[XTIMECARD]

A SOFTWARE FOR TIME BILLING AND PROJECT MANAGEMENT

Yeshi Silvano Namkhai
Daria Mikhaylova

Last modified 07/10/14 with revision 1485

48 pages, Statistics paragraphs, 7879 words, Statistics tables, Statistics graphs.

Created on Sun, Jun 16, 13 by YN.
Last change Thu, Jul 10, 14 by Yeshi Silvano Namkhai with revision 1485.

Yeshi Silvano Namkhai
Daria Mikhaylova

/export/hda3/borglet/local_ram_fs_dirs/13.prod.changeling-worker-libreoffice.apps-docs-changeling-worker-libreoffice.1325611248271.549eedb1e9d07f8c/ramdisk/dird5d7d305e0e38be2c5c78070955458/filed5d7d305e0e38be2c5c78070955458.odt

TABLE OF CONTENTS

This manual guides the user to successful time billing and project management using LibreOffice Base, the software **xTimeCard** was developed by HEXAGON for its own activity.

PURPOSE

This application does not pretend to fulfill accounting requirements, it helps IT or business consulting firms to manage their projects and bill effectively by week or month; in other words for a managers and project managers to be paid for their team's work and achievements.

SIMPLY

This tutorial does not pretend to explain time billing in details, rather aims to introduce the topic and show how can be easily approached by our software. As a matter of fact **xTimeCard** simplifies tracking of work, control over budget and preparation of bills and reports for customer.

CLARIFY

Any project manager face the initial difficulty of getting the structure of their project properly balanced between budget, tasks and WBSs. **xTimeCard** helps to have done quickly and clearly.

LEARN AND CUSTOMIZE

With this software you received this tutorial that contains information and step-by-step guide on how to create this software, configure and use successfully. Moreover HEXAGON provides support to customize for your specific needs as part the learning process, as a matter of fact LibreOffice Base was chosen as technology to allow both developers and users to customize for specif needs.

SCALE

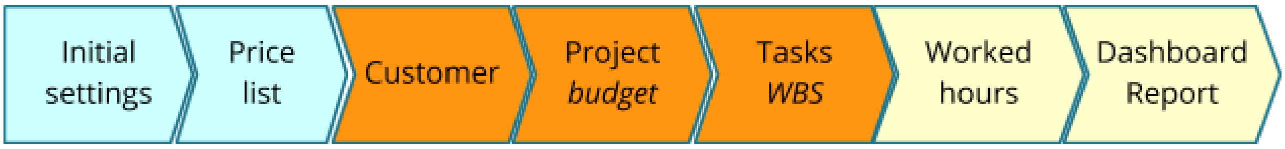
The advanced section of this tutorial also describe how to move from a personal use to a multi-user environment, for instance integrating to a largely used database such as MySQL or connecting to a workflow based on cloud technologies like Google Apps.

ORGANIZATION

As you may expect **xTimeCard** distinguishes between:

- settings
- master data
- operations

Settings are needed for the software to know more about you, while master data refer to customers and projects and they do not change, finally operations record your work and allow to account and be paid for.



INITIAL SETTINGS

To get started with **xTimeCard** you need at least to have your employee and the price of your activities to be paid clearly defined. For the sake of simplicity **xTimeCard** do not feature extensive information like resources, call/info center and do not consider discounts or customer's price lists, but these optional features may be a good reason for customization.

MASTER DATA

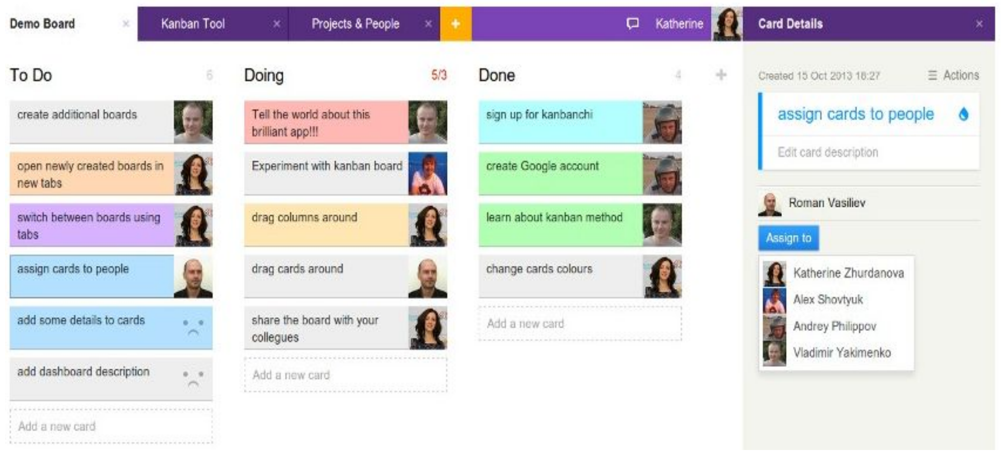
The essential information to properly run **xTimeCard** are the projects your team performs for your customer and the customer himself. Generally speaking any project has a well defined project plan, which is a task list, milestones, assigned resources (human by hourly rate or material), time frame, risk and critical path.

For this tutorial we work the project schedule and Gantt chart using Google spreadsheet and the add-on called ProjectSheet pretending to be a well organized team that work in real-time. Some of competencies are missing in our project therefore few collaborators are to be considered external employee/consultant, therefore the hypothetical project manager will be looking after as they bill us their worked time. For more info review the project template provided for this tutorial.

OPERATIONS

Every activity is recorded by the team member or the project manager in **xTimeCard** at the end of the day, therefore everyday week-end a delta comparison of actual over budget is available to the project manager and this may lead to report and bill the customer.

In addition to project management tools for this tutorial we use a dashboard that track progress, communicate objectives and manage changes between member all members, this is a Google add-on called Kanbanchi.



For more info review the card template provided for this tutorial.

FEATURES

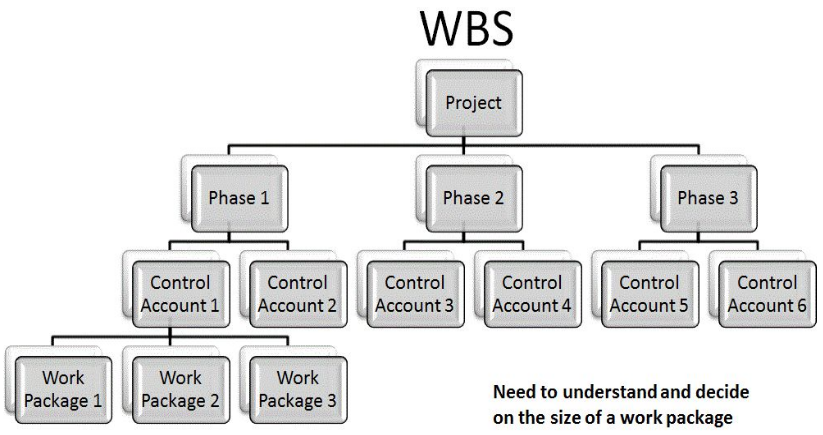
However this software was meant for educational purpose, allows a project manager to perform tracking over employees or external consultants within the project and prepare for billing or control.

DASHBOARD

When you first run **xTimeCard**, double click on “Dashboard” and look the performance of your business. Every information is available and dynamically updated while you select between customers and projects.

PROJECT CONTROL

Within **xTimeCard** a project is identified by a name and a full description, is organized by phases or stages and comprehends a task list. A task is identified by name and description, by an estimated effort and cost (which gives the budget of the project) and a WBS code.



The ability to aggregate more tasks by the WBS code or milestone allow to account the project properly and evaluate the progress from a delta budget point of view, which is not easy to track within project management software or tools.

Project Management Institute, A Guide to the Project Management Body of Knowledge, (PMBOK® Guide) – Fifth Edition, Project Management Institute Inc., 2013 Page 126

WBS approach breaks the work of the project down into progressively more detail, at the lowest level of *work package* can be properly and reliably scheduled, estimated, monitored and controlled.

TRACKING

Every team member will record their worked time and assign properly to project and task, for each project the WBS view will track delta over budget dynamically. As you may expect key performance of earned value management may be compared to reality as you track your team's work with **xTimeCard**. The creation of reports with KPIs like SPI and CPI compared to results and achievements may be a reason for meaningful customization of **xTimeCard**.

BILLING

To account a project means to evaluate WBSs, generally speaking software like Microsoft Project allow only to track estimate progress therefore to control actual value over budget; does not allow to verify your team's work, achievements and to eventually bill and get paid. With **xTimeCard** you may easily bill the work done and justify it over your project plan.

LIBREOFFICE

LibreOffice is a freely available, fully-featured office productivity suite. Its native file format is Open Document Format (ODF), an open standard format that is being adopted by governments worldwide as a required file format for publishing and accepting documents. LibreOffice can also open and save documents in many other formats, including those used by several versions of Microsoft Office.

LibreOffice includes the components. Learn more from the chapter 1 of the book "Getting Started with LibreOffice 4.2" .

WRITER (WORD PROCESSOR)

Writer is a feature-rich tool for creating letters, books, reports, newsletters, brochures, and other documents. You can insert graphics and objects from other components into Writer documents. Writer can export files to HTML, XHTML, XML, Adobe Portable Document Format (PDF), and several versions of Microsoft Word files. It also connects to your email client.

CALC (SPREADSHEET)

Calc has all of the advanced analysis, charting, and decision making features expected from a high-end spreadsheet. It includes over 300 functions for financial, statistical, and mathematical operations, among others. The Scenario Manager provides "what if" analysis. Calc generates 2D and 3D charts, which can be integrated into other LibreOffice documents. You can also open and work with Microsoft Excel workbooks and save them in Excel format. Calc can also export spreadsheets in several formats, including for example Comma Separated Value (CSV), Adobe PDF and HTML formats.

IMPRESS (PRESENTATIONS)

Impress provides all the common multimedia presentation tools, such as special effects, animation, and drawing tools. It is integrated with the advanced graphics capabilities of LibreOffice Draw and Math components. Slideshows can be further enhanced using Fontwork special effects text, as well as sound and video clips. Impress is compatible with Microsoft PowerPoint file format and can also save your work in numerous graphics formats, including Macromedia Flash (SWF).

DRAW (VECTOR GRAPHICS)

Draw is a vector drawing tool that can produce everything from simple diagrams or flowcharts to 3D artwork. Its Smart Connectors feature allows you to define your own connection points. You can use Draw to create drawings for use in any of the LibreOffice components, and you can create your own clip art then add it to the Gallery. Draw can import graphics from many common formats and save them in over 20 formats, including PNG, HTML, PDF, and Flash.

BASE (DATABASE)

Base provides tools for day-to-day database work within a simple interface. It can create and edit forms, reports, queries, tables, views, and relations, so that managing a relational database is much the same as in other popular database applications. Base provides many new features, such as the ability to analyze and edit relationships from a diagram view. Base incorporates HSQLDB as its default relational database engine. It can also use dBASE, Microsoft Access, MySQL, or Oracle, or any ODBC compliant or JDBC compliant database. Base also provides support for a subset of ANSI-92 SQL.

MATH (FORMULA EDITOR)

Math is the LibreOffice formula or equation editor. You can use it to create complex equations that include symbols or characters not available in standard font sets. While it is most commonly used to create formulae in other documents, such as Writer and Impress files, Math can also work as a standalone tool. You can save formulae in the standard Mathematical Markup Language (MathML) format for inclusion in web pages and other documents not created by LibreOffice.

BEGINNER

In the beginner section of the **xTimeCard** tutorial we review the basics of LibreOffice Base and we build the skeleton. A working knowledge of LibreOffice or any office automation software is suggested.

DOWNLOAD AND INSTALL

For this tutorial you need to install LibreOffice 4.2 or superior, you may use previous version but is not advisable because require rework of extensions and additional patches. Download from LibreOffice website and install for your operating system. Learn more about setuping up LibreOffice from chapter 2 of “Getting Started with LibreOffice 4.2”.

OFFLINE HELP

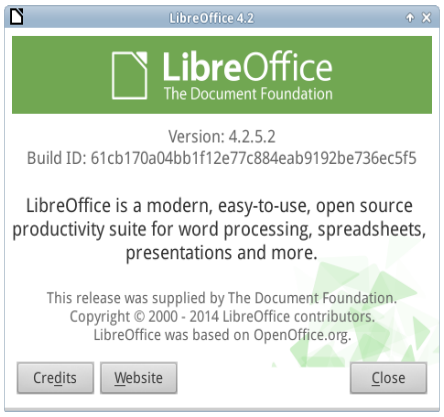
Add also a help in your favorite language, which comes handy while learning Base.

ACCESS2BASE

The main extension, contained in any version superior to LibreOffice 4.2, to code and customize **xTimeCard** is called Access2Base, which allows advanced user and developers to code using Microsoft Access simplified basic. If eventually you prefer to use OpenOffice, please download and install also the extension; follow instructions.

VERSION

If you already have LibreOffice installed, please check your version go in Help→About LibreOffice, eventually update or add extension following instructions.



Version: 4.2.5.2
Build ID: 61cb170a04bb1f12e77c884eab9192be736ec5f5

LibreOffice is a modern, easy-to-use, open source productivity suite for word processing, spreadsheets, presentations and more.

This release was supplied by The Document Foundation.
Copyright © 2000 - 2014 LibreOffice contributors.
LibreOffice was based on OpenOffice.org.

