

TD3 Base de données : le langage SQL

Diagne Moustapha id°21011573

6 mai 2022

Exercice 1 : Création de tables

- Créer la table EMP dont la clé est NoEMP et les attributs ont pour types :

Attribut	Type	Signification
NoEMP	NUMBER(4)	nombre sur au maximum 4 chiffres
NomEMP	VARCHAR2(10)	chaîne de caractères d'au maximum 10 caractères
EMPLOI	VARCHAR2(15)	chaîne de caractères d'au maximum 15 caractères
MGR	NUMBER(4)	nombre sur au maximum 4 chiffres
DateEMB	DATE	
SAL	NUMBER(7, 2)	nombre sur au maximum 7 chiffres dont deux après la virgule
COMM	NUMBER(7, 2)	nombre sur au maximum 7 chiffres dont deux après la virgule
NoDEPT	NUMBER(2)	nombre sur au maximum 2 chiffres

```
ma_bdd=# CREATE TABLE EMP(
ma_bdd=# NoEMP NUMERIC(4) NOT NULL,
ma_bdd=# NomEMP VARCHAR(10),
ma_bdd=# EMPLOI VARCHAR(15),
ma_bdd=# MGR NUMERIC(4),
ma_bdd=# DateEMB DATE,
ma_bdd=# SAL NUMERIC(7,2),
ma_bdd=# COMM NUMERIC(7,2),
ma_bdd=# NoDEPT NUMERIC(2),
ma_bdd=# CONSTRAINT EMP_clé_primaire PRIMARY KEY (NoEMP));
CREATE TABLE
ma_bdd=# \d emp
```

Column	Type	Collation	Nullable	Default
noemp	numeric(4,0)		not null	
nomemp	character varying(10)			
emploi	character varying(15)			
mgr	numeric(4,0)			
dateemb	date			
sal	numeric(7,2)			
comm	numeric(7,2)			
nodept	numeric(2,0)			

```
Indexes:
    "emp_clé_primaire" PRIMARY KEY, btree (noemp)
ma_bdd=#
```

- Remplir cette table par les données suivantes :

NoEMP	NomEMP	EMPLOI	MGR	DateEMB	SAL	COMM	NoDEPT
7369	SERGE	FONCTIONNAIRE	7902	17-DEC-1980	800		20
7499	BRAHIM	VENDEUR	7698	20-FEB-1981	1600	300	30
7521	NASSIMA	VENDEUR	7698	22-FEB-1981	1250	500	30
7566	LUCIE	GESTIONNAIRE	7839	2-APR-1981	2975		20
7654	MARTIN	VENDEUR	7698	28-SEP-1981	1250	1400	30
7698	BENJAMIN	GESTIONNAIRE	7839	1-MAY-1981	2850		30
7782	DAYANE	GESTIONNAIRE	7839	9-JUN-1981	2450		10
7788	ARIJ	ANALYSTE	7566	09-DEC-1982	3000		20
7839	MAYAR	PRESIDENT		17-NOV-1981	5000		10
7844	ROI	VENDEUR	7698	8-SEP-1981	1500	0	30
7876	VIRGINIE	FONCTIONNAIRE	7788	12-JAN-1983	1100		20
7900	LYNA	FONCTIONNAIRE	7698	3-DEC-1981	950		30
7902	ASMA	ANALYSTE	7566	3-DEC-1981	3000		20
7934	SIMONE	FONCTIONNAIRE	7782	23-JAN-1982	1300		10

```

ma_bdd=# INSERT INTO EMP
ma_bdd=# VALUES (7499, 'BRAHIM', 'VENDEUR', 7698, '20-FEB-1981', 1600, 300, 30),
ma_bdd=# (7521, 'NASSIMA', 'VENDEUR', 7698, '22-FEB-1981', 1250, 500, 30),
ma_bdd=# (7566, 'LUCIE', 'GESTIONNAIRE', 7839, '2-APR-1981', 2975, NULL, 20),
ma_bdd=# (7654, 'MARTIN', 'VENDEUR', 7698, '28-SEP-1981', 1250, 1400, 30),
ma_bdd=# (7698, 'BENJAMIN', 'GESTIONNAIRE', 7839, '1-MAY-1981', 2850, NULL, 30),
ma_bdd=# (7782, 'DAYANE', 'GESTIONNAIRE', 7839, '9-JUN-1981', 2450, NULL, 10),
ma_bdd=# (7788, 'ARIJ', 'ANALYSTE', 7566, '09-DEC-1982', 3000, NULL, 20),
ma_bdd=# (7839, 'MAYAR', 'PRESIDENT', NULL, '17-NOV-1981', 5000, NULL, 10),
ma_bdd=# (7844, 'ROI', 'VENDEUR', 7698, '8-SEP-1981', 1500, 0, 30),
ma_bdd=# (7876, 'VIRGINIE', 'FONCTIONNAIRE', 7788, '12-JAN-1983', 1100, NULL, 20),
ma_bdd=# (7900, 'LYNA', 'FONCTIONNAIRE', 7698, '3-DEC-1981', 950, NULL, 30),
ma_bdd=# (7902, 'ASMA', 'ANALYSTE', 7566, '3-DEC-1981', 3000, NULL, 20),
ma_bdd=# (7934, 'SIMONE', 'FONCTIONNAIRE', 7782, '23-JAN-1982', 1300, NULL, 10);
INSERT 0 13
ma_bdd=# SELECT * FROM EMP;

```

noemp	nomemp	emploi	mgr	dateemb	sal	comm	nodept
7369	SERGE	FONCTIONNAIRE	7902	1980-12-17	800.00		20
7499	BRAHIM	VENDEUR	7698	1981-02-20	1600.00	300.00	30
7521	NASSIMA	VENDEUR	7698	1981-02-22	1250.00	500.00	30
7566	LUCIE	GESTIONNAIRE	7839	1981-04-02	2975.00		20
7654	MARTIN	VENDEUR	7698	1981-09-28	1250.00	1400.00	30
7698	BENJAMIN	GESTIONNAIRE	7839	1981-05-01	2850.00		30
7782	DAYANE	GESTIONNAIRE	7839	1981-06-09	2450.00		10
7788	ARIJ	ANALYSTE	7566	1982-12-09	3000.00		20
7839	MAYAR	PRESIDENT		1981-11-17	5000.00		10
7844	ROI	VENDEUR	7698	1981-09-08	1500.00	0.00	30
7876	VIRGINIE	FONCTIONNAIRE	7788	1983-01-12	1100.00		20
7900	LYNA	FONCTIONNAIRE	7698	1981-12-03	950.00		30
7902	ASMA	ANALYSTE	7566	1981-12-03	3000.00		20
7934	SIMONE	FONCTIONNAIRE	7782	1982-01-23	1300.00		10

