SQL-Ledger User Guide

Written by
Sebastian Weitmann
Armaghan Saqib

International SQL-Ledger Network Association

September 2, 2013

Contents

Preface

TODO

Chapter 1

Introduction

1.1 Introducing SQL-Ledger

SQL-Ledger is an open source accounting/ERP solution written by Dieter Simader. Its version 1.0 was released in Jan. 29, 1999. So as of this writing in 2013, it is 14 years old software which is under constant development and enhancement during this period. This makes it suitable enough for small as well as for large businesses.

SQL-Ledger has an impressive feature set which even many commercial / properietory ERP solutions don't provide. It internal design and user interface are simple which make it easy to learn.

SQL-Ledger is an open source software which means that it comes with full source code and you can modify it as you wish and need not to worry about discovering any undocumented bug which may cost you tons of money to get fixed or force you to some other solution when the company behind it goes out of business.

Ledger123 is an enhanced version of SQL-Ledger. It was created to fix some of the bugs in SQL-Ledger which were not getting fixed. We still call Ledger123 as 'Enhanced SQL-Ledger'.

1.1.1 Versions

The current release of stock SQL-Ledger is 3.0.5. Ledger123 is based upon 3.0.3 with its enhancements. We call it Ledger123 release 3. Ledger123 tries to incorporate all the goodness which comes from stock SQL-Ledger. So you get

best of both worlds.

To make things simple, we assume that you are using Ledger123 release 3 (Enhanced SQL-Ledger release 3). Though most of the sections would apply equally well to the stock SQL-Ledger 3 as well as older version. This is particularly true if you are not using inventory related functions because most of the enhancements in Ledger 123 are related to inventory.

Website and other resources on Internet

1.2 Getting up and running

1.2.1 Installing enhanced sql-ledger using 'git clone'

The recommended way to download and install our enhanced SQL-Ledger is to use 'git' package. To install git on Ubuntu, you run 'sudo apt-get install git-core'. Once git is successfully installed, you can follow following steps:

1. Download the sql-ledger github repository. You will get a fully working sql-ledger installation which includes our enhancements. (The default 'master' branch)

```
git clone git://github.com/ledger123/ledger123.git
```

2. From now onwards you can upgrade to our latest enhancements (which includes any latest releases from sql-ledger.com) with the following simple commmand:

```
git pull
```

3. Let us say you are not interested in our enhancements and just want to maintain and upgrade to the sql-ledger release from sql-ledger.com. Switch to the sql-ledger branch first time:

```
git checkout -b sql-ledger origin/sql-ledger
```

4. From now onwards upgrading to official sql-ledger from sql-ledger.com is as easy as:

```
git pull
```

5. Note that you can always switch back and forth between our enhanced sql-ledger and the official one by git checkout as:

```
git checkout master # enhanced sql-ledger
git checkout sql-ledger # official sql-ledger
```

6. You can switch back to any past sql-ledger version. First see a log of all commits and 40 chars hashes:

```
git log --pretty=oneline
```

7. To revert to sql-ledger 2.8.17

```
git checkout 7b15e9b
```

SQL-Ledger Virtual Machine

1.3 Our enhancements to standard SQL-Ledger

1.3.1 Departments

1. Restrict user to a particular department using admin.pl.
2. Default department for user.
3. Department is mandatory on invoices/orders/quotes if there is at least one department defined.

1.3.2 Warehouses

1. Warehouse transfers module.
2. Restrict user to a particular warehouse using admin.pl.
3. Default warehouse for user.

4. Track warehouse inventory from sales and purchase invoices.
5. Track inventory-in-transit between warehouse movement.
6. Warehouse is mandatory on invoices if there is at least one warehouse defined.
7. Warehouse onhand and activity reports.

1.3.3 COGS

1. Re-posting script which corrects cogs errors due to invoice editing.
2. Invoice and invoice-item cogs/revenue information with gross profit %age.
3. Onhand value report which shows the inventory onhand quantities and value based upon fifo costing.

1.3.4 Reports

1. Per-invoice and per-item cogs/revenue information.
2. Enhanced tax reports. Audit trail report.
3. Drill-down to transactions from income statement.
4. Invoice date and customer/vendor filter in 'All Items' report.
5. Account description in 'GL Reports'.
6. Account activity report using 'GL Reports'.
7. Save report search conditions and layout in user menu. Recall with a single click.

1.3.5 Others

1. 'Add Customer', 'Add Vendor' links on invoices/orders/quotes/POS screens. These links appear only if allowed by access control settings.

2. Enhanced assemblies. You can get a report of all stock-assembly actions. Warehouses are correctly updated with any assemblies made and components used.
3. Enhanced bank reconciliation.
4. Added back the 'Shipping–Transfer' function from sql-ledger 2.6.
5. LedgerDoctor script which identifies potential problems with data entry.
6. CSV data import. (invoices,transactions,gl,orders,customers,vendors,parts,chart)
7. Disabled incorrect item weight update from orders and invoices
8. Parts group is mandatory if there is at least on group defined.

1.4 Explanation of bugs and gotchas in official version

1.4.1 Orders

1. Warehouse information is not updated when you receive orders by editing Rcvd quantity on orders.
2. When you make changes to invoice created from an order (add/remove item, quantities), inventory onhand count goes out of order. This is caused any invoice created from order does not update the inventory onhand.

We have fixed these issues by not allowing to receive orders by editing them. We also do not allow editing invoices created from orders to avoid corrupting onhand quantities.

1.4.2 COGS

1. Incorrect accounting transaction is posted for sale returns. COGS gets corrupted when you edit an invoice.
2. We have modified the posting of sale returns to post correct cogs.

We have added a reposting script to correct any corrupted cogs values when you edit an invoice.

1.4.3 Warehouses

Default warehouse functionality is broken in many ways. If somebody is successfully using it without ledger123 enhancements, I would love to know how?

Chapter 2

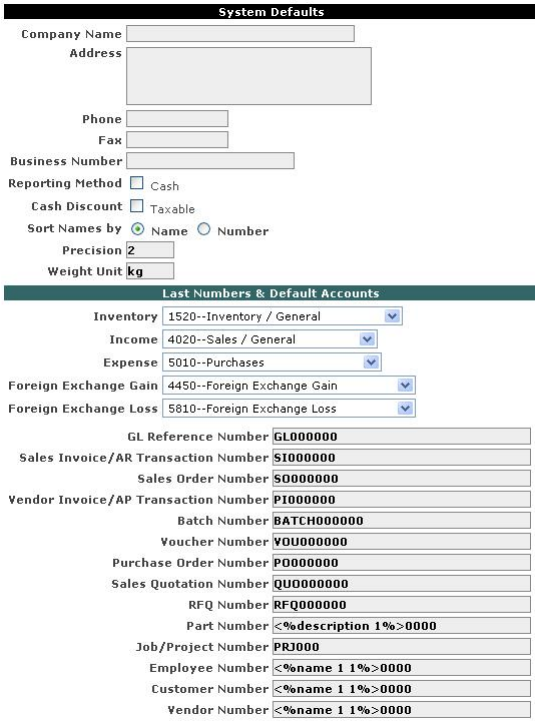
Setting up your business on SQL-Ledger

The next step after successful SQL-Ledger installation is to setup your initial business data. You need do do this before you start making your day to day transactions.

2.1 Defaults

System–Defaults menu allows you to setup your company, address and related information in SQL-Ledger. Document numbering is also controlled by system defaults.

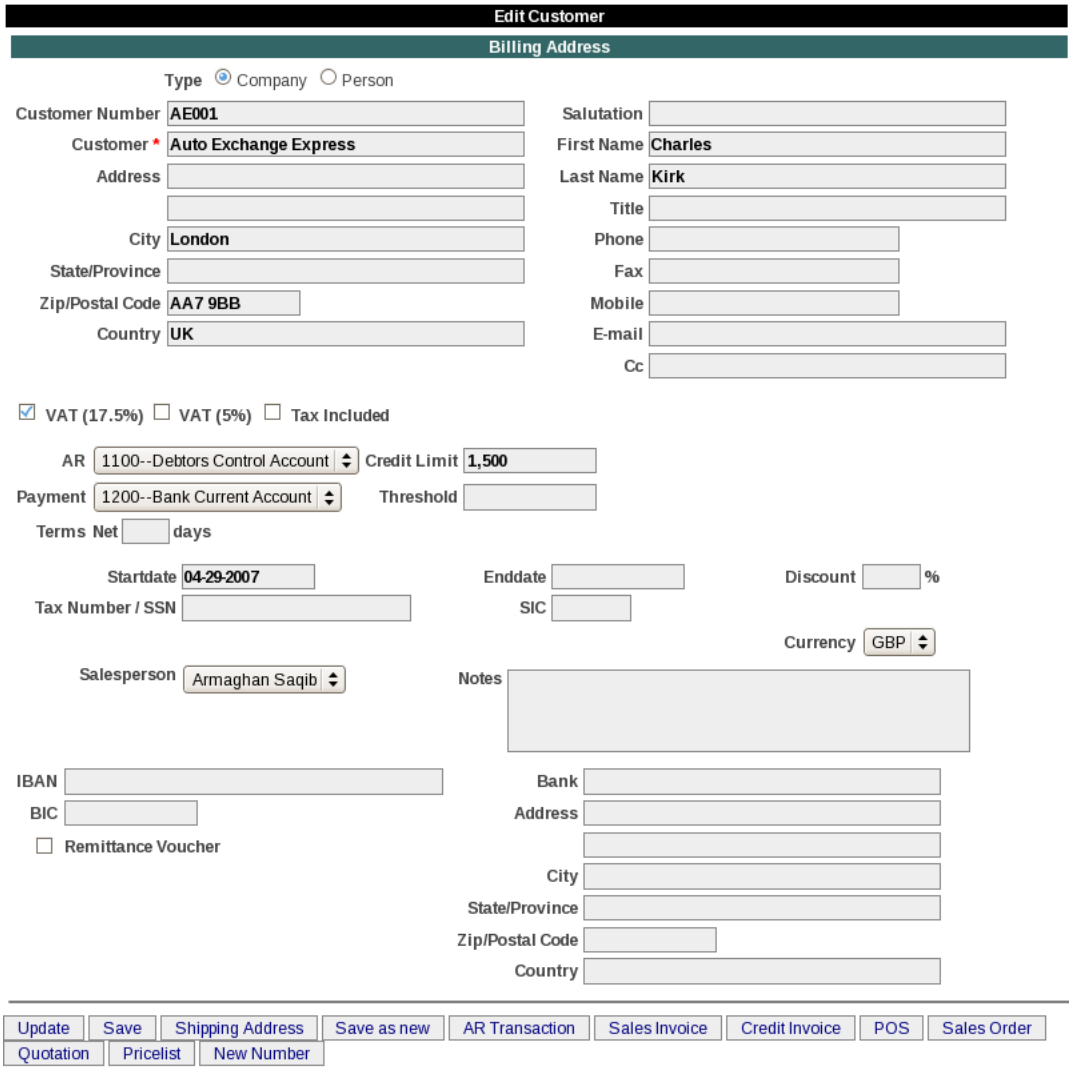
We setup defaults for document numbers as shown on the following screen shot. You can change these to your liking or organizational needs.



2.2 Customers

You need to add at least one customer before creating invoices. Use Customers–Add Customer to add new customers.

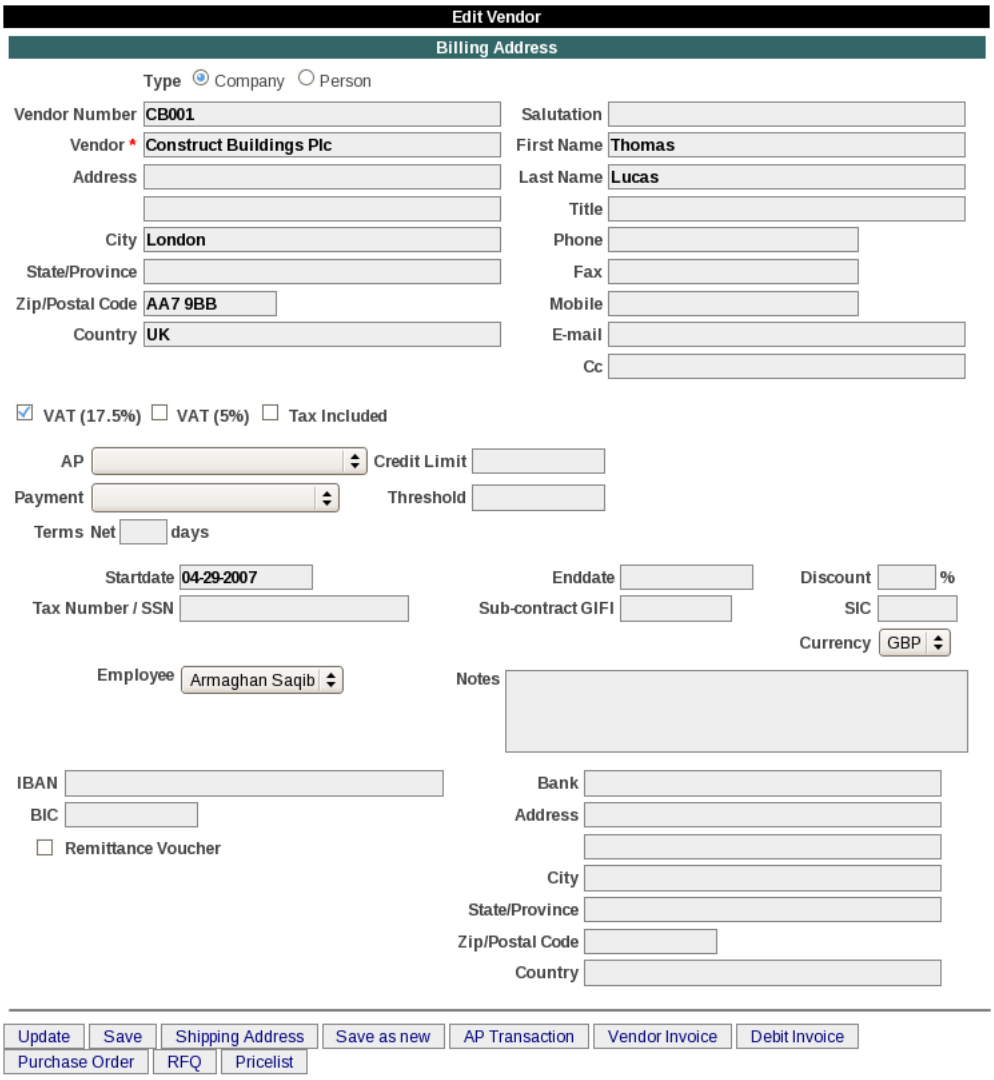
To change existing customers, first you list them using Customers–Reports–Search. Customers are listed with hyperlinks to edit each customer.



