

# What is a BusinessObject?

Any data you wish to store in your App will be stored in one or more data-tables. These are surfaced into your App, making them visible and usable by turning them into "BusinessObjects".

Converting a data-table into a BusinessObject is done by adding records to the "BusinessObjects," "Resources" and "Searches" framework data-tables. The "BusinessObjects" record should have a "Name" which is the same as the name of the new data-table, Orixia uses this correspondence when it builds the App. Other fields in the BusinessObjects system-table are set so that the App can understand what it should show. Searches and Resources can be added with the "LinkTable" column set to the name of the data-table. Again, Orixia uses this correspondence to associate the Searches and Resources with the BusinessObject / data-table(s).

For the App to work well the data-table must be created following Orixia Framework data-table creation standards, and the SQL-statements in the BusinessObjects system-table must also be well-written. Failure to follow standards will result in bugs in your App.

Tools exist such as the "Create BusinessObject Form" to help in this process and make the creation of new BusinessObjects easier.

Once a BusinessObject has been created, when your Orixia App starts a new Entity will be added to the System Entities Screen, with some basic standard features plus any of the optional features or extensions you have chosen to include, such as searches, reports, dashboards, child-entities etc.

You will be able to open Edit Forms to edit and update the data in the new BusinessObject, view records from the data-table in Grids and other Orixia user-interfaces such as the diary or worksurfaces, include data from the new BusinessObject in reports, charts, dashboards and so on.

## Database-level definitions

As well as settings from the Orixia Framework tables, Orixia also reads a lot of **meta data** from the data-definition of the data-table and uses this when it builds your App. This includes basic aspects of the data-table, such as the data-type of COLUMNS, but also **decorations** which are additional DESCRIPTIONS added to parts of the data-table definition. Both Framework-table and Data-Schema-Decoration are used for specific reasons, which are a bit too complicated to explain here. Overall efforts have been made to try to place as much control of the App as possible "close" to the data, ie within the database structure definition, however in a number of cases this is either not practical or not desirable, so Framework-tables are used instead.

## Settings of the BusinessObject

The screenshot shows the 'BusinessObjects data' edit form for a 'People' entity. The form includes the following fields and options:

- Name:** People
- BusObjType:** M
- Left:** 7
- Top:** 2
- Color:** (Green)
- Icon:** (Person icon)
- SecurityLevel:** 0

**System Scripts:**

- Buttons: ListScript, DisplayScript, ViewScript, SummaryScript, DiaryScript, Settings
- SQL Script:

```
1 SELECT
2   P.ID,
3   P.FirstName,
4   P.LastName,
5   O.Name as Organisation,
6   P.JobDescription + COALESCE(' (' + T.Name + ')', '') as PersonType,
7   P.DateCreated,
8   P.Current
9 FROM
10  People P
11 LEFT JOIN Organisations O ON (P.OrganisationsID = O.ID)
12 LEFT JOIN Types T ON (P.PersonTypeID = T.ID)
13
14
```

**Linking and Settings:**

- LinkToPhones:
- LinkToEmailAddresses:
- LinkToImages:
- LinkToAddresses:
- AddNewButton:
- LinkToComments:
- AcceptsLinks:
- CreatesLinks:

**Date Edited:** 20/08/2021 19:54:10

**Date Created:** 09/12/2019 09:53:48

**Current:**

BusinessObjects edit form

The above image shows the BusinessObject editing screen. Note that when a BusinessObject is newly created it is usual to use Orixa tools to create it. Once a BusinessObject has been created, it's properties can be edited in the above screen to change values for SQL.

## Relational Data Structures

While it is possible that you may be able to add a BusinessObject to an App with just a single data-table, it is more likely that your new BusinessObject will be an Entity, created by linking together a number of BusinessObjects.

Take for example an App designed to show Sales of Products to Customers. In a simple world the programmer might try to use a single data-table to create such an App. In fact, for the App to be well designed a number of data-tables are needed these might include, but would not be limited to: Customers, Suppliers, Products, Sales and SalesItems. Each of these data-tables would also have an entry in the "BusinessObjects system-table, and the linkages between them would be used to build the entities that form the App.

Understanding how all this works is difficult, but it is always possible to use existing templates as a base for extending and building your own system.

Orixa uses well understood concepts from Relational Database Design as its standard. Programmers with a background in relational databases will find this fairly easy to understand. Orixa also has a "Links" system-table, which can be used to create promiscuous linkages between otherwise unrelated data-tables. This "Links" concept is closer in its operation to Graph-database design. Orixa allows a fluid mix of these different modes of inter-connection between data in an app.

Details of how to structure relational BusinessObjects are included in this Help Article:

[Relational Data in Orixa](#)