```
CREATE TABLE Editeurs (
    Num_Editeur INT PRIMARY KEY,
    Nom VARCHAR(50) NOT NULL,
    Prenom VARCHAR(50) NOT NULL,
    Rue VARCHAR(50) NOT NULL,
    Localite VARCHAR(50) NOT NULL
);



CREATE TABLE Categorie (
    Num_Categorie INT PRIMARY KEY,
    Nom VARCHAR(50) NOT NULL,
    Categorie_parent INT REFERENCES Categorie(Num_Categorie)
);




CREATE TABLE Auteur (
    id_Auteur INT PRIMARY KEY,
    Nom VARCHAR(50) NOT NULL,
    Date_de_naissance DATE NOT NULL
);






CREATE TABLE Livres (
    NumLivre INT PRIMARY KEY,
    Titre VARCHAR(50) NOT NULL,
    ISBN INT NOT NULL UNIQUE,
    NbPages INT NOT NULL,
    Langue VARCHAR(50) NOT NULL,
    Cote INT NOT NULL,
    Date_de_publication DATE NOT NULL,
    Num_Editeurs INT REFERENCES Editeurs(Num_Editeur) NOT NULL,
    Num_Categorie INT REFERENCES Categorie(Num_Categorie) NOT NULL
);
```

```
CREATE TABLE Ecriture (
    id_Auteur INT REFERENCES Auteur(id_Auteur) NOT NULL,
    NumLivre INT REFERENCES Livres(NumLivre) NOT NULL,
    primary key(id_Auteur,  NumLivre)
);
```

DROP TABLE dans le sens inverse de la création

```
INSERT INTO Categorie(num_categorie, nom, categorie_parent)
    VALUES (4, 'SQL', 3);
```

```
INSERT INTO Editeurs (num_editeur, Nom, Prenom, Rue, Localite)
    VALUES (1, 'Gallimard', ' ', 'Rue Sebastien-Bottin 5', '75328 Paris Cedex 07');
```

```
INSERT INTO Auteur (id_auteur, nom, date_de_naissance)
   VALUES (1, 'J.K. Rowling', TO_DATE('310765', 'ddmmyy'));
```

```
SELECT *
FROM Auteur
```

Labo 2 :

1.
   ```
   ALTER TABLE TYPECONTRAT
   MODIFY ty_code NOT NULL ;
   ```

2.
   ```
   SELECT CA_NOM, PRENOM
   FROM CANDIDAT
   WHERE datenaissance LIKE '%93' ;
   ```

3.
   ```
   SELECT titre
   FROM JOB
   WHERE EXTRACT(year FROM publie) < 2019;
   ```

4.
   ```
    SELECT CA_NOM, PRENOM
   FROM candidat
   WHERE CA_CODEPOSTAL > 4300 ;
   ```

5.
   ```
    SELECT et_nom
   FROM etablissement
   WHERE et_nom LIKE 'Universit%' ;
   ```