2.3 Vendors

You need to add at least one vendor before creating invoices. Use Vendors–Add Vendor to add new vendors.

To change existing vendors, first you list them using Vendors–Reports–Search. Vendors are listed with hyperlinks to edit each vendor.


Edit Vendor
Billing Address
Type Company Person
Vendor Number CB001
Vendor * Construct Buildings Plc
Address
City London
State/Province
Zip/Postal Code AA7 9BB
Country UK
Salutation
First Name Thomas
Last Name Lucas
Title
Phone
Fax
Mobile
E-mail
Cc
VAT (17.5%) VAT (5%) Tax Included
AP
Payment
Terms Net days
Credit Limit
Threshold
Startdate 04-29-2007
Enddate
Discount %
Tax Number / SSN
Sub-contract GIFI
SIC
Currency GBP
Employee Armaghan Saqib
Notes
IBAN
BIC
Remittance Voucher
Bank
Address
City
State/Province
Zip/Postal Code
Country
Update Save Shipping Address Save as new AP Transaction Vendor Invoice Debit Invoice
Purchase Order RFQ Pricelist

2.4 Type of Business

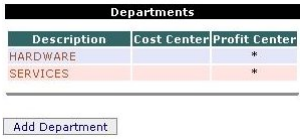
2.5 Employees

2.6 Departments

Departments are optional and can be used to classify transactions according to a department code.

2.6.1 Managing Departments

Departments can be added, changed or deleted using 'System–Departments' menu option.



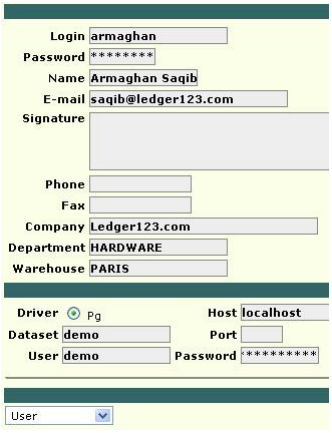
SQL-Ledger departments can be mapped to the various departments (sales, purchase etc.), branches (London, Oxford etc.) or product divisions (Product 1, Product2 etc.) within your organization.

Departments can be marked as 'Cost Center' or 'Profit Center'. Cost center departments appear only in purchasing module. Profit center departments appear both in purchasing and sales modules.

You can also change 'Department' to anything you like (eg.Branch) using the sql-ledger language customization feature. Note: Departments lookup does not appear on transaction forms unless you define at least one department from System→Departments menu option.

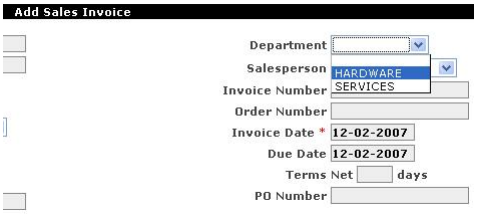
2.6.2 Default Department

You can define a default default for users through sql-ledger administrative interface. You can also restrict the user to view and make transactions to his department only by setting his role to User. Users with role Administrator, Manager, Supervisor always have access to all departments.



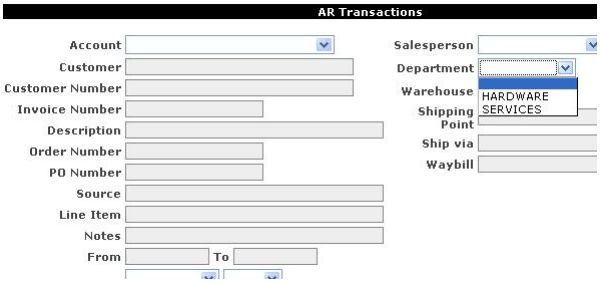
2.6.3 Using Departments

Once departments are defined you can specify them in your invoices, orders, quotations and other transactions.

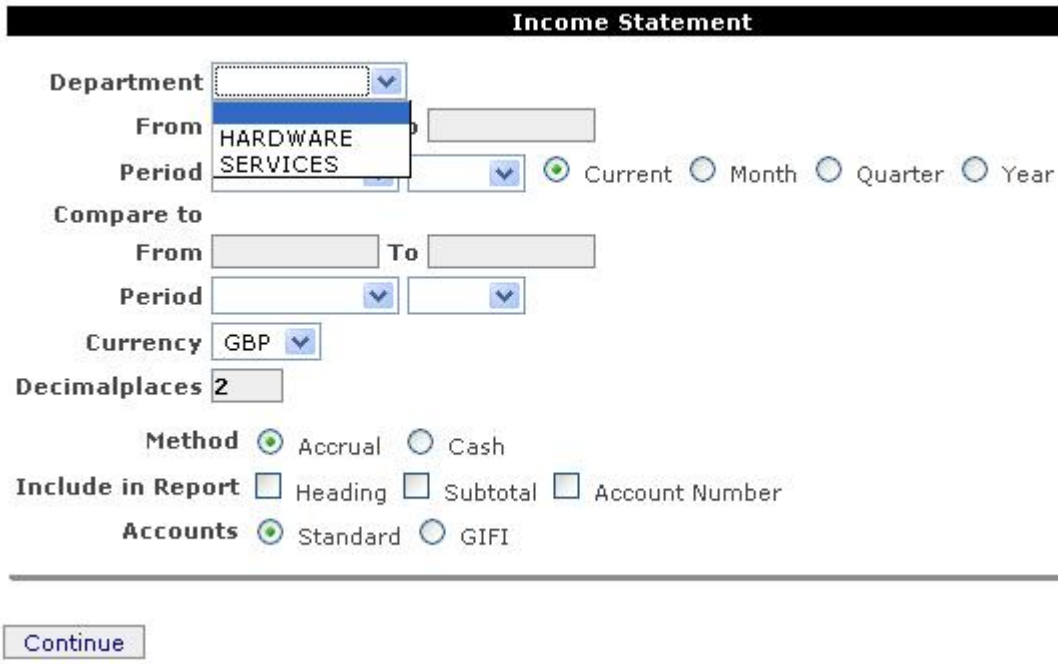


2.6.4 Reports

Reports allow you to view all or department specific transactions.



Income Statement and Balance sheet can also be compared and displayed by department.



2.7 Projects

Projects are optional and can be used for following things:

1. Track income and expenses to specific projects using invoices and general ledger transactions.

2. Enter time card data.

Notes Projects lookup appears on transactions forms only if you have created at least one project.

2.7.1 Managing Projects

You can add or change projects through Projects menu.

2.7.2 Using Projects

Once you have defined projects, you can use them in sales and purchase invoices.

2.8 Chart of Accounts

2.9 Templates

Print forms for invoices, orders, quotations and financial reports can be customized by you by editing form templates. There are three type of templates:

2.9.1 HTML Templates

HTML templates are easier to modify because it of wide spread knowledge of html. Only basic html knowledge is required to edit html templates.

TODO: Attach:templates1.jpg

2.9.2 Latex Templates

Latex templates are bit complex to understand and modify but offer complete control over printed invoice, order or quotation forms. See below for basic introduction to latex.

TODO: Attach:templates3.jpg

2.9.3 Text Templates. Used only with Point-of-Sale interface

Text templates are used only for Point-of-Sale receipts printing. These templates allow you to print on 40 character receipt printers.

TODO: Attach:templates4.jpg

2.9.4 Editing Templates

Templates can be edit through sql-ledger. When you click on a template, it is displayed with 'Edit' button at the end of the template. Clicking the 'Edit' button will open the template in a text box where it can be edited and saved.

Attach:templates2.jpg

2.9.5 Template Variables

Sql-ledger replaces actual data into templates using variables which we call template variables. Template variables are enclosed within <% and %>.

Here are some template variables to give you an idea. The best way to view all these template variables and understand their usage is by going through existing templates.

```
<%name%>
<%address1%>
<%address2%>
<%city%>
<%state%>
<%zipcode%>
<%country%>
<%contact%>
<%invnumber%>
<%invdate%>
<%duedate%>
<%ordnumber%>
```

```
<%employee%>
<%shippingpoint%>
<%shipvia%>
<%runningnumber%>
<%number%>
<%description%>
<%deliverydate%>
<%qty%>
<%unit%>
<%sellprice%>
<%discountrate%>
<%linetotal%>
```

2.9.6 Template control commands

Template processing engine in sql-ledger allows simple if statement and loops. Example of these are described below:

2.9.6.1 'if' is used to print a column data conditionally

```
<%if contact%>
  <br><%contact%>
  <br>
<%end contact%>

<%if taxincluded%>
   <th colspan=7 align=right>Total</th>
   <td colspan=2 align=right><%invtotal%></td>
<%end taxincluded%>

<%if not taxincluded%>
   <th colspan=7 align=right>Subtotal</th>
   <td colspan=2 align=right><%subtotal%></td>
<%end taxincluded%>

<%if paid%>
   <tr>
      <th colspan=7 align=right>Paid</th>
      <td colspan=2 align=right>- <%paid%></td>
   </tr>
<%end paid%>
```

2.9.6.2 'for' loop to print all lines on an invoice

```
<%foreach number%>
        <tr valign=top>
          <td align=right><%runningnumber%>.</td>
          <td><%number%></td>
          <td><%description%></td>
          <td><%deliverydate%></td>
          <td align=right><%qty%></td>
          <td><%unit%></td>
          <td align=right><%sellprice%></td>
          <td align=right><%discountrate%></td>
          <td align=right><%linetotal%></td>
        </tr>
<%end number%>

<%foreach tax%>
  <tr>
     <th colspan=7 align=right><%taxdescription%> on <%taxbase%> @ <%taxrate%> %
     <td colspan=2 align=right><%tax%></td>
  </tr>
<%end tax%>
```

2.9.7 An Introduction to Latex

Latex is a complete collection of software tools to create high quaility print documents. Latex templates are used in SQL-Ledger to create high quality print forms like invoices, purhcase orders etc.

Latex is included with Redhat distributions (rpm -qa | grep tetex).

For FreeBSD, you can install the teTex port from /usr/ports/print/teTEX.

Latex migh seem overwhelming to a new comer but it is really a simple toolkit to use for customizing the SQL-Ledger templates. In this very short introduction of Latex, we shall go through the basic document format and its use in SQL-Ledger.

Here is 'Hello world!' in latex.

2.9.7.1 Create a text file (hello.tex) in your home folder with following text:

```
\documentclass[a4paper,11pt]{article}
\begin{document}
Hello world!
\end{document}
```

2.9.7.2 Compile this tex file into dvi file and use xdvi to view it:

```
latex hello.tex
xdvi hello.dvi
```

2.9.7.3 You can also convert it to pdf:

```
pdflatex hello.tex
xpdf hello.pdf
```

2.9.8 Structure of a Latex Document

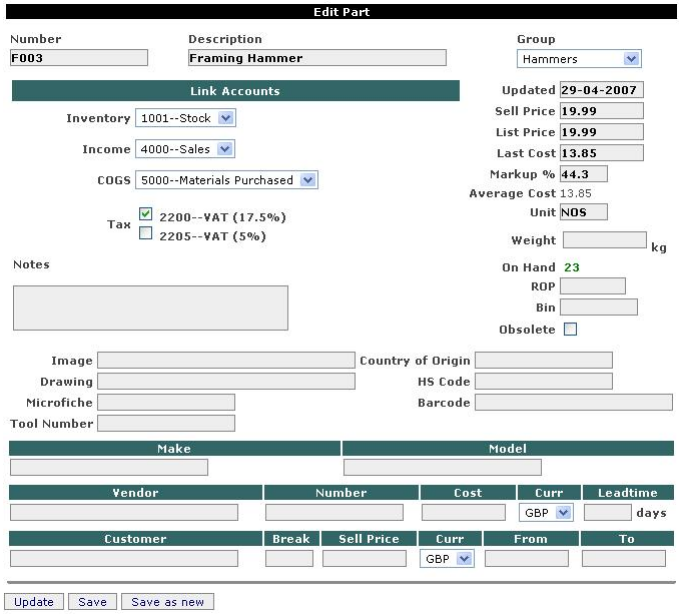
Latex commands start with a backslash (\). Parameters can follow the command. Optional parameters are enclosed in [] while mandatory ones are enclosed in {}. {} can also be used to terminated a command mixed within some text (to make it easier to understand the command for the compiler). Special characters in latex (#, $, %, ^, &, _, {, }, ~) are escaped with \ except for the \ character itself (because is used to break a line). To use literal backslash (\) use can use special command $\backslash$.

Single line comments start with % while multi-line comments can be enclosed between \begin{comment} and \end{comment} structure.

