

# VISTAS Y PERMISOS EN UNA BASE DE DATOS

Trabajo de Alejandro Sainz Sainz

BD-  
ACTIVIDAD 4.3

COMIENZO .....	3
CONSULTA Y VISTA NUMERO 1 .....	3
CONSULTA Y VISTA NÚMERO 2 .....	5
CONSULTA Y VISTA NÚMERO 3 .....	6
CONSULTA Y VISTA NUMERO 4 .....	7
PRUEBAS CON USUARIOS ROLES Y PERMISOS .....	8

Alejandro Sainz Sainz

1 Vista 1 .....	3
2 Resultado Vista 1 .....	4
3 Vista 2 .....	5
4 Resultado vista 2.....	5
5 Consulta y Vista 3 .....	6
6 Resultado vista 3.....	6
7 Consulta y vista 4.....	7
8 Resultados Vista 4.....	7
9 Creación de Usuarios y Contraseñas .....	8
10 Creación de Role .....	8
11 Asignación de Roles a Usuarios.....	8
12 Role por defecto .....	9
13 Grants .....	9
14 Drops.....	9
15 Login con usuario estandar .....	10
16 Script de usuario .....	10
17 Show DB.....	10
18 Show tables.....	11
19 Resultado Show tables.....	11
20 Select vistas.....	11
21 Select 1 .....	12
22 Select 2.....	12

## COMIENZO

---

En esta actividad realizaremos una serie de actividades que son una continuación de todo lo que hemos hecho hasta ahora relativo a consultas, vistas, usuarios, roles y permisos.

Así que sin más, vamos a ello.

### CONSULTA Y VISTA NUMERO 1

Como ya cogí un poco de soltura en el ejercicio anterior, algunas vistas puedo generarlas desde el comienzo con la línea de crear vista, pero por si acaso sigo creando la mayoría de ellas generando la consulta primero, así es más fácil comprobar los resultados. De todas formas, para ahorrar tiempo, lo que haré será mostrar los comandos completos, incluida la creación de la vista.

La primera vista nos pide que mostremos todos los empleados y su antigüedad en la empresa.

```
• create or replace view antiguedadTrabajadores as  
  select e.first_name, e.last_name, timestampdiff(year, e.hire_date, current_date())  
  from employees e;
```

#### 1 Vista 1

Para esta vista sólo he tenido que seleccionar el nombre y apellido de los trabajadores. Para calcular la antigüedad he usado la función `timestampdiff`. Como fechas se usarán la fecha de contratación y la fecha actual, pero para mostrar sólo necesito los años, por lo tanto, como primer parámetro uso `year`. Se me hace siempre raro en este tipo de funciones se indique primero la fecha más baja y luego la más actual. Yo para mi gusto es más intuitivo de la otra forma.

Una vez creada tendré que usar un `select *` de esta vista para ver el resultado. Podría añadir un `order by` la diferencia de fechas, pero como lo haría de toda la tabla he decidido dejarlo así.

Alejandro Sainz Sainz

	first_name	last_name	timestampdiff(year, e.hire_date, current_date())
▶	Georgi	Facello	38
	Bezalel	S 38	39
	Parto	Bamford	38
	Chirstian	Koblick	38
	Kyoichi	Maliniak	35
	Anneke	Preuslin	35

Trabajadores 1 x

2 Resultado Vista 1

Como vemos aquí, obtenemos el resultado que queremos. Eso sí, después de ver el resultado he decido ponerle alias al campo de antigüedad, que tal como está es hasta duro de mirar.

## CONSULTA Y VISTA NÚMERO 2

Para la segunda vista necesitamos obtener los cambios en los salarios de los trabajadores, así como las fechas correspondientes de inicio y fin de obtención de esos salarios.

Esta es una consulta bastante sencillita, paso directamente a mostrar el comando.

```
create or replace view evolucionSalarioTrabajadores as
select e.emp_no, e.first_name, e.last_name, s.salary, s.from_date, s.to_date
from employees e
join salaries s on e.emp_no = s.emp_no
order by e.emp_no;
```

### 3 Vista 2

Nada que no hayamos visto ya hasta ahora, bastante sencilla. No hay nada adicional que explicar, salvo que en este caso si he optado por ordenar por el número de empleado, más que nada, porque hace más fácil la comprobación al observar la consulta.

10001	Georgi	Facello	60117	1986-06-26	1987-06-26
10001	Georgi	Facello	62102	1987-06-26	1988-06-25
10001	Georgi	Facello	66074	1988-06-25	1989-06-25
10001	Georgi	Facello	66596	1989-06-25	1990-06-25
10001	Georgi	Facello	66961	1990-06-25	1991-06-25
10001	Georgi	Facello	71046	1991-06-25	1992-06-24
10001	Georgi	Facello	74822	1992-06-24	1993-06-24

### 4 Resultado vista 2

Ejecutando un comando SELECT sobre la vista vemos el resultado. Como digo, con el order by se hace mucho más sencillo ver bien que hemos obtenido lo que nos interesa.

