

Cours NSI	Thème : Base de données Fiche SQL	Date :
-----------	--------------------------------------	--------

### Typage des données

Voir table ci-contre

### Création d'une table avec clé primaire

```
CREATE TABLE nom_table (
    colonne_1 INTEGER PRIMARY KEY,
    colonne_2 text NOT NULL
);
```

```
CREATE TABLE nom_table (
    colonne_1 INTEGER PRIMARY KEY,
    colonne_2 TEXT NOT NULL,
    colonne_3 INTEGER NOT NULL
);
```

```
CREATE TABLE nom_table (
    colonne_1 INTEGER NOT NULL,
    colonne_2 INTEGER NOT NULL,
    ...
    PRIMARY KEY(colonne_1, colonne_2,...)
);
```

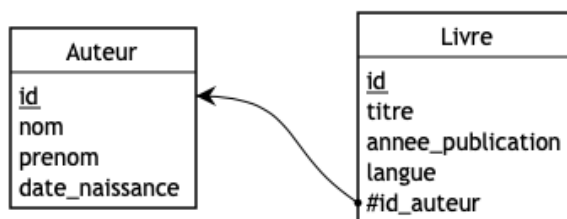
### Création d'une table avec clés étrangères

```
CREATE TABLE LIVRE (
    id INTEGER PRIMARY KEY,
    titre TEXT NOT NULL,
    annee_parution INTEGER NOT NULL,
    langue TEXT NOT NULL,
    id_auteur INTEGER NOT NULL,
    FOREIGN KEY (id_auteur) REFERENCES AUTEUR (id)
);
```

### Suppression d'une table

```
DROP TABLE nom_table;
```

Types courants	Type sqlite3
INT INTEGER TINYINT SMALLINT MEDIUMINT BIGINT UNSIGNED BIGINT INT2 INT8	INTEGER
CHARACTER(20) VARCHAR(255) VARYING CHARACTER(255) NCHAR(55) NATIVE CHARACTER(70) NVARCHAR(100) TEXT CLOB	TEXT
BLOB <i>no datatype specified</i>	BLOB
REAL DOUBLE DOUBLE PRECISION FLOAT	REAL
NUMERIC DECIMAL(10, 5) BOOLEAN DATE DATETIME	NUMERIC



Cours NSI	Thème : Base de données Fiche SQL	Date :
-----------	--------------------------------------	--------

### **Sélection de données**

toutes les données d'une table :

```
SELECT * FROM nom_table;
```

certaines colonnes d'une table :

```
SELECT colonne_1, colonne_2 FROM nom_table;
```

sans doublons :

```
SELECT DISTINCT colonne FROM nom_table;
```

en ordonnant :

```
SELECT colonne_1, colonne_2 FROM nom_table ORDER BY colonne ASC;
```

```
SELECT colonne_1, colonne_2 FROM nom_table ORDER BY colonne DESC;
```

en filtrant :

```
SELECT colonne FROM nom_table WHERE condition;
```

*condition* est une expression utilisant :

- les opérateurs classiques : <, <=, >, >=, =, <>, +, -, \*, /, AND, OR, NOT, ...
- l'opérateur LIKE : colonne LIKE '%motif%', valeur contenant un motif particulier

### **Jointure**

sélection de toutes les données de deux tables en relation :

```
SELECT * FROM table_1 JOIN table_2 ON table_1.colonne_1 = table_2.colonne_2;
```

sélection de certaines données de deux tables en relation :

```
SELECT table_1.colonne FROM table_1 JOIN table_2 ON table_1.colonne_1 = table_2.colonne_2;
```

### **Fonctions d'agrégation**

```
SELECT FONCTION(colonne) FROM nom_table;
```

Les fonctions usuelles sont : COUNT(), SUM(), AVG(), MIN(), MAX(), ...

### **Insertion de données**

```
INSERT INTO nom_table (colonne_1, colonne_2) VALUES (valeur_1a, valeur_2a), (valeur_1b, valeur_2b);
```

### **Mise à jour de données**

```
UPDATE nom_table SET colonne_1 = valeur_1, colonne_2 = valeur_2 WHERE condition;
```

### **Suppression de données**

suppression d'une ligne :

```
DELETE FROM nom_table WHERE condition;
```

suppression de toutes les lignes d'une table :

```
DELETE * FROM nom_table;
```