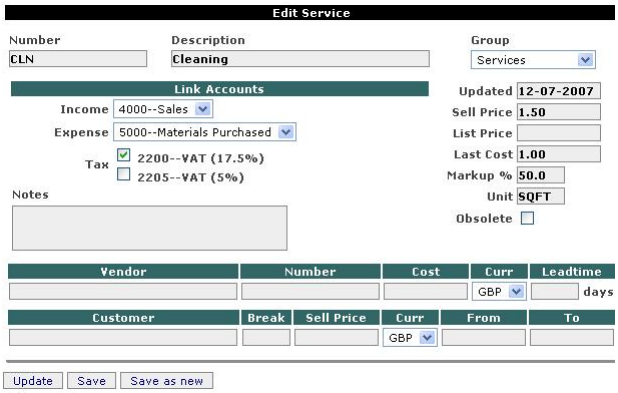
Every latex document starts with \documentclass with parameters ([a4paper,11pt]{article}) following it.

2.10 Parts

Parts are tangible items you keep in your stock. You purchase them from your vendors and sell them to your customers for profit.



2.11 Services



2.12 Assemblies

An assembly is composed of components which are individual parts in the inventory or other sub-assemblies. Assemblies in SQL-Ledger allow you to do manage your manufacturing process.

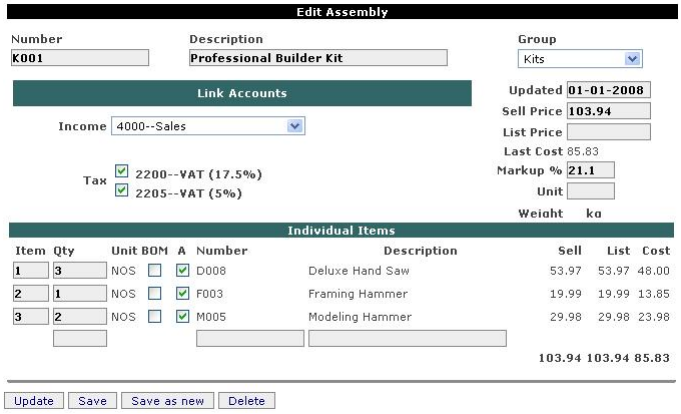
Work flow for using assemblies:

1. Define assemblies. Goods & Services–Add Assembly.
2. Build assemblies. Goods & Services–Stock Assembly. Individual parts are removed and assemblies are added to the stock inventory.
3. Sell assembly items like any other item.

Please note that you cannot buy parts defined as assemblies.

2.12.1 Define assemblies

As assembly is just like any other inventory item in your sql-ledger with the additional information about its components. You define new assemblies using Goods and Service – Add Assembly.



2.12.2 Stock assemblies

This option reduces the quantities of the components and increases the onhand quantity of the assemblies. COGS is not recorded at this point.

COGS for the assembly is recorded from individual components when you sell the assembly. FIFO allocation also occurs at the time of sale. (Rows are inserted in invoice table for component parts with assemblyitem=TRUE)

2.12.3 Reports

More Reports–Goods and Services–Stock Assembly gives you a list of your Stock Assembly actions. This report lists the parts taken out of assembly as well as assemblies built.

Stock Assembly

Reference	Date	Department	Warehouse	Number	Description	Qty	Unit
P031	11-01-2008		LONDON	K001	Professional Kit	5	
P031	11-01-2008		LONDON	D008	Deluxe Hand Saw	-15	NOS
P031	11-01-2008		LONDON	F003	Framing Hammer	-5	NOS
P031	11-01-2008		LONDON	M005	Modeling Hammer	-10	NOS
						-25	
build1	15-03-2008		LONDON	K001	Professional Kit	3	
build1	15-03-2008		LONDON	M005	Modeling Hammer	-6	NOS
build1	15-03-2008		LONDON	F003	Framing Hammer	-3	NOS
build1	15-03-2008		LONDON	D008	Deluxe Hand Saw	-9	NOS
						-15	
						-40	

Goods and Services–Assemblies gives you list of all or selected assemblies with their components.

Assemblies /

Active :

Number		Description	Qty	Unit	Sell Price	Last Cost	Avg Cost
K001		Professional Kit	7		103.94	85.83	
	D008	Deluxe Hand Saw		NOS	17.99	16.00	16.00
	F003	Framing Hammer		NOS	19.99	13.85	13.85
	M005	Modeling Hammer		NOS	14.99	11.99	11.99
K002		Standard Builder Kit	6		32.98	27.99	
	M005	Modeling Hammer		NOS	14.99	11.99	11.99
	D008	Deluxe Hand Saw		NOS	17.99	16.00	16.00

Goods and Servers–Components gives you a list order by partnumber and the assembly in which it is used.

Components /

Active :

Number	Assembly	Description	Qty	Unit	Sell Price	Last Cost	Avg Cost
D008	K001	Deluxe Hand Saw	3	NOS	17.99	16.00	16.00
	K002	Deluxe Hand Saw	1	NOS	17.99	16.00	16.00
F003	K001	Framing Hammer	1	NOS	19.99	13.85	13.85
M005	K001	Modeling Hammer	2	NOS	14.99	11.99	11.99
	K002	Modeling Hammer	1	NOS	14.99	11.99	11.99

Work Order You can print work order for a sales orders. Work order lists all component parts required to fullfil a given order of assembly items.

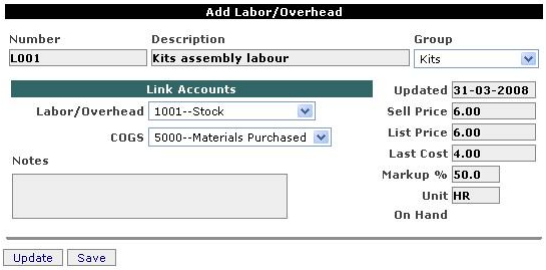
WORK ORDER

To	Ship To
Big Porridge Ltd. London AA7 9BB UK Attn: Milton Bear	Big Porridge Ltd. London AA7 9BB UK Milton Bear

Order #	Order Date	Required by	Salesperson	Shipping Point	Ship Via
SO-002	01-04-2008		Armaghan Saqib		

Item	Number	Description	Qt'y		Bin	Serial #
1.	K001	Professional Kit	3			
.	D008	Deluxe Hand Saw	9	NOS		
.	F003	Framing Hammer	3	NOS		
.	M005	Modeling Hammer	6	NOS		
2.	K002	Standard Builder Kit	5			
.	M005	Modeling Hammer	5	NOS		
.	D008	Deluxe Hand Saw	5	NOS		

2.13 Labour/Overhead



2.14 Groups

Groups are used to group togather the parts and services. You can filter parts and services reports by selecting a group on search screens.

Groups have another useful functionality. When you check the POS button box during group add or change, they appear as buttons on POS (point-of-sale) screens making it easier to select items within each group.

2.15 Pricegroups

SQL-Ledger has very flexiable pricing mechanism. For example:

1. You can define customer specific prices for each part.
2. You can define quantity breaks. (If someone buys 10 units instead of 1, he/she can automatically gets lower price.)
3. And you can specify start and end dates to offer a special price during, for example, Christmas season.

Price groups take this concept further and allow you to define 'groups' of special prices. Let us say you sell to distributor, dealer and end-user. Each of these groups of customers gets tiered discount/price.

There are three steps to use price groups:

1. Create three price groups; distributor, dealer and enduser. (Goods & Services-Add Pricegroup)

2. Define item prices for these price groups. To do this, open the item for editing and select the price group and set the price according to the price group tier. Leave the customer column blank. Repeat this for all items. (Clicking 'Update' will allow you to set prices for multiple pricegroups for a single item.)

3. Open the customer record for editing and set the applicable price group for that customer.

2.16 Warehouses

Warehouses are optional and can be used to manage your inventory at more than one physical place.

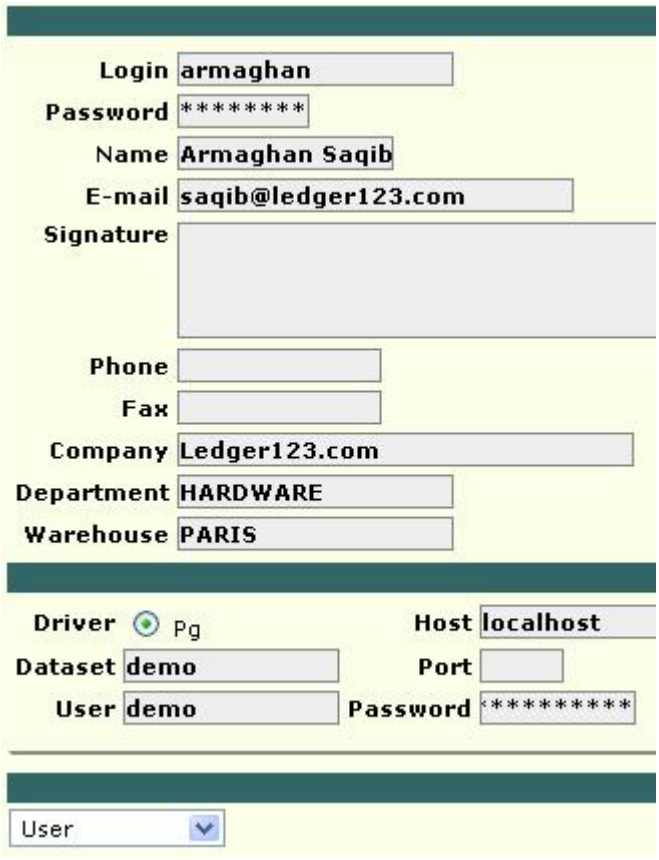
Important: Once you have defined warehouses, these are no longer optional and you cannot save a transaction (invoice or transfer) without specifying a warehouse.

2.16.1 Adding warehouses

You can add, change or delete warehouses through 'System–Warehouses' option.

2.16.2 Default warehouse

You can define a default warehouse for users through administrative interface. You can restrict a user to view and make transactions to his warehouse by setting his role to User. Users with role Administrator, Manager, Supervisor always have access to all warehouses.



2.16.3 Using warehouses

Warehouse drop down is enabled on relevant transactions forms once you define at least one warehouse. When you purchase goods, quantity is added to the specified warehouse. When you sell goods, quantity is subtracted from the specified warehouse.

2.16.4 Warehouse transfers

You can move inventory between warehouses by using 'Warehouses–Add Transfer' menu option.

2.16.5 Transfers delivered

Some companies also need to track the in-transit goods between warehouse transfers. Delivered date is usually different from transfer date.

When you login, you will see the number of transfers which have been sent to your default warehouse but not received by you yet.

To 'receive' the transfers, click the 'Warehouses–Reports–Deliveries' menu option, specify criteria and click Continue to display the transfers pending to be received. Here you specify the dates when the goods were delivered at 'your' warehouse and click 'Save Delivered'.

Transfers to be Delivered

My Warehouse LONDON

Date	Transfer Number	Description	Notes	From WH	Delivered Date
01-18-2008	12311	Hand saw and brush sets	New shipment.	HONGKONG	
01-19-2008	12322	Hand saw and hand trencher	Quantity at Paris	PARIS	

Save Delivered

2.16.6 Reports

1. Goods & Services–Parts report provides summary of your on hand quantity at selected or all warehouses. Click 'Warehouse' check box to display onhand by warehouse.

2. Warehouses–Reports–Transfers gives you a list of transfers. Summary lists transfer transactions and Detail lists all items in each transfer transaction. You can click on transfer number hyper link to edit the transfer.

3. Warehouses–Reports–Onhand gives you inventory onhand for all warehouses or for a particular warehouse.

4. Warehouses–Reports–Activity gives you all activity of a particular item or all items. Select warehouse to see the activity in a particular warehouse. Activity report shows activity from purchase invoices, sales invoices and transfers.

Parts /

Active :
Number : D008

Number	Description	Qty	Unit	Sell Price	Last Cost	Avg Cost	Warehouse
D008	Deluxe Hand Saw	38	NOS	17.99	16.00	16.00	
		1					LONDON
		37					PARIS

2.16.7 Enabling multiple warehouses for old dataset

If you have upgraded your sql-ledger installation with our enhanced version, you need to run few queries to bring your old data in sync with the new warehouses structure.

Assemblies are a special case. In standard sql-ledger, 'Stock Assembly' action does not create any transaction/log and directly updates the onhand quantities in parts table. If you are using assemblies, you will almost always need to adjust the components and assemblies quantities after running these queries. See step 4 below.

Important: Make sure you have a current backup before doing this.

TODO: Copy queries and other text here. See how code can be formatted properly

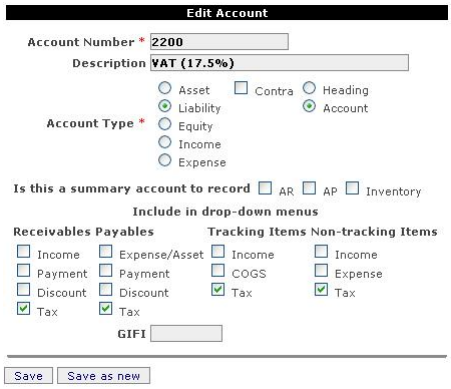
2.17 Translations

2.18 Taxes

Defining and using taxes in sql-ledger is a four step process:

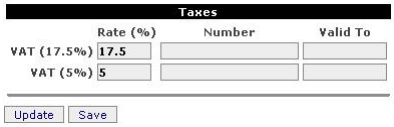
2.18.1 Define tax accounts in chart

You create (or edit) tax accounts in chart of accounts using System–Accounts menu option.