JAVA

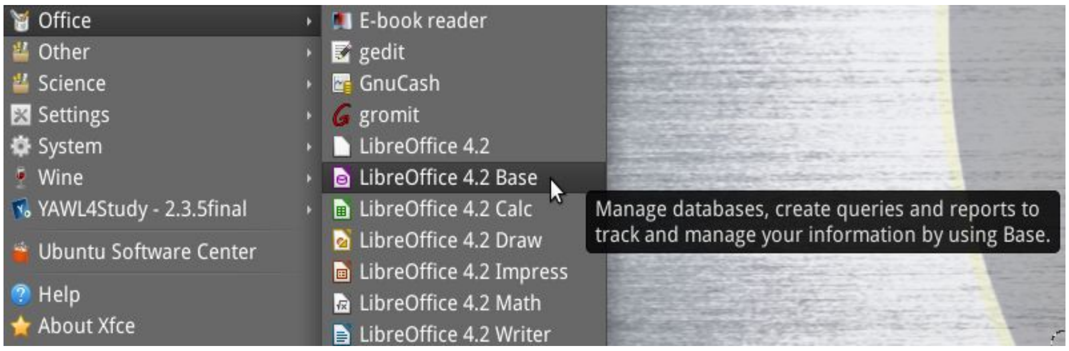
For this tutorial you will need to work with JAVA only in the advanced section, therefore for now you do not need to worry about.

START LIBREOFFICE BASE

Many people use office automation software but not everyone has experience with open-source and most of all with database concepts. In beginner section of this tutorial we will review the essential information to create the skeleton of **xTimeCard**.

RUN

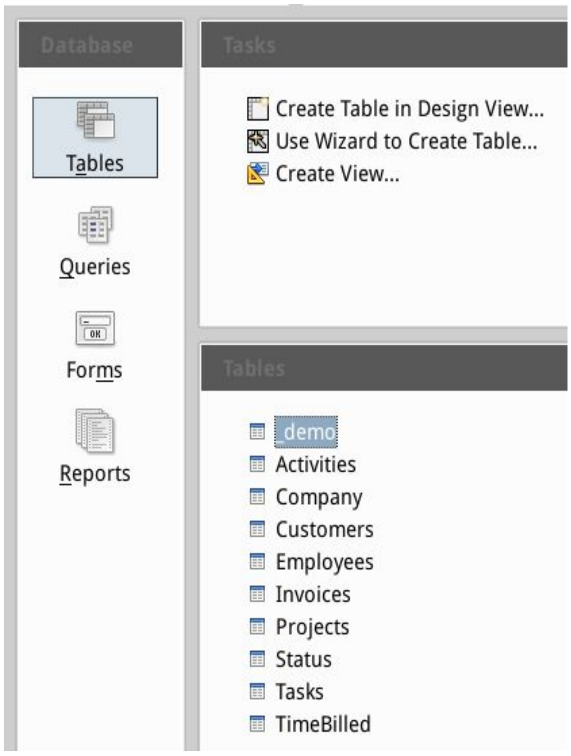
Now is the time to open LibreOffice Base and learn how to create a table, fill in data, query for results and filter. Go to your main menu and run LibreOffice Base. HEXAGON promotes open-source therefore pictures will be taken from Linux, for this tutorial UbuntuStudio version 12.04.



Manage databases, create queries and reports to track and manage your information by using Base.

VIEWS

A LibreOffice Base file contains a database, which can be flat or relational; it contains four objects: tables, queries, forms and reports. Each object has view, you may select from the menu or the left bar, every view has tasks on the right and objects in the bottom right.



SAMPLE DATA

Before filling our own data, you may try out our sample database and learn how xTimeCard works. This is a complete set of data with masters and transactions, we provide now a short description of characters which come as courtesy of Matt Groening and the Fox Broadcasting Company.

EMPLOYEES

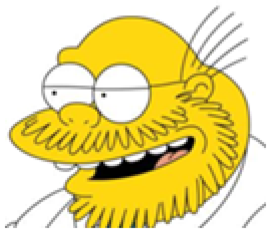
The owner of the company is Artie Ziff (first boy friend of Margie), a successful business and his main activity is software development.

Artie Ziff, male, age 35, software developer and owner of Ziffcorp *and for this tutorial user of xTimeCard*. His email is artieboss@ziffcorp.com, taxcode 9900770088. His address is 901 North Grand, Springfield, IL 62702 with phone number +1(217)522-6670. http://www.ziffcorp.com

Jonathan Frink, male, age 42, professor *and for this tutorial R&D for Ziffcorp*. His email is proff@ziffcorp.com , taxcode 3333222277. His address is 3035 S Chatman Rd, Springfield, IL 62704 with phone number +1(217)546-9136. http://rd.ziffcorp.com

Chloe Talbot, female, age, a successful journalist *and for this tutorial freelance PR for Ziffcorp*. Her email is chloet@freepress.org , taxcode 5544455511. Her address is 3151 Dirksen Parkway, Springfield, IL 62703 with phone number +1(217)529-1928. http://chloet.freepress.org

CUSTOMERS



Dr. Marvin Monroe, male, age 45, psychiatrist, owns a private clinic in Springfield. His email is drmarvin@mclinic.com, taxcode: 1100220033. His address is 3208 CLEAR LAKE, Springfield, IL, 62702 with phone number +1(217)744-3550. http://www.mclinic.com

Kirk Van Houten, male, age 42, middle management of 'Southern Cracker'. His email is kvh@scracker.com , taxcode 3322779955. His address is 1825 South MacArthur Blvd., Springfield, IL, 62704 with phone number +1(217)546-5323. http://www.scracker.com

Howard K. Duff VIII, male, age over 50', owner of Duff Beer (stadium and president of Springfield Isotopes American football team). His email is hk8@duff.com , taxcode 8888811111. His address is 3420 Freedom Drive, Springfield, IL, 62703 with phone number +1(217)793-1305. http://www.duff-stadium.com

PROJECTS

Each customer ask Ziffcorp to develop a customized solution to their business, therefore our sample data features three main projects. For Dr. Monroe a management software for their clinic based on mobile technology, while for V.Houten an implementation of open-source ERP (Enterprise Resource Planning) software to improve their production plants and sales, finally for Duff's Beer design of website, billboards and new marketing campaign.

TASKS AND WBS

Some projects of Ziffcorp are requested by their customers against a defined budget, therefore in this tutorial we may evaluate performance with printed reports as expected. Both numeric and literal approach to coding WBS are contained in the sample data to show the pro/cons.

DOCUMENTATION

To create the skeleton of **xTimeCard**, you need to be familiar with the basics of LibreOffice Base and the underlying concept of relational database; to move further and build the logic (intermediate) and customize (advanced) **xTimeCard**, you need to be an advanced user of LibreOffice Base and make your hands dirty BASIC programming and scripting.

GETTING STARTED GUIDE

Please review chapter 8 of “Getting Started with LibreOffice 4.2”, if you feel familiar with the explanations and exercises move ahead and create the tables for **xTimeCard**.

Otherwise follow step-by-step the tutorial contained in the guide and learn the basics. For your convenience the sample database is available for download.

BASE HANDBOOK

LibreOffice Base has an extensive guide for people who understand the basics and wish for in-depth discussion called Base Handbook, for this tutorial you may need this knowledge to customize **xTimeCard** therefore only for intermediate and advanced sections. For your convenience the sample database described in the book is available for download.

MACRO

A macro is a saved sequence of commands or keystrokes that are stored for later use. The LibreOffice macro language is very flexible, allowing automation of both simple and complex tasks. Macros are very useful when you have to repeat the same task in the same way over and over again.

LibreOffice macros are usually written in a language called LibreOffice Basic, sometimes abbreviated to OOoBasic (deriving from OpenOffice.org Basic) or Basic. Although you can learn LibreOffice Basic and write macros, there is a steep learning curve to writing macros from scratch. The best reference for OooBASIC is the book OpenOffice.org Macros Explained written by Andrew Pitonyak.

The usual methods for a beginner are to use macros that someone else has written or use the built-in macro recorder, which records keystrokes and saves them for use. Most tasks in LibreOffice are accomplished by “dispatching a command” (sending a command), which is intercepted and used.

Please review chapter 13 of getting started guide to learn more on macro recorder.

OOOBASIC

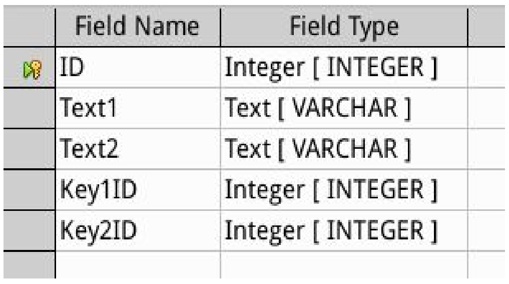
Originally developed within OpenOffice and derived in LibreOffice, OOoBASIC macro language is inspired by Microsoft VBA (Visual Basic for Applications) macro language integrated in Microsoft Office suite of applications. The best reference is the book https://wiki.openoffice.org/w/images/c/c1/BasicGuide_OOo3.2.0.pdf which fully introduces the programming language and the helps VBA programmers to move to OooBASIC.

TABLES

Create a testing table and the other 9 tables for the **xTimeCard** application that will be built with this tutorial.

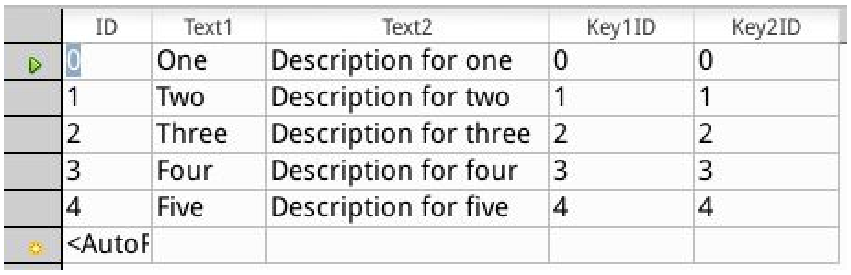
_TEST

First of all create a table called "_test", add an auto-value primary key of length 10, two



Field Name	Field Type
ID	Integer [INTEGER]
Text1	Text [VARCHAR]
Text2	Text [VARCHAR]
Key1ID	Integer [INTEGER]
Key2ID	Integer [INTEGER]

text fields of length 50 and two more numeric fields as integer of length 10.



ID	Text1	Text2	Key1ID	Key2ID
0	One	Description for one	0	0
1	Two	Description for two	1	1
2	Three	Description for three	2	2
3	Four	Description for four	3	3
4	Five	Description for five	4	4
<AutoF				

May be created also with this SQL script:

```
CREATE CACHED TABLE "_test"("ID" INTEGER NOT NULL PRIMARY KEY,"Text1"
VARCHAR(50),"Text2" VARCHAR(50),"Key1ID" INTEGER,"Key2ID" INTEGER)

INSERT INTO "_test" VALUES(0,'One','Description for one',0,0)
INSERT INTO "_test" VALUES(1,'Two','Description for two',1,1)
INSERT INTO "_test" VALUES(2,'Three','Description for three',2,2)
INSERT INTO "_test" VALUES(3,'Four','Description for four',3,3)
INSERT INTO "_test" VALUES(4,'Five','Description for five',4,4)
```

ACTIVITIES

This table is a master data, has the purpose to keep information about the work activity performed, has a key, a description and a billing rate. It is connected to master data of tasks and transactions of worked time. It may be created also with this SQL script:

```
CREATE CACHED TABLE "Activities"("ActivityID" INTEGER GENERATED BY DEFAULT AS
IDENTITY(START WITH 0) NOT NULL PRIMARY KEY,"Description" VARCHAR(200) NOT
NULL,"Rate" DECIMAL(10,2) NOT NULL)
```

STATUS

This table is a master data, has the purpose to describe the roadmap of projects, which is meant to be standard. It is also used by tasks to identify the belonging to project's phase. It is obviously connected to master data of projects and tasks. It may be created also with this SQL script:

```
CREATE CACHED TABLE "Status"("Status" TINYINT NOT NULL PRIMARY KEY,"Name"
VARCHAR(20),"Description" VARCHAR(200))
```

INVOICES

This table is a transaction data, has the purpose to bill weekly our customers by using a form that allow to select worked hours by week and sign as billed. It contains information of the company (= first record of Employee table), invoice date, payment date, invoice number, and invoice values. Once an invoice has been created, scan and record it. It may be created also with this SQL script:

```
CREATE CACHED TABLE "Invoices"("InvoiceID" INTEGER GENERATED BY DEFAULT AS
IDENTITY(START WITH 0) NOT NULL PRIMARY KEY,"CustomerID"
INTEGER,"EmployeeID" INTEGER,"Number" VARCHAR(50),"Date" DATE,"DueDate"
DATE,"NetValue" DECIMAL(10,2),"Taxes" DECIMAL(10,2),"Total" DECIMAL(10,2),"Note"
VARCHAR(100),"Scan" LONGVARBINARY)

ALTER TABLE "Invoices" ADD CONSTRAINT SYS_FK_510 FOREIGN KEY("CustomerID")
REFERENCES "Customers"("CustomerID")
ALTER TABLE "Invoices" ADD CONSTRAINT SYS_FK_522 FOREIGN KEY("EmployeeID")
REFERENCES "Employees"("EmployeeID")
ALTER TABLE "Projects" ADD CONSTRAINT SYS_FK_507 FOREIGN KEY("CustomerID")
REFERENCES "Customers"("CustomerID")
ALTER TABLE "WorkedHrs" ADD CONSTRAINT SYS_FK_525 FOREIGN KEY("EmployeeID")
REFERENCES "Employees"("EmployeeID")
```

