



## Plan

---

- Introduction
- Le modèle relationnel
- L'algèbre relationnelle
- **Le langage SQL**
- Le MEA et le modèle relationnel

1



## Le langage SQL

---

2

## Généralités

- SQL (Structured Query Language) est le Langage de Requêtes standard pour les SGBD relationnels
- Il permet de créer, modifier, supprimer et interroger tous les types d'objets d'une BDR (domaines, les données, les relations, les vues, les index, les règles etc.)

3

## Types de valeur

- Booléens : `BOOLEAN`
  - `TRUE`, `FALSE`
- Numériques
  - Nombres entiers : `INT` ou `INTEGER`
  - Nombres flottants : `REAL` ou `FLOAT`
  - Nombres décimaux (nbre chiffres `d` après la virgule fixé) : `DECIMAL(n, d)`
- Textes
  - Chaîne longueur fixe `n` : `CHAR(n)`
  - Chaîne longueur max `n` : `VARCHAR(n)`
- Temporels
  - `DATE`, `TIME` ...

4

# Définition de schémas

5

## Exemple Vision tabulaire

| EMPLACEMENT |         |             |             |
|-------------|---------|-------------|-------------|
| <u>numE</u> | Surface | nbrePersMax | TypeEmpl    |
| 1           | 12      | 4           | tente       |
| 2           | 12      | 4           | tente       |
| 3           | 15      | 6           | tente       |
| 4           | 20      | 8           | tente       |
| 15          | 14      | 4           | caravane    |
| 16          | 16      | 6           | caravane    |
| 17          | 19      | 6           | caravane    |
| 33          | 16      | 4           | caravane    |
| 34          | 16      | 4           | camping-car |
| 58          | 30      | 4           | camping-car |
| 59          | 36      | 4           | bungalow    |
| 60          | 50      | 5           | bungalow    |
| 61          | 60      | 6           | bungalow    |

| SPORT         |             |            |
|---------------|-------------|------------|
| <u>nomS</u>   | UniteLoc    | TarifUnite |
| tennis        | 1 heure     | 6          |
| VTT           | 1/2 journée | 7,5        |
| planche-voile | 1 journée   | 11,5       |
| canoë         | 2 heures    | 3,75       |
| pétanques     | 2 heures    | 2          |

| SEJOUR     |           |            |            |        |               |
|------------|-----------|------------|------------|--------|---------------|
| <u>num</u> | nomClient | DebSejour  | FinSejour  | nbPers | <u>#Num E</u> |
| 1          | Merle     | 01/07/2003 | 15/07/2003 | 3      | 4             |
| 2          | Poussin   | 01/07/2003 | 10/07/2003 | 4      | 2             |
| 3          | Rossignol | 01/07/2003 | 10/07/2003 | 5      | 17            |
| 4          | Canaris   | 02/07/2003 | 09/07/2003 | 2      | 34            |
| 5          | Piaf      | 02/07/2003 | 21/07/2003 | 2      | 1             |
| 6          | Rose      | 05/01/2003 | 20/07/2003 | 6      | 60            |

| LOCATION    |              |            |            |
|-------------|--------------|------------|------------|
| <u>#num</u> | <u>#NomS</u> | Date       | nbreUnites |
| 3           | tennis       | 03/07/2003 | 1          |
| 5           | canoë        | 03/07/2003 | 2          |
| 5           | tennis       | 04/07/2003 | 1          |
| 3           | VTT          | 05/07/2003 | 2          |
| 11          | tennis       | 11/07/2003 | 2          |

6



## Illustration de la création de domaine

---

```
CREATE DOMAIN typeEmplacement  
AS VARCHAR(20)  
CHECK (VALUE IN ('tente', 'bungalow', 'camping-car',  
    'caravane'));
```

7



## Création de table

---

```
CREATE TABLE <nom>  
(  
    <Nom_attribut> <type> <valeur_par_défaut> <contrainte>,  
    <Nom_attribut> <type> <valeur_par_défaut> <contrainte>,  
    <Nom_attribut> <type> <valeur_par_défaut> <contrainte>,  
    <Nom_attribut> <type> <valeur_par_défaut> <contrainte>,  
    ...,  
    <Contrainte>,  
    <Contrainte>,  
    <Contrainte>  
    ...  
);
```

