

Bases de Datos

Clase 8: Lógica en la BDD

QUÉ VAMOS A VER HOY



- . VISTAS
- . TRIGGERS
- . STORE PROCEDURES

Lógica en la BDD

- En desarrollo de software le llamamos *business logic*, lógica del negocio o simplemente lógica a todas las reglas, algoritmos, etc. que definen el sistema y que están dados por lo que quiere lograr la aplicación.
- Los motores relacionales más complejos como Postgres o MySQL tienen funcionalidades que permiten implementar parte de esa lógica directamente en la BDD.

Vistas

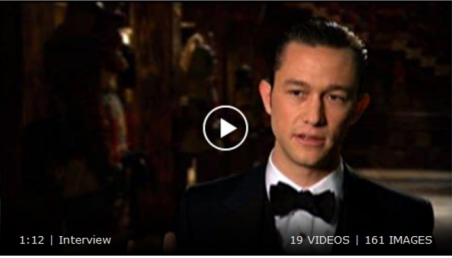

Motivación

 Menu All ▾ Search IMDb  IMDbPro Watchlist

Christopher Nolan (1)


Top 500

Writer | Producer | Director

1:12 | Interview19 VIDEOS | 161 IMAGES

Best known for his cerebral, often nonlinear, storytelling, acclaimed writer-director [Christopher Nolan](#) was born on July 30, 1970, in London, England. Over the course of 15 years of filmmaking, Nolan has gone from low-budget independent films to working on some of the biggest blockbusters ever made. At 7 years old, Nolan began making short movies ... [See full bio »](#)

Born: July 30, 1970 in London, England, UK

More at IMDbPro »
 **Contact Info:** View agent, publicist, legal on IMDbPro

Nominated for 5 Oscars. Another 139 wins & 227 nominations. [See more awards »](#)




Quick Links

Biography
Awards
Photo Gallery

Filmography (by Job)
Trailers and Videos




[Explore More](#)

Reboots & Remakes We Can't Wait to See



From "Dexter" to *The Suicide Squad*, here are our picks for the reboots and remakes we're most excited for in 2021 and beyond.

[Browse our picks »](#)

Share this page:   

Motivación

The image is a screenshot of the IMDb website, showing two overlapping views. The top view is the profile page for Christopher Nolan, and the bottom view is the movie page for 'The Prestige' (2006).

Christopher Nolan Profile:

- Name:** Christopher Nolan (1)
- Roles:** Writer | Producer | Director
- Top 500:** Indicated by a red ribbon icon.
- Quick Links:** Biography, Awards, Photo Gallery, Filmography (by Job), Trailers and Videos.
- Explore More:** A button to explore more content.
- Reboots & Remakes We Can't Wait to See:** A section featuring movie posters for 'Inception', 'The Dark Knight', and 'The Matrix'.
- Video:** A video player showing an interview with Christopher Nolan, with a play button icon and a duration of 1:12.
- Statistics:** 19 VIDEOS | 161 IMAGES.
- Description:** Best known for his cerebral, often nonlinear, storytelling, acclaimed writer-director Christopher Nolan was born on July 30, 1970, in London, England. Over the course of 15 years of filmmaking, Nolan has directed some of the biggest blockbusters of the 21st century.
- More at IMDb:** A link to more information.
- Contact:** A link to contact the user.
- Nominated:** A yellow badge indicating nominations.

The Prestige Movie Page:

- Header:** FULL CAST AND CREW | TRIVIA | USER REVIEWS | IMDbPro | MORE | SHARE
- Title:** The Prestige (2006)
- Rating:** 8.5/10 (1,207,788 votes)
- Genre:** PG-13 | 2h 10min | Drama, Mystery, Sci-Fi
- Release Date:** 20 October 2006 (USA)

Motivación

The image is a screenshot of the IMDb website, overlaid with a semi-transparent version of the same site to show a transition or comparison. The top section shows the IMDb homepage with a search bar and navigation links. Below this, the profile of Christopher Nolan is displayed, featuring a photo, a video player, and a brief biography. The bottom section shows the movie page for 'The Prestige' (2006), including the cast and crew, a rating of 8.5/10, and a 'Rate This' button.

