

BASES DE DONNÉES

COURS 3



DDL

DATA DEFINITION LANGUAGE

CREATE TABLE
ALTER TABLE, ADD CONSTRAINT
ETC...



CRÉER UNE TABLE

CREATE TABLE

GÉNÉRALITÉS

- La commande CREATE TABLE permet de créer une table en SQL. La création d'une table sert à définir les colonnes et le type de données qui seront contenus dans chacune des colonnes (entier, chaîne de caractères, date, valeur binaire ...).
- La syntaxe générale pour créer une table est la suivante :

```
1 CREATE TABLE nom_de_la_table (  
2     colonne1 type_donnees,  
3     colonne2 type_donnees,  
4     colonne3 type_donnees,  
5     colonne4 type_donnees  
6 )
```

- Dans cette requête, 4 colonnes ont été définies. Le mot-clé “type_donnees” sera à remplacer par un mot-clé pour définir le type de données (INT, DATE, TEXT ...).

GÉNÉRALITÉS

```
1 CREATE TABLE utilisateur (  
2   id INT,  
3   nom VARCHAR(100),  
4   prenom VARCHAR(100),  
5   email VARCHAR(255),  
6   date_naissance DATE,  
7   pays VARCHAR(255),  
8   ville VARCHAR(255),  
9   code_postal VARCHAR(5),  
10  nombre_achat INT,  
11  commentaires TEXT  
12 )
```

- **id** : identifiant unique
- **nom** : nom de l'utilisateur dans une colonne de type VARCHAR avec un maximum de 100 caractères au maximum
- **prenom** : idem mais pour le prénom
- **email** : adresse email enregistré sous 255 caractères au maximum
- **date_naissance** : date de naissance enregistré au format date
- **pays** : nom du pays de l'utilisateur sous 255 caractères au maximum
- **ville** : idem pour la ville
- **code_postal** : 5 caractères du code postal
- **nombre_achat** : nombre d'achat de cet utilisateur sur le site
- **commentaires** : texte à longueur variable

A decorative wavy line in a mustard yellow color runs vertically along the left side of the image, separating a white area from a dark grey area.

MODIFIER UNE TABLE

ALTER TABLE

GÉNÉRALITÉS

- La commande ALTER TABLE en SQL permet de modifier une table existante. Idéal pour ajouter une colonne, supprimer une colonne ou modifier une colonne existante, par exemple pour changer le type.
 - La syntaxe générale pour créer une table est la suivante :

```
1 ALTER TABLE nom_table  
2   <instruction>,  
3   <instruction>;
```

- Le mot-clé “instruction” ici sert à désigner une commande supplémentaire, qui sera détaillée selon l’action que l’ont souhaite effectuer : ajouter, supprimer ou modifier une colonne.

AJOUTER UNE COLONNE

- L'ajout d'une colonne dans une table est relativement simple et peut s'effectuer à l'aide d'une requête ressemblant à ceci :

```
1 ALTER TABLE utilisateur
2   ADD adresse_rue VARCHAR(255);
3
4 ALTER TABLE utilisateur
5   ADD adresse_ville VARCHAR(255),
6   ADD adresse_code_postal VARCHAR(5);
```


SUPPRIMER UNE COLONNE

- La suppression d'une colonne dans une table est également relativement simple et peut s'effectuer à l'aide d'une requête ressemblant à ceci :

```
1 ALTER TABLE utilisateur
2     DROP COLUMN adresse_rue;
3
4 ALTER TABLE utilisateur
5     DROP COLUMN adresse_cp,
6     DROP COLUMN adresse_ville;
```

MODIFIER UNE COLONNE

- La modification d'une colonne dans une table est encore une fois simple, et modifier son type peut s'effectuer à l'aide d'une requête ressemblant à ceci :

```
1 ALTER TABLE utilisateur  
2 ALTER COLUMN numero_telephone TYPE VARCHAR(10);
```

- Attention toutefois, modifier un type ne pourrait éventuellement pas être possible si des données d'un type incompatible étaient déjà présentes.
 - Exemple : il ne serait pas possible de modifier le type du champ « prénom » de TEXT à INTEGER (car on peut se douter que des prénoms déjà renseignés ne pourraient pas être convertis en nombre entier)
- Nous verrons ensuite qu'il est possible de modifier beaucoup plus de choses que le type d'une colonne

RENOMMER UNE COLONNE

- Pour renommer une colonne, il convient d'indiquer l'ancien nom de la colonne et le nouveau nom de celle-ci.
-

```
1 ALTER TABLE utilisateur  
2   RENAME COLUMN prnom TO prenom;
```

- Attention toutefois, renommer une colonne ne pourrait pas être possible si cette action devait créer un doublon dans les noms de colonne.