8

## Illustration : création d'une table (sans contraintes)

```
CREATE TABLE EMPLACEMENT
(
NumE VARCHAR(5),
Surface INTEGER DEFAULT 6,
NbrePersMax INTEGER,
TypeEmpl typeEmplacement
);
```

9

## Expression des contraintes d'intégrité

- Contraintes sur les définitions de noms d'attributs (portent sur un seul attribut)
  - Valeur nulle impossible → NOT NULL
  - Clé → PRIMARY KEY
  - Clé étrangère → REFERENCES <table\_référencée>[(<colonnes>)]
  - Contrainte de domaine → CHECK <condition>
- Les contraintes peuvent être nommées (optionnel)
  - Si la contrainte est violée, le nom de la contrainte est présente dans les messages d'erreur

```
CONSTRAINT <nom_contrainte>
```

10

# Expression des contraintes d'intégrité

- Contraintes de relations (sur plusieurs attributs)
  - Clé → PRIMARY KEY
  - Contrainte générale → CHECK <condition>

11

## Illustration

```
CREATE TABLE EMPLACEMENT
(
  NumE VARCHAR(5) NOT NULL PRIMARY KEY,
  Surface INTEGER DEFAULT 6,
  NbrePersMax INTEGER,
  TypeEmpl TypeEmplacement,
);
```

12



## Illustration

```
CREATE TABLE SEJOUR
(
  Num INTEGER NOT NULL PRIMARY KEY,
  NomClient VARCHAR(20) NOT NULL,
  DateDeb DATE,
  DateFin DATE,
  NbrePers INTEGER,
  NumE VARCHAR(5) NOT NULL REFERENCES EMBLACEMENT,

  CONSTRAINT dates
  CHECK (DateDeb < DateFin)
);
```

13



## Illustration

```
CREATE TABLE SPORT
(
  NomS VARCHAR(20) NOT NULL,
  UnitéLoc VARCHAR(10) NOT NULL,
  TarifUnite real NOT NULL,

  PRIMARY KEY(NomS)
);
```

```
CREATE TABLE LOCATION
(
  Num INTEGER NOT NULL REFERENCES
    SEJOUR(num),
  NomS VARCHAR(20)
    REFERENCES SPORT(NomS),
  Date DATE,
  NbreUnites INTEGER DEFAULT 1
    CHECK (NbreUnites BETWEEN 1 AND 3),

  PRIMARY KEY(Num, NomS, Date),
);
```

14



## Alimentation de la base de données

---

- INSERT

- Insertion de lignes dans une relation

```
INSERT INTO <relation>[( attribut [,attribut] ... )]  
VALUES (<valeur> [, < valeur >])
```

```
INSERT INTO SPORT  
VALUES ('ping-pong', 'heure', 3)
```

15



## Recherche de données

---

16



## Structure d'une requête SQL

- Une requête SQL est composée à partir d'un opérateur de base qui produit une table

```
SELECT
FROM
[WHERE]
[ORDER BY]
```

- Des opérateurs d'union, d'intersection ou de différence
- Des opérateurs de jointures
- Des expressions d'attributs et des fonctions de calcul

17

## Expression des projections

```
SELECT    <liste_attributs>
FROM      <relation>
```

*Q1. Visualiser toutes les informations des séjours ?*

```
SELECT    *
FROM      SEJOUR
```

18

## Expression des projections

*Q2. Quels sont les noms des sports proposés par le camping ?*

```
SELECT    nomS
FROM      SPORT
```

*Q3. Quels sont les noms des sports proposés et leur tarif à l'unité TTC?*

```
SELECT    nomS, TarifUnité*1.196
FROM      SPORT
```

19

## Expression des projections

```
SELECT    DISTINCT <liste_attributs>
FROM      <relation>
```

*Q3bis. Quels sont les sports loués ?*

```
SELECT    DISTINCT nomS
FROM      LOCATION
```

