

Instituto FOC
Módulo de desarrollo de aplicaciones web

BRUNO MARENCO CERQUEIRA

Tarea Individual 5: DML: Consultas y tratamiento de datos

Marzo/2018

Índice

Índice.....2

Planteamiento3

Se pide realizar las siguientes consultas:.....3

Primera consulta3

Segunda consulta.....4

Tercera consulta4

Cuarta consulta5

Quinta consulta6

Sexta consulta7

Inserción.....8

Modificación8

Eliminación9

Planteamiento

El objetivo principal de esta tarea es la realización de consultas complejas sobre un esquema de la base de datos, es decir, se realizarán consultas en la que se mostraran consultas a varias tablas. Para la realización de esta tarea, se deberá usar el esquema “HR” con el que se viene trabajando a lo largo del curso.

Se pide realizar las siguientes consultas:

Primera consulta

- Escriba una consulta para el departamento de recursos humanos que genere las direcciones de todos los departamentos. Utilice las tablas LOCATIONS y COUNTRIES. Muestre el ID, dirección, ciudad, estado y país en la salida. Utilice NATURAL JOIN en la consulta.

SELECT location_id, street_address, city, state_province, country_name FROM locations NATURAL JOIN countries;

```

2800 Rua Frei Caneca 1360
Sao Paulo Sao Paulo
Brazil

2900 20 Rue des Corps-Saints

LOCATION_ID STREET_ADDRESS
-----
CITY STATE_PROVINCE
-----
COUNTRY_NAME
-----
Geneva Geneve
Switzerland

3000 Murtenstrasse 921 BE
Bern
Switzerland

LOCATION_ID STREET_ADDRESS
-----
CITY STATE_PROVINCE
-----
COUNTRY_NAME
-----
3100 Pieter Breughelstraat 837
Utrecht Utrecht
Netherlands

3200 Mariano Escobedo 9991
Mexico City Distrito Federal,
Mexico

LOCATION_ID STREET_ADDRESS
-----
CITY STATE_PROVINCE
-----
COUNTRY_NAME
-----

23 rows selected.

SQL>

```

Se realiza un Select para realizar la consulta, indicando primero los atributos de la consulta por orden (con el ID location en primer lugar), se determina la tabla donde se realiza la consulta con el comando FROM en la tabla locations y usamos Natural join para unir la tabla con el último dato necesario (nombre del país), la tabla countries.

Segunda consulta

- El departamento de recursos humanos necesita un informe de todos los empleados. Escriba una consulta para mostrar el apellido, el número y el nombre de departamento de todos los empleados.

SELECT last_name, department_id, department_name FROM employees JOIN departments USING (department_id);

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Smith	80	Sales
Sully	80	Sales
Taylor	80	Sales
Tucker	80	Sales
Tuvault	80	Sales
Vishney	80	Sales
Zlotkey	80	Sales
Kochhar	90	Executive
King	90	Executive
De Haan	90	Executive
Popp	100	Finance
Urman	100	Finance
Chen	100	Finance
Faviet	100	Finance
Sciarra	100	Finance
Greenberg	100	Finance
Gietz	110	Accounting
Higgins	110	Accounting

106 rows selected.

Se realiza un Select para realizar la consulta, indicando primero los atributos de la consulta por orden (con el last_name en primer lugar), se determina la tabla donde se realiza la consulta con el comando FROM en la tabla employees y usamos Join para unir la tabla con datos necesarios de otra tabla (nombre del departamento), la tabla departments. Usamos Using para especificar que la unión la hacemos con la columna department_id que está en las dos tablas, ya que hay otra columna que hay otra columna con los mismos datos y nombre en las dos tablas manager_id.

Tercera consulta

- Muestre el apellido, cargo, número y nombre de departamento de todos los empleados que trabajan en Toronto.

