

Depuis 2006, Oracle propose une version gratuite de son produit phare. Il est donc possible de librement télécharger une version du SGBD Oracle appelé **Oracle Express Edition** (*Oracle XE*). Il s'agit en fait d'une version limitée de *Oracle 10g*. Limitée, certes, mais très bien pour découvrir cet outil ou pour travailler à la maison.

Dans cet article, nous allons voir comment **installer Oracle XE en environnement Linux/Debian**. Par ailleurs nous découvrirons aussi deux clients fournis avec le serveur, à savoir **Oracle Application Express** et **SQL*Plus**.

Sommaire

- 1 Installer le serveur
 - 1.1 Première tentative: pas assez de swap (!)
 - 1.2 Seconde tentative: libaio manquante
 - 1.3 Configuration
- 2 Créer un utilisateur
- 3 Créer une table
 - 3.1 Avec Oracle Application Express (interface graphique)
 - 3.2 Avec Oracle Application Express (Commande SQL)

Installer le serveur

Pour commencer, nous allons procéder à l'installation d'**Oracle Express Edition**. Ce produit est disponible en différentes versions: Linux (RPM/Deb) ou Windows, avec ou sans prise charge internationale. Pour ma part, comme je vais installer Oracle sur mon portable sous Debian, j'ai choisi pour cet article la version ".deb" Linux Universal: `oracle-xe-universal_10.2.0.1-1.0_i386.deb`. A strictement parler, si vous n'utilisez que des langues latines, le support *universal* n'est pas indispensable. Néanmoins, c'est un peu dommage de s'en priver. D'autant que la tendance actuelle est à la généralisation de l'UTF-8. (*utf8 everywhere*).



Note:

Au moment de télécharger *Oracle XE*, vous devrez vous inscrire sur le site d'Oracle si ce n'est pas déjà fait. L'inscription est gratuite – et n'engage à rien. C'est juste long et ennuyeux...

Première tentative: pas assez de swap (!?)

Comme vous l'avez deviné, la version *.deb* est un paquet Debian. Donc à installer avec `dpkg`:

```
sh# dpkg -i /home/sylvain/Desktop/oracle-xe-universal_10.2.0.1-1.0_i386.deb
Selecting previously deselected package oracle-xe-universal.
(Reading database ... 103587 files and directories currently installed.)
Unpacking oracle-xe-universal (from .../oracle-xe-universal_10.2.0.1-1.0_i386.deb) ...
This system does not meet the minimum requirements for swap space. Based on
the amount of physical memory available on the system, Oracle Database 10g
```

```
Express Edition requires 1024 MB of swap space. This system has 919 MB
of swap space.  Configure more swap space on the system and retry the installation.
dpkg: error processing /home/sylvain/Desktop/oracle-xe-universal_10.2.0.1-1.0_i386.deb (--install):
 subprocess pre-installation script returned error exit status 1
Errors were encountered while processing:
 /home/sylvain/Desktop/oracle-xe-universal_10.2.0.1-1.0_i386.deb
```

Bon, première tentative infructueuse. Pas assez de swap. Heu, sauf que j'ai 2GiB (pour 512MiB de RAM). Pourquoi le script d'installation ne rapporte que 919MB? L'explication semble être que ce script vérifie la quantité de swap *restant disponible*. En quittant quelques applications j'ai finalement réussi à satisfaire aux conditions minimum requises sans avoir à étendre ma partition de swap.

Seconde tentative: libaio manquante

Une fois ces soucis de swap réglés, nous pouvons retenter l'installation:

```
sh# dpkg -i /home/sylvain/Desktop/oracle-xe-universal_10.2.0.1-1.0_i386.deb
(Reading database ... 106181 files and directories currently installed.)
Preparing to replace oracle-xe-universal 10.2.0.1-1.0 (using .../oracle-xe-universal_10.2.0.1-1.0_i386.deb) ...
Unpacking replacement oracle-xe-universal ...
dpkg: dependency problems prevent configuration of oracle-xe-universal:
 oracle-xe-universal depends on libaio (>= 0.3.96); however:
  Package libaio is not installed.
  Package libaio1 is not installed.
dpkg: error processing oracle-xe-universal (--install):
 dependency problems - leaving unconfigured
Processing triggers for man-db ...
Errors were encountered while processing:
 oracle-xe-universal
```

