



Introducción a Mysql

Final del curso



Clase 4

Temas

Repaso clase 3

Introducción a:

- Subconsultas
- Vistas
- Procedimientos almacenados
- Triggers

Cierre:

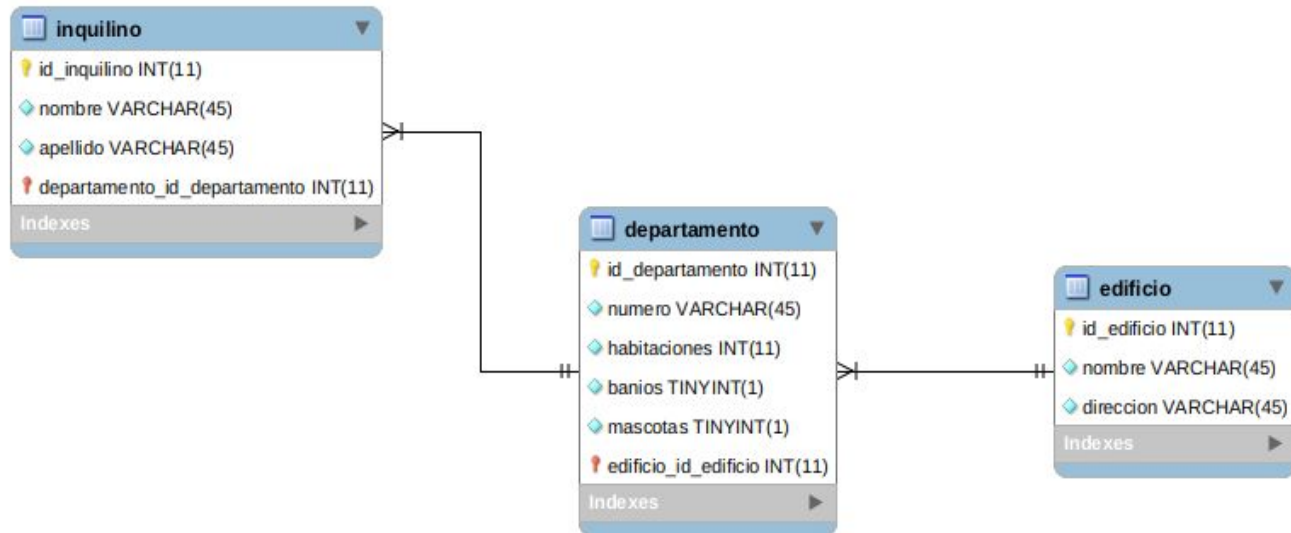
- Backup BD
- Restaurar BD



Empecemos



Repaso



Join

SELECT * FROM edificio

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

#	id_edificio	nombre	direccion
1	1	Torres A	Direccion Falsa 123
2	2	Torres B	Direccion Falsa 1234
3	3	Torres C	Direccion Falsa 12345
*	NULL	NULL	NULL



edificio 6



Action Output

	#	Time	Action	Message	Duration / Fetch
✓	1	15:00:44	SELECT * FROM edificio LIMIT 0, 1000	3 row(s) returned	0,00044 sec / 0,000...


Join


SELECT i.nombre, i.apellido FROM inquilino i

Result Grid   Filter Rows:

Export:  Wrap Cell Content: 

#	nombre	apellido
1	Juan	Perez
2	Andrea	Gomez
3	Raul	Garcia
4	Marcos	Borja
5	Sabrina	Pescce
6	Tatiana	Lezza




inquilino 18 



Action Output 

#	Time	Action	Message	Duration / Fetch
1	15:11:42	SELECT i.nombre, i.apellido FROM inquilino i LIMIT 0, 1000	6 row(s) returned	0,00055 sec / 0,000...


Join


```
SELECT i.nombre, i.apellido, d.numero, d.habitaciones, d.banios, d.mascotas
FROM inquilino i INNER JOIN departamento d
ON i.departamento_id_departamento = d.id_departamento
```

Result Grid   Filter Rows: 

Export:  Wrap Cell Content: 

#	nombre	apellido	numero	habitaciones	banios	mascotas
1	Juan	Perez	1-A	1	1	0
2	Andrea	Gomez	2-A	3	1	0
3	Raul	Garcia	1-B	2	1	0
4	Marcos	Borja	2-B	1	1	0
5	Sabrina	Pescce	3-A	4	2	1
6	Tatiana	Lezza	3-B	2	1	1

Result 16 

Action Output 

#	Time	Action	Message	Duration / Fetch
✓ 1	15:09:40	SELECT i.nombre, i.apellido, d.numero, d.habitaciones, d.bani...	6 row(s) returned	0,00060 sec / 0,000...

