

ASOCIANDO TABLAS

DAW 1 – Bases de Datos

Práctica 3.5

Continuación a Consultas en bases de datos

Marco Valiente Rodríguez
Mvaliente2501@educantabria.es

Índice

Contenido

Índice	1
Planteamiento del Problema	2
a. Corrección del Código al Ejecutar	2
b. Consultas	3
1. Lista el nombre del departamento actual de cada empleado (mostrando el nombre y apellido del empleado).....	3
2. Muestra el salario actual de cada empleado y su nombre y apellidos.....	4
3. Obtén la lista completa de jefes actuales de departamento y sus títulos.....	4
4. Enumera los 3 departamentos con más empleados en la actualidad.	5
5. Enumera a todos los empleados que han trabajado en el mismo departamento durante más de 2 años.....	6
6. Calcula el salario medio por departamento entre los empleados que actualmente trabajan en él.	7
7. Muestra al empleado mejor pagado de cada departamento entre los que trabajan actualmente.	8
8. Muestra los detalles del jefe actual de departamentos con el salario más bajo.	9
9. Muestra el salario, título, nombre y apellido del primer empleado contratado por la empresa.....	10
Bibliografía	11

Planteamiento del Problema

Nos encontramos con un gran código dentro de la base de datos, que, al ejecutarlo, nos da un error. Tenemos, pues, que intentar leer y entender el código sin poder realizar una lectura general al código de por sí, ya que este código consiste en más de 1400 líneas, así que cualquier tipo de consultas van a estar hechas a la ciega.

a. Corrección del Código al Ejecutar

Tras una ejecución para detectar fallos, el error que nos salta es el siguiente:

```
✖ 11 10:04:47 INSERT INTO `dept_emp` VALUES (10001,'d005','1986-06-26','9999-01-01'), (10002,'d007','1996-08-03','9999-01-01'), (10003,'d004','1995-12-03','9999-01-01')...
```

Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('employees`.`dept_emp', CONSTRAINT `dept_emp_ibfk_1` FOREIGN ... 0.000 sec

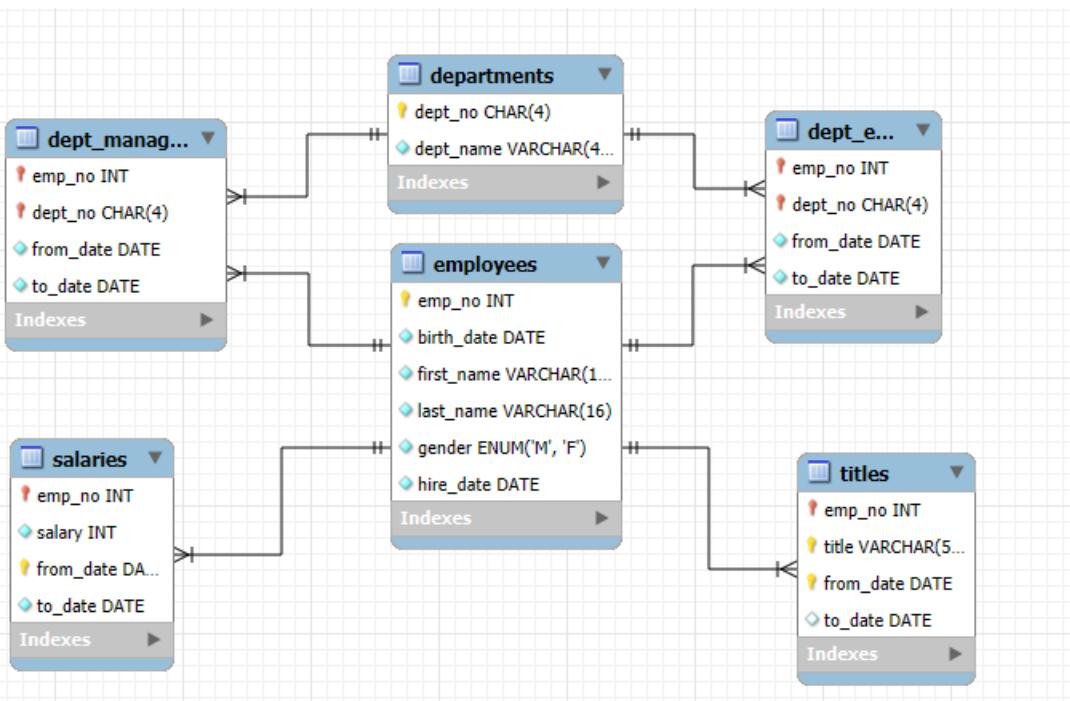
De estos datos, sabemos que el fallo ocurre en el comando “`INSERT INTO `dept_emp` VALUES`”, y que el error consiste en que no puede detectar las Foreign Key.