A decorative wavy line in a mustard yellow color runs vertically along the left side of the image, separating a white area from the dark grey background.

SUPPRIMER UNE TABLE

DROP TABLE

GÉNÉRALITÉS

- Encore une fois, c'est très simple. Pour supprimer une table “nom_table” il suffit simplement d'utiliser la syntaxe suivante :

```
1 DROP TABLE nom_table;
```

- A savoir : s'il y a une dépendance avec une autre table, il est recommandé de les supprimer avant de supprimer la table. C'est le cas par exemple s'il y a des clés étrangères.



VIDER UNE TABLE

TRUNCATE TABLE

GÉNÉRALITÉS

- Une dernière fois, c'est très simple. Pour vider une table “nom_table” de ses entrées il suffit simplement d'utiliser la syntaxe suivante :

```
1 TRUNCATE nom_table;
```

A decorative graphic on the left side of the slide consisting of two parallel, wavy, vertical lines. The inner line is a dark olive green color, and the outer line is a light cream color. They start from the top left and extend towards the bottom left, creating a stylized, organic shape.

LES INDEXES

A QUOI SERVENT LES INDEXES ?

- Un index nous permet de manière générale de rechercher et de trouver plus rapidement des informations.
- Un index implique une manière précise et connue à l'avance de « classer » les éléments à chercher, afin de les retrouver beaucoup plus rapidement.
 - Dans le cas du dictionnaire, par exemple, l'indexation se fait par ordre alphabétique des mots définis.
 - Comment faire si nous voulions tous les mots qui ont dans leur définition le mot « kiwi » ?
 - Réponse 1 : Lorsqu'on nous le demande, lire le dictionnaire en entier, et récupérer les mots dont les définitions contiennent le mot kiwi
 - Réponse 2 : Ecrire un second livre qui met en face de chaque mot unique trouvé dans les descriptions, le mot auquel appartient cette définition

UN INDEX EN BASE DE DONNÉES

- Un index en base de données, c'est exactement la même chose.
- C'est une manière définie à l'avance de classer des éléments afin de les retrouver plus rapidement.
- Il implique plusieurs choses
 - Un index prend de la place en mémoire, car il faut stocker ce nouveau modèle recherche
 - Si une recherche se fait sur des champs multiples, la définition de l'index doit se faire sur cette liste de champs.
 - Un index est avant tout une optimisation. Il n'est pas nécessaire de définir un index si son utilisation est anecdotique, ou que les performances sont sensiblement les mêmes.

UN INDEX EN BASE DE DONNÉES

- Imaginons la table d'utilisateurs suivante possède 10 millions d'entrées :

Utilisateur
<u>id</u> : <i>integer</i>
email : <i>text</i>

- Sans avoir indexé quoi que ce soit, que se passerait-il si nous recherchions un email « x » ?
 - Le moteur de base de données parcourt toutes les entrées de la table utilisateurs
 - Compare chaque champ email à la valeur « x », 10 millions de fois
 - Retourne les champs correspondants.
 - Temps estimé d'exécution : 1000 ans.
- Si on indexe le champ email, voici ce qu'il se passerait si nous recherchions un email « x » :
 - Retourne les champs correspondants.
 - Temps estimé d'exécution : 1 ms.

