

The background is a solid blue color. It is decorated with various abstract geometric elements: a vertical dashed white line on the left; a grid of small purple dots in the top left; concentric purple circles at the top center; a horizontal line with a small white circle at its left end in the top right; a dark blue and purple quarter-circle shape on the right; a horizontal line with a small black circle at its left end below the quarter-circle; a dotted purple circle and a solid white circle on the bottom left; a horizontal purple bar with a black dot at its right end; three small dots (white, purple, purple) at the bottom center; a series of purple diagonal slashes on the bottom right; and two vertical bars (one dark blue, one white) on the far right.

MYSQL 2ITECH



Réactivation mémoire



.....



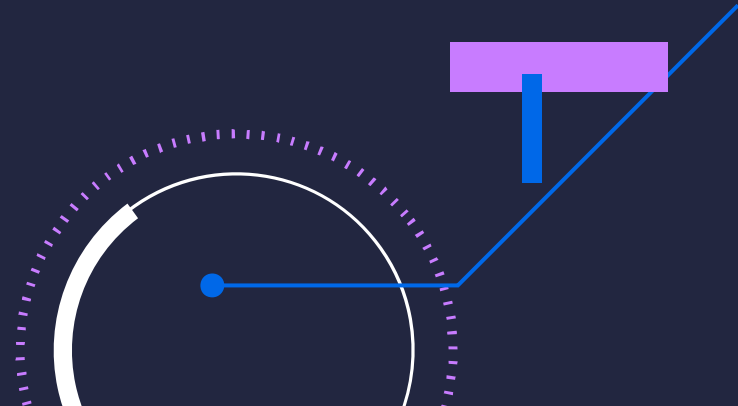
Les composants SQL

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
- Opérateurs



JSON

Uploaded using RayThis Extension








































```
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

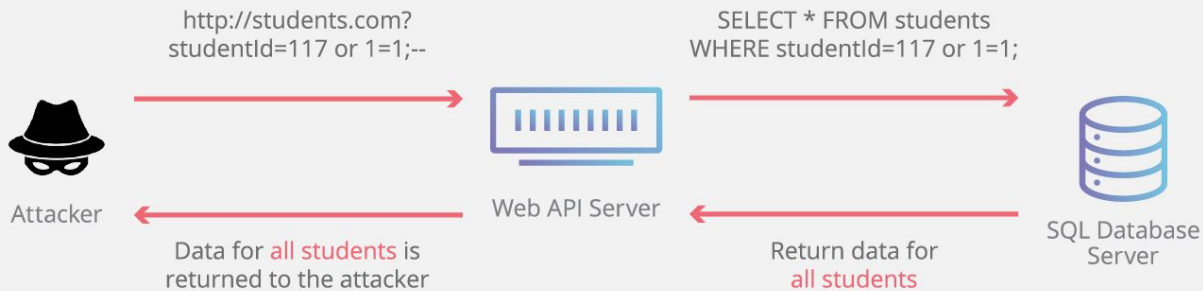
```
CREATE TABLE produits (  
  id INT PRIMARY KEY,  
  nom VARCHAR(255),  
  details JSON  
);  
  
INSERT INTO produits (id, nom, details) VALUES (1, 'Produit A', '{"description": "Un produit  
de qualité", "prix": 99.99, "en_stock": true}');  
  