Tras una breve revisión, vemos que el fallo es evidente, que la tabla de `'dept_emp'` está siendo introducida los datos antes que los propios datos a los que referencia existan, por lo que lo que necesitamos hacer es copiar la inserción de datos y ejecutarla más abajo. Por debajo de employees and departments.

Output:

#	Time	Action	Message
25	10:09:18	INSERT INTO `dept_manager` VALUES (10002,'d007','1985-01-01','1991-10-01'), (10003,'d004','1991-10-01','9999-01-01'), (10006,'d005','1985-01-01','1991-10-01')	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0
26	10:09:18	INSERT INTO `dept_emp` VALUES (10001,'d005','1986-06-26','9999-01-01'), (10002,'d007','1996-08-03','9999-01-01'), (10003,'d004','1995-12-03','9999-01-01')	110 row(s) affected Records: 110 Duplicates: 0 Warnings: 0
27	10:09:18	INSERT INTO `salaries` VALUES (10001,60117,'1986-06-26','1987-06-26'), (10001,62102,'1987-06-26','1988-06-25'), (10001,66074,'1988-06-25','1989-01-01')	999 row(s) affected Records: 999 Duplicates: 0 Warnings: 0

Y efectivamente, la base de datos ha sido creada de forma correcta. Este es nuestro esquema:



b. Consultas

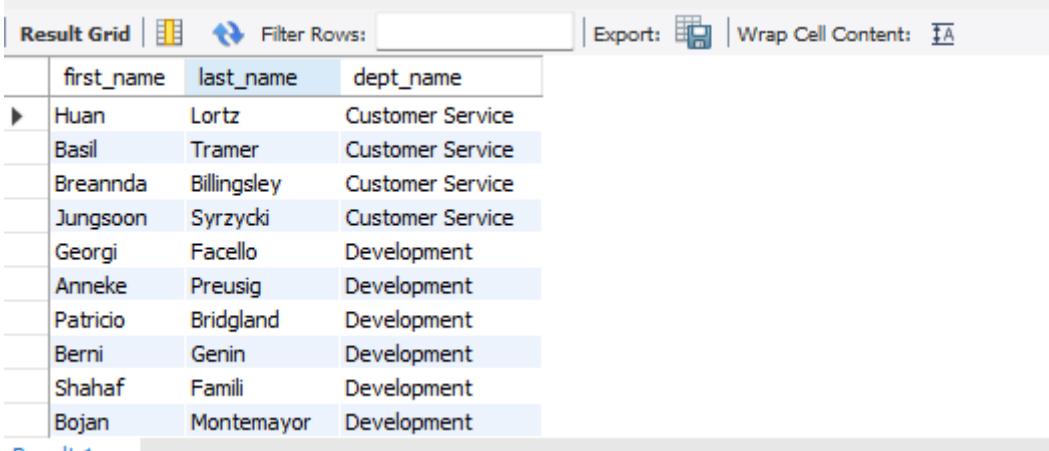
Este apartado es el que más interesa realizar, ya que es el grueso de la actividad. Para cada consulta, vamos a realizar su apropiado apartado, ya que de esta hay bastantes datos. También aplicaremos en cada apartado, exactamente cómo de válida es el resultado, para poder argumentar si igual existe algún error en el código.

1. Lista el nombre del departamento actual de cada empleado (mostrando el nombre y apellido del empleado)

Se nos piden el departamento actual de cada empleado. Para esto, vamos a tener que destacar en el uso de la palabra actual. Quiere decir, que vamos a tener que filtrar cualquier tipo de asignación a un departamento cuya fecha límite sea inferior a la actual.

