

SQL



KEEPCODING
Tech School

Datos de varias tablas

En el día de ayer buscábamos con una `SELECT` datos de la tabla de series o de temporadas. O bien de la tabla de persona o empleados.

¿Cómo puedo hacer para en una única sentencia SQL poder sacar datos de varias tablas?

Para ello hacemos uso de las Joins.

Por ejemplo al sacar los datos de una temporada sacar el nombre de la serie.

JOINS en SQL

Las JOINS en SQL permiten combinar registros de 2 o más tablas, relacionando dichas tablas mediante un campo común entre ellas.

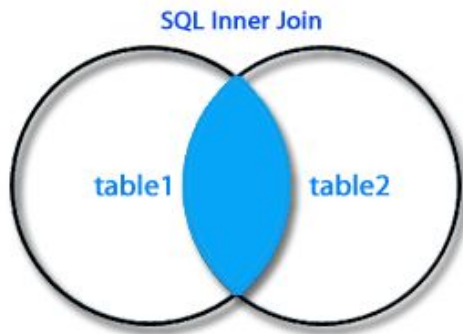
De esta forma podemos devolver datos de diferentes tablas.

Una JOIN se produce cuando 2 o más tablas se juntan en una única sentencia SQL

Ejemplo al sacar las temporadas y querer sacar el titulo de la serie, la join entre ambas tablas será mediante el campo “idserie”

Tipos de JOINS – INNER JOIN

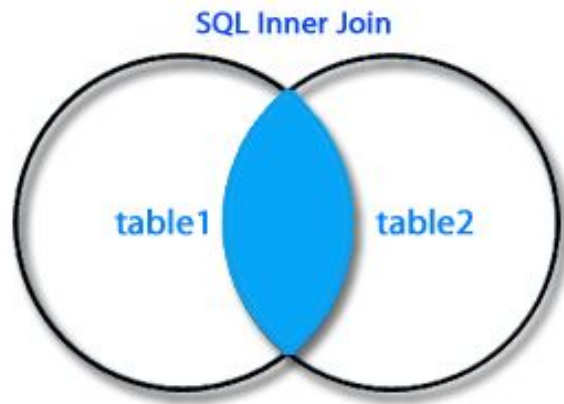
Este tipo de JOIN es la más usada y selecciona todas las filas que hay de relación entre las 2 tablas.



Tipos de JOINS – INNER JOIN

INNER JOIN (Combinación interna)

Este tipo de JOIN es la más usada y selecciona todas las filas que hay de relación entre las 2 tablas según indicamos en el predicado



Tipos de JOINS – INNER JOIN

Ejemplo series y temporadas. Sacamos las temporadas y el nombre de la serie.

Paso 1. Seleccionamos los datos de temporada. Ojo usamos alias de la tabla

```
select a.num_temporada , a.anio  
from prueba_ddl.temporadas a
```

Tipos de JOINS – INNER JOIN

Paso 2. Añadimos en el From la tabla de series y le ponemos como alias la siguiente letra del abecedario.

```
select a.num_temporada , a.anio  
from prueba_ddl.temporadas a , prueba_ddl.series b
```

Paso 3. Añadimos el campo de serie (nombre) usando el alias B

```
select a.num_temporada , a.anio, b.nombre  
from prueba_ddl.temporadas a , prueba_ddl.series b
```

Si ejecutamos esta sentencia (sin poner el predicado), estamos combinando todas las filas de las 2 tablas haciendo un **producto cartesiano**. El producto cartesiano se soluciona añadiendo el predicado o filtro de relación entre las tablas

Tipos de JOINS – INNER JOIN

Paso 3. Añadimos el predicado en el WHERE. El predicado nos indica por cada registro de la tabla de temporadas, como se relaciona con la de series. En este caso es por el campo idserie de ambas tablas.

```
select a.num_temporada , a.anio, b.nombre  
from prueba_ddl.temporadas a , prueba_ddl.series b  
where a.idserie = b.idserie
```

Ahora salen las temporadas y el nombre de la serie, sin producirse el producto cartesiano.

Tipos de JOINS – INNER JOIN

El inner join se puede escribir de 2 formas siendo lo mismo ambas.

```
select a.num_temporada , a.anio, b.nombre  
from prueba_ddl.temporadas a , prueba_ddl.series b  
where a.idserie = b.idserie
```

```
select a.num_temporada , a.anio, b.nombre  
from prueba_ddl.temporadas a inner join prueba_ddl.series b  
on a.idserie = b.idserie
```

Tipos de JOINS – OUTER JOIN

OUTER JOIN (Combinación externa)

Mediante los outer join no es necesario que una fila de una tabla tenga relación con la de otra tabla que se combina.

