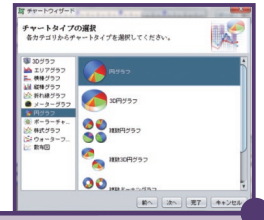


# 「Elixir Report」を使って クラウド上のデータを グラフ化してみよう



GrapeCity®

OSDN Magazine

最近ではSalesforceなどのクラウドサービスを使ってデータを管理しているケースも少なくない。今回は「Elixir Report」を使って、このようなクラウド上にあるデータを表やグラフといった形で出力する方法を紹介する。

## クラウド上のデータをデータソースとして利用する

「Elixir Report」は、さまざまなデータを元に表やグラフを作成し、それをPDFや

画像ファイル、PowerPoint、Excelなどの形式で出力できる帳票ツールだ(図1)。

Elixir Reportの特徴の1つに、さまざまなデータソースから表やグラフの作成に必要なデータを取得できる点がある。表やグラフを作成する際、その元データはデータベース内に格納されていることが多い。Elixir Reportではデータベース内のデータにJDBC (Java向けのデータベース接続API) 経由でアクセスできるが、それ以外にもExcelやXML、テキスト (CSVやTSVなど)

といった形式のファイルやJavaオブジェクトへのアクセスが可能になっている(表1)。

Excelやテキスト、XMLといったファイルをデータソースとして利用する場合、そのファイルは必ずしもローカルマシン上にある必要はなく、HTTPなどを使ってリモートサーバー上にあるファイルを動的に取得して使用することが可能だ。これを利用すれば、既存のほかのアプリケーションが出力するデータを表やグラフ化するシステムを容易に実装できる。

また、Elixir Reportでデータソースとして利用できるJDBCは、サードパーティからさまざまなドライバが提供されている。これらのドライバは、MySQLやPostgreSQL、Oracleといったデータベースにアクセスするものだけに限られず、たとえば特定の形式のファイル内のデータにアクセスしたり、さまざまなアプリケーションやシステムにアクセスするためのものもある。これらを使用することで、さまざまなプラットフォーム内にあるデータをElixir Reportで扱えるようになる。

今回はElixir Reportでクラウド上のデータをドキュメント化する例として、クラウドサービスであるSalesforce上のデータにアクセスできるJDBCドライバ「CData (シーデータ) JDBC Driver for Salesforce 4J」を使ってSalesforce上のデータを表・グラフ化するというものを紹介する。

## Salesforce上のデータをJavaアプリケーションから利用できる「CData JDBC Driver for Salesforce 4J」

今回使用するCData JDBC Driver for Salesforce 4J (以下、JDBC Driver for Salesforce) は、Salesforce上のデータにアクセスするためのAPIを提供するJava用のJDBCドライバだ。JDBC Driver for Salesforceでは、Javaで広く使われているJDBC APIを使用して、Salesforce上にある取引先データベースや商談データベースな