CREATE INDEX

- Créer un index peut s'effectuer à l'aide d'une requête ressemblant à ceci :

```
1 CREATE INDEX index_1 ON ma_table (colonne_1);  
2 CREATE INDEX index_2 ON ma_table (colonne_2, colonne_3);
```

- Il n'existe pas de convention de nommage spécifique sur le nom des index, juste des suggestions de quelques développeurs et administrateurs de bases de données.
- Voici une liste de suggestions de préfixes à utiliser pour nommer un index
 - Préfixe “**PK_**” pour **P**rietary **K**ey
 - Préfixe “**FK_**” pour **F**oreign **K**ey
 - Préfixe “**UX_**” pour **U**nique **I**ndex
 - Préfixe “**IX_**” pour chaque autre **I**ndex

```
1 CREATE INDEX ix_foo ON ma_table (bar);  
2 CREATE INDEX ix_bar ON ma_table (baz);
```



CONTRAINTES D'INTÉGRITÉ

CONTRAINTES ET OPTIONS

- Les contraintes d'intégrité et options servent à garantir un des aspects fondamentaux des SGBD :
 - L'intégrité des données
- L'intégrité des données représente l'exactitude, l'exhaustivité et la cohérence globales des données dans une base de données.
- Par exemple, grâce à l'intégrité des données, une base de données (correctement définie) garantira
 - Que 100% des bons de commande sont liés à au moins un compte client
 - Que deux comptes clients ne possèdent pas la même adresse email
 - Qu'aucun bon de commande ne possède un numéro de commande vide
 - Etc...

PRIMARY KEY

- Définition :
 - La contrainte PRIMARY KEY spécifie qu'une ou plusieurs colonnes d'une table ne peuvent contenir que des valeurs uniques (non dupliquées) et non nulles.
 - Une seule clé primaire peut être spécifiée pour une table, qu'il s'agisse d'une contrainte de colonne ou d'une contrainte de table.
 - PRIMARY KEY applique les mêmes contraintes de données qu'une combinaison de UNIQUE et NOT NULL.
- Où trouver cette contrainte :
 - Définition de la colonne
 - Définition de la table
- Syntaxe :

```
1 CREATE TABLE exemple_primary_key_1 (  
2   id SERIAL PRIMARY KEY  
3 );  
4 CREATE TABLE exemple_primary_key_2 (  
5   id VARCHAR(255) CONSTRAINT contrainte_1 PRIMARY KEY  
6 );  
7 CREATE TABLE exemple_primary_key_3 (  
8   champs_1 INTEGER,  
9   champs_2 TEXT,  
10  CONSTRAINT contrainte_2 PRIMARY KEY (champs_1, champs_2)  
11 );  
12  
13 ALTER TABLE exemple_primary_key_3  
14   DROP CONSTRAINT contrainte_2;  
15  
16 ALTER TABLE exemple_primary_key_3  
17   ADD CONSTRAINT contrainte_2 PRIMARY KEY (champs_1);  
18
```

FOREIGN KEY

- Définition :
 - La clause FOREIGN KEY exige qu'un groupe d'une ou plusieurs colonnes de la table ne puisse contenir que des valeurs correspondant aux valeurs de la ou des colonnes référencées d'une ligne de la table référencée.
- Où trouver cette contrainte :
 - Définition de la colonne
 - Définition de la table
- Syntaxe :

```
1 CREATE TABLE table_1 (  
2   id INTEGER PRIMARY KEY  
3 )  
4  
5 CREATE TABLE exemple_foreign_key_1 (  
6   id_table_1 INTEGER REFERENCES table_1  
7 )  
8  
9 CREATE TABLE exemple_foreign_key_2 (  
10  id_table_1 INTEGER REFERENCES table_1(id)  
11 )  
12  
13 CREATE TABLE exemple_foreign_key_3 (  
14   id_table_1 INTEGER,  
15   CONSTRAINT fk_fk3_t1 FOREIGN KEY (id_table_1) REFERENCES table_1(id)  
16 )  
17  
18 ALTER TABLE exemple_foreign_key_3  
19   DROP CONSTRAINT fk_fk3_t1;  
20  
21 ALTER TABLE exemple_foreign_key_3  
22   ADD CONSTRAINT fk_fk3_t1 FOREIGN KEY (id_table_1) REFERENCES table_1(id);
```


