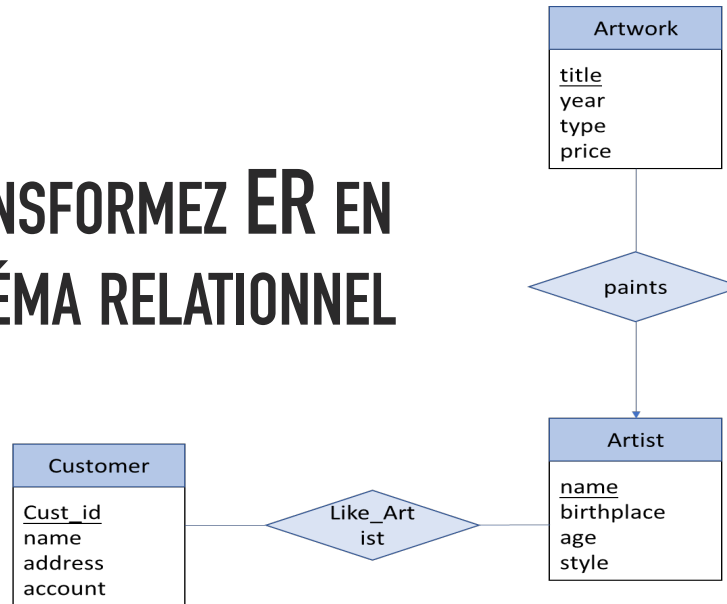


Plus de pratique
pour **lire**, **écrire** et
modifier vos
données avec SQL?

OBJECTIFS

- Réexaminer CREATE, INSERT, SELECT, UPDATE, DELETE
- Modification des bases de données avec ALTER TABLE
- Pratiquer à écrire des requêtes SQL

TRANSFORMEZ ER EN SCHÉMA RELATIONNEL



*PAS VRAIMENT, MAIS POUR AUJOURD'HUI OUI

Rappelez comment nous transformons les diagrammes ER en schémas relationnels.

CRÉER VOTRE BASE DE DONNÉES

Artist		
AName	Varchar(20)	Primary Key
Birthplace	Varchar(20)	
Style	Varchar(20)	
DateOfBirth	Date	

Customer		
CustId	Integer	Primary Key
Name	Varchar(20)	
Address	Varchar(20)	
Amount	Numeric(8,2)	

Artwork		
Title	Varchar(20)	Primary Key
Year	Integer	
Type	Varchar(20)	
Price	Numeric(8,2)	
Aname	Varchar(20)	Foreign Key

LikeArtist			
CustId	Integer	Primary Key	Foreign Key
AName	Varchar(20)	Primary Key	Foreign Key

Un diagramme relationnel peut être exprimé sous la forme d'un schéma SQL. Les noms dans le diagramme diffèrent du SQL ci-dessous.

```
CREATE TABLE artists (
  name VARCHAR(20),
  birthplace VARCHAR(20),
  style VARCHAR(20),
  dateofbirth DATE,
  PRIMARY KEY (name)
);
```

```
CREATE TABLE customers (
  id INTEGER,
  name VARCHAR(20),
  address VARCHAR(20),
  amount numeric(8,2),
```

PRIMARY KEY (id)

);

CREATE TABLE artworks (

title VARCHAR(20),

year INTEGER,

type VARCHAR(20),

price numeric(8,2),

artist_name VARCHAR(20),

PRIMARY KEY (title),

FOREIGN KEY(artist_name) REFERENCES artists(name)

);

CREATE TABLE likeartists (

customer_id INTEGER,

artist_name VARCHAR(20),

PRIMARY KEY(artist_name, customer_id),

FOREIGN KEY (artist_name) REFERENCES artists(name),

FOREIGN KEY (customer_id) REFERENCES customers(id)

);

