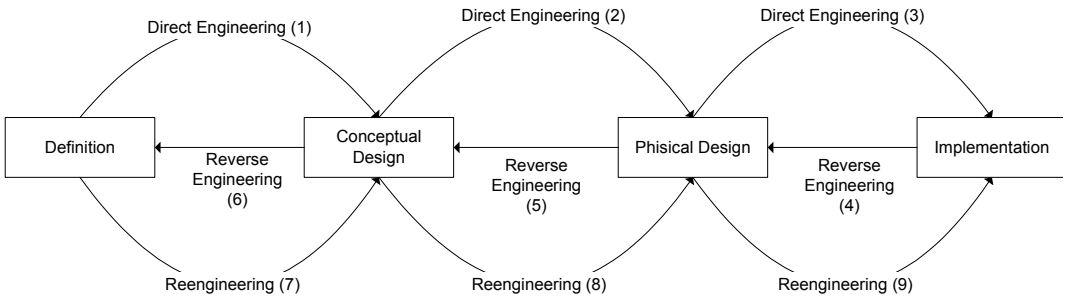


Fig. 3: Database Reengineering Steps

The database is not normalized. Normalizing a database is to improve the structure of our database so as to ensure the best possible structure for the data that it contains. Every database needs to be normalized for two principal goals: removing redundant data (for example, when the same data are stored in more than one table) and ensuring data dependencies make sense (storing only related data in every table). Normalizing is a useful concept because when the database design is good, the results obtained from the exploitation of the data is much better.

In order to normalize a database, it is important to apply a series of criteria, the normal forms, consisting of six rules (two of them forming one only set and known as third normal form and Boyce-Codd normal form respectively) that lead to a normalized database. Among these rules, the four first (first, second and Boyce-Codd normal forms) should be applied to have a normalized database. It is also possible to go further in order to obtain a more refined database model applying fourth and fifth normal forms wherever possible. In any case, we will first apply first to Boyce-Codd normal form –thus having a normalized database– and then we can apply the other two criteria.

Description of the Basic normal forms

First normal form (1NF) sets the very basic rules for an organized database:
- Eliminating duplicate columns in the same table.
- Creating separate tables for each group of related data and identifying every row with one column or set of columns (the primary key).

Second normal form (2NF) further addresses the concept of removing duplicate data:
- Removing subsets of data that apply to multiple rows of a table and placing them in separate rows.
- Creating relationships between these new tables and their predecessors through the use of foreign keys.

Third normal form (3NF) goes one large step further:
- Removing columns that are not dependent upon the primary key.
- There is one final requirement, also known as Boyce-Codd normal form (BCNF):
A relation is in BCNF if and only if every determinant is a candidate key.
We need to remember that these normalization guidelines are cumulative. Thus, for a database to be in 2NF, it must first fulfil all the criteria of a 1NF database.

Comments on Development Steps
Whenever we have to normalize a database, the process will generally be more easily followed if we work at the design level. This is so because at this level it is easier to understand all the attributes, primary keys, foreign keys, and relationships between the tables.

Extending the current model to include new semantics. The usual reason for extending a model is generally to prepare our database for supporting a new process or including a new system. These new process will become new parts from one existing database and will have to be developed anew. As a result, the most adequate way to proceed in extending the model is to apply the general steps for designing a database. Also, studying the existing

databases becomes essential in order to avoid creating redundant data or designing again some already designed part.

Integrating database designs. The goal in this case is to integrate two or more databases in one only database. We can establish two ways to proceed when integrating two different databases:

The first one uses some kind of automatic system within the theories of string matching. When comparing two different databases, we are confronted with a great deal of different data. The easiest system for doing so is to convert all data into one single format. The format that can represent the others is the string format. Therefore, we have to find a good system for comparing strings (for example the Levenshtein distance method).

There is also another way in which the integration is done manually. The key idea here is to integrate the models while respecting the normal forms, and not to introduce any redundant data. When we are integrating a database manually, we have to study first the two databases and fully understand the meaning of the attributes. Secondly, we have to ensure that the contents of every database share the same semantics. Afterwards, we have to redesign the tables and the relationships among them so that they are in the same semantic area, having in mind one only final database. After having redesigned the databases, it is important to check out that the semantics present in the two initial databases can be found in the resulting database, and that the resulting database is normalized.

3.4 Reengineering the CARSID database

Introduction. When we first studied the information system being used by CARSID, we discovered that there were three different databases in the system. The most important one, called *Cokerie*, is the database which controls the most important steps in the production process. Another database controls the system of weigh and is called *Bascule*, and there is still one third database which controls the transport for introducing raw material in the plant and taking out the final product (coke) and is called *Traction*. In this section we will focus on the reengineering of these databases pertaining to the CARSID information system.

As for the reasons underlying the reengineering, we have already pointed out that such a decision can be taken on three different grounds. In the case of CARSID, all three reasons were relevant, although this does not always have to be the case. The three reasons were:

1. Normalizing the databases.
2. Creating new semantics for the database model.
3. Integrating the previously existing databases.

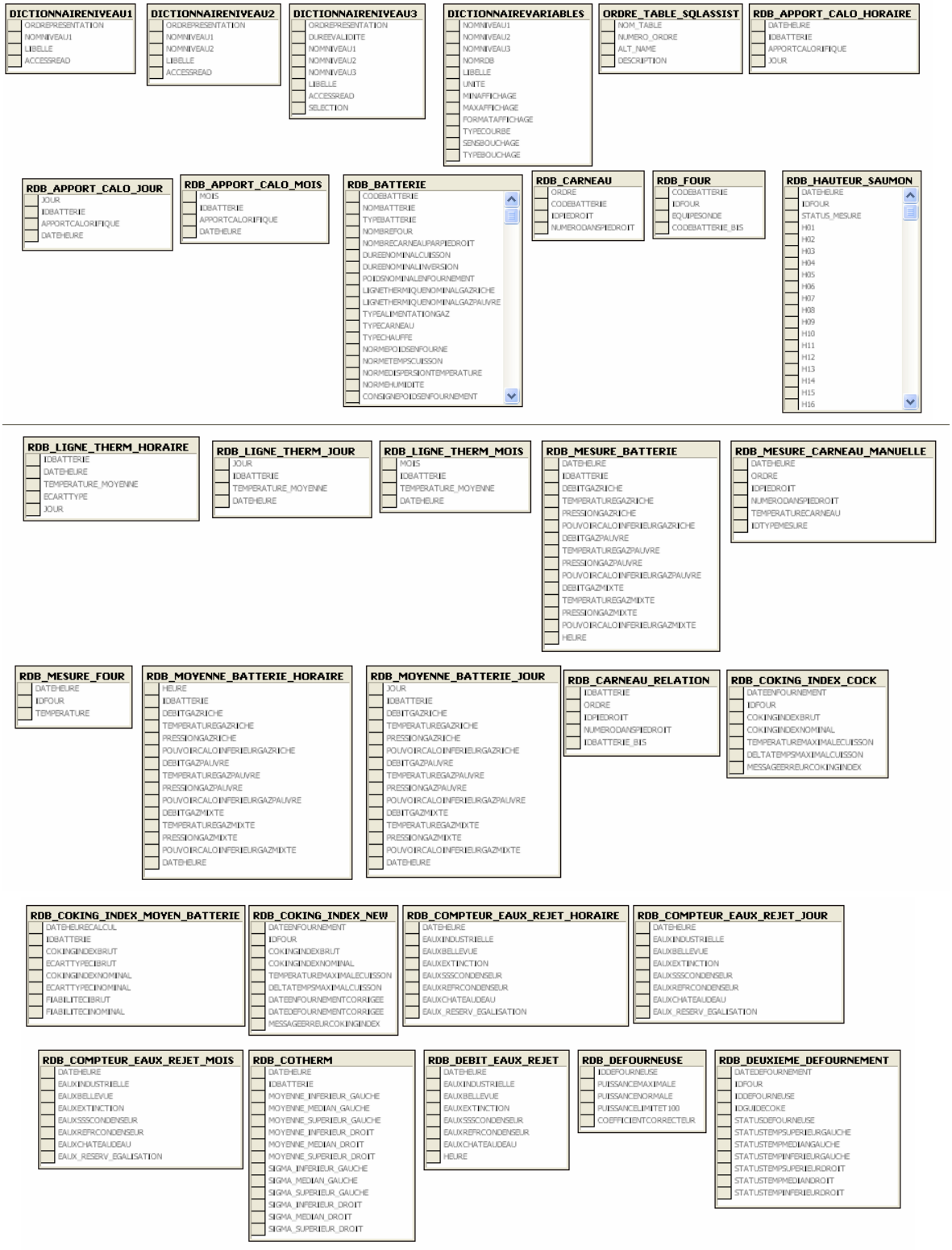
The steps we followed were aimed first at normalizing and second at integrating. We applied both steps in this order because it is much more difficult to integrate a database which is not normalized. Even with normalized databases, integrating is a complex task because it is possible to find redundant data in the databases or the tables we are working with.

The method that we used in the reengineering was based on the interviews had with the technical staff from CARSID. We will however start explaining some steps of reverse engineering we had to take at the beginning of the job because they will be needed in the explanation of the interviews.

The process of reverse engineering

In order to start our work, we applied a process of reverse engineering for obtaining a physical model of the databases using the scripts that the staff at CARSID gave to us. We took the script that they had in SQL for the databases in sysbase and SQL for the ones in VAX; then, using the rational rose tool, we regenerated a model. After this, we generated a new script for Microsoft SQL Server. You can find enclosed the global physical model.
The models that we can see in the next sections are in SQL Server 7.0 and SQL Server 2000.

Reengineering of the database *Cokerie*. The redesigning of this database was the most difficult one, because this is the most complex and biggest database. At this point, we will just comment the most relevant aspects of the new database and we will show the original database (Fig. 4).