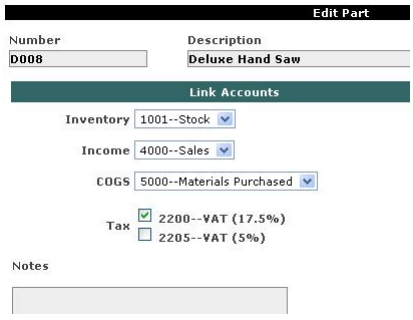
2.18.2 Define tax percentages

You set percentages for each tax using System–Taxes menu option.



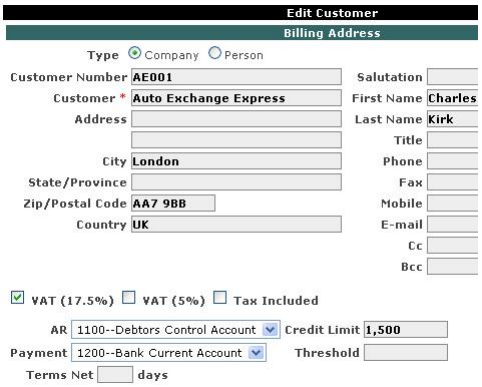
2.18.3 Mark Items/services as taxable

You mark each part or service taxable during add or edit process. You do this using Goods & Services menu option.



2.18.4 Mark customers/vendors for applicable taxes

Tax will not be calculated for your customers or vendors unless you mark them as taxable.



2.19 Data import from other applications

Sometimes you need to import your sales data into sql-ledger which was produced elsewhere.

You might have a web store where you download your daily sales in CSV format and want to import it into Sql-Ledger. Or you are just moving to sql-ledger from your legacy accounting software and want to move all existing data from old software to sql-ledger.

Following sections provide detailed steps for importing CSV text files.

2.19.1 Sale invoices

Sales invoices can be imported from text files.

2.19.1.1 Format your data

Here is a sample import data. You prepare data in this format and save it in a text file. The last column AR is accounts receivable account number which is 1100 in UK chart of accounts.

If your data contains invoices with more than one item, repeat the row with same invoice header information and change the item number and price information. SQL-Ledger will import all these rows as a single invoice. (See invoice number A100 above)

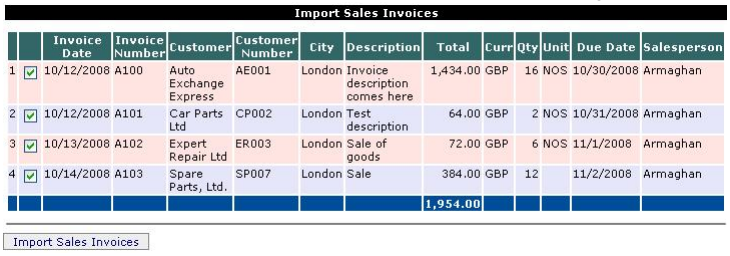
For list of additional data columns that can be imported see step 4.

```
invnumber,transdate,duedate,customernumber,curr,invoicedescription,
    partnumber,
qty,sellprice,employeenumber,AR,department,warehouse
```

```
A100,10/12/2008,10/30/2008,AE001,GBP,Invoice description comes here,B001
    ,10,102,E-001,1100,HARDWARE,LONDON
A100,10/12/2008,10/30/2008,AE001,GBP,Invoice description comes here,F003
    ,6,69,E-001,1100,HARDWARE,LONDON
A101,10/12/2008,10/31/2008,CP002,GBP,Test description,F003,2,32,E-002,1100,
    SERVICES,PARIS
A102,10/13/2008,11/1/2008,ER003,GBP,Sale of goods,T007,6,12,E-003,1100,
    SERVICES,LONDON
A103,10/14/2008,11/2/2008,SP007,GBP,Sale,K001,12,32,E-004,1100,HARDWARE,
    PARIS
```

2.19.1.2 Upload and preview

Using Impor–Sales Invoices menu option, upload this file into Sql-Ledger. You will be shown what will be imported before actual import is done. At this point you can check and uncheck the invoices to be imported.



Import Sales Invoices

		Invoice Date	Invoice Number	Customer	Customer Number	City	Description	Total	Curr	Qty	Unit	Due Date	Salesperson
1	☑	10/12/2008	A100	Auto Exchange Express	AE001	London	Invoice description comes here	1,434.00	GBP	16	NOS	10/30/2008	Armaghan
2	☑	10/12/2008	A101	Car Parts Ltd	CP002	London	Test description	64.00	GBP	2	NOS	10/31/2008	Armaghan
3	☑	10/13/2008	A102	Expert Repair Ltd	ER003	London	Sale of goods	72.00	GBP	6	NOS	11/1/2008	Armaghan
4	☑	10/14/2008	A103	Spare Parts, Ltd.	SP007	London	Sale	384.00	GBP	12		11/2/2008	Armaghan
								1,954.00					

2.19.1.3 Confirm data import

When you click the Import Sales Invoices button, invoices will be imported. You will be show which invoices were imported successfully.

1. Posting Invoice ... A102, Sale of goods, ER003, Expert Repair Ltd, London, 72 ... ok
2. Posting Invoice ... A100, Invoice description comes here, AE001, Auto Exchange Express, London, 1,434 ... ok
3. Posting Invoice ... A101, Test description, CP002, Car Parts Ltd, London, 64 ... ok
4. Posting Invoice ... A103, Sale, SP007, Spare Parts, Ltd., London, 384 ... ok

Total: 1,954

2.19.1.4 Additional data which can be imported

Sample csv file provided above contains only the most commonly used columns. Here is the complete list.

transdate
invnumber
customernumber
curr
duedate
employeenumber

ordnumber
quonumber
datepaid
shippingpoint
shipvia
waybill
terms
notes
intnotes
language_code
ponumber
cashdiscount
discountterms
partnumber
description
sellprice
discount
qty
unit
serialnumber
projectnumber
deliverydate
AR
taxincluded

2.19.2 Receipts and Payments

You can import payments and match them to invoices using 'Import–Payments'. Following points should be kept in mind.

1. Payments are matched first on Invoice DCN column and then, if no match is found, on payment amount.

2. Both AR and AP invoices are matched with payments.

3. The amount matched is calculated as debit minus credit.

2.19.2.1 Format your data

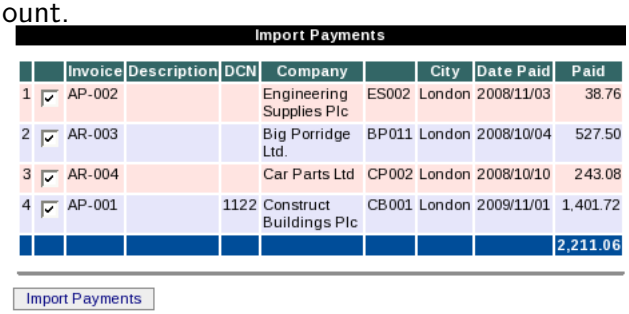
Create or format the data in a CSV file with structure similar to the given below.

```
datepaid,memo,debit,credit,dcn
2008/11/03,"payment ref 2121",,38.76,
2008/10/04,"cash payment",,527.5, 2008/10/10,"CC Receipt",,243.08,
2009/11/01,"Payment matched by DCN",,1401.72,1122
```

2.19.2.2 Upload and perview

Import script will read the CSV file and match the payments to AR or AP invoices first on DCN Number and then on invoice due amount, if needed.

In this example, one AP invoice is matched on amount and the other one is matched on DCN number. The other two are AR invoices which are matched on amount.



Import Payments

		Invoice	Description	DCN	Company		City	Date Paid	Paid
1	☑	AP-002			Engineering Supplies Plc	ES002	London	2008/11/03	38.76
2	☑	AR-003			Big Porridge Ltd.	BP011	London	2008/10/04	527.50
3	☑	AR-004			Car Parts Ltd	CP002	London	2008/10/10	243.08
4	☑	AP-001		1122	Construct Buildings Plc	CB001	London	2009/11/01	1,401.72
									2,211.06

Import Payments

2.19.2.3 Confirm data import

Once you click 'Import Payments', payments are imported and applied to the matched invoices.

```
1. Posting Payment ... AP-002, , ES002, Engineering Supplies Plc, London, 38.76 ...
ok
2. Posting Payment ... AR-003, , BP011, Big Porridge Ltd., London, 527.50 ... ok
3. Posting Payment ... AR-004, , CP002, Car Parts Ltd, London, 243.08 ... ok
4. Posting Payment ... AP-001, , CB001, Construct Buildings Plc, London,
1,401.72 ... ok
```

2.19.2.4 Advanced receipts/payments import

1. You can easily change the script to match the payments on other invoice columns like invoice number. The procedures to modify are 'sub payments' in 'SL/IM.pm' and 'sub im_payment' in 'bin/mozilla/im.pl'.

2. To match payments only to AR (or AP) invoices, change the UNION queries in SL/IM.pm to select invoices from AR or AP only as required.

2.19.3 AR/AP Transactions

You can import AR and AP transactions.

For AR transactions, format your data using following sample:

```
invnumber,customernumber,transdate,amount,description,notes,source,memo
00003,AE001,10-11-07,2030,"desc1","notes1","source1","memo1"
00004,CP002,07-12-07,3213,"desc1","notes2","source2","memo2"
00005,SP007,09-12-07,-200,"desc1","notes3","source3","memo3"
```

For AP transactions, format your data using following sample:

```
invnumber,vendornumber,transdate,amount,description,notes,source,memo
00003,CB001,10-10-08,2030,"desc1","notes1","source1","memo1"
00004,ES002,10-12-08,3213,"desc2","notes2","source2","memo2"
00005,SA003,12-12-08,-200,"desc3","notes3","source3","memo3"
```

2.19.4 General Ledger

This feature will help you to move your data from most of the accounting software to sql-ledger in few easy steps:

2.19.4.1 Format your data

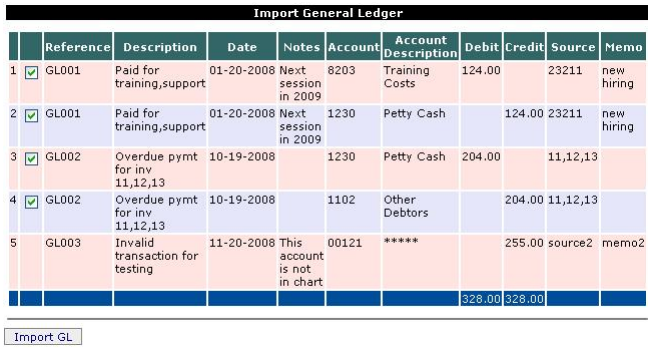
Format your data according to following sample. Keep in mind that:

1. Import script creates one GL transaction for each unique 'reference' number.
2. There can be any number of lines (rows) in each transaction.
3. Account must exist in chart of accounts Debits and credits must be equal before the CSV file can be imported.

```
reference,transdate,description,notes,accno,debit,credit,source,memo
GL001,01-20-2008,"Paid for training,support",Next session in
    2009,8203,124,0,23211,new hiring
GL001,01-20-2008,"Paid for training,support",Next session in
    2009,1230,0,124,23211,new hiring
GL002,10-19-2008,"Overdue pymt for inv 11,12,13",,1230,204,0,"11,12,13",
GL002,10-19-2008,"Overdue pymt for inv 11,12,13",,1102,0,204,"11,12,13",
GL003,11-20-2008,Invalid transaction for testing,This account is not in
    chart,00121,0,255,source2,memo2
```

2.19.4.2 Upload and preview

Using 'Imports–GL Transaction' load the CSV file into sql-ledger. Import script will show the rows which contain valid account number and can be imported.



Import General Ledger

		Reference	Description	Date	Notes	Account	Account Description	Debit	Credit	Source	Memo
1	☑	GL001	Paid for training,support	01-20-2008	Next session in 2009	8203	Training Costs	124.00		23211	new hiring
2	☑	GL001	Paid for training,support	01-20-2008	Next session in 2009	1230	Petty Cash		124.00	23211	new hiring
3	☑	GL002	Overdue pymt for inv 11,12,13	10-19-2008		1230	Petty Cash	204.00		11,12,13	
4	☑	GL002	Overdue pymt for inv 11,12,13	10-19-2008		1102	Other Debtors		204.00	11,12,13	
5		GL003	Invalid transaction for testing	11-20-2008	This account is not in chart	00121	*****		255.00	source2	memo2
								328.00	328.00		

Import GL

2.19.4.3 Confirm data import

Click Import GL to finish the import script. Transactions successfully imported will be show on the next page.

3. Posting gl transaction ... GL001 ... ok
4. Posting last gl transaction ... GL002 ... ok

2.19.5 Customers and Vendors

Customer and Vendor import is similar (except the number column which is customernumber or vendornumber).

