



Universidad Nacional del Litoral
Facultad de Ingeniería y Ciencias Hídricas
Departamento de Informática

Bases de Datos

SQL: Guía de Trabajo Nro. 3
Consultas avanzadas
Ejercicios

Ejercicio 1. Obtenga un listado de apellidos y nombres de autores junto a los títulos de las publicaciones de su autoría. Ordene el listado por apellido del autor.

au_lname	au_fname	Titulo
Bennet	Abraham	The Busy Executive's Database
Blotchet-Halls	Reginald	Fifty Years in Buckingham Pala
Carson	Cheryl	But Is It User Friendly?
DeFrance	Michel	The Gourmet Microwave Second E
del Castillo	Innes	Silicon Valley Gastronomic Tre
Dull	Ann	Secrets of Silicon Valley
Green	Marjorie	The Busy Executive's Database
Green	Marjorie	You Can Combat Computer Stress



Guía para resolver la consulta

Los datos solicitados residen en tablas diferentes. Extraemos el apellido y nombre de los autores de la tabla Authors, pero necesitamos el título de las publicaciones, de la tabla Titles. Tenemos que hacer un JOIN que abarque las tablas necesarias.

Luego, ordenar la relación resultado por apellido del autor.

Ejercicio 2. Obtenga un listado que incluya los nombres de las editoriales (tabla `publishers`) y los nombres y apellidos de sus empleados (tabla `employee`) pero sólo para los empleados con `job_level` de 200 o más.

pub_name	Empleado	job_lvl
Scootney Books	Francisco Chang	227
Scootney Books	Philip Cramer	215
Luceme Publishing	Carlos Hemandez	211
New Moon Books	Matti Karttunen	220
Ramona Publishers	Maria Pontes	246

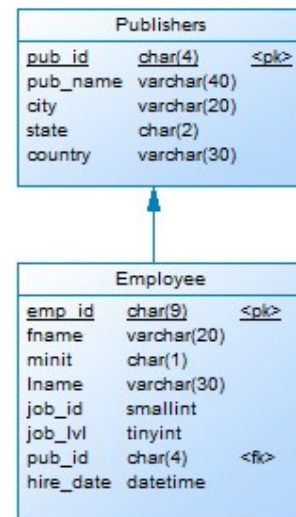


Guía para resolver la consulta

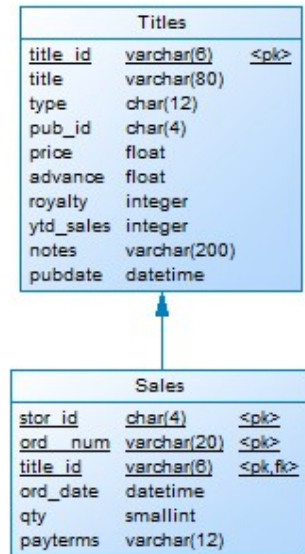
Recordemos la relación entre Editoriales y Empleados:

Aquí también necesitamos hacer un JOIN, pero además agregar una condición que deben cumplir los empleados a recuperar.

También, la columna "Empleado" en la lista de salida del SELECT es una concatenación del apellido y nombre del empleado, separados por un espacio.



Ejercicio 3. Recordemos que tabla `sales` contiene información de ventas de publicaciones. Cada venta era identificada unívocamente por un número de orden (`ord_num`), el almacén donde se produjo (`stor_id`) y la publicación vendida (`title_id`). La venta poseía también la fecha de venta (`ord_date`) y la cantidad vendida de la publicación (`qty`).



Se desea obtener un listado como el siguiente, que muestre los ingresos (precio de publicación * cantidad vendida) que ha proporcionado cada autor a partir de las ventas de sus publicaciones. Ordene el listado por orden descendente de ingresos.

au_lname	au_fname	Ingresos
Ringer	Albert	1357.6000
Ringer	Anne	1302.2000
Dull	Ann	1000.0000
Hunter	Sheryl	1000.0000
Panteley	Sylvia	838.0000
MacFeather	Stearns	730.5500



Guía para resolver la consulta

Necesitamos hacer un JOIN que nos permita vincular los datos de Ventas (tabla `Sales`) con los autores de las publicaciones vendidas.

Además, el atributo "Ingresos" en la lista de salida del SELECT es un atributo calculado. Ese atributo calculado...¿es un atributo de grupo?
¿Cuál es el grupo?

Por último, el atributo "Ingresos" debe ser ordenado de manera descendente.

Ejercicio 4. Obtenga los tipos de publicaciones (columna `type`) cuya media de precio sea mayor a \$12.