DICTIONNAIRENIVEAU1
ORDREREPRESENTATION
NOMNIVEAU1
LIBELLE
ACCESSREAD
DICTIONNAIRENIVEAU2
ORDREREPRESENTATION
NOMNIVEAU1
NOMNIVEAU2
LIBELLE
ACCESSREAD
DICTIONNAIRENIVEAU3
ORDREREPRESENTATION
DUREEVALIDITE
NOMNIVEAU1
NOMNIVEAU2
NOMNIVEAU3
LIBELLE
ACCESSREAD
SELECTION
DICTIONNAIREVARIABLES
NOMNIVEAU1
NOMNIVEAU2
NOMNIVEAU3
NOMBDB
LIBELLE
UNITE
MINAFFICHAGE
MAXAFFICHAGE
FORMATAFFICHAGE
TYPECOURBE
SENSBOUCHAGE
TYPEBOUCHAGE
ORDRE_TABLE_SQLASSIST
NOM_TABLE
NUMERO_ORDRE
ALT_NAME
DESCRIPTION
RDB_APPORT_CALO_HORAIRE
DATEHEURE
IDBATTERIE
APPORTCALORIFIQUE
JOUR
RDB_APPORT_CALO_JOUR
JOUR
IDBATTERIE
APPORTCALORIFIQUE
DATEHEURE
RDB_APPORT_CALO_MOIS
MOIS
IDBATTERIE
APPORTCALORIFIQUE
DATEHEURE
RDB_BATTERIE
CODEBATTERIE
NOMBATTERIE
TYPEBATTERIE
NOMBREFOUR
NOMBRECARNEAUPARPIEDROIT
DUREENOMINALCUISSON
DUREENOMINALINVERSION
POIDSNOMINALENFOURNEMENT
LIGNETHERMIQUENOMINALGAZRICHE
LIGNETHERMIQUENOMINALGAZPAUVRE
TYPEALIMENTATIONGAZ
TYPECARNEAU
TYPECHAUFFE
NORMEPOIDSENFOURNE
NORMETEMPSCUISSON
NORMEDISPERSIONTEMPERATURE
NORMEHUMIDITE
CONSIGNEPOIDSENFOURNEMENT
RDB_CARNEAU
ORDRE
CODEBATTERIE
IDPIEDROIT
NUMERODANSPIEDROIT
RDB_FOUR
CODEBATTERIE
IDFOUR
EQUIPESONDE
CODEBATTERIE_BIS
RDB_HAUTEUR_SAUMON
DATEHEURE
IDFOUR
STATUS_MESURE
H01
H02
H03
H04
H05
H06
H07
H08
H09
H10
H11
H12
H13
H14
H15
H16
RDB_LIGNE_THERM_HORAIRE
IDBATTERIE
DATEHEURE
TEMPERATURE_MOYENNE
ECARTTYPE
JOUR
RDB_LIGNE_THERM_JOUR
JOUR
IDBATTERIE
TEMPERATURE_MOYENNE
DATEHEURE
RDB_LIGNE_THERM_MOIS
MOIS
IDBATTERIE
TEMPERATURE_MOYENNE
DATEHEURE
RDB_MESURE_BATTERIE
DATEHEURE
IDBATTERIE
DEBITGAZRICHE
TEMPERATUREGAZRICHE
PRESSIONGAZRICHE
POUVOIRCALOINFERIEURGAZRICHE
DEBITGAZPAUVRE
TEMPERATUREGAZPAUVRE
PRESSIONGAZPAUVRE
POUVOIRCALOINFERIEURGAZPAUVRE
DEBITGAZMIXTE
TEMPERATUREGAZMIXTE
PRESSIONGAZMIXTE
POUVOIRCALOINFERIEURGAZMIXTE
HEURE
RDB_MESURE_CARNEAU_MANUELLE
DATEHEURE
ORDRE
IDPIEDROIT
NUMERODANSPIEDROIT
TEMPERATURECARNEAU
IDTYPEMESURE
RDB_MESURE_FOUR
DATEHEURE
IDFOUR
TEMPERATURE
RDB_MOYENNE_BATTERIE_HORAIRE
HEURE
IDBATTERIE
DEBITGAZRICHE
TEMPERATUREGAZRICHE
PRESSIONGAZRICHE
POUVOIRCALOINFERIEURGAZRICHE
DEBITGAZPAUVRE
TEMPERATUREGAZPAUVRE
PRESSIONGAZPAUVRE
POUVOIRCALOINFERIEURGAZPAUVRE
DEBITGAZMIXTE
TEMPERATUREGAZMIXTE
PRESSIONGAZMIXTE
POUVOIRCALOINFERIEURGAZMIXTE
DATEHEURE
RDB_MOYENNE_BATTERIE_JOUR
JOUR
IDBATTERIE
DEBITGAZRICHE
TEMPERATUREGAZRICHE
PRESSIONGAZRICHE
POUVOIRCALOINFERIEURGAZRICHE
DEBITGAZPAUVRE
TEMPERATUREGAZPAUVRE
PRESSIONGAZPAUVRE
POUVOIRCALOINFERIEURGAZPAUVRE
DEBITGAZMIXTE
TEMPERATUREGAZMIXTE
PRESSIONGAZMIXTE
POUVOIRCALOINFERIEURGAZMIXTE
DATEHEURE
RDB_CARNEAU_RELATION
IDBATTERIE
ORDRE
IDPIEDROIT
NUMERODANSPIEDROIT
IDBATTERIE_BIS
RDB_COKING_INDEX_COCK
DATEENFOURNEMENT
IDFOUR
COKINGINDEXBRUT
COKINGINDEXNOMINAL
TEMPERATUREMAXIMALECUISSON
DELTATEMPSMAXIMALCUISSON
MESSAGEERREURCOKINGINDEX
RDB_COKING_INDEX_MOYEN_BATTERIE
DATEHEURECALCUL
IDBATTERIE
COKINGINDEXBRUT
ECARTTYPECIBRUT
COKINGINDEXNOMINAL
ECARTTYPECINOMINAL
FIABILITECIBRUT
FIABILITECINOMINAL
RDB_COKING_INDEX_NEW
DATEENFOURNEMENT
IDFOUR
COKINGINDEXBRUT
COKINGINDEXNOMINAL
TEMPERATUREMAXIMALECUISSON
DELTATEMPSMAXIMALCUISSON
DATEENFOURNEMENTCORRIGEE
DATEDEFOURNEMENTCORRIGEE
MESSAGEERREURCOKINGINDEX
RDB_COMPTEUR_EAUX_REJET_HORAIRE
DATEHEURE
EAUXINDUSTRIELLE
EAUXBELLEVUE
EAUXEXTINCTION
EAUXSSSCONDENSEUR
EAUXREFRCONDENSEUR
EAUXCHATEAUDEAU
EAUX_RESERV_EGALISATION
RDB_COMPTEUR_EAUX_REJET_JOUR
DATEHEURE
EAUXINDUSTRIELLE
EAUXBELLEVUE
EAUXEXTINCTION
EAUXSSSCONDENSEUR
EAUXREFRCONDENSEUR
EAUXCHATEAUDEAU
EAUX_RESERV_EGALISATION
RDB_COMPTEUR_EAUX_REJET_MOIS
DATEHEURE
EAUXINDUSTRIELLE
EAUXBELLEVUE
EAUXEXTINCTION
EAUXSSSCONDENSEUR
EAUXREFRCONDENSEUR
EAUXCHATEAUDEAU
EAUX_RESERV_EGALISATION
RDB_COTHERM
DATEHEURE
IDBATTERIE
MOYENNE_INFERIEUR_GAUCHE
MOYENNE_MEDIAN_GAUCHE
MOYENNE_SUPERIEUR_GAUCHE
MOYENNE_INFERIEUR_DROIT
MOYENNE_MEDIAN_DROIT
MOYENNE_SUPERIEUR_DROIT
SIGMA_INFERIEUR_GAUCHE
SIGMA_MEDIAN_GAUCHE
SIGMA_SUPERIEUR_GAUCHE
SIGMA_INFERIEUR_DROIT
SIGMA_MEDIAN_DROIT
SIGMA_SUPERIEUR_DROIT
RDB_DEBIT_EAUX_REJET
DATEHEURE
EAUXINDUSTRIELLE
EAUXBELLEVUE
EAUXEXTINCTION
EAUXSSSCONDENSEUR
EAUXREFRCONDENSEUR
EAUXCHATEAUDEAU
HEURE
RDB_DEFOURNEUSE
IDDEFOURNEUSE
PUISSANCEMAXIMALE
PUISSANCENORMALE
PUISSANCELIMITET100
COEFFICIENTCORRECTEUR
RDB_DEUXIEME_DEFOURNEMENT
DATEDEFOURNEMENT
IDFOUR
IDDEFOURNEUSE
IDGUIDECOKE
STATUSDEFOURNEUSE
STATUSTEMPSUPERIEURGAUCHE
STATUSTEMPMEDIANGAUCHE
STATUSTEMPINFERIEURGAUCHE
STATUSTEMPSUPERIEURDROIT
STATUSTEMPMEDIANDROIT
STATUSTEMPINFERIEURDROIT

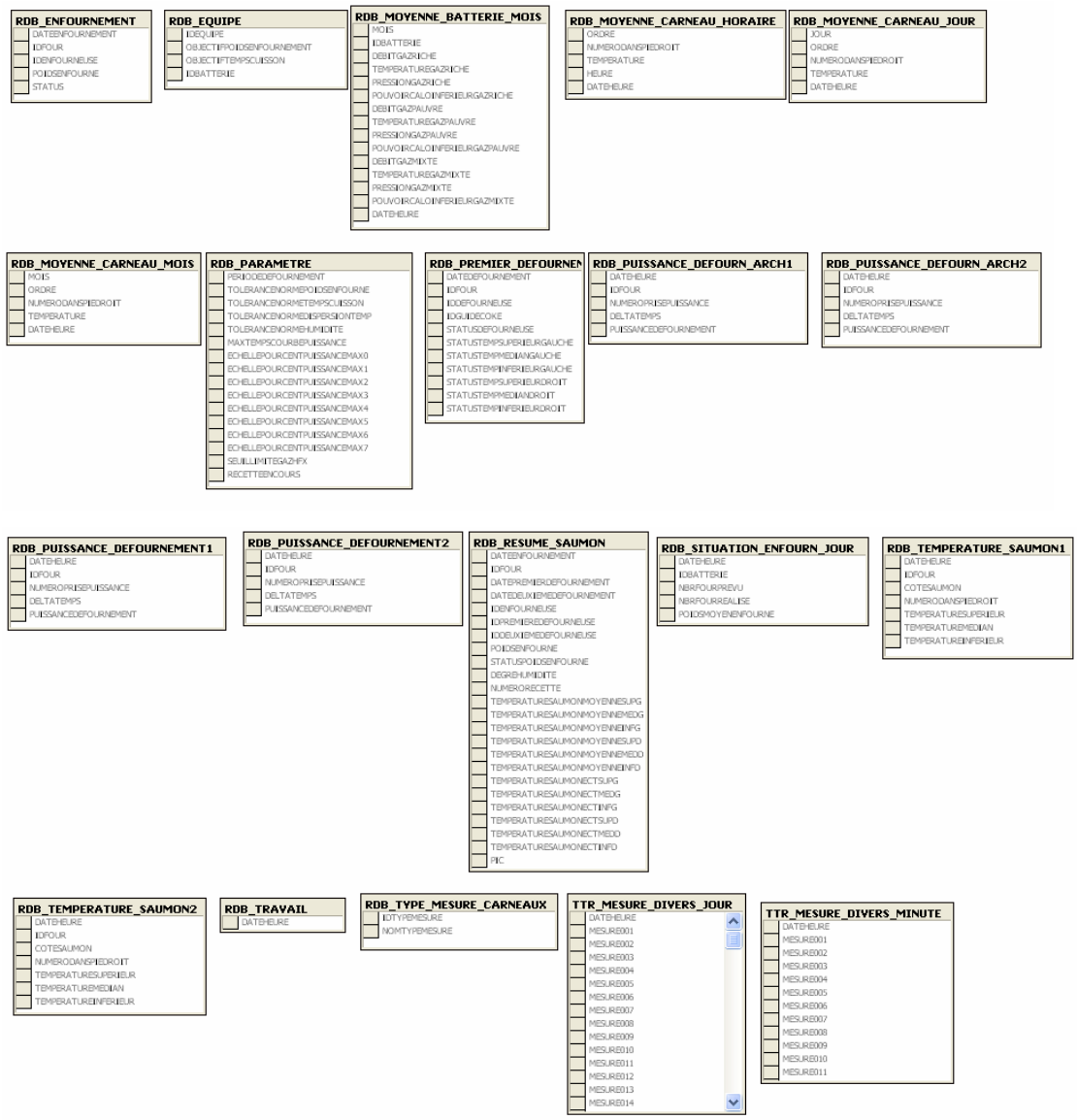


Fig. 4: Original Cokerie Database

The hardest part in redesigning this database were the tables 'Four', 'Batterie' and 'Piedorit' (Fig. 5). These critical tables represent the essential structure of a battery. A battery is composed by a set of ovens and piedroits. Every oven is related to two different piedroits.

One part of the database supports a planning system for oven charge (*enfournement*) and discharge (*defournement*). This part is a very important part of the database because it helps the system generate the organization of the tasks for oven charge and discharge. These tasks are really important because help the factory obtain a greater capacity of production. The new model (Fig. 6) stores the original planning in the table 'PlanningDefournement' and then the result is transferred to the table 'PdefResultat'. This procedure allows us to compare the planning to the real state of affairs and to improve and tune the planning accordingly. The system stores the results for every oven separately and includes a

great quantity of variables. With these results the enterprise can subsequently carry out research and studies in the production and the capacity and quality of every oven.

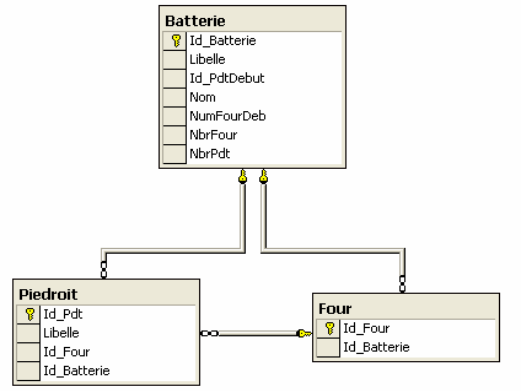


Fig. 5: The Heart of the New Cokerie Design

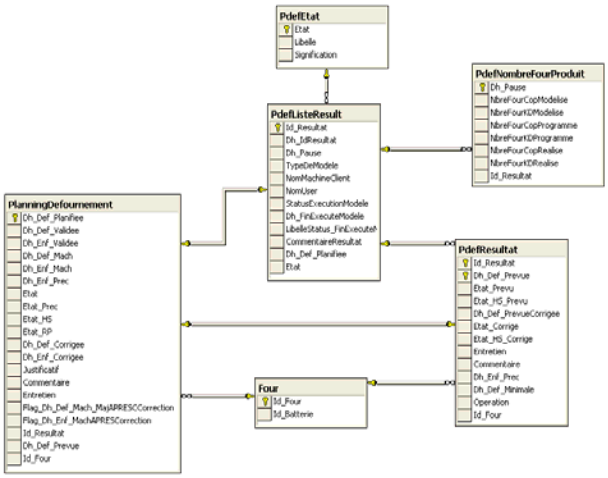


Fig. 6: The Planning of the New Cokerie Design

The most crucial point in the whole process is the oven discharge (Fig. 7), because this action is the most complex and it may cause some problems. In this step of the manufacturing process, the coal (at approximately 1200ºC) is taken out of the oven. The machines involved in the carrying out of this particular action can control the power they exert in 113 different points. These power adjustments are stored automatically with their respective 113 variables in the table 'Defournement_Puissance' every time the oven is discharged. The same machines also store the profile of the resulting coke in the table 'Defournement_PorfileCoke' with 88 more variables every time the oven is discharged. The 'cotherm' is stored in the table 'Defournament_Cotherm' with 177 variables.

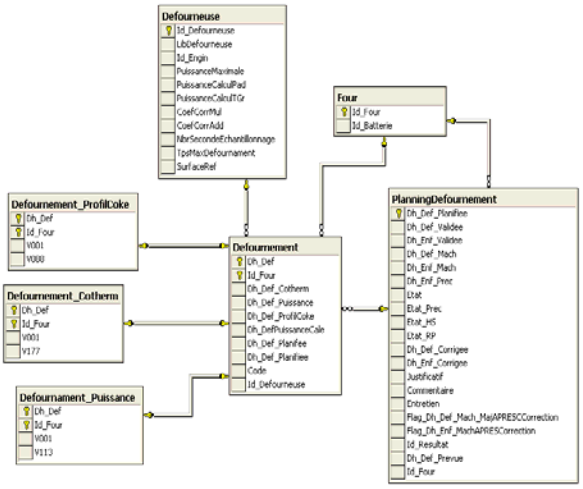


Fig. 7: The Oven Discharge of the New Cokerie Design

In the new database (Fig. 8 for the conceptual model) design there are some tables without relations. These tables are disconnected for different reasons. The tables 'MesureSynchroneBatterieMinute', 'MesureSynchroneDiversJour', 'MesureSynchroneDiversMinute' and 'MesureSynch-roneDiversPause' store data for historical reasons. This data are synchronically stored and are generally used to retrieve information about the behaviour of the battery or about any parameters of the production system in two periodical forms: one row for day or one row for minute. The tables 'RecapJourCoProduit', 'RecapPauseCoProduit' and 'RecapPause-Four' store the historical data about any special events of production and contain a summary of the production events occurred each day.