AJOUTEZ DES DONNÉES À VOTRE BASE DE DONNÉES

INSERT

Insérez les données suivantes dans votre base de données

```
INSERT INTO customers
(id, name, address, amount)
VALUES
(1, 'John', 'Ottawa', 8.5),
(2, 'Amy', 'Orleans', 9.0),
(3, 'Peter', 'Gatineau', 6.3);
```

```
INSERT INTO artists
(name, birthplace, style, dateofbirth)
VALUES
('Caravaggio', 'Milan', 'Baroque', '1571-09-28' ),
('Smith', 'Ottawa', 'Modern', '1977-12-12'),
('Picasso', 'Malaga', 'Cubism', '1881-10-25');
```

```
INSERT INTO artworks
(title, year, type, price, artist_name)
VALUES
('Blue', 2000, 'Modern', 10000.00, 'Smith'),
('The Cardsharps', 1594, 'Baroque', 40000.00, 'Caravaggio');
```

LISTE TOUS LES ARTISTES NÉS À OTTAWA

SELECT

```
SELECT *  
FROM artists  
WHERE birthplace = 'Ottawa';
```


LISTE DES **TITRES** ET DES
PRIX DE TOUTES LES
OUVRAGES D'ART EN **2000**.

SELECT

```
SELECT title, price  
FROM artworks  
WHERE year = 2000;
```

METTRE À JOUR LE MONTANT
POUR TOUS LES CLIENTS DE
GATINEAU À 9.8.

UPDATE

```
UPDATE customers  
SET amount = 9.8  
WHERE address = 'Gatineau';
```

SUPPRIMER LE
CLIENT JOHN.

DELETE

```
DELETE  
FROM customers  
WHERE name = 'John';
```

AJOUTER LA COLONNE COUNTRY À ARTISTS.

ALTER TABLE

La commande ALTER TABLE nous permet d'apporter plusieurs modifications à une table que nous avons créée auparavant. Nous pouvons ajouter / supprimer des colonnes et des contraintes, renommer le nom de la table, les colonnes et bien plus.

Ajouter une colonne à une table déjà créée

```
ALTER TABLE TableName
```

```
ADD COLUMN ColumnName ColumnType;
```

```
ALTER TABLE artists
```

```
ADD COLUMN country varchar(100);
```

AJOUTEZ UNE COLONNE **RATING**
À **CUSTOMERS**, AVEC LA
CONTRAINTE: LA VALEUR DOIT
ÊTRE ENTRE **1** ET **10**.

CHECK

Nous pouvons également ajouter une colonne avec une contrainte d'intégrité supplémentaire.

```
ALTER TABLE TableName  
ADD COLUMN ColumnName ColumnType  
CHECK ( Constraint );
```

```
ALTER TABLE customers  
ADD COLUMN rating integer CHECK (rating between 1 and 10);
```

GROUP BY HAVING ORDER BY SELECT

SELECT peut être étendue en ajoutant plus de clauses.

- * GROUP BY: regroupe tous les résultats de requête en termes d'un ou plusieurs attributs avec cette clause.
- * HAVING: Une clause "where" sur les attributs d'agrégat / groupe.
- * ORDER BY: Trier les données en fonction d'un ou plusieurs attributs

SELECT [**DISTINCT**] *select-list*
FROM *from-list*
WHERE *record-qualification*
GROUP BY *grouping-list*
HAVING *group-qualification*

AJOUTEZ (ENCORE) DES DONNÉES À VOTRE BASE DE DONNÉES

INSERT

Insérez les données suivantes dans votre base de données

```
INSERT INTO customers
(id, name, address, amount, rating)
VALUES
(4, 'Emre', 'Preston', 20000.00, 5),
(5, 'Saeid', null, 40000.00, 6);
```

```
INSERT INTO artists
(name, birthplace, style, dateofbirth, country)
VALUES
('Leonardo', 'Florence', 'Renaissance', '04-15-1452', 'Italy'),
('Michelangelo', 'Arezzo', 'Renaissance', '03-06-1475',
'Italy'),
('Josefa', 'Seville', 'Baroque', '09-09-1630', 'Spain'),
('Hans Hofmann', 'Weisenburg', 'Modern', '02-17-1966', 'Germany'),
```



```
('John', 'San Francisco', 'Modern', '02-17-1920', 'USA');
```

```
INSERT INTO artworks
```

```
(title, year, type, price, artist_name)
```

```
VALUES
```

```
('Waves', 2000, null, 4000.00, 'Smith'),
```

```
('Three Musicians', 1921, 'Modern', 11000.00, 'Picasso');
```

```
INSERT INTO likeartists
```

```
(customer_id, artist_name)
```

```
VALUES
```

```
(4, 'Picasso'),
```

```
(5, 'Picasso'),
```

```
(5, 'Leonardo');
```

SUPPRIMEZ L'ARTISTE SMITH

Supposons que l'artiste «Smith» ait déménagé dans une autre galerie et on veut supprimer de notre base de données. Notez que la table artworks a une clé étrangère vers la table artists.