UNIQUE

- Définition :
 - La contrainte UNIQUE spécifie qu'un groupe d'une ou plusieurs colonnes d'une table ne peut contenir que des valeurs uniques.
- Où trouver cette contrainte :
 - Définition de la colonne
 - Définition de la table
 - Définition des indexes
- Syntaxe :

```
1 CREATE TABLE exemple_unique (  
2   champs_1 TEXT UNIQUE,  
3   champs_2 TEXT,  
4   champs_3 TEXT,  
5   champs_4 TEXT,  
6   champs_5 TEXT,  
7   UNIQUE(champs_2),  
8   CONSTRAINT contrainte_1 UNIQUE(champs_3),  
9   CONSTRAINT contrainte_2 UNIQUE(champs_4, champs_5)  
10 );  
11  
12 ALTER TABLE exemple_unique  
13   DROP CONSTRAINT contrainte_1,  
14   ADD CONSTRAINT contrainte_3 UNIQUE (champs_3),  
15   ADD CONSTRAINT contrainte_4 UNIQUE (champs_3, champs_4, champs_5);  
16  
17 CREATE UNIQUE INDEX contrainte_5 ON exemple_unique(champs_3, champs_5);  
18 DROP INDEX contrainte_5;
```

CHECK

- Définition :
 - La contrainte UNIQUE permet de s'assurer que la valeur à insérer respecte des conditions logiques (ex : 'age > 18')
- Où trouver cette contrainte :
 - Définition de la colonne
 - Définition de la table
- Syntaxe :

```
1 CREATE TABLE products_a (  
2     price NUMERIC CHECK (price > 0),  
3     discounted_price NUMERIC CONSTRAINT discounted_price_positive CHECK  
4     (discounted_price > 0),  
5     CONSTRAINT valid_discount CHECK (price > discounted_price)  
6 );  
7  
7 CREATE TABLE products_b (  
8     price NUMERIC,  
9     discounted_price NUMERIC,  
10    CONSTRAINT positive_prices CHECK (price > 0 AND discounted_price > 0),  
11    CONSTRAINT valid_discount CHECK (price > discounted_price)  
12 );  
13  
14 ALTER TABLE products_b ADD CONSTRAINT max_price CHECK (price < 10000);  
15 ALTER TABLE products_b DROP CONSTRAINT max_price;  
16
```

NULL / NOT NULL

- Définition :
 - NOT NULL : La colonne ne doit pas contenir de valeurs nulles.
 - NULL : La colonne est autorisée à contenir des valeurs nulles. Il s'agit de la valeur par défaut.
- Où trouver cette option :
 - Définition de la colonne
- Syntaxe :

```
1 CREATE TABLE not_null_exemple (  
2   champs_1 TEXT NOT NULL,  
3   champs_2 INTEGER  
4 )  
5  
6 ALTER TABLE not_null_exemple  
7   ALTER COLUMN champs_1 DROP NOT NULL,  
8   ALTER COLUMN champs_2 SET NOT NULL;
```

DEFAULT

- Définition :
 - DEFAULT attribue une valeur par défaut si aucune données n'est indiquée pour cette colonne lors de l'ajout d'une ligne dans la table.
- Où trouver cette option :
 - Définition de la colonne
- Syntaxe :

```
1 CREATE TABLE exemple_default (  
2   champs_1 TEXT DEFAULT 'hello world',  
3   champs_2 INTEGER NOT NULL,  
4 );  
5  
6 ALTER TABLE exemple_default  
7   ALTER COLUMN champs_1 DROP DEFAULT,  
8   ALTER COLUMN champs_2 SET DEFAULT 0;
```

A decorative graphic on the left side of the slide consisting of two parallel, wavy, vertical lines. The inner line is a dark gold color, and the outer line is white. They start from the top left and extend towards the bottom left, creating a stylized, organic shape.