Christopher Nolan (1)
Writer | Producer | Director

Best known for his cerebral, often nonlinear, storytelling, acclaimed writer-director [Christopher Nolan](#) was born on July 30, 1970, in London, England. Over the course of 15 years of filmmaking, Nolan has directed some of the biggest blockbusters of the decade. [See full bio »](#)

Born: July 30, 1970

[More at IMDb](#)
[Contact Info](#)

Quick Links

- [Biography](#)
- [Awards](#)
- [Photo Gallery](#)
- [Filmography \(by Job\)](#)
- [Trailers and Videos](#)

[Explore More](#)

Reboots & Remakes We Can't Wait to See

[The Matrix](#) [The Dark Knight](#) [The Dark Knight Rises](#)

IMDb [Menu](#) [All](#)

Get a sneak peek of the new version of this page. [Check it out now](#)

FULL CAST AND CREW | **TRIVIA** | **USER REVIEWS** | **IMDbPro** | **MORE** | **SHARE**

The Prestige (2006)
PG-13 | 2h 10min | Drama, Mystery, Sci-Fi | 20 October 2006 (USA)

8.5 / 10
1,207,788

[Rate This](#)

Datos

Directores

<u>nombre</u>	<u>país</u>	<u>retirado</u>
Christopher Nolan	Inglaterra	false
Clint Eastwood	EE.UU	false
Ingmar Bergman	Suecia	true
John Favreau	EE.UU	false

Películas

<u>nombre</u>	<u>director</u>	<u>año</u>
The Prestige	Christopher Nolan	2006
Interstellar	Christopher Nolan	2014
Gran Torino	Clint Eastwood	2008
The Prestige	John Favreau	2021

Evaluaciones

<u>pelicula</u>	<u>director</u>	<u>fuelle</u>	<u>eval</u>
Gran Torino	Clint Eastwood	The Guardian	8.1
Gran Torino	Clint Eastwood	Rolling Stone	8.7
The Prestige	Christopher Nolan	The Observer	8.3
The Prestige	Christopher Nolan	Uncut	8.5
The Prestige	Christopher Nolan	Rolling Stone	8.7
Interstellar	Christopher Nolan	Rolling Stone	8.2
The Prestige	John Favreau	Uncut	9.8

Consulta Frecuente

Score promedio de la película "The Prestige" de "Christopher Nolan"

```
SELECT AVG(eval) AS promedio  
FROM Evaluaciones  
WHERE nombre = 'The Prestige' AND  
       director = 'Christopher Nolan'
```

promedio
8.5

Datos

Directores

<u>nombre</u>	<u>país</u>	<u>retirado</u>
Christopher Nolan	Inglaterra	false
Clint Eastwood	EE.UU	false
Ingmar Bergman	Suecia	true
John Favreau	EE.UU	false

Películas

<u>nombre</u>	<u>director</u>	<u>año</u>
The Prestige	Christopher Nolan	2006
Interstellar	Christopher Nolan	2014
Gran Torino	Clint Eastwood	2008
The Prestige	John Favreau	2021

Evaluaciones

<u>pelicula</u>	<u>director</u>	<u>fuelle</u>	<u>eval</u>
Gran Torino	Clint Eastwood	The Guardian	8.1
Gran Torino	Clint Eastwood	Rolling Stone	8.7
The Prestige	Christopher Nolan	The Observer	8.3
The Prestige	Christopher Nolan	Uncut	8.5
The Prestige	Christopher Nolan	Rolling Stone	8.7
Interstellar	Christopher Nolan	Rolling Stone	8.2
The Prestige	John Favreau	Uncut	9.8

Datos

Directores

<u>nombre</u>	<u>país</u>	<u>retirado</u>
Christopher Nolan	Inglaterra	false
Clint Eastwood	EE.UU	false
Ingmar Bergman	Suecia	true
John Favreau	EE.UU	false

