***UF2. [PAC03] Lenguaje DML.***

Actividades.

Usando la base de datos Liga que se encuentra subida a la plataforma, realiza las siguientes consultas en MySQL:

1. **Una correctamente cada instrucción con su explicación.**

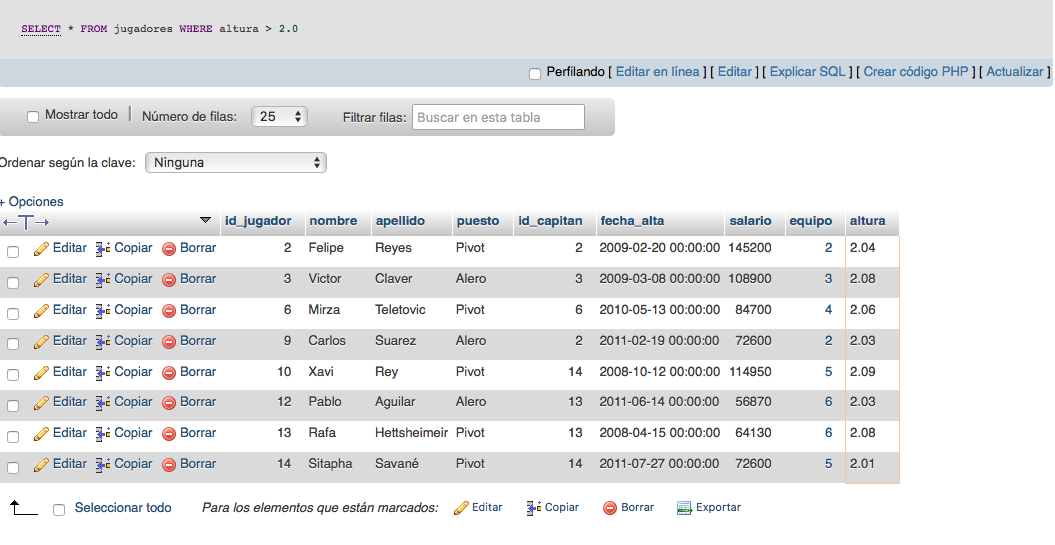
|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| B | C | D | A |

1. **Principales diferencias entre el comando TRUNCATE y el DELETE: marcar con x la columna del comando al que corresponda cada afirmación:**

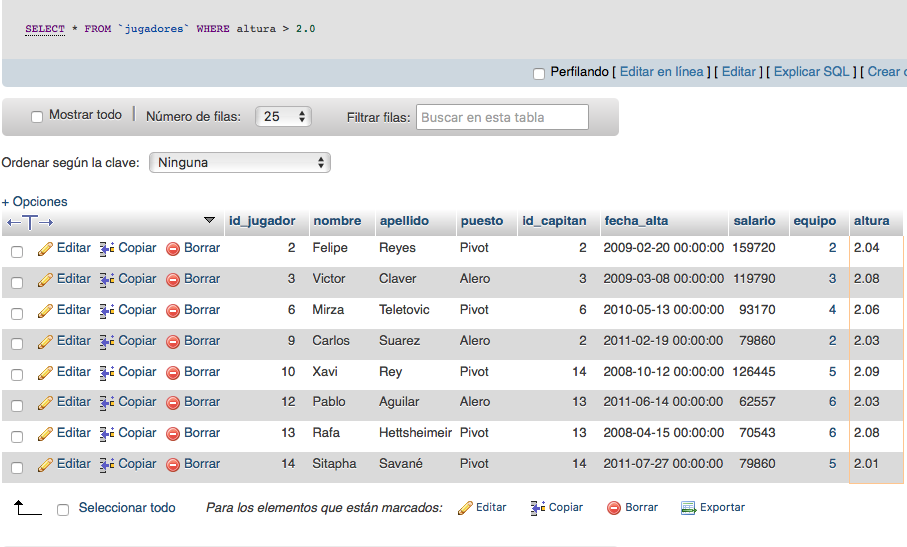
|  |  |  |
| --- | --- | --- |
|  | TRUNCATE | DELETE |
| Borra toda la tabla y la recrea | X |  |
| Elimina filas una a una |  | X |
| Es más rápido | X |  |
| No se puede deshacer |  | X |

1. **Indicar el comando necesario para aumentar el salario de los jugadores de más de dos metros un 10%.**