Toujours pas bon! Maintenant, il manque *libaio*. Ou *libaio1* – qui justement est disponible dans les paquets Debian:

```
sh# aptget install libaio1
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
 libaio1
0 upgraded, 1 newly installed, 0 to remove and 10 not upgraded.
1 not fully installed or removed.
```

```
Need to get 7316B of archives.  
After this operation, 73.7kB of additional disk space will be used.  
Get:1 http://10.129.36.102 lenny/main libaiol 0.3.107-3 [7316B]  
Fetched 7316B in 1s (4968B/s)  
Selecting previously deselected package libaiol.  
(Reading database ... 106181 files and directories currently installed.)  
Unpacking libaiol (from .../libaiol_0.3.107-3_i386.deb) ...  
Setting up libaiol (0.3.107-3) ...  
Setting up oracle-xe-universal (10.2.0.1-1.0) ...  
update-rc.d: warning: /etc/init.d/oracle-xe missing LSB information  
update-rc.d: see <http://wiki.debian.org/LSBInitScripts>  
Executing Post-install steps...  
You must run '/etc/init.d/oracle-xe configure' as the root user to configure the database.
```

Notez au passage que `dpkg` est *gentil*, puisqu'il a repris l'installation interrompue d'Oracle une fois libaiol installée.

Configuration

Bref, une fois Oracle XE installé, reste l'étape de la configuration:

```
sh# /etc/init.d/oracle-xe configure  
Oracle Database 10g Express Edition Configuration  
-----  
This will configure on-boot properties of Oracle Database 10g Express  
Edition. The following questions will determine whether the database should  
be starting upon system boot, the ports it will use, and the passwords that  
will be used for database accounts. Press <Enter> to accept the defaults.  
Ctrl-C will abort.  
  
Specify the HTTP port that will be used for Oracle Application Express [8080]:8181  
  
Specify a port that will be used for the database listener [1521]:  
  
Specify a password to be used for database accounts. Note that the same  
password will be used for SYS and SYSTEM. Oracle recommends the use of  
different passwords for each database account. This can be done after  
initial configuration:  
Confirm the password:  
  
Do you want Oracle Database 10g Express Edition to be started on boot (y/n) [y]:  
  
Starting Oracle Net Listener...Done  
Configuring Database...Done  
Starting Oracle Database 10g Express Edition Instance...Done
```

Installation Completed Successfully.
To access the Database Home Page go to "<http://127.0.0.1:8181/apex>"



Port 8181?

J'ai déjà sur ma machine un service sur le port 8080 (Tomcat). C'est pourquoi j'ai choisi – un peu arbitrairement – le port 8181. Vous retrouverez ce numéro à plusieurs reprises dans les exemples qui vont suivre. Bien sûr si vous configurez *Oracle Application Express* pour utiliser un autre port (ou le port par défaut 8080), c'est *votre* port qu'il faudra utiliser. Pas 8181!

Enfin, vous pouvez tout de suite vous connecter à la console pour voir si tout est vraiment OK:

```
sh$ firefox http://127.0.0.1:8181
```

Et là surprise: point de console, mais la licence d'utilisation du produit. A ben oui: j'ai oublié le nom de l'application (apex) dans l'URL...

```
sh$ firefox http://127.0.0.1:8181/apex
```

Créer un utilisateur

Lors de l'installation d'Oracle deux utilisateurs ont été créés: SYS et SYSTEM. En deux mots, SYS est un utilisateur qui a accès aux tables liées au fonctionnement de base d'Oracle. SYSTEM, lui, à plutôt accès aux tables liées aux tâches d'administration.

Il ne faut jamais utiliser les schémas associés à SYS ou

ORACLE Database Express Edition

Database Login

Enter your database username and password.

SYSTEM pour héberger les tables et les données de votre application. Celles-ci devront au contraire être créées dans le schéma d'un utilisateur *ordinaire*.

Par contre, comme la création d'un nouvel utilisateur fait partie des tâches administratives dévolues à *SYSTEM*, il faut d'abord se connecter sous cette identité pour créer un nouveau compte.

Username

Password

Login

[Click here to learn how to get started](#)

SYSTEM est chargé des tâches administratives comme la création de nouveaux utilisateurs.

ORACLE® Database Express Edition

User: SYSTEM

Home



Une fois connecté sous l'identité SYSTEM, choisissez **Administration > Manage Database Users > Create** pour créer un nouveau compte utilisateur.

Create Database User

Cancel

Create

* Username

SYLVAIN

* Password

●●●●●●●●

* Confirm Password

●●●●●●●●

Expire Password

☐

Account Status

Unlocked

Default Tablespace:

USERS

Temporary Tablespace:

TEMP

User Privileges

Roles:

☒ CONNECT

☒ RESOURCE

☐ DBA

Direct Grant System Privileges:

☐ CREATE DATABASE LINK

☐ CREATE MATERIALIZED VIEW

☐ CREATE PROCEDURE

☐ CREATE PUBLIC SYNONYM

☐ CREATE ROLE

☒ CREATE SEQUENCE

☐ CREATE SYNONYM

☒ CREATE TABLE

☒ CREATE TRIGGER

☒ CREATE TYPE

☒ CREATE VIEW

Check All

Uncheck All

Le formulaire de **création d'un utilisateur** permet de définir le nom de celui-ci, son mot de passe et les permissions qui lui sont associées. Pour la suite de cet article, vous aurez au moins besoin de donner les permissions *créer une table*, *créer une séquence* et *créer un trigger*.