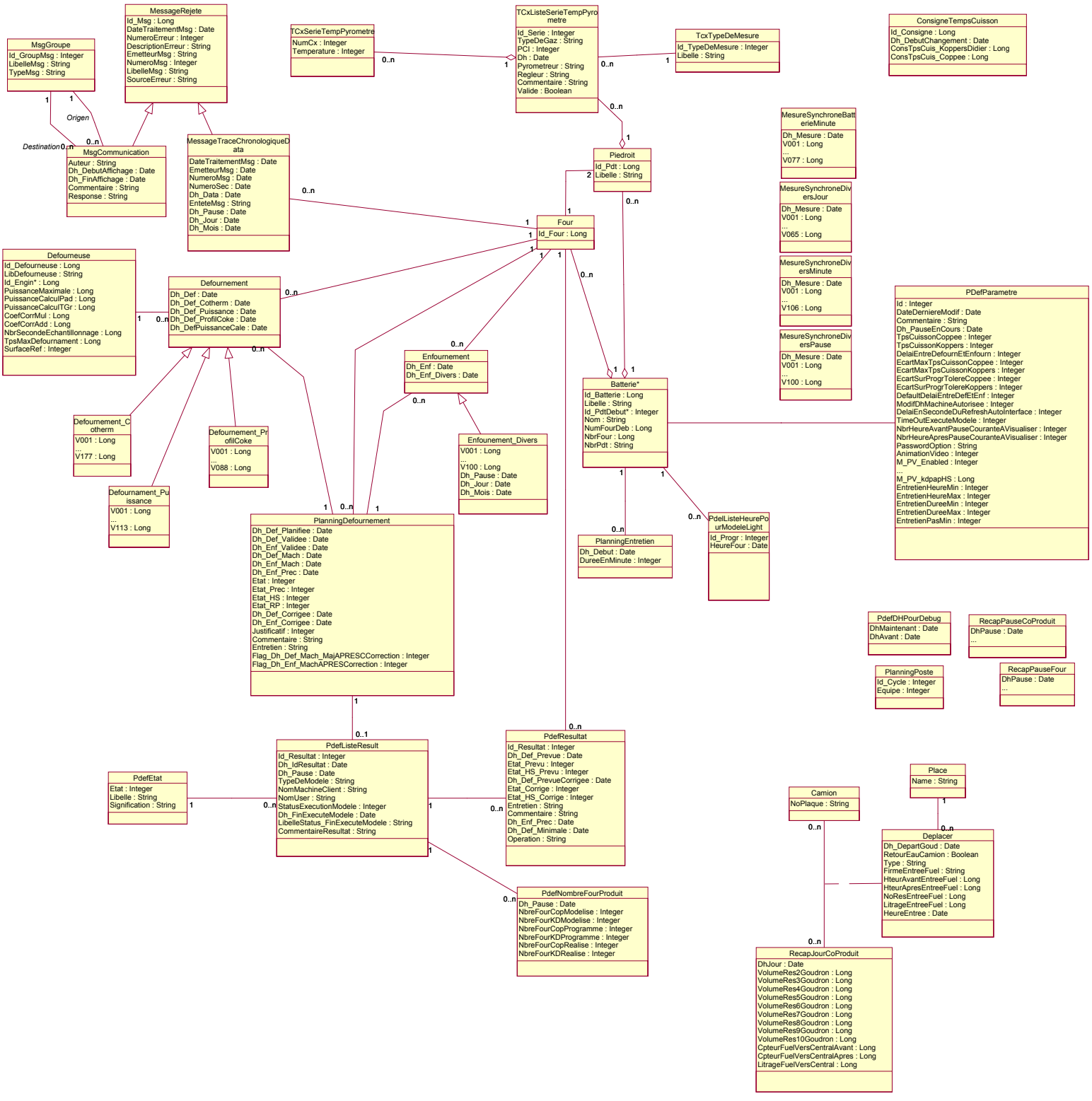


Figure 8: Conceptual model of the cokerie

Reengineering the database *Traction*. The database *Traction* is used for controlling the process through which the raw material comes into the factory by train and the one through which the coke goes out to another plant. Theoretically, it controls the situation of the wagons, the material that every wagon transports, and the composition of the trains with his locomotives, tenders and wagons.
We normalized the original database, and the resulting one can be seen in Fig. 9 We had some problems in normalizing the model because there were some tables that had not been used and some of the information was duplicated. Moreover, no integrity relationship had been established and, as a consequence, we had to introduce these relationships between all the tables.

In the redesigning of this database, the work was done at the conceptual level (Fig. 10). We had some external problems, since the staff from CARSID did not want to introduce the concept 'train', because major changes would be needed as a consequence of its introduction. Finally, we avoided using this concept by suggesting another solution in which the *class* 'train' was used. This new solution was finally accepted because it is most similar to the existing database and the changes needed in order to use the application with this model are less drastic.

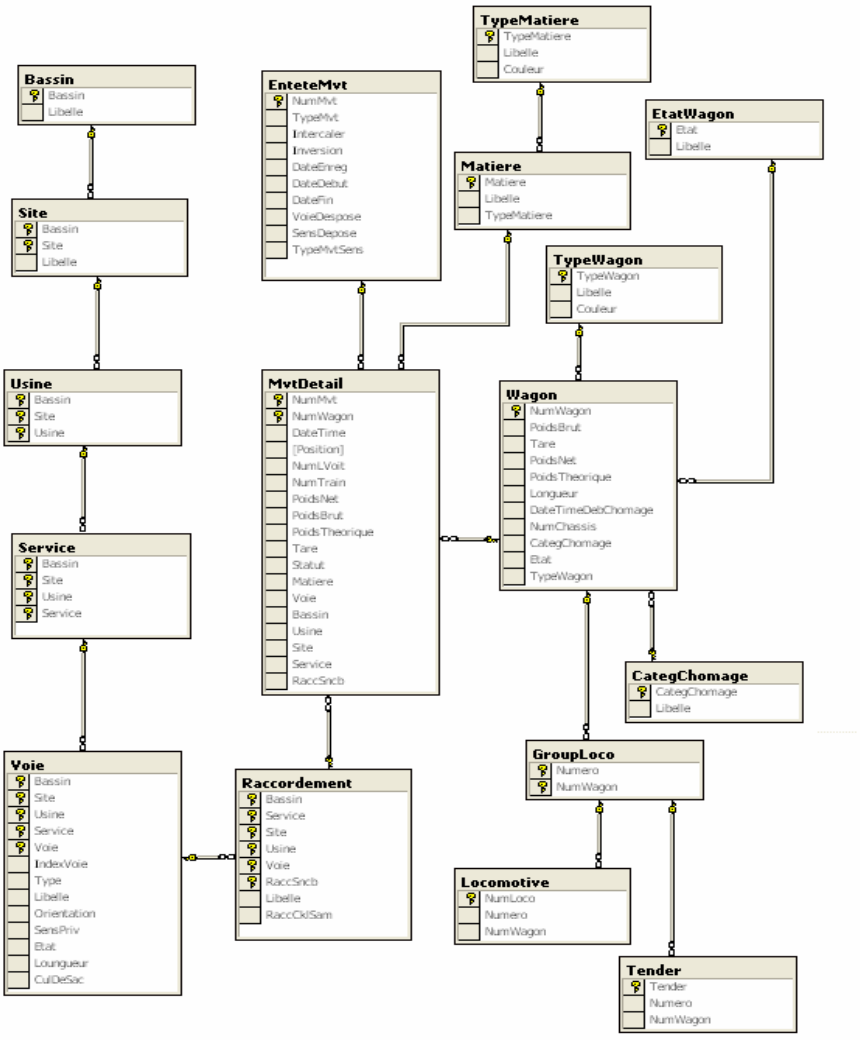


Fig. 9: New Traction Database (Physical Model)

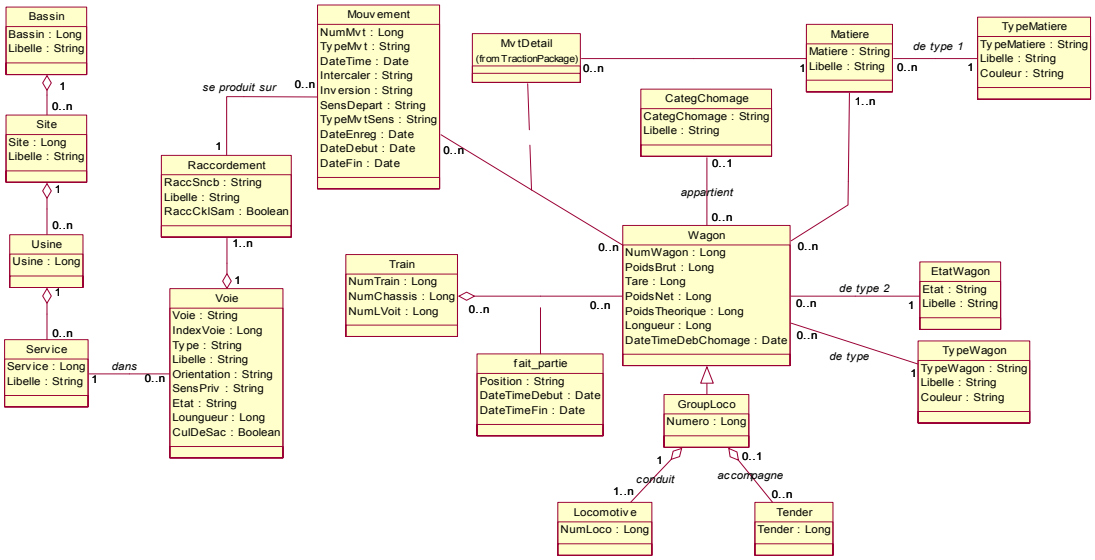


Fig. 10: New Traction Conceptual Model

Reengineering the database *Bascule*. This database is used to control all trucks and materials which come in and go out of the factory. Also, this database controls the administration process of the firm's clients and suppliers which send and receive material to or from CARSID. The original database was very denormalized and we had to redesign and modify all tables, because the system used to store the data was far from acceptable – all the more if we want to follow the normalization norms. Theoretically, the system used is aimed at introducing all the data in the 'badge' table and using the trigger technology to copy some data in other tables.

In the new version (Fig. 11), we introduced many changes. We deleted many fields from the existing

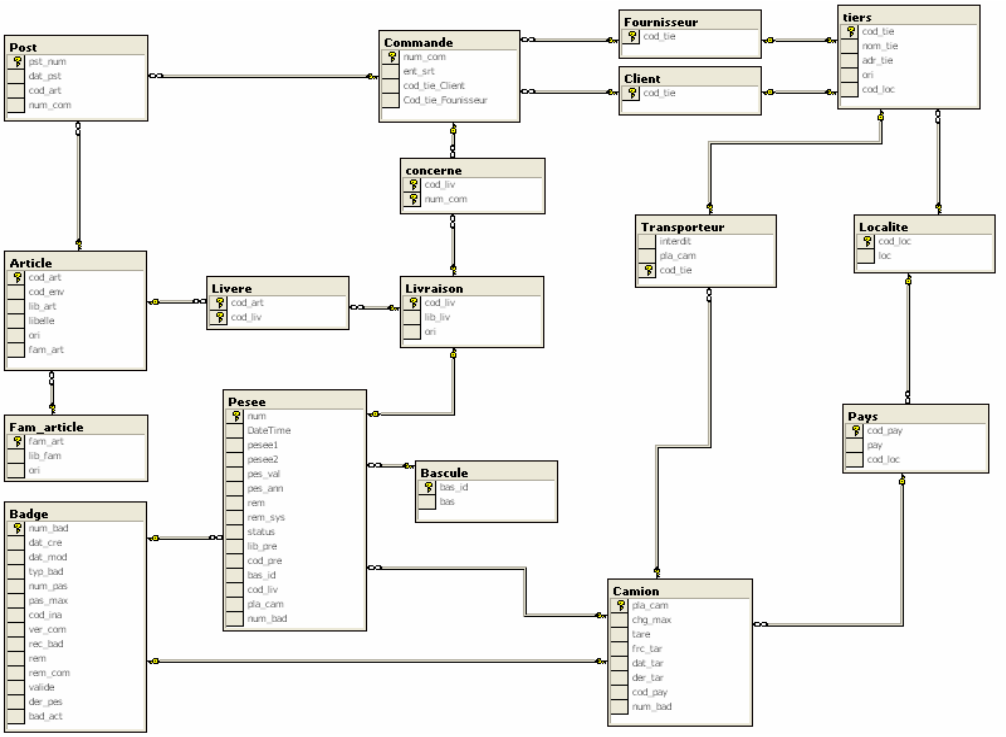


Fig. 11: New Bascule Database (Physical Model)

tables and we were forced to introduce some new tables. For instance, we deleted many fields from the table ‘badge’ and also moved many fields from the table ‘badge’ to the ‘Pesee’ table. We also decomposed the table ‘command’ in two different tables, ‘command’ and ‘post’. We introduced all the integrity relationships between all tables. We introduced the generalization/specialization of the table ‘tiers’ into ‘Client’, ‘Fournisseur’ and ‘Transporteur’ (see the generalization/specialization in Fig. 12). The relationship between 'badge' and 'Camion' was established in order to know what truck does actually have the 'badge'; this relationship is very necessary because when a truck comes and goes to the ‘badge’ for the material to be weighted we have to localize what truck is doing this action. We will not store the historical relationship between the badges and the trucks because the people at CARSID were not interested in that matter. We will however store the historical data of the table ‘weigh’ ('pesee') and the table ‘truck’ ('Camion') which did the action.