Join

```
SELECT i.nombre, i.apellido, d.numero, d.habitaciones, d.banios, d.mascotas, e.nombre, e.direccion
FROM inquilino i INNER JOIN departamento d
ON i.departamento_id_departamento = d.id_departamento
INNER JOIN edificio e ON d.edificio_id_edificio = e.id_edificio
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

#	nombre	apellido	numero	habitaciones	banios	mascotas	nombre	direccion
1	Juan	Perez	1-A	1	1	0	Torres A	Direccion Falsa 123
2	Andrea	Gomez	2-A	3	1	0	Torres A	Direccion Falsa 123
3	Raul	Garcia	1-B	2	1	0	Torres B	Direccion Falsa 1234
4	Marcos	Borja	2-B	1	1	0	Torres B	Direccion Falsa 1234
5	Sabrina	Pescce	3-A	4	2	1	Torres C	Direccion Falsa 12345
6	Tatiana	Lezza	3-B	2	1	1	Torres C	Direccion Falsa 12345

Result 17

Action Output

#	Time	Action	Message	Duration / Fetch
1	15:10:27	SELECT i.nombre, i.apellido, d.numero, d.habitaciones, d.bani...	6 row(s) returned	0.00078 sec / 0.000...

Join

```
SELECT i.nombre, i.apellido, d.numero, d.habitaciones, d.banios, d.mascotas, e.nombre as nombre_edificio,  
e.direccion FROM inquilino i INNER JOIN departamento d  
ON i.departamento_id_departamento = d.id_departamento  
INNER JOIN edificio e ON d.edificio_id_edificio = e.id_edificio
```

Result Grid									
Filter Rows: <input type="text"/>									
Export: <input type="text"/> Wrap Cell Content: <input type="text"/>									
#	nombre	apellido	numero	habitaciones	banios	mascotas	nombre_edificio	direccion	
1	Juan	Perez	1-A	1	1	0	Torres A	Direccion Falsa 123	
2	Andrea	Gomez	2-A	3	1	0	Torres A	Direccion Falsa 123	
3	Raul	Garcia	1-B	2	1	0	Torres B	Direccion Falsa 1234	
4	Marcos	Borja	2-B	1	1	0	Torres B	Direccion Falsa 1234	
5	Sabrina	Pescce	3-A	4	2	1	Torres C	Direccion Falsa 12345	
6	Tatiana	Lezza	3-B	2	1	1	Torres C	Direccion Falsa 12345	

Result 23

Action Output

#	Time	Action	Message	Duration / Fetch
1	15:26:27	SELECT i.nombre, i.apellido, d.numero, d.habitaciones, d.bani...	6 row(s) returned	0,00059 sec / 0,000...

Join

```
SELECT i.nombre, i.apellido, d.numero, d.habitaciones, d.banios, d.mascotas, e.nombre as nombre_edificio,  
e.direccion FROM inquilino i INNER JOIN departamento d  
ON i.departamento_id_departamento = d.id_departamento  
INNER JOIN edificio e ON d.edificio_id_edificio = e.id_edificio  
WHERE d.habitaciones = 2 AND d.mascotas = 1
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

#	nombre	apellido	numero	habitaciones	banio:	mascotas	nombre_edificio	direccion
1	Tatiana	Lezza	3-B	2	1	1	Torres C	Direccion Falsa 12345

Result 27

Action Output

#	Time	Action	Message	Duration / Fetch
1	15:28:34	SELECT i.nombre, i.apellido, d.numero, d.habitaciones, d.bani...	1 row(s) returned	0,00034 sec / 0,000...



Sub-Consultas



Sub-Consultas

- Un select dentro de otro
- Este select puede devolver un solo valor o una lista
- Estos valores se usan como condiciones de filtro
- Pueden pensarlo de esta forma:

	SELECT ...	
		SELECT ...



Sub-Consultas

- Hay varios tipos. Como introducción vamos a ver:

- 1) Sub-consultas escalonadas

- 2) Sub-consultas de listas



Sub-Consultas

- Hay varios tipos. Como introducción vamos a ver:
 - 1) Sub-consultas escalonadas
 - Devuelven un solo valor que se va a usar luego en una condición de filtro
 - 2) Sub-consultas de listas
 - Devuelve una lista de valores que luego se van a usar en una condición de filtro



Operadores

Lógicos:

- **AND** Se tienen que cumplir todas las condiciones para que sea True
- **OR** Con que se cumpla 1 ya está
- **NOT** Lo contrario

Comparación:

- **LIKE** Se usa para comprar string, buscar palabras dentro de texto, etc
- **<>** Distinto
- **<** Menor
- **>** Mayor
- **<=** Menor igual
- **>=** Mayor igual
- **BETWEEN** Intervalos
- **IN** En, dentro de, etc
- **ANY** Cualquiera
- **ALL** Todos



Sub-Consultas escalonada



Sub-Consultas escalonada

- Devuelve un solo valor que se usa como condición de filtro
- Querys de ejemplo:
 - 1) Obtener todos los jugadores cuyo sueldo **sea igual al mayor sueldo de la liga**
 - 2) Obtener todos los jugadores cuyo sueldo **sea igual al menor sueldo de la liga**
 - 3) Obtener todos los jugadores cuyo sueldo **sea mayor a la media de sueldos de la liga**


Query 1: Obtener todos los jugadores cuyo sueldo sea igual al MAYOR sueldo de la liga


```
SELECT max(sueldo_mensual) FROM jugador
```

Result Grid  Filter Rows:  Export:  Wrap Cell Content: 

#	max(sueldo_mensual)
1	200000

Result 37 

Action Output 

	#	Time	Action	Message	Duration / Fetch
	1	16:27:51	SELECT max(sueldo_mensual) FROM jugador LIMIT 0, 1000	1 row(s) returned	0,00027 sec / 0,000...