Créer une table

Oracle est livré avec une base de démonstration appelée HR pour Ressources Humaines (*Human Resources*).

Vous saurez, j'en suis certain, *jouer* avec cette base par vous-même. Mais si vous êtes comme moi, vous avez sans doute plutôt hâte de créer vos propres tables. C'est ce que nous allons faire dans la suite de cet article.

Pour cette prise en main d'Oracle, nous allons rester extrêmement modestes. Et nous allons nous contenter de créer une base contenant trois tables pour gérer les compétences professionnelles des employés d'une entreprise:

1. *COMPETENCES*: qui regroupera l'ensemble des compétences reconnues;
2. *EMPLOYES* qui contiendra la liste des employés;
3. et enfin la table d'association *EMPLOYES_COMPETENCES* qui permettra de savoir quel(s) employé(s) possède(nt) quelle(s) compétence(s).

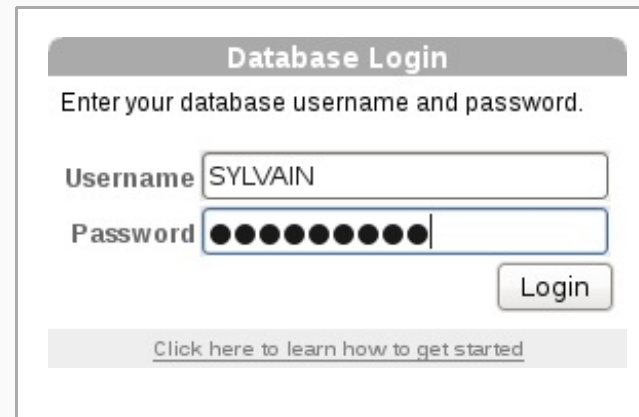
Afin d'avoir un panorama des possibilités offertes par *Oracle XE*, nous utiliserons trois différentes techniques pour créer ces tables:

- Tout d'abord nous utiliserons l'interface graphique conviviale d'*Oracle Application Express*;
- ensuite nous utiliserons toujours *Oracle Application Express*, mais cette fois en fournissant nous même les requêtes SQL;
- et enfin, nous utiliserons le client en mode texte *SQL*Plus* pour saisir directement les requêtes via un terminal.

Avec Oracle Application Express (interface graphique)

Pour cette première table nous allons utiliser l'**interface graphique d'Oracle Application Express**.

A partir de maintenant, nous allons travailler sous l'identité d'un utilisateur *ordinaire*. Si vous êtes encore connecté sous l'identité *SYSTEM*, déconnectez-vous (*logout*) et reconnectez-vous sous l'identité de l'utilisateur créé à l'étape précédente.

The image shows a web-based login form for a database. At the top, there is a header bar with the text "Database Login". Below this, a prompt says "Enter your database username and password." There are two input fields: "Username" with the text "SYLVAIN" entered, and "Password" with a series of black dots for masking. To the right of the password field is a "Login" button. At the bottom of the form, there is a link that says "Click here to learn how to get started".

Database Login

Enter your database username and password.

Username SYLVAIN

Password ●●●●●●●●

Login

[Click here to learn how to get started](#)

User: SYLVAIN

Home



Administration



Object Browser

Create
Browse



Utilities



Application Builder

Create

Table

View

Index

Sequence

Type

Package

Procedure

Function

Trigger

Database Link

Materialized View

Synonym

Pour créer une nouvelle table avec *Oracle Application Express*, choisissez **Object Browser > Create > Table**.

Create Table

Cancel

Next >

* Table Name

COMPETENCES

☐ Preserve Case

Column Name	Type	Precision	Scale	Not Null	Move
id	NUMBER			<input checked="" type="checkbox"/>	▼ ▲
description	VARCHAR2		40	<input checked="" type="checkbox"/>	▼ ▲
	- Select Datatype -				▼ ▲
	- Select Datatype -				▼ ▲
	- Select Datatype -				▼ ▲
	- Select Datatype -				▼ ▲
	- Select Datatype -				▼ ▲
	- Select Datatype -				▼ ▲

Add Column

La première étape va consister à renseigner les champs de la table *COMPETENCES*. Ici, ma table ne comporte que deux colonnes: *ID* l'identifiant unique (qui sera défini dans quelques instants comme la clé primaire) et *DESCRIPTION* ... la description.