(14 rows)

3. Créer la table DEPT dont la clé est NoDEPT et les attributs ont pour types :

Attribut	Type
NoDEPT	NUMBER(2)
NomDEPT	VARCHAR2(14)
LOC	VARCHAR2(13))

```

ma_bdd=# CREATE TABLE DEPT(
ma_bdd(# NoDEPT NUMERIC(2) NOT NULL,
ma_bdd(# NomDEPT VARCHAR(14),
ma_bdd(# LOC VARCHAR(13),
ma_bdd(# CONSTRAINT DEPT_clé_primaire PRIMARY KEY (NoDEPT));
CREATE TABLE
ma_bdd=# \d DEPT

```

Column	Type	Collation	Nullable	Default
nodept	numeric(2,0)		not null	
nomdept	character varying(14)			
loc	character varying(13)			

Indexes:

```

"dept_clé_primaire" PRIMARY KEY, btree (nodept)

```

4. Remplir cette table par les données suivantes :

NoDEPT	NomDEPT	VILLE
10	COMPTABILITE	BREST
20	RECHERCHE	RENNES
30	VENTES	DINAR
40	GESTION	DINAN

```
ma_bdd=# CREATE TABLE DEPT(
ma_bdd=# NODEPT NUMERIC(2) NOT NULL,
ma_bdd=# NOmDEPT VARCHAR(14),
ma_bdd=# LOC VARCHAR(13),
ma_bdd=# CONSTRAINT DEPT_clé_primaire PRIMARY KEY (NoDEPT));
CREATE TABLE
```

```
ma_bdd=# SELECT * FROM DEPT;
nodept | nomdept | loc
-----+-----+-----
10 | COMPTABILITE | BREST
20 | RECHERCHE | RENNES
30 | VENTES | DINAR
40 | GESTION | DINAN
(4 rows)
```

5. Faire en sorte que la clé étrangère soit correctement définie, après avoir défini et rempli les deux tables.

```
ma_bdd=# ALTER TABLE EMP ADD CONSTRAINT EMP_clé_étrangère FOREIGN KEY (NoDEPT) REFERENCES DEPT(NoDEPT);
ALTER TABLE
ma_bdd=# \d EMP
Table "public.emp"
Column | Type | Collation | Nullable | Default
-----+-----+-----+-----+-----
noemp | numeric(4,0) | | not null |
nomemp | character varying(10) | | |
emploi | character varying(15) | | |
mgr | numeric(4,0) | | |
dateemb | date | | |
sal | numeric(7,2) | | |
comm | numeric(7,2) | | |
nodept | numeric(2,0) | | |
Indexes:
    "emp_clé_primaire" PRIMARY KEY, btree (noemp)
Foreign-key constraints:
    "emp_clé_étrangère" FOREIGN KEY (nodept) REFERENCES dept(nodept)
```

Exercice 2 : Requêtes SQL simples

1. Afficher tous les tuples de la table EMP :

```
ma_bdd=# SELECT * FROM EMP;
noemp | nomemp | emploi | mgr | dateemb | sal | comm | nodept
-----+-----+-----+-----+-----+-----+-----+-----
7369 | SERGE | FONCTIONNAIRE | 7902 | 1980-12-17 | 800.00 | | 20
7499 | BRAHIM | VENDEUR | 7698 | 1981-02-20 | 1600.00 | 300.00 | 30
7521 | NASSIMA | VENDEUR | 7698 | 1981-02-22 | 1250.00 | 500.00 | 30
7566 | LUCIE | GESTIONNAIRE | 7839 | 1981-04-02 | 2975.00 | | 20
7654 | MARTIN | VENDEUR | 7698 | 1981-09-28 | 1250.00 | 1400.00 | 30
7698 | BENJAMIN | GESTIONNAIRE | 7839 | 1981-05-01 | 2850.00 | | 30
7782 | DAYANE | GESTIONNAIRE | 7839 | 1981-06-09 | 2450.00 | | 10
7788 | ARIJ | ANALYSTE | 7566 | 1982-12-09 | 3000.00 | | 20
7839 | MAYAR | PRESIDENT | | 1981-11-17 | 5000.00 | | 10
7844 | ROI | VENDEUR | 7698 | 1981-09-08 | 1500.00 | 0.00 | 30
7876 | VIRGINIE | FONCTIONNAIRE | 7788 | 1983-01-12 | 1100.00 | | 20
7900 | LYNA | FONCTIONNAIRE | 7698 | 1981-12-03 | 950.00 | | 30
7902 | ASMA | ANALYSTE | 7566 | 1981-12-03 | 3000.00 | | 20
7934 | SIMONE | FONCTIONNAIRE | 7782 | 1982-01-23 | 1300.00 | | 10
(14 rows)
```

2. À partir du contenu de la table EMP afficher les noms et les emplois de tous.

```
ma_bdd=# SELECT nomemp, emploi FROM EMP;
nomemp | emploi
-----+-----
SERGE | FONCTIONNAIRE
BRAHIM | VENDEUR
NASSIMA | VENDEUR
LUCIE | GESTIONNAIRE
MARTIN | VENDEUR
BENJAMIN | GESTIONNAIRE
DAYANE | GESTIONNAIRE
ARIJ | ANALYSTE
MAYAR | PRESIDENT
ROI | VENDEUR
VIRGINIE | FONCTIONNAIRE
LYNA | FONCTIONNAIRE
ASMA | ANALYSTE
SIMONE | FONCTIONNAIRE
(14 rows)
```

3. Afficher les tuples correspondant au département 20 dans la table EMP.

```
ma_bdd=# SELECT * FROM EMP WHERE nodept=20;
noemp | nomemp | emploi | mgr | dateemb | sal | comm | nodept
-----+-----+-----+----+-----+----+-----+-----
 7369 | SERGE | FONCTIONNAIRE | 7902 | 1980-12-17 | 800.00 | | 20
 7566 | LUCIE | GESTIONNAIRE | 7839 | 1981-04-02 | 2975.00 | | 20
 7788 | ARIJ | ANALYSTE | 7566 | 1982-12-09 | 3000.00 | | 20
 7876 | VIRGINIE | FONCTIONNAIRE | 7788 | 1983-01-12 | 1100.00 | | 20
 7902 | ASMA | ANALYSTE | 7566 | 1981-12-03 | 3000.00 | | 20
(5 rows)
```

4. Rechercher et afficher les noms et salaires des employés exerçant en tant que vendeur et ayant une commission.

```
ma_bdd=# SELECT nomemp, sal
ma_bdd-# FROM EMP
ma_bdd-# WHERE emploi= 'VENDEUR' AND comm IS NOT NULL ;
nomemp | sal
-----+----
BRAHIM | 1600.00
NASSIMA | 1250.00
MARTIN | 1250.00
ROI | 1500.00
(4 rows)
```

5. Afficher tous les numéros de département. Est-il possible d'améliorer le contenu d'affichage? Commenter.

```
ma_bdd=# SELECT nodept
ma_bdd-# FROM EMP
ma_bdd-# GROUP BY nodept /*Évite la redondance*/
ma_bdd-# ORDER BY nodept ASC;
nodept
-----
 10
 20
 30
(3 rows)
```

Il est effectivement possible d'améliorer l'affichage en utilisant GROUP BY (supprime les doublons) et ORDER BY (pour le tri).