Prepare your data file using the sample text provided below. (Change customernumber to vendornumber for vendor import)

```
customernumber,name,firstname,lastname,contacttitle,phone,fax,email,notes,
    address1,address2,city,state,zipcode,country
001,Ledger123,Armaghan,Saqib,Consultant,,,saqib@ledger123.com,"These are,
    just, sample notes",,,London,,"AA7 8BB",UK
```

2.19.6 Parts

2.19.6.1 Format your data

Format your data according to following sample format. Please note that:

1. Import procedure assigns a unique parts_id to each part imported or group created.

2. Duplicates are not allowed and duplicate check is done on partnumber.

```
partnumber,description,unit,partsgroup,listprice,sellprice,lastcost,rop,bin,
    image,drawing,notes
B002,"Brush Set",NOS,brush,9.99,9.99,7,150,TOP,noimage,brush.jpg,notes about
     brush set
D010,"Deluxe Hand Saw",NOS,SAW,17.99,17.99,16,50,TOP,saw.jpg,nodrawing,notes
     about hand saw
D011,"Digger Hand Trencher",NOS,Picks & Hatchets,18.99,18.99,15,200,TOP,,
    nodrawing,notes about hand saw
```

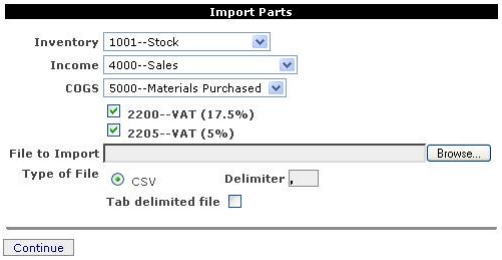
2.19.6.2 Upload and preview

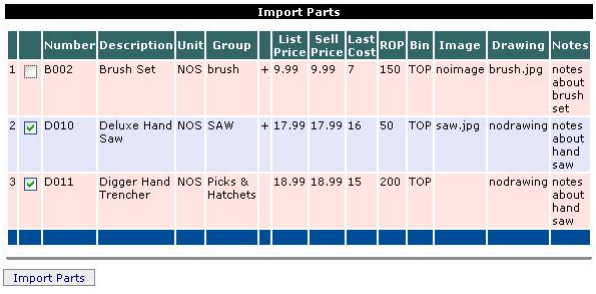
To start the import process, click 'Data Import–Parts' in the menu. Following page will be displayed. Click 'Browse' to select your CSV file, mark the taxes applicable and select the account links (Defaults are enough most of the time) Click 'Continue' when done. You will be presented with the following screen. On this screen you can mark the parts to be imported by checking or un-checking the checkbox on each line.

Please note:

1. The parts which are already in the system (based on partnumber) will not imported. (You will not see a check box with them)

2. Parts groups which are new will be added. These are marked by a '+' sign after group name.





2.19.6.3 Confirm data import

Click 'Import Parts'. Your CSV file will be processed and parts will be imported. Any new groups will also be added. You will see an output like the following:

```
1. Add part ... D010, Deluxe Hand Saw ... ok
2. Add part ... D011, Digger Hand Trencher ... ok
Parts imported
```

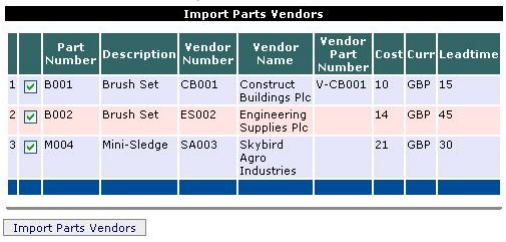
2.19.7 Vendor price list

2.19.7.1 Format your data

```
partnumber,vendornumber,vendorpartnumber,lastcost,curr,leadtime
B001,CB001,V-CB001,10,GBP,15 B002,ES002,,14,GBP,45 M004,SA003,,21,GBP,30
```

2.19.7.2 Upload and preview

Click 'Data Import–Parts Vendors', specify the file with the 'Browse' button and click 'Import Parts Vendors' button. Following page will be displayed. Here you can un-check the rows which you do not want to import. Rows with invalid vendor number or partnumber will not have the checkbox.



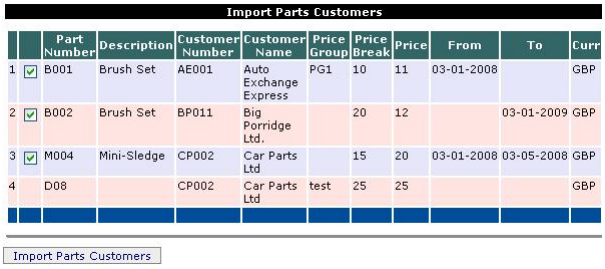
2.19.8 Customer price list

2.19.8.1 Format your data

```
partnumber,customernumber,pricegroup,pricebreak,sellprice,validfrom,validto,curr
B001,AE001,PG1,10,11,03-01-2008,,GBP
B002,BP011,,20,12,,03-01-2009,GBP
M004,CP002,,15,20,03-01-2008,03-05-2008,GBP
D08,CP002,test,25,25,,,GBP
```

2.19.8.2 Upload and preview

Click 'Data Import–Parts Customers', specify the file with the 'Browse' button and click 'Import Parts Customers' button. Following page will be displayed. Here you can un-check the rows which you do not want to import. Rows with invalid customer number or partnumber will not have the checkbox.



Import Parts Customers

		Part Number	Description	Customer Number	Customer Name	Price Group	Price Break	Price	From	To	Curr
1	☑	B001	Brush Set	AE001	Auto Exchange Express	PG1	10	11	03-01-2008		GBP
2	☑	B002	Brush Set	BP011	Big Porridge Ltd.		20	12		03-01-2009	GBP
3	☑	M004	Mini-Sledge	CP002	Car Parts Ltd		15	20	03-01-2008	03-05-2008	GBP
4		D08		CP002	Car Parts Ltd	test	25	25			GBP

Import Parts Customers

2.19.9 Chart of accounts

2.19.9.1 Format your data

1. Prepare your chart of accounts in your spreadsheet software according to the sample given below.

2. Upload the chart csv file using 'Import–Chart' menu option.

3. Check/uncheck the accounts to be imported and click continue to import the selected accounts.

```
accno,description,charttype,category,link
1000,"CURRENT ASSETS",H,A,
1060,"Checking Account",A,A,AR_paid:AP_paid
1065,"Petty Cash",A,A,AR_paid:AP_paid
1200,"Accounts Receivables",A,A,AR
1205,"Allowance for doubtful accounts",A,A,
1500,"INVENTORY ASSETS",H,A,
1520,"Inventory / General",A,A,IC
1530,"Inventory / Aftermarket Parts",A,A,IC
1800,"CAPITAL ASSETS",H,A,
```

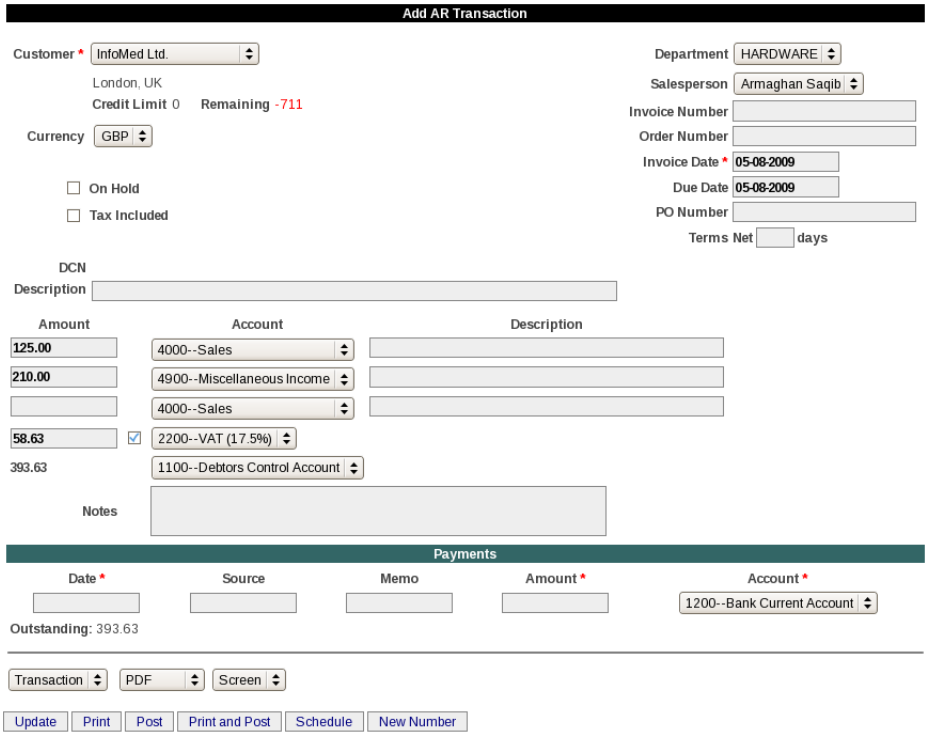
Chapter 3

Running your business on SQL-Ledger

3.1 AR

3.1.1 AR Transaction

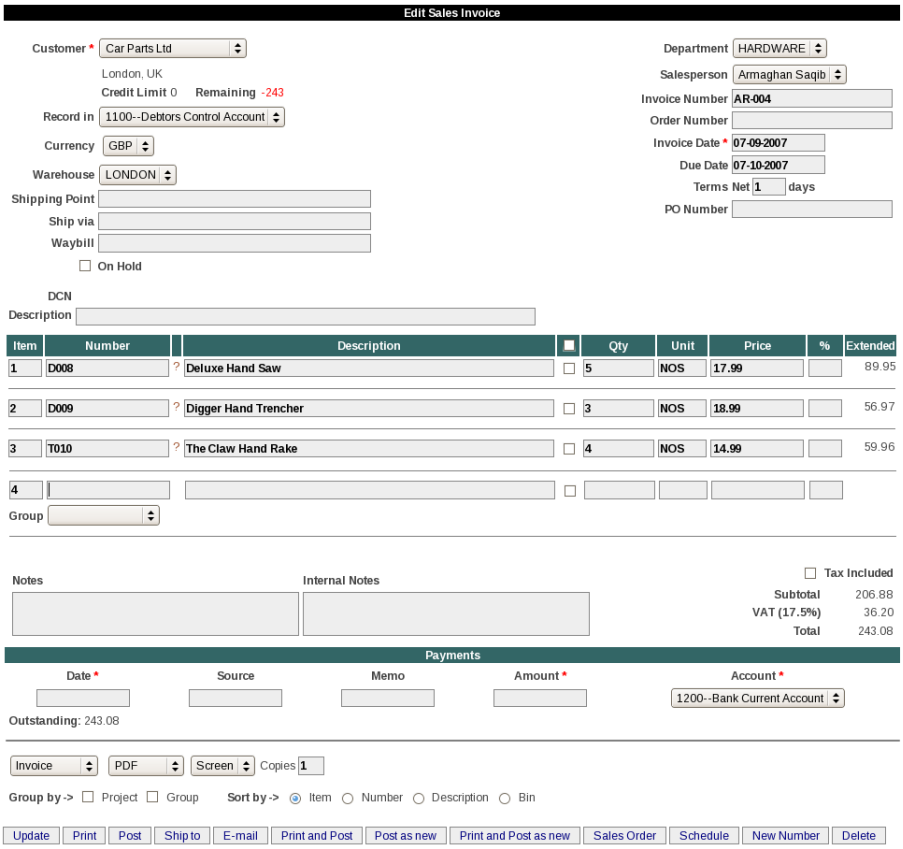
AR–Add Transaction menu option is used to create AR Transactions. These transactions allow you to record your sales in correct GL accounts without creating an invoice.



3.1.2 Sales Invoice

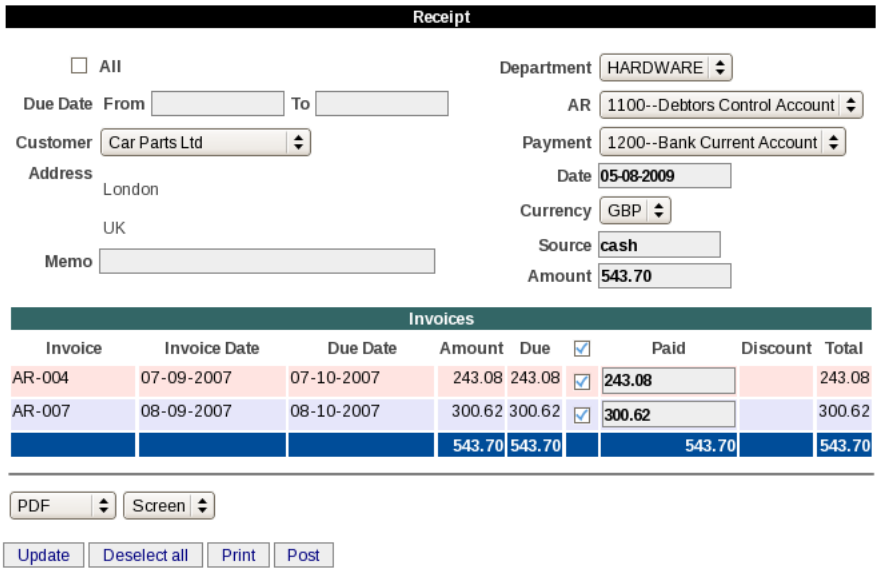
Sales invoices are created using AR–Sales Invoice menu option. The only mandatory columns are Customer and Invoice Date. Rest of the columns can be left blank.

Once you enter an item (part, service) and click 'Update', a new line opens. This way you can enter any number of items (parts, services etc.) in the detail portion of the invoice.



3.2 Receipts

You can record cash receipts from customer while creating invoices (for cash sales) or afterward using Cash–Receipt menu.



3.3 AP

3.4 Payments

3.5 General Ledger

3.6 Recurring Transactions

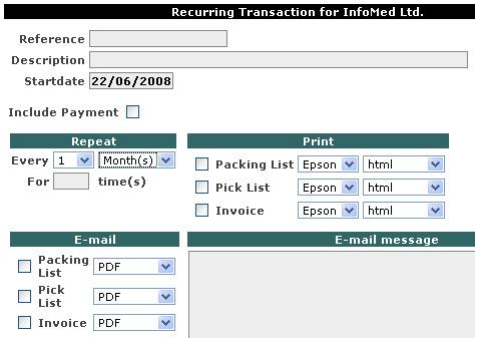
Recurring Transactions allow you to auto-generate pre-defined invoices, transactions and orders.

This feature can be used for the following:

1. Recurring billing to a customer (For rent, web hosting, school fee, installment etc.)
2. Recurring billing from your vendor
3. Monthly orders to your vendors or from your customers.
4. Monthly payroll posting using GL Recurring Transactions.
5. Month-end adjustments and allocations.