Query 1: Obtener todos los jugadores cuyo sueldo sea igual al MAYOR sueldo de la liga

```
SELECT nombre, apellido, sueldo_mensual FROM jugador
WHERE sueldo_mensual = (SELECT max(sueldo_mensual) FROM jugador)
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

#	nombre	apellido	sueldo_mensual
1	Lionel Andres	Messi	200000
2	Cristiano	Ronaldo	200000

jugador 41

Action Output

#	Time	Action	Message	Duration / Fetch
1	16:30:37	SELECT nombre, apellido, sueldo_mensual FROM jugador WH...	2 row(s) returned	0,00063 sec / 0,000...

Query 2: Obtener todos los jugadores cuyo sueldo sea igual al MENOR sueldo de la liga

```
SELECT min(sueldo_mensual) FROM jugador
```

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

#	min(sueldo_mensual)
1	55000

Result 42 

Action Output 

#	Time	Action	Message	Duration / Fetch
✓ 1	16:34:45	SELECT min(sueldo_mensual) FROM jugador LIMIT 0, 1000	1 row(s) returned	0,00040 sec / 0,000...

Query 2: Obtener todos los jugadores cuyo sueldo sea igual al MENOR sueldo de la liga

```
SELECT nombre, apellido, sueldo_mensual FROM jugador
WHERE sueldo_mensual = (SELECT min(sueldo_mensual) FROM jugador)
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

#	nombre	apellido	sueldo_mensual
1	Neymar	Jr	55000
2	Paulo	Dybala	55000

jugador 44

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 1	16:35:32	SELECT nombre, apellido, sueldo_mensual FROM jugador WH...	2 row(s) returned	0,00051 sec / 0,000...

Query 3: Obtener todos los jugadores cuyo sueldo sea mayor a la MEDIA de sueldos de la liga
(Promedio - Funciones de agregación)

```
SELECT avg(sueldo_mensual) FROM jugador
```

Result Grid			Filter Rows:		Export:		Wrap Cell Content:	
#	avg(sueldo_mensual)							
1	94500.0000							





Result 45	
-----------	--

Action Output	
---------------	--


#	Time	Action	Message	Duration / Fetch
1	16:37:38	SELECT avg(sueldo_mensual) FROM jugador LIMIT 0, 1000	1 row(s) returned	0,00047 sec / 0,000...


Query 3: Obtener todos los jugadores cuyo sueldo sea mayor a la MEDIA de sueldos de la liga

```
SELECT nombre, apellido, sueldo_mensual FROM jugador
WHERE sueldo_mensual > (SELECT avg(sueldo_mensual) FROM jugador)
```

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

#	nombre	apellido	sueldo_mensual
1	Lionel Andres	Messi	200000
2	Cristiano	Ronaldo	200000

jugador 47 

Action Output 

#	Time	Action	Message	Duration / Fetch
1	16:38:16	SELECT nombre, apellido, sueldo_mensual FROM jugador WH...	2 row(s) returned	0,00040 sec / 0,000...



Sub-Consultas de listas







Sub-Consultas de listas


- Devuelve una lista de valores que se usan como condición de filtro
- Querys de ejemplo:
 - 1) Obtener todos los jugadores que cobren **más que TODOS** los sueldos del Manchester City
 - 2) Obtener todos los jugadores que cobren **más que ALGUNO** de los sueldos del Manchester City
 - 3) Obtener todos los jugadores cuyo sueldo **este DENTRO** de todos los sueldos del Manchester City


Query 1: Obtener todos los jugadores que cobren **más que TODOS** los sueldos del Manchester City

```
SELECT j.nombre, j.apellido, j.sueldo_mensual, e.detalle_equipo
FROM jugador j INNER JOIN equipo e
ON j.equipo_id_equipo = e.id_equipo
WHERE e.detalle_equipo = 'Manchester City';
```

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

#	nombre	apellido	sueldo_mensual	detalle_equipo
1	Scott	Carson	45000	Manchester City
2	Kyle	Walker	47000	Manchester City
3	Sergio	Aguero	62000	Manchester City

Result 108 

Action Output 


#	Time	Action	Message	Duration / Fetch
1	17:15:17	SELECT j.nombre, j.apellido, j.sueldo_mensual, e.detalle_equipo...	3 row(s) returned	0.00061 sec / 0.000...


Query 1: Obtener todos los jugadores que cobren **más que TODOS** los sueldos del Manchester City

```
SELECT j.sueldo_mensual
FROM jugador j INNER JOIN equipo e
ON j.equipo_id_equipo = e.id_equipo
WHERE e.detalle_equipo = 'Manchester City';
```

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

#	sueldo_mensual
1	45000
2	47000
3	62000





Result 115 

Action Output 


#	Time	Action	Message	Duration / Fetch
1	17:26:13	SELECT j.sueldo_mensual FROM jugador j INNER JOIN equipo e...	3 row(s) returned	0.00052 sec / 0,000...