PROJECTS

This table is a master data, has the purpose to keep information about projects and is directly connected to master data of Customers. It contains information of the motivation, begin and end dates, customer's order and actual state. It connects with transaction data of worked hours. It may be created also with this SQL script:

```
CREATE CACHED TABLE "Projects"("ProjectID" INTEGER GENERATED BY DEFAULT AS
IDENTITY(START WITH 0) NOT NULL PRIMARY KEY,"CustomerID" INTEGER,"Name"
VARCHAR(50),"Begin" DATE,"End" DATE,"Description" LONGVARCHAR,"Status"
TINYINT,"Order" VARCHAR(50),CONSTRAINT SYS_FK_398 FOREIGN KEY("Status")
REFERENCES "Status"("Status")) .
```

TASKS

This table is a master data, has the purpose to keep information about tasks within a project, therefore directly connected to master data of Projects. It contains information of the motivation, begin and end dates, customer's order and actual state. It may be created also with this SQL script:

```
CREATE CACHED TABLE "Tasks"("TaskID" INTEGER GENERATED BY DEFAULT AS
IDENTITY(START WITH 0) NOT NULL PRIMARY KEY,"ProjectID" INTEGER NOT
NULL,"Description" VARCHAR(200),"EstHrs" DECIMAL(6),"EstCost"
DECIMAL(10,2),"WBS" VARCHAR(50) NOT NULL,"Status" TINYINT,CONSTRAINT
SYS_FK_412 FOREIGN KEY("ProjectID") REFERENCES
"Projects"("ProjectID"),CONSTRAINT SYS_FK_415 FOREIGN KEY("Status") REFERENCES
"Status"("Status"))
```

WORKEDHRS

This table is a transaction data, has the purpose to record daily work. It contains information of the employee, the activity performed and the matrix the project/task/wbs. Once an invoice is created, the single worked hour will be connected the the invoice id. It may be created also with this SQL script:

```
CREATE CACHED TABLE "WorkedHrs"("WorkedHrID" INTEGER GENERATED BY
DEFAULT AS IDENTITY(START WITH 0) NOT NULL PRIMARY KEY,"EmployeeID"
INTEGER,"ProjectID" INTEGER,"TaskID" INTEGER,"ActivityID" INTEGER NOT
NULL,"InvoiceID" INTEGER,"Date" DATE,"Hours" DECIMAL(6),"Notes"
LONGVARCHAR,CONSTRAINT SYS_FK_449 FOREIGN KEY("ActivityID") REFERENCES
"Activities"("ActivityID"),CONSTRAINT SYS_FK_452 FOREIGN KEY("ProjectID")
REFERENCES "Projects"("ProjectID"),CONSTRAINT SYS_FK_455 FOREIGN KEY("TaskID")
REFERENCES "Tasks"("TaskID"),CONSTRAINT SYS_FK_458 FOREIGN KEY("InvoiceID")
REFERENCES "Invoices"("InvoiceID"))
```

CUSTOMERS

This table is a master data, has the purpose to keep information of our customers. It may be created also with this SQL script:

```
CREATE CACHED TABLE "Customers"("CustomerID" INTEGER GENERATED BY
DEFAULT AS IDENTITY(START WITH 0) NOT NULL PRIMARY KEY,"Company"
VARCHAR(50),"FirstName" VARCHAR(50),"LastName" VARCHAR(50),"Address"
VARCHAR(100),"City" VARCHAR(50),"CountryID" INTEGER,"Phone"
VARCHAR(30),"Email" VARCHAR(50),"Web" VARCHAR(200),"TaxCode"
VARCHAR(50),"Notes" LONGVARCHAR,"Picture" LONGVARBINARY,CONSTRAINT
SYS_FK_513 FOREIGN KEY("CountryID") REFERENCES "Countries"("CountryID"))
```

EMPLOYEES

This table is a master data, has the purpose to keep information of our selves. The first record is to be considered as the company issuing invoices. It may be created also with this SQL script:

```
CREATE CACHED TABLE "Employees"("EmployeeID" INTEGER GENERATED BY
DEFAULT AS IDENTITY(START WITH 0) NOT NULL PRIMARY KEY,"Company"
VARCHAR(50),"FirstName" VARCHAR(50),"LastName" VARCHAR(50),"Address"
VARCHAR(100),"City" VARCHAR(50),"CountryID" INTEGER,"Phone"
VARCHAR(30),"Email" VARCHAR(50),"Web" VARCHAR(200),"TaxCode"
VARCHAR(50),"Notes" LONGVARCHAR,"Picture" LONGVARBINARY)
```

COUNTRIES

This table is a master data, has the purpose to allow selection of country while filling customer and employees forms. It may be created also with this SQL script:

```
CREATE CACHED TABLE "Countries"("CountryID" INTEGER NOT NULL PRIMARY
KEY,"Country" VARCHAR(100))
```

For your convenience, countries may be filled with this SQL script:

```
INSERT INTO "Countries" VALUES(0,'China')
INSERT INTO "Countries" VALUES(1,'India')
INSERT INTO "Countries" VALUES(2,'United States')
INSERT INTO "Countries" VALUES(3,'Indonesia')
INSERT INTO "Countries" VALUES(4,'Brazil')
INSERT INTO "Countries" VALUES(5,'Pakistan')
INSERT INTO "Countries" VALUES(6,'Nigeria')
INSERT INTO "Countries" VALUES(7,'Bangladesh')
INSERT INTO "Countries" VALUES(8,'Russia')
INSERT INTO "Countries" VALUES(9,'Japan')
INSERT INTO "Countries" VALUES(10,'Mexico')
INSERT INTO "Countries" VALUES(11,'Philippines')
INSERT INTO "Countries" VALUES(12,'Vietnam')
INSERT INTO "Countries" VALUES(13,'Ethiopia')
INSERT INTO "Countries" VALUES(14,'Egypt')
INSERT INTO "Countries" VALUES(15,'Germany')
INSERT INTO "Countries" VALUES(16,'Iran')
INSERT INTO "Countries" VALUES(17,'Turkey')
INSERT INTO "Countries" VALUES(18,'Democratic Republic of the Congo')
INSERT INTO "Countries" VALUES(19,'Thailand')
INSERT INTO "Countries" VALUES(20,'France')
INSERT INTO "Countries" VALUES(21,'United Kingdom')
INSERT INTO "Countries" VALUES(22,'Italy')
INSERT INTO "Countries" VALUES(23,'Burma')
INSERT INTO "Countries" VALUES(24,'South Africa')
INSERT INTO "Countries" VALUES(25,'South Korea')
INSERT INTO "Countries" VALUES(26,'Colombia')
INSERT INTO "Countries" VALUES(27,'Spain')
INSERT INTO "Countries" VALUES(28,'Ukraine')
INSERT INTO "Countries" VALUES(29,'Tanzania')
INSERT INTO "Countries" VALUES(30,'Kenya')
INSERT INTO "Countries" VALUES(31,'Argentina')
INSERT INTO "Countries" VALUES(32,'Algeria')
INSERT INTO "Countries" VALUES(33,'Poland')
INSERT INTO "Countries" VALUES(34,'Sudan')
INSERT INTO "Countries" VALUES(35,'Uganda')
INSERT INTO "Countries" VALUES(36,'Canada')
INSERT INTO "Countries" VALUES(37,'Iraq')
INSERT INTO "Countries" VALUES(38,'Morocco')
INSERT INTO "Countries" VALUES(39,'Peru')
INSERT INTO "Countries" VALUES(40,'Uzbekistan')
INSERT INTO "Countries" VALUES(41,'Saudi Arabia')
INSERT INTO "Countries" VALUES(42,'Malaysia')
INSERT INTO "Countries" VALUES(43,'Venezuela')
INSERT INTO "Countries" VALUES(44,'Nepal')
INSERT INTO "Countries" VALUES(45,'Afghanistan')
INSERT INTO "Countries" VALUES(46,'Yemen')
INSERT INTO "Countries" VALUES(47,'North Korea')
INSERT INTO "Countries" VALUES(48,'Ghana')
INSERT INTO "Countries" VALUES(49,'Mozambique')
INSERT INTO "Countries" VALUES(50,'Taiwan')
INSERT INTO "Countries" VALUES(51,'Australia')
INSERT INTO "Countries" VALUES(52,'Ivory Coast')
INSERT INTO "Countries" VALUES(53,'Syria')
INSERT INTO "Countries" VALUES(54,'Madagascar')
INSERT INTO "Countries" VALUES(55,'Angola')
INSERT INTO "Countries" VALUES(56,'Cameroon')
INSERT INTO "Countries" VALUES(57,'Sri Lanka')
INSERT INTO "Countries" VALUES(58,'Romania')
INSERT INTO "Countries" VALUES(59,'Burkina Faso')
INSERT INTO "Countries" VALUES(60,'Niger')
INSERT INTO "Countries" VALUES(61,'Kazakhstan')
INSERT INTO "Countries" VALUES(62,'Netherlands')
INSERT INTO "Countries" VALUES(63,'Chile')
INSERT INTO "Countries" VALUES(64,'Malawi')
INSERT INTO "Countries" VALUES(65,'Ecuador')
INSERT INTO "Countries" VALUES(66,'Guatemala')
INSERT INTO "Countries" VALUES(67,'Mali')
INSERT INTO "Countries" VALUES(68,'Cambodia')
INSERT INTO "Countries" VALUES(69,'Senegal')
INSERT INTO "Countries" VALUES(70,'Zambia')
INSERT INTO "Countries" VALUES(71,'Zimbabwe')
INSERT INTO "Countries" VALUES(72,'Chad')
INSERT INTO "Countries" VALUES(73,'South Sudan')
INSERT INTO "Countries" VALUES(74,'Belgium')
INSERT INTO "Countries" VALUES(75,'Cuba')
INSERT INTO "Countries" VALUES(76,'Tunisia')
INSERT INTO "Countries" VALUES(77,'Guinea')
INSERT INTO "Countries" VALUES(78,'Greece')
INSERT INTO "Countries" VALUES(79,'Portugal')
INSERT INTO "Countries" VALUES(80,'Rwanda')
INSERT INTO "Countries" VALUES(81,'Czech Republic')
INSERT INTO "Countries" VALUES(82,'Somalia')
INSERT INTO "Countries" VALUES(83,'Haiti')
INSERT INTO "Countries" VALUES(84,'Benin')
INSERT INTO "Countries" VALUES(85,'Burundi')
INSERT INTO "Countries" VALUES(86,'Bolivia')
INSERT INTO "Countries" VALUES(87,'Hungary')
INSERT INTO "Countries" VALUES(88,'Sweden')
INSERT INTO "Countries" VALUES(89,'Belarus')
INSERT INTO "Countries" VALUES(90,'Dominican Republic')
INSERT INTO "Countries" VALUES(91,'Azerbaijan')
INSERT INTO "Countries" VALUES(92,'Honduras')
INSERT INTO "Countries" VALUES(93,'Austria')
INSERT INTO "Countries" VALUES(94,'United Arab Emirates')
INSERT INTO "Countries" VALUES(95,'Israel')
INSERT INTO "Countries" VALUES(96,'Switzerland')
INSERT INTO "Countries" VALUES(97,'Tajikistan')
INSERT INTO "Countries" VALUES(98,'Bulgaria')
INSERT INTO "Countries" VALUES(99,'Hong Kong (China)')
INSERT INTO "Countries" VALUES(100,'Serbia')
INSERT INTO "Countries" VALUES(101,'Papua New Guinea')
INSERT INTO "Countries" VALUES(102,'Paraguay')
INSERT INTO "Countries" VALUES(103,'Laos')
INSERT INTO "Countries" VALUES(104,'Jordan')
INSERT INTO "Countries" VALUES(105,'El Salvador')
INSERT INTO "Countries" VALUES(106,'Eritrea')
INSERT INTO "Countries" VALUES(107,'Libya')
INSERT INTO "Countries" VALUES(108,'Togo')
INSERT INTO "Countries" VALUES(109,'Sierra Leone')
INSERT INTO "Countries" VALUES(110,'Nicaragua')
INSERT INTO "Countries" VALUES(111,'Kyrgyzstan')
INSERT INTO "Countries" VALUES(112,'Denmark')
INSERT INTO "Countries" VALUES(113,'Finland')
INSERT INTO "Countries" VALUES(114,'Slovakia')
INSERT INTO "Countries" VALUES(115,'Singapore')
INSERT INTO "Countries" VALUES(116,'Turkmenistan')
INSERT INTO "Countries" VALUES(117,'Norway')
INSERT INTO "Countries" VALUES(118,'Lebanon')
INSERT INTO "Countries" VALUES(119,'Costa Rica')
INSERT INTO "Countries" VALUES(120,'Central African Republic')
INSERT INTO "Countries" VALUES(121,'Ireland')
INSERT INTO "Countries" VALUES(122,'Georgia')
INSERT INTO "Countries" VALUES(123,'New Zealand')
INSERT INTO "Countries" VALUES(124,'Republic of the Congo')
INSERT INTO "Countries" VALUES(125,'Palestine')
INSERT INTO "Countries" VALUES(126,'Liberia')
INSERT INTO "Countries" VALUES(127,'Croatia')
INSERT INTO "Countries" VALUES(128,'Oman')
INSERT INTO "Countries" VALUES(129,'Bosnia and Herzegovina')
INSERT INTO "Countries" VALUES(130,'Puerto Rico')
INSERT INTO "Countries" VALUES(131,'Kuwait')
INSERT INTO "Countries" VALUES(132,'Moldov')
INSERT INTO "Countries" VALUES(133,'Mauritania')
INSERT INTO "Countries" VALUES(134,'Panama')
INSERT INTO "Countries" VALUES(135,'Uruguay')
INSERT INTO "Countries" VALUES(136,'Armenia')
INSERT INTO "Countries" VALUES(137,'Lithuania')
INSERT INTO "Countries" VALUES(138,'Albania')
INSERT INTO "Countries" VALUES(139,'Mongolia')
INSERT INTO "Countries" VALUES(140,'Jamaica')
INSERT INTO "Countries" VALUES(141,'Namibia')
INSERT INTO "Countries" VALUES(142,'Lesotho')
INSERT INTO "Countries" VALUES(143,'Qatar')
INSERT INTO "Countries" VALUES(144,'Macedonia')
INSERT INTO "Countries" VALUES(145,'Slovenia')
INSERT INTO "Countries" VALUES(146,'Botswana')
INSERT INTO "Countries" VALUES(147,'Latvia')
INSERT INTO "Countries" VALUES(148,'Gambia')
INSERT INTO "Countries" VALUES(149,'Kosovo')
INSERT INTO "Countries" VALUES(150,'Guinea-Bissau')
INSERT INTO "Countries" VALUES(151,'Gabon')
INSERT INTO "Countries" VALUES(152,'Equatorial Guinea')
INSERT INTO "Countries" VALUES(153,'Trinidad and Tobago')
INSERT INTO "Countries" VALUES(154,'Estonia')
INSERT INTO "Countries" VALUES(155,'Mauritius')
INSERT INTO "Countries" VALUES(156,'Swaziland')
INSERT INTO "Countries" VALUES(157,'Bahrain')
INSERT INTO "Countries" VALUES(158,'Timor-Leste')
INSERT INTO "Countries" VALUES(159,'Djibouti')
INSERT INTO "Countries" VALUES(160,'Cyprus')
INSERT INTO "Countries" VALUES(161,'Fiji')
INSERT INTO "Countries" VALUES(162,'Reunion (France)')
INSERT INTO "Countries" VALUES(163,'Guyana')
INSERT INTO "Countries" VALUES(164,'Comoros')
INSERT INTO "Countries" VALUES(165,'Bhutan')
INSERT INTO "Countries" VALUES(166,'Montenegro')
INSERT INTO "Countries" VALUES(167,'Macau (China)')
INSERT INTO "Countries" VALUES(168,'Solomon Islands')
INSERT INTO "Countries" VALUES(169,'Western Sahara')
INSERT INTO "Countries" VALUES(170,'Luxembourg')
INSERT INTO "Countries" VALUES(171,'Suriname')
INSERT INTO "Countries" VALUES(172,'Cape Verde')
INSERT INTO "Countries" VALUES(173,'Malta')
INSERT INTO "Countries" VALUES(174,'Guadeloupe (France)')
INSERT INTO "Countries" VALUES(175,'Martinique (France)')
INSERT INTO "Countries" VALUES(176,'Brunei')
INSERT INTO "Countries" VALUES(177,'Bahamas')
INSERT INTO "Countries" VALUES(178,'Iceland')
INSERT INTO "Countries" VALUES(179,'Maldives')
INSERT INTO "Countries" VALUES(180,'Belize')
INSERT INTO "Countries" VALUES(181,'Barbados')
INSERT INTO "Countries" VALUES(182,'French Polynesia (France)')
INSERT INTO "Countries" VALUES(183,'Vanuatu')
INSERT INTO "Countries" VALUES(184,'New Caledonia (France)')
INSERT INTO "Countries" VALUES(185,'French Guiana (France)')
INSERT INTO "Countries" VALUES(186,'Mayotte (France)')
INSERT INTO "Countries" VALUES(187,'Samoa')
INSERT INTO "Countries" VALUES(188,'Sao Tom and Principe')
INSERT INTO "Countries" VALUES(189,'Saint Lucia')
INSERT INTO "Countries" VALUES(190,'Guam (USA)')
INSERT INTO "Countries" VALUES(191,'Curacao (Netherlands)')
INSERT INTO "Countries" VALUES(192,'Saint Vincent and the Grenadines')
INSERT INTO "Countries" VALUES(193,'Kiribati')
INSERT INTO "Countries" VALUES(194,'United States Virgin Islands (USA)')
INSERT INTO "Countries" VALUES(195,'Grenada')
INSERT INTO "Countries" VALUES(196,'Tonga')
INSERT INTO "Countries" VALUES(197,'Aruba (Netherlands)')
INSERT INTO "Countries" VALUES(198,'Federated States of Micronesia')
INSERT INTO "Countries" VALUES(199,'Jersey (UK)')
INSERT INTO "Countries" VALUES(200,'Seychelles')
INSERT INTO "Countries" VALUES(201,'Antigua and Barbuda')
INSERT INTO "Countries" VALUES(202,'Isle of Man (UK)')
INSERT INTO "Countries" VALUES(203,'Andorra')
INSERT INTO "Countries" VALUES(204,'Dominica')
INSERT INTO "Countries" VALUES(205,'Bermuda (UK)')
INSERT INTO "Countries" VALUES(206,'Guernsey (UK)')
INSERT INTO "Countries" VALUES(207,'Greenland (Denmark)')
INSERT INTO "Countries" VALUES(208,'Marshall Islands')
INSERT INTO "Countries" VALUES(209,'American Samoa (USA)')
INSERT INTO "Countries" VALUES(210,'Cayman Islands (UK)')
INSERT INTO "Countries" VALUES(211,'Saint Kitts and Nevis')
INSERT INTO "Countries" VALUES(212,'Northern Mariana Islands (USA)')
INSERT INTO "Countries" VALUES(213,'Faroe Islands (Denmark)')
INSERT INTO "Countries" VALUES(214,'Sint Maarten (Netherlands)')
INSERT INTO "Countries" VALUES(215,'Saint Martin (France)')
INSERT INTO "Countries" VALUES(216,'Liechtenstein')
INSERT INTO "Countries" VALUES(217,'Monaco')
INSERT INTO "Countries" VALUES(218,'San Marino')
INSERT INTO "Countries" VALUES(219,'Turks and Caicos Islands (UK)')
INSERT INTO "Countries" VALUES(220,'Gibraltar (UK)')
INSERT INTO "Countries" VALUES(221,'British Virgin Islands (UK)')
INSERT INTO "Countries" VALUES(222,'Aland Islands (Finland)')
INSERT INTO "Countries" VALUES(223,'Caribbean Netherlands (Netherlands)')
INSERT INTO "Countries" VALUES(224,'Palau')
INSERT INTO "Countries" VALUES(225,'Cook Islands (NZ)')
INSERT INTO "Countries" VALUES(226,'Anguilla (UK)')
INSERT INTO "Countries" VALUES(227,'Wallis and Futuna (France)')
INSERT INTO "Countries" VALUES(228,'Tuvalu')
INSERT INTO "Countries" VALUES(229,'Nauru')
INSERT INTO "Countries" VALUES(230,'Saint Barthelemy (France)')
INSERT INTO "Countries" VALUES(231,'Saint Pierre and Miquelon (France)')
INSERT INTO "Countries" VALUES(232,'Montserrat (UK)')
INSERT INTO "Countries" VALUES(233,'Saint Helena, Ascension and Tristan da Cunha (UK)')
INSERT INTO "Countries" VALUES(234,'Svalbard and Jan Mayen (Norway)')
INSERT INTO "Countries" VALUES(235,'Falkland Islands (UK)')
INSERT INTO "Countries" VALUES(236,'Norfolk Island (Australia)')
INSERT INTO "Countries" VALUES(237,'Christmas Island (Australia)')
INSERT INTO "Countries" VALUES(238,'Niue (NZ)')
INSERT INTO "Countries" VALUES(239,'Tokelau (NZ)')
INSERT INTO "Countries" VALUES(240,'Vatican City')
INSERT INTO "Countries" VALUES(241,'Cocos (Keeling) Islands (Australia)')
INSERT INTO "Countries" VALUES(242,'Pitcairn Islands (UK)')
```

