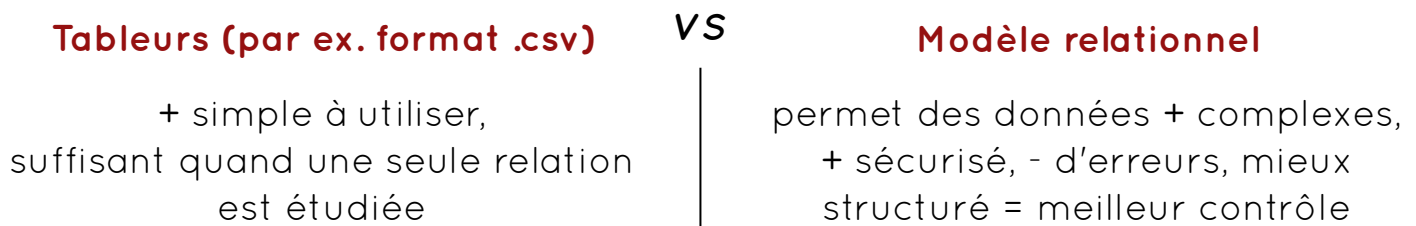
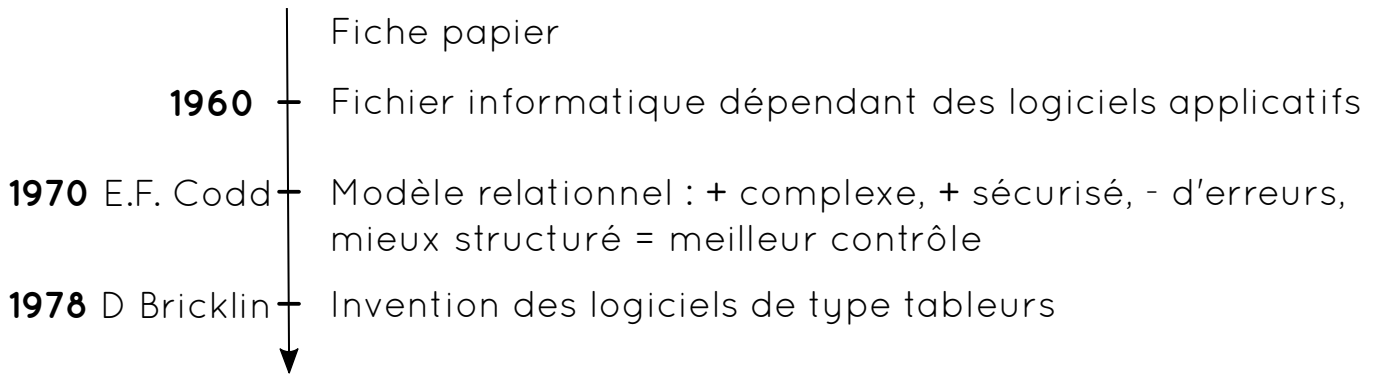
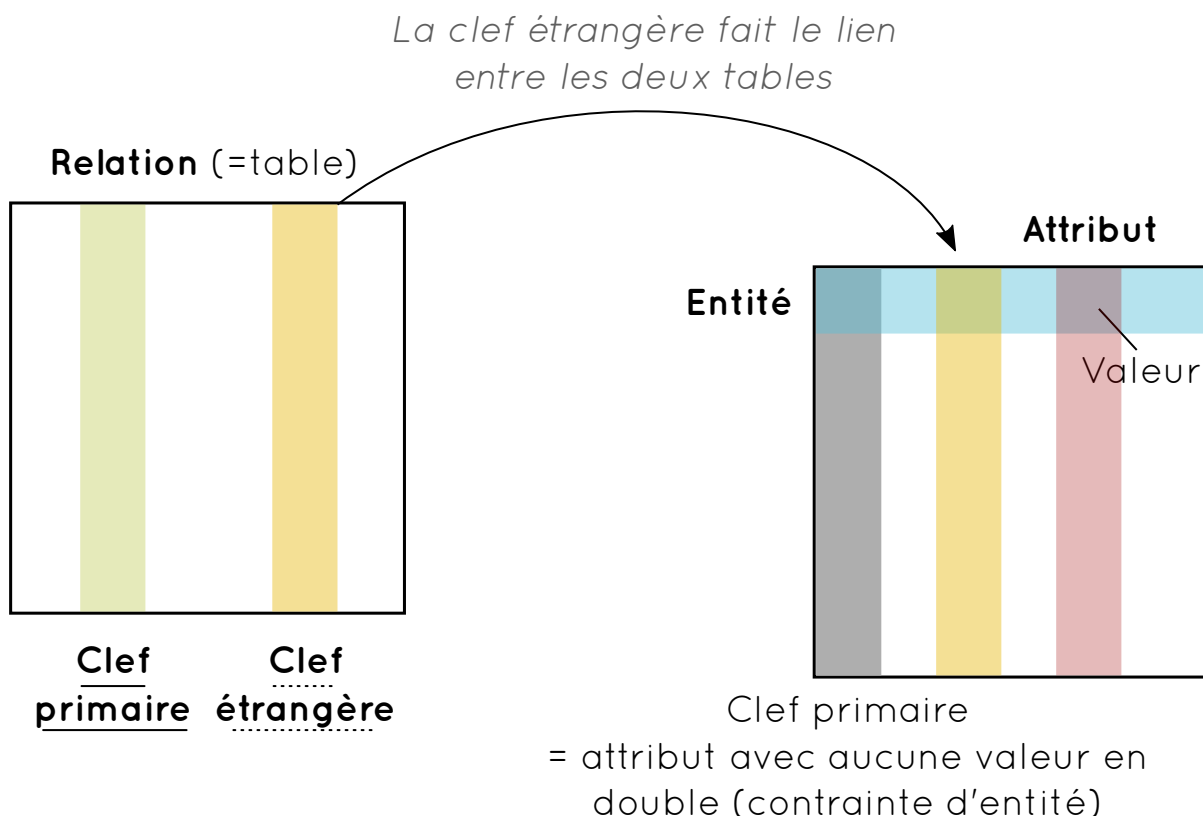