Query 1: Obtener todos los jugadores que cobren **más** que **TODOS** los sueldos del Manchester City

```
SELECT nombre, apellido, sueldo_mensual FROM jugador
WHERE sueldo_mensual > ALL (SELECT j.sueldo_mensual
FROM jugador j INNER JOIN equipo e ON j.equipo_id_equipo = e.id_equipo
WHERE e.detalle_equipo = 'Manchester City')
```

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

#	nombre	apellido	sueldo_mensual
1	Lionel Andres	Messi	200000
2	Cristiano	Ronaldo	200000
3	Eden	Hazard	63000

jugador 112 

Action Output 

#	Time	Action	Message	Duration / Fetch
✓ 1	17:20:42	SELECT nombre, apellido, sueldo_mensual FROM jugador WH...	3 row(s) returned	0,00077 sec / 0,000...

Query 1: Obtener todos los jugadores que cobren **más que TODOS** los sueldos del Manchester City

#	nombre	apellido	sueldo_mensual
1	Lionel Andres	Messi	200000
2	Cristiano	Ronaldo	200000
3	Neymar	Jr	55000
4	Jan	Oblak	44000
5	Eden	Hazard	63000
6	Marc-Andre	ter Stegen	60000
7	Virgil	Van Dijk	58000
8	Paulo	Dybala	55000
9	Scott	Carson	45000
10	Kyle	Walker	47000
11	Sergio	Aguero	62000







#	sueldo_mensual
1	45000
2	47000
3	62000




#	nombre	apellido	sueldo_mensual
1	Lionel Andres	Messi	200000
2	Cristiano	Ronaldo	200000
3	Eden	Hazard	63000


Query 2: Obtener todos los jugadores que cobren **más que ALGUNO** de los sueldos del Manchester City

```
SELECT nombre, apellido, sueldo_mensual FROM jugador
WHERE sueldo_mensual > ANY (SELECT j.sueldo_mensual
FROM jugador j INNER JOIN equipo e ON j.equipo_id_equipo = e.id_equipo
WHERE e.detalle_equipo = 'Manchester City')
```

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

#	nombre	apellido	sueldo_mensual
1	Lionel Andres	Messi	200000
2	Cristiano	Ronaldo	200000
3	Neymar	Jr	55000
4	Eden	Hazard	63000
5	Marc-Andre	ter Stegen	60000
6	Virgil	Van Dijk	58000
7	Paulo	Dybala	55000
8	Kyle	Walker	47000
9	Sergio	Aguero	62000

jugador 116 

Action Output 

#	Time	Action	Message	Duration / Fetch
1	17:29:25	SELECT nombre, apellido, sueldo_mensual FROM jugador WH...	9 row(s) returned	0,00065 sec / 0,000...

Query 2: Obtener todos los jugadores que cobren **más que ALGUNO** de los sueldos del Manchester City

#	nombre	apellido	sueldo_mensual
1	Lionel Andres	Messi	200000
2	Cristiano	Ronaldo	200000
3	Neymar	Jr	55000
4	Jan	Oblak	44000
5	Eden	Hazard	63000
6	Marc-Andre	ter Stegen	60000
7	Virgil	Van Dijk	58000
8	Paulo	Dybala	55000
9	Scott	Carson	45000
10	Kyle	Walker	47000
11	Sergio	Aguero	62000







#	sueldo_mensual
1	45000
2	47000
3	62000




#	nombre	apellido	sueldo_mensual
1	Lionel Andres	Messi	200000
2	Cristiano	Ronaldo	200000
3	Neymar	Jr	55000
4	Eden	Hazard	63000
5	Marc-Andre	ter Stegen	60000
6	Virgil	Van Dijk	58000
7	Paulo	Dybala	55000
8	Kyle	Walker	47000
9	Sergio	Aguero	62000


Query 3: Obtener todos los jugadores cuyo sueldo este **DENTRO** de todos los sueldos del Manchester City

```
SELECT nombre, apellido, sueldo_mensual FROM jugador
WHERE sueldo_mensual IN (SELECT j.sueldo_mensual
FROM jugador j INNER JOIN equipo e ON j.equipo_id_equipo = e.id_equipo
WHERE e.detalle_equipo = 'Manchester City')
```

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

#	nombre	apellido	sueldo_mensual
1	Scott	Carson	45000
2	Kyle	Walker	47000
3	Sergio	Aguero	62000

jugador 128 

Action Output 

#	Time	Action	Message	Duration / Fetch
1	17:34:26	SELECT nombre, apellido, sueldo_mensual FROM jugador WH...	3 row(s) returned	0,00063 sec / 0,000...