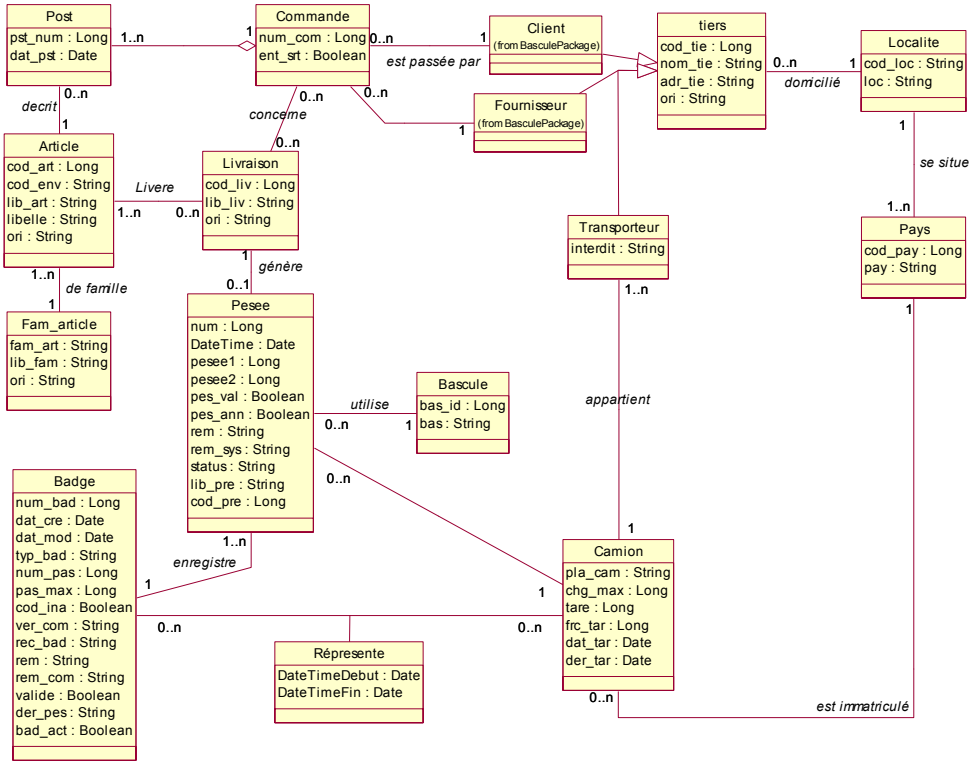


Fig. 12: New Bascule Conceptual Model

Engineering the database of *Stock*. The database 'Stock' (Fig. 13) is the only database we worked with from the beginning, since it did not exist in the previous information system. We found the necessity of creating this new database after trying to find a system to integrate the three existing databases (Bascule, Traction, and Cokerie) and speaking with the staff about the principal problems that they had. From our talks with the personnel we found out that a very important factor in the production process, which should be efficiently controlled through the whole of it in order to improve and minimize final costs, was the material (first as coal, and finally as coke). With this basic common aspect in mind, we went back to the databases we had been studying and realised a common basis may be identified, which was again the material. As a result, we decided to develop a proposal which allowed for the material to be controlled over the whole process and which could be used as the common factor to integrate the existing databases in a global data model. We based our study of the database model in a set of stock controlling tables (Microsoft Excel files) that they presented to us. In this model we tried to control all the materials involved in the manufacturing process, from the raw material to the final product. The material is divided into three groups: coal, the raw material, mixture (table 'Melange'), and coke. The mixture is made of different types of coal, and it is crucial for obtaining a good quality coke. Finally, the coke is the result of cooking this mixture.

We discomposed the evolution of the material into movements and sub-movements, so that the movements could account for the quantity of material and the sub-movements could represent the situation of the material. Sub-movements help see the places where the material is accumulated. When we control the situations where the material is accumulated we can take decisions in order to improve the production process and work to obtain a *just in time* system for minimizing the quantity of material needed to fabricate coke.

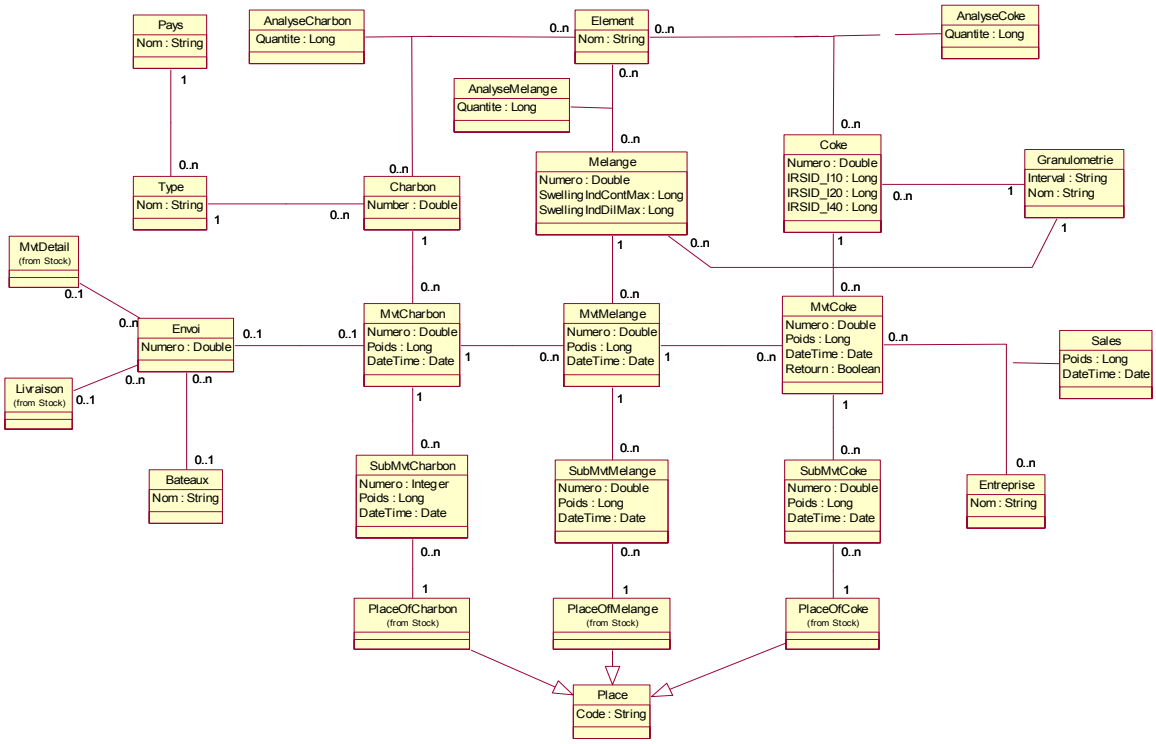


Fig. 13: Stock Conceptual Model

We also stored the granularity of every set of coke and mixture. The granularity is one of the parameters which are used to determine the quality of the product. With this database model the technical staff can know the evolution of the material and they can try to find the cause of either a good result or a bad result. We also added the table 'Sales' in order to know how much material is sold to other enterprises.

We can see the physical model in figure 14.

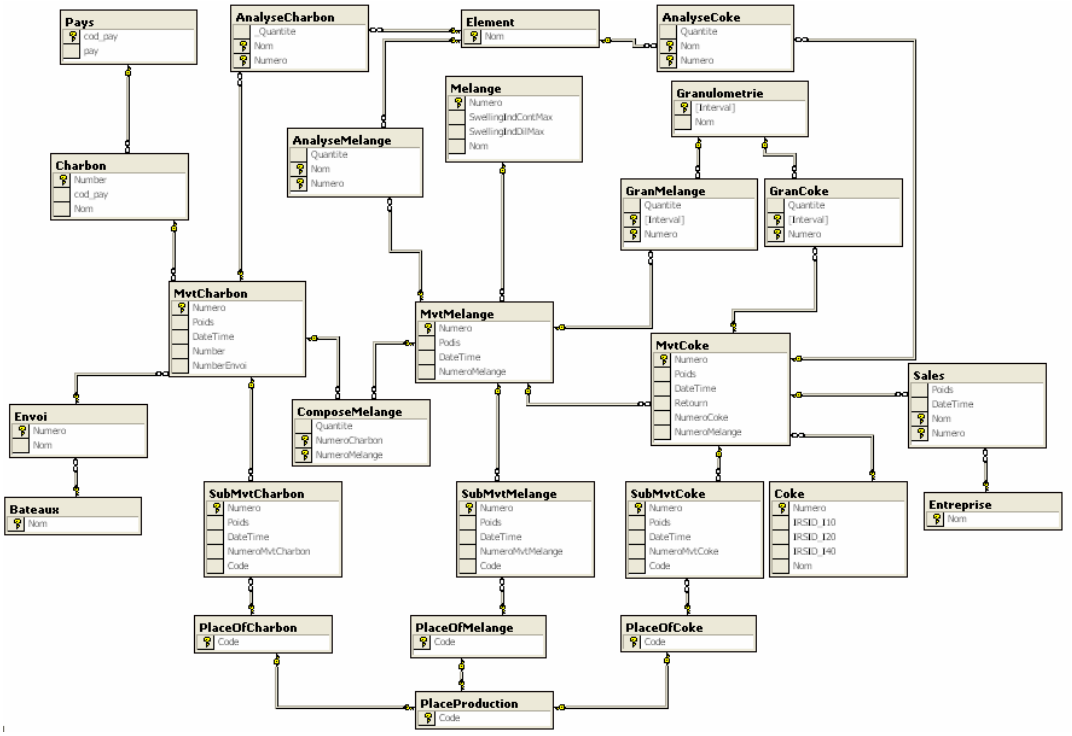


Fig. 14: Stock Database (Physical Model)

Integrating the databases. The integration of the four databases should not very difficult if the previous steps were well done. In this section, we will comment on the method that we used in order to integrate the databases and the way in which we established connecting points between the databases defined in the previous sections, as well as some modifications that could have been made in the whole process.

To start with, the database integration could have been done following an *automatic integration* (see section 3.3). This was not done because, in our case, the fields of the databases were very different, and we soon discarded this option. We could also have tried to apply this technique for the data in the databases, but we also discarded this possibility because there were data with very different meaning and very similar form. For these two reasons, we decided to adopt a *manual integration* (see section 3.3).

In order to find connecting points from which to start the integration of the three existing databases, we analysed the data contained in the databases and thought that the junction was the *material*. From this idea, we established two different points through which the databases could be connected. The first connecting point is the mean used to transport the coal (Fig. 15). The coal can come by ship (table 'Bateaux'), by truck (the database '*Bascule*'), and by train (the database '*Traction*'). This 'mean of transport' allowed us to connect all but one database

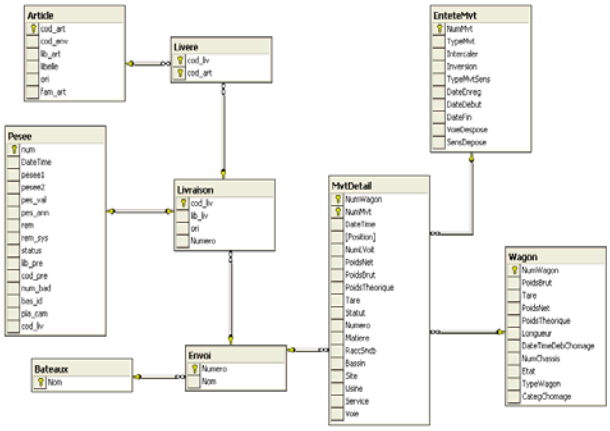


Fig. 15: Connection between Stock Database, Traction Database, and Bascule Database

('Cokerie'). The second connecting point found is the relationship between the actions and the places where they are performed (Fig. 16). The places of action contain the quantity of material which will be used in the action and can therefore be used to connect the databases 'Stock' and 'Cokerie'.

We had now a final database design which connects the models which we had been working with separately. This model is very big and for this reason we are not including the schema in this document. It can be found in the file '\NewDesigns\Schema\CARSIDGeneralDB.pdf' from the annexed CD; the conceptual model, on the other hand, can be found in a Rational Rose file, '\NewDesigns\models\DBModelVF.mdl'.

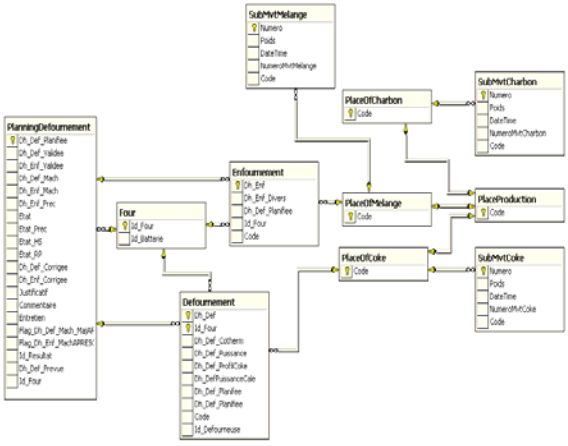


Fig. 16: Connection between the Databases Stock and Cokerie

We would like to make a final remark about some aspects of our work that may call the reader's attention. If we study the final solution we will discover some things which are not completely correct from a theoretical point of view although there are practical reason which required this to be done in this particular way. Namely, there are five different tables which contain data about material. These tables existed in the previous version of the information system and we did not modify them because the applications which use the databases need to work with the existing tables. Changing these tables would cause a need to change the applications that feed the databases.

3.5 Code

In this section, we have exposed the theoretical bases on which a good reengineering can be done and we have applied this theses in order to redesign the databases from the firm CARSID. We have worked with the databases from one of its plants, '*La Cokerie du Marchienne*', in order to develop a consistent and efficient database model which allowed the plant to improve both its information system and the subsequent information analyses. We have based the new model in a concept – the material – which we have considered a very useful link between the existing databases because of its impact on the production process and on its improvement possibilities regarding efficiency and costs. In respect to the possibilities of improving the firm's information analysis, the next chapter will focus on methods and proposals aimed at improving the results.

4. DATA WAREHOUSE AND INFORMATION ANALYSIS

4.1 Introduction

Nowadays, most enterprises have some kind of information system. These information systems include a great quantity of data which can be analyzed. The people responsible for those firms' sections and departments need to study their own behaviour, his suppliers', the way their sales fluctuate, among many other things. It is for this reason that the data warehouse technology appeared

and gained the popularity it has to this date in the business sector.

What is Data Warehousing? The term *data warehouse* was coined by W. H. Inmon. The concept of data warehouse is based on the idea of grouping all the necessary data for applying multiple systems of analysis and obtaining the maximum information. This information could be crucial for our business in terms of competitive power, in order to better understand customers' necessities and preferences, or market fluctuations, among many other data.

Typically, a data warehouse is housed on an enterprise mainframe server. Data from various online transaction processing (OLTP) applications and other sources is selectively extracted and organized on the data warehouse database for use by analytical applications and user queries. Data warehousing emphasizes the capture of data from diverse sources for useful analysis and access, but does not generally start from the point of view of the end user or knowledge worker who may need access to specialized, sometimes local databases. The latter idea is known as the data mart.

What are Data Marts? A data mart is a repository of data gathered from operational data and other sources that is designed to serve a particular community of knowledge workers. In scope, the data may derive from an enterprise-wide database or data warehouse or be more specialized. The emphasis of a data mart is on meeting the specific demands of a particular group of knowledge users in terms of analysis, content, presentation, and ease-of-use. Users of a data mart can expect to have data presented in terms that are familiar.