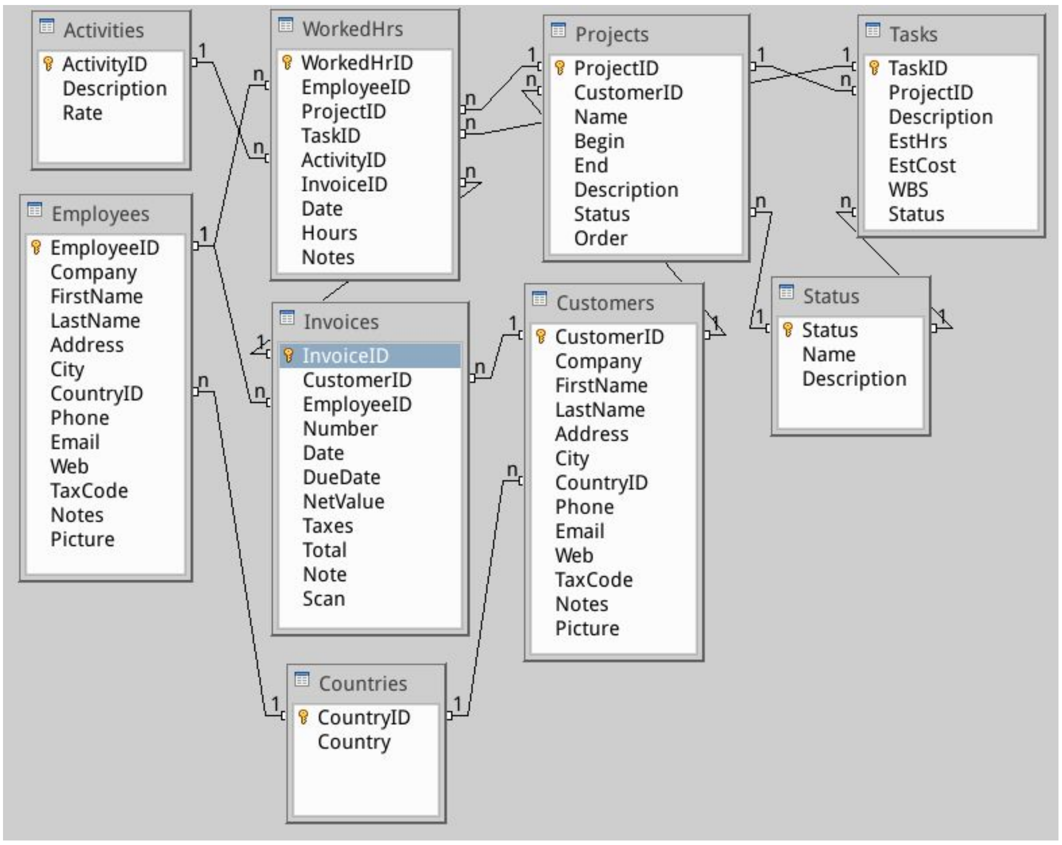
RELATIONSHIPS

Now is the time to look at the relationship between the tables we just created. generally speaking a master data table will likely have 1:n relationship while an operations table would have n:1.

Learn more on referential integrity which s the main principle a beginner approaches database systems.

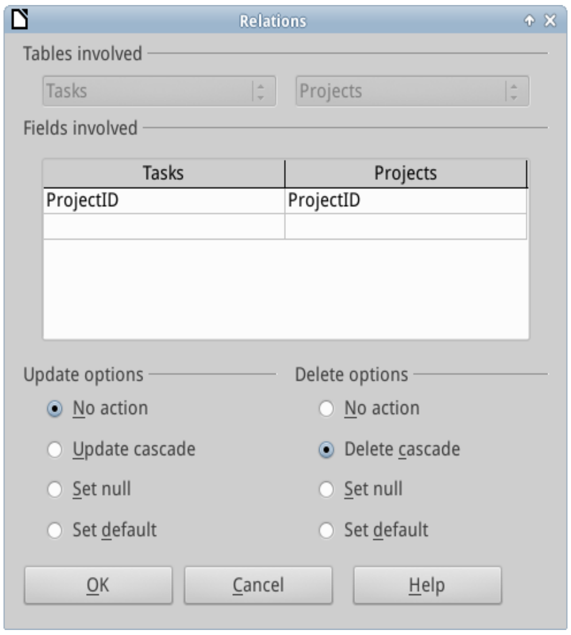
SET

From menu Tools → Relationships and for each table set the proper relationship, verify that dragging over to create connection actually create the correct relationship 1:1 or 1:n.



ACTION

To grant referential integrity action may be set within the relationship, for instance if an information is spread over few tables and deletion occurs we may have an action that simultaneously eliminate records from other tables. Let's say if I delete a project, I want to be sure that also its tasks are eliminated.


Relations

Tables involved

Tasks | Projects

Fields involved

Tasks	Projects
ProjectID	ProjectID

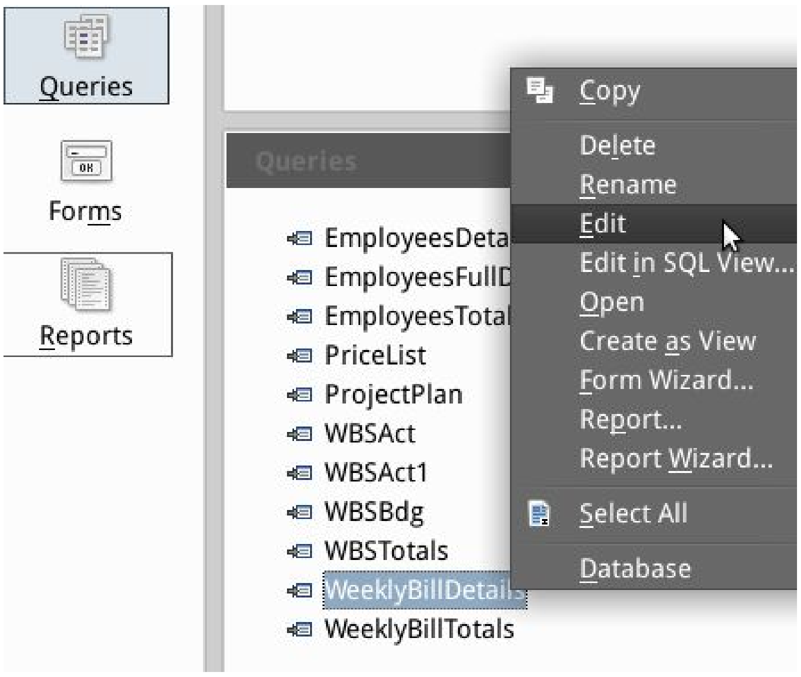
Update options
- [✓ No action]
- Update cascade
- Set null
- Set default

Delete options
- No action
- [✓ Delete cascade]
- Set null
- Set default

QUERIES

The simplest way to select data is to create a query, which allows us also to present item by item or sum data. SQL query may be created by coding the script or by design the relations and filters, for this tutorial we will create a query that shows all our billable worked hours in details and summed by weeks.

