

## Documentation SQL en ligne

<https://docs.oracle.com/en/database/oracle/oracle-database/21/sqlrf/index.html>

## Base de données

Nous allons travailler sur la Partie 3 du cours. Considérons la base de données dont le schéma et l'extension sont donnés ci-dessous.

EMP (EMPNO, ENAME, JOB, MGR#, HIREDATE, SAL, COMM, DEPTNO#)

DEPT (DEPTNO, DNAME, LOC)

Clés primaires

Clés étrangères#

MGR est le numéro d'employé (EMPNO) du manager de l'employé courant.

### EMP

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-dec-80	800.00	NULL	20
7499	ALLEN	SALESMAN	7698	20-feb-81	1600.00	300.00	30
7521	WARD	SALESMAN	7698	22-feb-81	1250.00	500.00	30
7566	JONES	MANAGER	7839	02-apr-81	2975.00	NULL	20
7654	MARTIN	SALESMAN	7698	28-sep-81	1250.00	1400.00	30
7698	BLAKE	MANAGER	7839	01-may-81	2850.00	NULL	30
7782	CLARK	MANAGER	7839	09-jun-81	2450.00	NULL	10
7839	KING	PRESIDENT	NULL	17-nov-81	5000.00	NULL	10
7844	TURNER	SALESMAN	7698	08-sep-81	1500.00	0.00	30
7876	ADAMS	CLERK	7788	23-sep-87	1100.00	NULL	20
7900	JAMES	CLERK	7698	03-dec-81	950.00	NULL	30
7902	FORD	ANALYST	7566	03-dec-81	3000.00	NULL	20
7934	MILLER	CLERK	7782	23-jan-82	1300.00	NULL	10

### DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW-YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

## Connexion

Nous allons utiliser Oracle Live SQL.

1. Allez sur <https://livesql.oracle.com> et cliquez sur START CODING.
2. Si vous n'avez pas de compte Oracle, créez un compte (c'est gratuit).
3. Connectez-vous et travaillez dans la SQL Worksheet.

### Exercice 1 : Création de la base de données, contraintes d'intégrité

1. Créer la table DEPT. Le numéro de département DEPTNO est la clé primaire. Sauvegarder le script SQL (Save) et exécuter le script (My scripts/Run).
2. Remplir la table DEPT avec des commandes SQL INSERT INTO (SQL Worksheet).
3. Télécharger le script SQL <https://eric.univ-lyon2.fr/jdarmont/docs/ora-emp.sql> (menu My Scripts / Upload Script). Exécuter le script. Il crée la table EMP.
4. **À partir de maintenant, on travaille uniquement dans la SQL Worksheet.** Dans EMP, ajouter les nouveaux employés :  
<7369, 'WILSON', 'MANAGER', 7839, '17-nov-1991', 3500.00, 600.00, 10> ;  
<7657, 'WILSON', 'MANAGER', 7839, '17-nov-1991', 3500.00, 600.00, 50> ;  
<7657, 'WILSON', 'MANAGER', 7000, '17-nov-1991', 3500.00, 600.00, 10> ;  
<7657, 'WILSON', 'MANAGER', 7839, '17-nov-1991', 3500.00, 600.00, 10>.  
Remarques ?

### Exercice 2 : Mise à jour de la base de données

1. Changer la LOCALisation du département SALES de CHICAGO à PITTSBURGH.
2. Dans EMP, augmenter de 10 % le SALaire des vendeurs (SALESMAN) dont la COMMission est supérieure à 50 % du salaire.
3. Dans EMP, donner aux employés en poste après le 01/01/1982 (HIREDATE) et ayant une commission non spécifiée (NULL) une commission égale à la moyenne des commissions.
4. Dans DEPT, supprimer le département n° 20 (DEPTNO). Remarque ?

### Exercice 3 : Interrogation de la base de données

Exprimer en SQL les requêtes suivantes.

1. Nom (ENAME), salaire, commission, salaire + commission de tous les vendeurs.
2. Nom des vendeurs par ordre décroissant du ratio commission/salaire.
3. Nom des vendeurs dont la commission est inférieure à 25 % de leur salaire.
4. Nombre d'employés du département n° 10.
5. Nombre d'employés ayant une commission.
6. Nombre de fonctions (JOB) différentes.
7. Salaire moyen par fonction (sans tenir compte des commissions).
8. Total des salaires du département SALES.
9. Nom des employés avec le nom de leur département.
10. Nom, fonction et salaire de l'employé qui a le salaire le plus grand.
11. Nom des employés qui gagnent plus que JONES.
12. Nom des employés qui ont la même fonction que JONES.
13. Nom des employés qui ont le même manager que CLARK.
14. Nom et fonction des employés qui ont la même fonction et le même manager que TURNER.
15. Nom des employés qui ont été embauchés avant tous les employés du département n° 10.
16. Liste des employés avec leur nom et celui de son manager.
17. Nom des employés qui ne travaillent pas dans le même département que leur manager.

