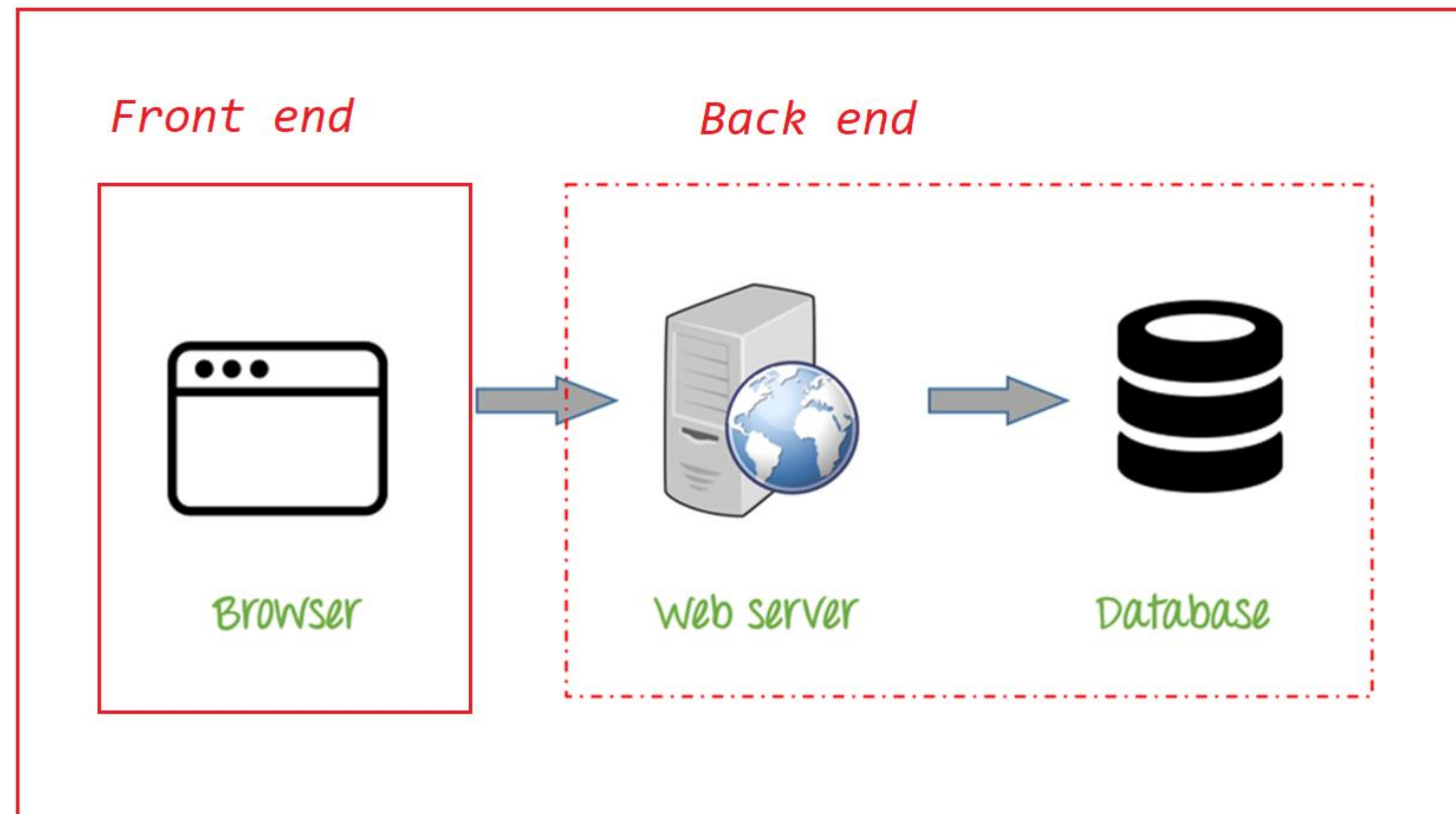


# DataBase

- ❑ 1. Giới thiệu hệ quản trị CSDL MySQL
- ❑ 2. Các câu lệnh hay sử dụng
- ❑ 3. Procedure
- ❑ 4. Trigger
- ❑ 5. Demo







SALES				
<u>purchase_number</u>	<u>date_of_purchase</u>	<u>customer_id</u>	<u>item_code</u>	
1	03/09/2016	1	A_1	
2	02/12/2016	2	C_1	
3	15/04/2017	3	D_1	
4	24/05/2017	1	B_2	
5	25/05/2017	4	B_2	
6	06/06/2017	2	B_1	
7	10/06/2017	4	A_2	
8	13/06/2017	3	C_1	
9	20/07/2017	1	A_1	
10	11/08/2017	2	B_1	

