

Unknown Title



Tú dijiste:

```
username es una foreign key de otra tabla: <?php include_once "bbdd.php"; session_start(); try {  
if(!isset($_SESSION['user']) || $_SESSION['user']==""){ header("Location: acceso.php"); } $conexion =  
mysqli_connect($server, $user, $password, $bd); $nombre = $_POST['nombre']; $apellidos =  
$_POST['apellidos']; $modelopc = $_POST['modelopc']; $error = $_POST['error']; $observaciones =  
$_POST['observaciones']; $username = $_SESSION['user']; $sql = "insert into averia3 (nombre,  
apellidos, modelopc, error, observaciones, username) values ('".$nombre."', '".$apellidos."',  
".$modelopc."', '".$error."', '".$observaciones."', '".$username."')"; //echo $sql; $resultado_sql =  
mysqli_query($conexion, $sql); if (!$resultado_sql){ echo "Error en la inserción: ".mysqli_connect(); } else  
{ echo "Avería registrada correctamente<br>"; } echo "<a href='consultar.php'>LISTA DE AVERIAS</a>";  
} catch (Exception $e) { echo "Error: ".$e->getMessage(); } ?> Error: Duplicate entry 'pepe' for key  
'averia3.username_3'
```

ChatGPT Plus

Vale, el error te está dando **una pista muy clara** 😊

Vamos por partes y al grano.



Error que te da

nginx

```
Duplicate entry 'pepe' for key 'averia3.username_3'
```

Esto **NO es un problema de foreign key**, es un **problema de índice UNIQUE**.

👉 MySQL te está diciendo que en la tabla averia3 **no se puede repetir el valor del campo username**.



Qué está pasando realmente

En averia3 tienes algo así como:

sql

```
UNIQUE KEY username_3 (username)
```

Eso significa que **cada username solo puede aparecer una vez** en esa tabla.

Pero tú estás haciendo esto:

php

```
$username = $_SESSION['user']; // ej. 'pepe'
```

Y cada vez que el mismo usuario registra otra avería →

 **violación de la clave UNIQUE**

Concepto importante (examen vibes)

- **FOREIGN KEY** → controla que el valor exista en otra tabla
- **UNIQUE** → controla que NO se repita

Tu campo username:

 es foreign key

 pero además es UNIQUE (y no debería)

Solución correcta (la más habitual)

 **Opción recomendada: quitar el UNIQUE de username**

Un usuario **puede tener muchas averías**, así que la relación correcta es:

scss

```
usuarios (1) ---- (N) averia3
```

Ejecuta en MySQL:

sql

```
ALTER TABLE averia3 DROP INDEX username_3;
```

 Mantienes la **foreign key**, pero permites repetidos.

Estructura ideal de la tabla

sql

```
CREATE TABLE averia3 (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(50),
```

```
apellidos VARCHAR(100),  
modelopc VARCHAR(50),  
error TEXT,  
observaciones TEXT,  
username VARCHAR(50),  
FOREIGN KEY (username) REFERENCES usuarios(username)  
);
```

- ✓ Un usuario → muchas averías
- ✓ Integridad referencial
- ✓ Sin errores por duplicado

⚠️ Opción NO recomendada (pero posible)

Si solo quieres una avería por usuario, entonces el error es correcto y tu código **funciona bien**. Pero en una app real de incidencias... eso no tiene mucho sentido 😅

💡 Extra importante (seguridad)

Ahora mismo tu SQL es vulnerable a **SQL Injection** 😬

Si quieres, te lo paso **con prepare() y bind_param()** (muy recomendable para clase).

¿Lo quieres seguro + limpio?

Tú dijiste:

no se ha solucionado

ChatGPT Plus

Ok, vamos a **ir a tiro**, porque si **sigue saliendo el Duplicate entry**, entonces **el índice UNIQUE sigue existiendo** (o hay otro parecido) aunque creas que lo quitaste.

Ese error **NO viene de PHP**, viene **100% de MySQL**.

