



Programación de Bases de Datos con SQL

6-4

Autouniones y Consultas Jerárquicas



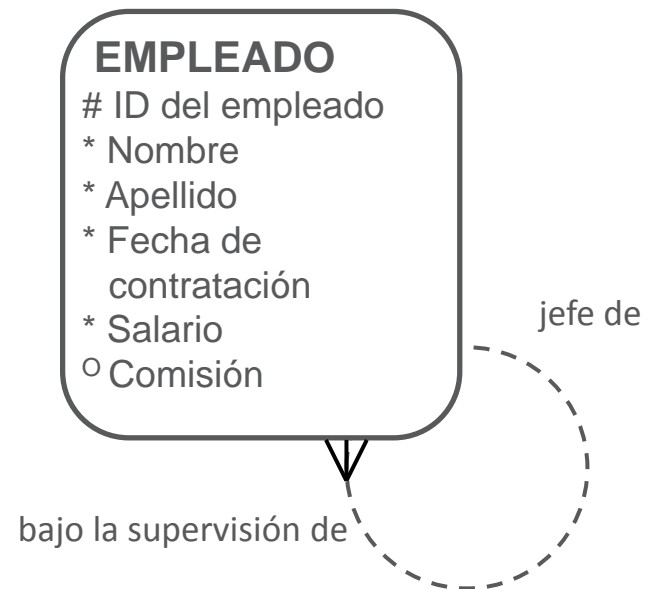
Objetivos

En esta lección se abordan los siguientes objetivos:

- Crear y ejecutar una sentencia SELECT para unir una tabla consigo misma mediante una autounión
- Interpretar el concepto de una consulta jerárquica
- Crear un informe con estructura de árbol
- Aplicar formato a datos jerárquicos
- Excluir ramas de la estructura de árbol

Objetivo

- En el modelado de datos, a veces es necesario mostrar una entidad con una relación consigo misma.
- Por ejemplo, un empleado también puede ser un jefe.
- Mostramos esto con la relación recursiva o de "oreja de cerdo".



Objetivo

- Una vez que tengamos una verdadera tabla employees, es necesario un tipo especial de unión denominada autounión para acceder a esos datos.
- Se utiliza una autounión para unir una tabla a sí misma como si se tratara de dos tablas.

```
SELECT worker.last_name || ' works for ' || manager.last_name  
  AS "Works for"  
FROM employees worker JOIN employees manager  
ON (worker.manager_id = manager.employee_id);
```

AUTOUNIÓN

- Para unir una tabla a sí misma, a la tabla se le asignan dos nombres o alias. Esto hará que la base de datos "crea" que hay dos tablas.

EMPLOYEES (worker)

employee_id	last_name	manager_id
100	King	
101	Kocher	100
102	De Haan	100
103	Hunold	102
104	Ernst	103
107	Lorentz	103
124	Mourgos	100

EMPLOYEES (manager)

employee_id	last_name
100	King
101	Kocher
102	De Haan
103	Hunold
104	Ernst
107	Lorentz
124	Mourgos

- Manager_id en la tabla worker es igual a employee_id en la tabla manager.

AUTOUNIÓN

- Seleccione los nombres de alias relacionados con la asociación de datos con esa tabla.

EMPLOYEES (worker)

employee_id	last_name	manager_id
100	King	
101	Kocher	100
102	De Haan	100
103	Hunold	102
104	Ernst	103
107	Lorentz	103
124	Mourgos	100



EMPLOYEES (manager)

employee_id	last_name
100	King
101	Kocher
102	De Haan
103	Hunold
104	Ernst
107	Lorentz
124	Mourgos



- Manager_id en la tabla worker es igual a employee_id en la tabla manager.

Ejemplo de SELF-JOIN

```
SELECT worker.last_name, worker.manager_id, manager.last_name
       AS "Manager name"
FROM employees worker JOIN employees manager
ON (worker.manager_id = manager.employee_id);
```