6. Lister les gestionnaires. On ne s'intéressera qu'à leurs noms, leurs salaires et leurs commissions.

```
ma_bdd=# SELECT nomemp, sal, comm
ma_bdd-# FROM EMP
ma_bdd-# WHERE emploi='GESTIONNAIRE';
nomemp | sal | comm
-----+----+-----
LUCIE | 2975.00 |
BENJAMIN | 2850.00 |
DAYANE | 2450.00 |
(3 rows)
```

7. Déterminer la ville du département de ASMA.

```
ma_bdd=# SELECT loc
ma_bdd-# FROM EMP, DEPT
ma_bdd-# WHERE EMP.nodept = DEPT.nodept
ma_bdd-# AND nomemp='ASMA';
loc
-----
RENNES
(1 row)
```

8. On s'intéresse aux employés gagnant entre 1000 e et 3000 e. Proposer une requête assurant cette tâche.

```
ma_bdd=# SELECT *
ma_bdd=# FROM EMP
ma_bdd=# WHERE sal BETWEEN 1000 AND 3000;
```

noemp	nomemp	emploi	mgr	dateemb	sal	comm	nodept
7499	BRAHIM	VENDEUR	7698	1981-02-20	1600.00	300.00	30
7521	NASSIMA	VENDEUR	7698	1981-02-22	1250.00	500.00	30
7566	LUCIE	GESTIONNAIRE	7839	1981-04-02	2975.00		20
7654	MARTIN	VENDEUR	7698	1981-09-28	1250.00	1400.00	30
7698	BENJAMIN	GESTIONNAIRE	7839	1981-05-01	2850.00		30
7782	DAYANE	GESTIONNAIRE	7839	1981-06-09	2450.00		10
7788	ARIJ	ANALYSTE	7566	1982-12-09	3000.00		20
7844	ROI	VENDEUR	7698	1981-09-08	1500.00	0.00	30
7876	VIRGINIE	FONCTIONNAIRE	7788	1983-01-12	1100.00		20
7902	ASMA	ANALYSTE	7566	1981-12-03	3000.00		20
7934	SIMONE	FONCTIONNAIRE	7782	1982-01-23	1300.00		10

(11 rows)

9. Lister les noms des départements dont le nom de ville commence par R ou D.

```
ma_bdd=# SELECT nomdept
ma_bdd=# FROM DEPT
ma_bdd=# WHERE loc LIKE 'R%' OR loc LIKE 'D%';
          nomdept
-----
 RECHERCHE
  VENTES
  GESTION
(3 rows)
```

10. Lister les noms des départements dont le nom de ville contient un R ou un D.

```
ma_bdd=# SELECT nomdept
ma_bdd=# FROM DEPT
ma_bdd=# WHERE loc LIKE '%R%' OR loc LIKE '%D%';
          nomdept
-----
COMPTABILITE
 RECHERCHE
  VENTES
  GESTION
(4 rows)
```

11. Dans l'ordre décroissant des salaires, lister les noms, les fonctions et les salaires des employés gagnant moins de 3000 e.

```
ma_bdd=# SELECT nomemp, emploi, sal
ma_bdd=# FROM EMP
ma_bdd=# WHERE sal < 3000
ma_bdd=# ORDER BY sal DESC;
```

nomemp	emploi	sal
LUCIE	GESTIONNAIRE	2975.00
BENJAMIN	GESTIONNAIRE	2850.00
DAYANE	GESTIONNAIRE	2450.00
BRAHIM	VENDEUR	1600.00
ROI	VENDEUR	1500.00
SIMONE	FONCTIONNAIRE	1300.00
MARTIN	VENDEUR	1250.00
NASSIMA	VENDEUR	1250.00
VIRGINIE	FONCTIONNAIRE	1100.00
LYNA	FONCTIONNAIRE	950.00
SERGE	FONCTIONNAIRE	800.00

(11 rows)

12. Dans l'ordre décroissant des numéros des départements ensuite dans l'ordre croissant des salaires, lister les noms, les fonctions, les salaires des employés et les numéros des départements.

```
ma_bdd=# SELECT nomemp, emploi, sal, nodept
ma_bdd=# FROM EMP
ma_bdd=# ORDER BY nodept DESC, sal ASC;
```

nomemp	emploi	sal	nodept
LYNA	FONCTIONNAIRE	950.00	30
MARTIN	VENDEUR	1250.00	30
NASSIMA	VENDEUR	1250.00	30
ROI	VENDEUR	1500.00	30
BRAHIM	VENDEUR	1600.00	30
BENJAMIN	GESTIONNAIRE	2850.00	30
SERGE	FONCTIONNAIRE	800.00	20
VIRGINIE	FONCTIONNAIRE	1100.00	20
LUCIE	GESTIONNAIRE	2975.00	20
ARIJ	ANALYSTE	3000.00	20
ASMA	ANALYSTE	3000.00	20
SIMONE	FONCTIONNAIRE	1300.00	10
DAYANE	GESTIONNAIRE	2450.00	10
MAYAR	PRESIDENT	5000.00	10

(14 rows)

13. Dans l'ordre décroissant des noms des départements ensuite dans l'ordre croissant des salaires, lister les noms, les fonctions, les salaires des employés et les noms des départements.

```
ma_bdd=# SELECT EMP.nomemp, EMP.emploi as fonction, sal, DEPT.nomdept
ma_bdd=# FROM EMP, DEPT
ma_bdd=# WHERE EMP.nodept = DEPT.nodept
ma_bdd=# ORDER BY DEPT.nomdept DESC, sal ASC;
```

nomemp	fonction	sal	nomdept
LYNA	FONCTIONNAIRE	950.00	VENTES
MARTIN	VENDEUR	1250.00	VENTES
NASSIMA	VENDEUR	1250.00	VENTES
ROI	VENDEUR	1500.00	VENTES
BRAHIM	VENDEUR	1600.00	VENTES
BENJAMIN	GESTIONNAIRE	2850.00	VENTES
SERGE	FONCTIONNAIRE	800.00	RECHERCHE
VIRGINIE	FONCTIONNAIRE	1100.00	RECHERCHE
LUCIE	GESTIONNAIRE	2975.00	RECHERCHE
ARIJ	ANALYSTE	3000.00	RECHERCHE
ASMA	ANALYSTE	3000.00	RECHERCHE
SIMONE	FONCTIONNAIRE	1300.00	COMPTABILITE
DAYANE	GESTIONNAIRE	2450.00	COMPTABILITE
MAYAR	PRESIDENT	5000.00	COMPTABILITE

