

## Définition des standards de dénomination SQL

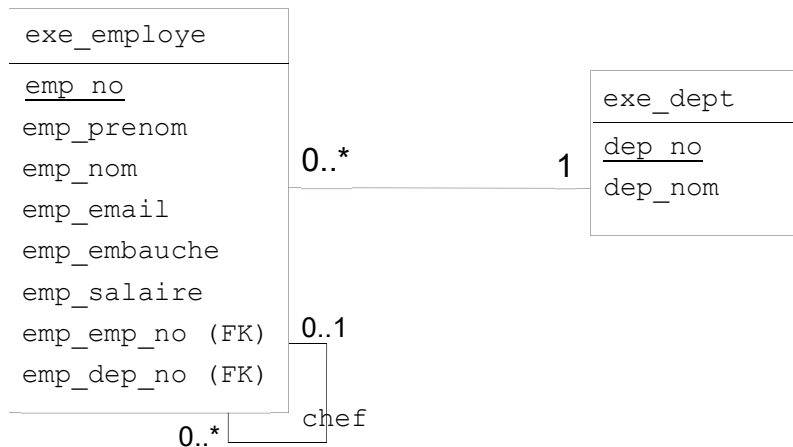
### Standards SQL

Les SGBD tels qu'Oracle peuvent manipuler des objets (tables, colonnes, vues, procédures) ayant des noms quelconques, pourvu que ceux-ci ne correspondent pas à des mots réservés du langage SQL.

D'autre part, et contrairement à d'autres, le langage SQL n'est pas sensible à la casse (minuscules / majuscules), cela permet donc en principe d'écrire la même requête SQL sous diverses formes. Néanmoins, il convient, pour faciliter la lecture et le débogage des requêtes et des scripts de suivre une convention d'écriture. Ainsi, dans le cadre de ce cours, nous définirons un standard pour le langage SQL, et pour le langage PL/SQL.

En ce qui concerne le langage SQL, la convention s'applique à la fois à la dénomination des objets ainsi qu'à la façon de rédiger des requêtes SQL.

Un certain nombre de standards, ayant pour but de vous faciliter la vie, se trouvent décrits dans le tableau ci-dessous.



Élément	Standard	Exemple
Mots-clés	Tous les mots clés du langage SQL seront écrits en <b>majuscule</b> , y compris les noms de fonctions utilisées.	<b>SELECT * FROM</b> exe_employe
Objets manipulés (tables, colonnes, vues, proc, ...)	Les objets manipulés seront écrits en <b>minuscule</b> , même si Oracle ne fait pas la différence.	SELECT emp_nom FROM exe_employe
Nom de tables	<ul style="list-style-type: none"> <li>Le nom dépend du contenu</li> <li>Néanmoins, il est utile de préfixer le nom de la table du contexte dans lequel on se trouve</li> </ul>	Si le contexte est la « Gestion d'une bibliothèque » on prefixera le nom des tables par 'bib_'; Les tables d' <b>exe</b> mple seront donc nommées <b>exe_employe</b> et <b>exe_dept</b>
Nom des champs	<ul style="list-style-type: none"> <li>En <b>minuscules</b></li> <li>Préfixés des 3 premières lettres du nom de la table suivi d'un « underscore »</li> </ul>	<b>emp_nom</b>

<u>Élément</u>	<u>Standard</u>	<u>Exemple</u>
Nom des clés étrangères	<ul style="list-style-type: none"> <li>En minuscules</li> <li>Préfixés des 3 premières lettres du nom de la table suivi d'un « underscore », puis du nom de la clé primaire dans la table référencée</li> </ul>	<b>emp_dep_no</b>
Nom des types de données	En <b>majuscules</b>	<pre>CREATE TABLE exe_dept (   dep_no  <b>NUMBER(6)</b>,   dep_nom <b>VARCHAR2(20)</b>,   dep_ville <b>VARCHAR2(20)</b> );</pre>
<u>Contraintes :</u> Clé primaire Clé étrangère Unique Check Not null	pk_nomTable fk_nomTable_nomTableEtrangère uk_nomColonne ch_nomColonne nn_nomColonne  NB : comme un nom de contrainte est unique, il faudra parfois adapter ces standards au cas particulier  Les contraintes de colonne suivent généralement la description du champ concerné (chaque fois que cela est possible),  le cas échéant après la déclaration de tous les champs, à la fin de la commande CREATE TABLE	<b>pk_exe_employe</b> <b>fk_exe_employe_dept</b> <b>uk_dep_nom</b> <b>ch_emp_npa</b> <b>nn_emp_nom</b>  NB : pk : Primary Key fk : Foreign Key uk : Unique Key ch : CCheck nn : Not Null  <pre>CREATE TABLE exe_dept (   dep_no NUMBER(6) <b>CONSTRAINT pk_exe_dept</b>                                 PRIMARY KEY,   dep_nom VARCHAR2 (20) <b>CONSTRAINT nn_dep_nom</b>                                 NOT NULL,   dep_ville VARCHAR2 (20) );</pre> <pre>CREATE TABLE exe_dept (   dep_no  NUMBER(6),   dep_nom VARCHAR2 (20),   dep_ville VARCHAR2 (20),   <b>CONSTRAINT pk_exe_dept</b> PRIMARY KEY (dep_no),   <b>CONSTRAINT nn_dep_nom</b> CHECK (dep_nom IS NOT NULL) );</pre>
Vues	vw_nomDeLaVue	<pre>CREATE VIEW <b>vw_employes_et_dept</b> AS   SELECT emp_nom, emp_prenom, dep_nom   FROM exe_employe   JOIN exe_dept ON dep_no = emp_dep_no;</pre>
Synonymes	nomDuSynonyme	<pre>CREATE SYNONYM <b>bidon</b> FOR sys.dual;</pre>
Séquences	sq_nomColonne ou sq_nomDuSchéma <ul style="list-style-type: none"> <li>si la séquence est utilisée par plusieurs tables dans un même schéma</li> </ul>	<pre>CREATE SEQUENCE <b>sq_emp_numero</b>   START WITH 100   INCREMENT BY 10   NOCYCLE;</pre>