SELECT \* FROM jugadores WHERE altura > 2.0

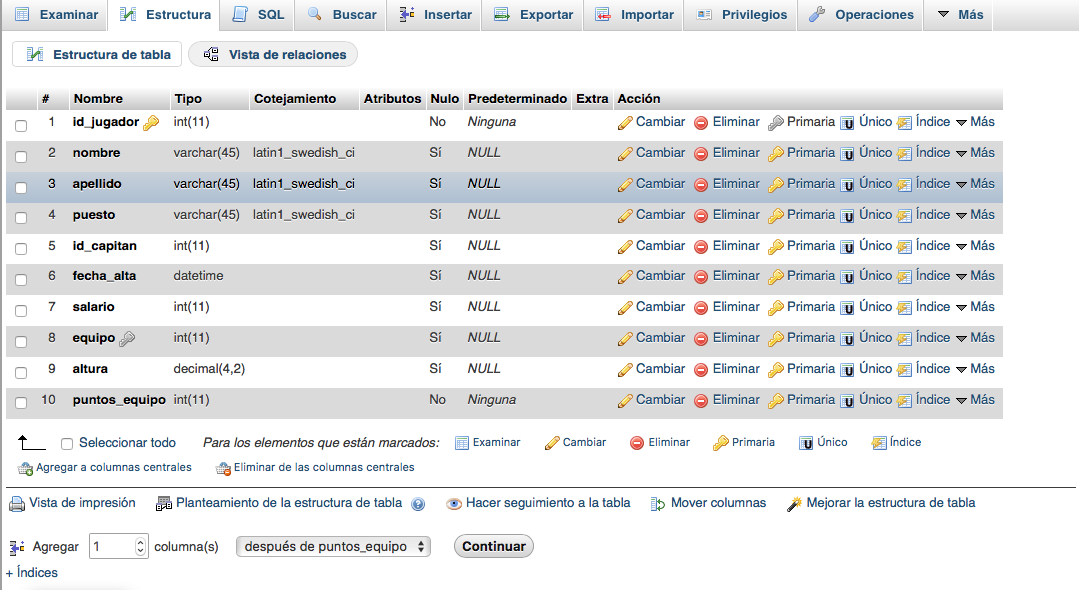


UPDATE jugadores SET salario = (salario\*1.1) WHERE altura > 2.0



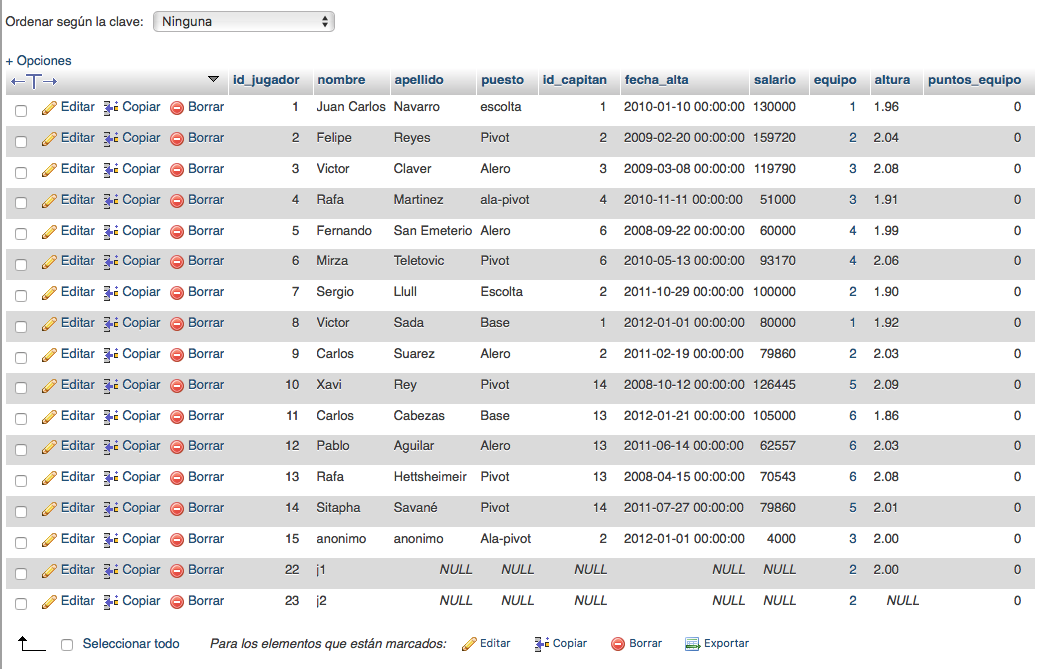
1. **Con el comando ALTER TABLE podemos añadir a la tabla jugadores la columna puntos\_equipo para reflejar todos los puntos de su equipo. Una vez creado el campo, ¿Cuál sería el comando para actualizar sus valores con los valores correctos en la tabla equipos?**

ALTER TABLE `jugadores` ADD `puntos\_equipo` INT(11) NOT NULL AFTER `altura`;