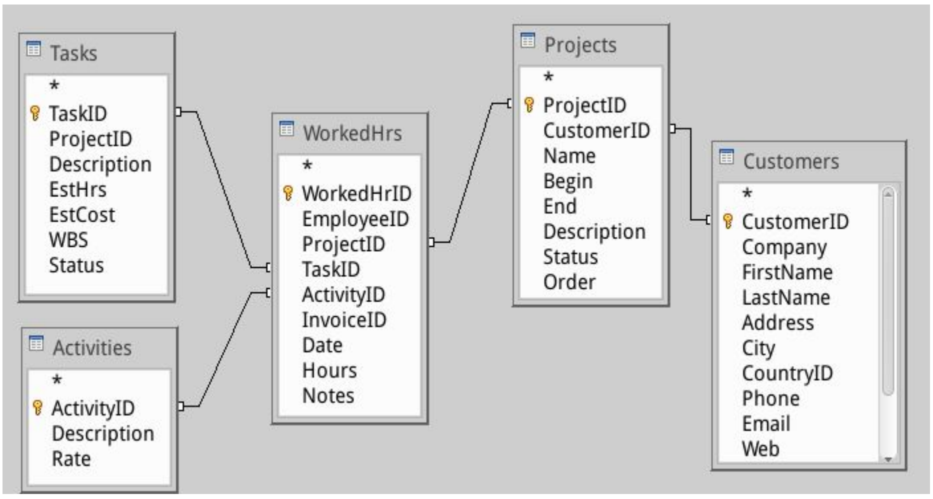
Now select view “Queries” and look for “WeeklyBillDetails” and “WeeklyBillTotals”.



WORKED HOURS

This query will select data from table “WorkedHrs” linking with three key fields of project, task and activity. The primary order is based on the link to the key field of “CustomerID” found in the table “Projects”.

Run the query on the sample data, review the design by selecting tab “Queries” and editing “WeeklyBillDetails”. This will be use in forms and reports without the need to copy the SQL script every time.



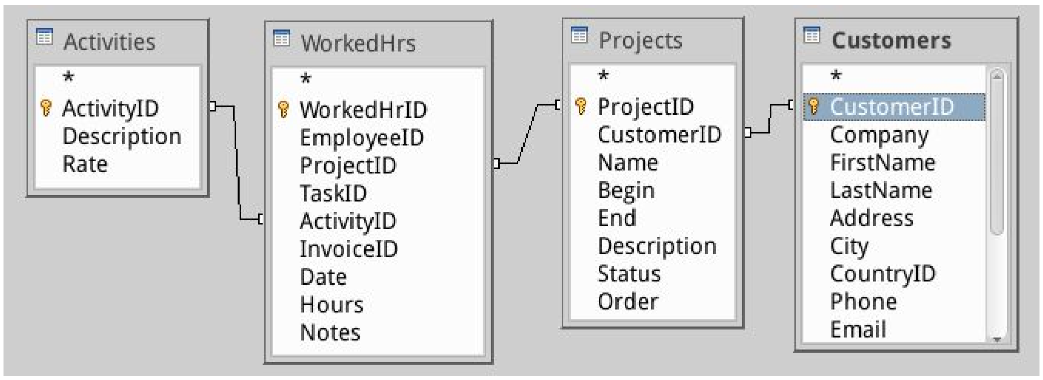
The SQL script derived form design is the following:

```
SELECT "Customers"."CustomerID", "WorkedHrs"."ProjectID", WEEK( "WorkedHrs"."Date" ) AS
"Week", "Tasks"."Status" AS "Phase", "Tasks"."WBS" AS "WBS", "Tasks"."Description" AS
"WBSDesc", "Activities"."Description" AS "Activity", AVG( "Activities"."Rate" ) AS "Rate", SUM(
"WorkedHrs"."Hours" ) AS "Hrs", SUM( "Rate" * "Hours" ) AS "Bill", "Projects"."Name" AS
"Project", "Customers"."Company" AS "Customer", "WorkedHrs"."Notes", "WorkedHrs"."Date"
FROM "WorkedHrs", "Projects", "Customers", "Activities", "Tasks" WHERE
"WorkedHrs"."ProjectID" = "Projects"."ProjectID" AND "Projects"."CustomerID" =
"Customers"."CustomerID" AND "WorkedHrs"."ActivityID" = "Activities"."ActivityID" AND
"WorkedHrs"."TaskID" = "Tasks"."TaskID" GROUP BY "Customers"."CustomerID",
"WorkedHrs"."ProjectID", "Tasks"."Status", "Tasks"."WBS", "Tasks"."Description",
"Activities"."Description", "Projects"."Name", "Customers"."Company", "WorkedHrs"."Notes",
"WorkedHrs"."Date", WEEK( "WorkedHrs"."Date" ) ORDER BY "Customers"."CustomerID" ASC,
"WorkedHrs"."ProjectID" ASC, "Week" ASC, "Phase" ASC, "WBS" ASC, "WBSDesc" ASC
```

WORKED HOURS BY WEEK

This query will select data from table “WorkedHrs” linking with two key fields of project, and activity. The primary order is based on the link to the key field of “CustomerID” found in the table “Projects”.

Again, run the query on the sample data, review the design by selecting tab “Queries” and editing “WeeklyBillTotals”.



The SQL script derived form design is the following:

```
SELECT "Customers"."CustomerID", WEEK( "WorkedHrs"."Date" ) AS "Week",
"Projects"."ProjectID", "Projects"."Name" AS "Project", "Customers"."Company" AS "Customer",
SUM( "WorkedHrs"."Hours" ) AS "Hrs", SUM( "Rate" * "Hours" ) AS "Bill" FROM "WorkedHrs",
"Projects", "Customers", "Activities" WHERE "WorkedHrs"."ProjectID" = "Projects"."ProjectID"
AND "Projects"."CustomerID" = "Customers"."CustomerID" AND "WorkedHrs"."ActivityID" =
"Activities"."ActivityID" GROUP BY "Customers"."CustomerID", "Projects"."ProjectID",
"Projects"."Name", "Customers"."Company", WEEK( "WorkedHrs"."Date" )
```

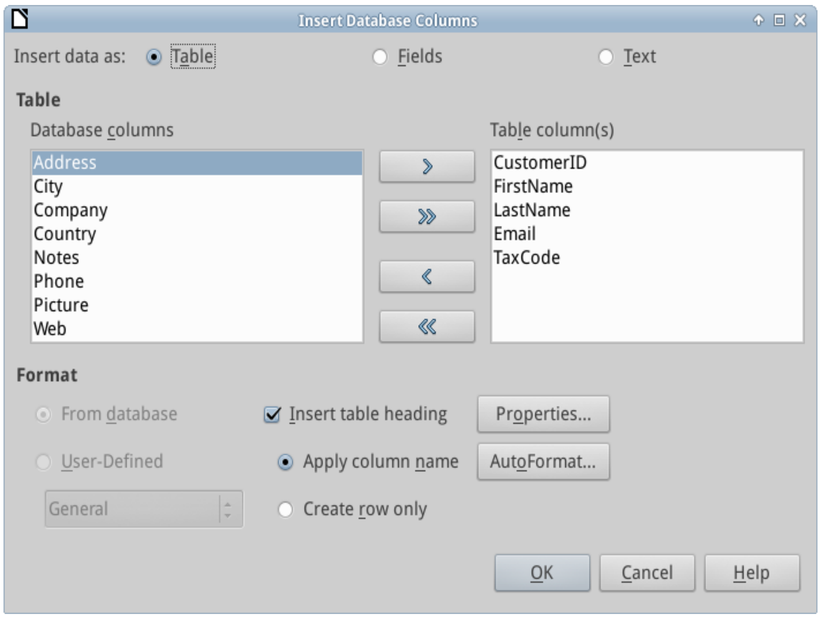
TRANSFER DATA TO OTHER APPLICATIONS

Working within LibreOffice allows you to transfer data between applications of the suite, there are different levels of integration which are well described in the chapter X of "Getting Started with LibreOffice" guide.

PRESENT DATA

Tables and data from LibreOffice BASE are easily transferred to Writer as usual with "CTRL+C" and "CTRL+V", however the purpose of placing data in a word processor is different from a spreadsheet.

For example pasting tables to Calc is the favorite way to export to .cvs format, but it doesn't create a nice report; if we want to present nicely data we should paste in Writer.



For this tutorial copy “Customers” table and paste in a Writer document.

While pasting select few fields to create a nice address book and click on “AutoFormat” to choose a color combination of your taste.

INTERMEDIATE

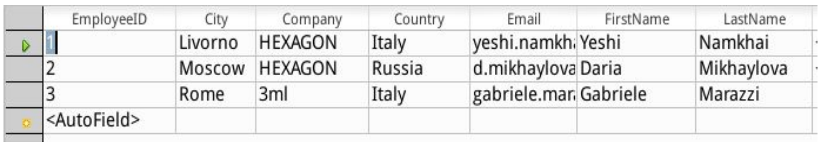
In this section of the **xTimeCard** tutorial we create the initial settings to allow **xTimeCard** to run properly and accept customers and projects.

INITIAL SETTINGS

With these few steps we create the minimal data for **xTimeCard** to work, at the end we will be able to print our price list and to issue invoices and add worked hours to projects.

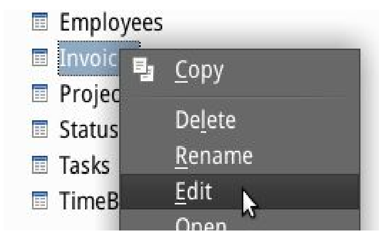
EMPLOYEES

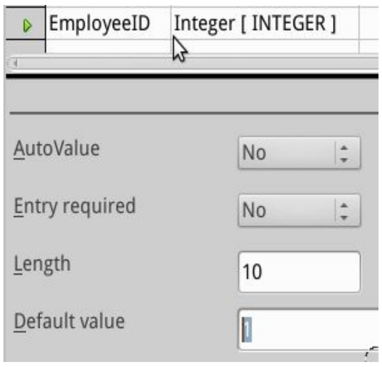
From the tables view, double click on "Employees" and add at least yourself or your company information.



EmployeeID	City	Company	Country	Email	FirstName	LastName
1	Livorno	HEXAGON	Italy	yeshi.namkh.	Yeshi	Namkhai
2	Moscow	HEXAGON	Russia	d.mikhaylova	Daria	Mikhaylova
3	Rome	3ml	Italy	gabriele.mar.	Gabriele	Marazzi
<AutoField>						

For the sake of simplicity the first employee is to be considered as the company that will issue invoice, eventually **xTimeCard** will allow to create invoices with any master data but we set a default to the table of invoices. Edit the table "Invoices" and scroll down to set the proper default.





EmployeeID Integer [INTEGER]

AutoValue: No

Entry required: No

Length: 10

Default value: 1

[..]

Invoices

To perform basic in

Activites

[..]

ALTER STRUCTURE

Many of us faced, during development of a database, mistakes or imperfect settings. For instance a missing field or even just wrong sequence. For this tutorial we will rebuild a table or alter it, without damaging data.

COPY AND PASTE

The simplest way to alter the structure of a table is to change the order of fields, if eventually we have missing fields we can add them before copy.

For example our “Employees” table is not correctly organize but unfortunately we have already used to store data, select and press “CTRL+C” and “CRTL+V”.

```
CREATE CACHED TABLE "Employees"("EmployeeID" INTEGER GENERATED BY
DEFAULT AS IDENTITY(START WITH 0) NOT NULL PRIMARY KEY,"City"
VARCHAR(50),"Company" VARCHAR(50),"Country" VARCHAR(50),"Email"
VARCHAR(50),"FirstName" VARCHAR(50),"LastName" VARCHAR(50),"Phone"
VARCHAR(30),"Address" VARCHAR(100),"Notes" LONGVARCHAR,"Web"
VARCHAR(200),"TaxCode" VARCHAR(50))
```

FORMS TEMPLATE

Now is the time to make your application operate more friendly by adding forms. Move to the forms view and create a nice looking template.

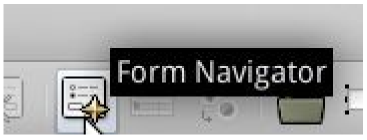
TEMPLATE

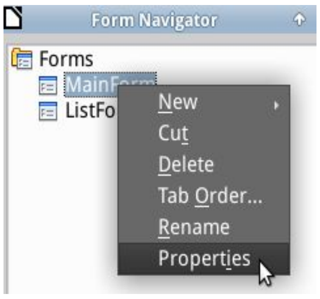
Always create a **MainForm** which may contain other sub-forms connected, this also allows to have master-detail interface, this is clearly explained in the Base Handbook.

[..picture of template form]

EMPTY FORM

Select forms view and click on “Create Form in Design View”, click on “Form Navigator” to add to our new empty form other two forms called “MainForm” and “ListForm”.

