

4. Inicjalizacja projektu - instalacja bazy danych i źródła ERP

Widzieliśmy wiele bardzo teoretycznych aspektów BI. Zaczęliśmy od ogólnego wprowadzenia, części poświęconej zarządzaniu projektami, a także części poświęconej wprowadzeniu do SQL. Ale koniec końców, do tej pory nie zrobiliśmy nic praktycznego, więc czas zabrać się za rozwój rozwiązania. Zanim jednak zaczniemy, musimy zainstalować nasz źródłowy ERP, skąd będą pochodzić dane. Po zainstalowaniu i skonfigurowaniu jeszcze trochę pokombinujemy, a następnie wybierzemy naszą bazę danych do przechowywania hurtowni danych.

Potrzeba danych

Wszystkie systemy BI wymagają pewnego źródła danych. Dane te mogą być ustrukturyzowane lub nieustrukturyzowane. Ponadto dane te mogą pochodzić z wielu źródeł i mieć wiele kształtów. Ponadto możemy napotkać pewne problemy z jakością danych, ponieważ możemy mieć brakujące i/lub nadmiarowe dane. Ale zawsze istnieje punkt wyjścia, w którym generowane są dane źródłowe. Ten tekst nauczy Cię, jak wyodrębniać, przekształcać, ładować i raportować swoje dane. Ale jest pierwszy wstępny krok, który polega na pozyskiwaniu tych danych. W tej książce zdecydowaliśmy się użyć Odoo (wcześniej znanego jako OpenERP) jako źródła naszych danych. Tak, korzystanie z dowolnych dostępnych bezpłatnie przykładowych schematów baz danych dostępnych w Internecie lub dostarczanych w pakiecie z prawie każdą bazą danych mogło być łatwiejsze. I tak znowu, użyjemy niektórych, aby pokazać ci kilka zaawansowanych punktów w dalszej części książki, takich jak indeksowanie i tak dalej. Ale chcemy pokazać prawdziwą aplikację i kompletny potok danych. A w prawdziwym życiu zazwyczaj oznacza to połączenie z systemem ERP. Większość projektów BI zaczyna pozyskiwać dane z SAP. SAP nie jest jednak przeznaczony dla małych i średnich firm. Złożoność systemu jest przytłaczająca, a na domiar złego ceny licencji są wysokie. Jest więc prawdopodobne, że będziesz musiał celować w inne systemy ERP. Jest ich wiele w przystępnej cenie, większość z nich działa nawet w chmurze, a płacisz na bieżąco (głównie na podstawie liczby użytkowników korzystających z nich lub zakupionych modułów lub na podstawie użycia). Uważamy jednak, że istnieją dobre projekty open source, dostępne za darmo, które mogą odpowiadać potrzebom Twojej firmy.

Konfigurowanie systemu za pomocą Odoo ERP

Jeśli masz już zainstalowany ERP w swojej firmie lub używasz jakiegoś programu do zarządzania klientami, wystawiania faktur itp. Gratulacje! masz już swoje źródło danych. Źródłem danych będzie rzeczywisty ERP lub rzeczywista transakcyjna baza danych, w której program lub programy, z których korzystasz, gromadzą dane. Dla tych, którzy jeszcze się nad tym zastanawiają, w tej części przedstawiamy, jak zainstalować i bawić się Odoo, aby zbierać dane do rozwoju naszego systemu BI. Jak wyjaśniono we wstępie, do zainstalowania Odoo użyjemy komputera z systemem Linux. Zdecydowaliśmy się na serwer Ubuntu w wersji 16.04 LTS. Dzieje się tak dlatego, że jest to wersja Long Term Support (LTS), co oznacza, że będzie obsługiwana przez dłuższy czas i powinna być bardziej stabilna. Instalacja Odoo może być nieco skomplikowana ze względu na różne wymagania wstępne, więc jest to raczej zadanie dla administratora systemu. Jeśli go nie masz lub nie możesz sobie pozwolić na jego wynajęcie, zalecamy skorzystanie z uproszczonej wersji, którą przedstawimy w kolejnych akapitach. Jeśli chcesz zainstalować go od zera, przed instalacją upewnij się, że zainstalowałeś bazę danych PostgreSQL, serwer WWW Apache i że wersja Pythona jest zgodna z wersją wymaganą przez instalowaną wersję Odoo.

Pakiet Bitnami Odoo

Do naszego wdrożenia użyjemy samodzielnego instalatora Bitnami, który zainstaluje i skonfiguruje dla nas Odoo. Ten pakiet zawiera bazę danych PostgreSQL, serwer Apache i pliki programu. Dla tych, którzy

są bardziej zainteresowani wypróbowaniem go najpierw i nie chcą wdrażać pakietu na istniejącej maszynie, Bitnami oferuje również maszynę wirtualną ze wszystkim już zainstalowanym, w tym z serwerem Ubuntu. Aby pobrać pakiet Bitnami, otwórz przeglądarkę i sprawdź następujący adres URL: <https://bitnami.com/stack/odoo/installer>. Jeśli chcemy zainstalować pakiet samodzielnie, to musimy wybrać opcję instalatorów z paska menu. A następnie wybierz ten odpowiedni dla naszego systemu operacyjnego. Dla tych, którzy chcą najpierw wypróbować, zamiast wybierać opcję Instalatory w menu, wybierz Maszyny wirtualne i pobierz obraz Ubuntu VM zawierający zainstalowany program. Aby użytkownik i hasło w pakiecie mogli zalogować się do urządzenia, przeczytaj uważnie stronę .

Pobieranie i instalacja Odoo

Odtąd omówimy podstawy instalacji pakietu . Po znalezieniu odpowiedniej wersji uzyskamy adres URL. Ponownie, w momencie pisania link do pobrania jest następujący:

```
https://downloads.bitnami.com/files/stacks/Odoo/9.0.20160620-1/bitnami-Odoo-9.0.20160620-1-linux-x64-installer.run
```

Ale mogło się to zmienić, jeśli dostępna jest nowa wersja. Za pomocą tego adresu URL przejdziemy do naszego systemu Linux i użyjemy wget lub curl, aby pobrać instalator. Najpierw przechodzimy do katalogu /tmp, aby tam pobrać pakiet:

```
cd / tmp
```

Następnie wywołujemy wget, aby pobrać instalator:

```
wget
```

I rozpocznie się pobieranie. Po kilku sekundach lub minutach (w zależności od połączenia sieciowego) plik zostanie pobrany do urządzenia.

Przed uruchomieniem pliku musimy uczynić go wykonywalnym, więc robimy chmod

```
+x:
```

```
bibook@bibook:/tmp$ chmod +x bitnami-Odoo-9.0-3-linux-x64-installer.run
```

Tam uruchomimy instalator, uruchamiając następujące czynności, używając sudo do uruchomienia go jako root, ponieważ będziemy go potrzebować do zainstalowania go w innym katalogu

```
bibook@bibook:/tmp$ sudo ./bitnami-Odoo-9.0-3-linux-x64-installer.run
```

I postępujemy zgodnie z instrukcjami na ekranie. Po uruchomieniu instalator zasugeruje ścieżkę w katalogu domowym. Używamy /opt, aby go zainstalować, więc zmień go na wypadek, gdybyś go nie miał. Pod koniec procesu instalacji instalator zapyta nas, czy chcemy uruchomić usługi i komponenty Odoo. Wybieramy tak (Y).