Películas

<u>nombre</u>	<u>director</u>	<u>año</u>	<u>promedio</u>
The Prestige	Christopher Nolan	2006	8.5
Interstellar	Christopher Nolan	2014	8.2
Gran Torino	Clint Eastwood	2008	8.4
The Prestige	John Favreau	2021	9.8

Evaluaciones

<u>pelicula</u>	<u>director</u>	<u>fuelle</u>	<u>eval</u>
Gran Torino	Clint Eastwood	The Guardian	8.1
Gran Torino	Clint Eastwood	Rolling Stone	8.7
The Prestige	Christopher Nolan	The Observer	8.3
The Prestige	Christopher Nolan	Uncut	8.5
The Prestige	Christopher Nolan	Rolling Stone	8.7
Interstellar	Christopher Nolan	Rolling Stone	8.2
The Prestige	John Favreau	Uncut	9.8

Datos

Directores

<u>nombre</u>	<u>país</u>	<u>retirado</u>
Christopher Nolan	Inglaterra	false
Clint Eastwood	EE.UU	false
Ingmar Bergman	Suecia	true
John Favreau	EE.UU	false

Películas

<u>nombre</u>	<u>director</u>	<u>año</u>	<u>promedio</u>
The Prestige	Christopher Nolan	2006	8.5
Interstellar	Christopher Nolan	2014	8.2
Gran Torino	Clint Eastwood	2008	8.4
The Prestige	John Favreau	2021	9.8

Evaluaciones

<u>pelicula</u>	<u>director</u>	<u>fuelle</u>	<u>eval</u>
Gran Torino	Clint Eastwood	The Guardian	8.1
Gran Torino	Clint Eastwood	Rolling Stone	8.7
The Prestige	Christopher Nolan	The Observer	8.3
The Prestige	Christopher Nolan	Uncut	8.5
The Prestige	Christopher Nolan	Rolling Stone	8.7
The Prestige	Christopher Nolan	The Guardian	8.7
Interstellar	Christopher Nolan	Rolling Stone	8.2
The Prestige	John Favreau	Uncut	9.8

Datos

Directores

<u>nombre</u>	<u>país</u>	<u>retirado</u>
Christopher Nolan	Inglaterra	false
Clint Eastwood	EE.UU	false
Ingmar Bergman	Suecia	true
John Favreau	EE.UU	false

Películas

<u>nombre</u>	<u>director</u>	<u>año</u>	<u>promedio</u>
The Prestige	Christopher Nolan	2006	8.6
Interstellar	Christopher Nolan	2014	8.2
Gran Torino	Clint Eastwood	2008	8.4
The Prestige	John Favreau	2021	9.8

Evaluaciones

<u>pelicula</u>	<u>director</u>	<u>fuelle</u>	<u>eval</u>
Gran Torino	Clint Eastwood	The Guardian	8.1
Gran Torino	Clint Eastwood	Rolling Stone	8.7
The Prestige	Christopher Nolan	The Observer	8.3
The Prestige	Christopher Nolan	Uncut	8.5
The Prestige	Christopher Nolan	Rolling Stone	8.7
The Prestige	Christopher Nolan	The Guardian	8.7
Interstellar	Christopher Nolan	Rolling Stone	8.2
The Prestige	John Favreau	Uncut	9.8

Vistas

CREATE VIEW PelEval AS

SELECT pelicula, director,

AVG(eval) AS promedio

FROM Evaluaciones

GROUP BY pelicula, director

PelEval

película	director	promedio
Interstellar	Christopher Nolan	8.4
Gran Torino	Clint Eastwood	8.2
The Prestige	Christopher Nolan	8.5
The Prestige	John Favreau	9.8

Vistas

La vista define una tabla "virtual"

PelEval

película	director	promedio
Interstellar	Christopher Nolan	8.4
Gran Torino	Clint Eastwood	8.2
The Prestige	Christopher Nolan	8.5
The Prestige	John Favreau	9.8

SELECT promedio

FROM PelEval

WHERE película = 'The Prestige' AND
director = 'Christopher
Nolan'



promedio
8.5

¿Cómo funcionan las vistas?