Bases de données relationnelles (1)



Contraintes d'intégrité (=règle de cohérence) :

- **Entité** : chaque enregistrement (chaque entité) est unique (=> clef primaire)
- **Référence** : les relations (tables) sont liés par des attributs (=> clef étrangère)
- **Domaine** : les attributs sont typés (INT, BOOL, CHAR ...)
- **Métier** : toutes autres contraintes (par ex. un mail doit contenir un @)



Bases de données relationnelles (2)

SCHEMA / RETOUR

Monstres = { ID Integer, Classe String, Niveau Integer, Race String }

SQL (SQLite)

```
CREATE TABLE Monstres (
  ID INT PRIMARY KEY,
  Classe VARCHAR(20),
  Niveau INT,
  Race VARCHAR(20)
);
```

```
INSERT INTO Monstres
VALUES (
  1, "Mage", 75, "Elf"
);
```

```
INSERT INTO Monstres
VALUES (
  2, "Voleur", 5, "Troll"
);
```

```
SELECT Classe, Niveau
FROM Monstres
WHERE Niveau>5;
```

```
CREATE TABLE Races (
  Race VARCHAR(20)
    PRIMARY KEY,
  Cris VARCHAR(40),
  Humanoïde BOOL,
  FOREIGN KEY (Type)
    REFERENCES Monstres (Type)
);
```

```
INSERT INTO Races
VALUES
  ("Elf", "Yo!", 1),
  ("Troll", "Grrr", 0)
;
```

```
SELECT * FROM Types;
```

```
SELECT * FROM Monstres
INNER JOIN Races
ON Monstres.Race = "Elf"
OR Monstres.Race = "Troll"
```

ÉTAT DE LA BASE DE DONNÉES

MONSTRES			
ID	Classe	Niveau	Race

MONSTRES			
ID	Classe	Niveau	Race
1	Mage	75	Elf

MONSTRES			
ID	Classe	Niveau	Race
1	Mage	75	Elf
2	Voleur	5	Troll

MONSTRES	
Classe	Niveau
Mage	75

Races = { Race String, Cris string, Humanoïde Bool }

RACES			MONSTRES			
Race	Cris	Humanoïde	ID	Classe	Niveau	Race
			1	Mage	75	Elf
			2	Voleur	5	Troll

RACES		
Race	Cris	Humanoïde
Elf	Yo!	1
Troll	Grrr!	0

RACES			MONSTRES			
Race	Cris	Humanoïde	ID	Classe	Niveau	Race
Elf	Yo!	1	1	Mage	75	Elf
Troll	Grrr!	0	2	Voleur	5	Troll

ID	Classe	Niveau	Type	Cris	Humanoïde
1	Mage	75	Elf	Yo!	1
2	Voleur	5	Troll	Grrr	1

Requêtes SQL à connaître (plus de détails dans les tableaux)

CREATE TABLE : créer une table

INSERT INTO ... VALUES : ajouter des entrées (=entités)

SELECT ... FROM : sélectionner les attributs ... de la table

WHERE : sélectionner des entités grâce à une condition booléenne

UPDATE : modifier une entrée

DELETE : supprimer une entrée

INNER JOIN : réaliser une jointure entre 2 tables