(14 rows)

Exercice 3 : Création et mise à jour

1. On souhaite revaloriser le salaire de tous les employés de 200e chacun. Proposer l'instruction permettant de le faire sur la table EMP.

```

ma_bdd=# UPDATE EMP SET sal = sal + 200;
UPDATE 14
ma_bdd=# SELECT * FROM EMP;

```

noemp	nomemp	emploi	mgr	dateemb	sal	comm	nodept
7369	SERGE	FONCTIONNAIRE	7902	1980-12-17	1000.00		20
7499	BRAHIM	VENDEUR	7698	1981-02-20	1800.00	300.00	30
7521	NASSIMA	VENDEUR	7698	1981-02-22	1450.00	500.00	30
7566	LUCIE	GESTIONNAIRE	7839	1981-04-02	3175.00		20
7654	MARTIN	VENDEUR	7698	1981-09-28	1450.00	1400.00	30
7698	BENJAMIN	GESTIONNAIRE	7839	1981-05-01	3050.00		30
7782	DAYANE	GESTIONNAIRE	7839	1981-06-09	2650.00		10
7788	ARIJ	ANALYSTE	7566	1982-12-09	3200.00		20
7839	MAYAR	PRESIDENT		1981-11-17	5200.00		10
7844	ROI	VENDEUR	7698	1981-09-08	1700.00	0.00	30
7876	VIRGINIE	FONCTIONNAIRE	7788	1983-01-12	1300.00		20
7900	LYNA	FONCTIONNAIRE	7698	1981-12-03	1150.00		30
7902	ASMA	ANALYSTE	7566	1981-12-03	3200.00		20
7934	SIMONE	FONCTIONNAIRE	7782	1982-01-23	1500.00		10

(14 rows)

2. Insérer un nouveau tuple dans la table DEPT. Le contenu du tuple est laissé à votre choix.

```

ma_bdd=# INSERT INTO DEPT /*NoDEPT, NomDEPT, VILLE */
ma_bdd=# VALUES (50, 'INFORMATIQUE', 'PARIS');
INSERT 0 1
ma_bdd=# SELECT * FROM DEPT;

```

nodept	nomdept	loc
10	COMPTABILITE	BREST
20	RECHERCHE	RENNES
30	VENTES	DINAR
40	GESTION	DINAN
50	INFORMATIQUE	PARIS

(5 rows)

3. Écrire et exécuter l'instruction de création de la table COMMISSION dont les attributs sont :

- (a) NomEmp de type chaîne de caractères (VARCHAR(10))
- (b) Emploi de type chaîne de caractères (VARCHAR(10))
- (c) Salaire de type réel (number(7,2))
- (d) ValComm de type réel (Number(4,2))

```

ma_bdd=# CREATE TABLE COMMISSION (
ma_bdd(#  NomEMP VARCHAR(10),
ma_bdd(#  Emploi VARCHAR(10),
ma_bdd(#  Salaire NUMERIC(7,2),
ma_bdd(#  ValComm NUMERIC(4,2)
ma_bdd(#  );
CREATE TABLE
ma_bdd=# \d COMMISSION

```

Table "public.commission"				
Column	Type	Collation	Nullable	Default
nomemp	character varying(10)			
emploi	character varying(10)			
salaire	numeric(7,2)			
valcomm	numeric(4,2)			

4. Insérer plusieurs tuples dans cette table.

```
ma_bdd=# INSERT INTO COMMISSION (nomemp, emploi, salaire, valcomm)
ma_bdd=# SELECT nomemp, emploi, sal, comm/1000 FROM EMP
ma_bdd=# WHERE emploi='VENDEUR';
INSERT 0 4
ma_bdd=# SELECT * FROM COMMISSION;
 nomemp | emploi | salaire | valcomm
-----+-----+-----+-----
 BRAHIM | VENDEUR | 1800.00 |    0.30
 NASSIMA | VENDEUR | 1450.00 |    0.50
  MARTIN | VENDEUR | 1450.00 |    1.40
   ROI   | VENDEUR | 1700.00 |    0.00
(4 rows)
```

5. Supprimer de la table COMMISSION, le tuple correspondant à un nom d'employé (au choix).

```
ma_bdd=# DELETE FROM COMMISSION WHERE valcomm = 0;
DELETE 1
ma_bdd=# SELECT * FROM COMMISSION;
 nomemp | emploi | salaire | valcomm
-----+-----+-----+-----
 BRAHIM | VENDEUR | 1800.00 |    0.30
 NASSIMA | VENDEUR | 1450.00 |    0.50
  MARTIN | VENDEUR | 1450.00 |    1.40
(3 rows)
```

6. Insérer dans cette table les données de la table EMP dont les employés ont une commission.

```
ma_bdd=# INSERT INTO COMMISSION (nomemp, emploi, salaire, valcomm)
ma_bdd=# SELECT nomemp, emploi, sal, comm/1000 FROM EMP
ma_bdd=# WHERE comm IS NOT NULL;
INSERT 0 4
ma_bdd=# SELECT * FROM COMMISSION;
 nomemp | emploi | salaire | valcomm
-----+-----+-----+-----
 BRAHIM | VENDEUR | 1800.00 |    0.30
 NASSIMA | VENDEUR | 1450.00 |    0.50
  MARTIN | VENDEUR | 1450.00 |    1.40
   ROI   | VENDEUR | 1700.00 |    0.00
(4 rows)
```

7. Vider la table COMMISSION ensuite supprimer là.

```
ma_bdd=# TRUNCATE TABLE COMMISSION;
TRUNCATE TABLE
ma_bdd=# SELECT * FROM COMMISSION;
 nomemp | emploi | salaire | valcomm
-----+-----+-----+-----
(0 rows)

ma_bdd=# DROP TABLE COMMISSION;
DROP TABLE
ma_bdd=# SELECT * FROM COMMISSION;
ERROR:  relation "commission" does not exist
LINE 1: SELECT * FROM COMMISSION;
                        ^
```

8. Recréer la table COMMISSION en y chargeant directement les données de la table EMP dont les employés ont une commission.

```

ma_bdd=# CREATE TABLE COMMISSION
ma_bdd=# AS SELECT nomemp, emploi, sal, comm
ma_bdd=# FROM EMP
ma_bdd=# WHERE comm IS NOT NULL;
SELECT 4
ma_bdd=# SELECT * FROM COMMISSION;
 nomemp | emploi |  sal  |  comm
-----+-----+-----+-----
 BRAHIM | VENDEUR | 1800.00 | 300.00
 NASSIMA | VENDEUR | 1450.00 | 500.00
 MARTIN  | VENDEUR | 1450.00 | 1400.00
 ROI     | VENDEUR | 1700.00 | 0.00
(4 rows)

```

Exercice 4 : Fonctions d'agrégat, ANY, ALL, GROUP BY, HAVING, etc.