Guía para resolver la consulta

¿Cómo sabemos el promedio de precios de un determinado tipo de publicación?

¿el promedio de precios es un atributo de tupla o de grupo?

Finalmente, debemos comparar esa media de precios contra un valor escalar.

Titles		
<u>title_id</u>	varchar(6)	<pk>
title	varchar(80)	
type	char(12)	
pub_id	char(4)	
price	float	
advance	float	
royalty	integer	
ytd_sales	integer	
notes	varchar(200)	
pubdate	datetime	

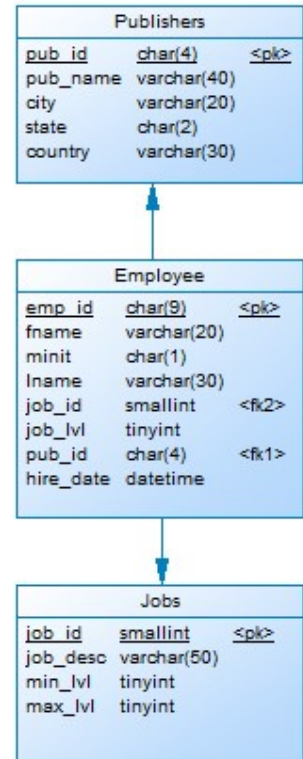
Ejercicio 5.

La tabla `employee` posee información de los empleados de cada editorial.

Por cada empleado tenemos su identificación (`emp_id`), su nombre (`fname`) y apellido (`lname`). Cada empleado pertenece a una editorial (`pub_id`) y posee una fecha de contratación (`hire_date`).

Las funciones de los empleados se describen en la tabla `jobs`. Cada empleado posee una función (`job_id`).

Obtenga el apellido y nombre del empleado contratado más recientemente.



Guía para resolver la consulta

Una solución sería buscar el valor escalar correspondiente a la fecha de contratación del empleado más "nuevo".

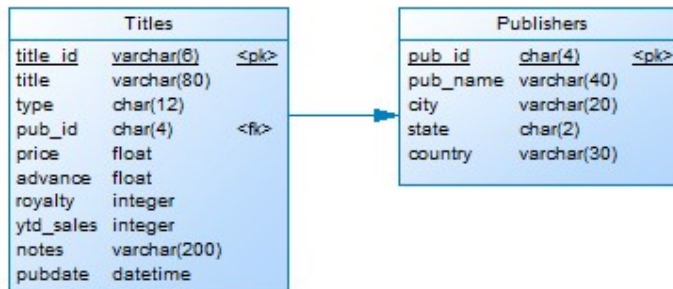
Luego, compararíamos las fechas de contratación de todos los empleados para evaluar si se trata de esa fecha más reciente.

Ver "2.1. Subqueries que producen valores escalares" en el documento 2 de este mismo tema.

Ejercicio 6. Obtenga un listado de nombres de editoriales que han editado publicaciones de tipo business.



Guía para resolver la consulta



Una solución sería buscar obtener una relación unaria que liste los códigos de editoriales que han editado títulos de tipo "business". Ese código de editorial está en la tabla titles, así que es una consulta sencilla.

Un query externo debería evaluar cada tupla de publishers comparando si su código de editorial está incluido en esa relación unaria de "códigos de editoriales que han editado publicaciones de tipo business".

Si esto fuera verdadero, debería listar el nombre de la editorial.

Ver "2.2. Condiciones que involucran relaciones" en el documento 2 de este mismo tema.

El ejercicio también se puede resolver con un JOIN.

Ejercicio 7. Obtenga un listado de títulos de las publicaciones que no se vendieron ni en 1993 ni en 1994. (Columna `ord_date` en tabla `Sales`)



Guía para resolver la consulta

Similar al ejercicio anterior.

Podría obtener una relación unaria que liste los códigos de publicaciones que se han vendido en los años 1993 y 1994. El código de publicación está en la tabla `Sales`, así que es una consulta sencilla.

Un query externo debería evaluar cada tupla de `titles` asegurándose que su código de publicación NO esté incluido en esa relación unaria de "códigos de publicaciones vendidas en 1993 y 1994". Si esa condición fuese verdadera (NO incluido), debería listar el título de la publicación.

Ver "2.2. Condiciones que involucran relaciones" en el documento 2 de este mismo tema.

Ver 4.1. Fechas en la Guía de trabajo Nro. 1.

