ODBC REPORT

- SALE.C

The first program was used to create a sale. You call the program, and also pass the suitable parameters to it, and it fills the table sale with them.

We implemented all the queries using the command *sprintf*, and passing it a string. Thus, it was relatively easy to pass the query the suitable parameters, because this command works the same way *printf* does.

After creating this "query-string", we executed it, and, if needed, we got the data obtained from it (only in those cases when we needed to obtain some data).

In this exercise, first of all, we insert a new sale, and then we get the saleid generated by *postgres*. After this, we fill the table *sale_isbn* with this saleid and all the ISBNs passed by argument. The program checks in the loop if there are any ISBNs, and if there are, it executes the query to add to *isbn_sales*. If the arguments are passed wrongly the queries will not work, and, thanks to the command *odbc_extract_error,* the user that is passing the arguments will know what is failing.

- BUY.C

This program was the most complex of all. We had to design two functionalities in only one file:

If we pass the program the parameter *add,* it adds a new purchase to the database based on the parameters chosen. First of all, we get the last purchaseid introduced in the table *purchase* and we increment it and use it to create the new purchase (this parameter is printed as Invoice). After that, we get the *userid,* using the parameter *screenname* in a query.

Once we do that, we enter in a loop (that is not executed if there are not any ISBNs). In this loop, we take information like the discount (previously getting the saleid) and the amount of each book. After all that, we insert into the table *purchase* all the information generated and the given by parameters (like the date).

Each iteration of the loop sums the total amount of buying each book (discounts included)

- FILL_SALES.C

This program was designed to insert purchases into the database from a text file similar to "ventas.txt". In order to do so, we used ODBC to connect to our *PostgreSQL* database. To create the queries, we stored them as a string in a char pointer using the *sprintf* function, and later executed them with the *SQLExecDirect*. This was not the best way to do it, but with the improved implementation, we could not get it to work.

In this program, before executing any queries, we check if the arguments are correct, by checking if *argc* is less than 2. If so, the input file has not been specified, so we exit the execution of the program.

Later on, we create a temporary table, in which we insert all the values given, as well as the price value, for which we look in the *edition* table, using the book ISBN.

When this table has been fully filled, we transfer all its content to the *purchase* table, casting the content, and checking if every ISBN is in the *edition* table. This is checked because if not, the foreign key constraint is violated