1. Lister le nombre d'employés gagnant plus que le minimum de tous les salaires.

```

ma_bdd=# SELECT COUNT(noemp)
ma_bdd=# FROM EMP
ma_bdd=# WHERE sal > (SELECT MIN(sal) FROM EMP);
 count
-----
      13
(1 row)

```

2. Afficher le nombre de départements différents dans la table EMP.

```

ma_bdd=# SELECT COUNT(DISTINCT nodept) FROM EMP;
 count
-----
      3
(1 row)

```

3. Afficher le salaire moyen dans la table EMP

```

ma_bdd=# SELECT ROUND(AVG(sal), 2) FROM EMP;
 round
-----
 2273.21
(1 row)

```

4. Lister tout employé dont le salaire est supérieur à au moins un salaire du département 30.

```

ma_bdd=# SELECT *
ma_bdd=# FROM EMP
ma_bdd=# WHERE sal > (SELECT MIN(sal) FROM EMP WHERE nodept=30);
 noemp | nomemp |  emploi  | mgr | dateemb |  sal  |  comm  | nodept
-----+-----+-----+----+-----+-----+-----+-----
 7499 | BRAHIM | VENDEUR  | 7698 | 1981-02-20 | 1800.00 | 300.00 | 30
 7521 | NASSIMA | VENDEUR  | 7698 | 1981-02-22 | 1450.00 | 500.00 | 30
 7566 | LUCIE  | GESTIONNAIRE | 7839 | 1981-04-02 | 3175.00 |      | 20
 7654 | MARTIN | VENDEUR  | 7698 | 1981-09-28 | 1450.00 | 1400.00 | 30
 7698 | BENJAMIN | GESTIONNAIRE | 7839 | 1981-05-01 | 3050.00 |      | 30
 7782 | DAYANE | GESTIONNAIRE | 7839 | 1981-06-09 | 2650.00 |      | 10
 7788 | ARIJ   | ANALYSTE  | 7566 | 1982-12-09 | 3200.00 |      | 20
 7839 | MAYAR  | PRESIDENT |      | 1981-11-17 | 5200.00 |      | 10
 7844 | ROI    | VENDEUR  | 7698 | 1981-09-08 | 1700.00 | 0.00   | 30
 7876 | VIRGINIE | FONCTIONNAIRE | 7788 | 1983-01-12 | 1300.00 |      | 20
 7902 | ASMA   | ANALYSTE  | 7566 | 1981-12-03 | 3200.00 |      | 20
 7934 | SIMONE | FONCTIONNAIRE | 7782 | 1982-01-23 | 1500.00 |      | 10
(12 rows)

```

5. Lister tout employé du département 10 dont le salaire est supérieur à tous ceux du département 20.

```
ma_bdd=# SELECT * FROM EMP
ma_bdd=# WHERE nodept = 10 AND sal > (SELECT MAX(sal) FROM EMP WHERE nodept = 20);
```

noemp	nomemp	emploi	mgr	dateemb	sal	comm	nodept
7839	MAYAR	PRESIDENT		1981-11-17	5200.00		10

(1 row)

6. Lister les salaires minimal et maximal pour chaque département.

```
ma_bdd=# SELECT nodept, MIN(sal), MAX(sal) FROM EMP
ma_bdd=# GROUP BY nodept;
```

nodept	min	max
10	1500.00	5200.00
30	1150.00	3050.00
20	1000.00	3200.00

(3 rows)

7. Lister le salaire moyen par emploi (EMPLOI).

```
ma_bdd=# SELECT emploi, ROUND(AVG(sal),2) as sal_moyen FROM EMP
ma_bdd=# GROUP BY emploi;
```

emploi	sal_moyen
VENDEUR	1600.00
GESTIONNAIRE	2958.33
PRESIDENT	5200.00
ANALYSTE	3200.00
FONCTIONNAIRE	1237.50

(5 rows)

8. Lister les salaires minimal et maximal pour chaque département ayant au moins 2 employés.

```
ma_bdd=# SELECT nodept, MIN(sal), MAX(sal) FROM EMP
ma_bdd=# GROUP BY nodept
ma_bdd=# HAVING COUNT(nodept) >= 2;
```

nodept	min	max
10	1500.00	5200.00
30	1150.00	3050.00
20	1000.00	3200.00

(3 rows)

9. Lister les départements ayant le minimum des salaires supérieur à la moyenne des salaires des employés de bureau (FONCTIONNAIRE).

```
ma_bdd=# SELECT nodept FROM EMP
ma_bdd=# GROUP BY nodept
ma_bdd=# HAVING MIN(sal) > (SELECT AVG(sal) FROM EMP WHERE emploi = 'FONCTIONNAIRE');
```

nodept
10

(1 row)

10. Donner les noms des employés par ordre alphabétique.

```

ma_bdd=# SELECT nomemp FROM EMP
ma_bdd=# ORDER BY nomemp ASC;
      nomemp
-----
ARIJ
ASMA
BENJAMIN
BRAHIM
DAYANE
LUCIE
LYNA
MARTIN
MAYAR
NASSIMA
ROI
SERGE
SIMONE
VIRGINIE
(14 rows)

```

11. Donner les noms des employés par ordre alphabétique inversé.

```

ma_bdd=# SELECT nomemp FROM EMP
ma_bdd=# ORDER BY nomemp DESC;
      nomemp
-----
VIRGINIE
SIMONE
SERGE
ROI
NASSIMA
MAYAR
MARTIN
LYNA
LUCIE
DAYANE
BRAHIM
BENJAMIN
ASMA
ARIJ
(14 rows)