PelEval

película	director	promedio
Interstellar	Christopher Nolan	8.4
Gran Torino	Clint Eastwood	8.2
The Prestige	Christopher Nolan	8.5
The Prestige	John Favreau	9.8

```
CREATE VIEW PelEval AS
SELECT película, director,
       AVG(eval) AS promedio
FROM Evaluaciones
GROUP BY película, director

SELECT promedio
FROM PelEval
WHERE película = 'The Prestige' AND
       director = 'Christopher Nolan'
```

=

```
SELECT promedio
FROM (
  SELECT película, director,
         AVG(eval) AS promedio
  FROM Evaluaciones
  GROUP BY película, director )
WHERE película = 'The Prestige' AND
       director = 'Christopher Nolan'
```


¿Cómo funcionan las vistas?

Las vistas no son tablas físicas!

Cuando consultamos una vista:

1. Se reescribe la consulta reemplazando la vista por la consulta original en el **FROM**.
2. Se ejecuta la consulta sobre las **tablas originales**.

¿Cómo funcionan las vistas?

En la práctica solo estamos guardando una consulta frecuente en el sistema para reutilizarla después:

- No estamos guardando una tabla
- No se crea una tabla en el disco
- Se trabaja con las tablas base
- Si se actualizan los datos no hay problemas

Vistas y actualización

Evaluaciones

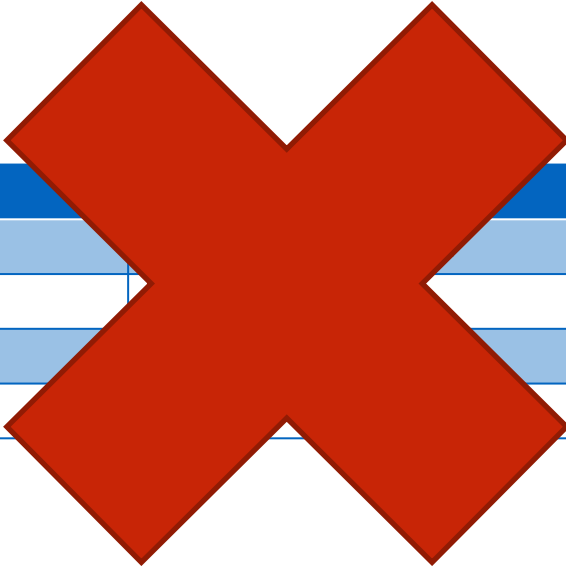
<u>pelicula</u>	<u>director</u>	<u>fuelle</u>	eval
Gran Torino	Clint Eastwood	The Guardian	8.1
Gran Torino	Clint Eastwood	Rolling Stone	8.7
The Prestige	Christopher Nolan	The Observer	8.3
The Prestige	Christopher Nolan	Uncut	8.5
The Prestige	Christopher Nolan	Rolling Stone	8.7
The Prestige	Christopher Nolan	The Guardian	8.7
Interstellar	Christopher Nolan	Rolling Stone	8.2
The Prestige	John Favreau	Uncut	9.8

PelEval

película	director	promedio
Interstellar	Christopher Nolan	8.4
Gran Torino	Clint Eastwood	8.2
The Prestige	Christopher Nolan	8.6
The Prestige	John Favreau	9.8

Eliminando Vistas

PelEval



película	promedio
Interstellar	8.4
Gran Torino	8.2
The Prestige	8.6
The Prestige	9.8

DROP VIEW PelEval

SELECT *
FROM PelEval

Error: No such table PelEval

¿Para qué sirven las vistas?

Abstracción:

- Reducir la complejidad de consultas grandes.
- Evitar repetición de consultas frecuentes.

Mantenibilidad:

- Más fácil de manejar que la gestión de datos duplicados o redundantes.
- Más fácil de optimizar y mantener que una consulta repetida.
- Es más lento que tener las tablas de verdad!

