Advanced Data Base

(8trd157)

Lab1 : *(project phase 1 of 4)*

Conceptual Data Model Design using E/R and Relational Methods

Paul Girard, Ph.D.

General objective of the database project *(4 phases)*

Model, design and implement a secure, interactive and multi-user system using 2 relational DBMS: Oracle 18c *(Unix)*, Access *(Windows)*. The development team will need 3 system analysts and one DBA *(Data Base Administrator)* for the design of data models *(conceptual and logical: global & all partial data models)* and the implementation of the complete system supporting a centralized and a distributed access using a client/server method.

First the Unix centralized database will use SQL, PL/SQL, C and PHP programming languages to access the database ; then the distributed database will be implemented with a Unix server and a Win/PC client having Access linked to an odbc interface to the server.

1 Objective of lab1: project phase 1 of 4

Create a global conceptual data model and 5 conceptual partial data models using 2 different types of data model: network *(Entity/Relationship)* and relational.

2 Description of the actual physical system

The general manager of an industry wants to improve the productivity of the actual manual system whose purpose is the control of the inventory of parts in order to satisfy the needs of the three departments involved :

2.1 In the **Maintenance Department** ~~technicians~~ go to the central store to get the necessary parts to repair some equipments. **Technicians** working for this department need to know the part number, the part name, the cost of one unit, the unit *(ex. gram, box, unit, kg, ...)*, the minimum quantity that should be in stock, the actual quantity in stock and the quantity on order of each part or any component of this part *(information are kept manually by a storekeeper).* There is no difference between a part and a component which is also a part. A part may or may not be a component of another part. For example an engine and a crankshaft are both a part but the crankshaft is a component of the engine and the engine is a component of a given car. Technicians would like to have a system that could display all the components *(part_number, part_name)* of a given part_number *(explosion of a part)* and all parts where a given part_number is a component *(implosion of a part).* Technicians are responsible to create or delete a component of a given part.

2.2 The **Purchasing Department** makes sure that parts are always available in the central store. If some parts under the responsibility of a purchasing agent have to be re-ordered, a ~~**Purchasing Agent** *(PA)*~~ will check different suppliers product *(price, delivery delay, unit, ...)* before preparing a purchase order and sending it to a given supplier.

All purchasing agents work under the responsability of the ~~**Purchasing Department Supervisor *(PDS)***~~ in the same department.

Each PA is responsible of a set of logically related parts *(ex. office parts, vehicles, network equipments, electrical parts, plumbing* parts, *...)*. Only the agent responsible for a given part may prepare a purchase order for that part. A part is assigned by the PDS to a PA *(no_part, part_name)* and the PA will specify at the same time the minimum quantity to be in stock according to the technicians. The unit price of a part will be determined by a simple calculation as soon as a purchase order is created. The last purchase order will always fix the same price for all parts already in the store.

Actually daily physical reports coming from the ~~**StoreKeepers *(SK)***~~ mention to each purchasing agent the parts that need to be re-ordered *(quantity in stock <= minimum quantity)*. A PA can then check different physical suppliers catalog to choose a product for a particular part. The following information is available for each ~~supplier:~~

- supplier_number, supplier_name, supplier_address, supplier_contact_name and from the physical catalog of his ~~product~~ : *[product_number, product_name, product_unit (integer being a multiple of a unit part) and supplier_ unit_price]*

The serial number of the supplier product *(unique for each supplier*) is different from the part number *(unique inside your company)*. Each supplier is independant and uses its own codification system.

The purchasing agent (PA) could call the contact person of a given supplier to check the delivery delay and make sure that the data read from their physical catalog is correct. He will then prepare a ~~**Purchase Order (PO)**~~ with the following information and send it to the chosen ~~supplier~~ :

> purchasing_agent_name, purchase_order_number, purchase_order_date, supplier_number, supplier_name, supplier_address, supplier_contact_name and for each ~~product~~ ordered *: product_number, product_name, product_unit, quantity_ordered, unit_price* and finally the total of the purchase_order.

The PA then modifies manually the inventory file with the storekeeper by updating the quantity_ordered and the revised price of the cost of each part unit. The last purchase order always fixes the price of the corresponding part based on the unit cost of the supplier product unit which is an integer being a multiple of the part_unit *(alpha-numeric) (see annex 2)*. This knowledge gives you the possibility of modifying easily the ordered quantity of part and its unit price by adding, dividing or multiplying these data by the product_unit of the supplier.