```

12. Lister les noms des employés dont le nom de manager commence par un H ou un S ou un un M.

```

ma_bdd=# SELECT nomemp FROM EMP
ma_bdd=# WHERE (nomemp LIKE 'H%'
ma_bdd(# OR nomemp LIKE 'S%'
ma_bdd(# OR nomemp LIKE 'M%')
ma_bdd=# AND noemp IN (SELECT mgr FROM EMP);
      nomemp
-----
MAYAR
(1 row)

```

13. Lister les noms des employés dont le nom de manager se termine par un E ou un M ou un un A.

```

ma_bdd=# SELECT nomemp FROM EMP
ma_bdd=# WHERE (nomemp LIKE '%E'
ma_bdd(# OR nomemp LIKE '%M'
ma_bdd(# OR nomemp LIKE '%A')
ma_bdd=# AND noemp IN (SELECT mgr FROM EMP);
nomemp
-----
LUCIE
DAYANE
ASMA
(3 rows)

```

14. Donner les informations sur les employés par ordre décroissant de date d'embauche puis par ordre alphabétique de nom.

```

ma_bdd=# SELECT * FROM EMP
ma_bdd=# ORDER BY dateEmb DESC, nomemp ASC;
noemp | nomemp | emploi | mgr | dateemb | sal | comm | nodept
-----+-----+-----+----+-----+----+-----+-----
7876 | VIRGINIE | FONCTIONNAIRE | 7788 | 1983-01-12 | 1300.00 |  | 20
7788 | ARIJ | ANALYSTE | 7566 | 1982-12-09 | 3200.00 |  | 20
7934 | SIMONE | FONCTIONNAIRE | 7782 | 1982-01-23 | 1500.00 |  | 10
7902 | ASMA | ANALYSTE | 7566 | 1981-12-03 | 3200.00 |  | 20
7900 | LYNNA | FONCTIONNAIRE | 7698 | 1981-12-03 | 1150.00 |  | 30
7839 | MAYAR | PRESIDENT |  | 1981-11-17 | 5200.00 |  | 10
7654 | MARTIN | VENDEUR | 7698 | 1981-09-28 | 1450.00 | 1400.00 | 30
7844 | ROI | VENDEUR | 7698 | 1981-09-08 | 1700.00 | 0.00 | 30
7782 | DAYANE | GESTIONNAIRE | 7839 | 1981-06-09 | 2650.00 |  | 10
7698 | BENJAMIN | GESTIONNAIRE | 7839 | 1981-05-01 | 3050.00 |  | 30
7566 | LUCIE | GESTIONNAIRE | 7839 | 1981-04-02 | 3175.00 |  | 20
7521 | NASSIMA | VENDEUR | 7698 | 1981-02-22 | 1450.00 | 500.00 | 30
7499 | BRAHIM | VENDEUR | 7698 | 1981-02-20 | 1800.00 | 300.00 | 30
7369 | SERGE | FONCTIONNAIRE | 7902 | 1980-12-17 | 1000.00 |  | 20
(14 rows)

```

15. Présenter les employés par groupes de même valeur de salaire, et par ordre décroissant de ces valeurs.

```

ma_bdd=# SELECT sal, count(*) FROM EMP
ma_bdd=# GROUP BY sal
ma_bdd=# ORDER BY sal DESC;
sal | count
-----+-----
5200.00 | 1
3200.00 | 2
3175.00 | 1
3050.00 | 1
2650.00 | 1
1800.00 | 1
1700.00 | 1
1500.00 | 1
1450.00 | 2
1300.00 | 1
1150.00 | 1
1000.00 | 1
(12 rows)

```

16. Représenter les employés groupés selon les noms de département ordonnés dans l'ordre alphabétique inverse.

```

ma_bdd=# SELECT nomdept, count(EMP.nomemp) FROM DEPT, EMP
ma_bdd=# WHERE DEPT.nodept = EMP.nodept
ma_bdd=# GROUP BY nomdept
ma_bdd=# ORDER BY nomdept DESC;
nomdept | count
-----+-----
VENTES | 6
RECHERCHE | 5
COMPTABILITE | 3
(3 rows)

```

17. Représenter les employés groupés selon le mois et l'année de leur date d'embauche (ordre chronologique).

```
ma_bdd=# SELECT DATE_TRUNC('month', dateemb)::date AS date_arrivée, count(*) AS nb_employés FROM EMP
ma_bdd=# GROUP BY DATE_TRUNC('month', dateemb)
ma_bdd=# ORDER BY DATE_TRUNC('month', dateemb) ASC;
 date_arrivée | nb_employés 
-----+-----
1980-12-01    |          1
1981-02-01    |          2
1981-04-01    |          1
1981-05-01    |          1
1981-06-01    |          1
1981-09-01    |          2
1981-11-01    |          1
1981-12-01    |          2
1982-01-01    |          1
1982-12-01    |          1
1983-01-01    |          1
(11 rows)

ma_bdd=# SELECT to_char((DATE_TRUNC('month', dateemb)), 'MON-YYYY') AS date_arrivée, count(*) AS nb_employés FROM EMP
ma_bdd=# GROUP BY DATE_TRUNC('month', dateemb)
ma_bdd=# ORDER BY DATE_TRUNC('month', dateemb) ASC;
 date_arrivée | nb_employés 
-----+-----
DEC-1980     |          1
FEB-1981     |          2
APR-1981     |          1
MAY-1981     |          1
JUN-1981     |          1
SEP-1981     |          2
NOV-1981     |          1
DEC-1981     |          2
JAN-1982     |          1
DEC-1982     |          1
JAN-1983     |          1
(11 rows)
```

18. Afficher le salaire moyen par job.

```
ma_bdd=# SELECT emploi, ROUND(AVG(sal),2) FROM EMP
ma_bdd=# GROUP BY emploi;
   emploi   | round 
-----+-----
VENDEUR     | 1600.00
GESTIONNAIRE | 2958.33
PRESIDENT   | 5200.00
ANALYSTE     | 3200.00
FONCTIONNAIRE | 1237.50
(5 rows)
```

19. Afficher les noms des employés groupés par job puis par nom de département.

```
ma_bdd=# SELECT emploi, nomdept, nomemp FROM EMP, DEPT
ma_bdd=# WHERE DEPT.nodept = EMP.nodept
ma_bdd=# GROUP BY EMP.emploi, DEPT.nomdept, EMP.nomemp
ma_bdd=# ORDER BY emploi, nomdept, nomemp;
   emploi   | nomdept | nomemp 
-----+-----+-----
ANALYSTE     | RECHERCHE | ARIJ
ANALYSTE     | RECHERCHE | ASMA
FONCTIONNAIRE | COMPTABILITE | SIMONE
FONCTIONNAIRE | RECHERCHE | SERGE
FONCTIONNAIRE | RECHERCHE | VIRGINIE
FONCTIONNAIRE | VENTES    | LYNA
GESTIONNAIRE | COMPTABILITE | DAYANE
GESTIONNAIRE | RECHERCHE | LUCIE
GESTIONNAIRE | VENTES    | BENJAMIN
PRESIDENT    | COMPTABILITE | MAYAR
VENDEUR      | VENTES    | BRAHIM
VENDEUR      | VENTES    | MARTIN
VENDEUR      | VENTES    | NASSIMA
VENDEUR      | VENTES    | ROI
(14 rows)
```

