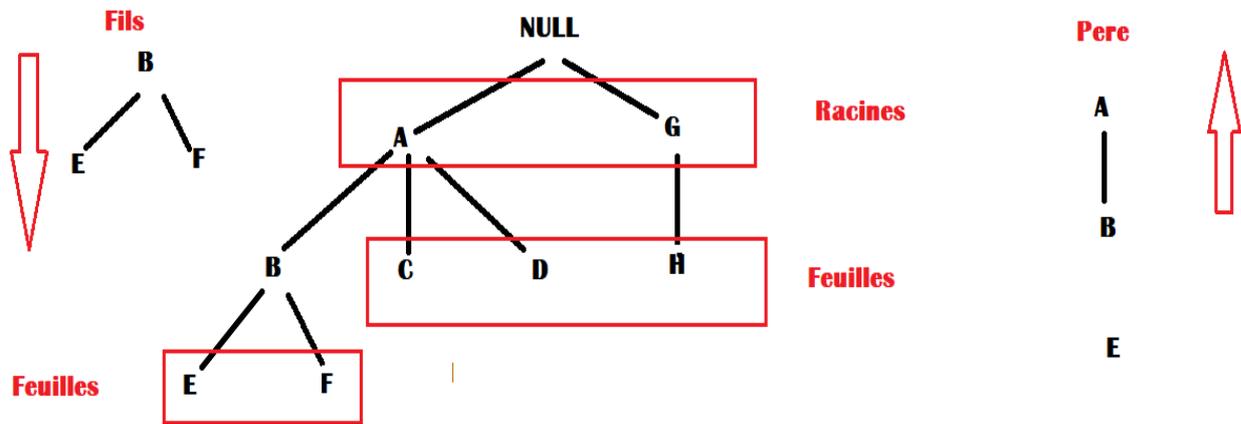


# Représentation et manipulation des Arbres en BD

## Un exemple



## Représentation relationnelle

Arbre(Fils, Père#, ...)

Attributs minimaux Fils/Père

Fils Clé primaire et Père Clé étrangère.

### La table Arbre

Arbre	<u>Fils</u>	Père#
	A	NULL
	B	A
	C	A
	D	A
	E	B
	F	B
	G	NULL
	H	G

## Création de la table Arbre

```
CREATE TABLE Arbre
( Fils Varchar2(2),
  Pere Varchar2(2),
  Constraint Pk_Arbre PRIMARY KEY (Fils),
  Constraint Fk_Arbre FOREIGN KEY (Pere) REFERENCES Arbre(Fils));
```

## Remplissage de la table Arbre

```
INSERT INTO Arbre VALUES ('A', NULL);
INSERT INTO Arbre VALUES ('B', 'A');
INSERT INTO Arbre VALUES ('C', 'A');
INSERT INTO Arbre VALUES ('D', 'A');
INSERT INTO Arbre VALUES ('E', 'B');
INSERT INTO Arbre VALUES ('F', 'B');
INSERT INTO Arbre VALUES ('G', NULL);
INSERT INTO Arbre VALUES ('H', 'G');
```

## Les cinq principales requêtes sur les arbres

### Nouvelles clauses SQL

**Level** : Un attribut système de type entier donne les numéros des niveaux

**LPAD()** : Fonction permettant une indentée de l'arbre

```
SELECT LPAD('azerty', 10, 'x'); -- retourne 'xxxxazerty'
```

### Parcours PERE-FILS

CONNECT BY **PRIOR** FILS = PERE

[AND Condition]

START WITH Condition : Point de départ du parcours

### Parcours FILS-PERE

CONNECT BY FILS = **PRIOR** PERE

[AND condition]

START WITH Condition : Point de départ du parcours

### Recherche des racines de l'Arbre

```
SELECT Fils
FROM Arbre
WHERE Pere IS NULL;
```

### Recherche des feuilles de l'Arbre

```
SELECT Fils
FROM Arbre
WHERE Fils NOT IN ( SELECT Pere FROM Arbre
                    WHERE Pere IS NOT NULL);
```

### Recherche des Fils (directs et indirects) d'un élément A (avec level)

```
SELECT Fils, Level
FROM Arbre
CONNECT BY PRIOR Fils = Pere
START WITH Fils = 'A';
```

### Recherche des Fils de A sauf B (visualisation indentée)

```
SELECT LPAD('-', 2*Level, ' ') || Fils
FROM Arbre
WHERE Fils <> 'B'
CONNECT BY PRIOR Fils = Pere
START WITH Fils = 'A';
```

### Recherche des Fils de A sauf B et les fils de B

```
SELECT Fils
FROM Arbre
CONNECT BY PRIOR Fils = Pere
AND Fils <> 'B'
START WITH Fils = 'A' ;
```