Note: No profit is added and no other *cost (no transportation cost, no custom fee, no broker cost and all taxes already included)* need to be considered to simplify all calculations.

A PA is responsible for creating a new part whose content will be accessible to all storekeepers and technicians. A PA also wants to access all previous purchase orders already sent to a given supplier in the past for a given part number. Before authorizing the payment of the supplier for a purchase order, the PA will wait to confirm that the shipping has been completed by storekeepers *(all products purchased in that purchase order has been received)*. A status "completed" will be written on a copy of this purchase order and sent to the account payable department. After the payment to the supplier, a status "paid" will be stamped on this copy *(this transaction is not considered in this lab).*

Finally the **Purchasing Department Supervisor (PDS)** has the following needs but some are difficult to support in the actual system :

- Hires a purchasing agent (emp_number, PA_name)
- Assign a given part under the responsability of a given PA
- Modify the responsability of a part from one PA to another PA.
- Display the number of purchase orders and the total value sent by a given PA *(employee_number, PA_name)* since a given date.
- Display all PA *(employee_number, PA_name)* with the total number of parts and the list of all parts *(part_number, part_name)* under the responsability of each one.

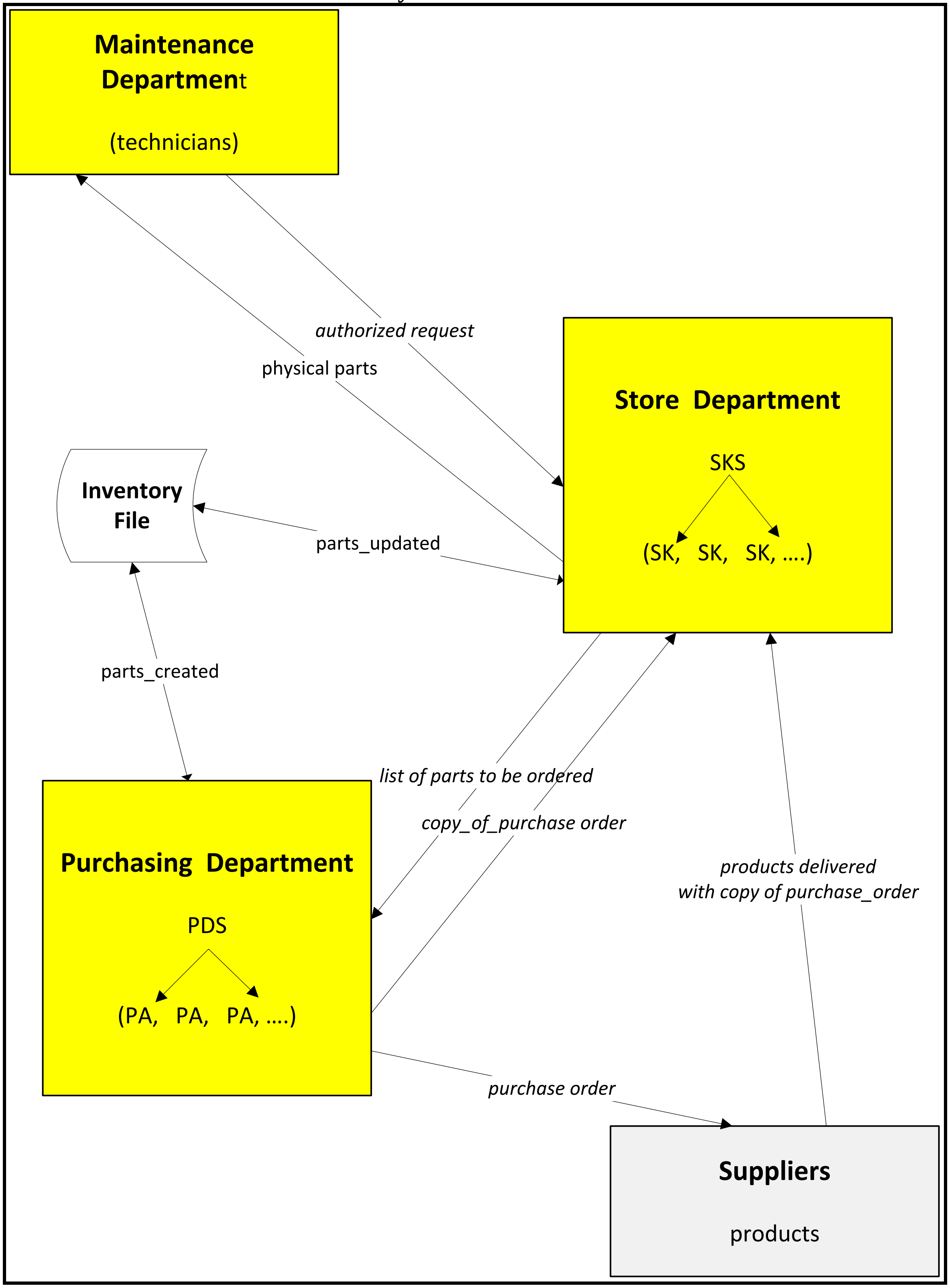
2.3 In the **Store Department** there are many ~~**Storekeepers (SK)**~~ under the responsability of the ~~**Storekeeper Supervisor (SKS)**~~**.** All SK's are responsible for the parts in the store. When a technician needs a new part, he checks with a StoreKeeper (SK) if the required quantity of this part *(part_number, quantity_stock)* is available in the store. Then the technician goes to the store and gives them to a SK an authorization request form specifying the part_number and the quantity of each part needed. The SK gives the requested parts to the technician and updates the physical inventory file.