In practice, the terms *data mart* and *data warehouse* each tend to imply the presence of the other in some form. However, most writers using the term seem to agree that the design of a data mart tends to start from an analysis of user needs and that a data warehouse tends to start from an analysis of what data already exists and how it can be collected in such a way that the data can later be used. A data warehouse is a central aggregation of data (which can be distributed physically); a data mart is a data repository that may derive from a data warehouse or not and that emphasizes ease of access and usability for a particular designed purpose. In general, a data warehouse tends to be a strategic but somewhat unfinished concept; a data mart tends to be tactical and aimed at meeting an immediate need.

4.2 Data Warehouse Design

Designing software may follow different steps and methodologies according to different factors, but anyway there is a great vantage of possibilities available for one particular design. It is therefore difficult to explain the designing process by reference to one single methodology. In one follows, we will introduce two different methodologies which can be followed in designing the software for our case study.

SAS *Rapid Warehouseing Methodology*. The more general of these two methodologies was proposed by the SAS Institute (Criado & Sánchez 2002). It is called *SAS Rapid Warehouseing Methodology*. This methodology is iterative and adequate for incremental project development of data warehouse. It may be divided into five main phases (Fig.).

1. Aim definition: In this phase the workgroup is defined. It will be made up of the staff at the Information Technologies (I.T.) department, representative people from the end user profile, the person in charge of the departments which want the data warehouse to be developed and the project manager.
In this phase the workgroup will define the functions and domain of the data warehouse. Also, they will define the parameters through which the success of the project will be assessed.

2. Requirement definition: This phase is critical because defining the requirements we are also defining the essential mechanisms which will allow us to control and analyse the business we are working with. If this definition fails to be correct, the aims defined in the first phase are at stake.

Generally, meetings will be held in this phase where the people form the I.T. department, the people representing the end user, and the responsible for the department where the data warehouse will operate will discuss about several aspects and details. The I.T. staff will study the existing information systems, and they will detect present deficiencies and opportunities, as well as future possibilities.

The result of this phase is a document which sets out the information necessities of the business, as well as the source from where the data can be obtained, and the design of the database architecture in relation to the data warehouse.

3. Designing and Modelling: The requirements identified in the previous phase will serve as the basis for designing and modelling the data warehouse. In this phase the data sources will be identified (internal, external), and the necessary modifications of the databases will be sketched in order to obtain a logical model for the data warehouse. This model will be composed by the entities and relationships which will allow for the business necessities of the organization to be solved. The logical model will further be developed into the physical model which will be stored in the data warehouse. This physical model will also be used in defining the storing architecture of the data warehouse, taking into account the kind of data exploitation we want to conduct with the data.

4. Implementation: The implementation of a data warehouse could be carried out following the next steps:

- Extracting the data from the operational systems and transforming them.
- Charging the validated data in a data warehouse. This action will be planned according to the refreshment necessities identified in the requirement definition and designing phases.
- Exploiting the data warehouse. This can be carried out through a multiplicity of techniques. The techniques adopted will vary according to the necessities of the business. The usual techniques are:
  - Query & Reporting
  - On-line analytical processing (OLAP)
  - Executive Information Systems (EIS)
  - Decision Support Systems (DSS)
  - Information visualizing
  - Data Mining

This phase will end up with a data warehouse capable of being used both by end users and the I.T. department.

5. Reviewing: The developing of a data warehouse does not come to an end when the implementation is carried out. On the contrary, it is an iterative process which increases its efficiency and scope with the lessons from experience.

Six to nine months after the implementation of the data warehouse, a revision of its functions should be carried out in order to identify new aspects to be improved or existing aspects to be further developed according to the actual use of the system.

Ralph Kimball's designing process. Another designing process is defined by Ralph Kimball in the book *The data warehouse toolkit* (1996). It can be structured in the following steps.

1. Choosing a business process to model. A business process is a major operational process in the organization that is supported by some kind of legacy system from which data can be collected for the purposes of the data warehouse. Examples of business processes are orders, invoices, shipments, inventory, account administration, sales, manufacturing process, and general ledger.

2. Choosing the grain of the business process. The grain is, according to Kimball, the fundamental atomic level of data to be represented in the fact table for this process. Typical grains are individual transactions, individual daily snapshots, or individual monthly snapshots. It is impossible to proceed to step 3 without defining the grain.

3. Choosing the dimensions that will apply to each fact table record. Typical dimensions are time, product, costumer, promotion, warehouse, transaction type, and status. With the choice of each dimension, all discrete, textlike dimensional attributes that fill out each dimension table should be described.

4. Choosing the measured facts that will populate each fact table record. Typically measured facts are numeric additive quantities like money, quantity sold, etc.

4.3. Data Mining

Data mining is the extraction of data from any database aimed at identifying patterns and establishing relationships among them. The process is based on different statistical models and can be applied to huge volumes of data in order to find trends and patterns. The difference between Data Mining and statistics is that the techniques used in Data Mining allow for the models to be created automatically whereas the more "classical" statistic technique has to be applied by a professional statistician. As a consequence, data mining shows increased power for its easiness of use, which allows all companies which have databases with big volumes of data to carry out data exploitation with relatively no difficulties.

Data Mining allows to automatically generate a model for data analysis with less manual work and the possibility of evaluating big quantity of data without any help from a professional statistician. There is a great deal of models which are generated

automatically, therefore increasing the probability to find a good model. Moreover, the analyst needs less formation on model construction and also less experience.

Data mining techniques are used in mathematics, cybernetics, genetics, and a multiplicity of businesses. A leading trend is to use data mining through web sites. This is called *web mining*, is used in customer relation management (CRM), and takes advantage from the huge amount of information gathered by a web site to look for patterns in user behaviour.

The KDD process. KDD (Knowledge Discovery in Database) is the not trivial process of identifying new, valid, potentially useful and definitely interpretable patterns in a data set. The knowledge discovery in databases is a process which takes advantage from several information technologies for extracting new and useful knowledge from great quantities of data through automatic or semiautomatic processes.

Data Mining is the central phase of the KDD process. Data Mining is the phase which integrates learning and statistical methods in obtaining pattern and model hypothesis.

The KDD process consists of 8 stages, all of them with a very important implication of the user (Fig 17).

1. Defining the application domain. Once the user has defined his or her work, it is necessary to develop an application with its respective database. With this database and the previous knowledge of the user, we will define the domain of our application.

2. Selecting the data that will be analyzed. In this phase, the data sources which can be needed for abstracting the information defined in the domain will be established. Also, a data warehouse scheme will be designed in order to unify the whole collected data.

3. Cleaning of data & pre-processing. With the data, it will always appear some data which would not be sensibly used for several reasons (for example, because one particular datum is an outlier). These data are called *knots*. In this step the goal is to remove all knots which can be present in the data of the source databases. This cleaning step will allow us to avoid any deviations or any other kind of problems in the models.

4. Transformation: Reduction of the data and projection. We need to find a useful characteristic which represents the dependent data, according to the user's aims.

5. Selecting the model to use. In this phase, the

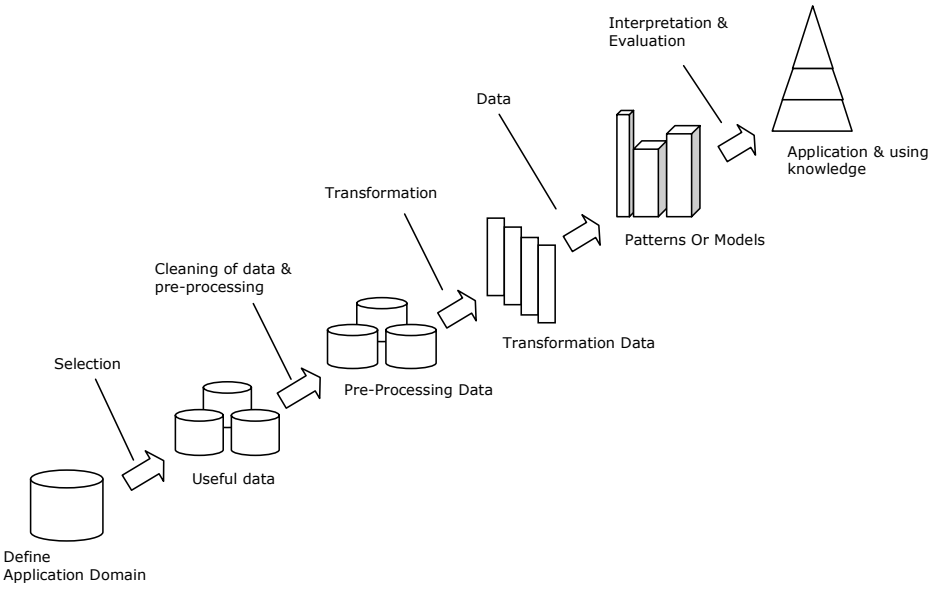


Fig. 17: The KDD Process

finding task should be selected and it should be determined whether the process aim is classifying, grouping, doing a regression, etc.

6. Choosing the correct algorithm for the data mining. The most suitable model or models should be chosen according to our aims.

7. Interpreting the resulting patterns. By interpretation we understand the verification and observation of the results. After interpreting the results, it may be necessary to go back to the previous point and change the model, in order to apply a new model to other data or to process the data again.

8. Consolidating the discovered knowledge. Here we will produce the results and the conclusions as help in taking decisions, solving problems, or defining new strategies.

4.4. What can a DWH do for CARSID?

Developing a Stock Control (Working with the OLAP tools). If we want to improve CARSID's analysing system, it will be useful to assess existing technologies and the latest methodologies. This is what we will try to carry out in the present section.

In order to take the maximum profit from data analysis, there is a very useful method called KDD. This methodology is developed through a number of steps which allow us to get a deep insight of the data analysis with greater success possibilities. The analysis is based in the use of a set of analysis technologies, with data warehouse and data mining as the main tools.

When we started studying the possibilities for analyzing the data with the SQL Server 7, we discovered that the version 2000 of this product includes substantial improvements for this kind of studies. The Microsoft SQL Server Enterprise Edition creates a new product called Microsoft Analysis Services. The Microsoft Analysis Services include all the possibilities for working with data warehouse that we could already find in version 7 and presents the Mining Model. The mining models are mechanisms for applying the algorithms to relational databases and multidimensional databases.

In what follows, we will present a group of examples of cubes in order to show the possibilities that they offer for CARSID's data warehouse. These cubes are not the only thing that is needed in this case, nor are they the solution they are looking for. They are just some examples of the possibilities that a data warehouse offers. The data from the examples are not real, but they have been introduced to illustrate in a more realistic way the examples that are presented.

First example:

This cube allows us to cross the information in function of the situation of coal in the production process, the type of coal, and its origin. In figure 18 we can see the scheme we needed in creating the cube. In this cube, the measure is the weight of the coal ('Poids' in the table called 'SubMvtCharbon') and the dimensions are (*a*) the places which can be found in the table 'PlaceOfCharbon', and (*b*) the origin of the coal, the type of the coal, and the movement of the entry material, as defined in the tables 'Pays', 'Charbon', and 'MvtCharbon' respectively.

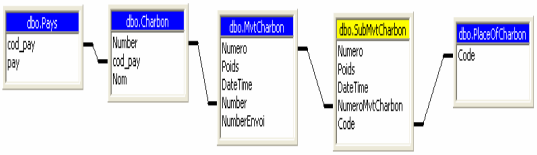


Fig. 18: Coal Movement Cube. Scheme

The result of calculating the cube can be appreciated in figure 19. The origin country is placed in the rows and the columns have the places where the coal can be.

	Code		
+ Pay	- Todas PlaceOfCharbon	Silos	Terre
- Todas PaysCharbon	7.000.000	1.000.000	6.000.000
+ Austrelia	3.000.000		3.000.000
+ Canada			
+ United States of Americ	4.000.000	1.000.000	3.000.000

Fig. 19: Coal Movement Cube. General Results

In figure 20 we can see the result of applying the *drill-down* operation. The advantage of this new

view is that now we can observe the type of coal and the quantities present in every place.

		Code		
- Pay	+ Nom	- Todas PlaceOfCharbon	Silos	Terre
- Todas PaysCharbon	Todas PaysCharbon Total	7.000.000	1.000.000	6.000.000
+ Australia	Australia Total	3.000.000		3.000.000
+ Canada	Canada Total			
- United States of America	United States of America T	4.000.000	1.000.000	3.000.000
	+ Blue Creek	4.000.000	1.000.000	3.000.000

Fig. 20: Coal Movement Cube: Specific Results

Second example:

This second cube gives us the possibility to know the information in relation to the consumption of coal. More specifically, we can cross the origin of the coal and the type of coal with the temporary moment in which the coal entered or left the production process. In figure 21 we can see the tables 'Pays' and 'Charbon' for the dimension country of origin and type of coal. The temporal dimension can be found in the fact table, and therefore we do not need an extra table for the temporal dimension. Also in the fact table, we can see the field 'Poids', which is the measure.

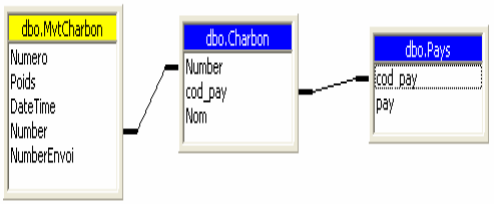


Fig. 21: Temporal Coal Movement Cube. Scheme

The general results in this case can be found by crossing the origin of coal and the years. The results may be appreciated in figure 22.

	+ Año		
+ Pay	Todas MvtCharbonTiempc	+ 2001	+ 2002
- Todas PaysCharbon	14.000.000	1.500.000	12.500.000
+ Australia	2.000.000		2.000.000
+ Canada	8.000.000	500.000	7.500.000
+ United States of Americ	4.000.000	1.000.000	3.000.000

Fig. 22: Temporal Coal Movement Cube. General Results

In figure 23 we can see the possibilities that are available through the operation *drill down*. This cube let us know the quantity of one type of coal for one specific day. And this information is available for all the days, months, terms, or years. Thus, controlling the stock of coal which comes in or goes out is really easy.