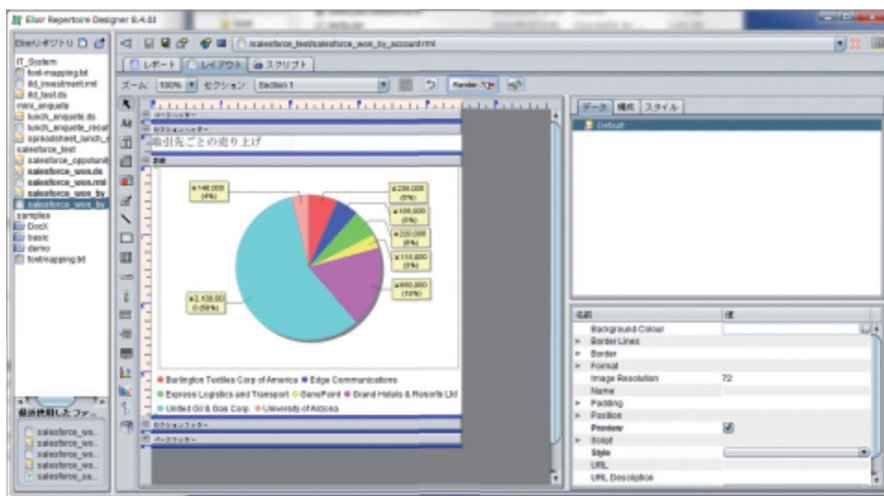


図1 Elixir Reportのレポートデザイナー画面

表1 Elixir Reportでサポートされている主なデータソース

データソース名	対象とするデータなど
JDBC	JDBC (Javaで広く使われているデータベースアクセスインターフェイス) に対応するデータベース内のデータ
Excel	Microsoft Excel (XLSおよびXSLX形式) ファイル内のデータ
テキスト	CSVやTSVといったテキストベースのファイル内のデータ
XML	XMLファイル内のデータ
オブジェクトデータソース	Javaオブジェクト内のデータ
表形式	レポートデザイナー内で作成された表形式のデータ
Google スプレッドシート	Google スプレッドシート上のデータ
ランダム	指定されたフォーマットに従ってランダムなデータを生成して使用する
キャッシュ	指定されたデータソースをキャッシュして使用する
プロパティ	レポートデザイナー内で作成された表形式のデータ。コンポジットデータソースと組み合わせて使用する
リファレンス	別のデータソースを参照する
Google スプレッドシート	Google スプレッドシート上のデータ

どにSQLでアクセスできる。データの読み取りだけでなく、作成や更新、削除といった操作もサポートされている。

JDBC Driver for SalesforceはJava (JDK 5.0~8.0) が動作する環境であれば、プラットフォームに関係なく動作する。評価目的で無償で利用できるトライアル版も提供されているので、興味のある方はまずはこちらを試してみることをおすすめする。

**今回作成するサンプル**

今回は、Salesforce上のデータを表およびグラフ形式で出力する例を紹介する。Elixir Reportについては『コードを記述することなしにデータを表やグラフとして出力できる帳票ツール「Elixir Report」』記事で紹介しているので、そちらを参照してほしい。

各種作業では、Elixir Reportのレポートデザイナーおよびレポートサーバーの全機能を制限無しで利用できる無償のトライアル版を使用した。トライアル版のライセンス申請はGrapeCityのElixir Reportページから行える。また、作業に使用したOSはWindows 7 Professional (64ビット) だ。

**「CData JDBC Driver for Salesforce 4J」を使ってSalesforce上のデータをレポートとして出力する**

近年、ビジネスの現場での顧客関係管理 (CRM) ツールとしてよく利用されているのがSalesforceだ。Salesforceはセールスフォース・ドットコムが提供しているクラウドサービスの総称で、営業支援ツール「Sales Cloud」やサポート支援ツール「Service Cloud」、マーケティングツール「Marketing Cloud」といったビジネスアプリケーション、そしてさまざまなアプリケーションを構築できる「Salesforce Platform」などが提供されている (図2)。



図2 Salesforceの「レポート&ダッシュボード」画面

Salesforceで使用するデータベースや各

種アプリケーションサーバーはすべてSalesforce側で管理されており、Webブラウザやクライアントアプリケーション経由でこれらにアクセスすることになる。そのため、ユーザーはこのようなインフラの管理を行うことなしにアプリケーションを利用できる。

今回使用するJDBC Driver for Salesforceは、このSalesforce上のデータにアクセスできるJDBCドライバだ。以下ではこれを利用し、Salesforce上にある商談データにアクセスして、取引先ごとの売り上げをグラフ形式で表示するレポートを作成する流れを紹介する。

また、Salesforce自体は有償のサービスだが、開発者向けに無料で利用できる「Force.com Developer Edition」も提供されている。今回はこちらを利用してテストを行った。商談データおよび顧客データとしては、あらかじめ用意されているサンプルデータを使用している。Force.com Developer Editionは、Salesforceの開発者向けサイトの「サインアップ」から登録が行える。

**JDBC Driver for Salesforceのインストールと設定**

JDBC Driver for Salesforceは、ZIP形式のアーカイブで配布されている。アーカイブを展開し、含まれているJAR形式のインストーラを実行するとインストールが開始される。Windowsの場合、Javaランタイム (JRE) がインストールされていればJARファイルをダブルクリックするだけで実行できるはずだ。

インストーラはウィザード形式となっており、指示に従って操作していくことでインストーラが完了する (図3)。

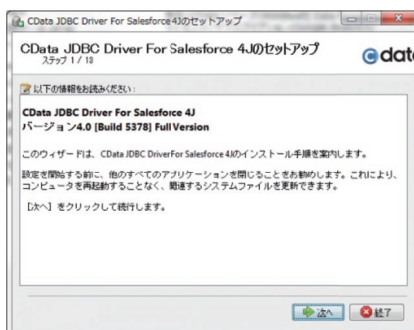


図3 JDBC Driver for Salesforceのインストーラ

インストール時にこのインストーラ上でプロダクトキーの入力が必要となるので、あらかじめ確認しておこう (図4)。なお、評価版ではプロダクトキーの入力は不要で、ライセンスアクティベーション画面自体も表示されない。