## CONSULTA Y VISTA NÚMERO 3

Vista que muestra el jefe actual de cada departamento con el nombre del departamento al que corresponde.

Esta consulta es un poco más larga que las anteriores, pero nada que no hayamos hecho ya. Lo único que hay que tener en cuenta es que la fecha\_hasta de la tabla de jefes de departamento sea igual a la fecha máxima.

```
create or replace view jefesDepartamentoActual as
select e.first_name, e.last_name, d.dept_name
from employees e
join dept_manager dem on e.emp_no = dem.emp_no
join dept_emp de on e.emp_no = de.emp_no
join departments d on de.dept_no = d.dept_no
where dem.to_date = '9999-01-01';
```

### 5 Consulta y Vista 3

Lo dicho, un poco más larga pero lo único que no se nos puede olvidar es la clausula del where. Eso sí, supongo que por los inserts de la tabla, el resultado es un poco raro.

	first_name	last_name	dept_name
►	Parto	Bamford	Production
	Saniya	Kalloufi	Development
	Yinghua	Dredge	Finance
	Yinghua	Dredge	Sales

### 6 Resultado vista 3

Como ya digo, un poco raro, dado que una persona es jefe o jefa de dos departamentos al mismo tiempo. Pero en principio, creo que es el resultado que buscaba.

## CONSULTA Y VISTA NUMERO 4

Para la última consulta, de la misma forma que hicimos con los salarios, se nos pide que hagamos con los títulos. Debemos de mostrar la evolución de los títulos de los trabajadores incluyendo las fechas en las que ostentaban dichos títulos.

```
• create or replace view evolucionTitulos as
  select e.first_name, e.last_name, t.title, t.from_date as 'Fecha Desde', t.to_date as 'Fecha Hasta'
  from employees e
  join titles t on e.emp_no = t.emp_no
  order by e.emp_no;
```

### 7 Consulta y vista 4

De nuevo en esta vista uso el order by, pues hace más fácil la comprobación de los que buscamos.

Como en consultas anteriores me parece que no queda muy bien el from\_date y el to\_date les he dado alias, más que nada, por legibilidad.

			Desde	Hasta
Georgi	Facello	Senior Engineer	1986-06-26	9999-01-01
Bezalet	Simmel	Staff	1996-08-03	9999-01-01
Parto	Bamford	Senior Engineer	1995-12-03	9999-01-01
Chirstian	Koblick	Engineer	1986-12-01	1995-12-01
Chirstian	Koblick	Senior Engineer	1995-12-01	9999-01-01
Kyoichi	Maliniak	Senior Staff	1996-09-12	9999-01-01

### 8 Resultados Vista 4

Aquí los resultados. Como podemos comprobar, al haber usado el order by nos es más fácil ver los cambios en algunos de los trabajadores.



## PRUEBAS CON USUARIOS ROLES Y PERMISOS

De esto ya había hecho alguna cosa adicional en la actividad anterior, pero ahora en esta he entrado un poco más en profundidad. Así que vamos paso a paso.

De primeras se nos pide que creamos dos usuarios, deptrrhh y deptconta.

```
1  
• create user deptrrhh@localhost identified by 'recursos';  
• create user deptconta@localhost identified by 'conta';
```

### 9 Creación de Usuarios y Contraseñas

Aquí podemos ver la creación de los dos usuarios que nos indica el ejercicio.

También se nos pide que creamos un role, llamado administración.

```
5 • create role administracion;  
7
```

### 10 Creación de Role

Se nos pide que añadamos los usuarios creados al role que también hemos creado.

```
• grant administracion to deptrrhh@localhost, deptconta@localhost;
```

### 11 Asignación de Roles a Usuarios

Con este comando podemos asignar el role a los usuarios creados.

Ahora en este momento se nos pedía que diésemos permisos de select al usuario deptrrhh sobre la primera consulta, al usuario deptconta sobre la última y al role administración sobre la segunda.

Pero aquí me surgió un problema. Al hacer después las pruebas, a ninguno de los usuarios se les permitía realizar tareas sobre la segunda vista, cosa que no debería de ser así.

Alejandro Sainz Sainz

Investigando un poco encontré que debía de hacer lo siguiente antes de continuar.

```
set default role administracion to deptrrh@localhost, deptconta@localhost;
```

## 12 Role por defecto

Tenía que indicar que administración fuese el role por defecto para los dos usuarios a los que se lo había garantizado. En principio, con esto, ya no tenía problemas.

Sabiendo esto, reordené el orden de ejecución del script, por lo tanto lo último lo muestro a continuación.

```
grant select on antiguedadTrabajadores to deptrrh@localhost;  
  
grant select on evolucionTitulos to deptconta@localhost;  
  
grant select on evolucionSalarioTrabajadores to administracion;
```

## 13 Grants

Ahora ya podía realizar los grants de permisos sobre las vistas requeridas a los usuarios que yo necesitase.