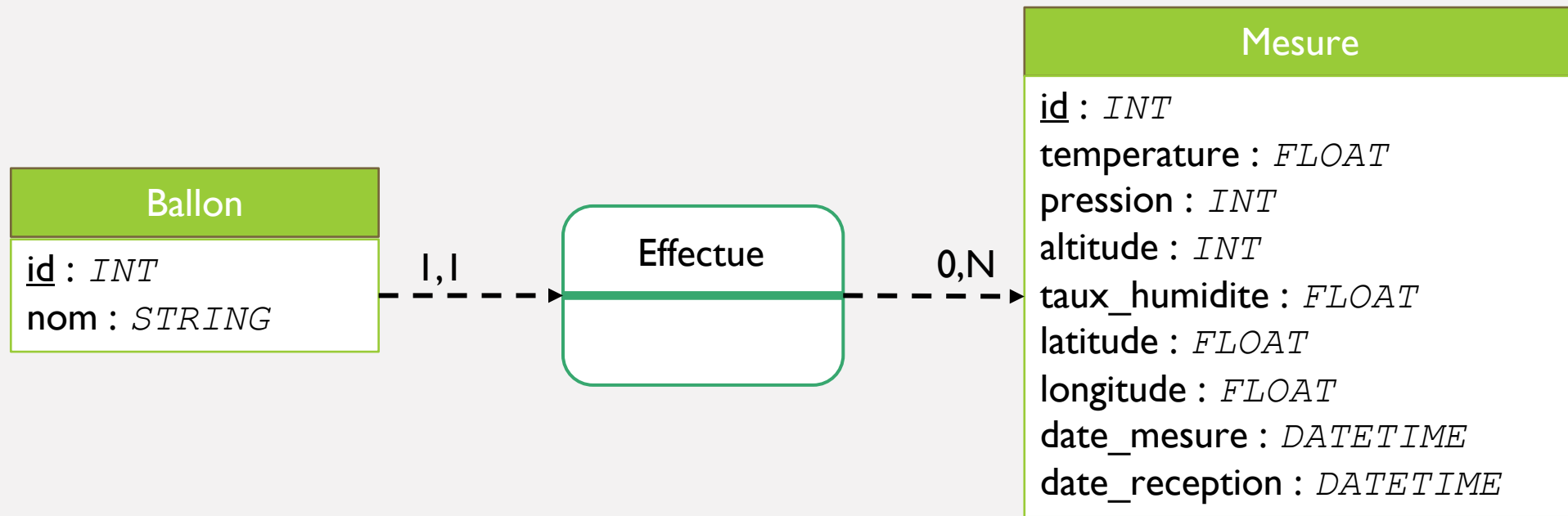
EXERCICES

EXERCICE 1/3

Un météorologue envoie des ballon-sonde pour effectuer des mesures de l'atmosphère telles que la température, la pression atmosphérique (en Pa), le taux d'humidité (en %), l'altitude (en mètre), et des coordonnées GPS. Ces mesures sont enregistrées toutes les 10 minutes et s'étalent sur une période de 7 jours. Les mesures sont envoyées 2 fois par jour au poste de commande, qui enregistre ensuite les données. Chaque ballon-sonde possède un nom qui lui est propre.

À l'aide du MCD suivant, et de la modélisation des tables vu dans le cours précédent, proposer les requêtes de création de table

EXERCICE 1 – MCD



EXERCICE 2

Imaginons maintenant que nous souhaitions représenter les différents postes de commandes qui possèdent et déploient les ballons-sondes.

Chaque poste de commande possède un nom obligatoire unique, et une adresse (rue, ville, code postal, pays).

Les ballons-sondes sont équipés de nouveaux capteurs qui permettent maintenant de récupérer la concentration de CO₂ dans l'atmosphère (en PPM), au détriment du taux d'humidité.

Les postes de commandes sont dirigés par des scientifiques avec un nom et un prénom obligatoire, qui peuvent administrer plusieurs postes de commande à la fois. Chaque poste de commande peut être géré par plusieurs scientifiques.

A partir de la solution de l'exercice précédent,

- Proposer une évolution du MCD qui comporterait ces modifications.
- Puis proposer une évolution du schéma des tables SQL.
- Enfin, rédiger les requêtes SQL de modification de la structure de la base de données.

EXERCICE BONUS

Modéliser les entités et les relations des fonctionnalités principales d'un site de e-commerce.
Minimum 7 tables, 50 attributs et 10 relations.