Pliki konfiguracyjne Bitnami i Odoo

W tym podrozdziale przejrzymy miejsce na wszystkie pliki konfiguracyjne i skrypty startowe, abyśmy mogli kontrolować naszą instalację Odoo. W przypadku, gdy chcemy zatrzymać lub rozpocząć później na stosie, istnieje skrypt rządzący całą instalacją o nazwie ctlscrip.sh znajdujący się na górze ścieżki instalacyjnej, w naszym przypadku /opt/odoo-9.0-3. Uruchomienie skryptu jako użytkownik root pozwala nam uruchamiać i zatrzymywać powiązane usługi:

```

bibook@bibook:/opt/odoo-9.0-3$ sudo ./ctlscript.sh
usage: ./ctlscript.sh help
       ./ctlscript.sh (start|stop|restart|status)
       ./ctlscript.sh (start|stop|restart|status)
Postgresql
       ./ctlscript.sh (start|stop|restart|status)

Apache
       ./ctlscript.sh (start|stop|restart|status)
openerp_background_worker
       ./ctlscript.sh (start|stop|restart|status)
openerp_gevent

help      - this screen
start     - start the service(s)
stop      - stop  the service(s)
restart   - restart or start the service(s)
status    - show the status of the service(s)

```

Ale w tym momencie mamy już uruchomione usługi, więc nie musimy nic robić. Oprócz usług jest jeszcze jeden ważny plik, który należy znać, czyli domyślna konfiguracja usług, w tym porty, adresy URL, ścieżki lokalne i inne ważne rzeczy. Plik nazywa się `properties.ini` i ponownie potrzebujemy uprawnień administratora, aby go zobaczyć. Można go znaleźć w katalogu głównym. Domyślnie porty używane przez aplikację to standardowe porty używane przez zainstalowane usługi. W naszym przypadku serwer Apache nasłuchuje na porcie 80, podczas gdy baza danych PostgreSQL nasłuchuje na domyślnym porcie 5432. Oprócz tych plików konfiguracyjnych usługi istnieją inne ważne pliki. Być może najważniejszym z nich jest plik konfiguracyjny aplikacji, który można znaleźć w `/opt/odoo-9.0-3/apps/odoo/conf/openerp-server.conf`. Jest to ważny plik, ponieważ zawiera kilka losowo wygenerowanych haseł niezbędnych do połączenia się z aplikacją. W tym pliku będziemy mogli znaleźć nazwę użytkownika i hasło do bazy danych, nazwę bazy danych oraz hasło lub konto administratora. Wartości, które musimy zanotować z tego pliku, wraz z ich znaczeniem, znajdują się w tabeli

Wartość: Opis

`admin_passwd` : Domyślne hasło administratora do logowania w interfejsie WWW

`db_host` : Adres IP serwera bazy danych, którym będzie host lokalny

`db_name` : Nazwa bazy danych, domyślnie `bitnami_openerp`

`db_password` : domyślne hasło do bazy danych

`db_port` : domyślny port, 5432

`db_user` : użytkownik bazy danych, domyślnie `bn_openerp`

Instalowanie `psql` i sprawdzanie połączenia z bazą danych

Ostatnim zadaniem do wykonania przed połączeniem z programem jest instalacja klienta `psql`, dzięki czemu możemy uzyskać dostęp do bazy danych PostgreSQL z linii poleceń. Jeśli używamy Ubuntu, można to osiągnąć, uruchamiając następujące polecenia:

```
sudo apt-get install postgresql-client-common postgresql-clientv
```

Następnie możemy przetestować połączenie i sprawdzić, czy wszystko działa poprawnie, łącząc się z klientem psql:

```
bibook@bibook:/opt/Odoo-9.0-3/apps/Odoo/conf$ psql
```

```
-U bn_openerp -h host lokalny -d bitnami_openerp
```

Zostaniemy poproszeni o wartość db_password; wprowadzamy go i powinniśmy zobaczyć monit klienta psql. Jeśli do tego dojdziemy, wszystko jest w porządku.

Password for user bn_openerp:

```
psql (9.5.2, server 9.4.6)
```

```
Type "help" for help.
```

```
bitnami_openerp=#
```

Teraz skończyliśmy. Jesteśmy gotowi do odwiedzenia naszego nowo zainstalowanego ERP!

Uwaga: Jeśli próbujesz uzyskać dostęp do bazy danych z komputera zewnętrznego, musisz najpierw włączyć dostęp zdalny. W takim przypadku musisz zmodyfikować plik konfiguracyjny PostgreSQL, aby baza danych nasłuchiwała pod wszystkimi interfejsami (*) i zezwoliła użytkownikom na zdalne logowanie

Dostęp do aplikacji

Po wykonaniu wszystkich czynności konfiguracyjnych możemy zalogować się do aplikacji. Domyślnie możemy zalogować się za pomocą adresu URL hosta lokalnego: <http://localhost> Zostanie wyświetlony ekran powitalny z prośbą o podanie danych uwierzytelniających. W tym momencie musimy użyć nazwy użytkownika wybranej podczas instalacji oraz użyć wartości admin_passwd z pliku konfiguracyjnego openerp-server.conf. Po wejściu właśnie zalogowaliśmy się przy użyciu konta administratora.

Konfigurowanie i instalowanie modułów

Domyślnie instalacja konfiguruje nasze Odoo z włączonymi kilkoma modułami, ale może się zdarzyć, że będziemy chcieli je dostosować, wybierając moduły do zainstalowania, wybierając, czy chcemy zainstalować przykładowe dane i tak dalej. Jak widać w dolnej części obrazka, dostępna jest opcja Zarządzaj bazami danych, która pozwala nam rozpocząć nowy projekt Odoo od zera. Wybierzemy tę opcję, ponieważ pozwoli nam to założyć nową bazę danych Odoo i zainstalujemy moduł Sales, aby używać go jako przykładu podczas rezerwacji. Jednocześnie w nowej instalacji możemy polecić Odoo załadowanie niektórych danych testowych. Jest to interesujące, ponieważ możemy go użyć do celów ilustracyjnych, chociaż niestety wstawia bardzo mało rekordów. Okno dialogowe tworzenia bazy danych pyta nas o hasło główne, nazwę nowej bazy danych, język bazy danych i hasło administratora. Mamy również pole wyboru, aby załadować przykładowe dane, więc będziemy je sprawdzać. Hasło główne to admin_passwd natomiast hasło administratora może być wybrane przez nas. Ten proces zajmuje trochę czasu, ponieważ program tworzy i wypełnia nową bazę danych, więc prosimy o cierpliwość i oczekiwanie do końca. Po zainstalowaniu bazy danych możemy wrócić do ekranu głównego i zalogować się nazwą użytkownika: admin i hasłem, które ustawiliśmy dla administratora w oknie dialogowym nowej bazy danych. Po zalogowaniu system wyświetli listę aplikacji dostępnych do zainstalowania. W tym projekcie książki będziemy używać głównie modułu sprzedaży, więc zainstalujemy go i skonfigurujemy, naciskając Zainstaluj w module Zarządzanie sprzedażą na ekranie głównym. Ponownie będziemy musieli trochę poczekać, ponieważ niektóre skrypty muszą utworzyć tabele potrzebne dla modułu i wstępnie wypełnić niektóre z nich danymi testowymi. Czas na przerwę