6.
   ```
    SELECT titre
   FROM job
   WHERE description LIKE '%restaurant%' ;
   ```

```
7.   SELECT prenom
     FROM candidat
     WHERE prenom LIKE '%e' ;
```

```
8.   SELECT ca_nom, prenom
     FROM candidat
     WHERE email IS NULL ;
```

```
9.   SELECT ca_nom, prenom
     FROM candidat
     WHERE ca_numero < ca_codepostal ;
```

```
10.  SELECT titre
     FROM job
     WHERE MONTHS_BETWEEN(SYSDATE, publie) < 4;
```

```
11.  SELECT ca_nom, prenom
     FROM candidat
     WHERE ca_localite LIKE 'Battice' ;
```

```
12.  SELECT ca_nom, prenom
     FROM candidat
     WHERE ca_localite in('Herve', 'Charleroi') ;
```

```
13.  SELECT ca_nom, prenom, EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM
     datenaissance)
     FROM candidat ;
```

```
14.  SELECT DISTINCT ca_localite
     FROM candidat
     WHERE prenom LIKE 'M%' ;
```

```
15.  SELECT titre
     FROM job
     WHERE publie = LAST_DAY(publie)
```

```
16.  SELECT TO_CHAR(publie, 'DAY')
     FROM job ;
```

```
17.  SELECT ca_nom, prenom
     FROM candidat
     WHERE TO_CHAR(datenaissance, 'D') = 1;
```

```
18.  SELECT id_typecontrat, ty_libelle
     FROM typecontrat
     WHERE ty_code LIKE 'CD%';

19.  SELECT titre
     FROM job
     WHERE EXTRACT(DAY FROM publie) = 28 ;

20.  SELECT ca_nom
     FROM candidat
     WHERE EXTRACT(YEAR FROM datenaissance) BETWEEN 1990 AND 2000;

21.  SELECT ca_nom, prenom
     FROM candidat
     WHERE TO_CHAR(datenaissance, 'MM') = 12;

22.  SELECT ent_nom
     FROM entreprise
     ORDER BY ent_nom #On peut ajouter ASC mais il est implicite ici

23.  SELECT ty_libelle
     FROM typecontrat
     WHERE ty_code NOT LIKE 'CD_';

24.  UPDATE candidat
     SET ca_nom = REPLACE(ca_nom, 'Roussel', 'Rousselle')
     WHERE ca_nom LIKE 'Roussel';

25.  UPDATE qualification
     SET qu_libelle = UPPER(SUBSTR(qu_libelle, 1, 1) || SUBSTR(qu_libelle, 2))
                                                  #Les || veulent dire ‘CONCATENATION’
     WHERE SUBSTR(qu_libelle, 1, 1) <> UPPER(SUBSTR(qu_libelle, 1, 1))

26.  UPDATE qualification
     SET qu_libelle = UPPER(SUBSTR(qu_libelle, 1, 1)) || LOWER(SUBSTR(qu_libelle, 2))
     WHERE SUBSTR(qu_libelle, 2) <> LOWER(SUBSTR(qu_libelle, 2))

27.  ALTER TABLE entreprise
     ADD ent_codepostal INT;
     ADD ent_localite VARCHAR(50);

     UPDATE entreprise
     SET ent_codepostal = 1000
     WHERE id_entreprise = 13;

     UPDATE entreprise
     SET ent_localite = 'Bruxelles'
     WHERE id_entreprise = 13;
```

28.
```
UPDATE candidat
SET ca_rue = 'rue des chars', ca_numero = 48, ca_codepostal = 4602, ca_localite = 'Cheratte'
WHERE ca_nom = 'Leroy' AND prenom = 'Nathalie';
```

29.
```
DELETE FROM Nationalite
WHERE nat_libelle in('Albanie', 'Biélorussie', 'Saint-Marin', 'Liechtenstein');
```

Labo 3 :

1.
```
SELECT c.ca_nom, c.prenom, n.nat_libelle
FROM candidat c
JOIN nationalite n ON n.id_nationalite = c.id_nationalite;
```

2.
```
SELECT j.titre, e.ent_nom
FROM entreprise e
JOIN job j ON j.id_entreprise = e.id_entreprise
WHERE e.ent_nom IN ('IPES', 'Axa', 'RenoBati');
```

3.
```
SELECT DISTINCT c.ca_nom, c.prenom, p.datepostule
FROM candidat c
JOIN postuler p ON c.id_candidat = p.id_candidat
WHERE p.datepostule > TO_DATE('18052019', 'ddmmyyyy');
```

4.
```
SELECT 'Le candidat ' || c.prenom || c.ca_nom || ' a postulé le ' || p.datepostule || ' à ' ||
j.titre AS Proposition
FROM candidat c
JOIN postuler p ON c.id_candidat = p.id_candidat
JOIN job j ON j.id_job = p.id_job
WHERE p.datepostule > TO_DATE('18052019', 'ddmmyyyy');
```

5.
```
SELECT c.prenom, c.ca_nom
FROM candidat c
JOIN etatcivil ec ON c.id_ec = ec.id_etatcivil
WHERE c.id_ec = 1;
```

