

Creamos la base de datos, los usuarios y las tablas:

```
CREATE DATABASE PUERTO;
USE PUERTO;

/*Crear los usuarios*/
CREATE USER 'makina'@'localhost' IDENTIFIED BY 'makina';
CREATE USER 'pica'@'localhost' IDENTIFIED BY 'pica';

/*Crear todas las tablas*/
CREATE TABLE BARCO (
    MAT INT(10) PRIMARY KEY,
    NOMBRE VARCHAR(25),
    FECHABOT DATE,
    PESO INT(6))
    ENGINE=INNODB;

CREATE TABLE MARINERO (
    DNI VARCHAR(9) PRIMARY KEY,
    NOMBRE VARCHAR(25),
    APELLIDOS VARCHAR(35),
    SEXO VARCHAR(1),
    EDAD INT(3),
    DIRECCION VARCHAR(40),
    CP INT(5))
    ENGINE=INNODB;

CREATE TABLE TRIPULACION (
    MAT INT(10),
    DNI VARCHAR(9),
    FECHA DATE,
    SUELDO INT(5),
    ESTADO VARCHAR(1),
    CONSTRAINT FK_MAT FOREIGN KEY (MAT) REFERENCES BARCO(MAT),
    CONSTRAINT FK_DNI FOREIGN KEY (DNI) REFERENCES MARINERO(DNI),
    CONSTRAINT PRIMARY KEY(MAT,DNI,FECHA))
    ENGINE=INNODB;
```

Y asignamos los permisos para los usuarios:

```
/*Asignar los permisos correspondiente a cada usuario*/
GRANT select, insert ON PUERTO.BARCO TO 'pica'@'localhost';
GRANT select, insert ON PUERTO.marinero TO 'pica'@'localhost';
GRANT EXECUTE ON PUERTO.* TO 'pica'@'localhost';
#Le he dado permisos de ejecución para hacer las inserciones desde un procedimiento
GRANT ALL ON PUERTO.* TO 'makina'@'localhost';

flush privileges; #Recargando los privilegios.
```

Comprobación de permisos:

```
:Alberto@DESKTOP-DLKALCK c:\xampp
# mysql -u pica -p
Enter password: ****
mysql> use puerto;
```

Sin permisos en la tabla tripulación:

```
mysql> select * from tripulación;
ERROR 1142 (42000): SELECT command denied to user 'pica'@'localhost' for table
'tripulación'
```

Con permisos para usar SELECT en la tabla barco.

```
mysql> select * from barco;
+----+-----+-----+-----+
| MAT | NOMBRE | FECHABOT | PESO |
+----+-----+-----+-----+
| 1   | Jeny 1 | 2015-02-13 | 65000 |
| 2   | Jeny 2 | 2000-03-05 | 15000 |
+----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Creamos el procedimiento de inserción para todas las tablas con el control de errores de inserción para registro duplicado:

```
/*Barco*/
DROP PROCEDURE IF EXISTS insbarco;
DELIMITER $$ 
CREATE PROCEDURE insbarco(mat int(10), nombre varchar(25), fechabot date, peso int(6))
BEGIN
    DECLARE CONTINUE HANDLER FOR 1062 SELECT 'Clave principal duplicada' AS 'ERROR';
    INSERT INTO barco VALUES (mat, nombre, fechabot, peso);
END $$ 
DELIMITER ;

/*Marinero*/
DROP PROCEDURE IF EXISTS insmarinero;
DELIMITER $$ 
CREATE PROCEDURE insmarinero(DNI VARCHAR(9), NOMBRE VARCHAR(25),
                           APELLIDOS VARCHAR(35), SEXO VARCHAR(1), EDAD INT(3),
                           DIRECCION VARCHAR(40), CP INT(5))
BEGIN
    DECLARE CONTINUE HANDLER FOR 1062 SELECT 'Clave principal duplicada' AS 'ERROR';
    INSERT INTO marinero VALUES (dni, nombre, apellidos, sexo, edad, direccion, cp);
END $$ 
DELIMITER ;