Tenemos que seleccionar dept_name y el nombre y apellido de los empleados, centrándonos en el to_date siendo o “el fin de los tiempos”, la fecha ‘9999-01-01’, o que dicha fecha sea menor que la actual.

```
SELECT e.first_name, e.last_name, d.dept_name
FROM departments d
JOIN dept_emp de ON d.dept_no = de.dept_no
JOIN employees e ON e.emp_no = de.emp_no
WHERE de.to_date > CURRENT_DATE();
```



The screenshot shows a database query results grid. At the top, there are buttons for 'Result Grid' (highlighted), 'Filter Rows', 'Export', and 'Wrap Cell Content'. The grid itself has three columns: 'first_name', 'last_name', and 'dept_name'. The data rows are as follows:

	first_name	last_name	dept_name
▶	Huan	Lortz	Customer Service
	Basil	Tramer	Customer Service
	Breannda	Billingsley	Customer Service
	Jungsoon	Syrzycki	Customer Service
	Georgi	Facello	Development
	Anneke	Preusig	Development
	Patricio	Bridgland	Development
	Berni	Genin	Development
	Shahaf	Famili	Development
	Bojan	Montemayor	Development

2. Muestra el salario actual de cada empleado y su nombre y apellidos.

Para mostrar esta información, necesitamos tomar de los empleados actuales el salario actual, que solo existirá de aquellos empleados que sigan trabajando.

```
SELECT e.first_name, e.last_name, s.salary
FROM employees e
JOIN salaries s ON s.emp_no = e.emp_no
WHERE s.to_date > CURRENT_DATE();
```

The screenshot shows a database result grid with three columns: first_name, last_name, and salary. The data includes 14 rows of employee information, such as Georgi Facello (salary 88958) and Bezalel Simmel (salary 72527). The grid has a header row and a footer row with dashed lines.

	first_name	last_name	salary
►	Georgi	Facello	88958
	Bezalel	Simmel	72527
	Parto	Bamford	43311
	Chirstian	Koblick	74057
	Kyoichi	Maliniak	94692
	Anneke	Preusig	59755
	Tzvetan	Zielinski	88070
	Sumant	Peac	94409
	Duangkaew	Piveteau	80324
	Patricio	Bridgland	54423
	Eberhardt	Terkki	68901
	Berni	Genin	60598
	Kazuhito	Cappelletti	77935
	Cristinel	Bouloucos	99651
...	----

El Código es sencillo, funcional, y sin mayores complicaciones.

3. Obtén la lista completa de jefes actuales de departamento y sus títulos.

Para poder determinar cuales son los jefes actuales, necesitamos tomar de varias tablas, más en particular la tabla dept_manager, así como unirla con cada departamento para poder visualizar los datos de mejor manera.

```
SELECT d.dept_name, e.first_name, e.last_name, t.title
FROM departments d
JOIN dept_manager dm ON d.dept_no = dm.dept_no
JOIN employees e ON e.emp_no = dm.emp_no
JOIN titles t ON t.emp_no = e.emp_no
WHERE t.to_date > CURRENT_DATE();
```

No necesitamos comprobar que el jefe actual esté en la base de datos como tal, ya que el título nos sirve para esa comprobación, y asumimos que solo existe un único jefe actual en cada departamento.

The screenshot shows a database result grid with the following columns: dept_name, first_name, last_name, and title. The data is as follows:

	dept_name	first_name	last_name	title
▶	Finance	Yinghua	Dredge	Senior Staff
	Human Resources	Alejandro	Brender	Senior Staff
	Production	Parto	Bamford	Senior Engineer
	Development	Anneke	Preusig	Senior Engineer
	Sales	Bezalel	Simmel	Staff

4. Enumera los 3 departamentos con más empleados en la actualidad.

Para esta consulta, usaremos ORDER BY y LIMIT 3. Para obtener cuantos empleados, usamos COUNT(), y contamos un dato único de dept_emp, sabiendo que solo vamos a tomar los últimos. Usamos AS num_empleados para declarar como se va a poder leer más abajo en el ORDER BY. Como estamos contando casillas no únicas, sabemos que COUNT funciona.

```
SELECT d.dept_name, COUNT(de.emp_no) AS num_empleados
FROM departments d
JOIN dept_emp de ON d.dept_no = de.dept_no
WHERE de.to_date > CURRENT_DATE()
GROUP BY d.dept_no
ORDER BY num_empleados DESC
LIMIT 3;
```

Y obtenemos los 3 resultados.