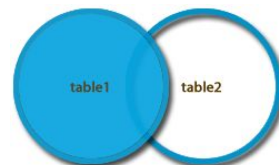
Esto se utiliza cuando combinamos una tabla 1 y 2, pero puede o no haber datos en la tabla 2. El resultado es todos los datos de la tabla1 y de la tabla 2 solo si tienen datos, o viceversa.

Uso real: Cuando una tabla1 puede o no tener datos para un registro de la tabla 2 o viceversa.

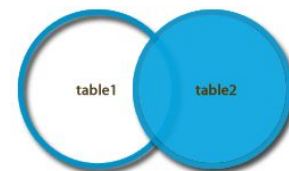
Tipos de JOINS – OUTER JOIN

Existen principalmente 2 tipos de outer join:

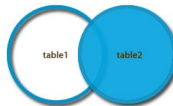
1. Left join. Devuelve todas las filas de la tabla 1. y las filas de la tabla 2 se mostrarán si hay coincidencia con la de la tabla 1 (izquierda). Sino no hay valores de la tabla2 , se mostrarán las columnas a NULL



2. Right join. Devuelve todas las filas de la tabla 2. y las filas de la tabla 1 se mostrarán si hay coincidencia con la de la tabla 2



Tipos de JOINS – RIGHT JOIN



Ejemplo:

1. Añadimos una serie nueva
2. Si ejecutamos la inner Join anterior, vemos que no sale la serie creada, porque no tiene temporadas.
3. Hacemos la right Join, sacamos todas las series y las temporadas que tenga

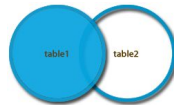
```
INSERT INTO prueba_ddl.series  
(idserie, nombre, anio_creation)  
VALUES('0005', 'Los Simpsons', 1991);
```

```
select a.num_temporada , a.anio, b.nombre  
from prueba_ddl.temporadas a inner join prueba_ddl.series b  
on a.idserie = b.idserie
```

```
select a.num_temporada , a.anio, b.nombre  
from prueba_ddl.temporadas a right join prueba_ddl.series b  
on a.idserie = b.idserie
```

num_temporada	anio	nombre
1	2.008	Breaking Bad
2	2.009	Breaking Bad
1	2.011	Juego de tronos
2	2.012	Juego de tronos
3	2.013	Juego de tronos
1	2.001	Hermanos de sangre
3	2.010	Breaking Bad
[NULL]	[NULL]	Los Simpsons

Tipos de JOINS – LEFT JOIN



Ejemplo:

1. Creamos una Inner join como tabla base series y metemos campos de temporadas
2. Como no sale los Simpson, hacemos Left Join

```
select b.num_temporada , b.anio, a.nombre , a.idserie
from prueba_ddl.series a inner join prueba_ddl.temporadas b
on a.idserie = b.idserie
```

```
select b.num_temporada , b.anio, a.nombre , a.idserie
from prueba_ddl.series a left join prueba_ddl.temporadas b
on a.idserie = b.idserie
```

num_temporada	anio	nombre	idserie
1	2.008	Breaking Bad	0001
2	2.009	Breaking Bad	0001
1	2.011	Juego de tronos	0002
2	2.012	Juego de tronos	0002
3	2.013	Juego de tronos	0002
1	2.001	Hermanos de sangre	0003
3	2.010	Breaking Bad	0001
INDUL	INDUL	Los Simpson	0005

Ejercicio1 - Enunciado

Sacar los empleados del departamento “MOBILE” con los siguientes datos:

- id persona
- nombre completo (concatenar nombre, apell1, apell2),
- fecha alta
- puesto

Nota: para concatenar en la sentencia SELECT se puede usar “||” entre las columnas, siempre que sean de tipo cadena.

