

Tableaux multiples

Jointure externe

Une jointure externe combinera des lignes de différentes tables même si la condition de jointure n'est pas remplie. Dans un `LEFT JOIN`, chaque ligne de la table de *gauche* est renvoyée dans le jeu de résultats, et si la condition de jointure n'est pas remplie, `NULL` les valeurs sont utilisées pour remplir les colonnes de la table de *droite*.

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
    ON table1.column_name
    = table2.column_name;
```

WITH Clause

La `WITH` clause stocke le résultat d'une requête dans une table temporaire (`temporary_movies`) à l'aide d'un alias.

Plusieurs tables temporaires peuvent être définies avec une instance du mot- `WITH` clé.

```
WITH temporary_movies AS (
    SELECT *
    FROM movies
)
SELECT *
FROM temporary_movies
WHERE year BETWEEN 2000 AND 2020;
```

UNION Clause

La `UNION` clause est utilisée pour combiner les résultats qui apparaissent à partir de plusieurs

`SELECT` instructions et filtrer les doublons.

Par exemple, étant donné une `first_names` table avec une colonne `name` contenant des lignes de données "James" et "Hermione", et une `last_names` table avec une colonne `name` contenant des lignes de données "James", "Hermione" et "Cassidy", le résultat de cette requête contiendrait trois `name` s: "Cassidy", "James" et "Hermione".

```
SELECT name
FROM first_names
UNION
SELECT name
FROM last_names
```

CROSS JOIN Clause

La `CROSS JOIN` clause est utilisée pour combiner chaque ligne d'une table avec chaque ligne d'une autre dans le jeu de résultats. Ceci `JOIN` est utile pour créer toutes les combinaisons possibles pour les enregistrements (lignes) dans deux tables.

La requête donnée sélectionnera les colonnes

`shirt_color` et `pants_color` dans le jeu de résultats,

qui contiendra toutes les combinaisons de

combinaisons de lignes dans les tables `shirts` et

`pants`. S'il y a 3 couleurs de chemises différentes dans

le `shirts` tableau et 5 couleurs de pantalons

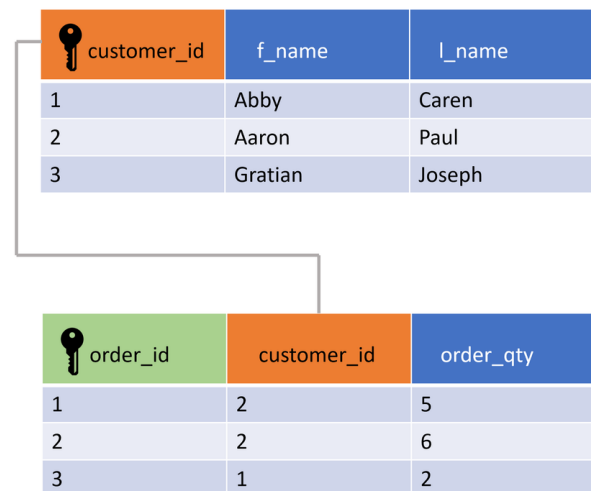
différentes dans le `pants` tableau, le jeu de résultats

contiendra $3 \times 5 = 15$ lignes.

```
SELECT shirts.shirt_color,
    pants.pants_color
FROM shirts
CROSS JOIN pants;
```

Clé étrangère

Une *clé étrangère* est une référence dans les enregistrements d'une table à la clé primaire d'une autre table. Pour conserver plusieurs enregistrements pour une ligne spécifique, l'utilisation d'une clé étrangère joue un rôle essentiel. Par exemple, pour suivre toutes les commandes d'un client spécifique, le tableau `order` (illustré au bas de l'image) peut contenir une clé étrangère.



customer_id	f_name	l_name
1	Abby	Caren
2	Aaron	Paul
3	Gratian	Joseph

order_id	customer_id	order_qty
1	2	5
2	2	6
3	1	2

Clé primaire

Une colonne de *clé primaire* dans une table SQL est utilisée pour identifier de manière unique chaque enregistrement de cette table. Une clé primaire ne peut pas être `NULL`. Dans l'exemple, `customer_id` est la clé primaire. La même valeur ne peut pas se reproduire dans une colonne de clé primaire. Les clés primaires sont souvent utilisées dans `JOIN` les opérations.

customer_id	f_name	l_name
1	Abby	Caren
2	Aaron	Paul
3	Gratian	Joseph

Jointure interne

La `JOIN` clause permet de renvoyer les résultats de plusieurs tables en les joignant à d'autres résultats basés sur des valeurs de colonne communes spécifiées à l'aide d'une `ON` clause. `INNER JOIN` est la valeur par défaut `JOIN` et ne renverra que les résultats correspondant à la condition spécifiée par `ON`.

```
SELECT *
FROM books
JOIN authors
  ON books.author_id = authors.id;
```