Query 3: Obtener todos los jugadores que cobren **más que ALGUNO** de los sueldos del Manchester City

#	nombre	apellido	sueldo_mensual
1	Lionel Andres	Messi	200000
2	Cristiano	Ronaldo	200000
3	Neymar	Jr	55000
4	Jan	Oblak	44000
5	Eden	Hazard	63000
6	Marc-Andre	ter Stegen	60000
7	Virgil	Van Dijk	58000
8	Paulo	Dybala	55000
9	Scott	Carson	45000
10	Kyle	Walker	47000
11	Sergio	Aguero	62000



#	sueldo_mensual
1	45000
2	47000
3	62000



#	nombre	apellido	sueldo_mensual
1	Scott	Carson	45000
2	Kyle	Walker	47000
3	Sergio	Aguero	62000

Pero si modifico un sueldo

#	nombre	apellido	sueldo_mensual
1	Paulo	Dybala	62000
2	Scott	Carson	45000
3	Kyle	Walker	47000
4	Sergio	Aguero	62000



Vistas






Vistas


Una vista es una **tabla virtual** que se genera a partir de una consulta de selección. Dicho de otro modo. Escribimos una consulta de selección (sobre una o más tablas) para leer los datos, y **almacenamos el resultado en una vista**.


Vistas

```
SELECT j.nombre, j.apellido, j.sueldo_mensual, e.detalle_equipo
FROM jugador j INNER JOIN equipo e
ON j.equipo_id_equipo = e.id_equipo
```

Result Grid   Filter Rows: Export:  Wrap Cell Content: [TA](#)

#	nombre	apellido	sueldo_mensual	detalle_equipo
1	Lionel Andres	Messi	200000	FC Barcelona
2	Marc-Andre	ter Stegen	60000	FC Barcelona
3	Cristiano	Ronaldo	200000	Juventus
4	Paulo	Dybala	62000	Juventus
5	Neymar	Jr	55000	Paris Saint-Germain
6	Jan	Oblak	44000	Atletico Madrid
7	Eden	Hazard	63000	Real Madrid
8	Virgil	Van Dijk	58000	Liverpool
9	Scott	Carson	45000	Manchester City
10	Kyle	Walker	47000	Manchester City
11	Sergio	Aguero	62000	Manchester City

Result 134 

Action Output 

#	Time	Action	Message	Duration / Fetch
1	18:02:26	SELECT j.nombre, j.apellido, j.sueldo_mensual, e.detalle_equipo...	11 row(s) returned	0,00060 sec / 0,000...





Vistas

```
CREATE VIEW sueldo_jugadores AS  
SELECT j.nombre, j.apellido, j.sueldo_mensual, e.detalle_equipo  
FROM jugador j INNER JOIN equipo e  
ON j.equipo_id_equipo = e.id_equipo;
```


Views		#	Time	Action	Message	Duration / Fetch
▼	sueldo_jugadores	✓ 1	18:03:06	CREATE VIEW sueldo_jugadores AS SELECT j.nombre, j.apellid...	0 row(s) affected	0,033 sec
	◆ nombre					
	◆ apellido					
	◆ sueldo_mensual					
	◆ detalle_equipo					


Vistas

SELECT * FROM sueldo_jugadores

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

#	nombre	apellido	suelo_mensual	detalle_equipo
1	Lionel Andres	Messi	200000	FC Barcelona
2	Marc-Andre	ter Stegen	60000	FC Barcelona
3	Cristiano	Ronaldo	200000	Juventus
4	Paulo	Dybala	62000	Juventus
5	Neymar	Jr	55000	Paris Saint-Germain
6	Jan	Oblak	44000	Atletico Madrid
7	Eden	Hazard	63000	Real Madrid
8	Virgil	Van Dijk	58000	Liverpool
9	Scott	Carson	45000	Manchester City
10	Kyle	Walker	47000	Manchester City
11	Sergio	Aguero	62000	Manchester City

Result 134 



Action Output 



#	Time	Action	Message	Duration / Fetch
1	18:02:26	SELECT j.nombre, j.apellido, j.suelo_mensual, e.detalle_equip...	11 row(s) returned	0.00060 sec / 0.000...

Vistas


```
ALTER VIEW sueldo_jugadores AS  
SELECT nombre, apellido, sueldo_mensual  
FROM jugador
```


```
SELECT * FROM sueldo_jugadores
```

Result Grid   Filter Rows:

Export:  Wrap Cell Content: 

#	nombre	apellido	sueldo_mensual
1	Lionel Andres	Messi	200000
2	Cristiano	Ronaldo	200000
3	Neymar	Jr	55000
4	Jan	Oblak	44000
5	Eden	Hazard	63000
6	Marc-Andre	ter Stegen	60000
7	Virgil	Van Dijk	58000
8	Paulo	Dybala	62000
9	Scott	Carson	45000
10	Kyle	Walker	47000
11	Sergio	Aguero	62000

sueldo_jugadores 141 

Action Output 

#	Time	Action	Message	Duration / Fetch
1	18:13:14	SELECT * FROM sueldo_jugadores LIMIT 0, 1000	11 row(s) returned	0,00020 sec / 0,000...



Vistas

DROP VIEW sueldo_jugadores

✓	7	18:10:19	DROP VIEW sueldo_jugadores
---	---	----------	----------------------------

0 row(s) affected

0,00059 sec



Vistas

- **SEGURIDAD:** Ocultar tablas y que solo tengan acceso a las listas
- **OPTIMIZACIÓN:** Si se hace la misma query todos los días, se arma una lista y se consulta esa vista
- **PRUEBA:** En vez de hacer consultas a todas las tablas se consulta la vista y se evita la posibilidad que por error rompamos una tabla



Procedimientos almacenados



Procedimientos almacenados

Supongamos que hay muchas aplicaciones que consultan a nuestra base de datos, no le vamos a dar acceso a todo el mundo a las tablas, lo que hacemos es crear un procedimiento que encapsule una query y que las app llamen a ese procedimiento

Los procedimientos pueden aceptar parámetros (dinámico) o no.