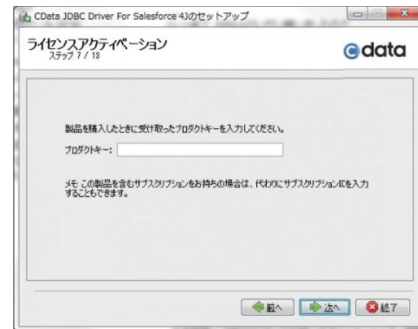


図4 「ライセンスアクティベーション」画面

JDBC Driver for Salesforceはデフォルトでは「C:\Program Files\CData\CData JDBC Driver for Salesforce 4J」というディレクトリにインストールされる。また、JDBCドライバ本体はインストールディレクトリ内の「lib」ディレクトリ以下にライセンスファイルなどとともに格納される。JDBC Driver for Salesforceを使用する際は、アプリケーションからこれらのファイルにアクセスできるように設定を行う必要がある。

Elixir Reportの場合、インストールディレクトリ (デフォルトでは「C:\Program Files\ElixirReport8.4JT」) 直下の「ext」ディレクトリ内にこれらのファイルをすべてコピーすることで、JDBC Driver for Salesforceが利用できるようになる (図5)。

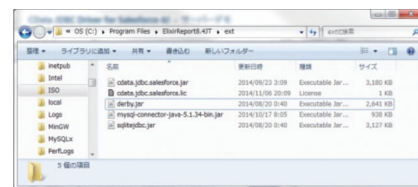


図5 JDBC Driver for SalesforceのjarファイルなどをElixir Reportのインストールディレクトリ以下にあるextディレクトリにコピーすることで、ElixirからJDBC Driver for Salesforceを利用できるようになる

**JDBC Driver for Salesforceをデータソースとして利用する**

JDBC Driver for Salesforceのインストールが完了したら、続いてElixir Reportのレポートデザイナーを起動し、データソースとして利用する設定を行って行く。まずレポートデザイナーでデータソースを作成し、データソースウィザードでデータソースタイプとして「JDBC」を選択する (図6)。

次に、JDBCデータソースのパラメータを入力する画面が表示されるので、ここで作成するデータソースの名前や説明、ドライバの指定などを行って行く (図7)。

ここで、「名前」および「説明」は任意の適当な文字列で構わない。今回はそれぞれ「salesforce\_opportunity」、「Salesforce・

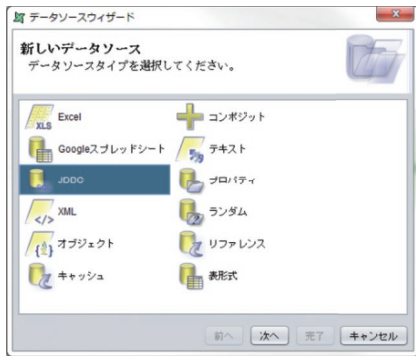


図6 データソースウィザードの「新しいデータソース」画面

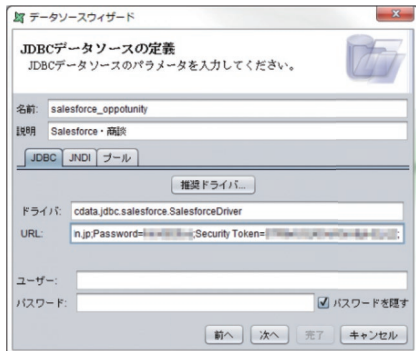


図7 「JDBC データソースの定義」画面

商談」とした。「ドライバ」には「cdata.jdbc.salesforce.SalesforceDriver」を指定する。また、「URL」は以下ようになる。

```
jdbc:salesforce:User=<ユーザー名>;Password=<パスワード>;SecurityToken=<セキュリティトークン>;
```

ここで、「ユーザー名」および「パスワード」にはSalesforceに登録しているユーザー名およびパスワードを、「セキュリティトークン」にはSalesforceから取得したセキュリティトークンを指定する。セキュリティトークンの取得方法はSalesforceのヘルプドキュメントに記載されているのでそちらを参照してほしい。

また、「ユーザー」および「パスワード」欄は空白のまま構わない。これらの入力が完了したら「次へ」をクリックし、「JDBCデータソースの定義」画面を表示されたら「クエリビルダー」ボタンをクリックする(図8)。