Vistas materializadas

```
CREATE MATERIALIZED VIEW PelEval AS  
  SELECT pelicula, director,  
         AVG(eval) AS promedio  
  FROM Evaluaciones  
  GROUP BY pelicula, director
```

PelEval

película	director	promedio
Interstellar	Christopher Nolan	8.4
Gran Torino	Clint Eastwood	8.2
The Prestige	Christopher Nolan	8.5
The Prestige	John Favreau	9.8

Vistas materializadas

PelEval

película	director	promedio
Interstellar	Christopher Nolan	8.4
Gran Torino	Clint Eastwood	8.2
The Prestige	Christopher Nolan	8.5
The Prestige	John Favreau	9.8

```
SELECT promedio  
FROM PelEval  
WHERE pelicula = 'The Prestige' AND  
director = 'Christopher Nolan'
```



promedio
8.5

Vistas materializadas

Evaluaciones

<u>pelicula</u>	<u>director</u>	<u>fuelle</u>	eval
Gran Torino	Clint Eastwood	The Guardian	8.1
Gran Torino	Clint Eastwood	Rolling Stone	8.7
The Prestige	Christopher Nolan	The Observer	8.3
The Prestige	Christopher Nolan	Uncut	8.5
The Prestige	Christopher Nolan	Rolling Stone	8.7
The Prestige	Christopher Nolan	The Guardian	8.7
Interstellar	Christopher Nolan	Rolling Stone	8.2
The Prestige	John Favreau	Uncut	9.8

Vistas materializadas

Evaluaciones

<u>película</u>	<u>director</u>	<u>fuelle</u>	eval
Gran Torino	Clint Eastwood	The Guardian	8.1
Gran Torino	Clint Eastwood	Rolling Stone	8.7
The Prestige	Christopher Nolan	The Observer	8.3
The Prestige	Christopher Nolan	Uncut	8.5
The Prestige	Christopher Nolan	Rolling Stone	8.7
The Prestige	Christopher Nolan	The Guardian	8.7
Interstellar	Christopher Nolan	Rolling Stone	8.2
The Prestige	John Favreau	Uncut	9.8

PelEval

película	director	promedio
Interstellar	Christopher Nolan	8.4
Gran Torino	Clint Eastwood	8.2
The Prestige	Christopher Nolan	8.5
The Prestige	John Favreau	9.8

Vistas materializadas

Evaluaciones

<u>pelicula</u>	<u>director</u>	<u>fuelle</u>	eval
Gran Torino	Clint Eastwood	The Guardian	8.1
Gran Torino	Clint Eastwood	Rolling Stone	8.7
The Prestige	Christopher Nolan	The Observer	8.3
The Prestige	Christopher Nolan	Uncut	8.5
The Prestige	Christopher Nolan	Rolling Stone	8.7
The Prestige	Christopher Nolan	The Guardian	8.7
Interstellar	Christopher Nolan	Rolling Stone	8.2
The Prestige	John Favreau	Uncut	9.8

PelEval

película	director	promedio
Interstellar	Christopher Nolan	8.4
Gran Torino	Clint Eastwood	8.2
The Prestige	Christopher Nolan	8.5
The Prestige	John Favreau	9.8

SELECT promedio

FROM PelEval

WHERE pelicula = 'The Prestige' AND

director = 'Christopher Nolan'



promedio
8.5

Vistas materializadas

Evaluaciones

<u>película</u>	<u>director</u>	<u>fuelle</u>	eval
Gran Torino	Clint Eastwood	The Guardian	8.1
Gran Torino	Clint Eastwood	Rolling Stone	8.7
The Prestige	Christopher Nolan	The Observer	8.3
The Prestige	Christopher Nolan	Uncut	8.5
The Prestige	Christopher Nolan	Rolling Stone	8.7
The Prestige	Christopher Nolan	The Guardian	8.7
Interstellar	Christopher Nolan	Rolling Stone	8.2
The Prestige	John Favreau	Uncut	9.8

PelEval

película	director	promedio
Interstellar	Christopher Nolan	8.4
Gran Torino	Clint Eastwood	8.2
The Prestige	Christopher Nolan	8.5
The Prestige	John Favreau	9.8