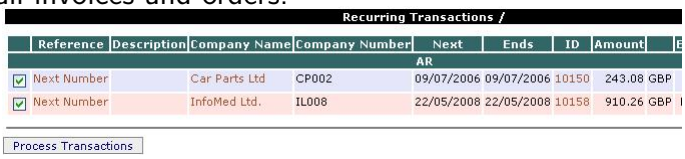
3.6.1 Scheduling

To generate the next number for a given transaction, leave the Next Number blank.



3.6.2 Generating

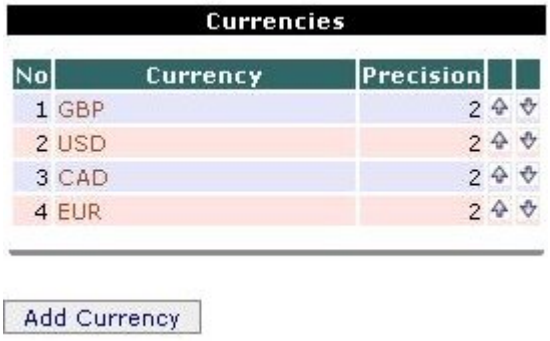
When recurring transactions are due you are reminded when you login to sql-ledger. With a single click you can generate all recurring transactions, print or email invoices and orders.



3.7 Exchange Rates

You can define and use multiple currencies in SQL-Ledger.

3.7.1 Defining currencies



3.7.2 Buying and selling in foreign currencies

3.7.3 Reports

AR Transactions /

Open

Date	Invoice	Description	Customer	Total		Paid		Curr
07-05-2007	AR-001		Auto Exchange Express	186.72	186.72			GBP
07-06-2007	AR-003		Big Porridge Ltd.	1,527.50	1,527.50	1,000.00	1,000.00	GBP
07-09-2007	AR-004		Car Parts Ltd	243.08	243.08			GBP
07-12-2007	AR-005		Electronics Ltd.	119.78	119.78			GBP
07-12-2007	AR-006		InfoMed Ltd.	317.11	317.11			GBP
12-09-2007	AR-007		InfoMed Ltd.	11.52	23.51			USD
				2,405.71		1,000.00		

3.7.4 Exchange rate difference

3.7.5 Funds transfers in foreign currencies

Let us say the exchange rate is 1 GBP = 2.0289 (or reverse 1 USD = 0.4929 GBP)

Add FX Adjustment

Reference | Date * 12-09-2007

Department

Description

Notes

Account	FX	Debit	Credit
1200--Bank Current Account - GBP			100.00
1201--Bank Current Account - USD		202.89	
1299--Foreign Currency Adjustments	x		102.89
0010--Freehold Property	☐		
		202.89	202.8

Update | Post | Schedule

3.8 Quotations

3.9 RFQ

3.10 Sales Order

3.11 Purchase Order

Here is the default work flow to use purchase orders.

1. Create a purchase order to inform vendor your intent to purchase goods.
2. To records the goods received, use Shipping–Receive.
3. Create a vendor invoice: Open the order and click the Vendor Invoice button. You can create invoice from a partially received order.

Note: An alternate work flow is also supported with some code changes (available as orders2 branch at github.com/ledger123). This allows you to partially/fully receive orders by editing them. 'Shipping-Receive' and 'Shipping-Ship' are not available in this branch.

3.11.1 Notes

Here are few points to remember:

1. When you create an invoice from order, you cannot edit the quantities on invoice screen or add or remove items.

2. When you create invoice from a partially received order, this order is marked closed and a new order with same number but remaining quantities and new order date is created.

When you are using the alternate workflow: (using orders2 code branch)

1. Stock onhand is increased when you save a PO with quantity in Rcvd column. No accounting entries are made. (COGS/expense, Vendor balances etc.)

2. You can create an invoice directly from PO by entering the qty received in Rcvd column and clicking the Vendor Invoice button. This automatically saves the order, updates stock and opens Add Vendor Invoice screen with information carried forward from the PO.

3.12 Shipping

Shipping module allows you to ship from and receive to warehouses from your orders. Here is the work flow to use the shipping module.

1. Create a sales or purchase order.

2. Ship/Receive this order from/to a warehouses.

3. Open the order and create invoice from it.

See also the documentation of orders entry module.

Shipping module serves the same purpose as putting the quantity in Ship or Recd column of a sales order or a purchase order but allows a different warehouse to be specified and maintain inventory quantities at warehouses.

Following paragraphs discuss the correct work flow to use the shipping module for purchases and sales.

3.12.1 Purchases

1. Create purchase orders for the inventory you want to purchase. If you do not specify a warehouse with order, you can receive the order to any warehouse.

2. Receive inventory using Shipping–Receive menu option. Select the desired warehouse during this process.

3. Create AP/Vendor invoice by opening the purchase order which has been received in the above step and clicking the Purchase Invoice button. Do not make any change to partnumber or quantity. Just click the Post button.

Note: When you create invoice from a partially received order, SL closes that order and creates a new one with the remaining order quantities but with same order number.

3.12.2 Sales

1. Create a sales order for the inventory you want to sell. If you do not specify a warehouse with order, you can ship the order from any warehouse.

2. Ship the order using Shipping–Ship. Shipping warehouse cannot be changed if you have specified one on the order.

3. Create AR/Customer invoice by opening the sales order which has been shipped in the above step and clicking the Sale Invoice button. Do not make any change to partnumber or quantity. Just click the Post button.

Note: When you create from a partially received order, SL closes that order and creates a new one with the remaining order quantities but with same order number.

3.12.3 Reports

Inventory onhand at warehouses:

1. Goods & Services–All Items report. Check the 'Warehouse' checkbox on search screen.

2. Warehouses–Reports–Onhand

Inventory receive/ship activity

1. Warehouses–Reports–Activity report.

3.12.4 Precautions

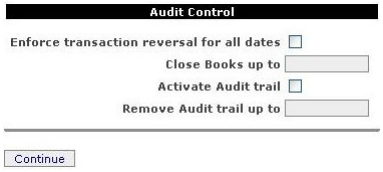
Do not do any of the following things when using the shipping module. It will make your inventory records incorrect.

1. Creating any new sale or purchase invoices directly (that is, without going through the order/ship/receive steps)
2. Editing any existing invoices.
3. Receiving purchase orders directly by putting the qty received in Recd column.
4. Shipping sales orders directly by putting the qty shipped in Ship column.

3.13 Time Cards

3.14 Audit Control

You can use System-Audit Control menu to enforce transaction control and log user activities.



Enforce transaction reversal for all dates

You can check this option to prevent any change to any transaction. You can however add a reverse transaction to correct some mistake. This option is highly recommended.

Close Books up to

When you close books upto a certain date, system does not allow changing any transaction prior to this date. Please note that this is not a year end process.

Activate Audit trail

All user activity (adding, changing, deleting transactions) is logged. You can view this log using 'Others–Audit Trial' report.

Remove Audit trail up to

You can use this option to remove audit trail from database up to a certain date. Useful to make your backups small.

Chapter 4

Keeping track of your business in SQL-Ledger

4.1 Income Statement

4.2 Balance Sheet

4.3 Trial Balance

4.4 Tax Report

4.5 Outstanding

4.6 Aging

4.7 Reconciliation

4.8 List Projects

4.9 Customer History

4.10 Vendor History

4.11 Year End

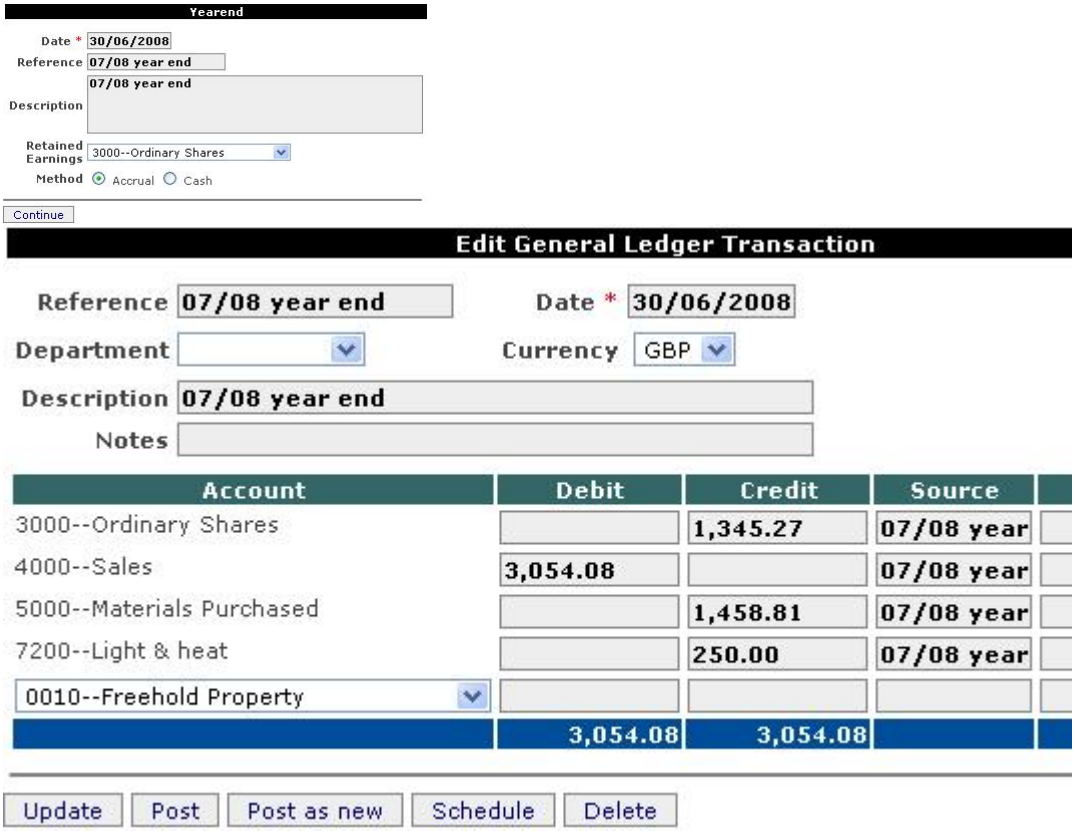
'System–Yearend' menu does the period closing in MyLedger. It creates a GL transaction which clears the income accounts and posts the difference (which is

income or loss) to the specified retained earnings account.

Please note that:

1. Year-end process can be run daily, weekly, monthly, quarterly or yearly.
2. Year-end GL transaction is not included in the income statement which covers period containing a closing transactions.
3. The year-end GL transaction can be viewed through GL reports and edited or deleted as required.

This is year end screen and the GL transaction created by year-end process.



4.12 Data backup

You can backup your data directly through sql-ledger. There are two ways to get your backup using the 'System–Backup' menu.

System–Backup–Send by Email

Backup is sent to your email address through email. You can add or change this email address through Preferences.

System–Backup–Save to File

When you click this menu option your browser will display the save file dialog and you can save backup file on your computer.

4.13 Basics of double-entry accounting system

4.13.1 Introduction

Double entry accounting system, although much feared by non-accountants, is a very simple but extremely powerful method of managing money.

SL does much of the double entry accounting itself linking all parts of the application through a chart of accounts. You need to know about double entry system only when you are going to make general ledger transactions. Basic Principle

Every business transaction affects at least two heads of accounts.

For example:

When you buy a car, you cash is decreased and your assets are increased. When you sell a item on cash, your sale is increased and your cash is also increased.

4.13.2 Account types

There are five basic types of accounts which are given below:

1. Assets

2. Liabilities
3. Capital
4. Sales
5. Expenses

4.13.3 Accounting rules

- Assets (1) and Expenses (5) are increased by debit and decreased by credit
- Liabilities (2), Capital (3) and Sales (4) are increased by credit and decreased by debit.

4.13.4 Examples

You invest $1000 to start a new business:

- Debit: Your bank account
- Credit: Capital account

You pay $100 check for office rent:

- Debit: Office rent expense account
- Credit: Your bank account

You build a website for a customer asking him to pay $200. Customer promises to pay after 20 days.

- Debit: Accounts Receivables (Debtors)
- Credit: Sales

Your customer pays you $200 after 20 days.

- Debit: Your bank account
- Credit: Accounts Receivables (Debtors)

Here is a really simple and useful accounting tutorial: http://www.a-systems.net/accounting.htm

4.14 Cost of Goods Sold (COGS)

Cost of Goods Sold (COGS) is the purchase price of the goods you just sold. Your sales minus the COGS is your gross profit. COGS is an important accounting information. Correct COGS gives you a clear picture of the profitability of your products.

Tip: To view the debit and credit accounting transactions for any sale or purchase invoice, enter the invoice number on General Ledger–Reports screen and click Continue button.

4.14.1 Sale invoices and COGS

Let us make it clear with an example:

You purchase 10 iPhones for $400 each.

- Debit: Inventory $4000
- Credit: AP $4000

A customer comes in and purchases 2 of these at $500 each.

- Debit: AR $1000 Credit: Sales $1000
- Debit: COGS $800 Credit: Inventory $800

So your gross profit is $200.

SQL-Ledger posts COGS automatically with each sale invoice. It calculates COGS on First-In First-Out (FIFO) basis. This means is that if you purchase 5 more iPhones at $430 each, MyLedger will keep calculating COGS @ $400 each until all 10 iPhones of first purchase transaction are depleted. Afterward it will calculate COGS @ $430.

4.14.2 Sales before purchases

SQL-Ledger allows you to sell goods without purchasing these in advance. This is a common practice in many businesses where you have received the goods but do not have the vendor invoice.

This results in negative stock quantity on Goods & Services–Reports–All Items report. No COGS is posted for such transactions at the time of sale. Later when you record purchases, COGS is automatically recorded for these oversold items.