Titles		
<u>title_id</u>	varchar(6)	<pk>
title	varchar(80)	
type	char(12)	
pub_id	char(4)	
price	float	
advance	float	
royalty	integer	
ytd_sales	integer	
notes	varchar(200)	
pubdate	datetime	

Sales		
<u>stor_id</u>	char(4)	<pk>
<u>ord_num</u>	varchar(20)	<pk>
<u>title_id</u>	varchar(6)	<pk, fk>
ord_date	datetime	
qty	smallint	
payterms	varchar(12)	

Podríamos vernos tentados a resolver esta consulta a través de un `INNER JOIN` entre las tablas de títulos (para obtener el título de la publicación) y la de ventas (a fin de evaluar la condición de fecha de la venta). La consulta sería la siguiente:

```
SELECT titles.title_id, title
FROM titles INNER JOIN sales
      ON titles.title_id = sales.title_id
WHERE YEAR(sales.ord_date) NOT IN (1993, 1994)
```

Sin embargo, esta solución no retorna los datos esperados, ya que obtenemos el listado de las publicaciones que tuvieron ventas en años diferentes a 1993 y 1994, pero no obtenemos las publicaciones que no tuvieron ventas en absoluto. Ante requerimientos de este tipo, la consulta se debe resolver a través de un predicado `IN`.

Ejercicio 8. Obtenga un listado como el siguiente con las publicaciones que poseen un precio menor que el promedio de precios de las publicaciones de la editorial a la que pertenecen.

title	pub_name	price
You Can Combat Computer Stress!	New Moon Books	2.9900
Life Without Fear	New Moon Books	7.0000
Emotional Security: A New Algorithm	New Moon Books	7.9900
The Gourmet Microwave	Binnet & Hardley	2.9900
Fifty Years in Buckingham Palace Kitchen	Binnet & Hardley	11.9500
Sushi, Anyone?	Binnet & Hardley	14.9900
Cooking with Computers: Surreptitious Ba	Algodata Infosystems	11.9500



Guía para resolver la consulta

El requerimiento nos pide columnas de titles y de publishers, así que tendremos que hacer un JOIN.

Por otro lado, SOLO debemos listar una publicación si cumple con lo solicitado: su precio debe ser menor al promedio de precios de las publicaciones de la editorial a la que pertenece.

Aquí necesitaremos comparar cada precio contra un valor escalar (el promedio de precios de las publicaciones de la editorial a la que pertenece la publicación que estamos evaluando).

Ver "2.2.5. Subqueries correlacionados" en el documento "Consultas avanzadas - Parte 2"

Ejercicio 9. La tabla `authors` posee una columna llamada `contract` con valores 0 ó 1 indicando si el autor posee o no contrato con la editora. Se desea obtener un listado como el siguiente para los autores de California (columna `state` con valor CA).

Nombre	Apellido	Posee contrato?
Johnson	White	Si
Marjorie	Green	Si
Cheryl	Carson	Si
Michael	O'Leary	Si
Dean	Straight	Si
Abraham	Bennet	Si



Guía para resolver la consulta

El siguiente es un ejemplo de los datos de autores:

Debemos aplicar una transformación a la columna `contract` de manera tal de mostrar un valor Si o No en reemplazo de los ceros y unos existentes.

Results		Messages		
	au_id	au_lname	au_fname	contract
1	172-32-1176	White	Johnson	1
2	213-46-8915	Green	Marjorie	1
3	238-95-7766	Carson	Cheryl	1
4	267-41-2394	O'Leary	Michael	1
5	274-80-9391	Straight	Dean	1
6	341-22-1782	Smith	Meander	0
7	409-56-7008	Bennet	Abraham	1

Ver "8. La expresión CASE" en el documento "Consultas avanzadas - Parte 3"

Ejercicio 10. La columna `job_lvl` indica el puntaje del empleado dentro de su área de especialización. Se desea obtener un reporte como el siguiente, ordenado por puntaje y apellido del empleado:

lname	Nivel
Ashworth	Puntaje entre 100 y 200
Brown	Puntaje entre 100 y 200
Domingues	Puntaje entre 100 y 200
...	
Chang	Puntaje mayor que 200
Cramer	Puntaje mayor que 200
Devon	Puntaje mayor que 200
...	
Schmitt	Puntaje menor que 100
Smith	Puntaje menor que 100
Tonini	Puntaje menor que 100



Guía para resolver la consulta

Aquí debemos evaluar el valor de la columna `job_lvl` para ver en que rango de valores cae:

- Entre 100 y 200
- Mayor que 200
- Menor que 100

Employee		
<u>emp_id</u>	char(9)	<pk>
fname	varchar(20)	
minit	char(1)	
lname	varchar(30)	
job_id	smallint	
job_lvl	tinyint	
pub_id	char(4)	
hire_date	datetime	

Ver "8. La expresión CASE" en el documento "Consultas avanzadas - Parte 3"