SELECT E.puntos FROM equipos E, jugadores J INTO J.puntos\_equipo WHERE E.id\_equipo = J.equipo

UPDATE jugadores SET jugadores.puntos\_equipo = equipos.puntos FROM jugadores INNER JOIN equipos ON jugadores.equipo = equipos.id\_equipo



1. **Escribir en el orden correcto cuáles serían los comandos necesarios para eliminar todos los datos de las tablas en la base liga. Justificar la respuesta.**

DELETE FROM partidos WHERE 1;

DELETE FROM jugadores WHERE 1;

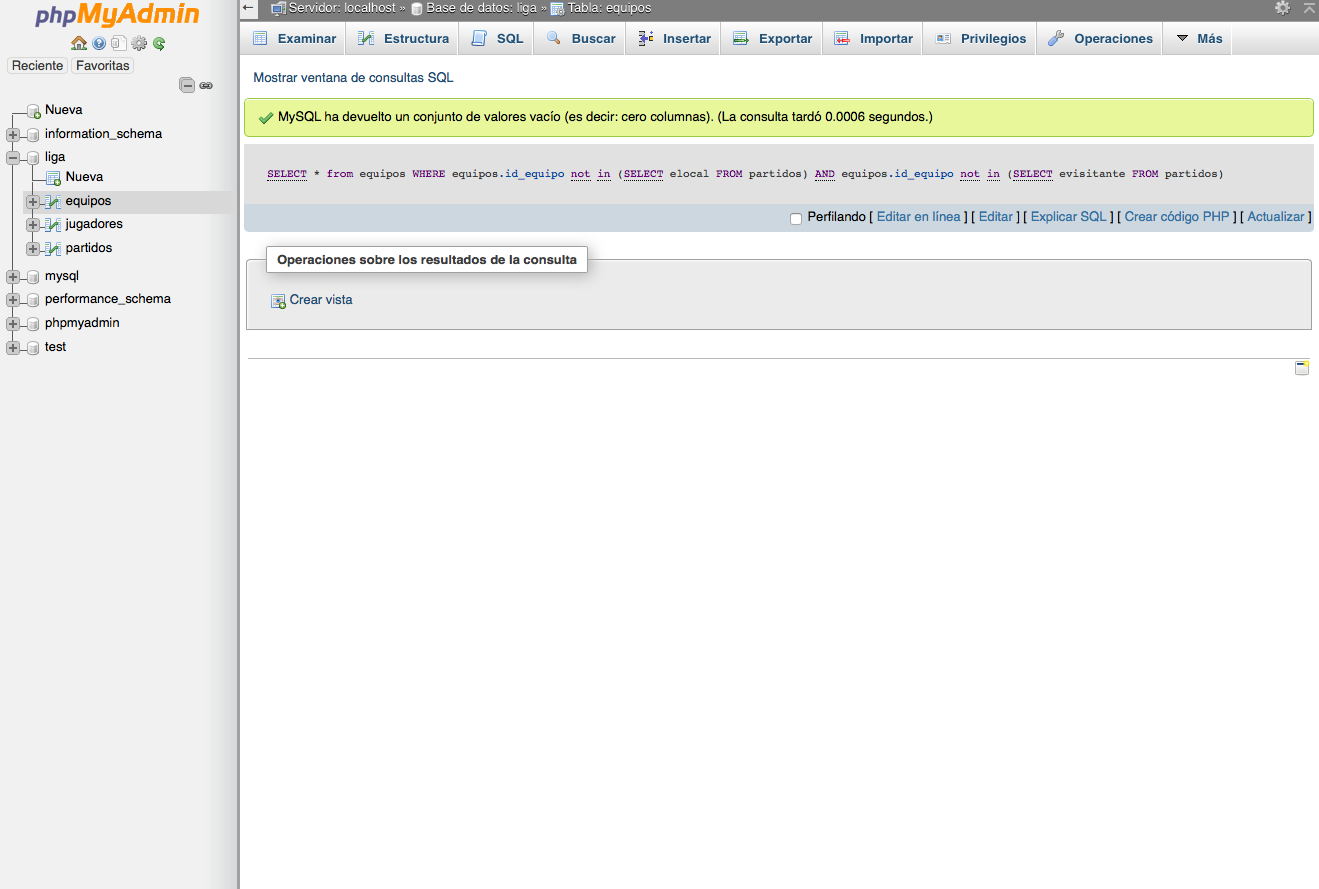
DELETE FROM equipos WHERE 1;

Eliminamos en orden partidos, jugadores y equipos debido a que las tablas partidos y jugadores tienen restricciones de tipo foreign key que referencian a equipos.