4.14.3 Editing Sale Invoices

When you edit and repost an already posted sale invoice, COGS goes out of sync and incorrect accounting entries are posted. This causes incorrect income statement.

To confirm this, display your income statement and write down the COGS amount. Now open and repost any past sales invoice. Compare the new COGS in income statement with the old one.

Ideally you should never edit an invoice. Instead post a reversal of the invoice (using a credit invoice) and create a new invoice. Check the box Enforce transaction reversal for all dates on System–Audit Control screen.

If you do need to edit invoices, you can correct COGS transactions by running the re-posting of invoices through menu System–Repost COGS.

Chapter 5

Ledger Cart

5.1 Introduction

LedgerCart instantly creates an online store and order system using information in your SQL-Ledger. You just drop the cgi scripts into your webserver, install few cpan modules, configure your db connection and you are ready to go.

Users can browse products and services, add items to their cart and checkout in a familiar way. New order is added to SQL-ledger sales orders.

5.1.1 Features

1. Extemely simple to install and configure.
2. Can be installed on dedicated or shared hosting.
3. No additional database required. Retrieves and saves all data from/to sql-ledger dataset.
4. Easy to customize. All pages are standard html pages with template toolkit tokens.
5. Add new pages by creating standard html files and linking them in header.html or sidebar.html.
6. Look and feel can be customized using css and templates.
7. A single script 'index.pl' allows you to easily add more features by adding new actions.

8. Add item descriptions. These are displayed on product detail page and are stored in item notes. Item descriptions can use markdown syntax.
9. Add item images. LedgerCart automatically creates thumbnails and shows full image on item detail.
10. Visitors can now add items to their cart and checkout with their billing and shipping address.
11. New customers can register during checkout.
12. Existing customers can get a new password to their email using 'forgot password'. They can login with their email address and place orders.
13. Customers can browse their orders and invoices when logged-in.

5.1.2 Limitations

No payment gateways support yet.

5.1.3 Using LedgerCart as an online store

LedgerCart can instantly turn your SL installation into an online store with little or no effort. Customers can place order using the familiar shopping cart interface. Your existing customers can generate a new password using 'Forgot password' feature.

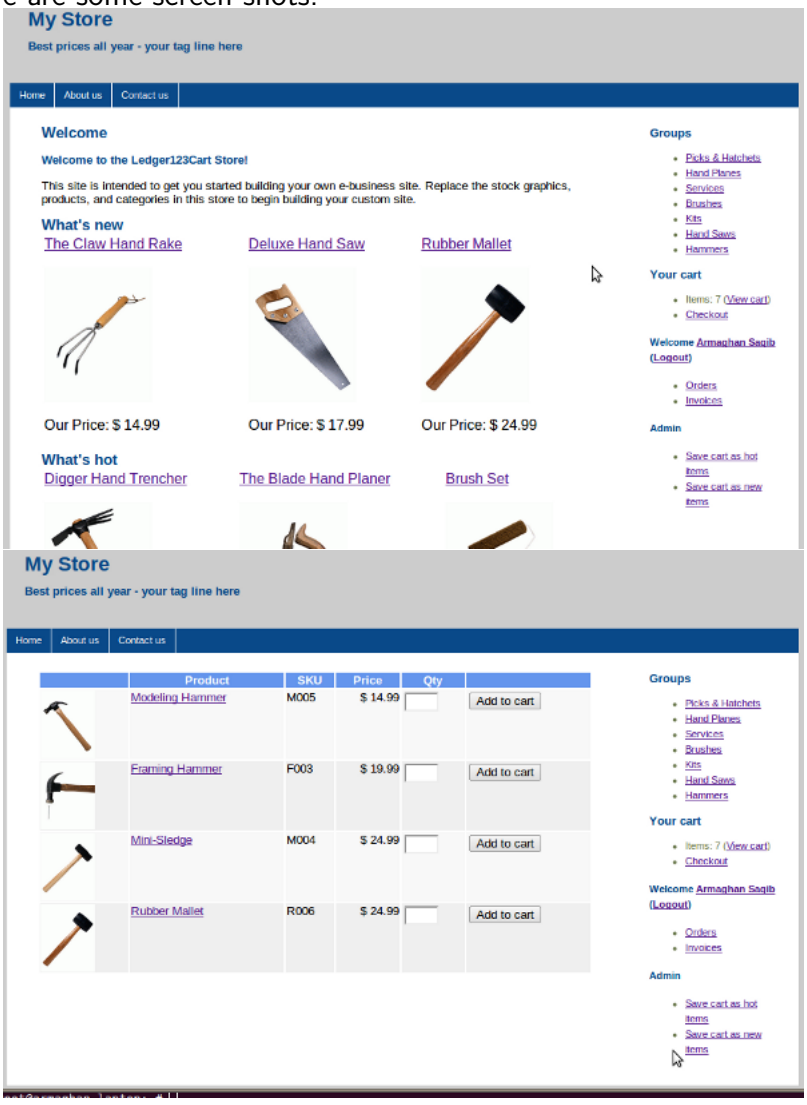
5.1.4 Using LedgerCart as Self service portal

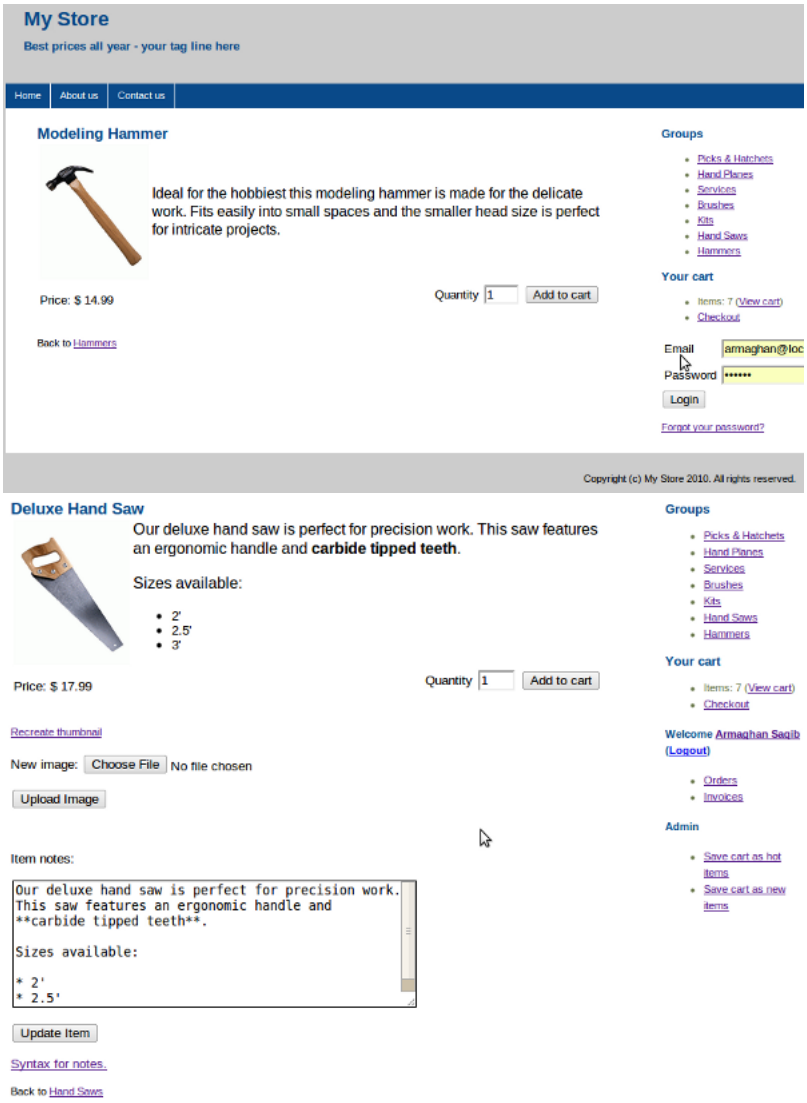
LedgerCart can be used to serve as a self-service internet portal just like the self-service internet banking. Your customers can view:

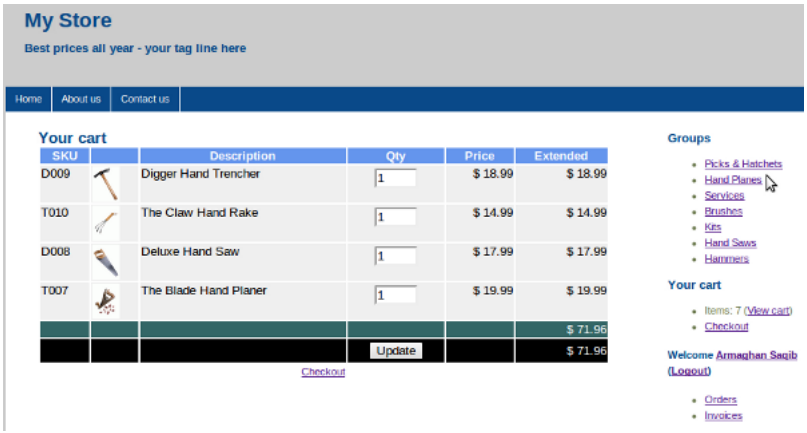
1. Their orders summary, order details and status
2. Invoices summary and details
3. Statements (payment summary and detail)

5.1.5 Screen shots

Here are some screen shots.




My Store
Best prices all year - your tag line here
Home
About us
Contact us
Modeling Hammer
Ideal for the hobbiest this modeling hammer is made for the delicate work. Fits easily into small spaces and the smaller head size is perfect for intricate projects.
Price: $ 14.99
Quantity 1
Add to cart
Back to Hammers
Groups
Picks & Hatchets
Hand Planes
Services
Brushes
Kits
Hand Saws
Hammers
Your cart
Items: 7 (View cart)
Checkout
Email armaghan@loc
Password
Login
Forgot your password?
Copyright (c) My Store 2010. All rights reserved.
Deluxe Hand Saw
Our deluxe hand saw is perfect for precision work. This saw features an ergonomic handle and carbide tipped teeth.
Sizes available:
2'
2.5'
3'
Price: $ 17.99
Quantity 1
Add to cart
Recreate thumbnail
New image: Choose File No file chosen
Upload Image
Item notes:
Our deluxe hand saw is perfect for precision work.
This saw features an ergonomic handle and
carbide tipped teeth.
Sizes available:
* 2'
* 2.5'
Update Item
Syntax for notes.
Back to Hand Saws
Welcome Armaghan Saqib (Logout)
Orders
Invoices
Admin
Save cart as hot items
Save cart as new items



5.2 Installation

5.2.1 Software packages

Login to the server with your user name and password. To be able to install the software, we have to change to the "root" account. In this way, we get administrator rights. Type:

```
su -
```

and enter your password.

With the following command, we install the packages we need for LedgerCart:

```
apt-get install libcgi-simple-perl libdbi-perl libtemplate-
    perl libobject-signature-perl libnumber-format-perl
    libmime-lite-perl libdbix-simple-perl libtext-markdown-
    perl libdate-calc-perl libgd-gd2-perl libdatetime-perl
    libhtml-format-perl apg
```

After that you need to install some further cpan modules:

```
cpan GD cpan GD::Thumbnail cpan MIME::Lite::TT::HTML
```

Then install LedgerCart in your SQL-Ledger directory:

```
git clone git://github.com/ledger123/ledgercart.git
    ledgercart
```

5.2.2 Configuration and Admin access

To configure LedgerCart for your installation, edit the config.pl file and change the appropriate lines for your database connection information. You can also change default thumbnail sizes here.

5.2.2.1 Admin User

To enable admin access, create a customer using SQL-Ledger with your email address and specify its id in $form{admin_id}. Now using "forgot password" link, generate a new password which will be sent to your email address.

5.2.2.2 Editing item descriptions, images and thumbnails

When you are logged in as admin and visit item detail pages, you can edit item descriptions as well as upload images and auto-create thumbnails.

Item descriptions text uses simple markup language 'markdown' for html elements. No html is allowed for security reasons. See http://daringfireball.net/projects/markdown/dingus for markdown syntax. Item descriptions are stored in item notes column and can be editing from within SQL-Ledger as well.

5.2.2.3 Editing pages through admin access

Once you login as admin, you can see 'Edit' links. Pages can be edited right away. You can use standard html and template toolkit tokens to edit pages.

5.2.2.4 Marking 'hot' and 'new' items

When you are logged in as admin, add items to your cart and click the 'Save cart as hot items' or 'Save cart as new items'. This will mark those items as hot or new and will display them on man page (in default templates). In future, hot/new functionality will be made to work based upon actual 'hot' or 'new' items.

5.2.3 Customization

LedgerCart is extremely easy to customize. LedgerCart consists of one big gateway script 'index.pl' which processes html templates created with Template::Toolkit.

1. Template::Toolkit templates are standard html files which can include Perl variables within [% and %] delimiters. You can copy the default templates and modify them as you please.

2. New pages can be added by creating standard html files and linking them to 'templatesfolder/header.html' or 'templatesfolder/sidebar.html'.

3. You can also customize the theme.css to change the colors and other look and feel according to your taste.

4. Expert users can modify the 'index.pl' file to add their own variables which can be interpolated within your LedgerCart templates.

Chapter 6

Development and Customization

6.1 Customization

SQL-Ledger can be customized in three ways:

6.1.1 custom_xx.pl files

You can create your own functions or override any existing function by creating custom scripts in custom_xx.pl files and putting them in bin/mozilla folder. For example, to add new functions to gl.pl file, add these functions to custom_gl.pl file and put this file into bin/mozilla/ folder. This file will be automatically loaded by sql-ledger before running any functions in gl.pl files.