		- Año	- Trimestre	- Mes	Dia	
			- 2001			
		2001 Total		- Trimestre 3		
			Trimestre 3 Total	- Agosto		
- Pay	+ Nom			Agosto Total	1	10
- Todas PaysCharbon	Todas PaysCharbon Total	1.500.000	1.500.000	1.500.000	1.000.000	500.000
+ Australia	Australia Total					
+ Canada	Canada Total	500.000	500.000	500.000		500.000
- United States of America	United States of America T	1.000.000	1.000.000	1.000.000	1.000.000	
	+ Blue Creek	1.000.000	1.000.000	1.000.000	1.000.000	

Fig. 23: Temporal Coal Movement Cube. Specific Results

Third example:

In order to analyze the quality of the coke, it is interesting to have some mechanism for establishing some kind of control. We suggest this cube because it let us see the quantity of coke and its granularity. In figure 24 we can see the scheme of the cube. We have the table 'Granulometrie' for the dimension granulometry, and the table 'MvtCoke' for establishing the temporal dimension; the fact table is 'Gran Coke'.

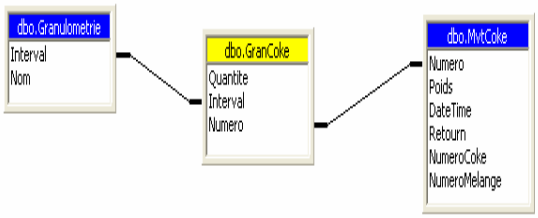


Fig. 24: The Coke Granularity. Scheme

The general result (Fig 25) of this cube is the crossing of the year and the granularity. This cube lets us know the quantity of coke which is produced by years distributed by diameter intervals.

The *drill-down* operation lets us see the quantity of coke which is produced by day, month, term or year (Fig. 26). This specificity lets us find the deviations of quality and the term or month when they occurred.

	Interval				
+ Año	- Todas Granulometrie	>80	0-20	20-40	40-80
- Todas TimeMvtCoke	4.649.703	299.703	850.000	3.100.000	400.000
+ 2002	3.099.802	199.802	600.000	2.000.000	300.000
+ 2003	1.549.901	99.901	250.000	1.100.000	100.000

Fig. 25: The Coke Granularity. General Results

				Interval				
- Año	- Trimestre	- Mes	Día	- Todas Granulometrie	>80	0-20	20-40	40-80
- Todas TimeMvtCoke	Todas TimeMvtCoke Total			4.649.703	299.703	850.000	3.100.000	400.000
- 2002	2002 Total			3.099.802	199.802	600.000	2.000.000	300.000
	- Trimestre 3	Trimestre 3 Total		3.099.802	199.802	600.000	2.000.000	300.000
		- Septiembre	Septiembre Total	3.099.802	199.802	600.000	2.000.000	300.000
			15	3.099.802	199.802	600.000	2.000.000	300.000
+ 2003	2003 Total			1.549.901	99.901	250.000	1.100.000	100.000

Fig. 26: The Coke Granularity. Specific Results

4.5. Code

In this section we have exposed what data warehouse and data mining technologies are. Also, we have tried to point out the advantages that may be derived from applying these technologies to the data collected with CARSID's information system, and, in so doing, we have tried to justify the convenience of actually applying some kind of data exploitation system in order to obtain useful information from these data.

5. CONCLUSIONS AND FURTHER IMPROVEMENTS

Conclusions. The aim of this project was to redesign the data models from the Marchienne plant of CARSID and to integrate all of them in one single data model, while improving its design and showing the possibilities offered by OLAP technologies in the bettering of the data exploitation carried out in this plant. To accomplish our objective we have carried out a project presented here in 5 sections. From these, section 3 and 4 should be remarked as they constitute the core part of this work. After the introduction in which the reader was presented with the intention and context of this project and the structure of this work, section 2 introduced the company for which the project was developed, its structure and that of the plant where the project was to be implemented, together with the processes carried out. Section 3 commented on the database reengineering processes conducted. Finally, section 4 showed through examples the possibilities posed by OLAP technologies in exploiting the data produced in the plant.

Before focusing on the results of this project, we will explain in more detail the core sections, which are, as said, sections 3 and 4. The first of these sections exposed how the databases were improved. We explained how we applied database reengineering techniques in order to improve three databases which were being used in the coke plant. In this process, we opted for a compromise solution which allowed us to go on working with the application already in use in this plant, and at the same time to enhance the performance of these databases according to the ideal objectives of a good database design. This section further remarked the nature of the solution adopted in the unification of the three databases, since it constitutes a mechanism through which one of the main factors involved in production cost can be controlled, namely *stock control*. This mechanism will allow for a reduction of cost in the final product as long as it is conveniently used and, as a consequence, the company may become more competitive.

This increased competitiveness may be notably fostered through a better exploitation of data, obtained through the implementation of OLAP technologies and production process controlling systems, which are suggested and explained in section 4. These mechanisms would help improve the analysis and study of those data gathered by the information system. In order to facilitate the understanding of this proposal and show the applicability of the possibilities offered by these technologies, some examples applied to the case under study are pointed out.

In what follows, we will focus on the conclusions derived from this project which may be structured in three levels – technological, research and economic contexts. It should be remembered, however, in reading these conclusions, that the solution finally adopted was a compromise solution which forced us to build the implementation of our aims both on a good application of the theoretical bases of information systems and on a *usable* solution which best suits the actual necessities of the company, among which *cost* has been a key factor.

The main result which is derived from this project is the practical application of database engineering and reengineering methods and techniques in the modification of the data models already in use in the enterprise. The new, integrated model presented will help not only enhance its database efficiency and efficacy and reduce the space devoted to storing information, but it will also offer the possibility of conducting queries about data which were previously not related. Of utmost importance is the fact that this new model is already prepared for the application of OLAP technologies in the exploitation of information.

Moreover, we suggested a mechanism for enriching stock control. This system will allow the plant to control one of the key factors involved in final product cost rise or decline. Therefore, this system, in combination with a good stock control policy, will help reduce final product cost and, as a result, will enhance the company's situation within the market.

Also, we presented OLAP technologies in an attempt to let the company figure out what they may imply for the business's performance. After examining the theoretical background of these technologies, we reached the conclusion that the company may benefit highly from their application in interpreting those data related to their performance.

In a more general level, we have found interesting to remark the following observations:

Database reengineering
A conclusion we may reach after doing the present work is that even though the final aim of any database reengineering process is taking this database and modifying it until a data model is obtained which perfectly suits the company's necessities, in the case studied – and probably in most cases – this will not be possible, since the model will depend on a set of applications or elements which will force a compromise solution which betters the current situation even without becoming an ideal solution. Thus, the complexities of the real world impose a necessary separation between theory and practice.

A stock controlling system
We have also proposed an extension of the data model by including a system to collect the data related to material movements so as to have at the plant's disposal a stock controlling system. This system, as a concept, is something which we considered very interesting because what may decide the cost of a product at the end of the production process is the material used and/or lost, the investment dedicated to material acquisition, the planning of purchases so as to avoid breaking stock, etc. Although this is not a concept which we intend to apply as a universal system for any company from this industrial sector, a stock controlling system becomes essential nowadays for any enterprise in the industrial sector.

OLAP tools
The tools presented in section 4 are, in our opinion, fundamental tools so that any company with a good information system may exploit their data exhaustively. Through these technologies, it will be able not only to improve its knowledge of the company and have the information which may help in decision taking processes, but also studies on those data will be quickened and costs reduced.

Further improvements. We would like to leave an open door for future works which continue enhancing this company's performance in an ongoing process. In this future works we may identify two major lines. The first one is to be carried out on a short-term basis, but there is still another one for middle and long-term projects.

In the first place, from a short-term perspective, we would suggest that after applying the new database redesign, data warehouse, especially data mining, implementation would be worked in depth. This may help the company both improve product manufacturing and reduce costs. This work should depart from the basis set out in the frame of this project.

In the second place, middle or long-term projects may tackle requirement abstraction (using UML as a standard language) in combination with process reengineering policies. In carrying out these requirement abstraction and process reengineering, we would also insist on the necessity of working with quality engineering policies or with any similar policies. Once developed such a costly task, it would be advisable in our opinion to initiate the construction of a new information system which accounts for the whole working process carried out in the plant under study. It would also be highly advisable to implement a common policy in the long term in which the whole company would integrate all of its information systems.

REFERENCES

Arnold, Robert S., cop., 1993, *Software Reengineering*, Los Alamitos, Clif. IEEE Computer Society Press.

Bain, Tony; Mike Benkovich; Robin Dewson; Sam Ferguson; Christopher Graves; Terrence Joubert; Denny Lee; Robert Skoglund; Paul Turley; Sakhr Youness; Mark Scott, 2001, *SQL Server 2000 Data Warehouseing with Analysis Services*, Wrox Press Ltd. (Programmer to Programmer).

Deming, W. E., 1972, *Report to Management*, Quality Progress.

A. Donnay, F. Fouss, M. Kolp, D. Massart, *Analyse orienté objet de processus sidérurgiques de type cokier,* Working Paper IAG, Université Catholique de Louvain, 2003.

Dolan, Alan & Joan Aldous, 1993, *Networks and Algorithms (An Introductory Approach)*, John WIELY & Sons.

Dougerty, Edward R. & Charles R. Giardina, 1998, *Mathematical methods for artificial and autonomous systems*, Prentice-hall International Editions.

Goldratt, Eliyahu M. & Jeff Cox, 1984, *The Goal: A process of ongoing improvement*, The North River Press.

Gunderloy, Mike & Tim Sneath, 1994, *SQL Server Developer's Guide. to OLAP with Analysis Services*, SYBEX.

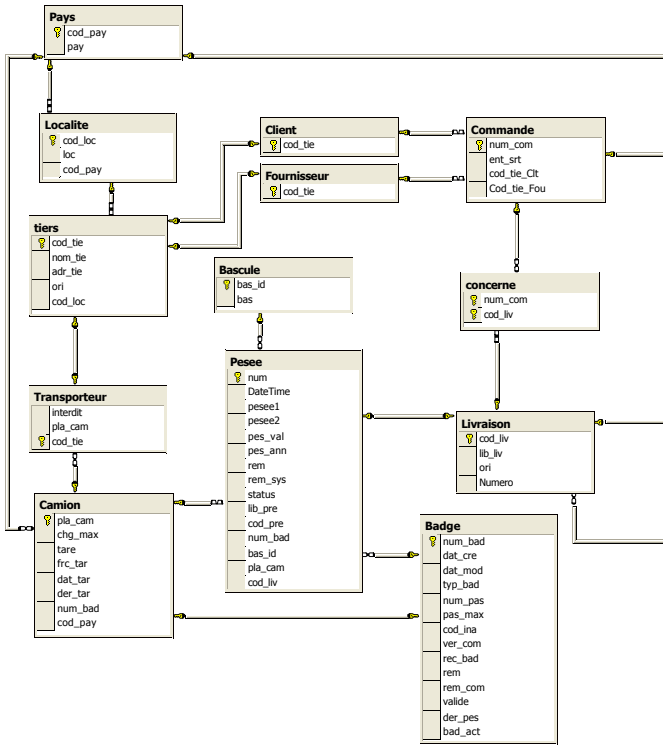
Hall, Patrick A. V. & Geoff R. Dowling, 1998, "Approximate String Matching", *ACM Computing Surveys*, vol. 12, no. 4 (1 December 1998), Prentice-hall International Editions.

Han, Jaiwei & Micheline Kamber, 2000, *Data Mining: Concepts and Techniques*, (online document) http://www.cs.su.ca/~han/dmbook, Morgan Kaufmann Publishers.

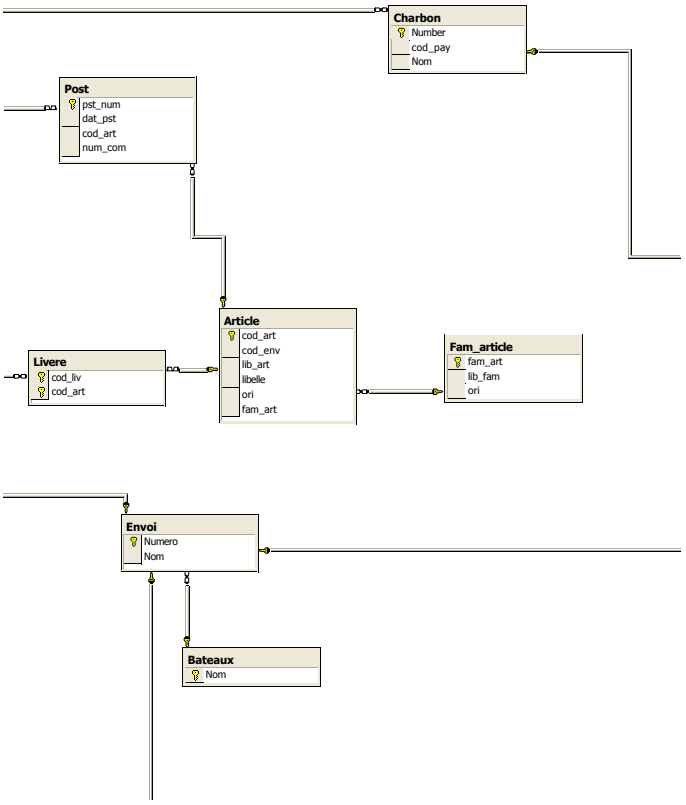
Inmon, W. H., 1996, *Building the Data Warehouse. (Second Edition)*, John Wiley & Sons, Inc.

Juran, Joseph M, 1992, *Juran on Quality by Design: The New Steps for Planning Quality into Goods and Services*, The Free Press.