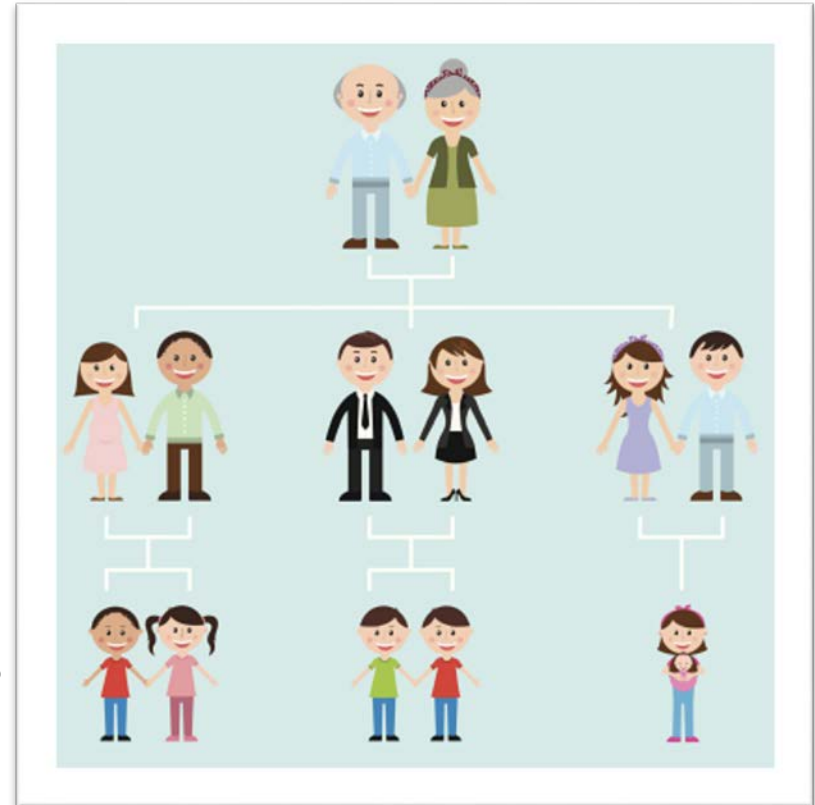
LAST_NAME	MANAGER_ID	Manager name
Kochhar	100	King
De Haan	100	King
Zlotkey	100	King
Mourgos	100	King
Hartstein	100	King
Whalen	101	Kochhar
Higgins	101	Kochhar
Hunold	102	De Haan
...

Consultas Jerárquicas

- Estrechamente relacionadas con las autouniones se encuentran las consultas jerárquicas.
- En las páginas anteriores, ha visto cómo puede utilizar las autouniones para ver al jefe directo de cada empleado.
- Con las consultas jerárquicas, también podemos ver para quién trabaja ese jefe, y así sucesivamente.
- Con este tipo de consulta, podemos crear un diagrama de organización que muestre la estructura de una compañía o de un departamento.

Consultas Jerárquicas

- Imagine un árbol familiar en el que los miembros más mayores de la familia se encuentran cerca de la base o el tronco del árbol y los más jóvenes representan las ramas del árbol.
- Las ramas pueden tener sus propias ramas y así sucesivamente.





Uso de Consultas Jerárquicas

- Con las consultas jerárquicas, puede recuperar datos según una relación jerárquica natural entre las filas de una tabla.
- Una base de datos relacional no almacena los registros de forma jerárquica.
- Sin embargo, cuando existe una relación jerárquica entre las filas de una única tabla, un proceso denominado recorrido por el árbol permite construir la jerarquía.
- Una consulta jerárquica es un método de creación de informes, con las ramas de un árbol en un determinado orden.