20

## Expression des sélections

On évalue dans l'ordre  
la clause `from` puis,  
la clause `where`  
et enfin la clause `select`

```
SELECT    <liste_attributs>
FROM      <relation>
WHERE     <condition>
```

*Q4. Quels sont les noms des sports proposés dont le tarif à l'unité est supérieur à 3 euros ?*

```
SELECT    nomS
FROM      SPORT
WHERE     TarifUnite > 3
```

21

## Expression des sélections

*Q5. Quels sont les noms des sports proposés dont le tarif est compris entre 1 et 3 euros ?*

```
SELECT    nomS
FROM      SPORT
WHERE     TarifUnite BETWEEN 1 AND 3
```

```
SELECT    nomS
FROM      SPORT
WHERE     TarifUnite >=1
AND       TarifUnité<=3
```

22

## Expression des sélections

*Q6. Quels sont les numéros des emplacements dont la surface est inférieure ou égal à 7m2 et pouvant accueillir tentes ou caravanes ?*

```
SELECT    numE
FROM      EMBLACEMENT
WHERE     Surface <=7
AND       (TypeEmpl='tente'
           OR TypeEmpl='Caravane')
```

23

## Expression des sélections

*Q6. Quels sont les numéros des emplacements dont la surface est inférieure ou égal à 7m2 et pouvant accueillir tentes ou caravanes ?*

```
SELECT    nomE
FROM      Emplacement
WHERE     Surface <=7
AND       TypeEmpl IN ('tente', 'Caravane')
```

24



## Expression des sélections

---

*Q7. Quels sont les numéros des clients dont le nom contient 'trop' ?*

```
SELECT    Num
FROM      SEJOUR
WHERE     NomCl LIKE '%trop%'
```

25



## Expression des sélections

---

*Q8. Quels sont les noms des clients et la date de début de leur séjour pour des clients ne sachant pas encore quelle est la date de leur départ ?*

```
SELECT    NomCl, Datedeb
FROM      Sejour
WHERE     DateFin IS NULL
```

26



## Fonctions de calcul et agrégats

---

*Q9. Combien y a-t-il de clients actuellement ?*

```
SELECT    count(num)
FROM      Sejour
```

```
SELECT    count(num) AS NbreClients
FROM      Sejour
```

27



## Fonctions de calcul et agrégats

---

*Q10. Combien y a-t-il de clients actuellement sans départ prévu ?*

```
SELECT    count(NomCl) AS NbreClSansDepartFixe
FROM      Sejour
WHERE     DateFin IS NULL
```

28

## Fonctions de calcul et agrégats

*Q12. Quel est le tarif minimum, maximum à l'unité d'un sport proposé ? Quelle est la moyenne des tarifs à l'unité également.*

```
SELECT      min(tarifUnite),
            max(tarifUnite),
            avg(tarifUnite)
FROM        Sport
```

29

## Expression des jointures

```
SELECT  <liste_attributs>
FROM    <relations>
WHERE   <condition de jointure>
[AND   <autres conditions>]
```

30

## Expression des jointures

*Q13. Quels sont toutes les informations des sports proposés par le camping et qui ont été choisis par des vacanciers ?*

```
SELECT      *
FROM        SPORT, LOCATION
WHERE       SPORT.NomS=LOCATION.NomS
```

31

## Expression des jointures

*Q14. Nom et tarif à l'unité des sports proposés qui ont été loués ainsi que le nombre d'unités loués chaque fois ?*

```
SELECT      SPORT.NomS,
            LOCATION.nbreUnites,
            SPORT.TarifUnite
FROM        SPORT, LOCATION
WHERE       SPORT.NomS=LOCATION.NomS
```

32



## Expression des jointures

*Q15. Identique à Q14 sans ambiguïté*

```
SELECT    LOCATION.NomS,nbreUnites,TarifUnite
FROM      SPORT, LOCATION
WHERE     SPORT.NomS=LOCATION.NomS
```

*Q16. Identique à Q14 avec alias*

```
SELECT    l.NomS, nbreUnites, TarifUnite
FROM      SPORT s, LOCATION l
WHERE     s.NomS=l.NomS
```