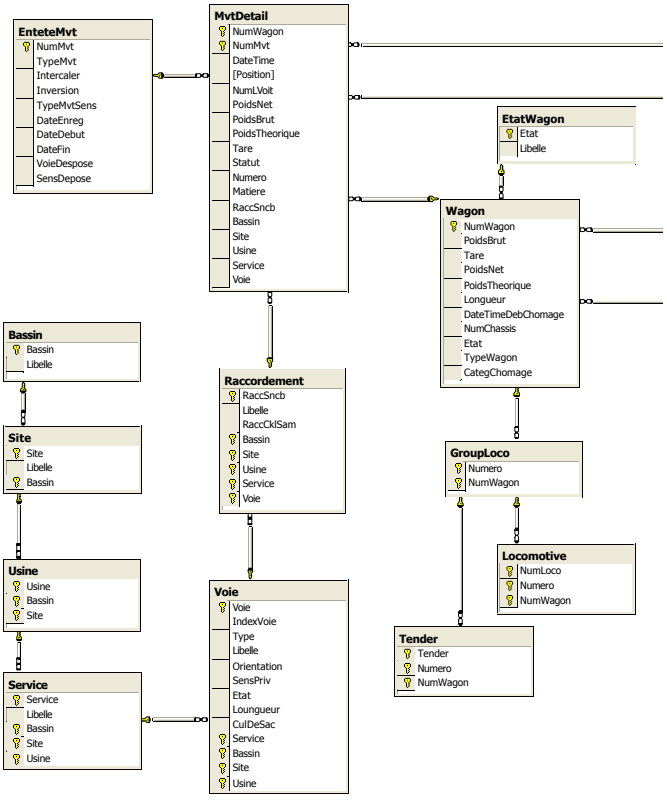
Kimball, Ralph, 1996, *The Data Warehouse Toolkit. Practical Techniques for building dimensional data warehouses*, John Wiley & Sons, Inc.


Pays
cod_pay
pay
Localite
cod_loc
loc
cod_pay
Client
cod_tie
Fournisseur
cod_tie
Commande
num_com
ent_srt
cod_tie_Clt
Cod_tie_Fou
tiers
cod_tie
nom_tie
adr_tie
ori
cod_loc
Bascule
bas_id
bas
concerne
num_com
cod_liv
Transporteur
interdit
pla_cam
cod_tie
Pesee
num
DateTime
pesee1
pesee2
pes_val
pes_ann
rem
rem_sys
status
lib_pre
cod_pre
num_bad
bas_id
pla_cam
cod_liv
Livraison
cod_liv
lib_liv
ori
Numero
Camion
pla_cam
chg_max
tare
frc_tar
dat_tar
der_tar
num_bad
cod_pay
Badge
num_bad
dat_cre
dat_mod
typ_bad
num_pas
pas_max
cod_ina
ver_com
rec_bad
rem
rem_com
valide
der_pes
bad_act

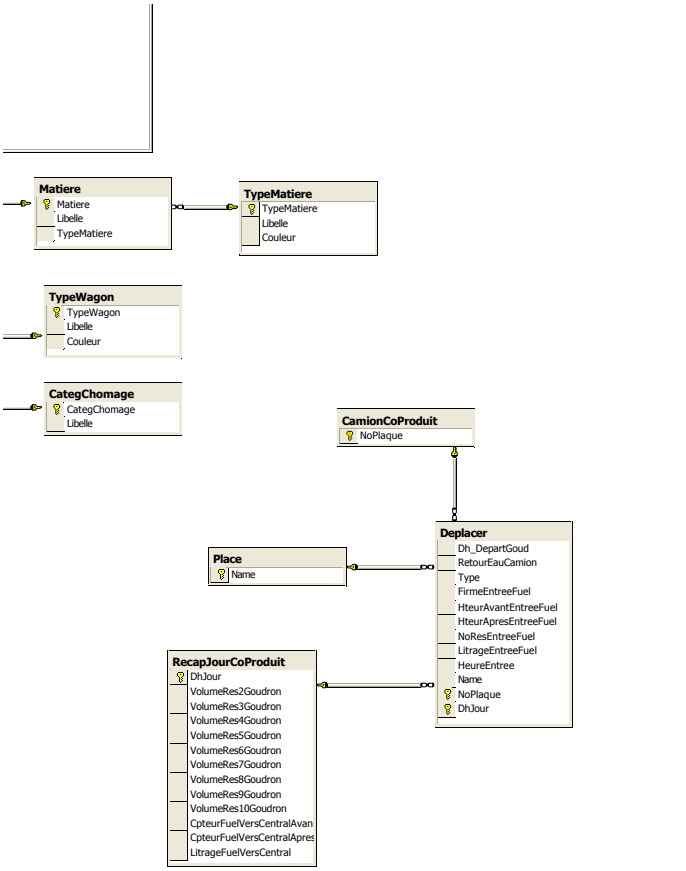
1-1


Charbon
Number
cod_pay
Nom
Post
pst_num
dat_pst
cod_art
num_com
Article
cod_art
cod_env
lib_art
libelle
ori
fam_art
Fam_article
fam_art
lib_fam
ori
Livere
cod_liv
cod_art
Envoi
Numero
Nom
Bateaux
Nom

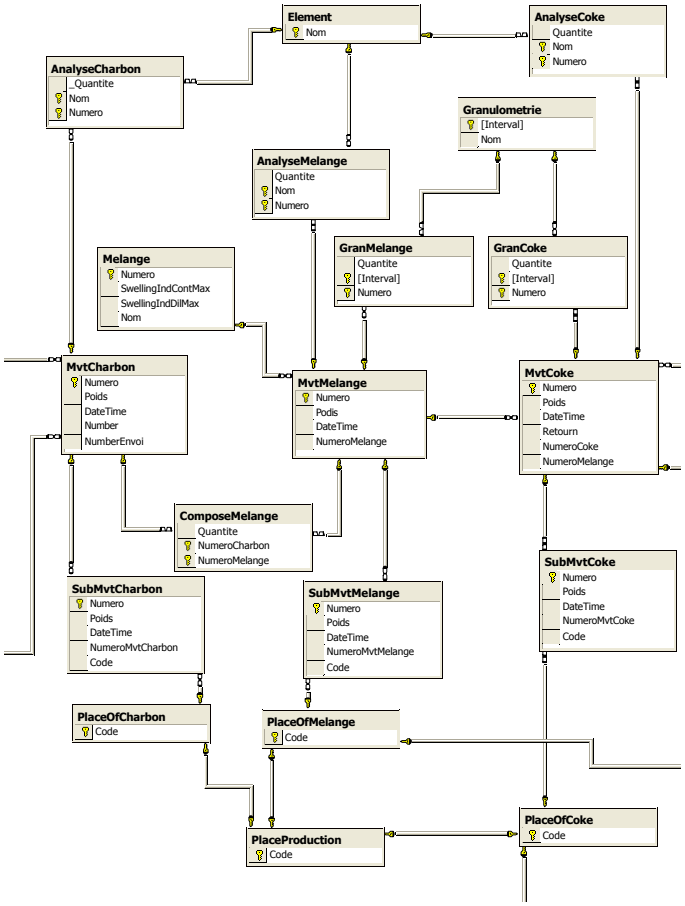
1-2


EnteteMvt
NumMvt
TypeMvt
Intercaler
Inversion
TypeMvtSens
DateEnreg
DateDebut
DateFin
VoieDespose
SensDepose
MvtDetail
NumWagon
NumMvt
DateTime
[Position]
NumLVoit
PoidsNet
PoidsBrut
PoidsTheorique
Tare
Statut
Numero
Matiere
RaccSncb
Bassin
Site
Usine
Service
Voie
EtatWagon
Etat
Libelle
Wagon
NumWagon
PoidsBrut
Tare
PoidsNet
PoidsTheorique
Longueur
DateTimeDebChomage
NumChassis
Etat
TypeWagon
CategChomage
Bassin
Bassin
Libelle
Site
Site
Libelle
Bassin
Usine
Usine
Bassin
Site
Service
Service
Libelle
Bassin
Site
Usine
Raccordement
RaccSncb
Libelle
RaccCklSam
Bassin
Site
Usine
Service
Voie
GroupLoco
Numero
NumWagon
Locomotive
NumLoco
Numero
NumWagon
Tender
Tender
Numero
NumWagon
Voie
Voie
IndexVoie
Type
Libelle
Orientation
SensPriv
Etat
Lounguеur
CulDeSac
Service
Bassin
Site
Usine

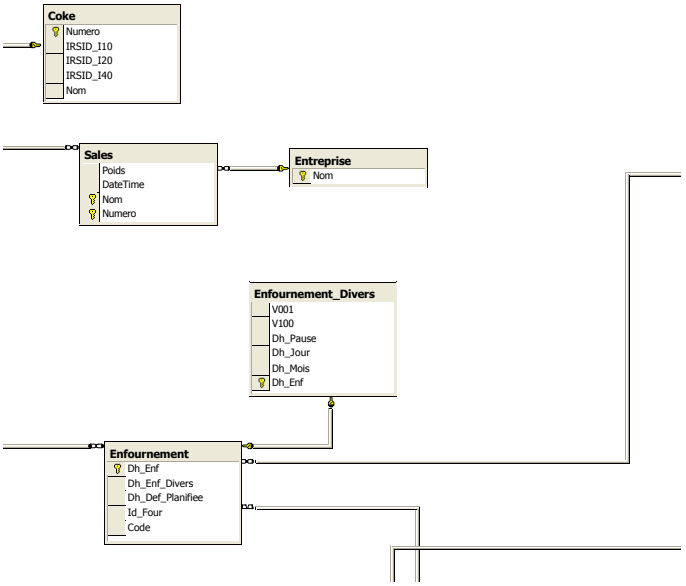
2-1


Matiere
Matiere
Libelle
TypeMatiere
TypeMatiere
TypeMatiere
Libelle
Couleur
TypeWagon
TypeWagon
Libelle
Couleur
CategChomage
CategChomage
Libelle
CamionCoProduit
NoPlaque
Deplacer
Dh_DepartGoud
RetourEauCamion
Type
FirmeEntreeFuel
HteurAvantEntreeFuel
HteurApresEntreeFuel
NoResEntreeFuel
LitrageEntreeFuel
HeureEntree
Name
NoPlaque
DhJour
Place
Name
RecapJourCoProduit
DhJour
VolumeRes2Goudron
VolumeRes3Goudron
VolumeRes4Goudron
VolumeRes5Goudron
VolumeRes6Goudron
VolumeRes7Goudron
VolumeRes8Goudron
VolumeRes9Goudron
VolumeRes10Goudron
CpteurFuelVersCentralAvan
CpteurFuelVersCentralApres
LitrageFuelVersCentral

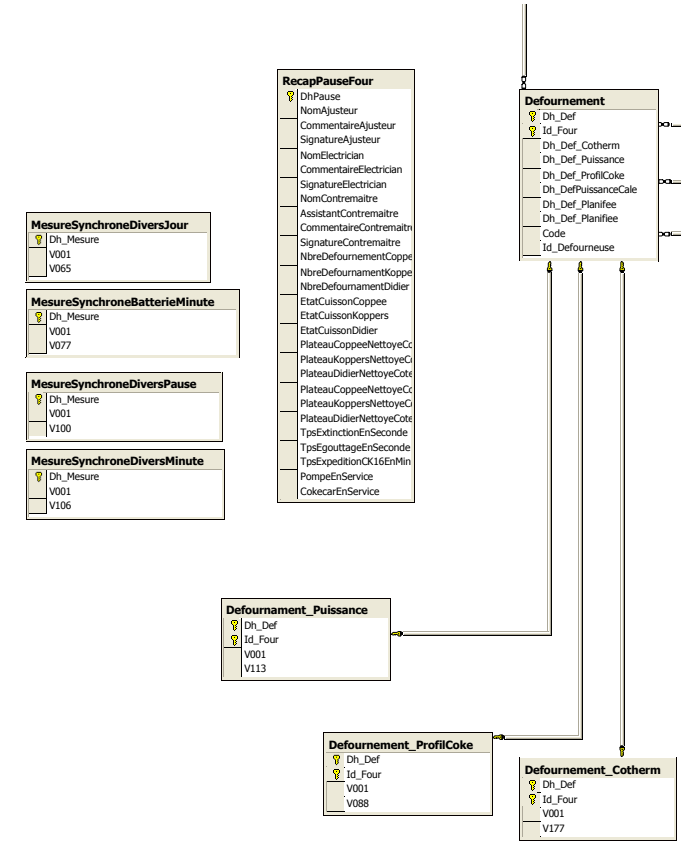
2-2


Element
Nom
AnalyseCoke
Quantite
Nom
Numero
AnalyseCharbon
Quantite
Nom
Numero
Granulometrie
[Interval]
Nom
AnalyseMelange
Quantite
Nom
Numero
GranMelange
Quantite
[Interval]
Numero
GranCoke
Quantite
[Interval]
Numero
Melange
Numero
SwellingIndContMax
SwellingIndDilMax
Nom
MvtCharbon
Numero
Poids
DateTime
Number
NumberEnvoi
MvtMelange
Numero
Podis
DateTime
NumeroMelange
MvtCoke
Numero
Poids
DateTime
Retourn
NumeroCoke
NumeroMelange
ComposeMelange
Quantite
NumeroCharbon
NumeroMelange
SubMvtCoke
Numero
Poids
DateTime
NumeroMvtCoke
Code
SubMvtCharbon
Numero
Poids
DateTime
NumeroMvtCharbon
Code
SubMvtMelange
Numero
Poids
DateTime
NumeroMvtMelange
Code
PlaceOfCharbon
Code
PlaceOfMelange
Code
PlaceProduction
Code
PlaceOfCoke
Code

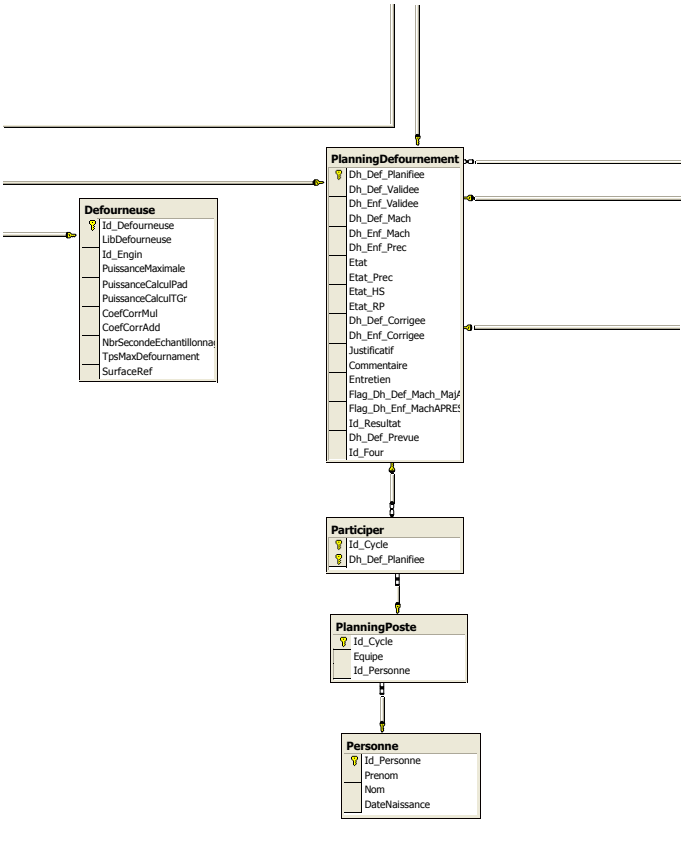
1-3


Coke
Numero
IRSID_I10
IRSID_I20
IRSID_I40
Nom
Sales
Poids
DateTime
Nom
Numero
Entreprise
Nom
Enfournement_Divers
V001
V100
Dh_Pause
Dh_Jour
Dh_Mois
Dh_Enf
Enfournement
Dh_Enf
Dh_Enf_Divers
Dh_Def_Planifiee
Id_Four
Code

