MYSQL 2ITECH

Réactivation mémoire

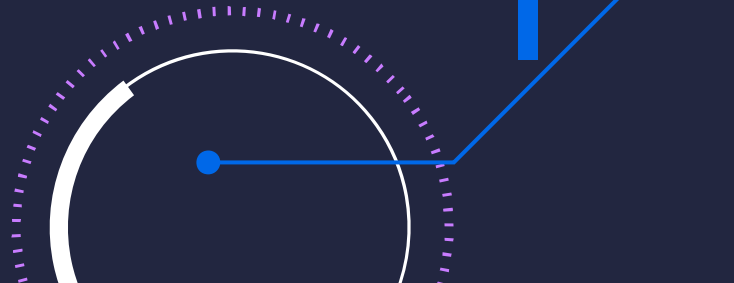
Les composants SQL

Les composants 5 principaux:

- DDL: data definition language (structure)
- DQL: data query language (lecture et sélection de data)
- DML: data manipulation language (écriture)
- DCL: data control language (contrôle d'accès)
- TCL: transaction control language

\+ 3 sous modules:

- Data types
- Functions
- Opérators



JSON

Uploaded using RayThis Extension

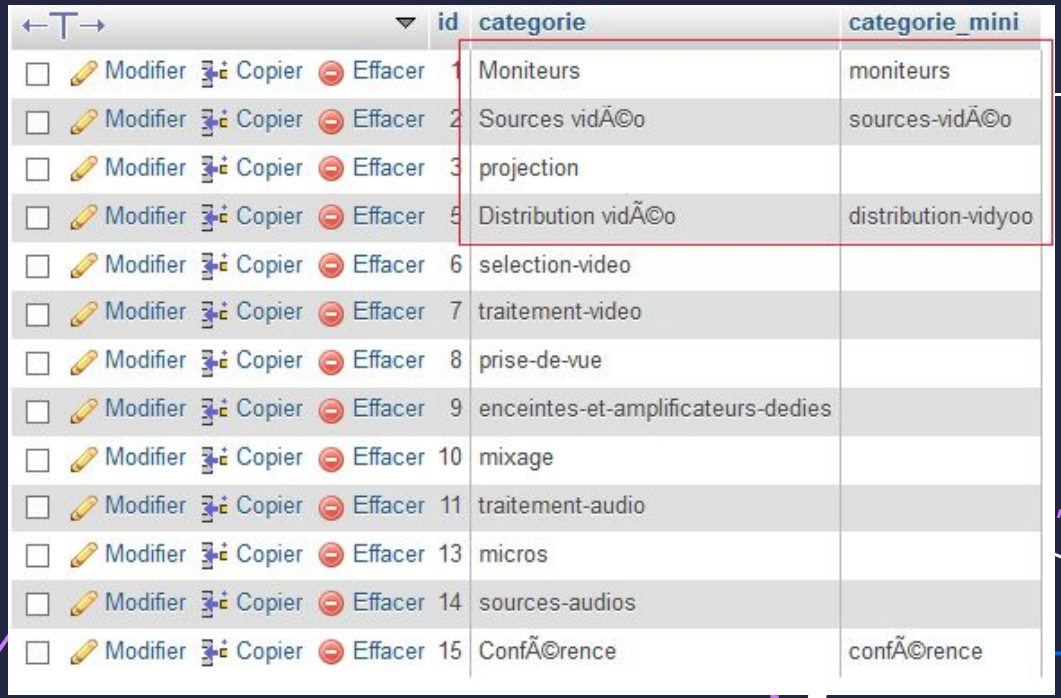
```sql
CREATE TABLE produits (
    id INT PRIMARY KEY,
    nom VARCHAR(255),
    details JSON
);

INSERT INTO produits (id, nom, details) VALUES (1, 'Produit A', '{"description": "Un produit
de qualité", "prix": 99.99, "en_stock": true}');

-- Sélection des détails du produit
SELECT nom, details->>'$.description' AS description, details->>'$.prix' AS prix
FROM produits
WHERE id = 1;
```