図8 「JDBC データソースの定義」画面

Salesforceへの接続が成功すれば、このクエリビルダー画面の左ペインにテーブル情報が表示されるはずだ(図9)。

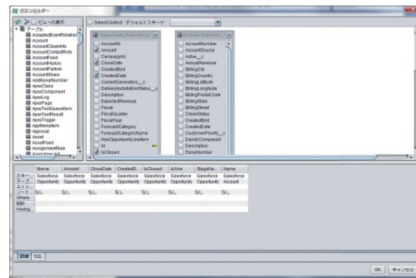


図9 クエリビルダー画面

あとは一般的なデータベースをデータソースとして使用する場合と同様、データが格納されているデータベースやカラム名を指定してクエリを作成すれば良い。なお、Salesforceの管理画面では各データの項目名が日本語で表示されるが、JDBC Driver for Salesforceではこれらに対応するテーブル名およびカラム名が英語で表示される点には注意したい。また、テーブル名/カラム名や各データの項目名一覧はJDBC Driver for Salesforceのヘルプで確認できる(図10)。

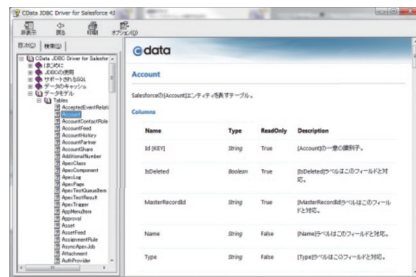


図10 JDBC Driver for Salesforceのヘルプ

今回は、商談データベース(JDBC Driver for Salesforce上では「Opportunity」テーブルに相当)および取引先データベース(「Account」テーブルに相当)に格納されている、表2の情報をクエリで取り出している。

表2 今回のクエリで取り出す

テーブル名	カラム名	説明
Opportunity	Name	商談名
Opportunity	Amount	金額
Opportunity	CloseDate	成立日
Opportunity	CreatedDate	作成日
Opportunity	IsClosed	商談が完了しているか
Opportunity	IsWon	契約を獲得できているか
Opportunity	StageName	商談の段階
Account	Name	取引先名

ここで使用するクエリは以下ようになる。

```
SELECT
Opportunity.Name, Opportunity.
Amount, Opportunity.CloseDate,
Opportunity.CreatedDate,
Opportunity.IsClosed, Opportunity.
IsWon,
Opportunity.StageName, Account.Name
FROM Salesforce.Opportunity,
Salesforce.Account
```

なお、Salesforceではあらかじめテーブルおよびカラム同士の関連性が設定されているため、複数のテーブルに対してクエリを実行した場合は自動的に対応するデータが結合されて出力される。このとき、結合は左側のテーブルを優先させた外部結合(LEFT JOIN)のようになる。今回は商談データベースを優先させるため、FROM節では次のように「Salesforce.Opportunity」を左側に書く必要がある。

```
FROM Salesforce.Opportunity,
Salesforce.Account
```

クエリビルダーを使ってSQLを作成した場合、カラム名やテーブル名といった順序が入れ替わっている場合があるので注意が必要だ。その場合、手動でSQL文を編集する必要がある。

クエリの作成後、「JDBCデータソースの定義」画面で「更新」ボタンをクリックし、データソースの名前と型を確認する(図11)。



図11 「JDBC データソースの定義」画面

ここで、「Name」というカラムはOpportunityテーブルおよびAccountテーブルの両方に含まれているので、片方のカラム名が「Name1」というものに変更されている点に注意したい。最後に「完了」ボタンをクリックすると、データソースが作成される。このデータソースで抽出されたデータは図12のようになる。

コンポジットデータソースを使って取得したデータを集計・加工する

続いて、取得したデータから取引先ごと

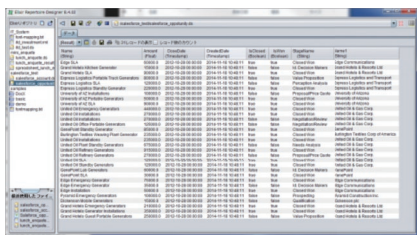


図12 作成したデータソースで抽出されたデータ

の契約金額を集計するための作業を行っていく。Salesforceの取引データベースには、「IsWon」という名前のカラムがあり、これが「true」であればその商談を獲得できていることになる。以下ではElixir Reportのコンポジットデータソース機能を使用し、「IsWon」が「true」のカラムを対象に、取引先ごとの合計金額を集計してみよう。

コンポジットデータソースは、指定したデータソースから取得したデータに対しフィルタやソート、フィールド(カラム)の追加といった操作を行い、その結果を新たなデータソースとして提供する機能だ。レポートデザイン上ではフローチャートのような画面でデータに対し適用する処理を指定することで、コンポジットデータソースを作成できる(図13)。