REFRESH MATERIALIZED VIEW PelEval

Vistas materializadas

Evaluaciones

<u>pelicula</u>	<u>director</u>	<u>fuelle</u>	eval
Gran Torino	Clint Eastwood	The Guardian	8.1
Gran Torino	Clint Eastwood	Rolling Stone	8.7
The Prestige	Christopher Nolan	The Observer	8.3
The Prestige	Christopher Nolan	Uncut	8.5
The Prestige	Christopher Nolan	Rolling Stone	8.7
The Prestige	Christopher Nolan	The Guardian	8.7
Interstellar	Christopher Nolan	Rolling Stone	8.2
The Prestige	John Favreau	Uncut	9.8

PelEval

película	director	promedio
Interstellar	Christopher Nolan	8.4
Gran Torino	Clint Eastwood	8.2
The Prestige	Christopher Nolan	8.6
The Prestige	John Favreau	9.8

SELECT promedio

FROM PelEval

WHERE pelicula = 'The Prestige' AND

director = 'Christopher Nolan'



promedio
8.6

Vistas materializadas vs tablas

```
CREATE MATERIALIZED VIEW PelEval AS
```

```
    SELECT pelicula, director,  
           AVG(eval) AS promedio  
    FROM Evaluaciones  
    GROUP BY pelicula, director
```

```
REFRESH MATERIALIZED VIEW  
PelEval
```

```
CREATE TABLE PelEval AS
```

```
    SELECT pelicula, director,  
           AVG(eval) AS promedio  
    FROM Evaluaciones  
    GROUP BY pelicula, director
```

- Las vistas materializadas generan la misma abstracción que las normales pero son más rápidas.
- Por otro lado, requieren espacio adicional e introducen una redundancia al esquema que nos debemos preocupar de mantener consistente, por ejemplo ejecutando `REFRESH MATERIALIZED VIEW` según sea necesario.
- Podemos automatizar eso en la misma DB? Si, mediante `TRIGGERS`.

Triggers

Triggers

- Procedimientos en la base de datos que se *gatillan* cada vez que se ejecuta algún evento.
- Contribuyen a forzar ciertas restricciones más complejas y a mantener la consistencia de la BD.

Triggers

Sintaxis (parte de)

- El evento puede ser {BEFORE | AFTER | INSTEAD OF} {CREATE| DELETE | UPDATE}. (ej: BEFORE CREATE)
- Podemos ejecutar el trigger para cada fila afectada por el evento (FOR EACH ROW), o 1 vez por evento (FOR EACH STATEMENT).

Triggers

Ejemplo

Por ejemplo: Queremos disminuir el stock de un producto cada vez que se le crea una venta .

La sintaxis cambia mucho entre sistemas pero es más o menos así:

```
CREATE TRIGGER reducir_stock
```

```
AFTER CREATE ON Ventas
```

```
FOR EACH ROW
```

```
BEGIN
```

```
UPDATE Productos
```

```
SET stock = Productos.stock - 1
```

```
WHERE NEW.id_productos = Productos.id
```

```
END
```

Triggers

Ejemplo

```
CREATE TRIGGER reducir_stock
```

```
AFTER CREATE ON Ventas
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    UPDATE Productos
```

```
    SET stock = Productos.stock - 1
```

```
    WHERE NEW.id_productos = Productos.id
```

```
END
```

Ejecutar cada vez que
se cree una venta.

Triggers

Ejemplo

```
CREATE TRIGGER reducir_stock
```

```
AFTER CREATE ON Ventas
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    UPDATE Productos
```

```
    SET stock = Productos.stock - 1
```

```
    WHERE NEW.id_productos = Productos.id
```

```
END
```

Para cada fila creada

Triggers

Ejemplo

```
CREATE TRIGGER reducir_stock
```

```
AFTER CREATE ON Ventas
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    UPDATE Productos
```

```
    SET stock = Productos.stock - 1
```

```
    WHERE NEW.id_productos = Productos.id
```

```
END
```

Actualizamos el stock en la tabla Productos

Triggers

Ejemplo

```
CREATE TRIGGER reducir_stock
```

```
AFTER CREATE ON Ventas
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    UPDATE Productos
```

```
    SET stock = Productos.stock - 1
```

```
    WHERE NEW.id_productos = Productos.id
```

```
END
```

En que el id del producto corresponda con la venta que se creó.