Como apunte adicional, comentar que, tuve que añadir otra serie de líneas al script para que funcionase siempre como debía.

```
drop user if exists deptrrh@localhost;  
drop user if exists deptconta@localhost;  
drop role if exists administracion;
```

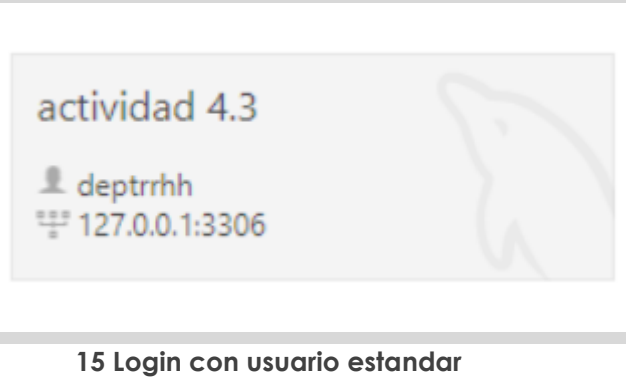
## 14 Drops

Es un comentario un poco off-topic, pero es necesario hacerlo por comodidad, sobre todo cuando es la primera vez que abres el script tras unas horas o días, que siempre va a dar error la primera vez que vuelve a intentar crear un usuario o role.

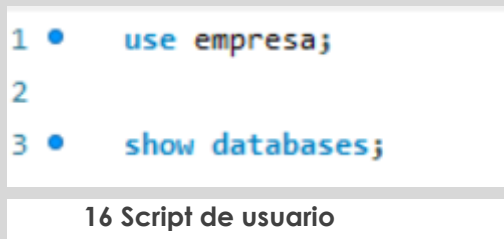
Alejandro Sainz Sainz

Una vez hecho todo esto debemos de hacer pruebas con los usuarios. En este caso creo que sólo necesitaremos realizar la prueba con uno de ellos, ya que con el otro es exactamente lo mismo pero con otra vista.

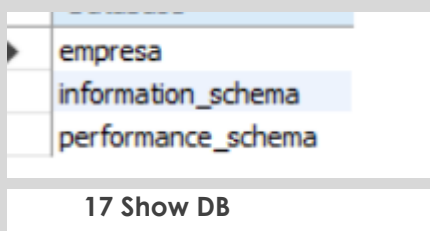
Vamos a ello.



Lo primero que hacemos es cambiar el usuario con el que nos logeamos, para probar su script personal.



Lo primero que vamos a hacer es indicar que BD vamos a usar, y que muestre todas.



Al usar show databases nos aseguramos que sólo tenga visión y acceso a las que nosotros queremos.

```
4  
5 • show tables;  
6  
18 Show tables
```

Después utilizamos el comando `show tables`. Este comando nos mostrará cuales son las tablas sobre las que el usuario puede llevar a cabo acciones.

	Tables_in_empresa
►	antiguedadtrabajadores
	evolucionsalariotrabajadores

19 Resultado Show tables

Gracias a este comando comprobamos que el usuario solo tiene acceso a la vista a la que le dimos acceso personalmente además de a la vista a la que tiene acceso por pertenecer al rol administración.

Después podemos usar el comando `select` en las dos vistas y ver que efectivamente el usuario puede tener acceso a estas vistas con ese comando.

```
• select * from antiguedadTrabajadores;  
  
• select * from evolucionSalarioTrabajadores;  
  
20 Select vistas
```

Ahora vamos a comprobar el resultado de cada uno de estos comandos.

first_name	last_name	Antigüedad
Georgi	Facello	38
Bezalel	Simmel	39
Parto	Bamford	38
Chirstian	Koblick	38
Kyoichi	Maliniak	35
Anneke	Preusig	35
Tzvetan	Zielinski	36

21 Select 1

emp_no	first_name	last_name	salary	from_date	to_date
10001	Georgi	Facello	60117	1986-06-26	1987-06-26
10001	Georgi	Facello	62102	1987-06-26	1988-06-25
10001	Georgi	Facello	66074	1988-06-25	1989-06-25
10001	Georgi	Facello	66596	1989-06-25	1990-06-25
10001	Georgi	Facello	66961	1990-06-25	1991-06-25
10001	Georgi	Facello	71046	1991-06-25	1992-06-24
10001	Georgi	Facello	74333	1992-06-24	1993-06-24

22 Select 2

Vemos que todo está como debería.

Cabe destacar que, a partir de ahora, el usuario puede también realizar consultas sobre las vistas creadas a las que se le ha concedido permiso.

Si repetimos este paso con el usuario dept\_conta, obtendremos el mismo resultado.

Con esto dicho, aquí termina esta actividad.