1. **Escribir una sentencia que borre registros de equipos que no hayan jugado partidos.**

SELECT \* from equipos WHERE equipos.id\_equipo not in (SELECT elocal FROM partidos) AND equipos.id\_equipo not in (SELECT evisitante FROM partidos)

DELETE \* from equipos WHERE equipos.id\_equipo not in (SELECT elocal FROM partidos) AND equipos.id\_equipo not in (SELECT evisitante FROM partidos)

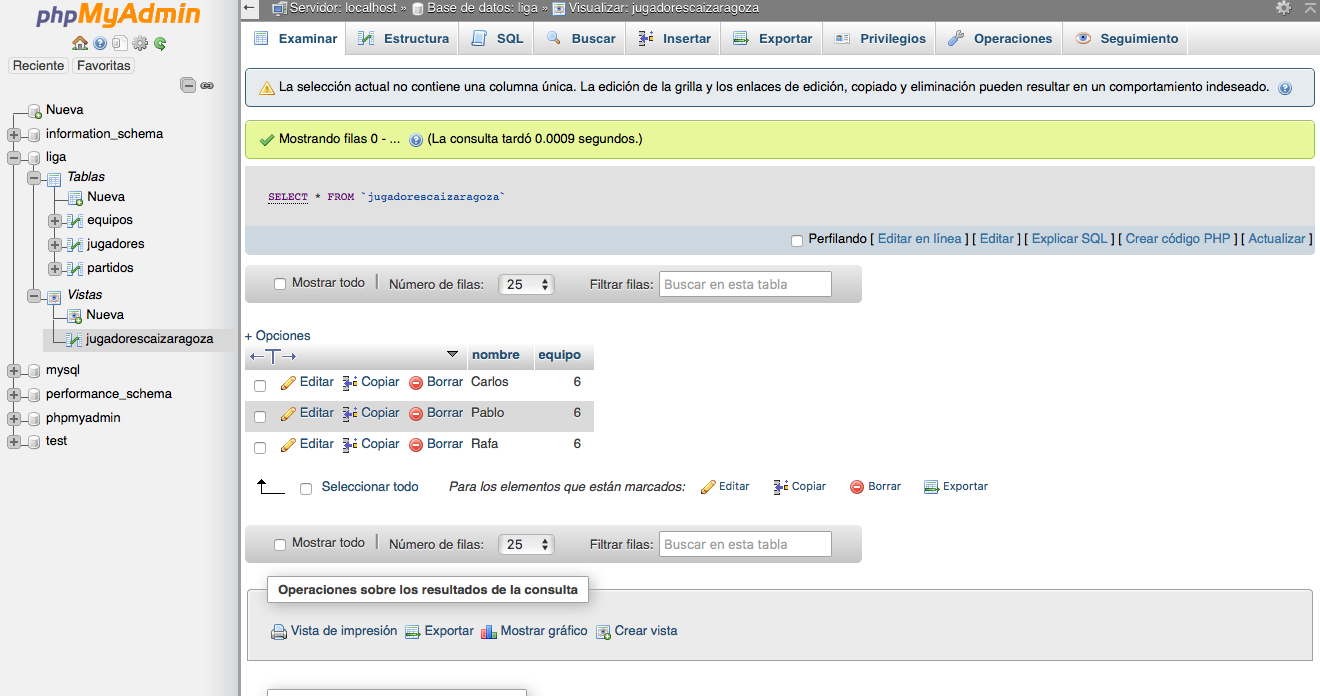


1. **Indicar si es verdadero (V) o falso (F) cada uno de los requisitos de los que se especifican para que una vista creada sea actualizable.**

|  |  |
| --- | --- |
| Correspondencia uno a uno entre las filas de la vista y las tablas subyacentes | V |
| Combinación de múltiples tablas | F |
| Existencia de cláusulas GROUP BY, HAVING, DITINCT, UNION o subconsultas en el predicado de SELECT | F |