na kawę! Po chwili instalacja zostanie zakończona i zobaczymy, że domyślnie moduł Sales Management zainstalował również jako warunek wstępny moduł Discuss i Fakturowanie. To idealnie, ponieważ będziemy pracować z fakturami. Teraz nadszedł czas, aby sprawdzić dane testowe i zobaczyć naszych klientów oraz powiązane z nimi faktury. Jeśli teraz odświeżymy główny ekran, zobaczymy, że mamy nowe opcje w górnym menu, w tym Sprzedaż, więc klikamy na to i przechodzimy do Klienci, aby przejrzeć naszą bazę klientów. Jak zobaczysz, mamy teraz kilku klientów dostępnych. Klikając na każdą firmę, pojawia się nowy ekran pokazujący nam kontakty firmy i kilka innych ciekawych opcji, takich jak dane klienta, suma zafakturowana, liczba faktur i tak dalej. Po zainstalowaniu i sprawdzeniu, czy aplikacja działa poprawnie, możemy przystąpić do nauki z modelu danych. Odoo ma złożony model danych, ale na szczęście kod źródłowy jest czysty, dostęp do bazy danych nie jest skomplikowany, a w Internecie jest kilka przydatnych zasobów, które nieco uszczegółwiają model. Dzięki tym fragmentom powinniśmy być w stanie zrozumieć podstawy podstawowych tabel i dwóch modułów, nad którymi będziemy pracować: Zarządzanie sprzedażą i Fakturowanie. W poniższych akapitach szczegółowo omówimy ważne tabele i model, do którego należą tabele; sprawdzimy kolumny i wiersze tych tabel i porównamy to, co jest zapisane w bazie danych, z tym, co widzimy w aplikacji. Jest jeszcze jedna rzecz do zrobienia. Przykładowe dane, które zainstalowaliśmy, są nieco ubogie i nie obejmują wszystkich możliwych scenariuszy. Na przykład nie ma opłaconych faktur: wszystkie mają status oczekujący. Potrzebujemy trochę więcej danych, aby móc wykonać raportowanie, więc utworzymy kilka rekordów i dołączymy je do bieżących danych testowych. W tym celu będziemy musieli utworzyć dane testowe, ale zajmiemy się tym później, gdy zaczniemy pobierać dane z Odoo.

Wybór naszej bazy danych hurtowni danych

Istnieje wiele opcji wyboru systemu RDBMS, który ma stać się naszą bazą danych hurtowni danych. Albo to, albo możemy użyć bazy danych NoSQL. Ale to jeszcze nie jest trend, a większość narzędzi do raportowania, w tym te, których będziemy używać, jest zaprojektowana do pracy z relacyjnymi bazami danych. Chociaż ten paradygmat może ulec zmianie w przyszłości, prawie wszystkie obecnie hurtownie danych oparte są na relacyjnych bazach danych. Istnieje dobra strona internetowa do śledzenia najczęściej używanych baz danych, dzięki czemu możesz sam zobaczyć, jakie trendy są obecnie na rynku i jakie opcje należy rozważyć. Kiedy już jesteśmy świadomi, że potrzebujemy relacyjnej bazy danych, nadszedł właściwy czas na jej wybór. Gdybyśmy mieli przyzwoity budżet i dużo danych do przetworzenia, prawdopodobnie będziemy patrzeć na komercyjną bazę danych. Te wydają się być potężne i pomimo faktu, że istnieją pewne bazy danych typu open source, które również dobrze się skalują, sensowne byłoby trzymanie się wersji prywatnej. Pomyśl szczególnie o wsparciu, aktualizacjach i rozwiązywaniu problemów, a także o liczbie ekspertów dostępnych na rynku, co jest bardzo ważnym czynnikiem, który musisz wziąć pod uwagę, decydując się na użycie jednej technologii lub aplikacji zamiast drugiej. Jeśli zdecydujemy się na małe wdrożenie, sensowne może być użycie PostgreSQL, jak widzieliśmy wcześniej; to ten, który jest używany do przechowywania metadanych Odoo i jest bezpośrednio dołączony do pakietu Bitnami. Jeśli zdecydowaliśmy się na ręczną instalację Odoo, mieliśmy również możliwość dostarczenia bazy danych MySQL dla metadanych. W większości przypadków ta decyzja dotyczy tego, jakie technologie już posiadamy w firmie (lub licencje) i jaka jest główna wiedza specjalistyczna, jaką posiadamy w firmie w zakresie baz danych. Nie ma sensu zakładać hurtowni danych PostgreSQL, jeśli nie mamy w firmie nikogo, kto mógłby się nią zająć, gdy pojawią się problemy (zaufaj mi, zrobią to!). Zalecamy zachowanie prostoty i wdrożenie bazy danych, do której jesteśmy przyzwyczajeni. Jeśli pracujemy w środowisku bazodanowym Oracle to instalacja bazy danych Oracle ma sens, ale czasami jest to niewykonalne. Jako mała firma możemy zainstalować jedną z bezpłatnych (i ograniczonych) wersji komercyjnej bazy danych, którą widzieliśmy w części 3. Oracle XE może być dobrym wyborem, ale należy zachować ostrożność, ponieważ, jak widzieliśmy w części 3, pewne ograniczenia pamięci, procesora i miejsca, więc myślenie o przyszłości zależy od naszego

uzasadnienia biznesowego, aby zdecydować, czy to wystarczy, czy nie. To samo dotyczy Microsoft SQL Server Express. Zakładając, że jesteśmy małą firmą i nie chcemy wydawać pieniędzy na żadną licencję, opcje ograniczają się do jednej z bezpłatnych wersji komercyjnych narzędzi lub open source lub bezpłatnej bazy danych. W naszym przypadku będziemy instalować MySQL/MariaDB dla naszego wdrożenia. Jest kilka dobrych powodów, aby to zrobić:

- * Jest to bezpłatna baza danych (lub open source w przypadku MariaDB).

- * Jest najbardziej rozbudowany i łatwy w administrowaniu i utrzymaniu. Jest wielu ludzi, którzy wiedzą, jak to zrobić, i mnóstwo informacji w sieci.

- * Dobrze się skaluje

Ale jak zawsze wybierz to, co najbardziej Ci odpowiada. W naszym przypadku będzie to MariaDB, ponieważ jest to prawdziwie otwarta baza danych, a najlepsze jest to, że jest kompatybilna z MySQL, więc przy użyciu tych samych narzędzi lub sterowników, które musisz połączyć z MySQL, można również użyć MariaDB. Zrezygnowaliśmy z używania PostgreSQL dołączonego do pakietu Bitnami OdoO jako bazy danych hurtowni danych, ponieważ chcemy, aby źródłowa baza danych ERP była odizolowana od bazy danych hurtowni danych. W bardzo małym wdrożeniu prawdopodobnie nie jest to bardzo ważne, ale myślenie w większym wdrożeniu może spowodować niepowodzenie całego projektu. Dostęp do bazy danych PostgreSQL będzie możliwy kilka razy na sekundę prawdopodobnie przez narzędzie ERP (transakcyjne). Więc istnieje pewna presja na tę bazę danych. Jeśli połączymy również nasze narzędzia BI i procesy ETL z tą samą bazą danych, może to mieć wpływ na narzędzie transakcyjne. Jest to niepożądana konsekwencja korzystania z tej samej bazy danych, ponieważ chcemy zapewnić jak najwyższą dostępność, zwłaszcza dla naszych programów operacyjnych. Ponieważ hurtownie danych są zwykle mniej wrażliwe niż nasze programy operacyjne, zainstalujemy dedykowaną bazę danych, która będzie działać jako hurtownia danych dla naszej platformy. Jest to dość powszechny scenariusz we wdrożeniach BI.