🔍 PASO 1 — Ver qué índices tiene REALMENTE la tabla

Ejecuta **esto tal cual** en MySQL:

sql

```
SHOW INDEX FROM averia3;
```

Busca una fila donde:

- Key_name sea algo como username, username_3, UNIQUE_username, etc.
- Non_unique = 0

👉 Ese es el culpable

🔥 PASO 2 — Eliminar TODOS los UNIQUE sobre username

Cuando veas el nombre exacto del índice, elimínalo.

Ejemplo (ajusta el nombre):

sql

```
ALTER TABLE averia3 DROP INDEX username_3;
```

⚠ Si ves más de uno, elimínalos **todos**.

🧪 PASO 3 — Confirmar que ya NO es único

Vuelve a ejecutar:

sql

```
SHOW INDEX FROM averia3;
```

- ✓ username debe aparecer con Non_unique = 1
- ✓ O directamente no tener índice

Si sigue saliendo como único → **no se ha borrado**

🧩 PASO 4 — Asegurarnos de que NO sea la PRIMARY KEY

Comprueba esto también:

sql

```
SHOW CREATE TABLE averia3;
```

Si ves algo como:

sql

```
PRIMARY KEY (username)
```

ERROR DE DISEÑO

- ✓ La PK debe ser un id

Solución:

sql

```
ALTER TABLE averia3 DROP PRIMARY KEY;  
ALTER TABLE averia3 ADD id INT AUTO_INCREMENT PRIMARY KEY;
```

Estructura FINAL correcta (la que debe quedar)

sql

```
id          → PRIMARY KEY  
username    → FOREIGN KEY (NO UNIQUE)
```

Relación correcta:

scss

```
usuarios (1) ---- (N) averia3
```

SI DESPUÉS DE TODO ESTO SIGUE FALLANDO

Pásame el resultado exacto de:

sql

```
SHOW CREATE TABLE averia3;
```

y te digo la línea exacta que está rompiendo todo.

Aquí no se escapa nada 😊

Tú dijiste:

Cannot drop column 'username': needed in a foreign key constraint
'averia3_ibfk_1';

ChatGPT Plus

Perfecto, ese mensaje ya nos dice **exactamente** qué está pasando 🔥

Y tranquilo: **no has roto nada**, es el orden lógico de MySQL.

Error

pgsql

```
Cannot drop column 'username':  
needed in a foreign key constraint 'averia3_ibfk_1'
```

👉 Traducción clara:

No puedes tocar username porque está siendo usado por una FOREIGN KEY.

Y MySQL aquí es muy estricto.

🧠 Regla CLAVE de MySQL (importantísima)

- 👉 Nunca puedes modificar / borrar un campo que participa en una foreign key
- 👉 Primero se elimina la FOREIGN KEY
- 👉 Luego ya puedes tocar índices, UNIQUE, columnas, etc.

✓ SOLUCIÓN PASO A PASO (orden correcto)

🔍 1 Ver el nombre exacto de la foreign key

Ejecuta:

sql

```
SHOW CREATE TABLE averia3;
```

Verás algo como:

sql

```
CONSTRAINT `averia3_ibfk_1`  
FOREIGN KEY (`username`) REFERENCES `usuarios`(`username`)
```

El nombre importante es:

nginx

```
averia3_ibfk_1
```

🔥 2 Eliminar SOLO la FOREIGN KEY (NO la columna)

sql

```
ALTER TABLE averia3 DROP FOREIGN KEY averia3_ibfk_1;
```

- ✓ Ahora username ya no está “protegido”

3 Eliminar el índice UNIQUE (el problema original)

Primero comprobamos índices:

sql

```
SHOW INDEX FROM averia3;
```

Localiza el índice UNIQUE (por ejemplo username_3) y bórralo:

sql

```
ALTER TABLE averia3 DROP INDEX username_3;
```

4 Volver a crear la FOREIGN KEY (correctamente)

Ahora sí, sin UNIQUE:

sql

```
ALTER TABLE averia3
ADD CONSTRAINT averia3_ibfk_1
FOREIGN KEY (username)
REFERENCES usuarios(username);
```

- ✓ username puede repetirse
- ✓ sigue siendo foreign key
- ✓ un usuario → muchas averías

Estructura FINAL correcta (modelo bueno)

sql

```
averia3
- id (PK)
- username (FK, NO UNIQUE)
```

Relación:

SCSS

usuarios (1) — (N) averia3

💣 Si algo sigue fallando

Pásame **tal cual** el resultado de:

sql

```
SHOW CREATE TABLE averia3;
```

y te digo **exactamente** qué línea sobra y cuál falta.