Triggers

Ejemplo

Para lo que comentábamos antes:

```
CREATE TRIGGER refresh_eval  
AFTER CREATE ON Evaluaciones  
FOR EACH STATEMENT  
BEGIN  
    REFRESH MATERIALIZED VIEW PelEval  
END
```

PL/pgSQL

-

Lenguaje procedural de SQL

PL/pgSQL

PL/pgSQL es un lenguaje procedural de SQL con el objetivo de:

- Crear funciones y activar **procedimientos almacenados**.
- Agregar estructuras de control al lenguaje SQL.
- Realizar cálculos complejos.
- Heredar todos los tipos, funciones y operadores definidos por el usuario.
- Se trabajan en archivos con extensión .sql

¡La mayoría de los procesos que se pueden definir en funciones de otros lenguajes también se puede hacer con PL/pgSQL!

PL/pgSQL - Sintaxis

CREATE OR REPLACE FUNCTION

<nombre_funcion> (<argumentos>)

RETURNS <tipo de dato> **AS \$\$** -- declaramos lo que retorna

DECLARE -- declaramos variables a utilizar

 <variable_1> -- en caso de ser necesario

 <variable_2>

...

BEGIN -- inicio de función

... -- sentencias SQL

END -- finalización de la función

\$\$ language plpgsql

PL/pgSQL - Sintaxis

```
BEGIN
```

```
IF condicion THEN
```

```
...
```

```
    ELSE IF otra condicion THEN
```

```
...
```

```
ELSE
```

```
...
```

```
END IF;
```

```
FOR variable IN iterable
```

```
LOOP
```

```
...
```

```
END LOOP;
```

```
RETURN valor -- ejemplo de valor → SELECT * FROM TABLE;
```

```
END
```

Stored Procedures

Stored Procedures

- Son funciones definidas mediante SQL, que quedan guardadas en el mismo DBMS y por lo tanto se pueden usar en consultas.
- Permiten ejecutar lógica compleja y repetitiva directamente en el DBMS.
- En un sólo proceso podemos ejecutar todas las consultas que queramos, o también hacer control de flujo con IFs o loops.

Stored Procedures vs API

Característica	API	Procedimientos Almacenados
Rendimiento	Generalmente más lento que los procedimientos almacenados debido a la latencia de red, serialización y deserialización de procesos.	Típicamente más rápido para operaciones de datos ya que se ejecutan dentro del servidor de base de datos, reduciendo la necesidad de transferencia de datos a través de la red.
Seguridad	La seguridad se gestiona a través de mecanismos de autenticación como OAuth, claves API, etc., y puede estar expuesta a mayores riesgos de seguridad debido a una mayor accesibilidad a través de internet.	Generalmente más seguro ya que la ejecución se limita al entorno de la base de datos; sin embargo, una codificación incorrecta puede llevar a ataques de inyección SQL.
Flexibilidad	Altamente flexible en el manejo de varios tipos de datos y en la conexión de diferentes sistemas o tecnologías.	Menos flexible, maneja principalmente la manipulación de datos y está confinado a las capacidades de la base de datos.
Escalabilidad	Escalable mediante técnicas como balanceo de carga, caché, etc., pero puede volverse complejo ya que escalar implica múltiples capas de infraestructura.	La escalabilidad está limitada a la capacidad del servidor de base de datos, aunque las bases de datos modernas admiten altos niveles de concurrencia y complejo balanceo de carga.
Mantenibilidad	Requiere actualizaciones y monitoreo regulares de diferentes capas (red, seguridad, interfaces de aplicación), lo cual puede ser complejo.	Más fácil de mantener ya que involucra solo la capa de base de datos; las actualizaciones y optimizaciones generalmente son directas.
Costo de Desarrollo	Potencialmente más alto debido a la necesidad de documentación detallada, medidas de seguridad rigurosas e interfaz entre múltiples sistemas.	Menores costos de desarrollo si las operaciones están estrictamente relacionadas con la manipulación de datos dentro del mismo sistema de base de datos.
Casos de Uso	Ideal para aplicaciones que requieren interacción entre diferentes sistemas o plataformas, integraciones externas y donde el acceso a datos en tiempo real no es crítico.	Mejor para la manipulación de datos compleja, tareas de procesamiento de datos rápidas y cuando las operaciones están confinadas dentro de un solo sistema de base de datos.