Pozyskiwanie i instalacja MariaDB

Istnieje wiele sposobów instalacji MariaDB. W przypadku systemu Linux kilka dystrybucji ma go już jako domyślną bazę danych w narzędziach menedżera oprogramowania. Inni nadal mają MySQL i będą tęsknić za repozytoriami MariaDB, a inni mogą zmusić cię do ręcznej instalacji pakietów, używając prekompilowanego pakietu lub samodzielnie kompilując źródła. Pomimo tego, co powiedzieliśmy w poprzednich podrozdziałach, kiedy wyjaśniliśmy, że naprawdę powinieneś zainstalować bazę danych hurtowni danych na osobnej maszynie, ponieważ mamy małe środowisko testowe, będziemy instalować ją na tej samej maszynie. Możemy to zrobić, ponieważ nie mamy żadnego innego serwera MySQL ani MariaDB działającego na tej maszynie, ponieważ pakiet Bitnami używa bazy danych PostgreSQL zamiast MySQL/MariaDB i używają innych portów, więc nie musimy nawet martwić się o nich. Dlatego dla uproszczenia tym razem będziemy używać tej samej maszyny, ale pamiętajmy o wcześniejszych rozważaniach.

Instalacja dla Windowsa

Jeśli korzystasz z systemu Windows, możesz bezpośrednio pobrać pliki binarne ze strony internetowej MariaDB i zainstalować je w zwykły sposób, na przykład dla aktualnej wersji w momencie pisania tej książki wyglądało to następująco:

<https://downloads.mariadb.org/mariadb/>

Będziesz musiał pobrać pakiet msi dla systemu Windows (32- i 64-bitowy, niezależnie od tego, jaki masz gust).

Instalacja na Linuksie

Chociaż możesz tam również pobrać pliki źródłowe dla Linuksa, będziemy pobierać już skompilowane pakiety dla naszej dystrybucji. Wyjaśniamy tutaj, jak dodać repozytoria MariaDB do obecnego pakietu menedżera oprogramowania dla kilku dystrybucji, najpopularniejszych, ale poszukaj instrukcji na stronie internetowej MariaDB dla każdego innego rodzaju wdrożenia. W przypadku obu dystrybucji będziemy używać narzędzia do konfiguracji repozytorium dostępnego na stronie internetowej MariaDB i dostępnego pod następującym adresem URL: <https://downloads.mariadb.org/mariadb/repositories>.

Instalacja w Ubuntu

Instalacja w Ubuntu jest prosta, ponieważ będziemy używać menedżera pakietów apt-get. Jednak domyślnie w starszych instalacjach Ubuntu musimy dodać repozytoria MariaDB, ponieważ nie są one domyślnie dostępne. Jeśli używamy aktualnej wersji lub nowszej niż 14.04, możemy pominąć tę część, ponieważ te pliki binarne są już załadowane na liście dystrybucyjnej pakietów. W przypadku wcześniejszych wersji musimy je najpierw dodać. Musimy przejść do narzędzia konfiguracji repozytorium i na liście dystrybucyjnej wybrać Ubuntu, wybrać wersję, którą mamy, wersję MariaDB i lustro, którego będziemy używać. Tak więc pierwszy krok będzie polegał na wykonaniu następujących poleceń w celu dodania repozytorium do naszej pamięci podręcznej apt:

```
sudo apt-get install software-properties-common
sudo apt-key adv --recv-keys --keyserver
hkp://keyserver.ubuntu.com:80 0xF1656F24C74CD1D8
sudo add-apt-repository 'deb
[arch=amd64,i386,ppc64el]
http://tedeco.fi.upm.es/mirror/mariadb/repo/10.1/ubunt
u xenial main'
```

Po zakończeniu możemy przystąpić do aktualizacji naszej pamięci podręcznej apt i zainstalowania programu, tak jak zrobimy to z aktualną wersją dystrybucji:

```
sudo apt update
sudo apt install mariadb-server
```

Gdy wykonamy następujące czynności, pojawi się nowy ekran z prośbą o pobranie nowego oprogramowania, a my potwierdzamy za pomocą „Y”. Po kilku sekundach (lub minutach), w zależności od mocy komputera, wszystko powinno zostać zainstalowane.

Po zakończeniu instalacji nasz serwer MariaDB działa, ale nadal należy wykonać kilka kroków. Jeśli instalacja jest w starej wersji i instalujesz MariaDB 5.5, musisz wykonać dodatkowy krok. Po instalacji musisz uruchomić (ten sam) skrypt, co w MySQL, aby zabezpieczyć instalację. Skrypt nazywa się `mysql_secure_installation` (wywołaj go `sudo mysql_secure_installation`) i powinien być dostępny z twojej ścieżki. Ten skrypt umożliwia zmianę hasła roota i usunięcie niektórych niepotrzebnych użytkowników.

Być może zostałeś wcześniej poproszony o ustawienie hasła; w takim przypadku możesz pominąć pierwszą część skryptu. Pozostałe rzeczy, o które zostaną poproszeni, to:

Change the root password? [Y/n]

Remove anonymous users? [Y/n]

Disallow root login remotely? [Y/n]

Remove test database and access to it? [Y/n]

Reload privilege tables now? [Y/n]

Sugerujemy odpowiedź Y, aby zmienić hasło roota, chyba że ustawiłeś je podczas instalacji, usunąć wszystkich anonimowych użytkowników, wyłączyć zdalne logowanie, JEŚLI instalacja Odoo nie została przeprowadzona na osobnym komputerze, w tym przypadku odpowiedź Nie; w przeciwnym razie będziesz musiał później dostosować uprawnienia dla użytkowników, aby usunąć testowe bazy danych, ponieważ nie są one potrzebne i przeladować tabele uprawnień po operacjach. Następnie nadszedł czas, aby spróbować zalogować się na serwerze i możemy użyć narzędzia poleceń, aby przetestować łączność i konfigurację. Nie jest to konieczne, ale zalecamy ponowne uruchomienie usług. Można to osiągnąć za pomocą następujących poleceń:

```
sudo service mysql stop && sudo service mysql start
```

Następnie możesz sprawdzić stan usługi, wykonując:

```
sudo service mysql status
```

I połącz się z bazą danych z klientem:

```
mysql -u root -p
```

Program poprosi o hasło roota, które właśnie ustawiłeś w poprzednich krokach. Jeśli zobaczysz coś w rodzaju banera MYSQL „Welcome to the MariaDB monitor...”, wszystko jest w porządku i gotowe do stworzenia pierwszej bazy danych!

Uwaga: w nowszych wersjach MariaDB może nie być konieczne uruchamianie skryptu sekurytyzacji po zakończeniu instalacji, ponieważ niektóre z tych zadań, takie jak zmiana hasła roota, są już wbudowane w główny instalator. W takim przypadku po prostu pomiń tę sekcję dokumentu i przejdź od razu do przetestowania łączności.