Primary Key Cancel < Previous Next >

Table name: **COMPETENCES**

Primary Key: ☐ No Primary Key
☒ Populated from a new sequence
☐ Populated from an existing sequence
☐ Not populated

* Primary Key Constraint Name

* Sequence Name

* Primary Key

La seconde étape nous permet de définir la **clé primaire**. Ici, il s'agit de la colonne *ID*. Remarquez aussi que j'indique que le contenu de cette colonne sera peuplé automatiquement à partir d'une nouvelle **séquence**.

Foreign Keys

Cancel< PreviousNext >

Foreign Key	Columns	Referenced Table	Referenced Columns	Action
-------------	---------	------------------	--------------------	--------

Add Foreign Key

Add

* NameCOMPETENCES_fk

☒ Disallow Delete

☐ Cascade Delete

☐ Set Null on Delete

Select Key Column(s)

IDDESCRIPTION

* Key Column(s)

* References Table

Constraints

Cancel< PreviousFinish

Constraint Name	Type	Column(s)/Check
COMPETENCES_UK1	Unique	DESCRIPTION

Add Constraint

Add

☐ Check ☒ Unique

Select Key Column(s)

IDDESCRIPTION

* Key Column(s)

* Name

COMPETENCES_uk2

A cette étape, vous pouvez définir d'autres éventuelles **contraintes** sur votre table. Ici, j'ai ajouté une contrainte d'unicité sur la colonne *DESCRIPTION*. Attention: une contrainte n'est réellement ajoutée qu'après avoir cliqué sur le bouton *Add*!

Create Table

CancelCreate

 Please confirm your request.

Schema: **SYLVAIN**

Table name: **COMPETENCES**

✓ SQL

```
CREATE table "COMPETENCES" (  
  "ID"          NUMBER NOT NULL,  
  "DESCRIPTION" VARCHAR2(40) NOT NULL,  
  constraint "COMPETENCES_PK" primary key ("ID")  
)  
/  
  
CREATE sequence "COMPETENCES_SEQ"  
/  
  
CREATE trigger "BI_COMPETENCES"  
  before insert on "COMPETENCES"  
  for each row  
begin  
  select "COMPETENCES_SEQ".nextval into :NEW.ID from dual;  
end;  
/  
  
alter table "COMPETENCES" add  
constraint COMPETENCES_UK1  
unique ("DESCRIPTION")
```

La dernière étape est la **confirmation**. Vous pouvez ici vérifier le code SQL qui sera exécuté. Et si tout vous semble correct, il ne vous reste alors qu'à cliquer sur le bouton *create*.

COMPETENCES

Table Data Indexes Model Constraints Grants Statistics UI Default

Add ColumnModify ColumnRename ColumnDrop ColumnRename

Column Name	Data Type	Nullable	Default	Primary Key
ID	NUMBER	No	-	1
DESCRIPTION	VARCHAR2(40)	No	-	-
				1 - 2

Et voilà votre table créée

Nous voici donc avec une table vide. Reste donc à la remplir.

COMPETENCES						
Table	Data	Indexes	Model	Constraints	Grants	Statistics
Query	Count Rows	Insert Row				

This table contains no data.

L'onglet **Data** permet de visualiser le contenu d'une table. Comme son nom le suggère, vous pouvez cliquer sur le bouton **Insert Row** pour saisir une nouvelle *ligne* dans la table.

Create Row
Cancel
Create
Create and Create Another

Table: COMPETENCES


* Id

* Description

▼ **Table Information**

Name	Type	Length	Precision	Scale	Null	Default	ID	Comments
ID	NUMBER	22			N		1	
DESCRIPTION	VARCHAR2	40			N		2	
row(s) 1 - 2 of 2								

Oracle Application Express vous permet de saisir les différents champs de l'enregistrement à ajouter. Les champs marqués avec une étoile correspondent aux champs marqués *NOT NULL* – et sont normalement obligatoires. Cependant, ici, le champ ID peut ne pas être rempli, puisque la table a été créée pour renseigner automatiquement cette valeur à partir d'une séquence.

COMPETENCES		
Table	Data	Indexes Model Constraints Grants Statistics UI
Query	Count Rows	Insert Row
EDIT	ID	DESCRIPTION
	1	C++
	2	Oracle
	3	Java
	4	Programmation orientée objets
	5	SQL
	6	Python
	7	JEE
row(s) 1 - 7 of 7		
Download		

Quand vous aurez terminé d'ajouter vos enregistrements, vous pourrez vérifier dans l'onglet **Data** que les données apparaissent bien. Remarquez que le champ *ID* a bien été rempli automatiquement.

Avec Oracle Application Express (Commande SQL)

Bon, l'interface graphique c'est sympa. Mais il s'agit d'Oracle, pas d'Access! Donc intéressons nous maintenant à ce qui fait le pain quotidien du développeur utilisant une base de données: SQL. Je suppose que vous connaissez un minimum ce langage. De toute façon, nous allons rester extrêmement basique, l'objectif étant surtout de voir comment utiliser l'interface web d'*Application Express* pour exécuter des requêtes SQL.