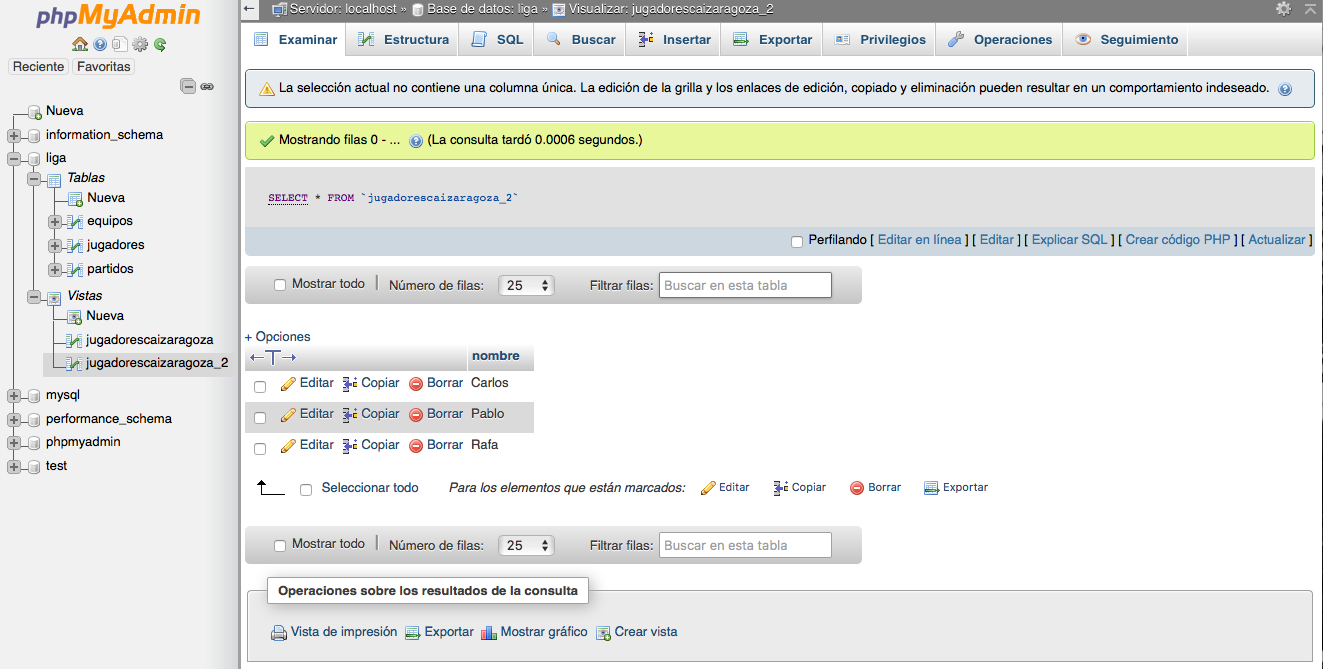
1. **A Escribir las sentencias necesarias para:**
   * Crear una vista con los campos nombre y equipo de los jugadores del CAI Zaragoza.

CREATE VIEW jugadoresCaiZaragoza AS (SELECT nombre, equipo FROM jugadores WHERE equipo = (SELECT id\_equipo FROM equipos WHERE nombre = "CAI Zaragoza"))