Procedimientos almacenados

```
SELECT j.nombre, j.apellido, j.sueldo_mensual, n.detalle_nacionalidad, e.detalle_equipo, p.detalle_posicion
FROM jugador j INNER JOIN nacionalidad n on j.nacionalidad_id_nacionalidad = n.id_nacionalidad
INNER JOIN equipo e on j.equipo_id_equipo = e.id_equipo
INNER JOIN posicion p on j.posicion_id_posicion = p.id_posicion
WHERE e.detalle_equipo = 'FC Barcelona'
```



Procedimientos almacenados sin parámetros

```
CREATE PROCEDURE all_info_jugadores_barcelona()
```

```
SELECT j.nombre, j.apellido, j.sueldo_mensual, n.detalle_nacionalidad, e.detalle_equipo, p.detalle_posicion  
FROM jugador j INNER JOIN nacionalidad n on j.nacionalidad_id_nacionalidad = n.id_nacionalidad  
INNER JOIN equipo e on j.equipo_id_equipo = e.id_equipo  
INNER JOIN posicion p on j.posicion_id_posicion = p.id_posicion  
WHERE e.detalle_equipo = 'FC Barcelona'
```

▼ Stored Procedures

all_info_jugadores_barcelona

#	Time	Action	Message	Duration / Fetch
1	18:37:37	CREATE PROCEDURE all_info_jugadores_barcelona() SELECT j...	0 row(s) affected	0,00069 sec


Procedimientos almacenados

CALL all_info_jugadores_barcelona()


Result Grid  Filter Rows:

Export:  Wrap Cell Content: 

#	nombre	apellido	sueldo_mensual	detalle_nacionalidad	detalle_equipo	detalle_posicion
1	Lionel Andres	Messi	200000	Argentina	FC Barcelona	Extremo derecho
2	Marc-Andre	ter Stegen	60000	Alemania	FC Barcelona	Arquero

Result 9 

Action Output 

	#	Time	Action	Message	Duration / Fetch
	1	18:38:50	CALL all_info_jugadores_barcelona()	2 row(s) returned	0.00077 sec / 0,000...

Procedimientos almacenados con parámetros

```
CREATE PROCEDURE all_info_jugadores(equipo_a_verificar varchar(45))
```

```
SELECT j.nombre, j.apellido, j.sueldo_mensual, n.detalle_nacionalidad, e.detalle_equipo, p.detalle_posicion  
FROM jugador j INNER JOIN nacionalidad n on j.nacionalidad_id_nacionalidad = n.id_nacionalidad  
INNER JOIN equipo e on j.equipo_id_equipo = e.id_equipo  
INNER JOIN posicion p on j.posicion_id_posicion = p.id_posicion  
WHERE e.detalle_equipo = equipo_a_verificar;
```

▼ Stored Procedures

all_info_jugadores_barcelona

#	Time	Action	Message	Duration / Fetch
1	18:37:37	CREATE PROCEDURE all_info_jugadores_barcelona() SELECT j,...	0 row(s) affected	0,00069 sec

Procedimientos almacenados

CALL all_info_jugadores('Real Madrid')

#	nombre	apellido	sueldo_mensual	detalle_nacionalidad	detalle_equipo	detalle_posicion
1	Eden	Hazard	63000	Belgica	Real Madrid	Extremo izquierdo

equipo 10 Result 16

Action Output

	#	Time	Action	Message	Duration / Fetch
✓	1	18:46:23	CALL all_info_jugadores('Real Madrid')	1 row(s) returned	0,00064 sec / 0,000...

Procedimientos almacenados

CALL all_info_jugadores('Manchester City')

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

#	nombre	apellido	sueldo_mensual	detalle_nacionalidad	detalle_equipo	detalle_posicion
1	Scott	Carson	45000	Inglaterra	Manchester City	Arquero
2	Kyle	Walker	47000	Inglaterra	Manchester City	Defensa central
3	Sergio	Aguero	62000	Argentina	Manchester City	Centro delantero

equipo 10


Result 17

Action Output



#	Time	Action	Message	Duration / Fetch
1	18:46:59	CALL all_info_jugadores('Manchester City')	3 row(s) returned	0,00075 sec / 0,000...


Procedimientos almacenados

CALL all_info_jugadores('FC Barcelona')

Result Grid  Filter Rows: Export:  Wrap Cell Content: 

#	nombre	apellido	sueldo_mensual	detalle_nacionalidad	detalle_equipo	detalle_posicion
1	Lionel Andres	Messi	200000	Argentina	FC Barcelona	Extremo derecho
2	Marc-Andre	ter Stegen	60000	Alemania	FC Barcelona	Arquero

equipo 10  Result 18 

Action Output 

#	Time	Action	Message	Duration / Fetch
1	18:47:44	CALL all_info_jugadores('FC Barcelona')	2 row(s) returned	0,00053 sec / 0,000...