User: SYLVAIN

Home



Pour exécuter une commande SQL à partir d'*Oracle Application Express*, choisissez **SQL > SQL Commands > Enter Command**.




Piège:

L'éditeur de commande d'Application Express, dans cette version (2.1.0.00.39) n'accepte qu'une seule commande SQL à la fois. Si vous voulez absolument exécuter plusieurs commandes à la suite, il faudra passer par l'éditeur de scripts (**SQL > SQL scripts > create**). De façon alternative, vous pouvez aussi grouper vos commandes dans un bloc PL/SQL délimité par `begin/end`. Vous trouverez un tel exemple un peu plus loin.

User: SYLVAIN

Home > SQL > SQL Commands

☒ Autocommit Display 10  Save Run

```
CREATE table "EMPLOYES" (  
  ID NUMBER NOT NULL,  
  NOM VARCHAR2(40) NOT NULL,  
  PRENOM VARCHAR2(40) NOT NULL,  
  constraint "EMPLOYES_PK" primary key("ID")  
)
```

Results Explain Describe Saved SQL History


Table created.

0.28 seconds

Tout comme précédemment, nous allons avoir besoin de créer une table, mais aussi une nouvelle séquence et un trigger pour initialiser la clé primaire. Or *Oracle Application Express* n'autorise qu'une seule requête à la fois. Nous allons donc devoir procéder en trois étapes. Tout d'abord en créant la table ...

User: SYLVAIN

Home > SQL > **SQL Commands**

☒ Autocommit Display 10  **Save** **Run**

```
CREATE sequence "EMPLOYES_SEQ"
```

Results Explain Describe Saved SQL History

Sequence created.

0.01 seconds

... puis en définissant une nouvelle séquence ...

User: SYLVAIN

Home > SQL > **SQL Commands**☒ Autocommit Display 10  **Save** **Run**

```
CREATE trigger "BI_EMPLOYES"
before insert on "EMPLOYES"
for each row
begin
    select "EMPLOYES_SEQ".nextval into :NEW.ID from DUAL
end;
```

Results Explain Describe Saved SQL History


Trigger created.

0.06 seconds

... et enfin en créant un *TRIGGER* pour utiliser la séquence pour initialiser le champ *ID*.

User: SYLVAIN

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 

Save

Run

```
begin
  INSERT INTO "EMPLOYES" ("PRENOM", "NOM") VALUES ('Paul', 'Smith');
  INSERT INTO "EMPLOYES" ("PRENOM", "NOM") VALUES ('John', 'Doe');
  -- etc
end|
```

Finalement, il nous reste à peupler la table nouvellement créée. Toujours parce qu'*Oracle Application Express* n'accepte qu'une seule commande SQL ou PL/SQL à la fois, j'ai groupé mes deux requêtes dans un bloc PL/SQL.

Avec SQL*Plus

Outre l'interface d'administration *Application Express*, *Oracle Express Edition* est livré avec un ensemble d'outils en ligne de commande. Vous les retrouverez dans le dossier `$ORACLE_HOME/bin`. Celui qui va nous intéresser ici est **SQL*Plus** (`$ORACLE_HOME/bin/sqlplus`). Il s'agit d'un client en ligne de commande pour un serveur de base de données Oracle.