JSON

Uploaded using RayThis Extension

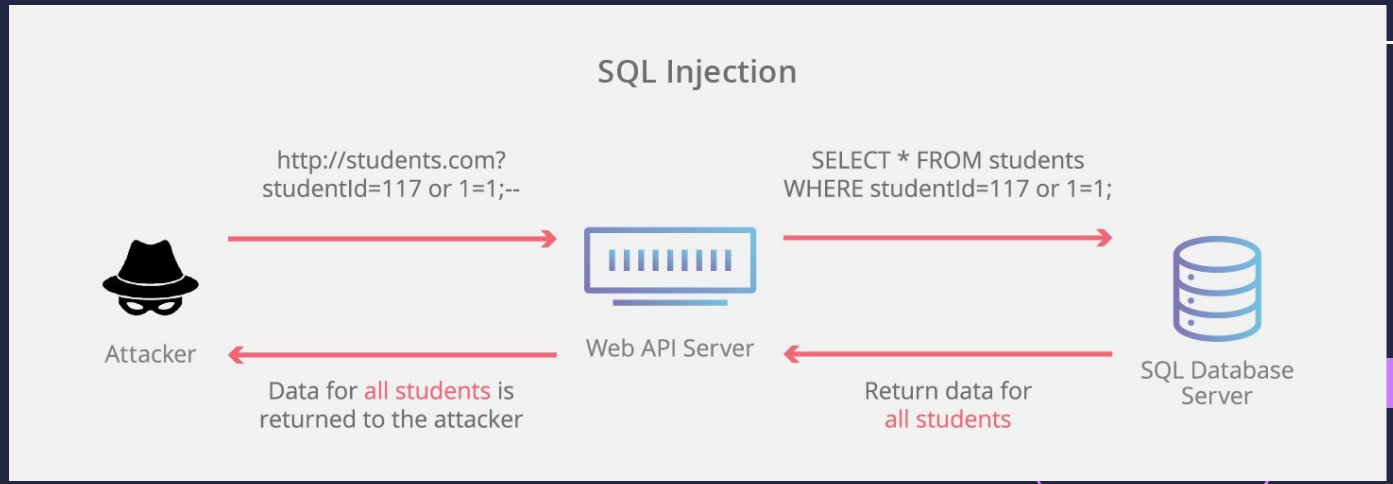
```sql
CREATE TABLE produits (
    id INT PRIMARY KEY,
    nom VARCHAR(255),
    details JSON
);

INSERT INTO produits (id, nom, details) VALUES (1, 'Produit A', '{"description": "Un produit
de qualité", "prix": 99.99, "en_stock": true}');

-- Sélection des détails du produit
SELECT nom, details->>'$.description' AS description, details->>'$.prix' AS prix
FROM produits
WHERE id = 1;
```

