



PRÁCTICA 3 2ºEVALUACIÓN

Hecho por: Jesús Padilla Crespo

Practica 3 2ªEvaluación

Ejercicios de Funciones de mysql

Parte1

Vamos a usar la BBDD del ejercicio1 de la evaluación (EMP y DEPT):

1. Muestra la versión de Mysql

```
mysql> SELECT VERSION();
+-----+
| VERSION() |
+-----+
| 5.7.17-log |
+-----+
1 row in set (0.00 sec)
```

2. Muestra la fecha y hora actual

```
mysql> SELECT NOW();
+-----+
| NOW() |
+-----+
| 2022-02-04 09:02:08 |
+-----+
1 row in set (0.02 sec)
```

3. Compara las cadenas ("Salesianos", y "SALESIANOS")

```
mysql> SELECT STRCMP ("Salesianos","SALESIANOS");
+-----+
| STRCMP ("Salesianos","SALESIANOS") |
+-----+
| 0 |
+-----+
1 row in set (0.00 sec)
```

4. Muestra la longitud de la cadena de los "ename" de la tabla EMP.

```
mysql> SELECT ENAME,CHAR_LENGTH(ENAME) AS "LONGITUD NOMBRES" FROM EMP;
+-----+-----+
| ENAME | LONGITUD NOMBRES |
+-----+-----+
| SMITH | 5 |
| ALLEN | 5 |
| WARD  | 4 |
| JONES | 5 |
| MARTIN | 6 |
| BLAKE | 5 |
| CLARK | 5 |
| SCOTT | 5 |
| KING  | 4 |
| TURNER | 6 |
| ADAMS | 5 |
| JAMES | 5 |
| FORD  | 4 |
| MILLER | 6 |
+-----+-----+
14 rows in set (0.00 sec)
```

5. Concatena los "Ename" y los "job" de todos los empleados, separándolos con un guión ("-").

```
mysql> SELECT CONCAT (ENAME,"-",JOB) FROM EMP;
+-----+
| CONCAT (ENAME,"-",JOB) |
+-----+
| SMITH-CLERK            |
| ALLEN-SALESMAN         |
| WARD-SALESMAN          |
| JONES-MANAGER          |
| MARTIN-SALESMAN        |
| BLAKE-MANAGER          |
| CLARK-MANAGER          |
| SCOTT-ANALYST          |
| KING-PRESIDENT         |
| TURNER-SALESMAN        |
| ADAMS-CLERK            |
| JAMES-CLERK            |
| FORD-ANALYST           |
| MILLER-CLERK           |
+-----+
14 rows in set (0.00 sec)
```

6. Muestra los "loc" alreves de la tabla DEPT.

```
mysql> SELECT REVERSE(LOC) FROM DEPT;
+-----+
| REVERSE(LOC) |
+-----+
| KROY_WEN     |
| SALLAD       |
| OGACIHC      |
| NOTSOB       |
+-----+
4 rows in set (0.03 sec)
```

7. Rellena con puntos "." por la derecha hasta 20 caracteres los ename de la tabla emp.

```
mysql> SELECT RPAD (ENAME,20,".")FROM EMP;
+-----+
| RPAD (ENAME,20,".") |
+-----+
| SMITH.....        |
| ALLEN.....          |
| WARD.....            |
| JONES.....           |
| MARTIN.....          |
| BLAKE.....            |
| CLARK.....            |
| SCOTT.....            |
| KING.....             |
| TURNER.....           |
| ADAMS.....            |
| JAMES.....            |
| FORD.....             |
| MILLER.....           |
+-----+
14 rows in set (0.00 sec)
```

8. Muestra en mayúsculas todos Dname de la tabla DEPT.

```
mysql> SELECT UPPER(DNAME)FROM DEPT;
+-----+
| UPPER(DNAME) |
+-----+
| ACCOUNTING   |
| RESEARCH     |
| SALES        |
| OPERATIONS   |
+-----+
4 rows in set (0.00 sec)
```