-- Sélection des détails du produit  
SELECT nom, details->'$.description' AS description, details->'$.prix' AS prix  
FROM produits  
WHERE id = 1;
```

Encodage

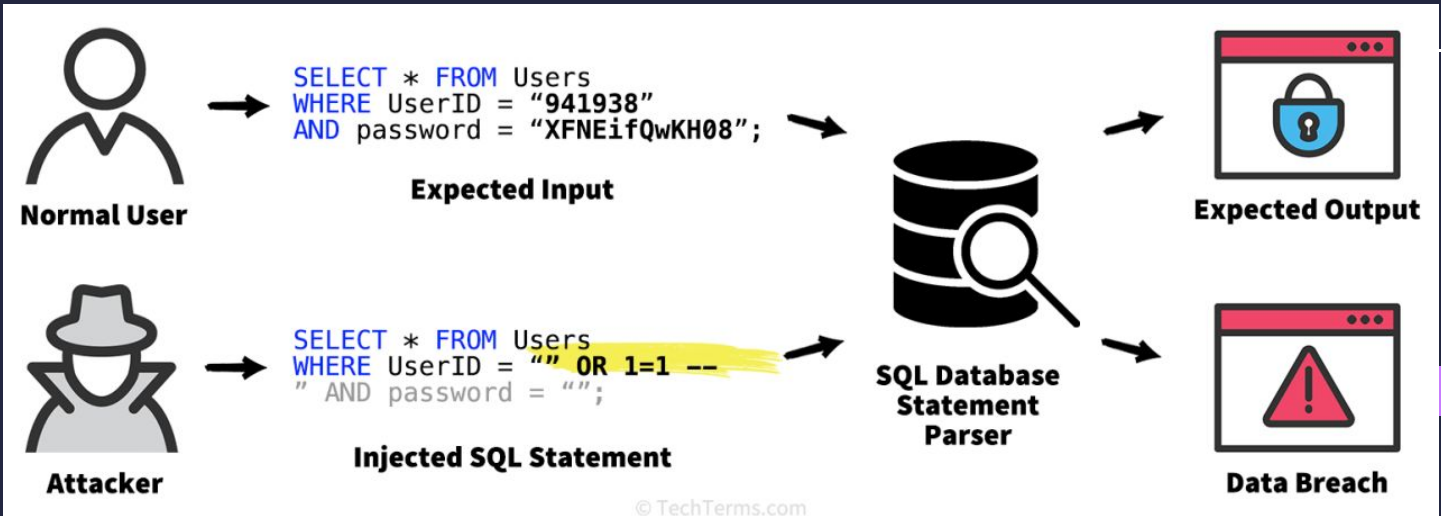
← T →							id	categorie	categorie_mini
<input type="checkbox"/>		Modifier		Copier		Effacer	1	Moniteurs	moniteurs
<input type="checkbox"/>		Modifier		Copier		Effacer	2	Sources vid��o	sources-vid��o
<input type="checkbox"/>		Modifier		Copier		Effacer	3	projection	
<input type="checkbox"/>		Modifier		Copier		Effacer	5	Distribution vid��o	distribution-vidyoo
<input type="checkbox"/>		Modifier		Copier		Effacer	6	selection-video	
<input type="checkbox"/>		Modifier		Copier		Effacer	7	traitement-video	
<input type="checkbox"/>		Modifier		Copier		Effacer	8	prise-de-vue	
<input type="checkbox"/>		Modifier		Copier		Effacer	9	enceintes-et-amplificateurs-dedies	
<input type="checkbox"/>		Modifier		Copier		Effacer	10	mixage	
<input type="checkbox"/>		Modifier		Copier		Effacer	11	traitement-audio	
<input type="checkbox"/>		Modifier		Copier		Effacer	13	micros	
<input type="checkbox"/>		Modifier		Copier		Effacer	14	sources-audios	
<input type="checkbox"/>		Modifier		Copier		Effacer	15	Conf��rence	conf��rence

Injection SQL

SQL Injection



Injection SQL



Les ORM