### Requêtes hiérarchiques

18. Structure hiérarchique des employés (NOM, JOB, EMPNO, MGR). L'employé au sommet de la hiérarchie n'a pas de manager.
19. Liste des employés qui dépendent de JONES. Indiquer leur niveau (LEVEL) dans la hiérarchie.
20. Salaire moyen pour chaque niveau d'employé.
21. Liste des employés qui travaillent pour JONES, sauf FORD.
22. Liste des employés qui travaillent pour JONES, sauf FORD et ceux qui travaillent pour FORD.

### Exercice 4 : Vues et catalogue du système

1. Créer la vue EMPDIR (EMPNO, ENAME) à partir de la table EMP. Vérifier son contenu.
2. Depuis la vue EMPDIR, modifier le nom de l'employé n° 7839 en 'DARMONT'. Consulter le contenu de la vue EMPDIR et de la table EMP.
3. Créer la vue EMPDEPT (EMPNO, ENAME, DEPTNO, DNAME) partir des tables EMP et DEPT. Vérifier son contenu. Quel est l'intérêt de définir cette vue ?
4. À travers la vue EMPDEPT, modifier le nom de l'employé n° 7698 en 'SINBAD'. Que se passe-t-il ?
5. Insérer un n-uplet quelconque dans la vue EMPDEPT. Que se passe-t-il ?
6. Lister toutes les tables qui vous sont accessibles (nom et propriétaire) en interrogeant la vue système ALL\_TABLES.
7. Lister les tables et les vues de votre compte, ainsi que leurs types (table ou vue), à l'aide de la vue système USER\_CATALOG.
8. Lister toutes les contraintes d'intégrité définies sur vos tables à l'aide de la vue système USER\_CONSTRAINTS. Afficher pour chaque contrainte son nom, la table à laquelle elle s'applique, son type, et sa « condition de recherche ».
9. À partir de la vue système USER\_TAB\_COLUMNS, afficher les attributs de la table EMP.
10. À partir de la vue système USER\_TAB\_COLUMNS, afficher le nom des tables et des vues qui ont pour attribut DEPTNO. Commentaire ?

## Correction Exercise 1

1. CREATE TABLE DEPT( DEPTNO NUMBER(2),  
DNAME CHAR(20),  
LOC CHAR(20),  
CONSTRAINT DEPT\_CLEP PRIMARY KEY (DEPTNO),  
CONSTRAINT DEPT\_DOM1 CHECK (DNAME IN  
( 'ACCOUNTING', 'RESEARCH', 'SALES', 'OPERATIONS')));
2. INSERT INTO DEPT VALUES (10, 'ACCOUNTING', 'NEW-YORK');  
INSERT INTO DEPT VALUES (20, 'RESEARCH', 'DALLAS');  
INSERT INTO DEPT VALUES (30, 'SALES', 'CHICAGO');  
INSERT INTO DEPT VALUES (40, 'OPERATIONS', 'BOSTON');
4. INSERT INTO EMP VALUES  
(7657, 'WILSON', 'MANAGER', 7839, '17-NOV-1991', 3500.00, 600.00, 10);

## Correction Exercise 2

1. UPDATE DEPT SET LOC = 'PITTSBURGH' WHERE DNAME = 'SALES';
2. UPDATE EMP SET SAL = SAL \* 1.1 WHERE COMM > 0.5 \* SAL;
3. UPDATE EMP  
SET COMM = (SELECT AVG(COMM) FROM EMP)  
WHERE HIREDATE > '01-JAN-1982'  
AND COMM IS NULL;
4. DELETE FROM DEPT WHERE DEPTNO=20;

## Correction Exercise 3

1. SELECT ENAME, SAL, COMM, SAL + COMM FROM EMP WHERE JOB = 'SALESMAN';
2. SELECT ENAME FROM EMP ORDER BY COMM / SAL DESC;
3. SELECT ENAME FROM EMP WHERE COMM < .25 \* SAL;
4. SELECT COUNT(EMPNO) FROM EMP WHERE DEPTNO = 10;
5. SELECT COUNT(EMPNO) FROM EMP WHERE COMM IS NOT NULL;
6. SELECT COUNT(DISTINCT JOB) FROM EMP;
7. SELECT JOB, AVG(SAL) FROM EMP GROUP BY JOB;
8. SELECT SUM(SAL) FROM EMP, DEPT WHERE EMP.DEPTNO = DEPT.DEPTNO  
AND DNAME = 'SALES';
9. SELECT ENAME, DNAME FROM EMP, DEPT WHERE EMP.DEPTNO = DEPT.DEPTNO;
10. SELECT ENAME, JOB, SAL FROM EMP WHERE SAL = (SELECT MAX(SAL) FROM EMP);
11. SELECT ENAME FROM EMP WHERE SAL > (SELECT SAL FROM EMP WHERE ENAME='JONES');
12. SELECT ENAME FROM EMP WHERE JOB = (SELECT JOB FROM EMP WHERE ENAME='JONES')  
AND ENAME <> 'JONES';