**ORACLE_HOME et PATH**

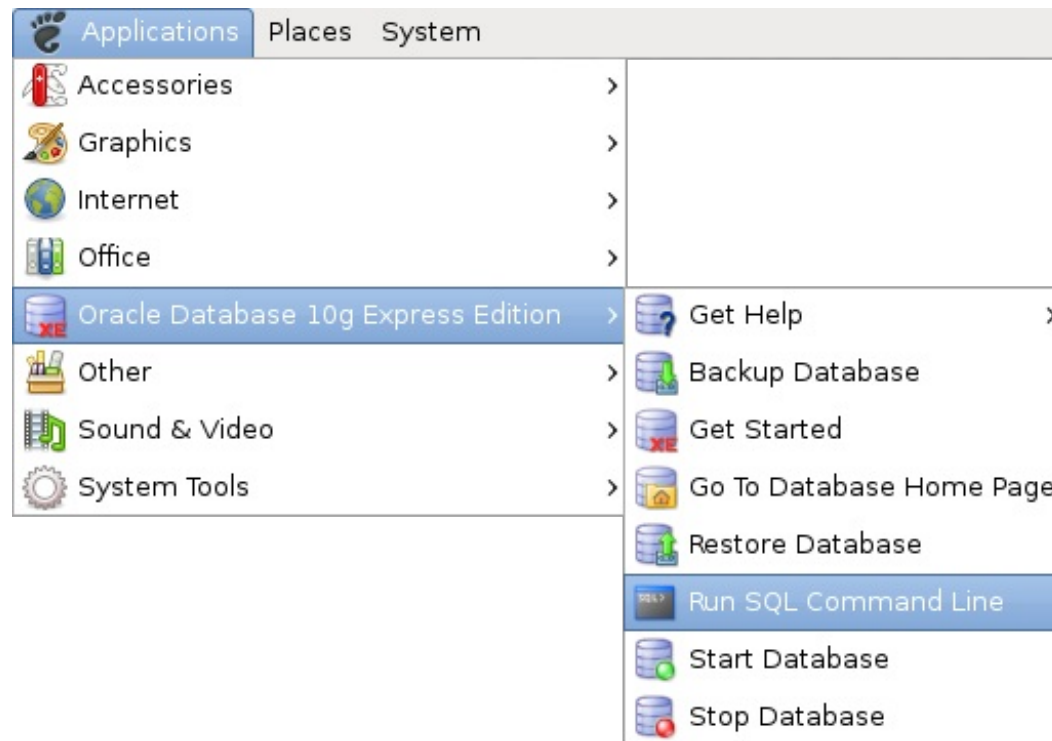
Pour être conforme à la *politique* de Debian, l'installation n'a pas modifié le *PATH* pour y ajouter le chemin vers les exécutables installés. Tout en restant conforme à cette politique, le paquet aurait pu créer des liens à partir de `/usr/bin`. Mais ce n'est pas le cas.

Or si vous souhaitez utiliser ces programmes à partir de la ligne de commande, vous n'aurez certainement pas envie de retaper le chemin complet à chaque fois...

Ajoutez donc ces définitions à la fin de votre *profile* (ou dans *profile.d* ou *bashrc* ou autre selon votre configuration):

```
export ORACLE_HOME=/usr/lib/oracle/xe/app/oracle/product/10.2.0/server
export PATH="$PATH:$ORACLE_HOME/bin"
```

Dorénavant, dans tout nouveau shell vous aurez directement accès aux *binaires* d'Oracle.



Sous Gnome, vous pouvez lancer **SQL*Plus** via le menu ajouté par l'installateur d'*Oracle XE*. C'est la solution de facilité, puisque le script associé à ce menu prend en charge, entre autres, l'initialisation des variables d'environnement *ORACLE_HOME* et *NLS_LANG*. L'inconvénient étant que vous ne souhaitez peut-être pas toujours utiliser **SQL*Plus** via une session interactive...

Nous allons nous servir de *SQL*Plus* pour créer notre dernière table. Celle qui fera l'association entre les employés et leurs compétences respectives. Mais tout d'abord lançons *SQL*Plus*:

```
sh$ sqlplus

SQL*Plus: Release 10.2.0.1.0 - Production on Sat Jan 16 18:09:48 2010

Copyright (c) 1982, 2005, Oracle. All rights reserved.

Enter user-name: sylvain@localhost
Enter password:
```



```
Connected to:
Oracle Database 10g Express Edition Release 10.2.0.1.0 - Production

SQL>
```

Comme vous le remarquez, à la connexion *SQL*Plus* demande sous quelle identité vous souhaitez vous connecter. La seule chose à noter ici, est qu'il faut aussi préciser sur quel hôte s'exécute le serveur auquel vous souhaitez vous connecter. D'où *sylvain@localhost*. Vous remarquerez aussi que dans Oracle, le nom d'utilisateur n'est pas sensible à la casse. *SYLVAIN* créé au début de cet article est bien le même utilisateur que *sylvain*.

```
SQL> CREATE TABLE "EMPLOYES_COMPETENCES" (
2      "EMPLOYEE_ID" NUMBER REFERENCES "EMPLOYES" ("ID") ,
3      "COMPETENCE_ID" NUMBER REFERENCES "COMPETENCES" ("ID") ,
4      CONSTRAINT "EMPLOYES_COMPETENCES_PK"
5              PRIMARY KEY ("EMPLOYEE_ID", "COMPETENCE_ID")
6  )
7  /
```

Table created.

Reste à peupler cette table. Histoire de se rappeler les *ID* de nos employés et des compétences (et de vérifier le fonctionnement des requêtes *SELECT* :))

```
SQL> SELECT * FROM EMPLOYES;
```

ID	NOM	PRENOM
1	Smith	Paul
2	Doe	John

```
SQL> SELECT * FROM COMPETENCES;
```

ID	DESCRIPTION
1	C++
7	JEE
3	Java
2	Oracle

```
4 Programmation orientée objets
6 Python
5 SQL
```

```
7 rows selected.
```



NLS et les accents

Non, il ne s'agit pas d'une erreur lors de la rédaction de cet article: L'accent dans *orientée objets* n'apparaît pas dans *SQL*Plus*. Pourtant il est bien présent dans les données enregistrées!