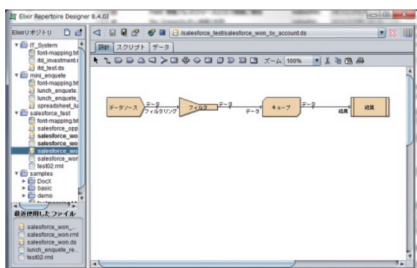


図13 コンポジットデータソースの「設計」画面

コンポジットデータソースを作成するには、データソースウィザードの「新しいデータソース」画面で「コンポジット」を選択する(図14)。今回作成するデータソース名は「salesforce\_won\_by\_account」、説明には「Salesforce・取引先ごとの売り上げ」とした。

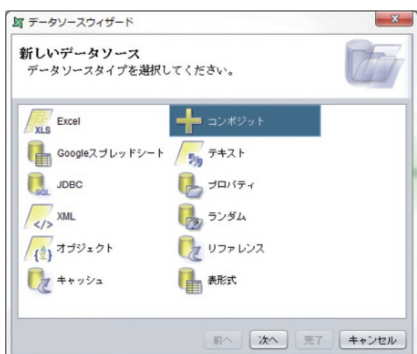


図14 「新しいデータソース」画面

コンポジットデータソースを作成すると、データをどのように処理するかを図で指定できる「設計」画面が表示される。ここでは、ツールバーから「データソース」や「フィルタ」といった要素を選択して画面をクリックすることでこれらの要素を配置し、要素同士を線(フロー)で接続することで、処理を定義できる。今回は、先ほど作成した「salesforce\_opportunity」データソースを元に、「フィルタ」を使って「IsWon」カラムが「true」のものを取り出し、さらに「キューブ」要素を使って集計を行ったものを結果として出力することとする。

設計画面では、配置した要素をダブルクリックすることでその設定を行える。まずデータソースでは、使用するデータソースとして「salesforce\_opportunity」を指定する(図15)。

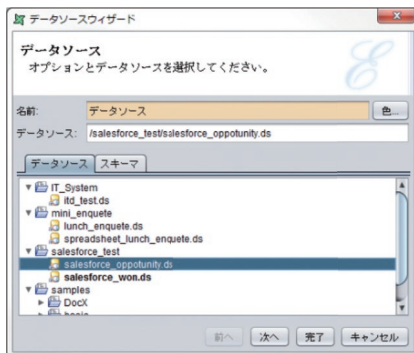


図15 データソースの設定

また、フィルタの設定では「IsWon」に対し、条件として「等しい」を意味する「Equals」を、比較する値に「true」を指定した(図16)。



図16 フィルタの設定

最後のキューブは、入力されたデータに対し、行/列を対象とした集計処理を行うものだ。今回は、取引先名ごとに行を作成し、取引先名とその取引金額合計という列を持つデータを出力することにする。

キューブの設定では、まず「キューブ階層」

を指定する画面が表示される。今回は階層化をしないので、ここは空のまま「次へ」をクリックする(図17)。

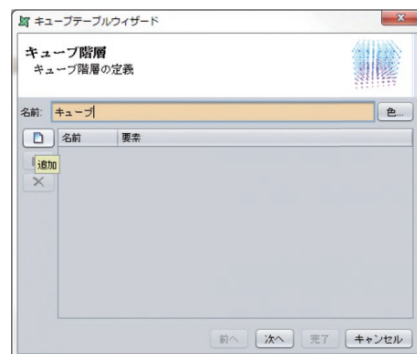


図17 「キューブ階層」画面

次の「キューブ軸」画面では、行、もしくは列として使用する項目を指定する。今回は取引先名ごとに行を作成するので、「Name1」フィールドをテーブル行に指定する(図18)。

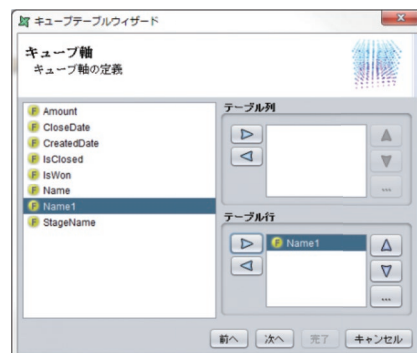


図18 「キューブ軸」画面

「キューブの指標」画面では、集計対象とするフィールドと集計方法を指定する。今回は取引金額が格納されている「Amount」フィールドに対し、それぞれの合計を出力する「合計値」演算を適用することにする(図19)。