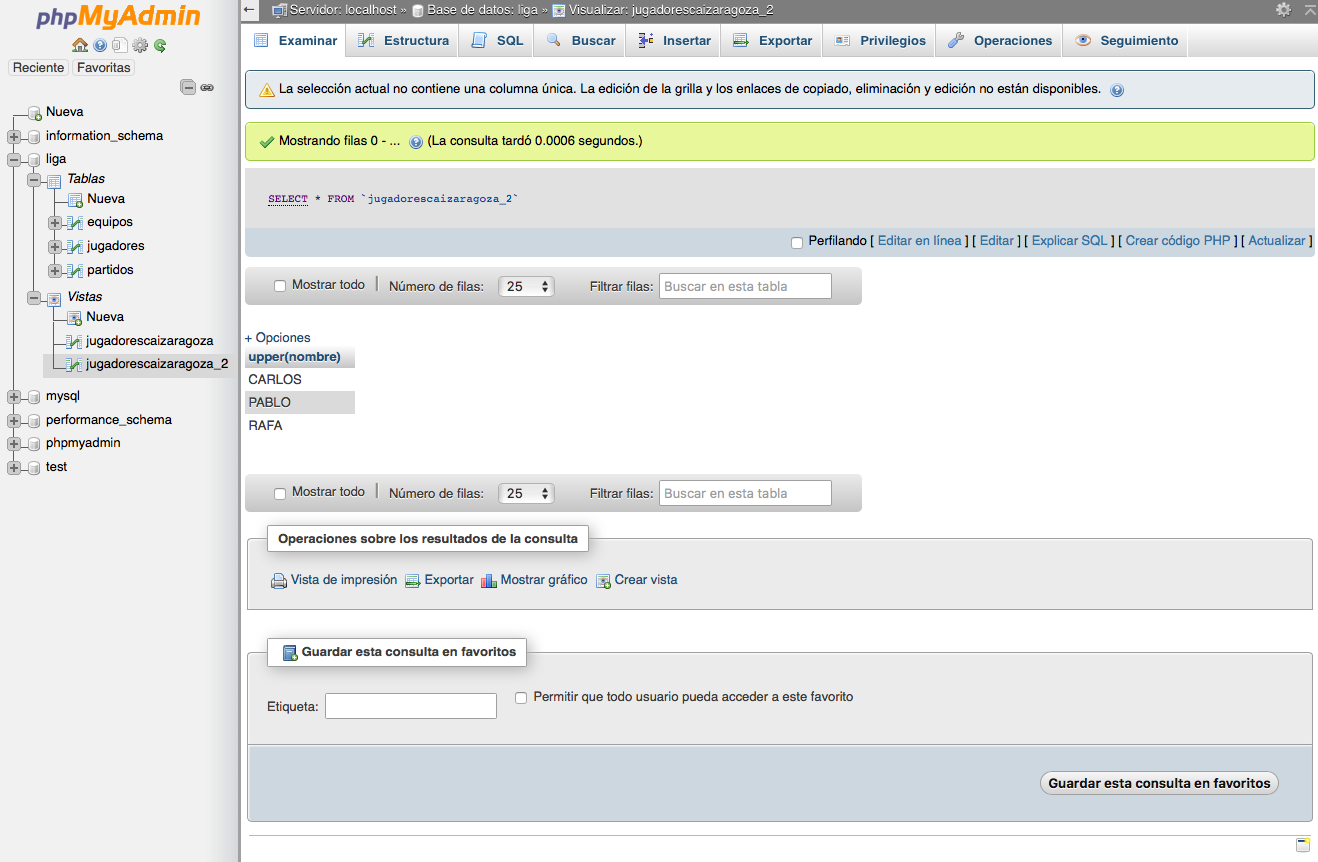


* + Crear otra basada en la anterior con solo los nombres de jugadores del CAI Zaragoza.

CREATE VIEW jugadoresCaiZaragoza\_2 AS (SELECT nombre FROM jugadoresCaiZaragoza)



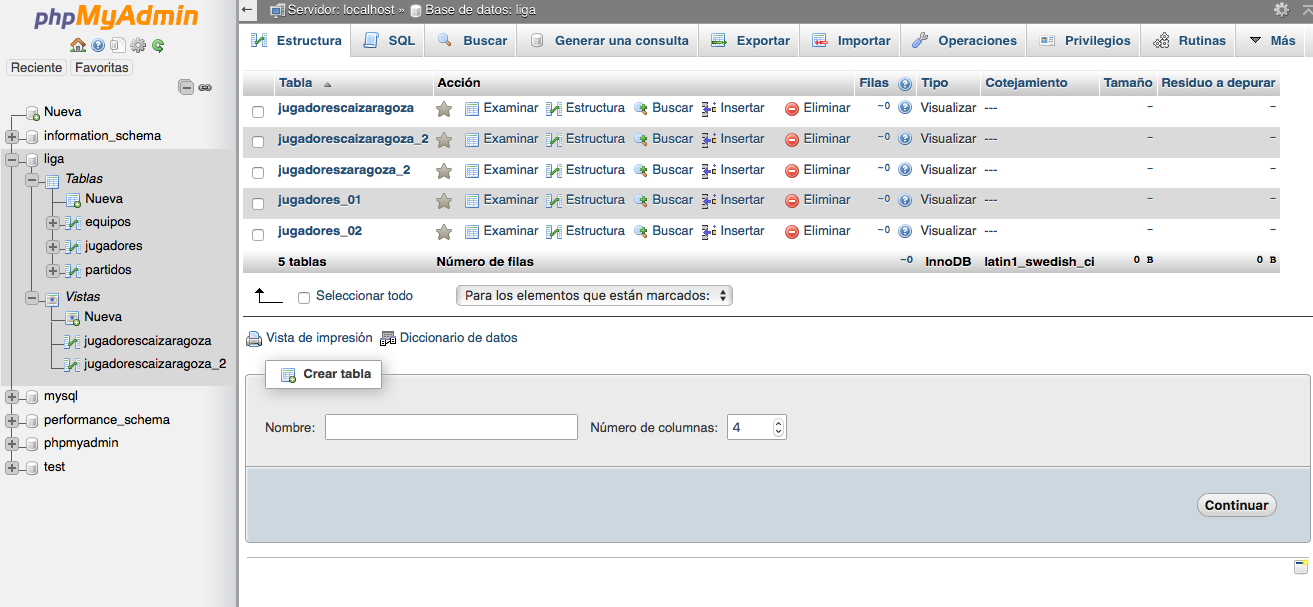
* + ¿A través de esta última vista, se podría modificar los nombres de los jugadores poniéndolos en mayúscula? En caso afirmativo, escribir como sería la sentencia.