Deux façons d'écrivez une requête DELETE:

Manuel: Nous supprimons tous les enregistrements de toutes les tables liées
à l'enregistrement 'Smith' dans artists

Automatisé: nous supprimons «Smith» de l'artiste et toutes les informations
associées sont supprimées par le DBMS.

Essayer les deux (mais vous devez sauvegarder et restaurer la base de données avant).

SUPPRIMEZ L'ARTISTE SMITH

"MANUELLE" DELETE

1. Supprimez toutes les œuvres d'art liées à Smith.
2. Supprimez ensuite Smith de la liste des artistes.

```
DELETE  
FROM artworks  
WHERE artist_name = 'Smith';
```

```
DELETE  
FROM artists  
WHERE name = 'Smith';
```

```
FOREIGN KEY (ARTIST_NAME)  
REFERENCES ARTISTS(NAME)  
ON DELETE CASCADE  
ON UPDATE CASCADE;  
  
ON CASCADE
```

Supprimez toutes les tables avec l'instruction DROP TABLE. Ou exécutez ALTER TABLE pour supprimer / rajouter la clé étrangère avec la propriété CASCADE.

```
BEGIN;  
ALTER TABLE artworks  
DROP CONSTRAINT artworks_artist_name_fkey;  
  
ALTER TABLE artworks  
ADD CONSTRAINT artworks_artist_name_fkey  
FOREIGN KEY (artist_name) REFERENCES artists(name)  
ON DELETE CASCADE  
ON UPDATE CASCADE;  
  
COMMIT;
```

SUPPRIMEZ L'ARTISTE SMITH

"AUTOMATIQUE" DELETE

```
SELECT *  
FROM artworks; -- Smith exists!!!
```

```
DELETE  
FROM artists  
WHERE name = 'Smith';
```

```
SELECT *  
FROM artworks; -- Smith will be gone!!!
```

Créez une branche **lab06**

```
nd (main)$ git checkout -b Lab06  
ch 'lab02'  
nd (lab02)$ git push -u origin HEAD  
used 0 (delta 0)  
  
request for Lab06 on GitHub by vis  
ithub.com/professor-forward/csi2532_  
  
r-forward/csi2532_plaground.git  
HEAD -> lab02  
to track remote branch 'lab02' from  
nd (lab02)$
```

1. Lister les name et birthplace de tous les artists
2. Lister le title et le price de toutes les artworks après 1600.
3. Lister le title et le type de toutes les artworks qui ont été peintes en 2000 ou peintes par Picasso.
4. Lister les name et birthplace de tous les artists nés entre 1880 et 1930. (ASTUCE: `EXTRACT(YEAR FROM dateofbirth)` vous donne l'année à partir d'un attribut DATE)
5. Lister les name et le country de naissance de tous les artists dont le style de peinture est Modern, Baroque or Renaissance. (ASTUCE: utilisez le mot-clé IN).
6. Lister tous les détails des artworks dans la base de données, triés par title.

Écrivez des requêtes SQL pour ces énoncés et afficher les résultats.

7. Lister les name et les customer ids de tous les customers qui aiment Picasso.
8. Lister les name de tous les customers qui aiment les artistes de style Renaissance et dont le price est supérieur à 30000.

Écrivez des requêtes SQL pour ces énoncés et afficher les résultats. Notez que les deux requêtes suivantes impliquent plusieurs tables.

SOUSSION

- Créer un schema.sql et seed.sql pour notre base de données artists
- Créer un lab06.md avec
 - Les requêtes SQL pour les énonces suivantes.
 - Affichez les résultats de chaque requête.
- Partager avec TA