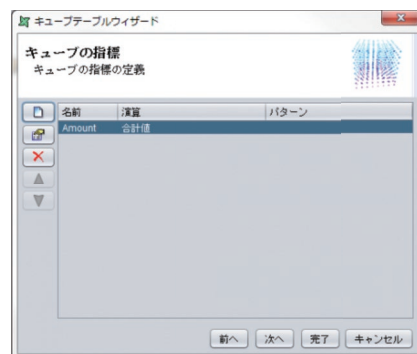


図19 「キューブの指標」画面

なお、指標を追加するには左側の新規作成アイコンをクリックし、「指標の追加」ウィンドウで「名前」と「演算」を指定すれば良

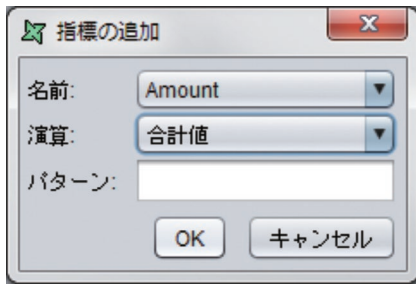


図20 「指標の追加」ウィンドウ

い(図20)。

次の「キューブオプション」画面では特に設定は不要だ。最後の「キューブスキーマのキャッシュ」で「更新」をクリックし、作成されるフィールドを確認して「完了」をクリックする(図21)。

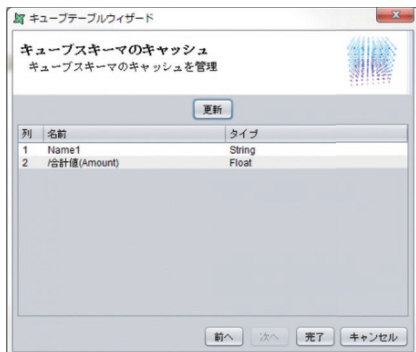


図21 「キューブスキーマのキャッシュ」画面

以上の作業が完了したら、「データ」タブをクリックし、最終的に出力されるデータをプレビューする。今回は「Name1」と「/合計値(Amount)」というフィールドが作成され、各行に取引先名とその取引金額合計が表示されていることが確認できる(図22)。

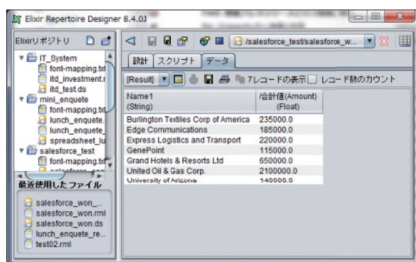


図22 「データ」タブで出力されるデータをプレビューできる

また、同様の手順で契約が成立した商談のみを出力するデータソース「salesforce\_won」も作成した(図23)。こちらは、先に作成した「salesforce\_won\_by\_account」コンポジットデータソースからキューブ集計処理を取り除き、「IsWon」カラムがtrueのデータのみを出力するものだ。

レポートテンプレートの作成

続いて、データソースを元に作成するレポー

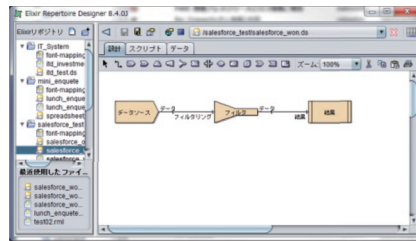


図23 「salesforce\_won」コンポジットデータソース

トのデザインを決めるレポートテンプレートを作成していく。今回は、「salesforce\_won」データソースを元に、表形式で成立した契約のみを出力する「salesforce\_won」レポートテンプレートと、「salesforce\_won\_by\_account」データソースを元に円グラフで取引先ごとの契約金額合計を出力する「salesforce\_won\_by\_account」レポートテンプレートを作成する。

まず表形式で成立した契約のみを出力する「salesforce\_won」レポートテンプレートだが、こちらはレポートウィザードで「表形式レポート」を選択して作成した(図24)。