Encodage

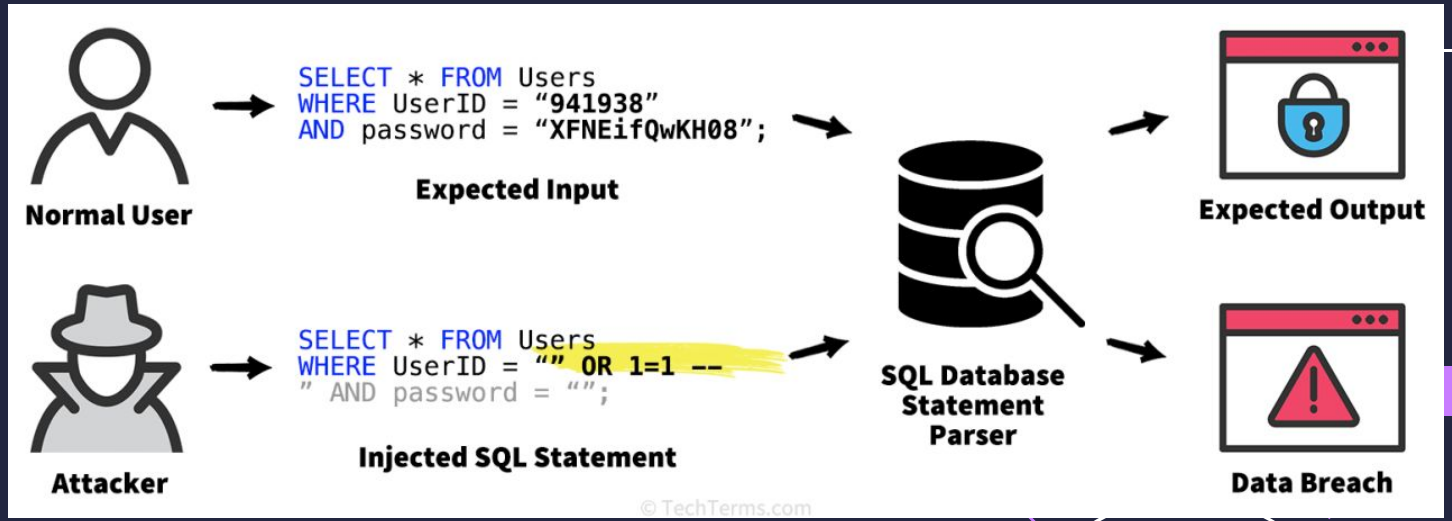


	id	categorie	categorie_mini
Modifier Copier Effacer	1	Moniteurs	moniteurs
Modifier Copier Effacer	2	Sources vidÃ©o	sources-vidÃ©o
Modifier Copier Effacer	3	projection	
Modifier Copier Effacer	5	Distribution vidÃ©o	distribution-vidyoo
Modifier Copier Effacer	6	selection-video	
Modifier Copier Effacer	7	traitement-video	
Modifier Copier Effacer	8	prise-de-vue	
Modifier Copier Effacer	9	enceintes-et-amplificateurs-dedies	
Modifier Copier Effacer	10	mixage	
Modifier Copier Effacer	11	traitement-audio	
Modifier Copier Effacer	13	micros	
Modifier Copier Effacer	14	sources-audios	
Modifier Copier Effacer	15	ConfÃ©rence	confÃ©rence

Injection SQL



Injection SQL



Les ORM

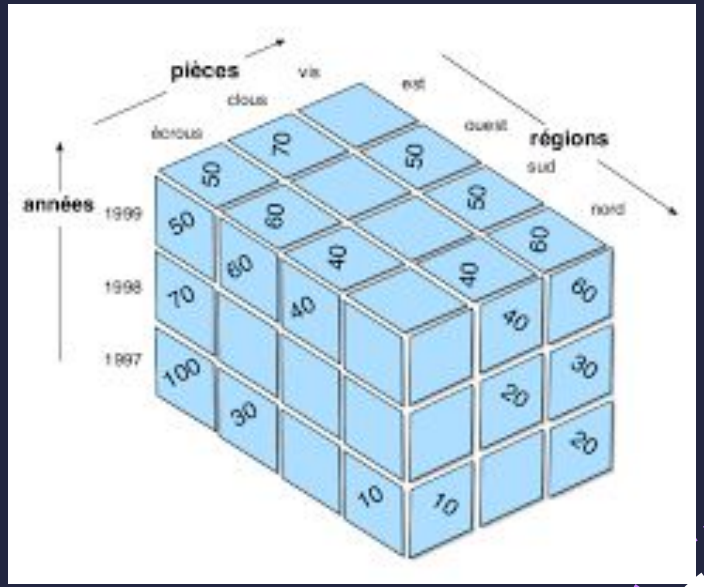

index.tsx
await prisma.user
.findUnique({
where: {email: 'ada@prisma.io' }
})
.posts({
where: {
title: {
}
}
})
contains
endsWith
equals
gt
gte

Insertion multi tables

Uploaded using RayThis Extension

```sql
INSERT INTO `toto` (`id`, `nom`) VALUES (1, 'toto');

INSERT INTO `toto_friends` (`friend_id`, `nom`) VALUES (LAST_INSERT_ID(), 'titi');
```