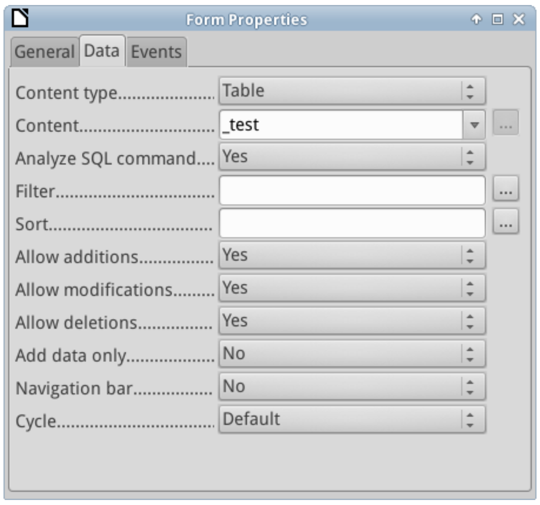




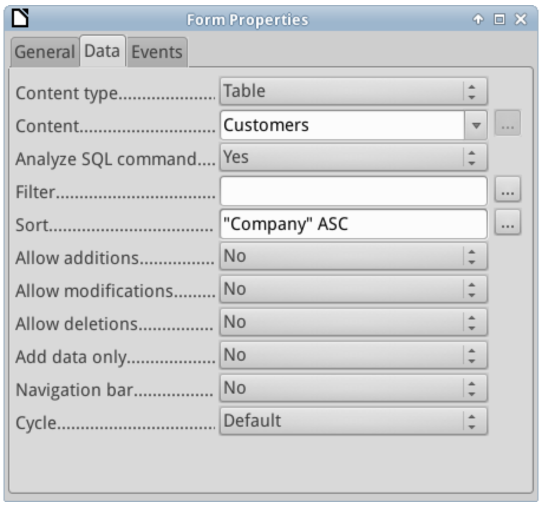
Right click on "Properties" of the "MainForm" and connect data to table "_test", repeat for "ListForm" and connect to table "Customers".

DATA OPERATIONS

For both forms do not show "navigation Bar" and for "ListForm" deactivate also other operations to data.



Content type	Table
Content	_test
Analyze SQL command	Yes
Filter	
Sort	
Allow additions	Yes
Allow modifications	Yes
Allow deletions	Yes
Add data only	No
Navigation bar	No
Cycle	Default

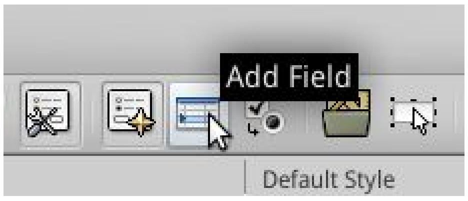


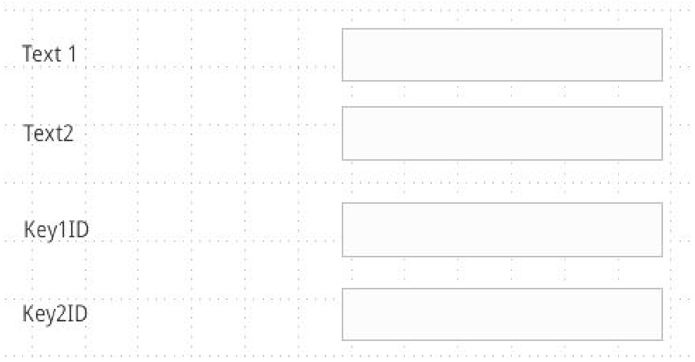
Content type	Table
Content	Customers
Analyze SQL command	Yes
Filter	
Sort	"Company" ASC
Allow additions	No
Allow modifications	No
Allow deletions	No
Add data only	No
Navigation bar	No
Cycle	Default

ADD FIELDS

Click on the "Add Field" and add all available fields beside "ID" from "_test" table, don't worry about type because you may change with right click selecting "Replace with".

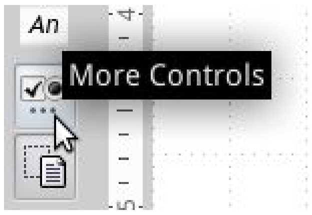
Now your form should look similar to this, add a "Navigation Bar" from "More controls" and customize like our example. Set width property to larger value like 5,70" and height to 0,39".

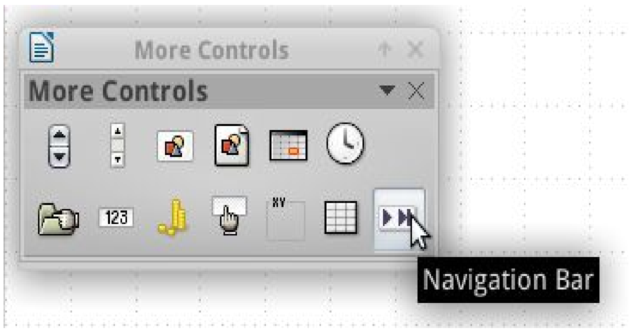




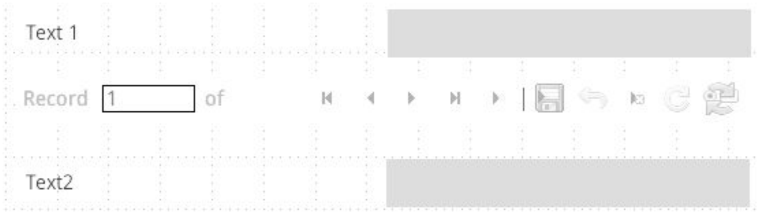
NAVIGATION BAR

Look for “More Controls” to a navigation bar, style properly with white background, no frame and hide filter/search.





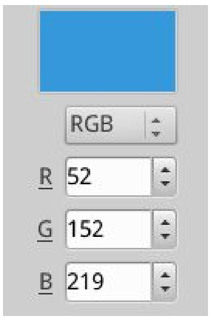
For this tutorial we set our forms with first line as a name/description of the master data, the second line as navigation bar.



Now save and give the name “_test”, double click to run.

COLORS

Add your favorite colors from menu Options → Options, for this tutorial add the color rgb 52,152,219 and rbg 34,104,147 and give a name to remember and use later on.



RGB

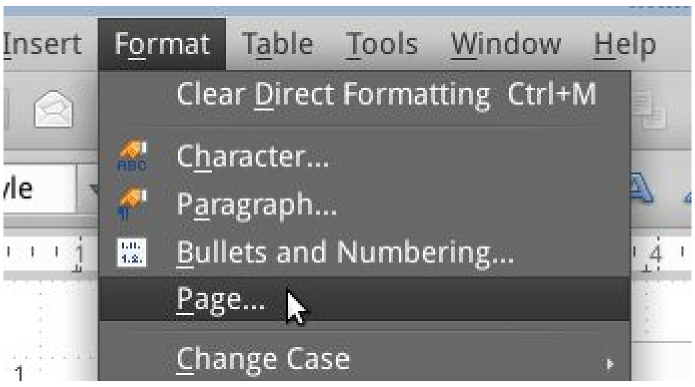
R 52

G 152

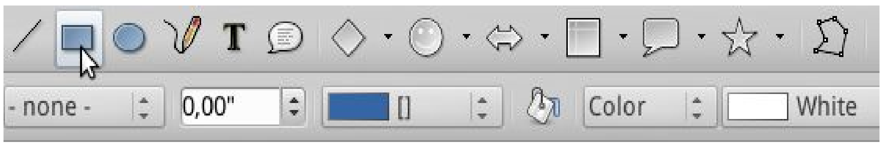
B 219

BACKGROUND

Our form looks nice but let's change the background of the page, from menu Format → Page → Background and select your previously created light blue color.

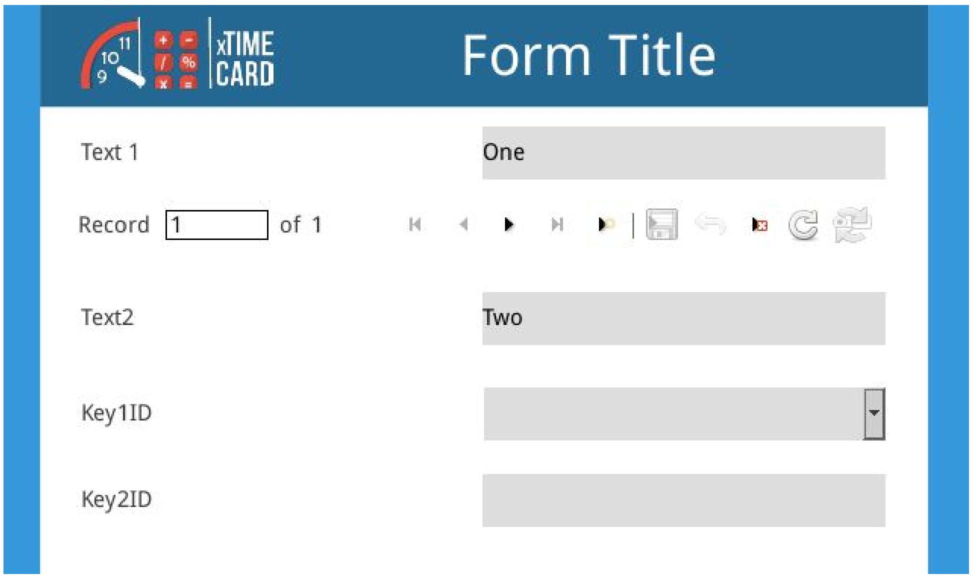


Activate the drawing toolbar from menu View → Toolbars → Drawing.



Now draw a white rectangle and send to background, this will be the main content area. Repeat on the top to create a header with the darker blue color previously created.

From menu Insert → Image add your favorite logo, re-size properly to fit the header. Add a label, enlarge font and set color to white write "Form Title", center it.



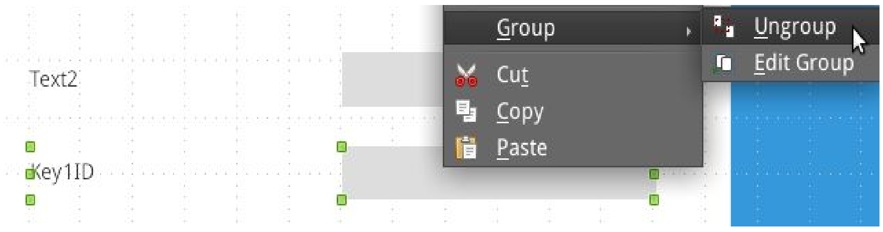
To complete the style of the form, set "Icons" of navigation bar to large, hide "Sort", change background of each field to "Gray 2" and deactivate frame border.

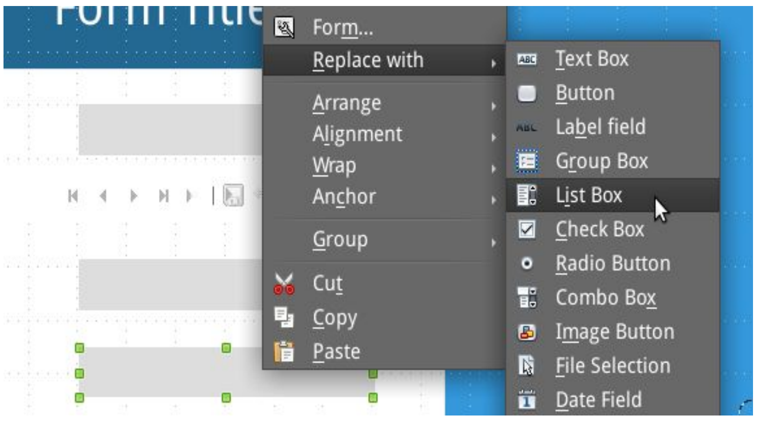
ANCHOR

As you may expect, pay attention when you deploy controls in the form; just like a word processor or a web page these controls need to be placed with an anchor. Remember to anchor properly, start with a wider anchor like “Page” and pay attention to the position X and Y.

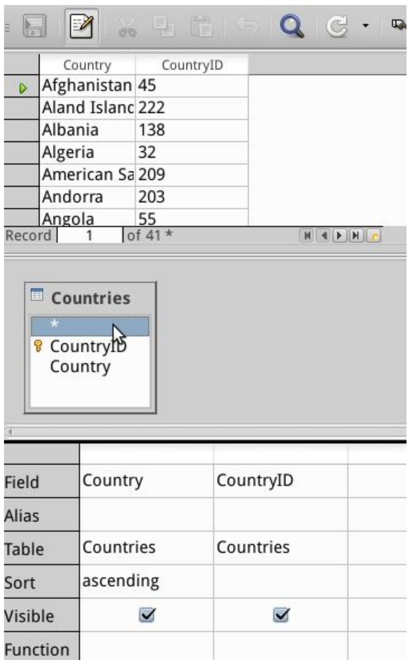
LISTBOX

Right click “Key1ID”, ungroup (separate from the label) and replace the text field control with a listbox. Right click and select “Replace with” and choose “List Box”, again right click and select “Control” to change properties.





Activate drop-down and go to tab “Data” to create SQL query to list from table “Countries”.