Ejercicio1 - Solución

A solución al ejercicio es la sentencia SQL:

```
select b.id_person , b."name" || ' ' || b.apell1 || ' ' || b.apelli2 as nombre, a.dt_hire, a.job
from employees a, person b
where a.id_person = b.id_person and a.department = 'MOBILE'
```

Y Hay 82 empleados que deben salir en el resultado

	id_person	nombre	dt_hire	job
56	000371	JUAN JOSE CAMPOS FRANCO	2001-06-13	ANDROID SENIOF
57	000372	MATTHEW FUENTES-RIVERA DOMINGUEZ	2001-11-05	ANDROID SENIOF
58	000373	CRISTINA TOSTADO GOMEZ	2002-08-30	ANDROID SENIOF
59	000375	PATRICIA GUTIERREZ DOMINGUEZ	2005-05-21	ANDROID SENIOF
60	000376	JAIME ALONSO CARRASCO	1997-07-13	PROGRAMADOR
61	000380	GONZALO SUAREZ VAZQUEZ	2005-05-28	ANDROID SENIOF
62	000460	OLGA MUÑOZ CHUECOS	2002-06-20	IOS JUNIOR
63	000465	JORDI AL DAIFI ILARZA	2001-09-22	IOS SENIOR
64	000467	PAULA ESTEVEZ (ant) HERNANDEZ	2008-12-30	IOS SENIOR
65	000469	FERNANDO GIL NIETO	2000-08-04	IOS SENIOR
66	000470	ALEJANDRA LUQUE JUAN	1999-08-30	IOS SENIOR
67	000471	MARIA ERIKA GRELA NULL	2000-03-16	IOS SENIOR
68	000478	NURIA JUÁREZ RIVERO	1990-04-17	PROGRAMADOR
69	000493	SILVIA MARTINEZ BERMEJO	1997-08-27	PROGRAMADOR
70	000505	JAVIER CERRAJERO MIRANDA	1997-07-07	PROGRAMADOR
71	000529	MONICA ANDREA SIRERA PEREZ	2000-05-18	ANDROID SENIOF
72	000531	AMAYA GOMEZ-VILLABOA NULL	2001-03-14	ANDROID SENIOF
73	000536	ESTER CHIRONA GIL	2001-02-14	ANDROID SENIOF
74	000539	LAURA ALZUARTE MARTINEZ	1999-07-18	ANDROID SENIOF
75	000541	LAURA CATALA CEJAS	1999-08-31	ANDROID SENIOF
76	000548	JUAN JOSE ANTQ ROSSI UNCAL	2000-06-19	ANDROID SENIOF
77	000605	LORENA ANGULO LOPEZ	1999-12-28	IOS SENIOR
78	000609	EVA AGUILERA FERNANDEZ	2000-11-17	IOS SENIOR
79	000610	AMELIA COLILLA RIESCO	2005-06-21	IOS SENIOR
80	000612	MARIA JOSE GUIJARRO VERDES	1999-11-26	IOS SENIOR
81	000613	LAURA JIMENEZ LORENZO	2005-03-02	IOS SENIOR
82	000617	MARIA CONCEPCION GOIRIZ SERRANO (A	2000-06-27	IOS JUNIOR



Normalización Real Base de datos

Modelo actual

- Actualmente tenemos un modelo con 2 tablas
 - Personas. Una persona solo puede estar aquí 1 vez
 - Empleados. Una persona puede ser empleado N veces en el tiempo.
 - Entra a la empresa y se va.
 - Entra a la empresa y se va.
 - Pluriempleados. Empleados que tienen 2 contratos a la vez en la misma empresa:
 - Ejemplo sector limpieza. Cada centro de trabajo que se limpia es un Empleado.
Hay personas que pueden tener hasta 10 registros en empleado a la vez en alta.
- Veamos el modelo
 - Modelo ER

Modelo actual □ No normalizado

person	
PK	<u>id_person</u>
	name
	apell1
	apell2
	dt_birth
	type_doc
	legal_id
	email
	phone
	gender
	description

employees	
PK,FK1	<u>id_Person</u>
PK	<u>dt_hire</u>
	dt_termination
	reason_hire
	reason_termination
	dt_seniority
	employee_type
	job
	departament
	company
	salary_gross_annual
	salary_variable_annual
	id_currency

Default '4000-01-001'

Default 'Sin motivo'

Default 'Sin motivo'

Default 'Empleado'

Default 0

Default 0

Default 'EUR'

```
create table public.person
(
  id_person varchar(10) not null,
  name varchar(50) not null,
  apell1 varchar(50) not null,
  apell2 varchar(50) not null,
  dt_birth date not null,
  type_doc varchar(20) not null,
  legal_id varchar(20) not null,
  email varchar(255) null,
  phone varchar(20) null,
  gender varchar(20) not null,
  description varchar(512) null
);
```

