

**Integridad:** Daño por modificaciones

**Disponibilidad:** Cuidar los objetos

**Confidencialidad:** Accesos autorizados

**Session Hijacking:** Consigue identificador de sesión activa

- **Fuerza bruta:** [Contra larga]
- **Sniffing:** [Usar SSL] Sniffer en la red
- **Propagación URL:** [No pasar info en URL]

- **RServidor compartido:** [Aislar]
- **XSS:** [noXSS.js] Roba cuck
- **Inject SQL:** [Sanitizar]

```
DELIMITER //
```

```
CREATE [OR REPLACE]
```

```
    TRIGGER [IF NOT EXISTS] <name> <when>
```

```
<event>
```

```
    ON tbl_name FOR EACH ROW BEGIN
```

```
        -- Contenido
```

```
    END; //
```

```
DELIMITER ;
```

```
DELIMITER //
```

```
CREATE TRIGGER log_uupdate BEFORE INSERT
```

```
    ON users FOR EACH ROW BEGIN
```

```
        -- Contenido
```

```
    END; //
```

```
DELIMITER ;
```

```
DROP TABLE IF EXISTS log_test;  
CREATE TABLE IF NOT EXISTS log_test(  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    hora TIMESTAMP DEFAULT(NOW()),  
    mensaje VARCHAR(100)  
)
```

```
INSERT [IGNORE] [INTO] tbl_name [PARTITION  
(partition_list)]  
{VALUES | VALUE} ({expr | DEFAULT},...),  
(...),...
```



```
DELIMITER //
CREATE TRIGGER log_change AFTER UPDATE
  ON employees FOR EACH ROW
  BEGIN
    INSERT INTO log_test (mensaje) VALUE ('');
  END; //
DELIMITER ;
```

```
DELIMITER //
```

```
CREATE PROCEDURE simpleproc (OUT param1 INT)
```

```
  BEGIN
```

```
    SELECT COUNT(*) INTO param1 FROM t;
```

```
  END; //
```

```
DELIMITER ;
```

  

```
-- CALL simpleproc(@a);
```