The screenshot shows a database result grid with the following columns: dept_name and num_empleados. The data is as follows:

	dept_name	num_empleados
▶	Development	22
	Production	20
	Sales	11

5. Enumera a todos los empleados que han trabajado en el mismo departamento durante más de 2 años

Para esta resolución, tenemos que usar la función `TIMESTAMPDIFF()`.

```
SELECT e.first_name, e.last_name, d.dept_name, de.from_date, de.to_date
FROM employees e
JOIN dept_emp de ON e.emp_no = de.emp_no
JOIN departments d ON d.dept_no = de.dept_no
WHERE TIMESTAMPDIFF(YEAR, de.from_date, de.to_date) > 2;
```

Y obtenemos el resultado

	first_name	last_name	dept_name	from_date	to_date
▶	Mary	Sluis	Customer Service	1990-01-22	1996-11-09
	Huan	Lortz	Customer Service	1989-09-20	9999-01-01
	Basil	Tramer	Customer Service	1992-05-04	9999-01-01
	Breannda	Billingsley	Customer Service	1992-11-11	9999-01-01
	Jungsoon	Syrzycki	Customer Service	1992-03-21	9999-01-01
	Sreekrishna	Servieres	Customer Service	1989-06-29	1992-12-11
	Georgi	Facello	Development	1986-06-26	9999-01-01
	Anneke	Preusig	Development	1990-08-05	9999-01-01
	Patricia	Bridgland	Development	1992-12-18	9999-01-01
	Berni	Genin	Development	1993-12-29	9999-01-01
	Kazuhide	Peha	Development	1987-04-03	1992-07-29
	Ramzi	Erde	Development	1988-02-10	2002-07-15
	Shahaf	Famili	Development	1999-09-03	9999-01-01
	Bojan	Montema...	Development	1999-09-27	9999-01-01

En esta consulta, se incluyen aquellos empleados que ya no trabajan en esta.

6. Calcula el salario medio por departamento entre los empleados que actualmente trabajan en él.

Para calcular el Salario medio, tomamos AVG(), y un GROUP BY, para declarar que se asocien estos salarios medios con el adecuado departamento.

```
SELECT d.dept_name, AVG(s.salary)
FROM departments d
JOIN dept_emp de ON d.dept_no = de.dept_no
JOIN employees e ON e.emp_no = de.emp_no
JOIN salaries s ON s.emp_no = e.emp_no
WHERE s.to_date > CURRENT_DATE()
AND de.to_date > CURRENT_DATE()
GROUP BY d.dept_name;
```

Y el resultado es tal que

Result Grid				Filter Rows:
	dept_name	AVG(s.salary)		
▶	Customer Service	76692.7500		
	Development	64348.4545		
	Finance	94161.0000		
	Human Resources	69650.3000		
	Marketing	86096.5000		
	Production	71607.3000		
	Quality Management	83941.0000		
	Research	72375.2857		
	Sales	87200.4545		

7. Muestra al empleado mejor pagado de cada departamento entre los que trabajan actualmente.

Para esta consulta, se requiere de hacer una subconsulta.

```
SELECT d.dept_name, e.first_name, e.last_name, s.salary
FROM departments d
JOIN dept_emp de ON d.dept_no = de.dept_no
JOIN employees e ON e.emp_no = de.emp_no
JOIN salaries s ON s.emp_no = e.emp_no
WHERE de.to_date > CURRENT_DATE()
AND s.to_date > CURRENT_DATE()
AND s.salary IN (SELECT MAX(s2.salary) FROM dept_emp de2
JOIN salaries s2 ON s2.emp_no = de2.emp_no
WHERE de2.dept_no = d.dept_no
AND de2.to_date > CURRENT_DATE()
AND s2.to_date > CURRENT_DATE());
```

Y después de esto, logramos tener el resultado deseado.