33

## Elimination de doublons

*Q16. Quels sont les noms et le tarif à l'unité des sports proposés qui ont été choisis pour une seule unité ? Eliminer les doublons*

```
SELECT    DISTINCT NomS, TarifUnite
FROM      SPORT, LOCATION
WHERE     SPORT.NomS=LOCATION.NomS
AND       nbreUnites= 1
```

34

## Jointure naturelle

*Q17. Quels sont toutes les informations des sports proposés qui ont été loués ?*

```
SELECT      *
FROM        Sport NATURAL JOIN Location
```

```
SELECT      *
FROM        Sport INNER JOIN Location
            on (nomS= nomSport)
```

35

## Jointures externes (full outer join - outer left | right join)

*Q18. Noms et Unité de location des sports qui ont été loués ainsi que le nombre d'unités loués chaque fois ?*

*Indiquer aussi ceux qui n'ont pas été loués*

| nomS          | UniteLoc    | nbreUnites |
|---------------|-------------|------------|
| tennis        | 1 heure     | 1          |
| tennis        | 1 heure     | 1          |
| canoë         | 2 heures    | 2          |
| tennis        | 1 heure     | 2          |
| VTT           | 1/2 journée | 2          |
| pétanques     | 2 heures    | nul        |
| planche-voile | 1 journée   | nul        |

```
SELECT NomS, nbreUnites, UniteLoc
FROM    Sport LEFT OUTER JOIN Location
ON      Sport.nomS=Location.nomS
```

36

## Sous-requêtes

- Imbrication de requêtes
- Opérateurs IN (semi-jointure) et NOT IN

*Q19. Quels sont les informations concernant les séjours ayant choisi de faire du sport ?*

```
SELECT  *
FROM    Sejour
WHERE   Num IN ( SELECT Num
                  FROM Location)
```

37

## Sous-requêtes

*Q20. Quels sont les noms des clients ayant choisi de faire du sport ?*

```
SELECT  NomCl
FROM    Sejour
WHERE   Num IN ( SELECT Num
                  FROM Location)
```

38



## Sous-requêtes

*Q20. Quels sont les noms des clients ayant choisi de faire du sport ?*

```
SELECT  NomCl
FROM    Sejour
WHERE   Num IN ( SELECT Num
                  FROM    Location)
```

```
SELECT  NomCl
FROM    Sejour, Location
WHERE   Sejour.Num=Location.Num
```

39



## Sous-requêtes

*Q20. Quels sont les noms des clients ayant choisi **de faire du sport** ?*

```
SELECT  NomCl
FROM    Sejour
WHERE   Num IN ( SELECT Num FROM LOCATION)
```

*Q21. Quels sont les noms des clients ayant choisi **de ne pas faire du sport** ?*

```
SELECT  NomCl
FROM    Sejour
WHERE   Num NOT IN ( SELECT Num FROM LOCATION)
```

40

## Opérateur de tri

```
SELECT      <liste_attributs>
FROM        <relations>
[WHERE      <conditions>]
[ORDER BY   <attribut> <ordre> [, <attribut> <ordre>]]
```

41

## Opérateur de tri

*Q24. Noms et tarif des sports choisis par les clients.  
Trier le résultat par ordre croissant sur le tarif à l'unité*

```
SELECT  nomS, TarifUnite
FROM    SPORT, LOCATION
WHERE   SPORT.NomS=LOCATION.NomS
ORDER BY TarifUnite ASC
```

42

## Opérateur de tri

*Q24. Noms et tarif des sports choisis par les clients.  
Trier le résultat par ordre croissant sur le tarif à l'unité puis sur le nom du sport*

```
SELECT  nomS, TarifUnite
FROM    SPORT, LOCATION
WHERE   SPORT.NomS=LOCATION.NomS
ORDER BY TarifUnite ASC, nomS DESC
```

43

## Expression des unions, intersection

*Q26. Num des séjours ayant choisi de faire du tennis et du canoe*

```
(SELECT      num
FROM        LOCATION
WHERE       nomS='Tennis')
INTERSECT
(SELECT      num
FROM        LOCATION
WHERE       nomS='Canoe')
```