Instalacja MariaDB w Centos

Omówiliśmy już instalację MariaDB w Windows i Ubuntu, ale dla innych osób korzystających z Centos, Fedory lub Redhat Enterprise, tutaj omawiamy podstawy instalacji. Domyślnie Centos 7 jest dostarczany z instalacją MariaDB 5.5 (taką samą, którą zainstalowaliśmy w Ubuntu), więc proces będzie podobny, ale w tym przypadku przy użyciu menedżera pakietów yum. Czas zacząć!


```

yum install mariadb-server mariadb
=====
Package Architecture Versio
n Repository Size
=====
Installing:
mariadb x86_64 1:5.5.47-
1.el7_2 updates 8.9 M
mariadb-server x86_64 1:5.5.47-
1.el7_2 updates 11 M
Instalando para las dependencias:
perl-Compress-Raw-Bzip2 x86_64 2.061-
3.el7 base 32 k
perl-Compress-Raw-Zlib x86_64 1:2.061-
4.el7 base 57 k
perl-DBD-MySQL x86_64 4.023-
5.el7 base 140 k
perl-DBI x86_64 1.627-
4.el7 base 802 k
perl-IO-Compress noarch 2.061-
2.el7 base 260 k
perl-Net-Daemon noarch 0.48-
5.el7 base 51 k
perl-PlRPC noarch 0.2020-
14.el7 base 36 k

Transaction summary
=====
Install 2 Packages (+7 Dependant packages)

Total size to download: 21 M
Total size installed: 107 M
Is this ok [y/d/N]:y

```

Następnie instalacja zostanie zakończona, a następnie możemy uruchomić serwer za pomocą:

```
systemctl start mariadb
```

A teraz możemy zabezpieczyć naszą instalację w taki sam sposób, jak w przypadku instalacji Ubuntu.

```
mysql_secure_installation
```

Odpowiadamy tak samo jak w wersji Ubuntu i po tym jesteśmy gotowi do uruchomienia klienta i przetestowania połączenia

```
mysql -u root -p
```

I znowu powinniśmy zobaczyć słynne „Witamy w monitorze MariaDB...” Jeśli nie, czytaj dalej, aby rozwiązać problemy.

Uwaga: ponownie zauważ, że jeśli zdecydowaliśmy się zainstalować MariaDB z jednego z oficjalnych repozytoriów, spowoduje to zainstalowanie wersji 10.1 z niewielkimi zmianami w procedurze instalacji

Rozwiązywanie problemów z łącznością

Możliwe, że domyślnie nie możemy połączyć się z komputera zewnętrznego z naszą nową instalacją MySQL. Niektóre pakiety wyłączają to, określając adres sprzężenia zwrotnego jako adres powiązania dla silnika MySQL. W takim przypadku rozwiązanie jest dość proste, musimy edytować plik `/etc/mysql/my.cnf` za pomocą naszego pożądanego edytora (`vi`, `nano`...) i szukać adresu, który określa, którego IP nasłuchuje baza danych, który będzie adresem IP sprzężenia zwrotnego, mniej więcej tak:

```
bind-address: 127.0.0.1
```

I zmień to, wiążąc go ze wszystkimi interfejsami lub z adresem IP interfejsu, który chcesz powiązać (zwykle `eth0`), ale może mieć inną nazwę, stąd plik powinien teraz czytać

```
bind-address: 0.0.0.0
```

Lub

```
bind-address: (wpisz tutaj ip twojej sieci interfejs, z którym chcesz się połączyć)
```

Potem pozostaje nam tylko zrestartować usługę...

```
sudo service mysql restart
```

I możemy przejść, określając adres IP, z którym chcemy się połączyć `mysql -u root -p` (twoje hasło tutaj bez spacji) - `h` (tutaj twoje ip ze spacją)

tj.

```
mysql -u root -pp4ssw0rd -h 192.168.2.10
```

Tworzenie naszej pierwszej bazy danych

Nadszedł czas, aby połączyć się z naszą instalacją MariaDB i zacząć się z nią bawić. Celem jest przekształcenie tej bazy danych w naszą nową hurtownię danych, z której będziemy pobierać dane z wielu źródeł. System transakcyjny, w naszym przypadku nasza nowa aplikacja Odoo, będzie zawierał szczegóły dotyczące klientów, produktów, zamówień, faktur itp., które chcemy przeanalizować. Te wraz z innymi informacjami zewnętrznymi zostaną wyodrębnione i załadowane do naszego

hurtownia danych. W kolejnych rozdziałach zobaczymy, jak wyodrębnić te informacje, i opracujemy kilka procesów, które zajmą się nimi w całej książce, ale najpierw musimy przygotować naszą bazę danych. Na potrzeby naszej realizacji postanowiliśmy stworzyć dwie bazy danych. Nie mamy bazy danych ODS, więc będziemy nazywać naszą inscenizację, nawet jeśli nie ma żadnych/kilka transformacji ze źródłowej bazy danych. W tym celu z dwóch baz danych, które mają zostać utworzone, jedna będzie tak zwaną bazą pomostową, która będzie emulować bazę danych ODS, w której zwykle umieszczane są surowe dane, a druga będzie zawierała bieżącą hurtownię danych. Jest to powszechna konfiguracja, ponieważ zwykle potrzebujemy miejsca, aby najpierw umieścić nasze dane, które będą pochodzić z wielu źródeł, zanim zostaną przekształcone w ostateczny kształt. W niektórych przypadkach możliwe jest napisanie skryptu ETL, który wykona całą tę pracę i może bezpośrednio umieścić dane w hurtowni danych, ale jest to zawsze dobre rozwiązanie. Argumentów przeciw jest wiele:

* Czasami trzeba mieszać dane z wielu źródeł, co utrudnia sprawę, ponieważ to samo narzędzie nie zawsze może mieszać wszystkie pojedyncze elementy danych.

* Wydajność jest zwykle gorsza, jeśli pracujemy z zewnętrznymi źródłami danych, a także dostępność może być zagrożona.

* Łatwiej jest wyleczyć się z błędów. Zwykle dane źródłowe są w formacie nieprzetworzonym i jak widzieliśmy w poprzednich rozdziałach, mogą nie być poprawne. Musimy trochę posprzątać, a wstawianie bezpośrednio do naszych tabel końcowych może naruszyć już sformatowane dane i znacznie utrudnić odzyskiwanie, a nawet spowodować utratę danych.

Mając to na uwadze, możemy przystąpić do tworzenia naszych dwóch baz danych i do tego nie będziemy bardzo oryginalni; nazwijmy je staging i dwh, co oznacza DataWareHouse. Polecenia, aby to osiągnąć, są następujące:

Najpierw musimy ponownie połączyć się z naszą bazą danych na wypadek rozłączenia:

```
mysql -u root -p
```

Utworzymy dwóch użytkowników z uprawnieniami tylko do tych samych baz danych. Każdy użytkownik będzie miał uprawnienia tylko do własnej bazy danych. Później w kolejnych rozdziałach utworzymy użytkowników, którzy będą mieli uprawnienia do interakcji między dwiema bazami danych, ale w tej chwili ich nie potrzebujemy.

Pokaż polecenia baz danych i tabel

Zacznijmy od utworzenia dwóch baz danych. Ponieważ jesteśmy zalogowani jako root, nie będziemy mieli żadnych problemów z utworzeniem dwóch baz danych. Polecenia, które musimy uruchomić, są następujące:

```
CREATE DATABASE IF NOT EXISTS dwh;
```

```
CREATE DATABASE IF NOT EXISTS staging;
```

Te dwa polecenia utworzą nam dwie bazy danych, ale nadal nic więcej. Żaden użytkownik, ponieważ jeszcze ich nie utworzyliśmy, nie będzie miał dostępu (z wyjątkiem naszego użytkownika root), a nawet jeśli użytkownicy już istnieją, nie będzie miał dostępu, ponieważ nie nadaliśmy im uprawnień. Dodajmy dwóch użytkowników, a następnie uprawnienia dla każdego z nich, aby uzyskać dostęp do własnej bazy danych.