9. Muestra el nombre y el año (solo el año de la fecha) de todos los empleados.

```
mysql> SELECT ENAME, YEAR(HIREDATE) FROM EMP;
+-----+-----+
| ENAME | YEAR(HIREDATE) |
+-----+-----+
| SMITH | 1980            |
| ALLEN | 1981            |
| WARD  | 1981            |
| JONES | 1981            |
| MARTIN | 1981            |
| BLAKE | 1981            |
| CLARK | 1981            |
| SCOTT | 1982            |
| KING  | 1981            |
| TURNER | 1981           |
| ADAMS | 1983            |
| JAMES | 1981            |
| FORD  | 1981            |
| MILLER | 1982           |
+-----+-----+
14 rows in set (0.00 sec)
```

10. Muestra la hora actual.

```
mysql> SELECT CURTIME();
+-----+
| CURTIME() |
+-----+
| 09:16:50  |
+-----+
1 row in set (0.00 sec)
```

11. Muestra el día de la semana de hoy.

```
mysql> SELECT DAYNAME(NOW());
+-----+
| DAYNAME(NOW()) |
+-----+
| Friday         |
+-----+
1 row in set (0.00 sec)
```

12. Muestra la fecha de hoy en formato: 7 de Febrero de 2021

```
mysql> SELECT DATE_FORMAT(NOW(), "%D OF %M OF %Y");
+-----+
| DATE_FORMAT(NOW(), "%D OF %M OF %Y") |
+-----+
| 4th OF February OF 2022              |
+-----+
1 row in set (0.00 sec)
```

13. Muestra la fecha de la tabla emp en formato: 7 de Febrero de 2021

```
mysql> SELECT DATE_FORMAT(HIREDATE, "%D OF %M OF %Y") FROM EMP;
+-----+
| DATE_FORMAT(HIREDATE, "%D OF %M OF %Y") |
+-----+
| 17th OF December OF 1980                |
| 20th OF February OF 1981                |
| 22nd OF February OF 1981                |
| 2nd OF April OF 1981                    |
| 28th OF September OF 1981               |
| 1st OF May OF 1981                      |
| 9th OF June OF 1981                     |
| 9th OF December OF 1982                 |
| 17th OF November OF 1981                |
| 8th OF September OF 1981                |
| 12th OF January OF 1983                 |
| 3rd OF December OF 1981                 |
| 3rd OF December OF 1981                 |
| 23rd OF January OF 1982                 |
+-----+
14 rows in set (0.00 sec)
```

14. Muestra el salario más alto de la tabla emp.

```
mysql> SELECT MAX(SAL) FROM EMP;
+-----+
| MAX(SAL) |
+-----+
| 5000.00 |
+-----+
1 row in set (0.00 sec)
```

15. Muestra el primer empleado contratado en la empresa.

```
mysql> SELECT ENAME FROM EMP WHERE HIREDATE = (SELECT MIN(HIREDATE) FROM EMP);
+-----+
| ENAME |
+-----+
| SMITH |
+-----+
1 row in set (0.02 sec)

mysql> SELECT ENAME FROM EMP ORDER BY HIREDATE LIMIT 1;
+-----+
| ENAME |
+-----+
| SMITH |
+-----+
1 row in set (0.00 sec)
```

16. Calcula el salario medio de la empresa.

```
mysql> SELECT AVG(SAL) FROM EMP;
+-----+
| AVG(SAL) |
+-----+
| 2073.214286 |
+-----+
1 row in set (0.00 sec)
```

17. Halla la media de los salarios de los departamentos cuyo salario mínimo supera a 1000.

```
mysql> SELECT AVG(SAL) FROM EMP GROUP BY DEPTNO HAVING MIN(SAL)>1000;
+-----+
| AVG(SAL) |
+-----+
| 2916.666667 |
+-----+
1 row in set (0.00 sec)
```