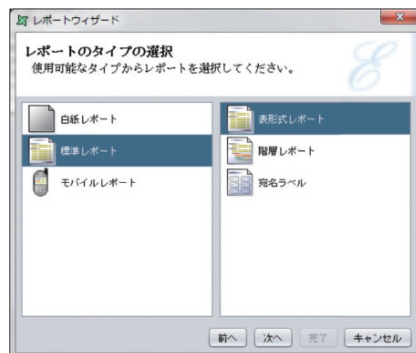


図24 レポートウィザードで「表形式レポート」を選択

表に出力するフィールドには「Name」(商談名)および「Name1」(取引先)、「Amount」(金額)、「CloseDate」(成立日)を指定した(図25)。

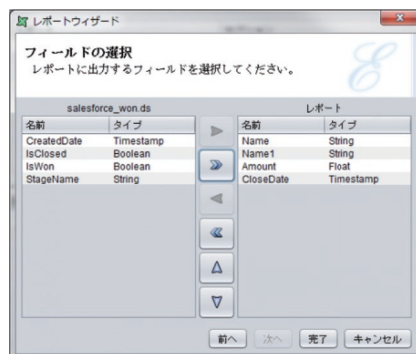


図25 「フィールドの選択」画面

以上の設定で生成されたレポートテンプレートは図26のようにになる。

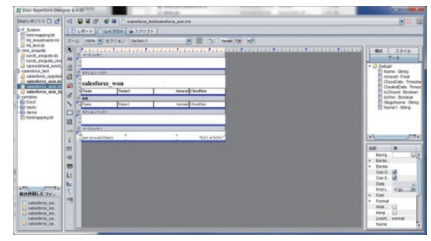


図26 出力されたレポートテンプレート

これにフォントサイズや要素の幅に関する若干の調整を加えたものが図27だ。

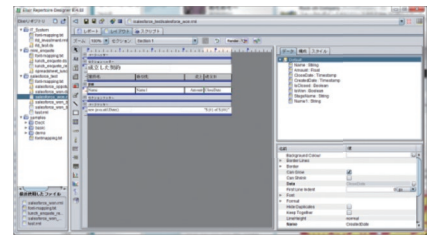


図27 完成したレポートテンプレート

このレポートテンプレートを使って実際に表を出力すると、図28のようになる。

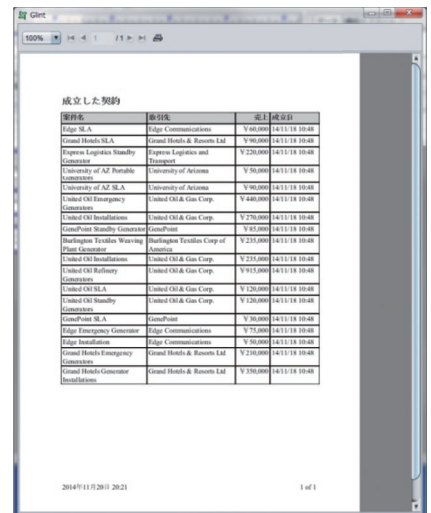


図28 レポートの出力例

続いて取引先ごとの契約金額合計を出力するレポートテンプレートだが、こちらはレポートのタイプとして「白紙レポート」を選択し、空のレポートテンプレートをまず生成する。続いて「セクションヘッダー」内にレポートのタイトル(「取引先ごとの売り上げ」)を入力したテキストエリアと、「チャート」要素を配置し、チャートの設定を行っていく。

チャート要素を配置すると、「チャートウィザード」が表示されるので、続いてチャートの設定を行っていく。まず「データソースの選択」画面では使用するデータソースを選択する(図29)。ここではデフォルトの「<セクションデータソース>」のままで良い。

「チャートタイプの選択」画面では、「円

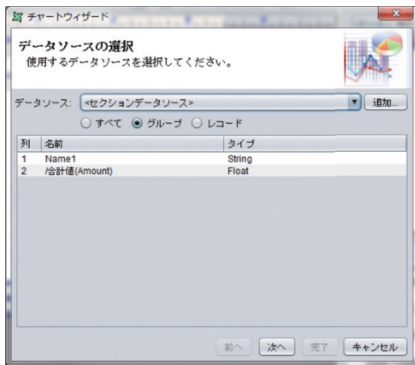


図29 「データソースの選択」画面

グラフ」を指定する(図30)。

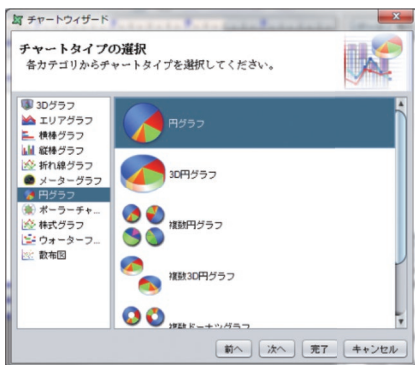


図30 「チャートタイプの選択」画面

続いてデータを指定する「円グラフ」画面となる。まず「キー」タブでキーとして使用するフィールドに「Name1」を、「値」タブで値として使用するフィールドに「/合計値(Amount)」を指定する(図31、32)。