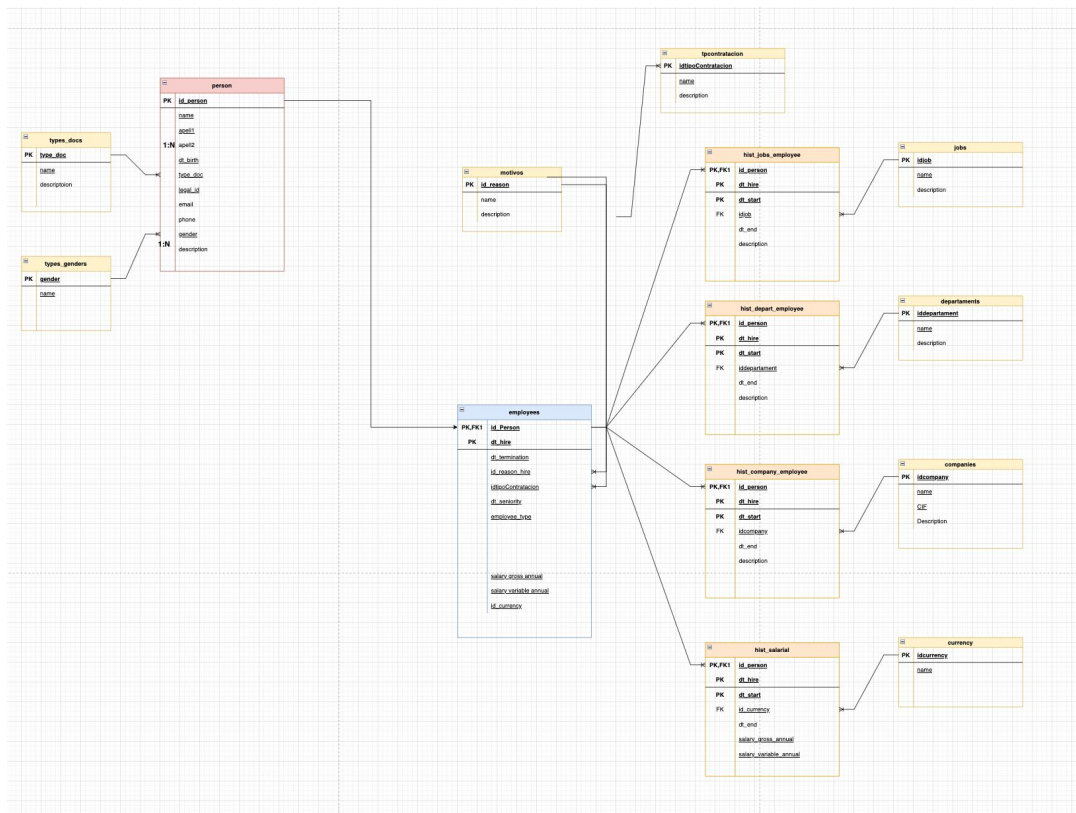
```
alter table public.person
add constraint person_PK primary key (id_person);
```

```
create table public.employees
(
  id_person varchar(10) not null,
  dt_hire date not null,
  dt_termination date null default '4000-01-01',
  reason_hire varchar(200) null default 'Sin motivo',
  reason_termination varchar(200) null default 'Sin motivo',
  dt_seniority date not null,
  employee_type varchar(200) not null default 'Empleado',
  job varchar(200) not null,
  department varchar(200) not null,
  company varchar(200) not null,
  salary_gross_annual numeric(12,5) null default 0,
  salary_variable_annual numeric(12,5) null default 0,
  id_currency varchar(10) default 'EUR'
);
```



Vamos a normalizar el modelo!

Ya tenemos el modelo ER!





**Vamos a crear el modelo con SQL y cargar
los datos desde el origen de datos de
“public”**

A background image of the anime character Goku from Dragon Ball Z. He is shown from the chest up, wearing his signature orange gi with a blue undershirt and a circular emblem on the chest. He has his signature spiky black hair and is smiling broadly with his eyes closed. His arms are raised in a celebratory gesture, with his fists clenched. The background is a bright blue sky with white clouds. The text "Ya somos unos super especialistas en SQL!!!" is overlaid in the center in a bold, black, sans-serif font.

Ya somos unos super especialistas en SQL!!!



KEEPCODING

Tech School

Madrid | Barcelona | Bogotá

Jose Luis Bustos Esteban
bejl@outlook.es