18. Muestra cuantos empleados no tienen comisión.

```
mysql> SELECT ENAME,COMM FROM EMP WHERE COMM IS NULL OR COMM= 0;
+-----+-----+
| ENAME | COMM |
+-----+-----+
| SMITH | NULL |
| JONES | NULL |
| BLAKE | NULL |
| CLARK | NULL |
| SCOTT | NULL |
| KING | NULL |
| TURNER | 0.00 |
| ADAMS | NULL |
| JAMES | NULL |
| FORD | NULL |
| MILLER | NULL |
+-----+-----+
11 rows in set (0.00 sec)
```

19. Muestra la media de salario de los departamentos agrupado por departamento.

```
mysql> SELECT AVG(SAL) FROM EMP GROUP BY DEPTNO;
+-----+
| AVG(SAL) |
+-----+
| 2916.666667 |
| 2175.000000 |
| 1566.666667 |
+-----+
3 rows in set (0.00 sec)
```

20. Muestra la suma de salario de los departamentos agrupado por departamento cuya suma sea mayor de 2000.

```
mysql> SELECT SUM(SAL) FROM EMP GROUP BY DEPTNO HAVING SUM(SAL)>2000;
+-----+
| SUM(SAL) |
+-----+
| 8750.00 |
| 10875.00 |
| 9400.00 |
+-----+
3 rows in set (0.00 sec)
```

21. Prueba las funciones con números de prueba: SIN(), COT(), ACOS(), ASIN(), EXP(), LN(), LOG(), POWER(), SQRT()

```
mysql> SELECT SIN(10),COT(11),ACOS(0.25),ASIN(0.25);
```

SIN(10)	COT(11)	ACOS(0.25)	ASIN(0.25)
-0.5440211108893698	-0.0044257413313241135	1.318116071652818	0.25268025514207865

```
1 row in set (0.00 sec)
```

```
mysql> SELECT EXP(14),LN(15),LOG(16),POWER(3,2),SQRT(17);
```

EXP(14)	LN(15)	LOG(16)	POWER(3,2)	SQRT(17)
1202604.2841647768	2.70805020110221	2.772588722239781	9	4.123105625617661

```
1 row in set (0.00 sec)
```

Parte2

Crea una base de datos "ejercicio3", en la que guardaremos información sobre selecciones nacionales de baloncesto.

Para ello tendremos: una tabla "PAIS" y una tabla "JUGADOR".

De cada país guardaremos el nombre (por ejemplo, "España") y un código que actuará como clave primaria (por ejemplo, "ESP"). De cada jugador anotaremos código, nombre, apellidos, posición y código de la selección a la que pertenece.

1. Añade los países:
 - ✦ ESP, España
 - ✦ ARG, Argentina
 - ✦ AUS, Australia
 - ✦ LIT, Lituania
2. Añade los jugadores:
 - ✦ RUB, Ricky, Rubio, Base,ESP
 - ✦ NAV, Sergio, Lull, Alero,ESP
 - ✦ SCO, Luis, Scola, Ala-Pivot,ARG
 - ✦ DEL, Carlos, Delfino, Escolta,ARG
 - ✦ MAC, Jonas, Maciulis, Alero,LIT
 - ✦ BOG, Andrew, Bogut, Pivot,AUS

Puedes añadir más datos de prueba.