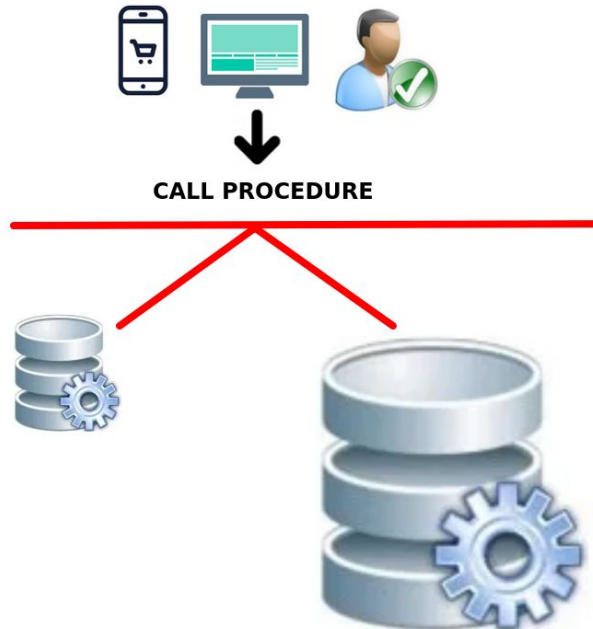


Procedimientos almacenados

DROP PROCEDURE `all_info_jugadores`;
DROP PROCEDURE `all_info_jugadores_barcelona`;

	#	Time	Action	Message	Duration / Fetch
✓	1	18:50:08	DROP PROCEDURE <code>all_info_jugadores</code>	0 row(s) affected	0,00057 sec
✓	2	18:50:08	DROP PROCEDURE <code>all_info_jugadores_barcelona</code>	0 row(s) affected	0,00040 sec

Procedimientos almacenados





TRIGGERS



Triggers o disparadores

- Un trigger es un código que se va a ejecutar cuando se ejecute "algo" o antes de que se ejecute ese "algo"
- Por ejemplo, un trigger se puede ejecutar **antes (BEFORE)** o **después (AFTER)** de que alguien **inserte (INSERT)**, **actualice (UPDATE)** o **elimine (DELETE)** datos en una tabla.
- Se usan para tareas de mantenimiento y administración en base de datos.



Triggers o disparadores

- Ej: Un usuario inserta en una tabla un producto y nosotros queremos que se desencadene un proceso que inserte en otra tabla el nombre del usuario que insertó ese producto, que código de producto inserto y en qué fecha lo hizo.



Triggers o disparadores

- Esto se haria **DESPUES** de insertar el producto ya que **ANTES** no existe el codigo de producto insertado

COD_PRODUCTO	DETALLE_PRODUCTO
1	Mesa Algarrobo

USUARIO_INSERT	COD_PRODUCTO	FECHA_INSERT
Juan Perez	1	24/08/2020 20:35



Triggers o disparadores

- Ej: Queremos verificar que cuando se inserte un usuario la edad sea un número positivo



Triggers o disparadores

- Esto se haría **ANTES** de hacer el insert

Juan	Perez	-15
Juan	Perez	30

ID_USUARIO	NOMBRE	APELLIDO	EDAD
1	Juan	Perez	30



Triggers o disparadores

EJ: Cada vez que alguien inserte un jugador en la tabla jugador, quiero que se desencadene un trigger que inserte en otra tabla de auditoría quien insertó el dato, que id de jugador inserto y en qué fecha lo hizo.



Triggers o disparadores

Lo primero que hay que hacer es crear la tabla donde va a insertar el Trigger luego del insert:

```
CREATE TABLE insert_jugadores(  
    id_insert_jugadores INT NOT NULL AUTO_INCREMENT,  
    usuario_inserto VARCHAR(50) NOT NULL,  
    id_jugador INT NOT NULL,  
    fecha_insercion DATETIME,  
    PRIMARY KEY (id_insert_jugadores)  
)
```

	#	Time	Action	Message	Duration / Fetch
✓	1	19:49:38	CREATE TABLE insert_jugadores(id_insert_jugadores INT ... 0 row(s) affected	0,280 sec



Triggers o disparadores

```
CREATE TRIGGER auditoria_insert_jugadores AFTER INSERT ON jugador FOR EACH ROW INSERT INTO  
insert_jugadores  
(usuario_inserto, id_jugador, fecha_insercion) values (CURRENT_USER(), NEW.id_jugador, NOW())
```

#	Time	Action	Message	Duration / Fetch
✓ 1	19:54:21	CREATE TRIGGER auditoria_insert_jugadores AFTER INSERT ON...	0 row(s) affected	0,067 sec



Triggers o disparadores

SHOW TRIGGERS

#	Trigger	Event	Table	Statement	Timing	Created	sql_mode	Definer	character_set_client	collation_connection	Database Collation
1	auditoria_insert_jugadores	INSERT	jugador	INSERT INTO insert_jugadores (usuario_inserto, id_jugador, fecha...	AFTER	2020-08-23 19:41:23.78	STRICT_TRANS_TABLES,STRI...	curso@localhost	utf8	utf8_general_ci	utf8_general_ci