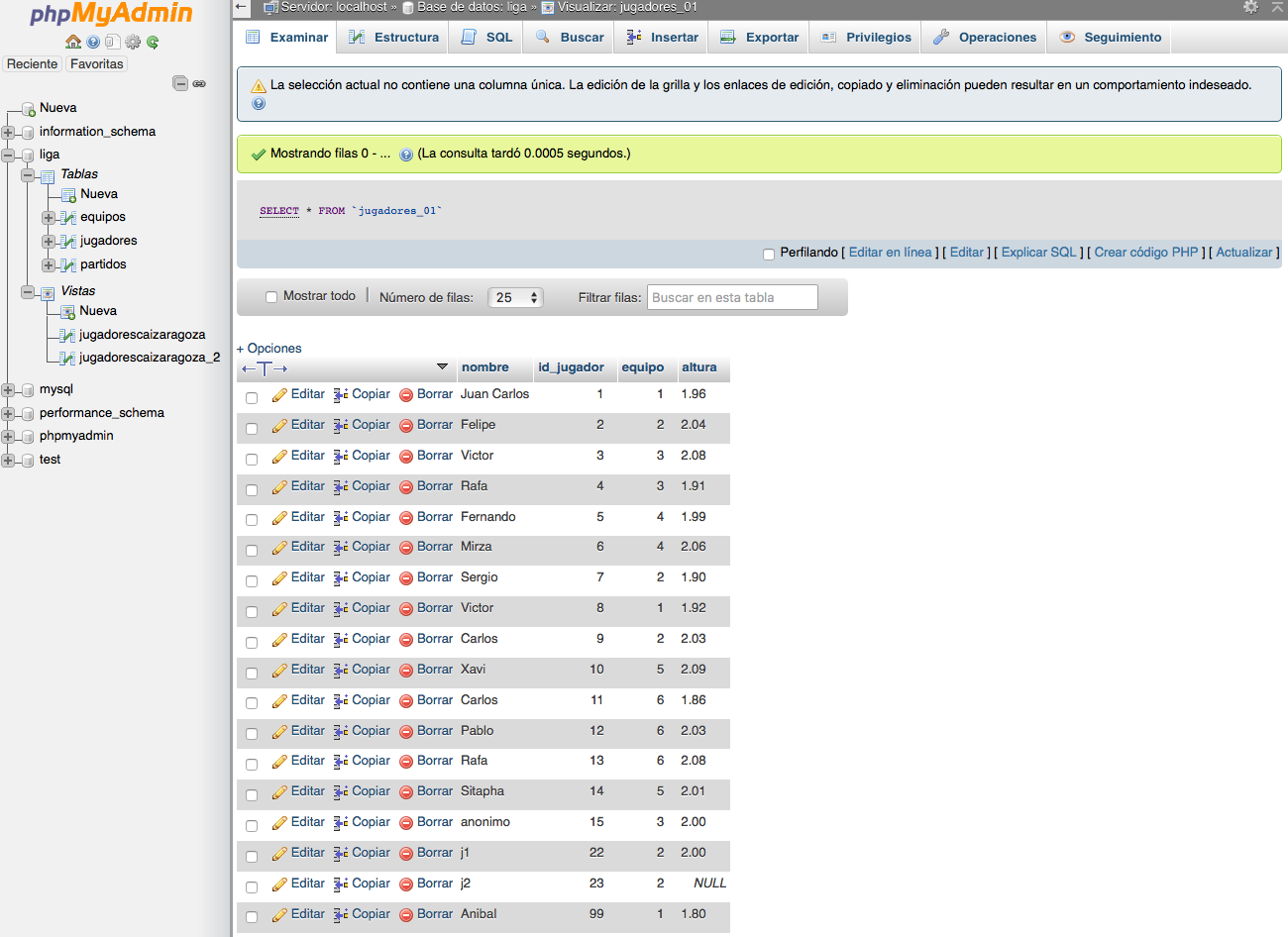
1. **Crear dos vistas, una con los campos nombre, id\_jugador, equipo y altura y otra con id\_jugador, equipo y nombre, ambas basadas en la tabla jugador. ¿Se podría insertar un nuevo jugador a través de cualquiera de ellas? En caso afirmativo, escribir como serían las sentencias.**

CREATE VIEW jugadores\_01 AS (SELECT nombre, id\_jugador, equipo, altura FROM jugadores );