Stored Procedures

Sintaxis (parte de)

CREATE or **REPLACE Function** <nombre_funcion> (<argumentos>) **RETURNS**

<tipo_retorno> **AS**

\$\$

DECLARE

<declaracion de variables>

BEGIN

<sentencias **SQL**>

END

\$\$ **language** plpgsql

Stored Procedures

Ejemplo

Procedimiento para insertar una fila a tabla personas:

```
CREATE OR REPLACE FUNCTION insertar_persona (rut varchar, nombre varchar, apellido  
varchar) RETURNS void AS  
$$  
BEGIN  
INSERT INTO personas VALUES (rut,nombre,apellido);  
END  
$$ language plpgsql
```


Stored Procedures

Ejemplo

Ahora para usarla hacemos:

```
SELECT insertar_persona('11.111.111-1', 'pepito', 'los palotes')
```

Stored Procedures

Ejemplo

Podemos iterar sobre resultados de consultas y usar eso para procesar e insertar datos a otras tablas:


```
CREATE OR REPLACE FUNCTION transferencia_nombres() RETURNS void AS $$  
DECLARE  
    tupla RECORD;  
    concat varchar;  
BEGIN  
    FOR tupla IN SELECT * FROM Personas LOOP  
        concat = tupla.nombre || tupla.apellido;  
        insert into personascompleto values (tupla.rut, concat);  
    END LOOP;  
END  
$$ language plpgsql
```

Stored Procedures

Consultas dinámicas

Podemos retornar consultas completas y además usar los argumentos de la función para generarlas de forma dinámica.

```
CREATE OR REPLACE FUNCTION vuelos_desde (c_origen varchar)
RETURNS TABLE (ciudad_destino varchar(50), horas integer) AS $$
BEGIN
RETURN QUERY EXECUTE '
    SELECT ciudad_destino, horas
    FROM Vuelo
    WHERE ciudad_origen = $1'
    USING c_origen;
RETURN; END
$$ language plpgsql
```



¿Por qué usar stored procedures?

- Ventaja principal: **velocidad**

Beneficios:

- Seguridad: Controla el acceso a los datos y a la lógica de la base de datos.
- Rendimiento: puede optimizarse para la velocidad, especialmente para consultas complejas.
- Reutilizabilidad: definido una vez, usado repetidamente en aplicaciones

Problemas con stored procedures...

- Desventaja principal: **Dificultad o imposibilidad de migrar a otra base de datos**

Problemas:

- Hace que el testing sea mucho más difícil.
- También hace que sea más difícil mantener el código, porque SQL tiende a ser menos legible a medida que se vuelve más complejo.
- El control de versiones puede ser un problema, incluso si el script está almacenado en el proyecto de Visual Studio usando GIT, cualquiera puede modificar el procedimiento en cualquier momento y nadie puede darse cuenta.

Lógica en la BDD

Conclusiones

- Escribir lógica compleja en SQL se puede volver inmanejable muy rápido. Además, triggers, vistas y procedimientos ya guardados en la DB requieren de ejecutar un `CREATE` para modificarse.
- Por otro lado, los frameworks web modernos, de manera opinionada presentan escaso soporte para este tipo de funcionalidades. Estas dos cosas hacen que usar views, procedures y triggers no se vean mucho en aplicaciones modernas.
- De todas formas, estas cosas aún se ven en sistemas legados (antiguos) o en sistemas de *Data engineering*, como *data warehouses* o *data lakes* (hablaremos más de lo que es eso en el futuro) y en ciertos servicios en la nube.