Everyday, SK's prepare the lists of all parts that need to be re-ordered to each PA if the actual quantity of a part number in stock is less than the specified minimum. Each list is sent to the PA responsible for this part. Twice a year, SK's check for each part the actual quantity in stock compared with the value *(quantity_in_stock)* in the physical inventory file. They will then update the value in their inventory physical file and prepare a report mentioning the loss for the company and the parts involved.

When a supplier delivers his products according to a given purchase order, the SK is responsible to check and validate this delivery by the supplier. With a copy of the purchase order from the PA and the one from the driver, each product will be verified according to the required and the delivered quantity. The corresponding inventory file *(part_number, quantity_ordered, quantity_stock)* will then be updated accordingly to the quantity received and the supplier_product_unit . If the quantity received is more than the requested quantity, it will be returned to the supplier. The purchase order of the SK will be updated at the same time *(status, quantity_received)*. Many products may be ordered in one purchase order to the same supplier.

The ~~StoreKeeper **Supervisor (SKS)**~~ would like to know 1) the number of parts in the store having a different part number, 2) the value and the total number of all parts actually in the store including those having the same part numbers which is a very time consuming operation.

Actual Physical Environment



3 Description of the desired environment

Each computer system analyst works in one of those departments to implement application programs for their local users. Each one of these 3 analysts could use a different software/hardware environment *(ex. VB, PHP, Pro*C, VC, Windows, Linux, MacOS, Android, Access, Oracle, Ingres, MySQL, ...).* A DBA *(Data Base Administrator)* is responsible for the final design of the centralized and the distributed database, its implementation and optimization after analyzing the security, confidentiality and the type of access for end users and developers. The DBA must fill the needs of each system analyst and programmer in the company by creating the appropriate data model and its corresponding data base that will support all user transactions ; the DBA is also responsible of the data base security. Each system analyst is responsible of identifying all user transactions and building all appropriate data models *(E/R and relational)* to support these transactions. All ~~SK~~'s will use the same program, all ~~PA~~'s will use also their own program and all ~~technicians~~ will use their own program. The ~~PDS~~ and the ~~SKS~~ will have access to their own private program. All programs will be linked to a central database first. ~~PA~~'s will also have an internet access.

Each program may be called on any user station connected to the company's intranet but some transactions must be accessible to the user by an HTTP client like " Firefox" after being authenticated; for example, only a purchasing agent could prepare and send a purchase order directly from outside by using an internet access.

4 Methodology recommended for the project

- Download visio_demo.vsd *(2007 visio file)* and the transcation table *(Word table)* from the web site. It's better to use Windows 7 and Visio 2007 or 2010.