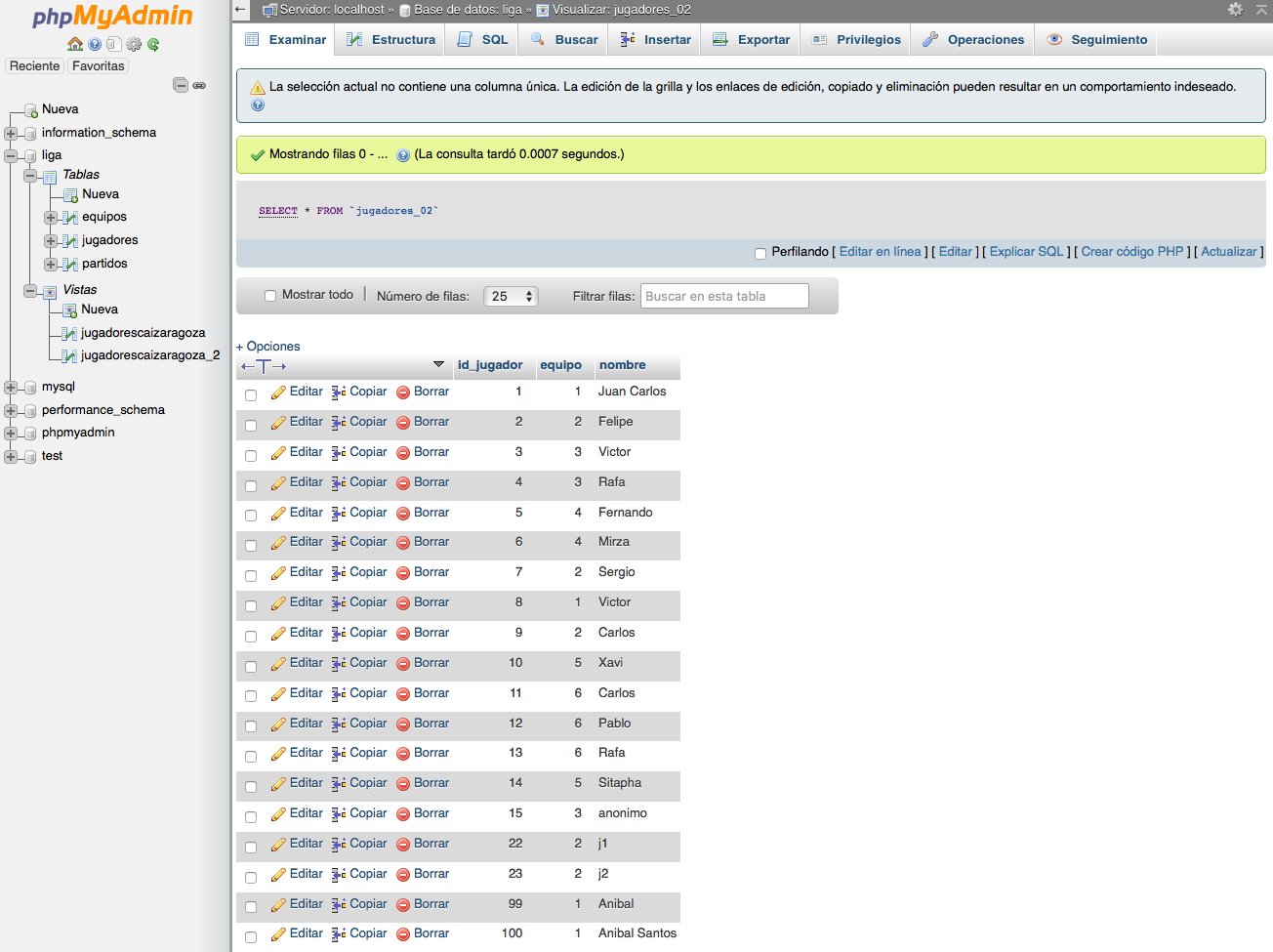
CREATE VIEW jugadores\_02 AS (SELECT id\_jugador, equipo, nombre FROM jugadores );



INSERT INTO jugadores\_01(nombre, id\_jugador, equipo, altura) VALUES ("Anibal", 99, 1, 1.80)



INSERT INTO jugadores\_02(nombre, id\_jugador, equipo) VALUES ("Anibal Santos", 100, 1)



1. Explicar qué es lo que haría la transacción siguiente:

**START TRANSACTION**

Inicia la transacción bancaria.

**USE ebanca;**

Indica que va a utilizar la base de datos ebanca.

**UPDATE cuentas SET saldo=saldo+2000 WHERE cod\_cuenta=3;**

Actualiza .la tabla cuentas aumentando el saldo en 2000 de la cuenta cuyo código es 3

**UPDATE cuentas SET saldo=saldo-2000 WHERE cod\_cuenta=4;**

Actualiza la tabla cuentas disminuyendo el saldo en 2000 de la cuenta cuyo código es 4.

**INSERT INTO MOVIMIENTOS VALUES(now(),2000,117,3,0);**

Inserta un nuevo movimiento de ingreso de 2000 en la tabla movimientos.

**INSERT INTO MOVIMIENTOS VALUES(now(),-2000,117,4,0);**

Inserta un nuevo movimiento de reintegro de 2000 en la tabla movimientos.

**COMMIT;**

Confirma los cambios realizados en la base de datos durante la transacción.