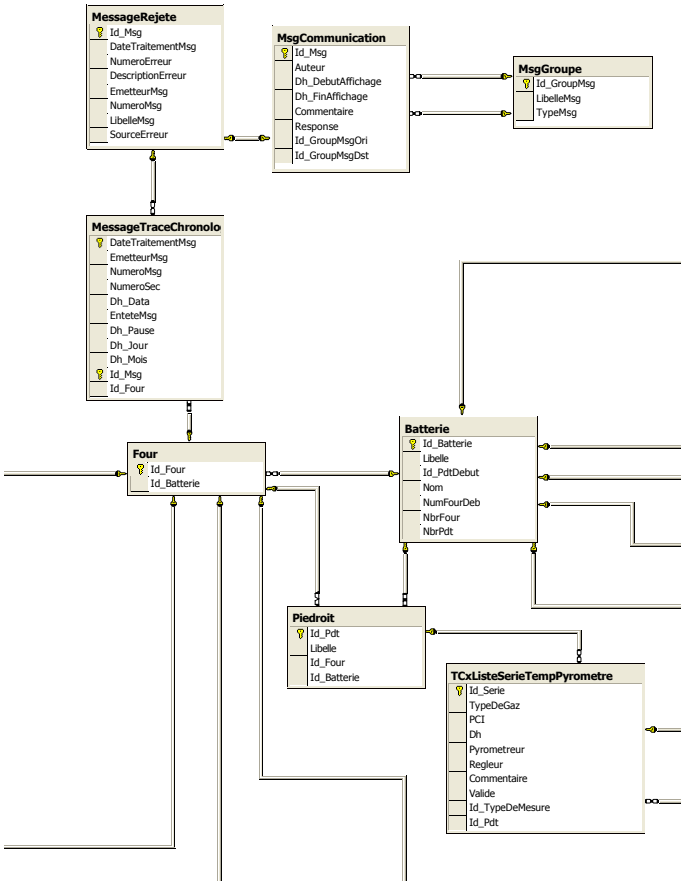
1-4


RecapPauseFour
DhPause
NomAjusteur
CommentaireAjusteur
SignatureAjusteur
NomElectrician
CommentaireElectrician
SignatureElectrician
NomContremaitre
AssistantContremaitre
SignatureContremaitre
EtatCuissonCoppee
EtatCuissonKoppers
EtatCuissonDidier
PompeEnService
CokecarEnService
Defournement
Dh_Def
Id_Four
Dh_Def_Cotherm
Dh_Def_Puissance
Dh_Def_ProfilCoke
Dh_DefPuissanceCale
Dh_Def_Planifee
Dh_Def_Planifiee
Code
Id_Defourneuse
MesureSynchroneDiversJour
Dh_Mesure
V001
V065
MesureSynchroneBatterieMinute
Dh_Mesure
V001
V077
MesureSynchroneDiversPause
Dh_Mesure
V001
V100
MesureSynchroneDiversMinute
Dh_Mesure
V001
V106
Defournament_Puissance
Dh_Def
Id_Four
V001
V113
Defournement_ProfilCoke
Dh_Def
Id_Four
V001
V088
Defournement_Cotherm
Dh_Def
Id_Four
V001
V177

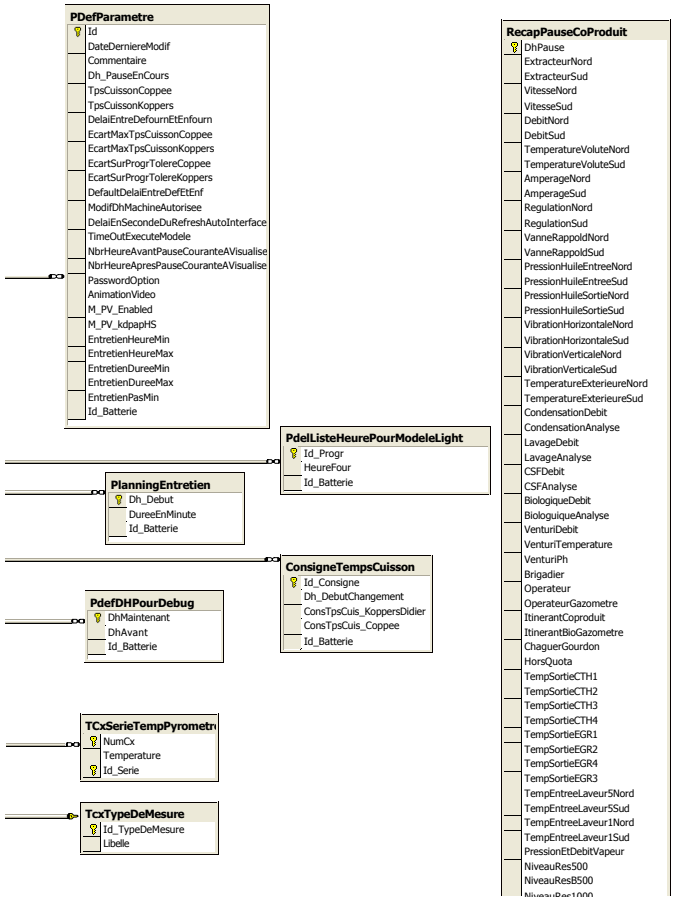
2-3


Defourneuse
Id_Defourneuse
LibDefourneuse
Id_Engin
PuissanceMaximale
PuissanceCalculPad
PuissanceCalculTGr
CoefCorrMul
CoefCorrAdd
TpsMaxDefournament
SurfaceRef
PlanningDefournement
Dh_Def_Planifiee
Dh_Def_Validee
Dh_Enf_Validee
Dh_Def_Mach
Dh_Enf_Mach
Dh_Enf_Prec
Etat
Etat_Prec
Etat_HS
Etat_RP
Dh_Def_Corrigee
Dh_Enf_Corrigee
Justificatif
Commentaire
Entretien
Id_Resultat
Dh_Def_Prevue
Id_Four
Participer
Id_Cycle
Dh_Def_Planifiee
PlanningPoste
Id_Cycle
Equipe
Id_Personne
Personne
Id_Personne
Prenom
Nom
DateNaissance

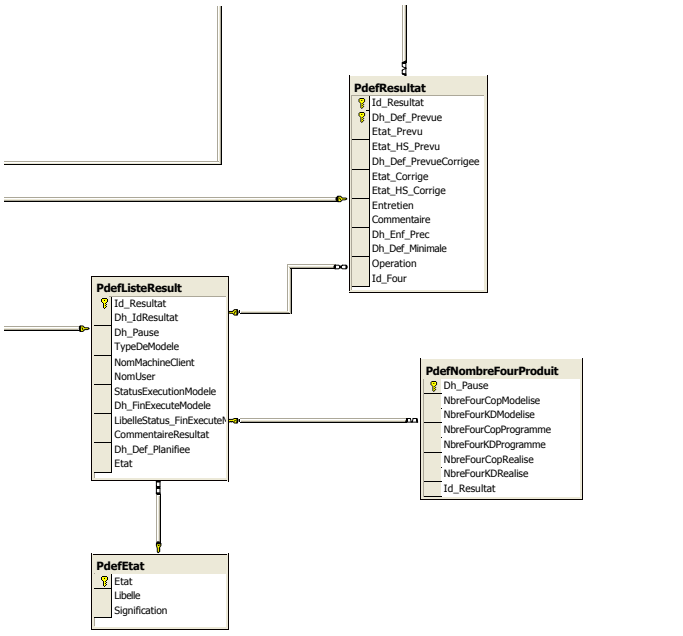
2-4


MessageRejete
Id_Msg
DateTraitementMsg
NumeroErreur
DescriptionErreur
EmetteurMsg
NumeroMsg
LibelleMsg
SourceErreur
MsgCommunication
Id_Msg
Auteur
Dh_DebutAffichage
Dh_FinAffichage
Commentaire
Response
Id_GroupMsgOri
Id_GroupMsgDst
MsgGroupe
Id_GroupMsg
LibelleMsg
TypeMsg
MessageTraceChronolo
DateTraitementMsg
EmetteurMsg
NumeroMsg
NumeroSec
Dh_Data
EnteteMsg
Dh_Pause
Dh_Jour
Dh_Mois
Id_Msg
Id_Four
Batterie
Id_Batterie
Libelle
Id_PdtDebut
Nom
NumFourDeb
NbrFour
NbrPdt
Four
Id_Four
Id_Batterie
Piedroit
Id_Pdt
Libelle
Id_Four
Id_Batterie
TCxListeSerieTempPyrometre
Id_Serie
TypeDeGaz
PCI
Dh
Pyrometreur
Regleur
Commentaire
Valide
Id_TypeDeMesure
Id_Pdt

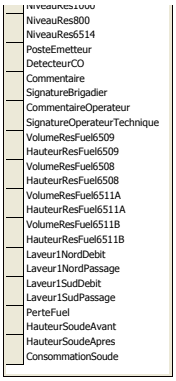
1-5


PDefParametre
Id
DateDerniereModif
Commentaire
Dh_PauseEnCours
TpsCuissonCoppee
TpsCuissonKoppers
DelaiEntreDefournEtEnfourn
EcartMaxTpsCuissonCoppee
EcartMaxTpsCuissonKoppers
EcartSurProgrTolereCoppee
EcartSurProgrTolereKoppers
DefaultDelaiEntreDefEtEnf
ModifDhMachineAutorisee
DelaiEnSecondeDuRefreshAutoInterface
TimeOutExecuteModele
NbrHeureAvantPauseCouranteAVisualise
NbrHeureApresPauseCouranteAVisualise
PasswordOption
AnimationVideo
M_PV_Enabled
M_PV_kdpapHS
EntretienHeureMin
EntretienHeureMax
EntretienDureeMin
EntretienDureeMax
EntretienPasMin
Id_Batterie
PdelListeHeurePourModeleLight
Id_Progr
HeureFour
Id_Batterie
PlanningEntretien
Dh_Debut
DureeEnMinute
Id_Batterie
ConsigneTempsCuisson
Id_Consigne
Dh_DebutChangement
ConsTpsCuis_KoppersDidier
ConsTpsCuis_Coppee
Id_Batterie
PdefDHPourDebug
DhMaintenant
DhAvant
Id_Batterie
TCxSerieTempPyrometr
NumCx
Temperature
Id_Serie
TcxTypeDeMesure
Id_TypeDeMesure
Libelle
RecapPauseCoProduit
DhPause
ExtracteurNord
ExtracteurSud
VitesseNord
VitesseSud
DebitNord
DebitSud
TemperatureVoluteNord
TemperatureVoluteSud
AmperageNord
AmperageSud
RegulationNord
RegulationSud
VanneRappoldNord
VanneRappoldSud
PressionHuileEntreeNord
PressionHuileEntreeSud
PressionHuileSortieNord
PressionHuileSortieSud
VibrationHorizontaleNord
VibrationHorizontaleSud
VibrationVerticaleNord
VibrationVerticaleSud
TemperatureExterieureNord
TemperatureExterieureSud
CondensationDebit
CondensationAnalyse
LavageDebit
LavageAnalyse
CSFDebit
CSFAnalyse
BiologiqueDebit
BiologuiqueAnalyse
VenturiDebit
VenturiTemperature
VenturiPh
Brigadier
Operateur
OperateurGazometre
ItinerantCoproduit
ItinerantBioGazometre
ChaguerGourdon
HorsQuota
TempSortieCTH1
TempSortieCTH2
TempSortieCTH3
TempSortieCTH4
TempSortieEGR1
TempSortieEGR2
TempSortieEGR4
TempSortieEGR3
TempEntreeLaveur5Nord
TempEntreeLaveur5Sud
TempEntreeLaveur1Nord
TempEntreeLaveur1Sud
PressionEtDebitVapeur
NiveauRes500
NiveauResB500

1-6


PdefResultat
Id_Resultat
Dh_Def_Prevue
Etat_Prevu
Etat_HS_Prevu
Dh_Def_PrevueCorrigee
Etat_Corrige
Etat_HS_Corrige
Entretien
Commentaire
Dh_Enf_Prec
Dh_Def_Minimale
Operation
Id_Four
PdefListeResult
Id_Resultat
Dh_IdResultat
Dh_Pause
TypeDeModele
NomMachineClient
NomUser
StatusExecutionModele
Dh_FinExecuteModele
LibelleStatus_FinExecuteM
CommentaireResultat
Dh_Def_Planifiee
Etat
PdefNombreFourProduit
Dh_Pause
NbreFourCopModelise
NbreFourKDModelise
NbreFourCopProgramme
NbreFourKDProgramme
NbreFourCopRealise
NbreFourKDRealise
Id_Resultat
PdefEtat
Etat
Libelle
Signification

2-5


NiveauRes800
NiveauRes6514
PosteEmetteur
DetecteurCO
Commentaire
SignatureBrigadier
CommentaireOperateur
SignatureOperateurTechnique
VolumeResFuel6509
HauteurResFuel6509
VolumeResFuel6508
HauteurResFuel6508
VolumeResFuel6511A
HauteurResFuel6511A
VolumeResFuel6511B
HauteurResFuel6511B
Laveur1NordDebit
Laveur1NordPassage
Laveur1SudDebit
Laveur1SudPassage
PerteFuel
HauteurSoudeAvant
HauteurSoudeApres
ConsommationSoude

2-6