Triggers o disparadores

INSERT INTO jugador (nombre, apellido, sueldo_mensual, nacionalidad_id_nacionalidad, equipo_id_equipo, posicion_id_posicion) VALUES ('Robert', 'Lewandowski', 150000, 9, 8, 5)



#	Time	Action	Message	Duration / Fetch
1	19:55:36	INSERT INTO jugador (nombre, apellido, sueldo_mensual, naci...	1 row(s) affected	0,030 sec







SELECT * FROM jugador

#	id_jugador	nombre	apellido	sueldo_mensual	nacionalidad_id_nacionalidad	equipo_id_equipo	posicion_id_posicion
1	1	Lionel Andres	Messi	200000	1	1	1
2	2	Cristiano	Ronaldo	200000	2	2	2
3	3	Neymar	Jr	55000	3	3	2
4	4	Jan	Oblak	44000	4	4	3
5	5	Eden	Hazard	63000	5	5	2
6	6	Marc-Andre	ter Stegen	60000	6	1	3
7	7	Virgil	Van Dijk	58000	7	6	4
8	8	Paulo	Dybala	62000	1	2	6
9	9	Scott	Carson	45000	8	7	3
10	10	Kyle	Walker	47000	8	7	4
11	11	Sergio	Aguero	62000	1	7	5
12	12	Robert	Lewando...	150000	9	8	5


Triggers o disparadores


SELECT * FROM insert_jugadores

Result Grid   Filter Rows:

Edit:    Export/Import:   Wrap Cell Content: 

#	id_insert_jugadores	usuario_inserto	id_jugador	fecha_insercion
1	1	curso@localhost	12	2020-08-23 19:55:36
*	NULL	NULL	NULL	NULL

insert_jugadores 10 

Action Output 

#	Time	Action	Message	Duration / Fetch
1	19:57:31	select * from insert_jugadores LIMIT 0, 1000	1 row(s) returned	0,00042 sec / 0,000...



Triggers o disparadores

EJ: Cada vez que alguien modifique los datos de un jugador, en este caso el sueldo_mensual, que se desencadene un trigger que inserte en otra tabla el usuario que hizo la modificación, el id del jugador modificado, el sueldo_mensual antes, el sueldo_mensual después y la fecha en que se realizó la transacción.

Triggers o disparadores

Lo primero que hay que hacer es crear la tabla donde va a insertar el Trigger luego del update:

```
CREATE TABLE update_jugadores(  
    id_update_jugadores INT NOT NULL AUTO_INCREMENT,  
    usuario_updateo VARCHAR(50) NOT NULL,  
    id_jugador_modificado INT NOT NULL,  
    sueldo_mensual_antes INT NOT NULL,  
    sueldo_mensual_despues INT NOT NULL,  
    fecha_modificacion DATETIME,  
    PRIMARY KEY (id_update_jugadores)  
)
```

#	Time	Action	Message	Duration / Fetch
✓ 1	20:38:45	create table update_jugadores(id_update_jugadores INT ... 0 row(s) affected	0,247 sec



Triggers o disparadores

```
CREATE TRIGGER auditoria_update_jugadores BEFORE UPDATE ON jugador FOR EACH ROW INSERT  
INTO update_jugadores  
(usuario_updateo, id_jugador_modificado, sueldo_mensual_antes, sueldo_mensual_despues,  
fecha_modificacion)  
values (CURRENT_USER(), NEW.id_jugador, OLD.sueldo_mensual, NEW.sueldo_mensual, NOW())
```

#	Time	Action	Message	Duration / Fetch
✓ 1	20:43:38	CREATE TRIGGER auditoria_update_jugadores BEFORE UPDAT...	0 row(s) affected	0,874 sec











Triggers o disparadores

SHOW TRIGGERS


#	Trigger	Event	Table	Statement	Timing	Created	sql_mode	Definer	character_set_client	collation_connection	Database Collation
1	auditoria_update_jugad...	UPD...	jugador	INSERT INTO update_jugadores (usuario_updateo, id_jugador_mod... values (CURRENT_USER(), NE...	BEF...	2020-08-23 20:43:38.48	ONLY_FULL_GROUP_BY,STRIC...	curso@localhost	utf8	utf8_general_ci	utf8_general_ci


Triggers o disparadores

SELECT * FROM `update_jugador`

Result Grid  Filter Rows:  Edit:    Export/Import:   Wrap Cell Content: 

#	id_insert_jugadores	usuario_inserto	id_jugador	fecha_insercion
1	1	curso@localhost	12	2020-08-23 19:55:36
*	NULL	NULL	NULL	NULL

insert_jugadores 10 

Action Output 

#	Time	Action	Message	Duration / Fetch
1	19:57:31	select * from insert_jugadores LIMIT 0, 1000	1 row(s) returned	0,00042 sec / 0,000...



Triggers o disparadores

```
DROP TRIGGER insert_jugadores;  
DROP TRIGGER update_jugadores;
```



BACKUP BD



Backup BD

```
mysqldump -u curso -p alquileres > alquiler.sql
```



RESTAURAR BD



Restaurar BD

(Si se elimina la base de datos, hay que crearla y luego ejecutar el script)

```
mysql -u curso -p alquileres < alquiler.sql
```



FIN

