

STRUCTURED QUERY LANGUAGE (SQL) — ADVANCED DataBase Foundations

Author: Eng. Carlos Andrés Sierra, M.Sc.
cavirguezs@udistrital.edu.co

Lecturer
Computer Engineer
School of Engineering
Universidad Distrital Francisco José de Caldas

2024-III



UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Outline

- 1 Store Procedures
- 2 Triggers
- 3 Performance



Outline

1 Store Procedures

2 Triggers

3 Performance



Store Procedures

Store Procedures are a **set** of **SQL statements** that are **stored** in the **database** and can be **executed** by **calling** the **procedure** name.

PostgreSQL Example

```
CREATE OR REPLACE FUNCTION myFunction()  
RETURNS SETOF myTable AS $$  
BEGIN  
    RETURN QUERY SELECT * FROM myTable;  
END;  
$$ LANGUAGE plpgsql;
```



Store Procedures

Store Procedures are a **set** of **SQL statements** that are **stored** in the **database** and can be **executed** by **calling** the **procedure** name.

MySQL Example

```
DELIMITER $  
CREATE PROCEDURE myProcedure()  
BEGIN  
    SELECT * FROM myTable;  
END$$  
DELIMITER ;
```



Store Procedures

Store Procedures can be **executed** by **calling** the **procedure** name.

PostgreSQL Example

```
SELECT * FROM myFunction ();
```

MySQL Example

```
CALL myProcedure ();
```



Store Procedures

Store Procedures can be **deleted** from the **database**.

PostgreSQL Example

```
DROP FUNCTION myFunction ();
```

MySQL Example

```
DROP PROCEDURE myProcedure ;
```



Outline

1 Store Procedures

2 Triggers

3 Performance



Triggers

Triggers are a **set** of **SQL statements** that are **executed automatically** when a **specified event** occurs in a **database**.

PostgreSQL Example

```
CREATE OR REPLACE FUNCTION my_trigger_function()  
RETURNS TRIGGER AS $$  
BEGIN  
    INSERT INTO myLog VALUES (NEW.id , NEW.name);  
    RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;  
  
CREATE TRIGGER myTrigger  
AFTER INSERT ON myTable  
FOR EACH ROW  
EXECUTE FUNCTION my_trigger_function();
```

Triggers

Triggers are a **set** of **SQL statements** that are **executed automatically** when a **specified event** occurs in a **database**.

MySQL Example

```
DELIMITER $  
CREATE TRIGGER myTrigger  
AFTER INSERT ON myTable  
FOR EACH ROW  
BEGIN  
    INSERT INTO myLog VALUES (NEW.id , NEW.name);  
END$$  
DELIMITER ;
```



Triggers

Triggers can be **deleted** from the **database**.

PostgreSQL Example

```
DROP TRIGGER myTrigger ON myTable;
```

MySQL Example

```
DROP TRIGGER myTrigger;
```



Outline

1 Store Procedures

2 Triggers

3 Performance



Indexes

Indexes are **data structures** that are used to **speed up** the **retrieval** of **data** from a **database**.

PostgreSQL Example — MySQL Example

```
CREATE INDEX myIndex ON myTable (name);
```



Views

Views are **virtual tables** that are **created** by **querying** one or more **tables** in a **database**.

PostgreSQL Example — MySQL Example

```
CREATE VIEW myView AS  
SELECT * FROM myTable WHERE country = 'USA';
```



Nested Queries

Nested Queries are **queries** that are **embedded** within **other queries**.

PostgreSQL Example — MySQL Example

```
SELECT * FROM myTable WHERE id IN  
  (SELECT id FROM myTable WHERE country = 'USA');
```



Outline

- 1 Store Procedures
- 2 Triggers
- 3 Performance



Thanks!

Questions?



Repo: <https://github.com/EngAndres/ud-public/tree/main/courses/databases-foundations>