/*Tripulación*/
DROP PROCEDURE IF EXISTS instripu;
DELIMITER $$ 
CREATE PROCEDURE instripu(MAT INT(10), DNI VARCHAR(9), FECHA DATE, SUELDO INT(5),
                        ESTADO VARCHAR(1))
BEGIN
    DECLARE CONTINUE HANDLER FOR 1062 SELECT 'Clave principal duplicada' AS 'ERROR';
    INSERT INTO tripulacion VALUES (mat, dni, fecha, sueldo, estado);
END $$ 
DELIMITER ;
```

Error si intentamos realizar una inserción con clave primaria duplicada en la tabla barco:

```
mysql> call insbarco(2, 'Jeny 2', '2000-03-05', 15000);
+-----+
| ERROR |
+-----+
| Clave principal duplicada |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

Tabla marineros:

```
mysql> select * from marinero;
+-----+-----+-----+-----+-----+-----+-----+
| DNI      | NOMBRE   | APELLIDOS | SEXO | EDAD | DIRECCION | CP    |
+-----+-----+-----+-----+-----+-----+-----+
| 13457896P | Luis     | Perez     | H    | 54   | C/Cazuelas | 29620 |
| 25736433X | Ana      | Mendiaz   | M    | 31   | C/Piruleta | 29620 |
| 96325418L | Alberto  | Martin    | H    | 26   | C/Tecleo   | 25874 |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> call insmarinero('25736433X','Ana','Mendiaz','M',31,'C/Piruleta',29620)
;
+-----+
| ERROR          |
+-----+
| Clave principal duplicada |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

Tabla tripulación:

```
mysql> select * from tripulacion;
ERROR 1142 (42000): SELECT command denied to user 'pica'@'localhost' for table
'tripulacion'
mysql> INSERT INTO TRIPULACION VALUES (24,'13457896P','2016-01-21',2500,'A');
ERROR 1142 (42000): INSERT command denied to user 'pica'@'localhost' for table
'tripulacion'
```

Hay un problema, que no sé cómo resolver, si la inserción se realiza desde un procedimiento se ejecuta la inserción sin problemas. Si se usa la sintaxis INSERT INTO directamente si que comprueba los permisos. La única forma que veo de que pueda controlar esto es añadiendo un TRIGGER a cada tabla para la comprobación del error.

Cambio al usuario makina que es el que tiene privilegios para todo en toda la base de datos:

```
# mysql -u makina -p
Enter password: *****
mysql> call instripu(24,'13457896P','2016-01-21',2500,'A');
+-----+
| ERROR          |
+-----+
| Clave principal duplicada |
+-----+
1 row in set (0.00 sec)

mysql> SELECT * FROM TRIPULACION;
+-----+-----+-----+-----+
| MAT | DNI      | FECHA     | SUELDO | ESTADO |
+-----+-----+-----+-----+
| 1  | 13457896P | 2016-01-21 | 2500   | A       |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Jose Alberto Martín Torres - Puerto

Tablas antes de las modificaciones:

SELECT * FROM puerto.barco;						
1	Jeny 1	2015-02-13	65000			
2	Jeny 2	2000-03-05	15000			
24	Titanic	2008-08-15	985000			
SELECT * FROM puerto.marinero;						
13457896P	Luis	Perez	H	54	C/Cazuelas	29620
25736433X	Ana	Mendiaz	M	31	C/Piruleta	29620
45126398O	Jose	Torres	M	62	C/Raton	29620
96325418L	Alberto	Martín	H	26	C/Tedeo	25874
SELECT * FROM puerto.tripulacion;						
1	13457896P	2016-01-21	2500	A		
2	13457896P	2015-07-13	2500	B		
2	25736433X	2015-07-13	6500	B		
24	25736433X	2016-01-21	8100	A		
24	96325418L	2015-01-21	2600	B		

Consultas y modificaciones:

/*Sacar el nombre y apellidos ordenados de forma descendente por apellido de todos aquellos marineros cuyo acumulado de sueldo sea mayor de 10000*/

```
SELECT MARINERO.nombre, MARINERO.apellidos, SUM(SUELDO) AS 'Sueldo total'  
FROM MARINERO  
INNER JOIN TRIPULACION ON MARINERO.dni=TRIPULACION.dni  
GROUP BY TRIPULACION.dni  
HAVING SUM(SUELDO)>10000  
ORDER BY MARINERO.apellidos DESC;
```

nombre	apellidos	Sueldo total
Ana	Mendiaz	14600

Creación de la función para aumentar sueldo y el procedimiento:

```

/*Crear la funcion aumenta el sueldo del marinero*/
DROP FUNCTION IF EXISTS premio;
DELIMITER $$ 
CREATE FUNCTION premio(s int)
RETURNS INT
BEGIN
    DECLARE total INT DEFAULT 0;
    SET total = s * 1.20;
    RETURN total;
END $$ 
DELIMITER ;

