

STRUCTURED QUERY LANGUAGE (SQL) — ADVANCED DataBase Foundations

Author: Eng. Carlos Andrés Sierra, M.Sc.
`carlos.andres.sierra.v@gmail.com`

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

2024-I



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 PROCEDURE myProcedure AS  
BEGIN  
    SELECT * FROM myTable;  
END;
```



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

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



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

```
CALL myProcedure ();
```

MySQL Example

```
CALL myProcedure ();
```



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

```
DROP PROCEDURE myProcedure ;
```

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



Triggers

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

MySQL Example

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



Triggers

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

PostgreSQL Example

```
DROP TRIGGER myTrigger;
```

MySQL Example

```
DROP TRIGGER myTrigger;
```



Outline

- 1 Store Procedures
- 2 Triggers
- 3 Performance



Views

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

PostgreSQL Example

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

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

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

MySQL Example

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



Indexes

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

PostgreSQL Example

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

MySQL Example

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



Outline

- 1 Store Procedures
- 2 Triggers
- 3 Performance



Thanks!

Questions?



Repo: github.com/engandres/ud-public/courses/databases-foundations