SELECT e.last_name, e.job_id, d.department_id, d.department_name FROM employees e JOIN departments d ON (e.department_id = d.department_id) JOIN locations l ON (d.location_id = l.location_id) WHERE (l.city) = 'toronto';

```
SQL> SELECT e.last_name, e.job_id, d.department_id, d.department_name FROM employees e JOIN de
LAST_NAME          JOB_ID    DEPARTMENT_ID
-----
DEPARTMENT_NAME
-----
Hartstein          MK_MAN      20
Marketing
Fay                MK_REP      20
Marketing
SQL> _
```

Se realiza un Select para realizar la consulta, indicando primero los atributos de la consulta por orden (con el last_name en primer lugar), se determina la tabla donde se realiza la consulta con el comando FROM en la tabla employees. Para la consulta se necesitan datos de otras dos tablas, departments y locations, por lo que se harán dos uniones con la condición ON para establecer el dato de unión entre tablas. Finalmente se coloca una condición a través de WHERE, indicando que sólo queremos los datos de los trabajadores de Toronto.

Cuarta consulta

- El departamento de recursos humanos desea determinar los nombres de todos los empleados contratados después de Davies. Cree una consulta para mostrar el nombre y la fecha de contratación de cualquier empleado después de Davies.

SELECT first_name, hire_date FROM employees WHERE hire_date > (SELECT hire_date FROM employees WHERE last_name = 'Davies');

```
FIRST_NAME          HIRE_DAT
-----
Alexis              20/02/05
Julia               24/06/06
Anthony             07/02/07
Kelly              14/06/05
Jennifer            13/08/05
Timothy             11/07/06
Randall             19/12/07
Britney             03/03/05
Samuel              01/07/06
Vance               17/03/07
Alana               24/04/06

FIRST_NAME          HIRE_DAT
-----
Kevin               23/05/06
Donald              21/06/07
Douglas             13/01/08
Pat                 17/08/05

81 rows selected.
SQL>
```

Se realiza un *Select* para realizar la consulta, indicando primero los atributos de la consulta por orden (con el *first_name* en primer lugar), se determina la tabla donde se realiza la consulta con el comando *FROM* en la tabla *employees*. Se aplica una condición a través de *WHERE*, indicando que sólo queremos los datos de los empleados contratados después de *Davies*, para saber el dato de la fecha de contratación hacemos una subconsulta con un *SELECT hire_date* de la tabla *empleados* de los empleados con apellido *Davies*.

Quinta consulta

- Cree un informe que muestre el id del empleado, apellido y salario de todos los empleados que ganan más del salario medio. Ordene los resultados en orden ascendente de salario.

SELECT employee_id, last_name, salary FROM employees WHERE salary > (SELECT AVG(salary) FROM employees);

EMPLOYEE_ID	LAST_NAME	SALARY
150	Tucker	10000
204	Baer	10000
169	Bloom	10000
149	Zlotkey	10500
162	Vishney	10500
174	Abel	11000
148	Cambrault	11000
114	Raphaely	11000
168	Ozer	11500
147	Errazuriz	12000
108	Greenberg	12000
EMPLOYEE_ID	LAST_NAME	SALARY
205	Higgins	12000
201	Hartstein	13000
146	Partners	13500
145	Russell	14000
102	De Haan	17000
101	Kochhar	17000
100	King	24000

51 rows selected.

Se realiza un *Select* para realizar la consulta, indicando primero los atributos de la consulta por orden (con el *employee_id* en primer lugar), se determina la tabla donde se realiza la consulta con el comando *FROM* en la tabla *employees*. Se aplica una condición a través de *WHERE*, indicando que sólo queremos los datos de los empleados con mayor salario que la media, para saber el dato del salario medio hacemos una subconsulta con un *SELECT AVG(salary)* de la tabla *empleados* obteniendo la media. Finalmente organizamos los datos en orden ascendente de salario con *ORDER BY salary*.