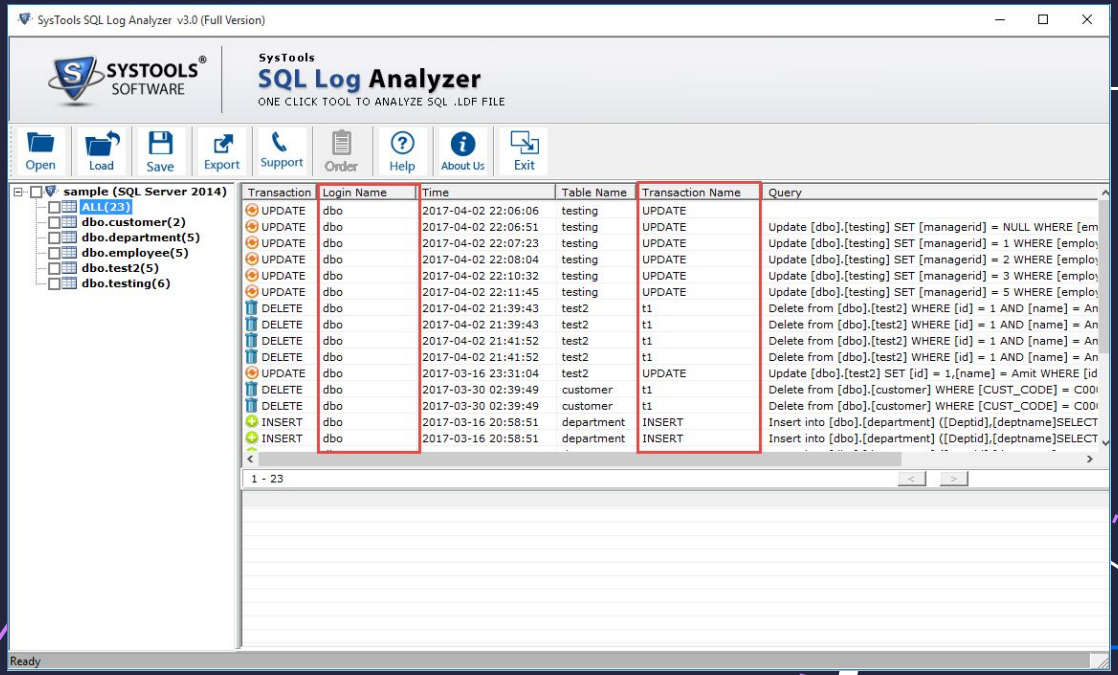
Cube complet / partiel



Les vues avancées

View	Description
Simple View	A view based on the only a single table, which doesn't contain GROUP BY clause and any functions.
Complex View	A view based on multiple tables, which contain GROUP BY clause and functions
Inline view	A view based on a subquery in FROM Clause, that subquery creates a temporary table and simplifies the complex query.
Materialized view	A view that stores the definition as well as data. It creates replicas of data by storing it physically.

Lecture des logs



Fonctions récursives

Uploaded using RayThis Extension

```sql
CREATE TABLE IF NOT EXISTS resultats (
    n INT,
    factoriel BIGINT
);

-- Créez la procédure stockée récursive
DELIMITER //
CREATE PROCEDURE CalculerFactoriel(n INT)
BEGIN
    IF n < 0 THEN -- gestion d'erreur
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Le nombre doit être non négatif';
    ELSEIF n = 0 THEN -- condition d'arrêt
        INSERT INTO resultats (n, factoriel) VALUES (0, 1);
    ELSE
        DECLARE fact BIGINT;
        CALL CalculerFactoriel(n - 1); -- Appel récursif
        SELECT factoriel INTO fact FROM resultats WHERE n = n - 1;
        INSERT INTO resultats (n, factoriel) VALUES (n, fact * n);
    END IF;
END;
//
DELIMITER ;

-- Appelez la procédure pour calculer le factoriel de 5, par exemple
CALL CalculerFactoriel(5);

-- Sélectionnez le résultat
SELECT factoriel FROM resultats WHERE n = 5;
```