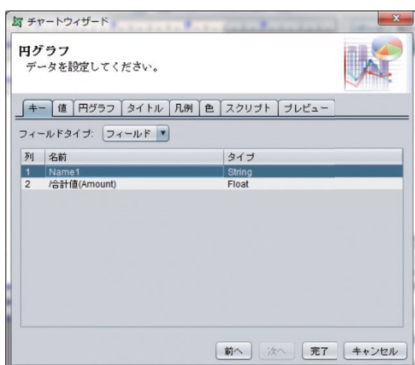


図31 「円グラフ」画面の「キー」タブ

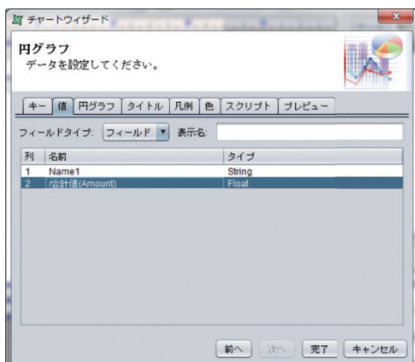


図32 「円グラフ」画面の「値」タブ

これで「プレビュー」タブを選択すると、指定したキーと値に応じたグラフがプレビュー表示される(図33)。

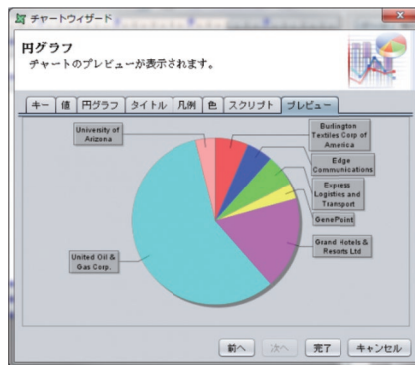


図33 「円グラフ」画面の「プレビュー」タブ

「判例」や「円グラフ」タブで円グラフや判例の表示方法も設定可能だ。これらを調整し、最終的に完成したレポートテンプレートは図34のようにになる。

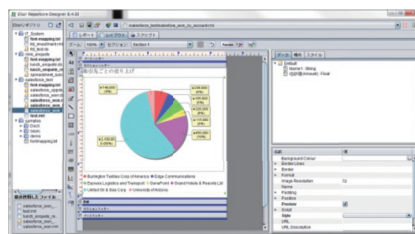


図34 完成したレポートテンプレート

このテンプレートを使って実際に出力したレポートは図35のようにになる。

あとはレポートサーバーにデータソースやレポートテンプレートが格納されたディレクトリを登録すれば、任意のアプリケーションからこの表やグラフが出力できるようになる。これらの作業については、『コードを記述することなしにデータを表やグラフとして出力できる帳票ツール「Elixir Report」』記事で紹介しているので、そちらを参照してほしい。

**SQLデータベースを使用するのとはほぼ同じ流れでクラウド上のデータも処理可能に**

さて、今回はSalesforce上にあるデータを元にレポートを作成する例を紹介した。Salesforce上にあるデータの参照には内部

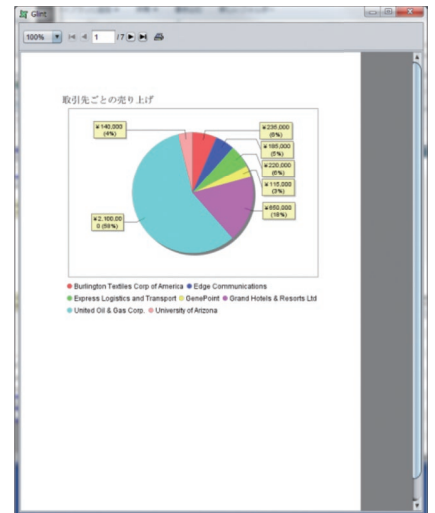


図35 出力されたレポート例

的にはSalesforceが提供しているAPIを利用しているのだが、JDBC Driver for Salesforceという外部ツールを併用することで、特にプログラミングを行うことなく、データベースをデータソースとして使用した場合と同じようにデータにアクセスできた。SalesforceのAPIに関する知識がなくても、SQLの知識だけでデータを利用できるのは非常に便利だろう。

また、データベース内のデータを集計しようとする場合、従来は個別にプログラムを記述するか、もしくは複雑なSQL文を組み合わせる必要があった。しかし、Elixir Reportのコンポジットデータソース機能を利用すれば、データの集計作業も簡単に実行できる。ただし、集計処理はレポートサーバー側で行われるため、処理内容によってはレポートサーバー側の負荷が大きくなる可能性がある点には注意したい。

さて、今回はデータソースやレポートテンプレートについて注目したが、実際にElixir Reportを利用する場合は、クライアントアプリケーションからレポートサーバーにアクセスしてレポートの生成・表示を行うことになる。今回はこのクライアントアプリケーション側から見た、Elixir Reportを利用するための方法を紹介する。

**製品購入やライセンスについてのお問い合わせ**

**グレイシティ株式会社 ツール事業部 営業部**

TEL : 048-222-3001

営業時間 : 月~金 9:00~12:00  
13:00~18:00

E-mail : sales@grapecity.com

Elixir Reportのトライアル版のお申し込み  
http://www.grapecity.com/tools/products/elixirreport84