**field** = a column in a table containing specific information about every record in the table

SALES				
purchase_number		date_of_purchase	customer_id	item_code
1		03/09/2016	1	A_1
2		02/12/2016	2	C_1
3		15/04/2017	3	D_1
4		24/05/2017	1	B_2
5		25/05/2017	4	B_2
6		06/06/2017	2	B_1
7		10/06/2017	4	A_2
8		13/06/2017	3	C_1
9		20/07/2017	1	A_1
10		11/08/2017	2	B_1



record

SALES				
purchase_number	date_of_purchase	customer_id	item_code	
1	03/09/2016	1	A_1	
2	02/12/2016	2	C_1	
3	15/04/2017	3	D_1	
4	24/05/2017	1	B_2	
5	25/05/2017	4	B_2	
6	06/06/2017	2	B_1	
7	10/06/2017	4	A_2	
8	13/06/2017	3	C_1	
9	20/07/2017	1	A_1	
10	11/08/2017	2	B_1	

➤ Create Database

```
CREATE DATABASE database_name;
```

➤ Create table

```
CREATE TABLE table_name (  
    name_column_1 datatype,  
    name_column_2 datatype,  
);
```



- Number: INT
- Characters: VARCHAR(50)
- Date: DATE
- ENUM

➤ Kiểu biểu diễn số nguyên

<u>numeric data type</u>	<u>size</u> (bytes)	<u>minimum value</u> (signed/unsigned)	<u>maximum value</u> (signed/unsigned)
TINYINT	1	-128	127
		0	255
SMALLINT	2	-32,768	32,767
		0	65,535
MEDIUMINT	3	-8,388,608	8,388,607
		0	16,777,215
INT	4	-2,147,483,648	2,147,483,647
		0	4,294,967,295
BIGINT	8	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
		0	18,446,744,073,709,551,615

- Kiểu biểu diễn số thực:
  - Float (4 bytes)
  - Double (8 bytes)



➤ Kiểu chuỗi:

So sánh CHAR và VARCHAR		
	CHAR	VARCHAR
Viết tắt	Character (Ký tự)	Variable Character (nghĩa là chưa biết length của Varchar)
Chiều dài tối đa	Lưu tối đa 255 bytes	Lưu tối đa 65,535 bytes
Length	Fix cứng size lưu trữ trong ổ cứng máy tính  VD: khai báo CHAR(10) thì 'Nguyen' chiếm 10 ô trong ổ cứng máy tính 'Nguyen Van' cũng chiếm 10 ô trong ổ cứng máy tính	Length của Varchar phụ thuộc vào giá trị  VD: khai báo VARCHAR(10) thì 'Nguyen' chiếm 6 ô trong ổ cứng máy tính 'Nguyen Van' chiếm 10 ô trong ổ cứng máy tính
Tốc độ	Nhanh hơn	Chậm hơn

- Kiểu ENUM:
  - ENUM ('Male', 'Female')

- Primary Key
- Foreign Key
- Unique Key
- Default
- Not Null
- Check



(8) SELECT *column\_name\_1, column\_name\_2, ...*

(1) FROM *table\_name*

(2) [JOIN]

(3) [WHERE]

(4) [GROUP BY]

(5) [HAVING]

(6) [ORDER BY]

(7) [LIMIT]

- Group By: Sử dụng để nhóm các dữ liệu tương đồng.
- Having: Sử dụng để giới hạn lại dữ liệu sau khi nhóm.

- COUNT()
- COUNT(field\_name), COUNT(DISTINCT field\_name), COUNT(\*), COUNT(1)
- SUM(),
- MIN(),
- MAX(),
- AVG()



- Method 1: Insert each record

```
INSERT INTO table_name (column1, column2, column3, ...)
```

```
VALUES (value1, value2, value3, ...);
```

- Method 2: Insert multiple records at once (recommended)

```
INSERT INTO table_name (column1, column2, column3, ...)
```

```
VALUES (value1.1, value1.2, value1.3, ...),
```

```
(value2.1, value2.2, value2.3, ...),
```

```
(value3.1, value3.2, value3.3, ...);
```

➤ Syntax:

```
UPDATE table_name
```

```
SET      column_name_1 = value_1,
```

```
        column_name_2 = value_2,
```

```
        ...
```

```
WHERE condition;
```

➤ Syntax:

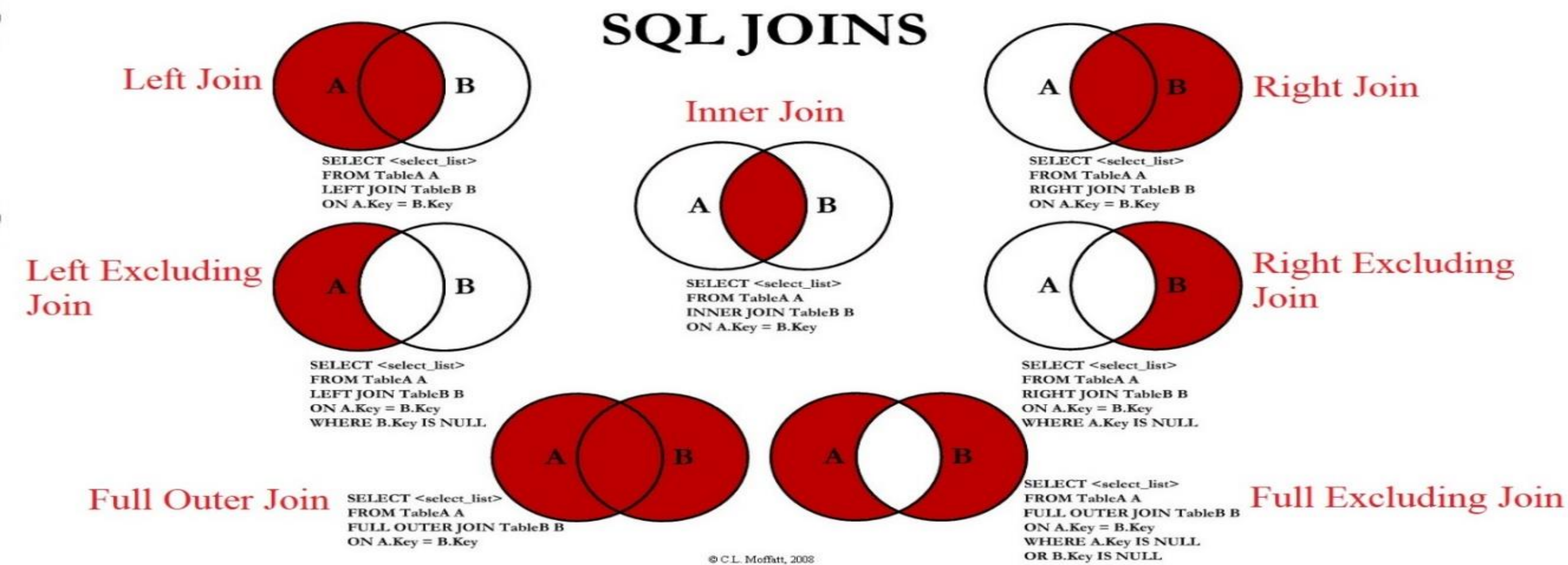
DELETE

FROM    table\_name

WHERE   condition;



- INNER JOIN (JOIN)
- LEFT JOIN
- RIGHT JOIN
  
- LEFT EXCLUDING JOIN
- RIGHT EXCLUDING JOIN
- FULL OUTER JOIN
- FULL EXCLUDING JOIN



```
SELECT      column_name
FROM        table_name
WHERE       EXISTS
            (SELECT      column_name
              FROM        table_name
              WHERE       condition
              ....);
```

```
WITH CTE_name (column_name_1, column_name_2, ...) AS (  
    CTE_query  
)
```

```
-- Statement using the CTE
```

```
SELECT | INSERT | DELETE | UPDATE | CREATE VIEW ...  
FROM    CTE_name;  
WHERE   condition
```

➤ Create

```
DELIMITER $$      (hoặc DELIMITER //)
CREATE PROCEDURE procedure_name (IN in_param_name_1 Datatype, OUT out_param_name_2 Datatype)
BEGIN
    SELECT ...;
END$$
DELIMITER ;
```

➤ Call

```
CALL procedure_name ();
```

➤ Drop

```
DROP PROCEDURE IF EXISTS procedure_name ();
```



## ➤ Create

```
DROP TRIGGER IF EXISTS trigger_name;
DELIMITER $$

CREATE TRIGGER trigger_name()
BEFORE/AFTER          INSERT/UPDATE/DELETE ON name_table
FOR EACH ROW
BEGIN
    -- logic
    SELECT ...;
END $$
DELIMITER ;
```

## ➤ Disable / Enable

```
DISABLE/ENABLE Trigger_name
```

## ➤ Drop

```
DROP TRIGGER IF EXISTS trigger_name;
```

Q & A