Tworzenie użytkowników dla dwóch nowych baz danych

Stworzyliśmy bazy danych, teraz potrzebujemy użytkowników. Tworzenie użytkowników w MariaDB jest dość proste, ale musimy określić, czy użytkownik będzie lokalny (będzie miał dostęp do bazy danych tylko z tego samego serwera), czy będzie to użytkownik zdalny. W zależności od wybranej konfiguracji musimy wykonać jedną z dwóch opcji: Jeśli nasz użytkownik będzie miał dostęp do bazy danych lokalnie, możemy utworzyć użytkownika lokalnie. Oświadczenia są następujące:

```
CREATE USER 'dwh'@'localhost' IDENTIFIED BY
```

```
'p4ssw0rd';
```

```
CREATE USER 'staging'@'localhost' IDENTIFIED BY
```

```
'p4ssw0rd';
```

Jeśli nasi użytkownicy potrzebują dostępu z zewnątrz, co jest prawdopodobne, możemy dodać symbol wieloznaczny (%) i będą mogli łączyć się z dowolnego miejsca. Obaj użytkownicy mogą współistnieć w tej samej bazie danych.

```
CREATE USER 'dwh'@'%' IDENTIFIED BY 'p4ssw0rd';
```

```
CREATE USER 'staging'@'%' IDENTIFIED BY 'p4ssw0rd';
```

Dzięki temu nasi użytkownicy są gotowi do dostępu. Ale w tym momencie nie będą mogli operować na bazach danych. Przydzielmy każdemu wymagane uprawnienia do ich baz danych.

Uwaga

Później, kiedy przejdziemy do części ETL i Reporting, ci użytkownicy będą wykorzystywani, więc jest bardzo prawdopodobne, że użytkownicy będą potrzebować dostępu z zewnątrz, więc weź to pod uwagę. W każdym razie można to zmienić w razie potrzeby, więc nie martw się o to teraz, ale po prostu miej to pod ręką na wypadek problemów z połączeniem.

Udział uprawnień do baz danych

Stworzyliśmy bazy danych i użytkowników, więc teraz czas na połączenie tych dwóch pojęć. W tym celu musimy zezwolić użytkownikom na manipulowanie bazami danych. Struktura uprawnień w MariaDB/MySQL jest dość łatwa do zrozumienia. Ma dobrze zdefiniowany format:

```
GRANT [type of permission] ON [database name].
```

```
[table name] TO '[username]'@'%';
```

Aby cofnąć uprawnienia, składnia jest bardzo podobna, zmieniając tylko słowa kluczowe:

```
REVOKE [type of permission] ON [database name].
```

```
[table name] FROM '[username]'@'%';
```

Zwróć uwagę na symbol wieloznaczny % na końcu, który musi zostać zastąpiony przez localhost, jeśli pracujemy z użytkownikami o dostępie lokalnym. W naszym przypadku przyznamy uprawnienia naszym lokalnym i zewnętrznym użytkownikom:

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON dwh.* TO
```

```
'dwh'@'localhost';
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON dwh.* TO
```

```
'dwh'@'%';
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON staging.*
```

```
TO 'staging'@'localhost';
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON staging.*
```

```
TO 'staging'@'%';
```

Query OK, 0 rows affected (0.00 sec)

Gdy to zostanie osiągnięte, możemy przystąpić do testowania naszych użytkowników, w tym celu opuszczamy naszą sesję root i testujemy obu użytkowników. Teraz zamiast tego należy nawiązać połączenie z nowym użytkownikiem:

```
mysql -u dwh -pp4ssw0rd
```

Wszystko idzie dobrze i powinniśmy zobaczyć naszego klienta zalogowanego:

Welcome to the MariaDB monitor. Commands end with

; or \g.

Your MariaDB connection id is 14

Server version: 5.5.47-MariaDB MariaDB Server

Copyright (c) 2000, 2015, Oracle, MariaDB

Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear

the current input statement.

Następnie możemy sprawdzić, które bazy danych użytkownik widzi

```
MariaDB [(none)]> show databases;
+-----+
| Database          |
+-----+
| information_schema|
| dwh                |
+-----+
2 rows in set (0.00 sec)
```

And then change to our database:

```
MariaDB [(none)]> use dwh;
Database changed
```

And then check that no tables are present yet:

```
MariaDB [dwh]> show tables;
Empty set (0.00 sec)
```

A następnie przejdź do naszej bazy danych:

```
MariaDB [(none)]> use dwh;
```

Database changed

A następnie sprawdź, czy nie ma jeszcze żadnych tabel:

```
MariaDB [dwh]> show tables;
```

Empty set (0.00 sec)

Następnie możemy wyjść i przetestować drugiego użytkownika, wykonując dokładnie tę samą procedurę:

```
MariaDB [dwh]> exit;
```

Bye

Połączymy się teraz jako użytkownik pomostowy i sprawdzimy to dopiero po zobaczeniu bazy danych pomostowych:


```
[root@localhost anogues]# mysql -u staging -
pp4ssw0rd
Welcome to the MariaDB monitor.  Commands end with
; or \g.
Your MariaDB connection id is 15
Server version: 5.5.47-MariaDB MariaDB Server

Copyright (c) 2000, 2015, Oracle, MariaDB
Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear
the current input statement.

MariaDB [(none)]> show databases;
+-----+
| Database          |
+-----+
| information_schema |
| staging            |
+-----+
2 rows in set (0.00 sec)

MariaDB [(none)]> use staging;
Database changed
MariaDB [staging]> show tables;
Empty set (0.00 sec)