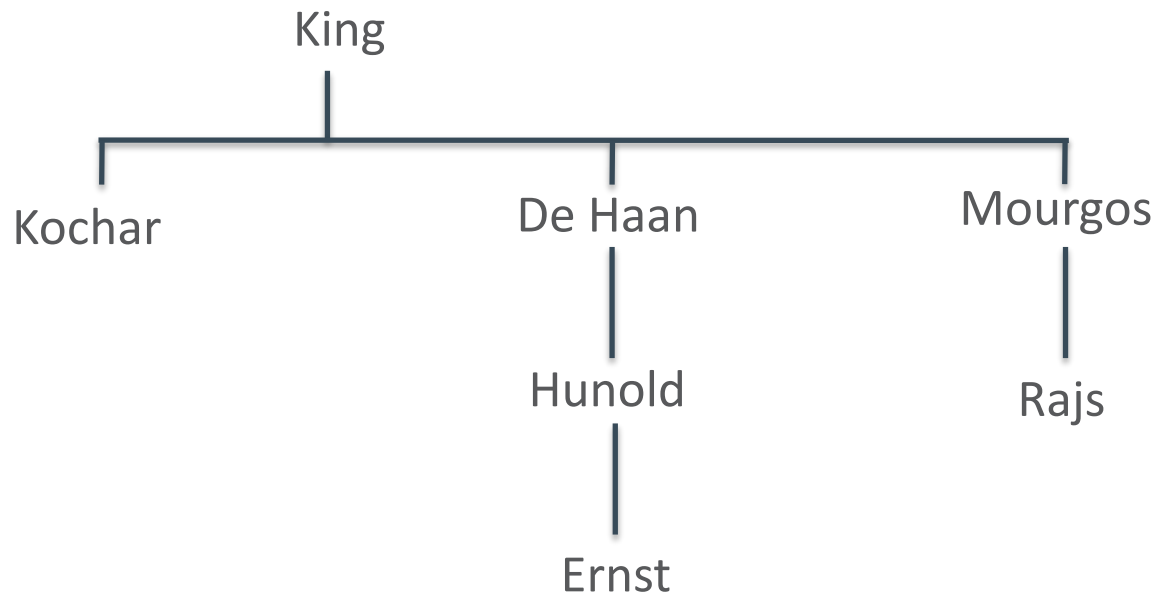
Datos de Consultas Jerárquicas

- Examine los datos de ejemplo de la tabla EMPLOYEES siguiente y observe cómo puede realizar manualmente las conexiones para ver quién trabaja para quién, empezando por Steven King y desplazándose por el árbol desde ahí.

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMM_PCT	MGR_ID	DEPT_ID
100	Steven	King	SKING	515.123.4567	17-Jun-1987	AD_PRES	24000	(null)	(null)	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-Sep-1989	AD_VP	17000	(null)	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	13-Jan-1993	AD_VP	17000	(null)	100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	03-Jan-1990	IT_PROG	9000	(null)	102	60
104	Bruce	Ernst	BERNST	590.423.4568	21-May-1991	IT_PROG	6000	(null)	103	60
124	Kevin	Mourgos	KMOURGOS	650.123.5234	16-Nov-1999	ST_MAN	5800	(null)	100	50
141	Trenna	Rajs	TRAJS	650.121.8009	17-Oct-1995	ST_CLERK	3500	(null)	124	50

Ilustración de Consultas Jerárquicas

- El diagrama de organización que podemos extraer a partir de los datos de la tabla EMPLOYEES será similar a este:



Palabras Clave de las Consultas Jerárquicas

- Las consultas jerárquicas tienen sus propias palabras clave nuevas: START WITH, CONNECT BY PRIOR y LEVEL.
- START WITH identifica qué fila se va a utilizar como raíz del árbol que se está creando; CONNECT BY PRIOR explica cómo realizar las uniones entre filas y LEVEL especifica cuántas ramas de profundidad recorrerá el árbol.



Ejemplo de Palabras Clave en Consultas Jerárquicas

```
SELECT employee_id, last_name, job_id, manager_id
FROM employees
START WITH employee_id = 100
CONNECT BY PRIOR employee_id = manager_id
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	MANAGER_ID
100	King	AD_PRES	-
101	Kochhar	AD_VP	100
200	Whalen	AD_ASST	101
205	Higgins	AC_MGR	101
206	Gietz	AC_ACCOUNT	205
102	De Haan	AD_VP	100
103	Hunold	IT_PROG	102
104	Ernst	IT_PROG	103

Otro Ejemplo de Consultas Jerárquicas

```
SELECT last_name || ' reports to ' || PRIOR last_name AS "Walk Top Down"  
FROM employees  
START WITH last_name = 'King'  
CONNECT BY PRIOR employee_id = manager_id;
```

Walk Top Down
King rinde cuentas a
Kochhar rinde cuentas a King
Whalen rinde cuentas a Kochhar
Higgins rinde cuentas a Kochhar
Gietz rinde cuentas a Higgins
De Haan rinde cuentas a King
Hunold rinde cuentas a De Haan
Ernst rinde cuentas a Hunold

Ejemplo de LEVEL en Consultas Jerárquicas

- LEVEL es una pseudocolumna que se utiliza con consultas jerárquicas y que cuenta el número de pasos que ha realizado desde la raíz del árbol.

LEVEL	Walk Top Down
1	King rinde cuentas a
2	Kochhar rinde cuentas a King
3	Whalen rinde cuentas a Kochhar
3	Higgins rinde cuentas a Kochhar
4	Gietz rinde cuentas a Higgins
2	De Haan rinde cuentas a King
3	Hunold rinde cuentas a De Haan
4	Ernst rinde cuentas a Hunold

```
SELECT LEVEL, last_name || ' reports to ' || PRIOR last_name
  AS "Walk Top Down"
FROM employees
START WITH last_name = 'King'
CONNECT BY PRIOR employee_id=manager_id;
```

Informe de Consulta Jerárquica

- Si deseara crear un informe que mostrara los niveles de dirección de la compañía, empezando por el nivel más alto y sangrando cada uno de los siguientes niveles, sería fácil hacer esto con la pseudocolumna LEVEL y la función LPAD para sangrar los empleados en función de su nivel.

```
SELECT LPAD(last_name, LENGTH(last_name)+(LEVEL*2)-2, '_')  
      AS "Org Chart"  
FROM employees  
START WITH last_name = 'King'  
CONNECT BY PRIOR employee_id = manager_id;
```

Niveles de Salida de la Consulta Jerárquica

- Como puede ver en el resultado de la derecha, cada fila se sangra con dos caracteres de subrayado por nivel.