Crea con funciones lo siguiente:

```
mysql> USE EJERCICIO3;
Database changed
mysql> CREATE TABLE PAIS (
  ->     CODIGO_PAIS VARCHAR(3) PRIMARY KEY,
  ->     NOMBRE VARCHAR(15) NOT NULL
  -> );
Query OK, 0 rows affected (0.05 sec)

mysql> CREATE TABLE JUGADOR (
  ->     CODIGO VARCHAR(3) PRIMARY KEY,
  ->     NOMBRE VARCHAR(15) NOT NULL,
  ->     APELLIDOS VARCHAR(15) NOT NULL,
  ->     POSICION VARCHAR(15) NOT NULL,
  ->     CODIGO_PAIS VARCHAR(3) NOT NULL REFERENCES PAIS(CODIGO_PAIS)
  -> );
Query OK, 0 rows affected (0.03 sec)
```



```
mysql> INSERT INTO PAIS VALUES ("ESP", "ESPAÑA");
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PAIS VALUES ("ARG", "ARGENTINA");
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO PAIS VALUES ("AUS", "AUSTRALIA");
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PAIS VALUES ("LIT", "LITUANIA");
Query OK, 1 row affected (0.00 sec)

mysql>
mysql> INSERT INTO JUGADOR VALUES ("RUB", "RICKY", "RUBIO", "BASE","ESP");
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO JUGADOR VALUES ("NAV", "SERGIO", "LULL", "ALERO","ESP");
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO JUGADOR VALUES ("SCO", "LUIS", "SCOLA", "ALA-PIVOT","ARG");
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO JUGADOR VALUES ("DEL", "CARLOS", "DELFINO", "ESCOLTA","ARG");
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO JUGADOR VALUES ("MAC", "JONAS", "MACIULIS", "ALERO","LIT");
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO JUGADOR VALUES ("BOG", "ANDREW", "BOGUT", "PIVOT","AUS");
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO JUGADOR VALUES ("LBN", "LA BOMBA", "NAVARRO", "RETIRADO","ESP");
Query OK, 1 row affected (0.01 sec)
```

1. Muestra los nombres y apellidos de todos los jugadores, en mayúsculas, ordenados por apellido y nombre.

```
+-----+-----+
| UPPER (NOMBRE) | UPPER(APELLIDOS) |
+-----+-----+
| ANDREW          | BOGUT             |
| CARLOS          | DELFINO           |
| SERGIO          | LULL              |
| JONAS           | MACIULIS          |
| LA BOMBA        | NAVARRO           |
| RICKY           | RUBIO             |
| LUIS            | SCOLA             |
+-----+-----+
7 rows in set (0.00 sec)
```

2. Muestra el nombre y apellidos del jugador o jugadores cuyo apellido es el más largo (formado por más letras).

```
mysql> SELECT NOMBRE,APELLIDOS FROM JUGADOR WHERE CHAR_LENGTH(APELLIDOS) = (SELECT MAX(CHAR_LENGTH(APELLIDOS))FROM JUGADOR);
+-----+-----+
| NOMBRE | APELLIDOS |
+-----+-----+
| JONAS  | MACIULIS  |
+-----+-----+
1 row in set (0.00 sec)
```


7. Muestra el nombre y apellidos de todos los jugadores cuyo país contenga una N en el nombre. Debes eliminar los espacios iniciales y finales de ambos campos, en caso de que existan.

```
mysql> SELECT TRIM(JUGADOR.NOMBRE),TRIM(APELLIDOS),TRIM(PAIS.NOMBRE) FROM JUGADOR,PAIS WHERE PAIS.CODIGO_PAIS=JUGADOR.CODIGO_PAIS AND PAIS.NOMBRE LIKE "%N%";
```

TRIM(JUGADOR.NOMBRE)	TRIM(APELLIDOS)	TRIM(PAIS.NOMBRE)
CARLOS	DELFINO	ARGENTINA
LA BOMBA	NAVARRO	ESPAÑA
JONAS	MACIULIS	LITUANIA
SERGIO	LULL	ESPAÑA
RICKY	RUBIO	ESPAÑA
LUIS	SCOLA	ARGENTINA

```
6 rows in set (0.00 sec)
```

8. Muestra al revés el apellido de los jugadores de Australia.

```
mysql> SELECT REVERSE(APELLIDOS) FROM JUGADOR WHERE CODIGO_PAIS="AUS";
```

REVERSE(APELLIDOS)
TUGOB

```
1 row in set (0.00 sec)
```

9. Muestra una cadena formada por 10 guiones,nombre y otros 10 guiones.

```
mysql> SELECT CONCAT("-----",NOMBRE,"-----") FROM JUGADOR;
```

CONCAT("-----",NOMBRE,"-----")
-----ANDREW-----
-----CARLOS-----
-----LA BOMBA-----
-----JONAS-----
-----SERGIO-----
-----RICKY-----
-----LUIS-----

```
7 rows in set (0.00 sec)
```