13. SELECT ENAME FROM EMP WHERE MGR = (SELECT MGR FROM EMP WHERE ENAME='CLARK')  
AND ENAME <> 'CLARK';
14. SELECT ENAME FROM EMP WHERE (JOB, MGR) IN  
(SELECT JOB, MGR FROM EMP WHERE ENAME = 'TURNER')  
AND ENAME <> 'TURNER';
15. SELECT ENAME FROM EMP WHERE HIREDATE < ALL  
(SELECT HIREDATE FROM EMP WHERE DEPTNO = 10);
16. SELECT SUBALTERNE.ENAME, SUPERIEUR.ENAME FROM EMP SUBALTERNE, EMP SUPERIEUR  
WHERE SUBALTERNE.MGR = SUPERIEUR.EMPNO;
17. SELECT SUB.ENAME FROM EMP SUB, EMP SUP WHERE SUB.MGR = SUP.EMPNO  
AND SUB.DEPTNO <> SUP.DEPTNO;
18. SELECT LEVEL, EMPNO, ENAME, JOB, MGR FROM EMP  
CONNECT BY MGR = PRIOR EMPNO  
START WITH MGR IS NULL  
ORDER BY LEVEL;
19. SELECT LEVEL, ENAME FROM EMP  
WHERE ENAME <> 'JONES'  
CONNECT BY MGR = PRIOR EMPNO  
START WITH ENAME = 'JONES'  
ORDER BY LEVEL;
20. SELECT LEVEL, AVG(SAL) FROM EMP  
CONNECT BY MGR = PRIOR EMPNO  
START WITH MGR IS NULL  
GROUP BY LEVEL;
21. SELECT ENAME FROM EMP  
WHERE ENAME <> 'JONES' AND ENAME <> 'FORD'  
CONNECT BY MGR = PRIOR EMPNO  
START WITH ENAME = 'JONES';
22. SELECT ENAME FROM EMP  
WHERE ENAME <> 'JONES'  
CONNECT BY MGR = PRIOR EMPNO AND ENAME <> 'FORD'  
START WITH ENAME = 'JONES';

#### Correction Exercise 4

1. CREATE VIEW EMPDIR AS (SELECT EMPNO, ENAME FROM EMP);  
SELECT \* FROM EMPDIR;
2. UPDATE EMPDIR SET ENAME = 'DARMONT' WHERE EMPNO = 7839;  
SELECT \* FROM EMPDIR;  
SELECT \* FROM EMP;
3. CREATE VIEW EMPDEPT AS ( SELECT EMPNO, ENAME, E.DEPTNO, DNAME  
FROM EMP E, DEPT D  
WHERE E.DEPTNO = D.DEPTNO );  
SELECT \* FROM EMPDEPT;  
-- Interroger la vue EMPDEPT permet à un utilisateur de ne pas exprimer la  
-- jointure entre EMP et DEPT.
4. UPDATE EMPDEPT SET ENAME = 'SINBAD' WHERE EMPNO = 7698;  
SELECT \* FROM EMPDEPT;  
SELECT \* FROM EMP;

5. INSERT INTO EMPDEPT VALUES(9999, 'NEWEMP', 99, 'NEWDEPT');  
-- Cela ne fonctionne pas car la requête concerne les deux tables jointes  
-- EMP et DEPT.
6. SELECT TABLE\_NAME, OWNER FROM ALL\_TABLES;
7. SELECT TABLE\_NAME, TABLE\_TYPE FROM USER\_CATALOG;
8. SELECT CONSTRAINT\_NAME, TABLE\_NAME, CONSTRAINT\_TYPE, SEARCH\_CONDITION  
FROM USER\_CONSTRAINTS;
9. SELECT COLUMN\_NAME FROM USER\_TAB\_COLUMNS WHERE TABLE\_NAME = 'EMP';
10. SELECT TABLE\_NAME FROM USER\_TAB\_COLUMNS WHERE COLUMN\_NAME = 'DEPTNO';  
-- Les vues sont également présentes dans USER\_TAB\_COLUMNS.