44

## Expression des unions, intersection

*Q26. Num des séjours ayant choisi de faire du tennis et du canoe*

```
(SELECT      num
FROM        LOCATION
WHERE       nomS='Tennis')
INTERSECT
(SELECT      num
FROM        LOCATION
WHERE       nomS='Canoe')
```

```
SELECT num
FROM  LOCATION
WHERE nomS='Tennis'
AND   num IN
      (SELECT num
FROM   LOCATION
WHERE  nomS='Canoe')
```

45

## Expression des unions, intersection et difference

*Q27. Num des séjours ayant choisi de faire du tennis et du canoe et pas de piscine*

```
(SELECT      num
FROM        LOCATION
WHERE       nomS='Tennis')
INTERSECT
(SELECT      num
FROM        LOCATION
WHERE       nomS='Canoe')
EXCEPT
(SELECT      num
FROM        LOCATION
WHERE       nomS='Piscine')
```

46

## Expression des unions, intersection et difference

*Q27. Num des séjours ayant choisi de faire du tennis et du canoe et pas de piscine*

```
(SELECT      num
FROM        LOCATION
WHERE       nomS= 'Tennis' )
INTERSECT
(SELECT      num
FROM        LOCATION
WHERE       nomS= 'Canoe' )
EXCEPT
(SELECT      num
FROM        LOCATION
WHERE       nomS= 'Piscine' )
```

```
SELECT num
FROM   LOCATION
WHERE  nomS= 'Tennis'
AND    num IN
      (SELECT num
FROM    LOCATION
WHERE   nomS= 'Canoe' )
AND    num NOT IN
      (SELECT num
FROM    LOCATION
WHERE   nomS= 'Piscine' )
```

47

## Mise à jour des données d'une relation





- **INSERT**
  - Insertion de lignes dans une table
- **UPDATE**
  - Modification de lignes dans une table
- **DELETE**
  - Suppression de lignes dans une table

49



## INSERT

```
INSERT INTO <relation>[( attribut [,attribut] ... )]
VALUES (<valeur> [, < valeur >])
```

```
INSERT INTO <relation>[( attribut [,attribut] ... )]
<requête>
```

```
INSERT INTO SPORT
VALUES ('ping-pong', 'heure', 3)
```

```
INSERT INTO SPORT (NomS,UniteLocation,TarifUnité)
SELECT nomSport, Unite, prix
FROM TousLesSports
WHERE TarifUnité < 10
```

50



## Update

---

```
UPDATE <relation>
    SET <attribut> =<valeur>,
        [<attribut> = <valeur>]
    [WHERE <condition>]
```

```
UPDATE SPORT
SET    TarifUnite = TarifUnite*1.1
WHERE  TarifUnite <5
```

```
UPDATE SPORT
SET    TarifUnite = TarifUnite*1.1
WHERE  NomS in (SELECT NomS FROM Location)
```

51



## Delete

---

```
DELETE FROM <relation>
    [WHERE <condition>]
```

```
DELETE FROM SPORT
WHERE TarifUnite <2
```

52



## Mise à jour de la structure de la relation

---

53



## ALTER TABLE

---

- ALTER TABLE permet de changer la structure d'une relation
  - ajouter ou supprimer des colonnes, des index,
  - changer le type des colonnes existantes,
  - renommer ces colonnes, ou la table elle-même

```
ALTER TABLE <relation>  
    <specification>
```

54



## Exemples

---

```
ALTER TABLE Sport
    RENAME COLUMN NomS TO NomSPort;
ALTER TABLE Sport
    ADD COLUMN ResponsableSport VARCHAR(10);
ALTER TABLE Sport
    DROP COLUMN ResponsableSport;
ALTER TABLE Sport
    ALTER COLUMN TarifUnite TYPE DECIMAL(3,1);
```

55



## Exemples

---

```
ALTER TABLE Sport
    ADD CONSTRAINT TarifUnite_Sport
    CHECK (TarifUnite >=0)

ALTER TABLE Sport
    ADD CONSTRAINT TarifUnite SET DEFAULT 0
```

56