20. Afficher les nombres d'employés dirigés directement par chaque manager (ici on comprend manager comme un employé ayant son numéro dans la colonne mgr).

```

ma_bdd=# SELECT COUNT(*) FROM (SELECT nomemp FROM EMP
ma_bdd=# EXCEPT SELECT nomemp FROM EMP
ma_bdd=# WHERE noemp IN (SELECT mgr FROM EMP)) as exception;
count
-----
      8
(1 row)

ma_bdd=# SELECT nomemp FROM EMP
ma_bdd=# EXCEPT SELECT nomemp FROM EMP
ma_bdd=# WHERE noemp IN (SELECT mgr FROM EMP);
nomemp
-----
MARTIN
SIMONE
NASSIMA
ROI
SERGE
BRAHIM
VIRGINIE
LYNA
(8 rows)

```

21. Afficher la ville et le nom des départements qui ont plus de quatre employés.

```

ma_bdd=# SELECT loc, nomdept, nomemp FROM DEPT, EMP
ma_bdd=# WHERE EMP.nodept = DEPT.nodept
ma_bdd=# AND EMP.nodept in (SELECT nodept FROM EMP
ma_bdd=# GROUP BY nodept
ma_bdd=# HAVING Count(nodept) > 4)
ma_bdd=# ORDER BY 1;
 loc | nomdept | nomemp
-----+-----+-----
DINAR | VENTES | NASSIMA
DINAR | VENTES | ROI
DINAR | VENTES | BRAHIM
DINAR | VENTES | LYNA
DINAR | VENTES | MARTIN
DINAR | VENTES | BENJAMIN
RENNES | RECHERCHE | ASMA
RENNES | RECHERCHE | LUCIE
RENNES | RECHERCHE | ARIJ
RENNES | RECHERCHE | VIRGINIE
RENNES | RECHERCHE | SERGE
(11 rows)

ma_bdd=# SELECT loc, nomdept, count(*) FROM DEPT, EMP
ma_bdd=# WHERE EMP.nodept = DEPT.nodept
ma_bdd=# AND EMP.nodept in (SELECT nodept FROM EMP
ma_bdd=# GROUP BY nodept
ma_bdd=# HAVING Count(nodept) > 4)
ma_bdd=# GROUP BY 1, 2
ma_bdd=# ORDER BY 1;
 loc | nomdept | count
-----+-----+-----
DINAR | VENTES |      6
RENNES | RECHERCHE |      5
(2 rows)

```

22. Afficher le nom des départements où les employés gagnent en moyenne plus de 2500e.

```

ma_bdd=# SELECT nomdept FROM DEPT, EMP
ma_bdd=# WHERE DEPT.nodept = EMP.nodept
ma_bdd=# GROUP BY nomdept
ma_bdd=# HAVING AVG(sal) > 2500;
          nomdept
-----
COMPTABILITE
(1 row)

```

23. Afficher le nombre d'employés par département et par profession.

```

ma_bdd=# SELECT EMP.nodept, nomdept, emploi, Count(*) as nb_employés FROM EMP, DEPT
ma_bdd=# WHERE DEPT.nodept = EMP.nodept
ma_bdd=# GROUP BY EMP.nodept, nomdept, emploi
ma_bdd=# ORDER BY 1 ASC, 2 ASC, 3 ASC;
 nodept |  nomdept |   emploi   | nb_employés
-----+-----+-----+-----
      10 | COMPTABILITE | FONCTIONNAIRE |          1
      10 | COMPTABILITE | GESTIONNAIRE |          1
      10 | COMPTABILITE | PRESIDENT     |          1
      20 | RECHERCHE   | ANALYSTE      |          2
      20 | RECHERCHE   | FONCTIONNAIRE |          2
      20 | RECHERCHE   | GESTIONNAIRE |          1
      30 | VENTES      | FONCTIONNAIRE |          1
      30 | VENTES      | GESTIONNAIRE |          1
      30 | VENTES      | VENDEUR       |          4
(9 rows)

```

24. Donner pour chaque département, son nom, le salaire minimum de ses employés, le salaire maximum, le salaire moyen, le nombre d'employés, le nombre d'employés touchant une commission (éventuellement nulle) et le nombre de jobs différents exercés dans ce département.

```

ma_bdd=# SELECT nomdept, min(sal) as sal_mini, max(sal) as sal_maxi, round(avg(sal),2) as sal_moyen, count(nomemp) as nb_employés, count(comm) as nb_commissionnés, count(DISTINCT emploi) as nb_emplois FROM EMP, DEPT
ma_bdd=# WHERE DEPT.nodept = EMP.nodept
ma_bdd=# GROUP BY nomdept;
 nomdept | sal_mini | sal_maxi | sal_moyen | nb_employés | nb_commissionnés | nb_emplois
-----+-----+-----+-----+-----+-----+-----
COMPTABILITE | 1500.00 | 5200.00 | 3116.67 |          3 |          0 |          3
RECHERCHE   | 1000.00 | 3200.00 | 2375.00 |          5 |          0 |          3
VENTES      | 1150.00 | 3050.00 | 1766.67 |          6 |          4 |          3
(3 rows)

```

Exercice 5 : Autres aspects de SQL

1. Créer une vue EMPDEPT qui contient la jointure naturelle des deux tables EMP et DEPT.

```

ma_bdd=# CREATE VIEW EMPDEPT
ma_bdd=# AS SELECT noemp, nomemp, emploi, mgr, dateemb, sal, comm, EMP.nodept, nomdept, loc
ma_bdd=# FROM EMP, DEPT
ma_bdd=# WHERE EMP.nodept = DEPT.nodept;
CREATE VIEW
ma_bdd=# SELECT * FROM EMPDEPT;