```



```

/*Procedimiento para aumentar el sueldo*/
DROP PROCEDURE IF EXISTS aumentar;
DELIMITER $$ 
CREATE PROCEDURE aumentar()
BEGIN
    DECLARE sueldo_i INT DEFAULT 0;
    DECLARE idmari VARCHAR(9);
    DECLARE matri INT DEFAULT 0;
    DECLARE salida INT DEFAULT 0;
    DECLARE cursormari CURSOR FOR
        SELECT TRIPULACION.dni, TRIPULACION.sueldo, TRIPULACION.mat
        FROM MARINERO
        INNER JOIN TRIPULACION ON TRIPULACION.dni=MARINERO.dni
        INNER JOIN BARCO ON TRIPULACION.mat=BARCO.mat
        WHERE MARINERO.sexo = 'H'
        AND MARINERO.cp = ( SELECT cp
                            FROM MARINERO
                            WHERE nombre = 'Ana'
                            AND apellidos = 'Mendiaz')
        OR TRIPULACION.mat = 24;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET salida = 1;
    OPEN cursormari;
    bucle: while salida = 0 do
        FETCH cursormari INTO idmari,sueldo_i,matri;
        SET sueldo_i = premio(sueldo_i);
        UPDATE tripulacion
        SET sueldo = sueldo_i
        WHERE dni = idmari
        AND mat = matri;
    END WHILE bucle;
    CLOSE cursormari;
END $$ 
DELIMITER ;

```

Antes y después de ejecutar:

SELECT * FROM puerto.tripulacion;						MAT	DNI	FECHA	SUELDO	ESTADO
1	13457896P	2016-01-21	2500	A	▶	1	13457896P	2016-01-21	3000	A
2	13457896P	2015-07-13	2500	B		2	13457896P	2015-07-13	3000	B
2	25736433X	2015-07-13	6500	B		2	25736433X	2015-07-13	6500	B
24	25736433X	2016-01-21	8100	A		24	25736433X	2016-01-21	9720	A
24	96325418L	2015-01-21	2600	B		24	96325418L	2015-01-21	3744	B

Jose Alberto Martín Torres - Puerto

```
/*Dar de baja a los tripulantes de los barcos mayores de 30 toneladas
y que cobran entre 2000 y 4000*/
DROP PROCEDURE IF EXISTS dardebaja;
DELIMITER $$
CREATE PROCEDURE dardebaja()
BEGIN
    DECLARE matri INT;
    DECLARE estado_i VARCHAR(1);
    DECLARE dni_i VARCHAR(9);
    DECLARE salida INT DEFAULT 0;
    DECLARE cursorbaja CURSOR FOR
        SELECT TRIPULACION.mat, TRIPULACION.dni, TRIPULACION.estado
        FROM TRIPULACION
        INNER JOIN BARCO ON TRIPULACION.mat=BARCO.mat
        WHERE BARCO.peso > 30000
        AND TRIPULACION.sueldo BETWEEN 2000 AND 4000;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET salida = 1;
    OPEN cursorbaja;
    bucle: while salida = 0 do
        FETCH cursorbaja INTO matri, dni_i, estado_i;
        SET estado_i = 'B';
        UPDATE TRIPULACION
        SET estado = estado_i
        WHERE TRIPULACION.mat = matri
        AND TRIPULACION.dni = dni_i;
    END WHILE bucle;
    CLOSE cursorbaja;
END $$
DELIMITER ;
```

Antes y después de ejecutar:

SELECT * FROM puerto.tripulacion;					MAT	DNI	FECHA	SUELDO	ESTADO
1	13457896P	2016-01-21	2500	A	1	13457896P	2016-01-21	3000	B
2	13457896P	2015-07-13	2500	B	2	13457896P	2015-07-13	3000	B
2	25736433X	2015-07-13	6500	B	2	25736433X	2015-07-13	6500	B
24	25736433X	2016-01-21	8100	A	24	25736433X	2016-01-21	9720	A
24	96325418L	2015-01-21	2600	B	24	96325418L	2015-01-21	3744	B