6.
```
SELECT j.titre, p.datepostule
FROM job j
LEFT OUTER JOIN postuler p ON j.id_job = p.id_job
WHERE p.datepostule IS NULL ;
```

```
7.  SELECT c.prenom, c.ca_nom, j.titre, p.datepostule
    FROM candidat c
    JOIN postuler p ON c.id_candidat = p.id_candidat
    JOIN job j ON j.id_job = p.id_job
    WHERE c.ca_nom IN('Bernard', 'Thomas') AND c.prenom IN('Marie', 'Pierre')
    ORDER BY p.datepostule;
```

```
8.  SELECT j.titre
    FROM job j
    LEFT OUTER JOIN REQUERIR r ON r.id_job = j.id_job
    WHERE r.id_job IS NULL;
```

```
9.  SELECT j.titre
    FROM job j
    JOIN necessiter n ON j.id_job = n.id_job
    JOIN qualification q ON n.id_qualification = q.id_qualification
    WHERE q.qu_libelle LIKE 'Master%';
```

```
10. SELECT j.titre
    FROM job j
    LEFT OUTER JOIN donner d ON j.id_job = d.id_job
    WHERE d.id_job IS NULL;
```

```
11. SELECT c1.ca_nom, c1.prenom, c2.ca_nom, c2.prenom
    FROM candidat c1
    JOIN postuler p1 ON c1.id_candidat = p1.id_candidat
    JOIN postuler p2 ON p1.id_job = p2.id_job
    JOIN candidat c2 ON p2.id_candidat = c2.id_candidat
    WHERE c1.id_candidat < c2.id_candidat ;
```

```
12.  SELECT c.ca_nom, j.titre, p.datepostule
    FROM candidat c
    JOIN postuler p ON p.id_candidat = c.id_candidat
    JOIN job j ON j.id_job = p.id_job;
```

```
13.  SELECT j.titre, av.av_libelle
    FROM job j
    JOIN donner d ON j.id_job = d.id_job
    JOIN avantage av ON d.id_avantage = av.id_avantage
    JOIN typecontrat tc ON j.id_typecontrat = tc.id_typecontrat
    WHERE av.av_libelle = 'Voiture' AND tc.ty_code = 'CDI';
```

```
14. SELECT c.ca_nom, c.prenom
    FROM conduire cond
    JOIN candidat c ON c.id_candidat = cond.id_candidat
    JOIN permisconduire p ON p.id_permis = cond.id_permis
    WHERE p.categorie = 'Voitures' AND cond.dateobtention < SYSDATE - 3650
    ORDER BY cond.dateobtention

15. SELECT c.ca_nom, c.prenom, q.qu_libelle
    FROM candidat c1
    JOIN postuler p1 ON c1.id_candidat = p1.id_candidat
    JOIN postuler p2 ON p1.id_job = p2.id_job
    JOIN candidat c2 ON p2.id_candidat = c2.id_candidat
    JOIN postuler p ON c.id_candidat = p.id_candidat
    JOIN posseder poss ON c.id_candidat = poss.id_candidat
    JOIN qualifcation q ON poss.id_qualification = q.id_qualification
    JOIN qualifcation q2 ON poss.id_qualification = q2.id_qualification
    WHERE c1.id_candidat < c2.id_candidat;

16. SELECT ent.ent_nom
    FROM candidat c
    JOIN postuler p ON p.id_candidat = c.id_candidat
    JOIN job j ON j.id_job = p.id_job
    JOIN entreprise ent ON ent.id_entreprise = j.id_entreprise
    WHERE c.ca_nom = 'Rousselle' AND c.prenom = 'Martine' ;

17.  SELECT j.titre, TO_NUMBER(SUBTR(tr.tr_libelle, INSTR(tr.tr_libelle, '-'), +1)) AS Salaire
    FROM job j
    JOIN tranchesalariale tr ON tr.id_tranchesalariale = j.id_tranchesalariale
    ORDER BY TO_NUMBER(SUBTR(tr.tr_libelle, INSTR(tr.tr_libelle, '-'), +1)) DESC

18.  SELECT j.titre
    FROM job j
    JOIN necessiter n ON j.id_job = n.id_job
    JOIN qualification q ON n.id_qualification = q.id_qualification
    WHERE q.qu_libelle = 'Marketing';

19.

20.

21.

22.
```