L'explication s'appelle *NLS – National Language Support*. Cette fonctionnalité fournit le support pour l'internationalisation. Avec en particulier, la gestion de l'encodage des caractères. Afin d'effectuer son travail correctement, NLS nécessite qu'on lui indique les règles à utiliser. A la fois sur le serveur (ce qui a été fait automatiquement à l'installation). Et sur le client (ce que nous n'avons pas fait).

Après cette introduction théorique, l'explication pratique que vous attendez. Il faut définir la variable d'environnement `NLS_LANG` pour préciser la *locale* de votre machine cliente. Par exemple:

```
sh$ export NLS_LANG=AMERICAN_AMERICA.AL32UTF8 # Correspond à en_US.utf8
```

La bonne nouvelle, c'est qu'*Oracle XE* est livré avec un script qui permet de déterminer la *bonne* valeur pour `NLS_LANG` à partir de la configuration des variables d'environnement standards `LANG` et `LC_ALL`:

```
sh$ export NLS_LANG=`$ORACLE_HOME/bin/nls_lang.sh`
```

Comme pour `ORACLE_HOME`, si vous envisagez d'utiliser fréquemment *SQL*Plus*, je vous conseille d'ajouter la définition ci-dessus dans votre *profile*.

Maintenant que la table est créée, essayons quelques variations autour de la requête *INSERT* pour la peupler:

```
SQL> INSERT INTO "EMPLOYES_COMPETENCES" VALUES (1,1);

1 row created.

SQL> BEGIN
2     INSERT INTO "EMPLOYES_COMPETENCES" VALUES (1,3);
3     INSERT INTO "EMPLOYES_COMPETENCES" VALUES (1,4);
4     INSERT INTO "EMPLOYES_COMPETENCES" VALUES (1,6);
5 END;
6 /

PL/SQL procedure successfully completed.

SQL> INSERT INTO "EMPLOYES_COMPETENCES" ("EMPLOYE_ID", "COMPETENCE_ID")
2     SELECT 2, ID FROM COMPETENCES WHERE DESCRIPTION IN ('Java', 'Oracle', 'SQL');

3 rows created.
```



Piège:

Si vous voulez rendre vos modifications immédiatement visibles aux autres clients (dont *Oracle Application Express*), n'oubliez pas de confirmer la transaction:

```
SQL> COMMIT;

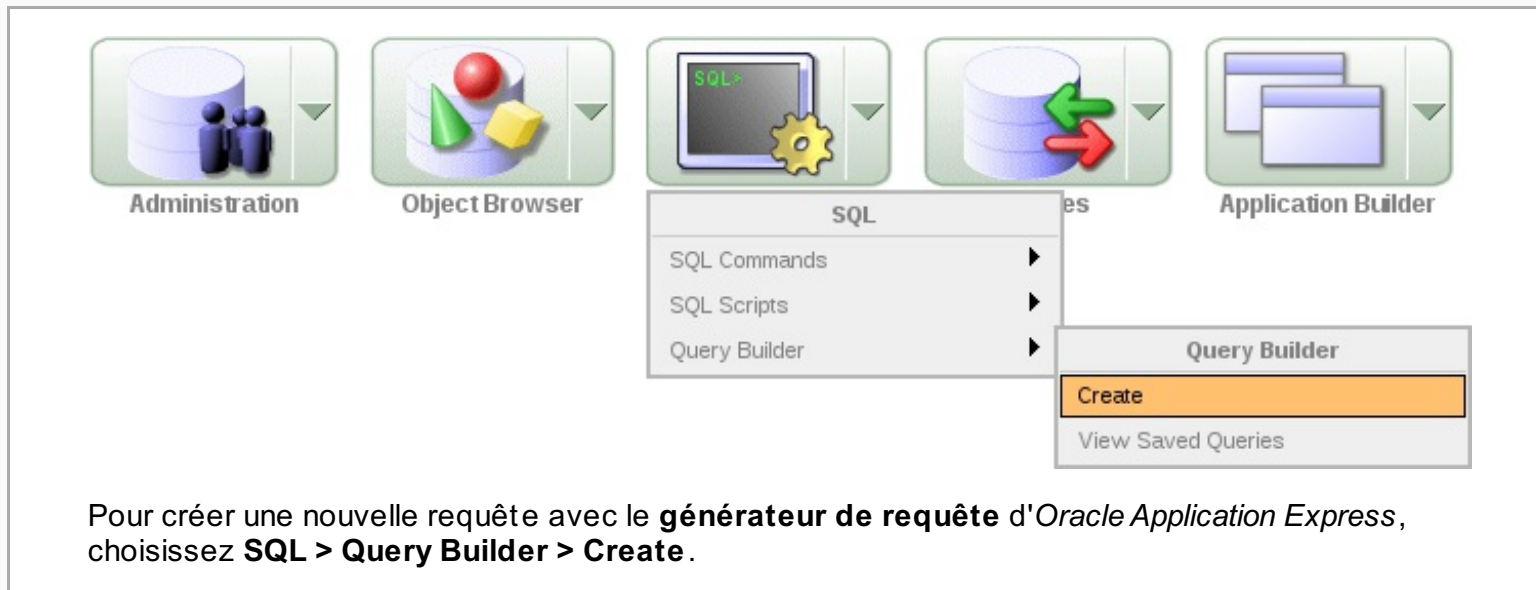
Commit complete.
```