Once your new functions are there, you can call them using your own custom menu. Custom menu entries are put in custom_menu.ini and follow the same syntax as that of menu.ini. This method of extending the sql-ledger is upgrade-safe and is the recommended way.

6.1.2 Custom Modules

You can build your own modules. To write a module, you need to create at least three files:

1. Module back-end code which will reside in ./sql-ledger/SL/MyModule.pm
2. Module front-end code which will reside in ./sql-ledger/bin/mozilla/mymodule.pl

3. Gateway script in ./sql-ledger. (You just need to make a copy of an existing one. For example cp gl.pl mymodule.pl in ./sql-ledger/ folder.

This method is also upgrade safe.

6.1.3 Modify the source code

Sometimes there is a need to directly alter the sql-ledger source code for particular needs. We have, for example, modified few reports (GL Transactions, All Items) in this way. Your changes, however, will be overwritten when you upgrade to new version and you will need to port these changes again to the new version.

A bit discipline and an SCM software like GIT can help manage such changes or patches with easy. We, at ledger123.com, use GIT to track and manage such changes across newer versions of sql-ledger.

6.2 SQL Queries

These sql queries for sql-ledger can be used in phpPgAdmin or psql.

6.2.1 Simple SQL Queries

6.2.1.1 Sales summary report

```
SELECT
        ar.invnumber,
        ar.transdate,
        c.name AS customer,
        ar.netamount,
        ar.amount - ar.netamount AS tax,
        ar.amount,
        ar.paid,
        ar.invoice
FROM ar
JOIN customer c ON (c.id = ar.customer_id);
```

6.2.1.2 Sales summary report with department and warehouse

```
SELECT
        ar.invnumber,
        ar.transdate,
        c.name AS customer,
        ar.netamount,
```

```
        ar.amount - ar.netamount AS tax,
        ar.amount,
        ar.paid,
        ar.invoice,
        d.description AS department,
        w.description AS warehouse
FROM ar
JOIN customer c ON (c.id = ar.customer_id)
JOIN department d ON (d.id = ar.department_id)
JOIN warehouse W ON (w.id = ar.warehouse_id);
```

6.2.1.3 Sales report with items

```
SELECT
        ar.invnumber,
        ar.transdate,
    c.name AS customer
        p.partnumber,
        ar.description,
        i.qty,
        i.sellprice,
        i.qty * i.sellprice AS extended
FROM ar
JOIN customer c ON (c.id = ar.customer_id)
JOIN invoice i ON (i.id = ar.trans_id);
```

6.2.1.4 List of customers

```
SELECT
        customernumber,
        name,
        creditlimit
FROM customer
WHERE LOWER(name) LIKE '%bank%'
ORDER BY name;
```

6.2.1.5 Cash accounts with current balances

```
SELECT
        accno,
```

```
        description,
        (
                SELECT SUM(amount) FROM acc_trans
                WHERE acc_trans.chart_id = chart.id
        ) AS balance
FROM chart
WHERE link LIKE '%_paid%';
```

6.2.1.6 Parts list

```
SELECT
        p.partnumber,
        pg.partsgroup,
        p.description,
        p.lastcost,
        p.rop,
        p.rop * p.lastcost AS reorder_amount
FROM parts p
JOIN partsgroup pg ON (pg.id = p.partsgroup_id)
WHERE inventory_accno_id IS NOT NULL
ORDER BY partnumber;
```

6.2.2 Advanced SQL Queries

6.2.2.1 Inventory onhand on specific date

```
SELECT
        p.partnumber,
        p.description,
        pg.partsgroup,
        p.unit,
        (
                SELECT SUM(0-i.qty) AS onhand
                FROM invoice i
                JOIN ap ON (ap.id = i.trans_id)
                WHERE ap.transdate <= '01-01-08' AND i.parts_id = p.id
        ) AS purchase,
        (
                SELECT SUM(i.qty) AS onhand
```

```
                FROM invoice i
                JOIN ar ON (ar.id = i.trans_id)
                WHERE ar.transdate <= '01-01-08'
                AND i.parts_id = p.id
        ) AS sale
FROM parts p
LEFT JOIN partsgroup pg
ON (pg.id = p.partsgroup_id);
```

6.2.2.2 Customer balances on a specific date

```
SELECT
        ct.id,
        ct.customernumber,
        ct.name,
        SUM(0 - ac.amount) AS balance
FROM customer ct
JOIN ar aa ON (ct.id = aa.customer_id)
JOIN acc_trans ac ON (aa.id = ac.trans_id)
JOIN chart c ON (c.id = ac.chart_id)
WHERE (ac.transdate <= '06-30-2007')
AND (c.link = 'AR')
GROUP BY 1,2,3
ORDER BY customernumber;
```

6.2.2.3 Sales summary by month

```
SELECT
        TO_CHAR(transdate, 'YY-MM') AS month,
        d.description AS department,
        SUM(netamount)
FROM ar
JOIN department d ON (d.id = ar.department_id)
WHERE (transdate BETWEEN '01.07.2005' AND '30.06.2006')
GROUP BY TO_CHAR(transdate, 'YY-MM'), d.description;
```

6.2.2.4 Sales Summary by group and month

```
SELECT
       d.description AS department,
       pg.partsgroup,
```

```
        TO_CHAR(ar.transdate, 'YY-MM') AS month,
        SUM(0 - i.qty * i.sellprice) AS amount
FROM invoice i
JOIN ar ON (ar.id = i.trans_id)
JOIN parts p ON (p.id = i.parts_id)
JOIN partsgroup pg ON (pg.id = p.partsgroup_id)
JOIN department d ON (d.id = ar.department_id)
WHERE ar.transdate BETWEEN '01.07.2005' AND '30.06.2006'
GROUP BY
        d.description,
        pg.partsgroup,
        TO_CHAR(ar.transdate, 'YY-MM')
ORDER BY 1, 2
```

6.2.2.5 Cash received today with age of AR in days

```
SELECT
        c.accno,
        c.description AS acc_title,
        d.description AS department,
        a.invnumber,
        ct.name,
        ac.transdate - a.transdate AS days,
        ac.source,
        ac.amount,
        e.name AS salesper,
        a.notes,
        ac.memo
FROM ar a
JOIN acc_trans ac ON (a.id = ac.trans_id)
JOIN chart c ON (ac.chart_id = c.id)
JOIN customer ct ON (a.customer_id = ct.id)
JOIN employee e ON (a.employee_id = e.id)
LEFT JOIN department d ON (d.id = a.department_id)
WHERE (ac.transdate = '30.05.06')
        AND(c.link LIKE '%AR_paid%')
        AND (
                a.department_id IN
                (SELECT id
                FROM department
                WHERE description IN ('LC','LS'))
        )
ORDER BY days;
```

6.2.2.6 Trial Balance with Month Headings

```
SELECT
        accno,
        description,
        (SELECT SUM(amount) FROM acc_trans ac   WHERE ac.chart_id = chart.id
               AND TO_CHAR(transdate, 'YY-MM') = '06-01') AS jan,
        (SELECT SUM(amount) FROM acc_trans ac   WHERE ac.chart_id = chart.id
               AND TO_CHAR(transdate, 'YY-MM') = '06-02') AS fab,
        (SELECT SUM(amount) FROM acc_trans ac   WHERE ac.chart_id = chart.id
               AND TO_CHAR(transdate, 'YY-MM') = '06-03') AS mar,
        (SELECT SUM(amount) FROM acc_trans ac   WHERE ac.chart_id = chart.id
               AND TO_CHAR(transdate, 'YY-MM') = '06-04') AS apr,
```

```
        (SELECT SUM(amount) FROM acc_trans ac   WHERE ac.chart_id = chart.id
               AND TO_CHAR(transdate, 'YY-MM') = '06-05') AS may,
        (SELECT SUM(amount) FROM acc_trans ac   WHERE ac.chart_id = chart.id
               AND TO_CHAR(transdate, 'YY-MM') = '06-06') AS jun,
        (SELECT SUM(amount) FROM acc_trans ac   WHERE ac.chart_id = chart.id
               AND TO_CHAR(transdate, 'YY-MM') = '05-07') AS jul,
        (SELECT SUM(amount) FROM acc_trans ac   WHERE ac.chart_id = chart.id
               AND TO_CHAR(transdate, 'YY-MM') = '05-08') AS aug,
        (SELECT SUM(amount) FROM acc_trans ac   WHERE ac.chart_id = chart.id
               AND TO_CHAR(transdate, 'YY-MM') = '05-09') AS sep,
        (SELECT SUM(amount) FROM acc_trans ac   WHERE ac.chart_id = chart.id
               AND TO_CHAR(transdate, 'YY-MM') = '05-10') AS oct,
        (SELECT SUM(amount) FROM acc_trans ac   WHERE ac.chart_id = chart.id
               AND TO_CHAR(transdate, 'YY-MM') = '05-11') AS nov,
        (SELECT SUM(amount) FROM acc_trans ac   WHERE ac.chart_id = chart.id
               AND TO_CHAR(transdate, 'YY-MM') = '05-12') AS dec,
FROM chart
WHERE charttype = 'A'
ORDER BY accno;
```

6.2.3 Queries to troubleshoot database problems

6.2.3.1 Transactions without departments

```
SELECT 'AR', id, invnumber AS reference, transdate
FROM ar
WHERE id NOT IN (SELECT DISTINCT trans_id FROM dpt_trans)
UNION ALL
SELECT 'AP', id, invnumber AS reference, transdate
FROM ap
WHERE id NOT IN (SELECT DISTINCT trans_id FROM dpt_trans)
UNION ALL
SELECT 'GL', id, reference, transdate
FROM gl
WHERE id NOT IN (SELECT DISTINCT trans_id FROM dpt_trans);
```

6.2.3.2 Unbalanced Journals

```
SELECT 'GL' AS mod, gl.reference, SUM(ac.amount)
FROM acc_trans ac
JOIN gl ON (gl.id = ac.trans_id)
GROUP BY 1, 2
HAVING SUM(ac.amount) <> 0
UNION ALL
SELECT 'AR' AS mod, ar.invnumber, SUM(ac.amount)
FROM acc_trans ac JOIN ar ON (ar.id = ac.trans_id)
```

```
GROUP BY 1, 2
HAVING SUM(ac.amount) <> 0
UNION ALL
SELECT 'AP' AS mod, ap.invnumber, SUM(ac.amount)
FROM acc_trans ac
JOIN ap ON (ap.id = ac.trans_id)
GROUP BY 1, 2  HAVING SUM(ac.amount) <> 0
ORDER BY 3
```

6.2.3.3 Orphan Transactions

```
SELECT *
FROM acc_trans
WHERE trans_id NOT IN (
        SELECT id FROM ar UNION ALL SELECT id FROM ap UNION ALL SELECT i
);
```

6.2.3.4 Correcting Assemblies Onhand

Due to a bug/gotcha in orders handling in official sql-ledger, parts onhand can go out of sync from actual transactions. Following query will help you find the correct onhand quantity for a given assembly.

```
SELECT 'Purchased', SUM(0-qty)  FROM invoice  WHERE parts_id = (SELECT id
    FROM parts WHERE partnumber='TW01') AND trans_id IN (SELECT id FROM ap)
UNION ALL
SELECT 'Sold', SUM(0-qty) FROM invoice WHERE parts_id IN (SELECT aid FROM
    assembly WHERE parts_id = (SELECT id FROM parts WHERE partnumber='TW01')
    ) AND trans_id IN (SELECT id FROM ar)
UNION ALL
SELECT 'Onhand', SUM(0-onhand) FROM parts WHERE id IN (SELECT aid FROM
    assembly WHERE parts_id = (SELECT id FROM parts WHERE partnumber='TW01')
    );
```

6.3 API

6.3.1 Introduction

SQL-Ledger allows you to call any of its functions from command line. An example will better illustrate this.

The following code run from your Linux/Unix shell will add a new customer to the customers table:

```
./ct.pl "
login=armaghan
&password=armaghan
&path=bin/mozilla
&db=customer
&action=save
&typeofcontact=company
&name=Ledger123
&firstname=Armaghan
&lastname=Saqib
&city=London
"
```

You could also insert this information using plain old SQL INSERT statement but here is the problem. Customer information is stored in at least three tables (customer, contact, address). You have to make sure you INSERT rows with correct id numbers in all three tables.

On the other hand API takes care of adding proper data rows in each tables with a single call like above. API also validates your data and runs any logic which is run when you are adding a customer through web interface. For example if you have defined a sequence for customer numbers, the next number is assigned automatically from that sequence.

6.3.2 API Uses

API can be used to "simulate" any sql-ledger function from command line. You can add customers, vendors, parts as well as any type of transaction (invoices, cash receipts and payments etc.)

This makes it very easy to integrate sql-ledger with any other application. For example you can integrate it with your CRM solution, POS system, or e-commerce solutions like AgoraCart or Interchange.

API also allows you to add new data entry interfaces with ease. All you need to develop is the code which will interact with users and leave the rest to the API.

Import invoices and payment functions built in new versions of sql-ledger are in fact "newer interfaces" built using the API.

6.3.3 Calling from PHP

You can make API calls from any language using its shell execution mechnisim. For example you can use the following php code to make SL api call.

```
<?php
$module = './ct.pl';
$params = 'login=armaghan';
$params .= '&password=armaghan';
$params .= '&path=bin/mozilla';
$params .= '&db=customer';
$params .= '&action=save';
$params .= '&typeofcontact=company';
$params .= '&name=Ledger123';
$params .= '&firstname=Armaghan';
$params .= '&lastname=Saqib';
$params .= '&city=London';
$output = shell_exec("$module \"$params\"");
echo "<pre>$output</pre>";
?>
```

asdfasdf