```
index.tsx

1  await prisma.user
2    .findUnique({
3      where: {email: 'ada@prisma.io' }
4    })
5    .posts({
6      where: {
7        title: {
8          |
9        }
10     }
11   })
```

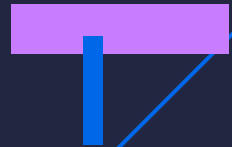
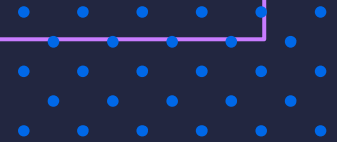
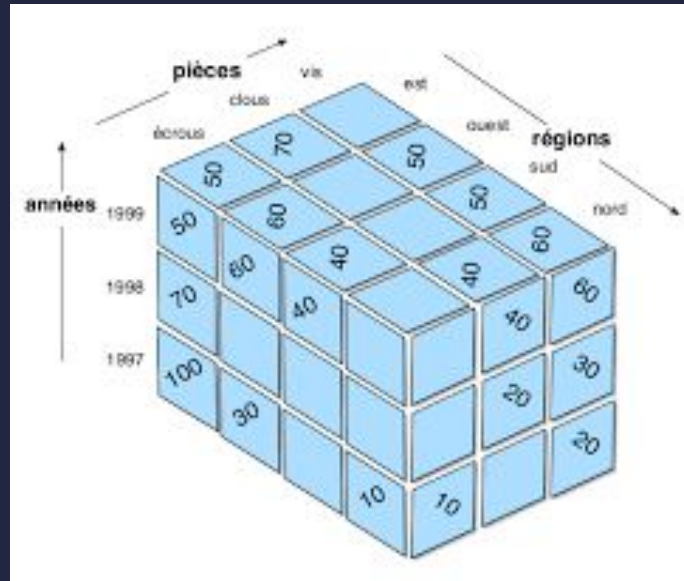
- contains
- endsWith
- equals
- gt
- gte

Insertion multi tables

Uploaded using RayThis Extension

```
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

SysTools SQL Log Analyzer v3.0 (Full Version)

SYSTOOLS® SOFTWARE

SysTools SQL Log Analyzer
ONE CLICK TOOL TO ANALYZE SQL .LDF FILE

Open Load Save Export Support Order Help About Us Exit

sample (SQL Server 2014)

- ALL (23)
- dbo.customer(2)
- dbo.department(5)
- dbo.employee(5)
- dbo.test2(5)
- dbo.testing(6)

Transaction	Login Name	Time	Table Name	Transaction Name	Query
UPDATE	dbo	2017-04-02 22:06:06	testing	UPDATE	Update [dbo].[testing] SET [managerid] = NULL WHERE [em
UPDATE	dbo	2017-04-02 22:06:51	testing	UPDATE	Update [dbo].[testing] SET [managerid] = 1 WHERE [emplo
UPDATE	dbo	2017-04-02 22:07:23	testing	UPDATE	Update [dbo].[testing] SET [managerid] = 2 WHERE [emplo
UPDATE	dbo	2017-04-02 22:08:04	testing	UPDATE	Update [dbo].[testing] SET [managerid] = 3 WHERE [emplo
UPDATE	dbo	2017-04-02 22:10:32	testing	UPDATE	Update [dbo].[testing] SET [managerid] = 5 WHERE [emplo
UPDATE	dbo	2017-04-02 22:11:45	testing	UPDATE	Update [dbo].[testing] SET [managerid] = 5 WHERE [emplo
DELETE	dbo	2017-04-02 21:39:43	test2	t1	Delete from [dbo].[test2] WHERE [id] = 1 AND [name] = An
DELETE	dbo	2017-04-02 21:39:43	test2	t1	Delete from [dbo].[test2] WHERE [id] = 1 AND [name] = An
DELETE	dbo	2017-04-02 21:41:52	test2	t1	Delete from [dbo].[test2] WHERE [id] = 1 AND [name] = An
DELETE	dbo	2017-04-02 21:41:52	test2	t1	Delete from [dbo].[test2] WHERE [id] = 1 AND [name] = An
UPDATE	dbo	2017-03-16 23:31:04	test2	UPDATE	Update [dbo].[test2] SET [id] = 1,[name] = Amit WHERE [id
DELETE	dbo	2017-03-30 02:39:49	customer	t1	Delete from [dbo].[customer] WHERE [CUST_CODE] = C001
DELETE	dbo	2017-03-30 02:39:49	customer	t1	Delete from [dbo].[customer] WHERE [CUST_CODE] = C001
INSERT	dbo	2017-03-16 20:58:51	department	INSERT	Insert into [dbo].[department] ([Deptid],[deptname])SELECT
INSERT	dbo	2017-03-16 20:58:51	department	INSERT	Insert into [dbo].[department] ([Deptid],[deptname])SELECT

1 - 23

Ready

Fonctions récurrentes

```
Upload using RayThis Extension

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

-- Créez la procédure stockée récurrente
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 récurrent
    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;
```