```
SELECT LPAD(last_name, LENGTH(last_name)+  
  (LEVEL*2)-2, '_') AS "Org_Chart"  
FROM employees  
START WITH last_name = 'King'  
CONNECT BY PRIOR employee_id = manager_id;
```

Org_Chart
King
__Kochhar
___Whalen
___Higgins
____Gietz
___De Haan
___Hunold
____Ernst
____Lorentz
___Rajs
___Davies
___Matos
___Vargas
___Zlotkey
___Abel
___Taylor
___Grant
___Hartstein
___Fay

Consulta Jerárquica de Abajo a Arriba

- Como puede ver en el resultado que aparece a continuación, en este ejemplo se muestra cómo crear una consulta jerárquica de abajo a arriba, moviendo la palabra clave PRIOR a la posición posterior al signo igual y mediante el uso de 'Grant' en la cláusula START WITH.

```
SELECT LPAD(last_name, LENGTH(last_name) + (LEVEL*2)-2, '_') AS  
      ORG_CHART  
FROM employees  
START WITH last_name = 'Grant'  
CONNECT BY employee_id = PRIOR manager_id
```

ORG_CHART
Grant
__Zlotkey
____King

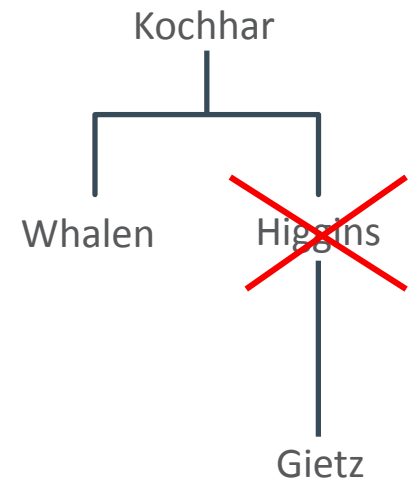
Depuración de Consultas Jerárquicas

- La depuración de ramas del árbol se puede realizar mediante la cláusula WHERE o la cláusula CONNECT BY PRIOR.
- Si se utiliza la cláusula WHERE, solo se excluye la fila especificada en la sentencia; si se utiliza la cláusula CONNECT BY PRIOR, se excluye toda la rama.

Depuración de Consultas Jerárquicas

- Por ejemplo, si desea excluir una única fila de su resultado, debería utilizar la cláusula WHERE para excluir esa fila; sin embargo, en el resultado, parecería que Gietz hubiera trabajado directamente para Kochhar, lo que no refleja la realidad.

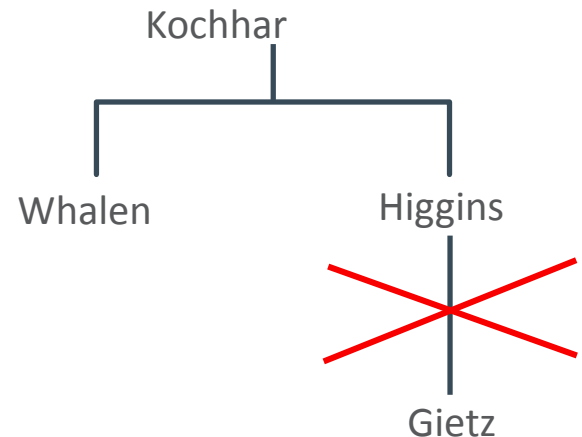
```
SELECT last_name  
FROM employees  
WHERE last_name != 'Higgins'  
START WITH last_name = 'Kochhar'  
CONNECT BY PRIOR employee_id = manager_id;
```



Depuración de Consultas Jerárquicas

- Sin embargo, si deseara excluir una fila y todas las filas por debajo de ella, debería realizar la parte de exclusión de la sentencia CONNECT BY.
- En este ejemplo que excluye a Higgins, también estamos excluyendo a Gietz del resultado.

```
SELECT last_name
FROM employees
START WITH last_name = 'Kochhar'
CONNECT BY PRIOR employee_id = manager_id
AND last_name != 'Higgins';
```



Terminología

Entre los términos clave utilizados en esta lección se incluyen:

- Connect By prior
- Consultas jerárquicas
- Nivel
- Autounión
- Start with
- Estructura del árbol
- Recorrido por el árbol
- Ramas

Resumen

En esta lección, debe haber aprendido lo siguiente:

- Crear y ejecutar una sentencia SELECT para unir una tabla consigo misma mediante una autounión
- Interpretar el concepto de una consulta jerárquica
- Crear un informe con estructura de árbol
- Aplicar formato a datos jerárquicos
- Excluir ramas de la estructura de árbol