MariaDB [staging]> exit;
Bye
```

Uwaga

Przedstawiliśmy wszystkie polecenia przy użyciu interakcji klienta z wiersza poleceń. Niedoświadczonym użytkownikom może się opłacać wypróbowanie narzędzia GUI do interakcji z bazami danych MySQL/MariaDB. Jest ich wiele za darmo, będąc oficjalnym środowiskiem roboczym MySQL, prawdopodobnie najbardziej znanym: <https://www.mysql.com/products/workbench/>, który działa w kilka systemów operacyjnych, ale istnieją inne dobre programy, takie jak HeidiSQL i dbForge dla systemów Windows, które są bezpłatne lub przynajmniej mają darmową edycję.

Analiza źródła danych

Nasza hurtownia danych jest już gotowa do działania. Nadszedł więc czas, aby zacząć dokładniej przyglądać się oprogramowaniu Odoo i jego strukturze bazy danych, ponieważ byłoby to źródłem

większości naszych danych w części ETL. Jeśli pamiętasz z początku tej części, zainstalowaliśmy oprogramowanie Odoo, używając PostgreSQL jako bazy danych do przechowywania metadanych aplikacji. Tutaj zobaczymy mały przegląd klienta PostgreSQL, dzięki czemu możemy sprawdzić źródłową bazę danych i tabelę Odoo. Tutaj przedstawimy tylko przegląd, które tabelę rozważymy później w naszym systemie, i wyjaśnimy podstawową relację między nimi a modułem, do którego należą.

Sprawdzanie naszego modelu

Możemy zacząć od bezpośredniego połączenia z bazą danych i rozpoczęcia przeglądania tabel. Jednak w zależności od aplikacji, które zainstalowaliśmy, znajdziemy wiele i wiele tabel. Na szczęście istnieje bardzo dobry internetowy zasób modułów Odoo na następującej stronie:

<http://useopenerp.com/v8>

Póki co dostępna dokumentacja dotyczy wersji 8.0, my będziemy używać wersji 9.0, ponieważ w naszym przypadku nie jest to bardzo ważne, ponieważ różnice są minimalne. Jak widać w sieci, aplikacja Odoo jest podzielona na zestaw kategorii, czyli modułów, które są podobne do aplikacji w Odoo. Chociaż nie tłumaczą się bezpośrednio, ponieważ istnieją pewne wspólne kategorie, które są używane przez wiele aplikacji i są uważane za rdzeń Odoo, niektóre z nich mają podobieństwa. Na przykład w Odoo możemy znaleźć kategorię Zarządzanie sprzedażą oraz aplikację Zarządzanie sprzedażą, ale na przykład nie możemy znaleźć kategorii Fakturowanie, podczas gdy w Odoo mamy aplikację do fakturowania. Wynika to z faktu, że Fakturowanie należy do kategorii Zarządzanie sprzedażą. Jeśli klikniemy dalej w link Zarządzanie sprzedażą, możemy zaobserwować następujący adres URL modułu:

<http://useopenerp.com/v8/module/sales-management>

Możemy również zobaczyć szczegóły dotyczące modułu Zarządzanie sprzedażą w Odoo. Możemy zobaczyć krótki opis aplikacji i przepływów obsługiwanych przez ten moduł wraz z kilkoma zrzutami ekranowymi menu i opcji objętych tym modułem, wszystkimi klasami Pythona zaangażowanymi w diagram klas UML, na wypadek, gdybyśmy chcieli dostosować ten moduł przez modyfikowanie kodu bezpośrednio w aplikacji i jeszcze kilka rzeczy, szczególnie interesujących dla naszego projektu, sekcja Modele. Model to jedna funkcjonalność zaimplementowana w Odoo. Mamy na przykład model sale.order, który zawiera wszystkie informacje dotyczące funkcjonalności zamówień aplikacji. Na przykład dla tego modułu możemy zobaczyć wszystkie kolumny, które są odwzorowane na różne tabelę bazy danych i będą zawierać wymaganą logikę do zaimplementowania tej funkcjonalności. Należy również zauważyć, że model może być używany przez wiele aplikacji. Diagram Modelu jest również ważny, ponieważ dostarcza nam informacji o relacjach między innymi modelami i klasami, ale w tym momencie bardziej interesują nas kolumny i typy tych kolumn; więc patrząc na tę sekcję, możemy zacząć opracowywać listę pól, które będziemy mogli wyodrębnić i skopiować do naszego obszaru przejściowego. Na przykład, w oparciu o potrzeby biznesowe, jesteśmy zainteresowani wydobywaniem informacji o naszych zamówieniach. Aby wesprzeć naszą analizę, nie jesteśmy zainteresowani wyodrębnieniem całego modelu zamówień, ponieważ prawdopodobnie znajdują się tam informacje, których nie potrzebujemy. Pierwszym krokiem będzie wyodrębnienie informacji zbiorczych dla tabel modeli Odoo do naszego obszaru przejściowego. Na przykład możemy rozważyć skopiowanie następujących pól z modelu zamówienia:

Nazwa kolumny: Opis

nazwa : Nazwa zamówienia

stan L Status zamówienia: wersja robocza, wysłane, anulowane, w trakcie realizacji, wykonane...

Date_order : Data realizacji zamówienia

User_id : Klucz do Sprzedawcy, który przyjął zamówienie w modelu res.users

Partner_id L Klucz do identyfikacji naszego klienta w modelu res.partner

Order_line : jeden lub więcej wierszy zamówień produktów, które odwołują się do modelu sale.order.line

Otrzymując te pola będziemy już mogli obliczyć w naszej hurtowni danych np. sumę zamówień klienta, sumę zamówień generowanych przez handlowca, liczbę sztuk na zamówienie jako średnią, sumę zamówień miesięcznie oraz na przykład procent zamówień, które faktycznie kończą się zaakceptowaną fakturą, żeby wymienić tylko kilka możliwych analiz. Prawdopodobnie, aby uzyskać bardziej znaczące informacje, musimy wyodrębnić więcej informacji z innych modeli. Na przykład prawdopodobne jest, że zamiast ich wewnętrznych identyfikatorów będziemy potrzebować nazwisk naszych pracowników lub klientów, prawda? Będzie to wymagało od nas nieco więcej ćwiczeń i wyszukania potrzebnych kolumn w res.users (dla naszych pracowników) i res.partners (dla naszych klientów), aby wyodrębnić wszystkie ich nazwiska, adresy i wszelkie istotne informacje potrzebujemy dla nich.

Uwaga

Próba zrozumienia modelu danych, zwłaszcza tak dużego jak ten, bez posiadania zbyt dużej wiedzy na temat UML, relacji między jednostkami i baz danych, może początkowo być nieco onieśmialająca. Ale przykłady, które zrobimy, są łatwe do zrozumienia i wszystkie będą prowadzone, więc nie denerwuj się, jeśli jesteś teraz trochę przytłoczony.

Spoglądając na res_users szybko zauważymy, że nie ma tam nazwisk pracowników. Stanowi to problem dla naszej próbki, ponieważ ich potrzebujemy. Musimy więc szukać gdzie indziej. Na szczęście znowu model przychodzi nam z pomocą i udaje nam się ich zlokalizować. Podążając za informacjami ze strony modelu widzimy następującą sekcję, po definicji kolumny wszystkich dostępnych kolumn dla użytkowników res_users: „Kolumny odziedziczone z bazy dodatków > res > res_partner.py > class res_partner”.