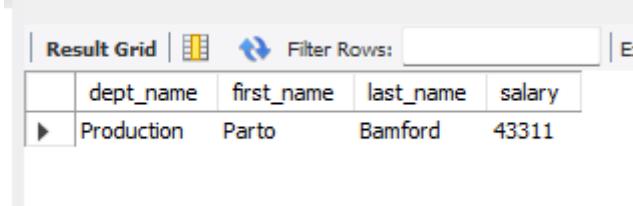
	dept_name	first_name	last_name	salary
▶	Customer Service	Jungsoon	Syrzycki	98003
	Development	Kwee	Schusler	103672
	Finance	Alejandro	McAlpine	94161
	Human Resources	Somnath	Foote	96333
	Marketing	Cristinel	Bouloucos	99651
	Production	Suzette	Pettey	96646
	Quality Management	Sumant	Peac	94409
	Research	Reuven	Garigliano	96322

8. Muestra los detalles del jefe actual de departamentos con el salario más bajo.

Para esta búsqueda, vamos a necesitar hacer una subconsulta de nuevo.

```
SELECT d.dept_name, e.first_name, e.last_name, s.salary
FROM departments d
JOIN dept_manager dm ON d.dept_no = dm.dept_no
JOIN employees e ON e.emp_no = dm.emp_no
JOIN salaries s ON s.emp_no = e.emp_no
WHERE dm.to_date > CURRENT_DATE()
AND s.to_date > CURRENT_DATE()
AND s.salary IN (SELECT MIN(s2.salary) FROM dept_manager dm2
JOIN salaries s2 ON s2.emp_no = dm2.emp_no
WHERE dm2.to_date > CURRENT_DATE())
AND s2.to_date > CURRENT_DATE());
```

Después de esta subconsulta, nos sale el único jefe actual, que es tal que



The screenshot shows a database query results grid. At the top, there are buttons for 'Result Grid' (selected), 'Filter Rows:', and an 'E' button. Below is a table with four columns: 'dept_name', 'first_name', 'last_name', and 'salary'. A single row of data is shown: Production, Parto, Bamford, 43311. There is also a navigation arrow pointing right below the first column.

	dept_name	first_name	last_name	salary
▶	Production	Parto	Bamford	43311

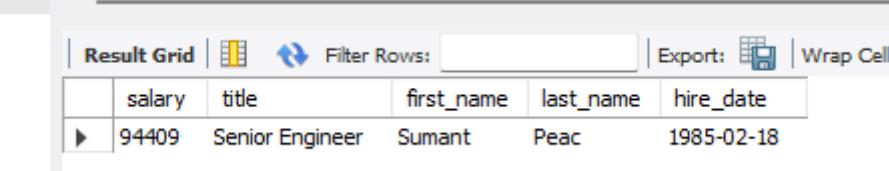
Dato que conocemos anteriormente de la búsqueda de todos los trabajadores.

9. Muestra el salario, título, nombre y apellido del primer empleado contratado por la empresa.

Terminamos realizando una consulta que requiere de una única subconsulta que podemos realizar tal que simplemente seleccionando de todos los existentes, el hire_date mínimo que podamos encontrar.

```
SELECT s.salary, t.title, e.first_name, e.last_name, e.hire_date  
FROM employees e  
JOIN salaries s ON s.emp_no = e.emp_no  
JOIN titles t ON t.emp_no = e.emp_no  
WHERE e.hire_date IN ( SELECT MIN(hire_date) FROM employees)  
AND s.to_date > CURRENT_DATE()  
AND t.to_date > CURRENT_DATE();
```

Y con esto, podemos encontrar nuestro primer empleado.



The screenshot shows a database query results grid. At the top, there are buttons for 'Result Grid' (selected), 'Filter Rows', 'Export' (with icons for CSV and Excel), and 'Wrap Cell'. Below the buttons is a table with the following data:

	salary	title	first_name	last_name	hire_date
▶	94409	Senior Engineer	Sumanth	Peac	1985-02-18

Bibliografía

- Recurso del curso de Desarrollo de Aplicaciones Web, Documentos del Curso.