	Country	CountryID
▷	Afghanistan	45
	Aland Islanc	222
	Albania	138
	Algeria	32
	American Sa	209
	Andorra	203
	Angola	55

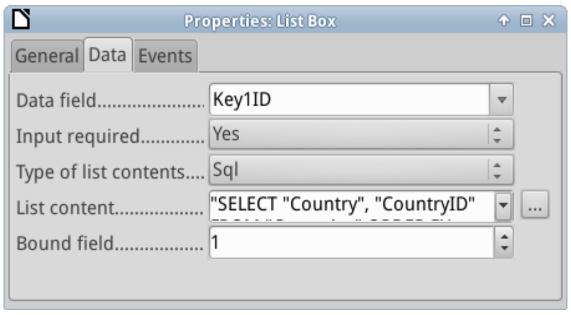
Record 1 of 41 *

Countries
*
CountryID
Country

Field	Country	CountryID
Alias		
Table	Countries	Countries
Sort	ascending	
Visible	✓	✓
Function		

Move to data to create a SQL query to list available projects. You may simply choose SQL and add this code:

```
SELECT "Country", "CountryID" FROM "Countries" ORDER BY "Country" ASC
```

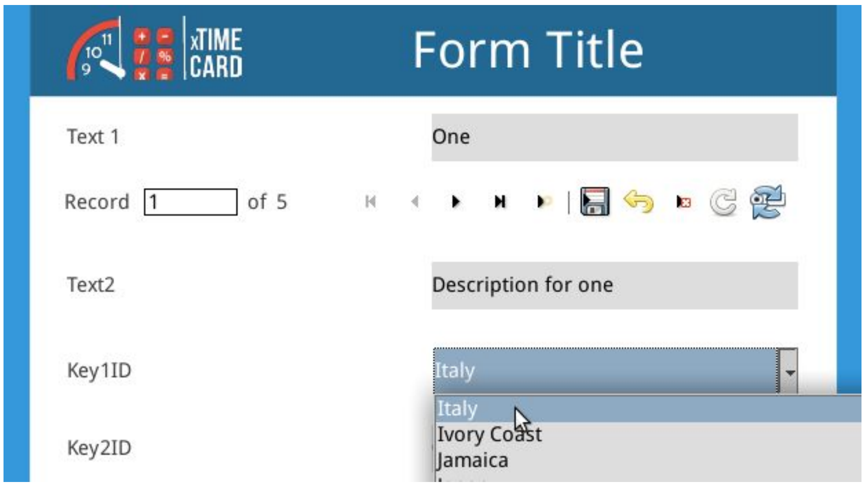


Properties: List Box

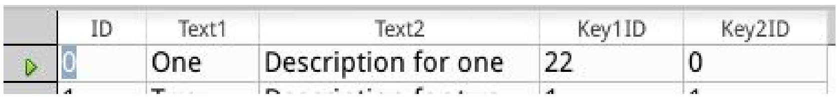
General | Data | Events

Data field	Key1ID
Input required	Yes
Type of list contents	Sql
List content	"SELECT "Country", "CountryID"
Bound field	1

Run your form and check the list box to properly show the name of all countries but save the “CountryID” in the “_test” table.



Click on “Save” button and go to “tables” and look into the table, for instance “Italy” is record 22.

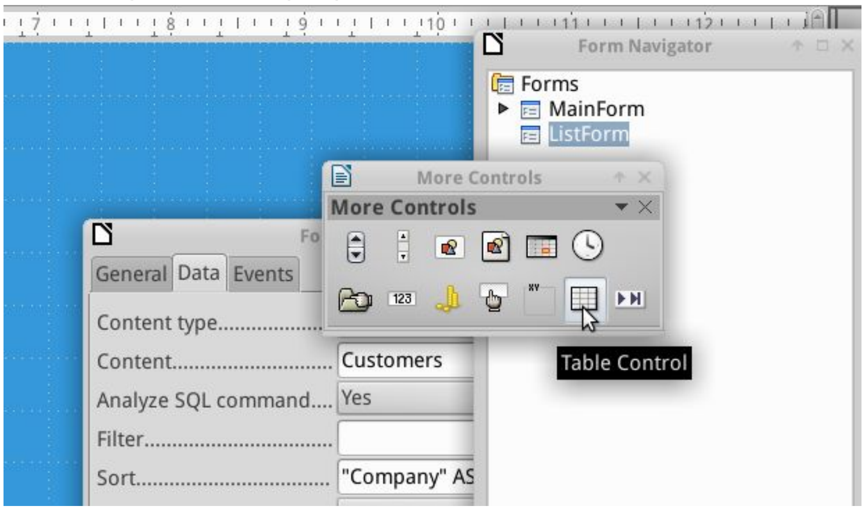


ID	Text1	Text2	Key1ID	Key2ID
0	One	Description for one	22	0

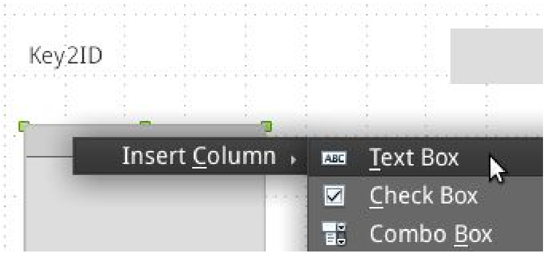
For this template we are almost done, next we will add a grid and connection between sub-forms (master-detail interface).

GRID

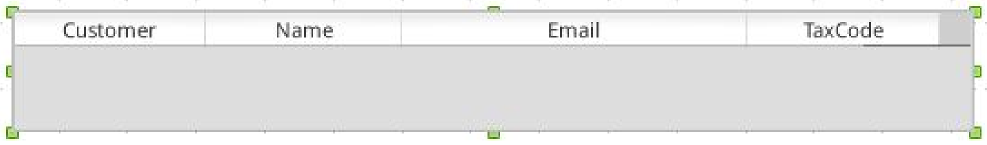
Drag a “Grid” control from the “More controls” dialog to our template form, draw a square in the lower frame. Pay attention to have your grid below “ListForm”, check in the “Navigator” or drag to set the proper level.



Right click on the control form the “Navigator” and select “Properties”. Set a nice “Gray1” background, give a flat appearance and call it “Grid1”. When you run your form, the list of your customers will be shown, for this tutorial review the sample data.



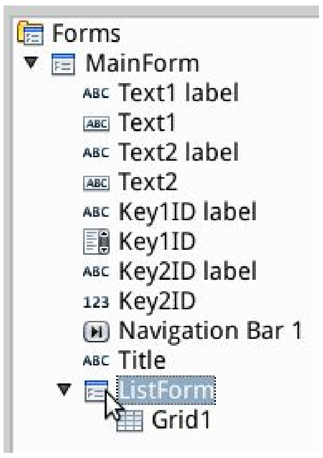
To columns to the grid, right click on the header and choose to insert "Text Box" connected to the fields "Company", "Name", "Email" and "Taxcode".

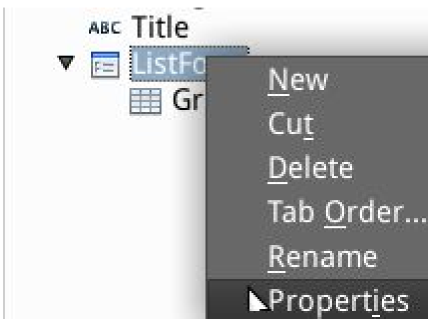


Save, run and see your customers shown in the grid, for this tutorial we would to connect the field "Key2ID" to this grid (therefore we will see only one row).

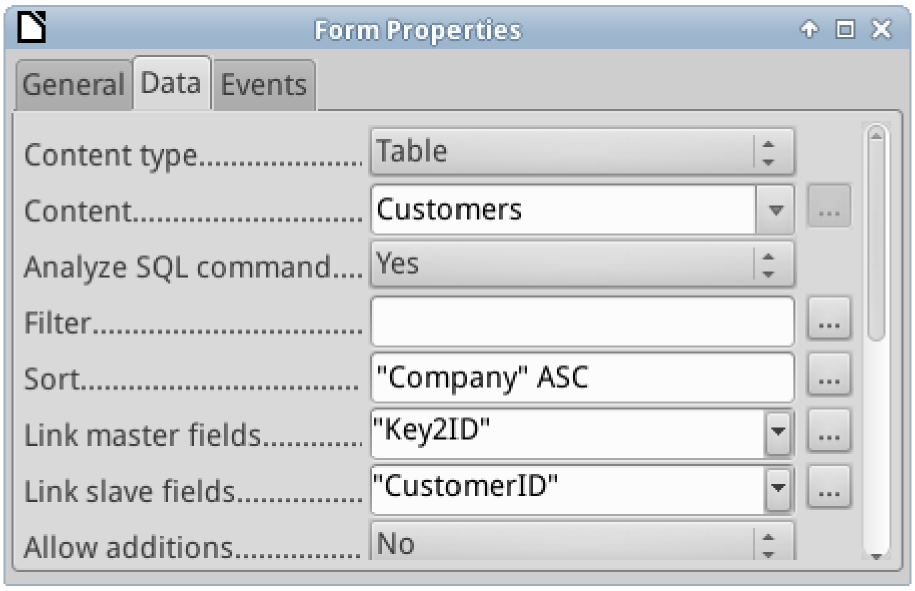
MASTER-DETAIL INTERFACE

To learn on how to connect a form to a sub-form and create the master-detail relation, let's start by opening the form "Navigator" and move the "ListFrom" under the "MainForm".



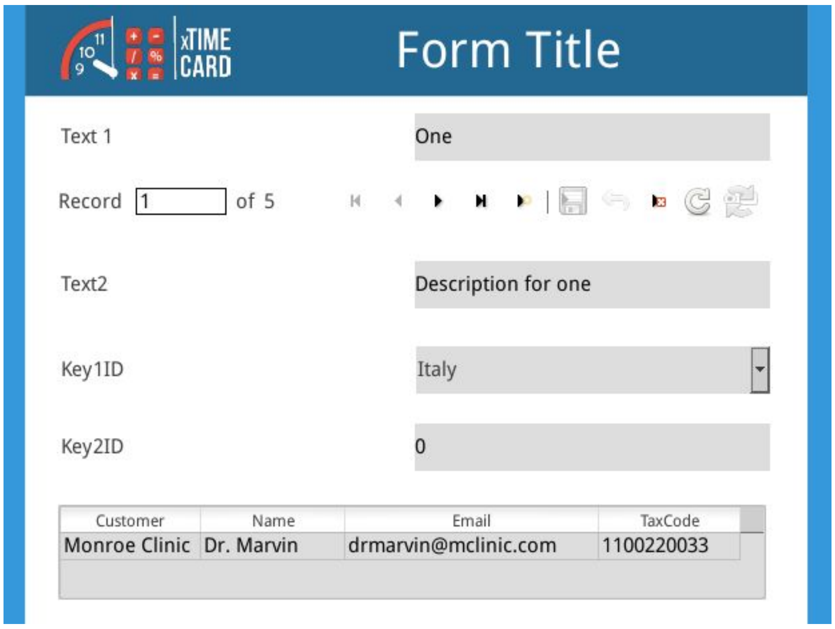


Right click for properties and set the relation between the tables "_test" and "Customers" by linking field "Key2ID" to field "CustomerID".



Content type	Table
Content	Customers
Analyze SQL command	Yes
Filter	
Sort	"Company" ASC
Link master fields	"Key2ID"
Link slave fields	"CustomerID"
Allow additions	No

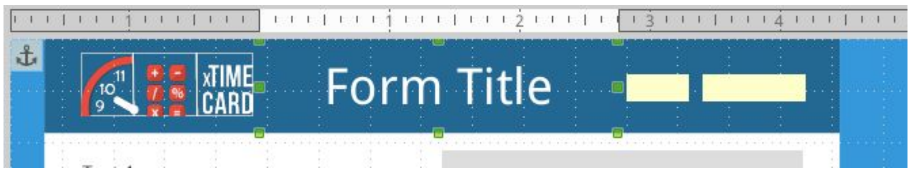
Save and run your form, notice now the grid shows only the proper customer record.



Text 1	One
Record 1 of 5	
Text2	Description for one
Key1ID	Italy
Key2ID	0

Customer	Name	Email	TaxCode
Monroe Clinic	Dr. Marvin	drmarvin@mclinic.com	1100220033

Now your form template looks nice, but still we are missing something in the header. Generally speaking the upper right should contain important information as our attention is often driven by “Z”.



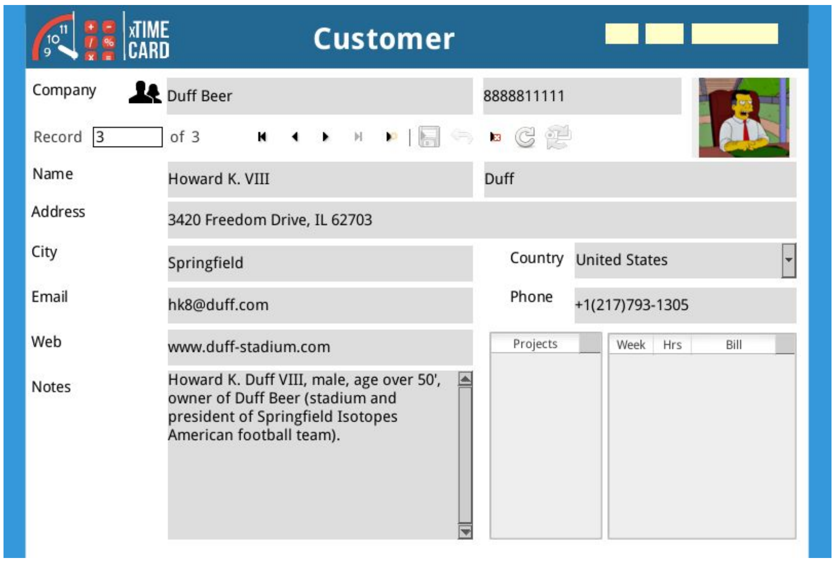
Let's create two fields for totals with a yellow color, don't connect to any for now.