```

noemp	nomemp	emploi	mgr	dateemb	sal	comm	nodept	nodept	nomdept	loc
7369	SERGE	FONCTIONNAIRE	7902	1980-12-17	1000.00		20	20	RECHERCHE	RENNES
7499	BRAHIM	VENDEUR	7698	1981-02-20	1800.00	300.00	30	30	VENTES	DINAR
7521	NASSIMA	VENDEUR	7698	1981-02-22	1450.00	500.00	30	30	VENTES	DINAR
7566	LUCIE	GESTIONNAIRE	7839	1981-04-02	3175.00		20	20	RECHERCHE	RENNES
7654	MARTIN	VENDEUR	7698	1981-09-28	1450.00	1400.00	30	30	VENTES	DINAR
7698	BENJAMIN	GESTIONNAIRE	7839	1981-05-01	3050.00		30	30	VENTES	DINAR
7782	DAYANE	GESTIONNAIRE	7839	1981-06-09	2650.00		10	10	COMPTABILITE	BREST
7788	ARIJ	ANALYSTE	7566	1982-12-09	3200.00		20	20	RECHERCHE	RENNES
7839	MAYAR	PRESIDENT		1981-11-17	5200.00		10	10	COMPTABILITE	BREST
7844	ROI	VENDEUR	7698	1981-09-08	1700.00	0.00	30	30	VENTES	DINAR
7876	VIRGINIE	FONCTIONNAIRE	7788	1983-01-12	1300.00		20	20	RECHERCHE	RENNES
7900	LYNA	FONCTIONNAIRE	7698	1981-12-03	1150.00		30	30	VENTES	DINAR
7902	ASMA	ANALYSTE	7566	1981-12-03	3200.00		20	20	RECHERCHE	RENNES
7934	SIMONE	FONCTIONNAIRE	7782	1982-01-23	1500.00		10	10	COMPTABILITE	BREST

(14 rows)

2. En utilisant cette vue, afficher les noms d'employés touchant plus que 2000 e avec leurs villes.

```

ma_bdd=# SELECT * FROM EMPDEPT;
ma_bdd=# SELECT nomemp, loc FROM EMPDEPT
ma_bdd=# WHERE sal > 2000;
   nomemp | loc
-----+-----
   LUCIE  | RENNES
 BENJAMIN | DINAR
  DAYANE  | BREST
   ARIJ   | RENNES
   MAYAR  | BREST
   ASMA   | RENNES
(6 rows)

```

3. Vider la table EMP.

```

ma_bdd=# TRUNCATE TABLE EMP;
TRUNCATE TABLE
ma_bdd=# SELECT * FROM EMP;
 noemp | nomemp | emploi | mgr | dateemb | sal | comm | nodept
-----+-----+-----+----+-----+----+-----+-----
(0 rows)

```

4. Supprimer la table EMP.

```

ma_bdd=# DROP TABLE EMP;
ERROR:  cannot drop table emp because other objects depend on it
DETAIL:  view empdept depends on table emp
HINT:   Use DROP ... CASCADE to drop the dependent objects too.
ma_bdd=# DROP TABLE EMP CASCADE;
NOTICE:  drop cascades to view empdept
DROP TABLE
ma_bdd=#

```

5. Créer à nouveau la table EMP avec les mêmes attributs et les mêmes contraintes d'intégrité.

```

ma_bdd=# CREATE TABLE EMP(
ma_bdd(# NoEMP NUMERIC(4) NOT NULL,
ma_bdd(# NomEMP VARCHAR(10),
ma_bdd(# EMPLOI VARCHAR(15),
ma_bdd(# MGR NUMERIC(4),
ma_bdd(# DateEMB DATE,
ma_bdd(# SAL NUMERIC(7,2),
ma_bdd(# COMM NUMERIC(7,2),
ma_bdd(# NoDEPT NUMERIC(2),
ma_bdd(# CONSTRAINT EMP_clé_primaire PRIMARY KEY (NoEMP),
ma_bdd(# CONSTRAINT EMP_clé_étrangère FOREIGN KEY (NoDEPT) REFERENCES DEPT(NoDEPT));
CREATE TABLE
ma_bdd=# \d emp

```

Column	Type	Collation	Nullable	Default
noemp	numeric(4,0)		not null	
nomemp	character varying(10)			
emploi	character varying(15)			
mgr	numeric(4,0)			
dateemb	date			
sal	numeric(7,2)			
comm	numeric(7,2)			
nodept	numeric(2,0)			

```

Indexes:
    "emp_clé_primaire" PRIMARY KEY, btree (noemp)
Foreign-key constraints:
    "emp_clé_étrangère" FOREIGN KEY (nodept) REFERENCES dept(nodept)

```

6. Supprimer la colonne COMM.

```

ma_bdd=# ALTER TABLE EMP DROP COLUMN COMM;
ALTER TABLE
ma_bdd=# \d emp

```

Column	Type	Collation	Nullable	Default
noemp	numeric(4,0)		not null	
nomemp	character varying(10)			
emploi	character varying(15)			
mgr	numeric(4,0)			
dateemb	date			
sal	numeric(7,2)			
nodept	numeric(2,0)			

```

Indexes:
    "emp_clé_primaire" PRIMARY KEY, btree (noemp)
Foreign-key constraints:
    "emp_clé_étrangère" FOREIGN KEY (nodept) REFERENCES dept(nodept)

```

7. Renommer la colonne COMM en BONUS.

```

ma_bdd=# ALTER TABLE EMP ADD COLUMN COMM NUMERIC(7,2);
ALTER TABLE
ma_bdd=# ALTER TABLE EMP RENAME COLUMN COMM TO BONUS;
ALTER TABLE
ma_bdd=# \d

```

Schema	Name	Type	Owner
public	commission	table	postgres
public	dept	table	postgres
public	emp	table	postgres

(3 rows)

```

ma_bdd=# \d emp

```

Column	Type	Collation	Nullable	Default
noemp	numeric(4,0)		not null	
nomemp	character varying(10)			
emploi	character varying(15)			
mgr	numeric(4,0)			
dateemb	date			
sal	numeric(7,2)			
nodept	numeric(2,0)			
bonus	numeric(7,2)			

```

Indexes:
    "emp_clé_primaire" PRIMARY KEY, btree (noemp)
Foreign-key constraints:
    "emp_clé_étrangère" FOREIGN KEY (nodept) REFERENCES dept(nodept)

```

8. Ajouter une colonne à la table DEPT.

```

ma_bdd=# ALTER TABLE DEPT ADD COLUMN BUDGET_in_k€ NUMERIC(11,2);
ALTER TABLE
ma_bdd=# \d dept

```

Column	Type	Collation	Nullable	Default
nodept	numeric(2,0)		not null	
nomdept	character varying(14)			
loc	character varying(13)			
budget_in_k€	numeric(11,2)			

```

Indexes:
    "dept_clé_primaire" PRIMARY KEY, btree (nodept)
Referenced by:
    TABLE "emp" CONSTRAINT "emp_clé_étrangère" FOREIGN KEY (nodept) REFERENCES dept(nodept)

```