- For the first user type *(ex. PA type)* identify the transactions needed in the transaction table *(transaction_name, description)* and draw the partial E/R data model with the downloaded visio symbols *(entity, relation)* to support those transactions. Do not forget to name each entity, attribute and relation with a meaningful identification. Save this visio file under the name *"partial_data_model_user***"*. Complete the transaction table *(entity, key, access, frequency, ...)* and verify mentally if each transaction is able to be executed using your data model, then save it under the name *"transaction_user***"*. Copy this data model to another visio file called *"global conceptual E/R Data Model" (hybrid method for complex data model).*

- Do the same for another user type *(ex. technician).* Make another copy of the transaction table, fill it with the needed transactions for this user and create the corresponding partial data model file with the name *"partial_data_model_user***"* and the *"transaction_user***"*. Update your global conceptual data model to support all actual partial data models.

- Proceed the same for the 3 remaining user types, each one having their own transaction table and partial data model. The global conceptual data model will now be complete.

- **Before mapping the E/R data model to the relational data model, contact an assistant professor or the professor to validate your models, it will save you a lot of time.**

- Proceed with the mapping of the global conceptual data model to the relational form. Underline the appropriate keys (1 line == 1 table). This part may be done in lab2.

- Normalize the global conceptual relational data model *(one table, one line).* Check if the 3rd , 4th or 5th normal form would be better. This part may be done in lab2.

- Extract from this global relational data model the partial relational data model for each one of the 5 user types. This part may be done in lab2.

After completing lab1, compare your data models and all transaction tables with the solution in the Download section of my web site http://www.uqac.ca/pgirard/english_version/trd157/partial_lab1.pdf

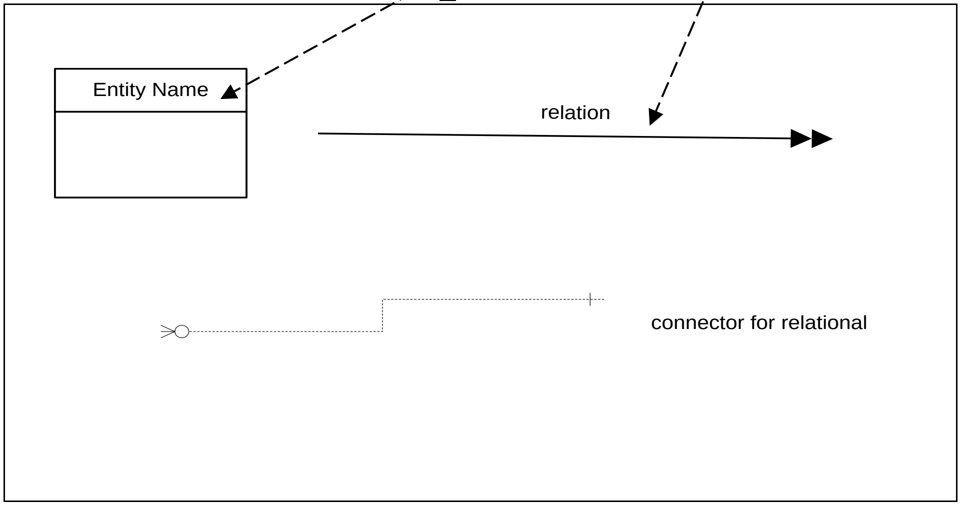
If you don't understand the solution, you don't understand the problem. Do it again.

Annex 1

1. E/R Data Model Symbols

This file can be downloaded from the web site. **You need only these 2 symbols** to create your E/R data models. Just duplicate them, fill attributes and use a meaningful name for each relation and entity names. To enter text, click on **A** in the top menu of Visio. Connect relations using the entity anchor points. Balance your data model to be easily understandable *(complex relations)* and try to prevent the crossing of lines.

visio_demo.vsd



2. Table (Word) defining each user transactions *(to be downloaded from web site)*

transaction_table.docx

Transaction	Description	Entity	Access type	Key

ANNEX 2

Example of a purchase order form

Company XXXXX

Purchase order number __________ **Date: ________**

VENDOR:
Supplier ID ___________________
Name: ___________________
Address: ___________________
____
Contact: __________________________________

AGENT
name _____________

product_id	product name	quantity	prod_unit	prod_unit_price	total

Total _____

Charge to account # ____________________

signature 1 : ________________ date ____________

signature 2 : ________________date ____________

signature 3 : ________________date ____________