Sexta consulta

- Escriba una consulta que muestre el número de empleado y el apellido de todos los empleados que trabajan en un departamento con un empleado cuyo apellido contiene la letra “u”.

```
SELECT e.employee_id, e.last_name FROM employees e JOIN departments d
ON (d.department_id=e.department_id) WHERE d.department_id IN (SELECT
d.department_id FROM employees e JOIN departments d ON (e.department_id =
d.department_id) WHERE e.last_name LIKE '%u%');
```

```
EMPLOYEE_ID LAST_NAME
-----
151 Bernstein
150 Tucker
149 Zlotkey
148 Cambrault
147 Errazuriz
146 Partners
145 Russell
119 Colmenares
118 Himuro
117 Tobias
116 Baida

EMPLOYEE_ID LAST_NAME
-----
115 Khoo
114 Raphaely

90 rows selected.

SQL> _
```

Se realiza un Select para realizar la consulta, indicando primero los atributos de la consulta por orden (con el employee_id en primer lugar), se determina la tabla donde se realiza la consulta con el comando FROM en la tabla employees. Utilizamos ON para juntar las tablas ya que puede haber ambigüedad, así unimos department y employees. Se aplica una condición a través de WHERE, indicando que sólo queremos los datos de los departamentos que tengan algún empleado con la letra ‘u’ en el apellido, para lo que usamos IN, ya que así sólo obtiene los datos de los valores atribuidos en IN, para obtener los números de los departamentos con trabajadores con la letra ‘u’, realizamos una subconsulta. Esta subconsulta se realiza con Select el department_id, de la tabla employees unida con la tabla departments con ON (para evitar nuevamente ambigüedad), y aplicando una condición WHERE para obtener sólo los departamentos con trabajadores con la letra ‘u’ en el apellido usando LIKE ‘%u%’.

Inserción

- Inserte un empleado llamado Juan Lopez con salario de 30000 en el departamento 100 y el resto de los datos a tu elección.

```
INSERT INTO employees(employee_id, first_name, last_name, email,
phone_number, hire_date, job_id, salary, manager_id, department_id) VALUES
(207, 'Juan', 'Lopez', 'juan@gmail.com', '913022232', '23/07/90', 'MK_REP', 30000,
101, 100);
```

- Inserte un empleado llamado Alberto Perez con salario de 25000 en el departamento 100 y el resto de los datos a tu elección.

```
INSERT INTO employees(employee_id, first_name, last_name, email,
phone_number, hire_date, job_id, salary, manager_id, department_id) VALUES
(208, 'Alberto', 'Perez', 'albperez@gmail.com', '913457232', '13/11/97', 'MK_MAN',
25000, 205, 100);
```

- Consulta los datos de empleados para ver que se han introducido correctamente.

Al haber insertado los dos datos con id 207 y 208, aplicamos el siguiente comando para ver si están en la tabla.

```
SELECT * FROM employees WHERE employee_id IN (207, 208);
```

```
SQL> SELECT * FROM employees WHERE employee_id IN (207, 208);
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
207	Juan	Lopez	juan@gmail.com	913022232	23/07/90	MK_REP	30000		101	100
208	Alberto	Perez	albperez@gmail.com	913457232	13/11/97	MK_MAN	25000		205	100

Modificación

- Modifique el salario de los empleados que sea superior a 20000 disminuyendolo en 1000.

UPDATE employees SET salary = salary - 1000 WHERE salary > 20000

- Modifique el departamento para que sea 110 a los empleados cuyo apellido acabe en "ez".

UPDATE employees SET department_id = 110 WHERE last_name LIKE '%ez';

- Consulta los datos de empleados para ver que se han modificado correctamente.

Eliminación

- Elimine el empleado de nombre Alberto Perez.

DELETE FROM employees WHERE employee_id = 208;

Elimine los empleado con salario mayor que 29000 y que sean del departamento 110.

DELETE FROM employees WHERE salary > 29000;

- Consulta los datos de empleados para ver que se han eliminado correctamente.

Para saber si ha funcionado, realizamos las siguientes consultas para buscar esos datos.

```
SQL> SELECT * FROM employees WHERE last_name='Perez';  
no rows selected  
  
SQL> SELECT * FROM employees WHERE salary > 29000;  
no rows selected  
  
SQL> _
```