W tej chwili musimy skupić się tylko na ostatnim kroku zdania. Daje nam to wskazówkę, gdzie się udać i znaleźć potrzebne dane. W tym przypadku system mówi nam dokładnie, że musimy wyszukać dane w tabeli res_partner. A patrząc na model widzimy, że tabela res_partner zawiera kolumnę name, która będzie identyfikować naszego pracownika. Ale jednocześnie ta tabela jest również używana dla klientów, których możesz się zastanawiać, jak wyjaśniliśmy wcześniej. Masz rację. Dla Odoo wszyscy użytkownicy, którzy tworzą część naszego systemu, a nawet organizacje, są uważani za członków modelu res_partner. Oznacza to, że wszystkie firmy, w tym nasza własna, wszyscy nasi pracownicy, wszyscy nasi klienci, bez względu na to, czy są to firmy, czy osoby fizyczne, będą miały co najmniej jeden wpis w tabeli res_partner. Fakt ten oznacza, że kiedy będziemy pracować nad ekstrakcją, będziemy musieli rozważyć wykonanie wielu połączeń do tej tabeli, oczywiście przy użyciu różnych kluczy i kolumn, w zależności od tego, jakie informacje chcemy wyodrębnić. Ale tym zajmiemy się dopiero później.

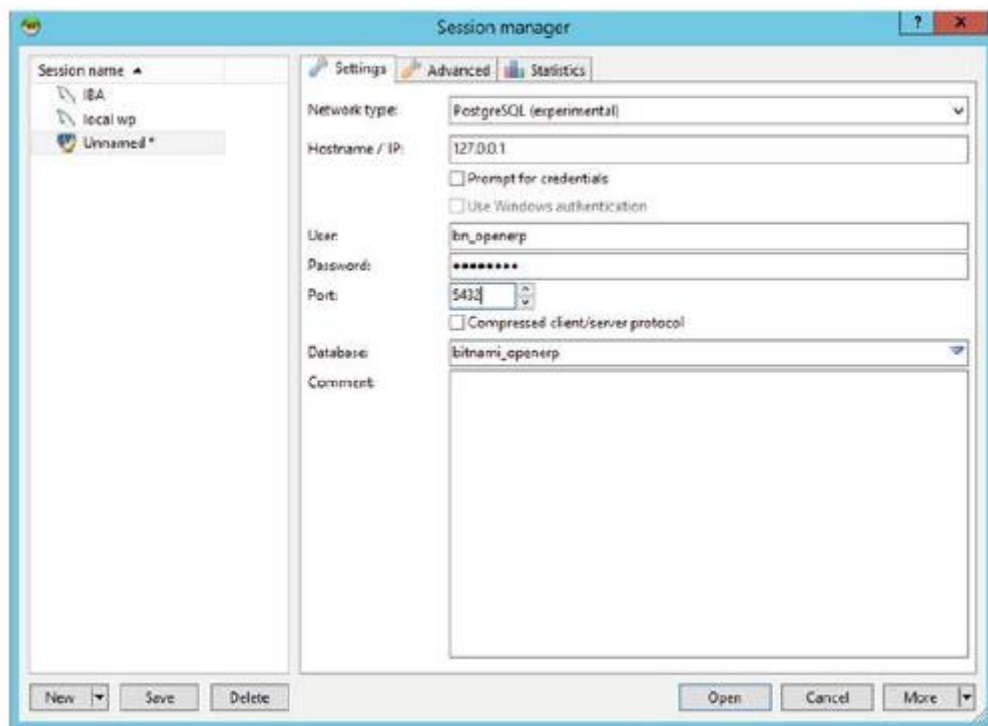
Konfigurowanie połączenia PostgreSQL z serwerem

Jeśli pamiętasz z pierwszej części tego rozdziału, kiedy instalowaliśmy pakiet Bitnami, zdecydowaliśmy się na użycie dołączonej bazy danych PostgreSQL. Oznacza to, że metadane wszystkich aplikacji i modułów Odoo będą przechowywane w bazie danych PostgreSQL. To dobre ćwiczenie, ponieważ w

dalszych rozdziałach, kiedy będziemy pracować nad częścią ETL, zobaczymy, jak połączyć się z różnymi bazami danych, ponieważ nasza hurtownia danych będzie w MariaDB. Jednak w tym rozdziale bardziej interesuje nas możliwość łączenia się z metadanymi aplikacji i przeglądania naszej bazy danych. Zaczynamy. Istnieje wiele zastosowań do tego celu iw różnych smakach. Możemy zdecydować się na ponowne użycie wiersza poleceń lub skorzystać z aplikacji GUI. Chociaż linia poleceń jest czasami przydatna, jeśli zasadniczo interesuje nas przeglądanie relacji i zawartości danych, dostrojenie klienta konsolowego do ładnego wyświetlania jest bardzo trudne. Z tego powodu tym razem użyjemy narzędzia GUI do sprawdzenia naszych metadanych Odoo. Istnieje wiele narzędzi GUI, które są bezpłatne i będą pasować do rachunku. W rzeczywistości PostgreSQL jest dostarczany z pakietem. My jednak wolimy używać HeidiSQL. Dobrą rzeczą jest to, że chociaż HeidiSQL jest przeznaczony dla baz danych MySQL i MariaDB, obsługuje również bazy danych PostgreSQL. I jest open source, więc można go używać za darmo!

Konfigurowanie połączenia PostgreSQL z serwerem

Jeśli pamiętasz z pierwszej części tego rozdziału, kiedy instalowaliśmy pakiet Bitnami, zdecydowaliśmy się na użycie dołączonej bazy danych PostgreSQL. Oznacza to, że metadane wszystkich aplikacji i modułów Odoo będą przechowywane w bazie danych PostgreSQL. To dobre ćwiczenie, ponieważ w dalszych rozdziałach, kiedy będziemy pracować nad częścią ETL, zobaczymy, jak połączyć się z różnymi bazami danych, ponieważ nasza hurtownia danych będzie w MariaDB. Jednak w tym rozdziale bardziej interesuje nas możliwość łączenia się z metadanymi aplikacji i przeglądania naszej bazy danych. Zaczynamy. Istnieje wiele zastosowań do tego celu iw różnych smakach. Możemy zdecydować się na ponowne użycie wiersza poleceń lub skorzystać z aplikacji GUI. Chociaż linia poleceń jest czasami przydatna, jeśli zasadniczo interesuje nas przeglądanie relacji i zawartości danych, dostrojenie klienta konsolowego do ładnego wyświetlania jest bardzo trudne. Z tego powodu tym razem użyjemy narzędzia GUI do sprawdzenia naszych metadanych Odoo. Istnieje wiele narzędzi GUI, które są bezpłatne i będą pasować do rachunku. W rzeczywistości PostgreSQL jest dostarczany z pakietem. My jednak wolimy używać HeidiSQL. Dobrą rzeczą jest to, że chociaż HeidiSQL jest przeznaczony dla baz danych MySQL i MariaDB, obsługuje również bazy danych PostgreSQL. I jest open source, więc można go używać za darmo! Aby zainstalować HeidiSQL, możemy kilka razy kliknąć Dalej. Jeśli chcesz zmienić ścieżkę lub inną konfigurację, możesz to zrobić. Jeśli podczas instalacji postępowałeś zgodnie z domyślnymi krokami, prawdopodobnie parametry połączenia będą takie same, jak te pokazane na rysunku



To jest ekran powitalny, który pojawia się, gdy po raz pierwszy otwierasz HeidiSQL i jest to miejsce do zdefiniowania wszelkich nowych połączeń, które chcesz utworzyć. Nasze połączenie byłoby z silnikiem PostgreSQL, więc upewnij się, że jest to ten wybrany w rozwijanym polu Typ sieci.

Po połączeniu zobaczymy kilka baz danych, z których większość jest wewnętrzna w bazie danych, ale wyróżnia się jedna o nazwie publiczna. Musimy kliknąć ten, a pojawi się lista tabel. Jednym z nich jest res_partner. Jeśli klikniesz go dwukrotnie, wejdziemy w tryb tabeli. W tym momencie nie chcemy edytować tabeli, więc nie zapisuj przypadkowo wprowadzonych zmian (jeśli w ogóle!). W tym widoku jesteśmy w stanie zobaczyć kolumny, relacje z innymi tabelami, a nawet przeglądać dane, a wszystko to za pomocą kilku kliknięć myszką. Karta danych będzie później bardzo przydatna, aby sprawdzić, czy dane, które wyodrębniamy, pasują do danych w systemie źródłowym. Lista kolumn daje nam wskazówki dotyczące kolumn, które możemy potrzebować wyeksportować, a żółte i zielone klucze przed numerami kolumn pokazują, czy kolumna jest kluczem podstawowym (żółty), czy kluczem obcym (zielony) odwołującym się do kolumny w innej tabeli. Informacje te uzupełniają informacje, które możemy znaleźć na stronie modelu i pomogą nam przez cały czas dokładnie wiedzieć, jakie informacje potrzebujemy pobrać i skąd je uzyskać.

Wniosek

Zobaczyliśmy, jak zainstalować system operacyjny, w naszym przypadku pakiet Odoo. Widzimy również, jak wykonać podstawową konfigurację aplikacji. Następnie zainstalowaliśmy bazę danych MariaDB do przechowywania naszego magazynu danych i utworzyliśmy dwie wymagane bazy danych: bazę danych ODS, którą nazwaliśmy staging; oraz baza danych hurtowni danych. Po instalacji naszej hurtowni danych przyjrzelśmy się strukturze Odoo, gdzie przechowuje informacje i jak uzyskać do nich dostęp z narzędzia GUI. Jeśli wykonaliśmy wszystkie kroki opisane w tym rozdziale, jesteśmy gotowi, aby przejść do bardziej zaawansowanych koncepcji i wreszcie zacząć bawić się naszymi danymi i je analizować. Z tego powodu ważne jest, aby wykonać wszystkie kroki opisane tutaj, ponieważ kolejne części zakładają, że instalacje i konfiguracje omówione w tej części zostały już zaimplementowane.