Dans le cas contraire, les modifications ne seront visibles que quand vous quitterez *SQL*Plus* avec la commande *EXIT* ou *QUIT* (ce qui entraîne un *commit* implicite):

```
SQL> EXIT
Disconnected from Oracle Database 10g Express Edition Release 10.2.0.1.0 - Production
```

Utiliser le générateur de requête

Pour terminer ce tour d'horizon des fonctionnalités élémentaires d'Oracle XE, nous allons jeter un oeil au **générateur de requête** (*Query Builder*) que vous trouverez dans *Application Express*:



Celui-ci vous permet de construire graphiquement une requête *SELECT complexe*. Son utilisation est intuitive, et je ne vais pas trop m'étendre dessus.



L'**éditeur de requête** (*Query builder*) d'*Oracle Application Express* permet de créer graphiquement des requêtes SELECT. Son utilisation est intuitive. En résumé, dans la liste située à gauche, vous sélectionnez les **tables** impliquées dans la requête (1). En cochant les champs dans les tables, vous choisissez les **colonnes** à afficher dans le résultat (2). Enfin, les **jointures** sont définies en cliquant sur la case à droite du nom des colonnes impliquées (3).

Pour donner un exemple assez simple, dans la copie d'écran ci-dessous, j'ai utilisé l'éditeur pour construire une requête permettant de lister les compétences de chaque employé. Pour cet article, cela me permet surtout de m'assurer que je retrouve bien les données insérées via *SQL*Plus* dans *Application Express*. Ou autrement dit, de vérifier que les données insérées ont bien été enregistrées de façon permanente dans la base de données – et sont donc accessibles aux autres *clients* d'Oracle:



Conditions SQL Results Saved SQL

```
select  "COMPETENCES"."DESCRIPTION" as "DESCRIPTION",
        "EMPLOYES"."NOM" as "NOM",
        "EMPLOYES"."PRENOM" as "PRENOM"
from    "EMPLOYES" "EMPLOYES",
        "EMPLOYES_COMPETENCES" "EMPLOYES_COMPETENCES",
        "COMPETENCES" "COMPETENCES"
where   "COMPETENCES"."ID"="EMPLOYES_COMPETENCES"."COMPETENCE_ID"
and     "EMPLOYES_COMPETENCES"."EMPLOYEE_ID"="EMPLOYES"."ID"
```

Une fois la requête construite, son exécution donne le

résultat ci-contre. Comme vous pourrez le vérifier, il s'agit bien du résultat attendu.

Conditions SQL Results Saved SQL

DESCRIPTION	NOM	PRENOM
C++	Smith	Paul
Java	Smith	Paul
Programmation orientée objets	Smith	Paul
Python	Smith	Paul
Oracle	Doe	John
Java	Doe	John
SQL	Doe	John
row(s) 1 - 7 of 7		

[Download](#)

Conclusion

Voilà: ce premier contact avec *Oracle Express Edition* touche à sa fin. D'un point de vue fonctionnel, nous n'avons fait qu'effleurer les possibilités offertes par *Oracle XE* et *Oracle Application Express*. Mais au moins, vous avez pu installer ce produit, vérifier qu'il était opérationnel, et découvrir les outils de base nécessaires à son utilisation. Mais comme souvent, rien ne remplace la pratique! Alors, à vous de jouer, maintenant...

Références

(en) [NLS FAQ](#)

(fr) <http://fadace.developpez.com/oracle/nls/> – Encore des informations sur NLS

Récupérée de « http://www.chicoree.fr/w/Installer_Oracle_XE »

Qu'est-ce que c'est?

Chicoree.fr est le site personnel de [Sylvain Leroux](#).

Mentions légales

- [À propos de Chicoree](#)
- [Avertissements](#)

Qu'y trouve-t-on?

- [Debian](#)
- [Derby](#)
- [Java](#)
- [Jython](#)
- [Python](#)
- [Pages liées](#)
- [Suivi des pages liées](#)
- [Lien historique](#)



Dernière modification de cette page le 16 avril 2010 à 21:14. Cette page a été consultée 1 063 fois.