MASTER DATA

Finally we are able to create

CUSTOMER'S FORM

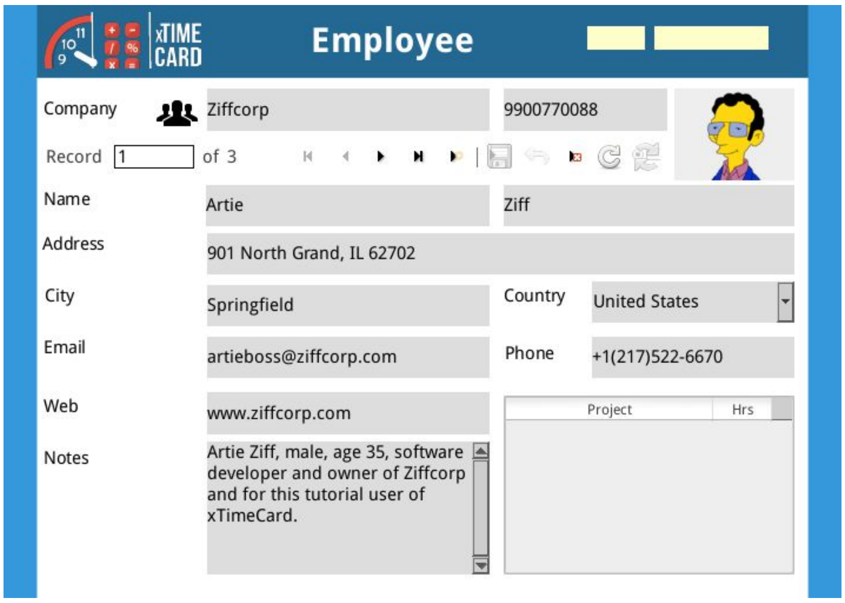
[...]



Company	Duff Beer	8888811111
Record 3 of 3		
Name	Howard K. VIII	Duff
Address	3420 Freedom Drive, IL 62703	
City	Springfield	Country United States
Email	hk8@duff.com	Phone +1(217)793-1305
Web	www.duff-stadium.com	
Notes	Howard K. Duff VIII, male, age over 50', owner of Duff Beer (stadium and president of Springfield Isotopes American football team).	

EMPLOYEE'S FORM

[..]



Company	Ziffcorp	9900770088
Record 1 of 3		
Name	Artie	Ziff
Address	901 North Grand, IL 62702	
City	Springfield	Country: United States
Email	artieboss@ziffcorp.com	Phone: +1(217)522-6670
Web	www.ziffcorp.com	
Notes	Artie Ziff, male, age 35, software developer and owner of Ziffcorp and for this tutorial user of xTimeCard.	

Project	Hrs

ADVANCED

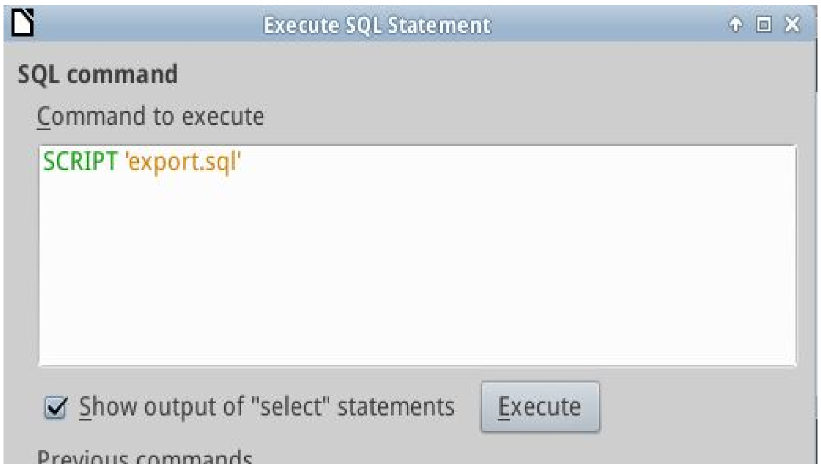
Our software is based on LibreOffice Base and make use of BASIC language with the Access2Base extension.

DATA IMPORT/EXPORT

Whenever we approach multi-user environements we need to be able to export and import the database structure and records. This a usually done in SQL by the use of standard commands like SCRIPT.

EXPORT

For this tutorial go to menu Tools → SQL to run command SCRIPT 'export.sql' and export your database in SQL format, open the file (that will be on your home folder) with a text editor to examine the content.



```
SCRIPT 'export.sql'
```

Generally speaking to import this file to MySQL or any other SQL engine you will need to adapt the dialect, as a matter every engine may have few little differences. Obviously this may also be done automatically to have a periodic backup.

NORTHWIND

Historically most of advanced users and developers learned about Microsoft Access and VBA experimenting with *NorthWind* sample database, for your convenience the attached code is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

1. Download the file.
2. Unzip all the files in one single directory of your choice.
3. Check the *class path* of your AOO/LibO Java Options as described in How do I setup Base+HSQLDB for non-embedded 'split database' access?.

4. *Open* the **"TT NorthWind.odb"** Base file, *enable* macros, *save* and *close* the file.
5. *Open* the **"TT NorthWind StandAlone.odt"** Writer file, *enable* macros - no data will be displayed -, *save* and *close* the file.
6. *Reopen* one or both files to explore the proposed forms and code. All forms, queries, Basic code in those files refer to one or more of next examples.

DO MORE WITH BASE

Software for database like Microsoft Access allows developers to customize via macro oriented programming language based on Visual Basic for Open/LibreOffice we use BASIC and function from the API called UNO. This is difficult for beginners therefore in this tutorial we reuse the knowledge previously developed programming Microsoft Access.

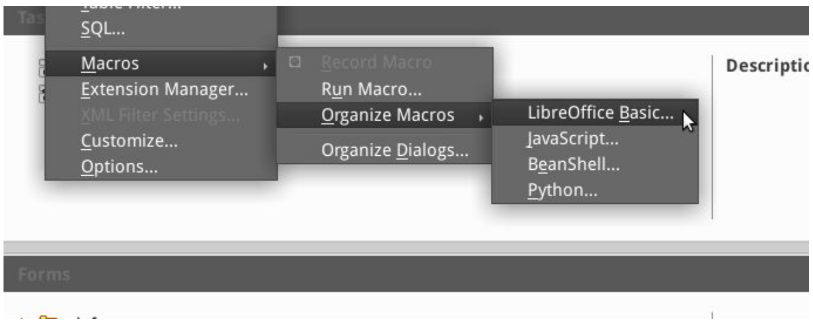
BASIC

LibreOffice Basic belongs to the family of Basic languages. Many parts of LibreOffice Basic are identical to Microsoft Visual Basic for Applications and Microsoft Visual Basic. Anyone who has already worked with these languages can quickly become accustomed to LibreOffice Basic.

You can also benefit from the advantages of object-oriented programming since an interface in LibreOffice Basic enables you to use external object libraries. LibreOffice Basic is an interpreter language.

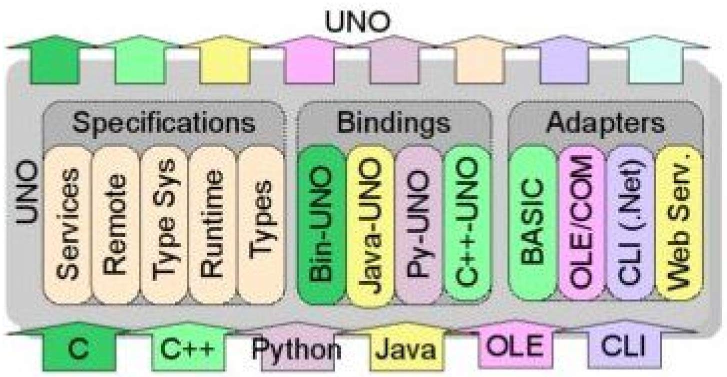
INTEGRATED DEVELOPMENT ENVIRONMENT

Like most of modern and advanced programming languages, also Open/LibreOffice BASIC has a full featured environment or IDE which provides an editor for creating and testing macros, the interpreter (to run macros) and the interfaces to various Open/LibreOffice applications (direct access to Office documents).



API AND UNO

LibreOffice objects and methods, such as paragraphs, spreadsheets, and fonts, are accessible to LibreOffice Basic through the LibreOffice application programming interface, or API. Through the API, for example, documents can be created, opened, modified and printed. The API can be used not only by LibreOffice Basic, but also by other programming



languages, such as Java and C++. The interface between the API and various programming languages is provided by something called **Universal Network Objects** (UNO).

Since LibreOffice Basic is a procedural programming language, several linguistic constructs have had to be added to it which enable the use of UNO.

To use a Universal Network Object in LibreOffice Basic, you will need a variable declaration for the associated object. The declaration is made using the `Dim` instruction (see The Language of LibreOffice Basic). The `Object` type designation should be used to declare an object variable:

```
Dim Obj As Object
```

The call declares an object variable named `Obj`.

The object variable created must then be initialized so that it can be used. This can be done using the `createUnoService` function:

```
Obj = createUnoService("com.sun.star.frame.Desktop")
```

This call assigns to the `Obj` variable a reference to the newly created object. `com.sun.star.frame.Desktop` resembles an object type; however in UNO terminology it is called a service rather than a type. In accordance with UNO philosophy, an Obj is described as a reference to an object which supports the

```
com.sun.star.frame.Desktop
```

service. The service term used in LibreOffice Basic therefore corresponds to the type and class terms used in other programming languages.

There is, however, one main difference: a Universal Network Object may support several services at the same time. Some UNO services in turn support other services so that, through one object, you are provided with a whole range of services. For example, that the aforementioned object, which is based on the

```
com.sun.star.frame.Desktop
```

service, can also include other services for loading documents and for ending the program.

ACCESS2BASE

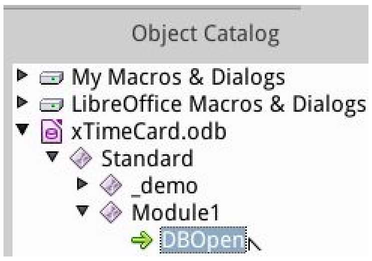
The extension Access2Base already available from LibreOffice 4.2, may be started by adding few lines of BASIC code in the opening of every database file. This gives the advanced users and developers the ability to use a simplified language in the style of Microsoft Access to perform complex task without bothering oneself with UNO interface.

ACTIVATE API

To be able to invoke the *Access2Base* API from your database file ".odb", you have to add few lines of code from the BASIC IDE into the a module of your database file.

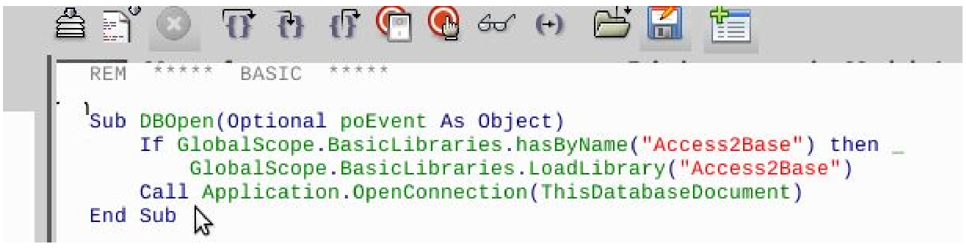
DBOPEN

Form menu Tools → Macros → Organize Macros → Open/LibreOffice Basic add the macro “DBOpen” to your main module, if you never created one is the right moment to add “Module1” to your database file.



```
Sub DBOpen(Optional poEvent As Object)
    If GlobalScope.BasicLibraries.hasByName("Access2Base") then _
        GlobalScope.BasicLibraries.LoadLibrary("Access2Base")
    Call Application.OpenConnection(ThisDatabaseDocument)
End Sub
```

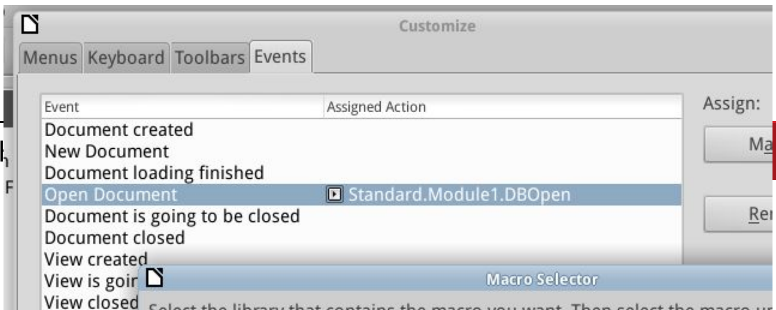
Create a module for your database application called for example “Module1” and edit to add the code; eventually you will end-up with a new macro called “DBOpen”.



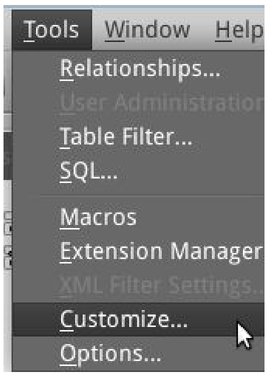
```
REM  *****  BASIC  *****

Sub DBOpen(Optional poEvent As Object)
    If GlobalScope.BasicLibraries.hasByName("Access2Base") then _
        GlobalScope.BasicLibraries.LoadLibrary("Access2Base")
    Call Application.OpenConnection(ThisDatabaseDocument)
End Sub
```

Assign in the main Base window with menu items Tools → Customize... (Events tab) the above Sub ("DBOpen" in the example but use the name of your choice) to the *OpenDocument* event. Save in the ".odb" file itself.



Event	Assigned Action
Document created	
New Document	
Document loading finished	
Open Document	Standard.Module1.DBOpen
Document is going to be closed	
Document closed	
View created	
View is goir	
View closed	



Close and re-open your database file ".odb", this will trigger the *OpenDocument* event and start programming macro's with Access2Base dialect. **Repeat for every new file to activate Access2Base API.**

TEST

Before creating cool things like a dashboard for our application or calculate values and update within the form, we need to check if the library works properly. For instance let's add the ability to launch a new form, copy "_test" into "....".

[..]