Esto ya está casi hecho 💪😊

Tú dijiste:

```
| averia3 | CREATE TABLE averia3 ( nparte int NOT NULL AUTO_INCREMENT, nombre varchar(50) NOT NULL, apellidos varchar(100) NOT NULL, modelopc varchar(100) CHARACTER SET utf8mb3 COLLATE utf8mb3_general_ci NOT NULL, error varchar(200) NOT NULL, observaciones varchar(500) NOT NULL, username varchar(10) NOT NULL, PRIMARY KEY (nparte), UNIQUE KEY username_5 (username), UNIQUE KEY username_6 (username), KEY username_2 (username), KEY username_4 (username), CONSTRAINT averia3_ibfk_1 FOREIGN KEY (username) REFERENCES usuario (username) ON DELETE CASCADE ON UPDATE CASCADE ) ENGINE=InnoDB AUTO_INCREMENT=13 DEFAULT CHARSET=utf8mb3 | +-----+
```

ChatGPT Plus

Perfecto, ahora **sí** tenemos la foto completa!

Y aquí está el problema clarísimo.

● **Qué está MAL en tu tabla (muy mal 😭)**

Tu tabla averia3 tiene esto:

sql

```
UNIQUE KEY `username_5` (`username`),
UNIQUE KEY `username_6` (`username`),
KEY `username_2` (`username`),
KEY `username_4` (`username`),
FOREIGN KEY (`username`) REFERENCES `usuario` (`username`)
```

👉 Tienes DOS índices UNIQUE sobre username

👉 Por eso **solo puedes insertar una avería por usuario**

👉 Da igual PHP, sesiones, o lo que hagas: MySQL lo bloquea

🧠 Diseño correcto (lo que debería ser)

- nparte → PRIMARY KEY ✓
 - username → FOREIGN KEY ✗ NO UNIQUE
 - Un usuario puede tener **muchas** averías
-

✓ SOLUCIÓN CORRECTA (ORDEN OBLIGATORIO)

1 Eliminar la FOREIGN KEY (temporalmente)

sql

```
ALTER TABLE averia3
DROP FOREIGN KEY averia3_ibfk_1;
```

2 Eliminar LOS DOS UNIQUE (los culpables)

sql

```
ALTER TABLE averia3 DROP INDEX username_5;
ALTER TABLE averia3 DROP INDEX username_6;
```

⚠ NO toques username_2 ni username_4 (son índices normales, esos sí valen)

3 Volver a crear la FOREIGN KEY (bien hecha)

sql

```
ALTER TABLE averia3
ADD CONSTRAINT averia3_ibfk_1
FOREIGN KEY (username)
REFERENCES usuario(username)
ON DELETE CASCADE
ON UPDATE CASCADE;
```

🔍 Comprobación FINAL

Ejecuta:

sql

```
SHOW INDEX FROM averia3;
```

Debe quedar algo así:

- username → Non_unique = 1
- ✗ ningún índice UNIQUE sobre username

Y luego prueba tu INSERT otra vez:

php

```
$username = $_SESSION['user']; // mismo usuario
```

👉 Ahora **sí** se insertarán varias averías del mismo usuario sin error 🎉

💡 Nota importante (esto es de examen)

Nunca pongas UNIQUE en una foreign key si la relación es 1:N

Tú la tenías convertida en **1:1 sin querer.**

Siquieres, en el siguiente mensaje:

- te limpio la tabla (quitar índices duplicados)
- o te dejo el **SQL final bonito**
- o te paso el INSERT con prepare() (pro + seguro)

Tú mandas 😊

Tú dijiste:

```
SHOW INDEX FROM averia3; +-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+ | Table | Non_unique |
Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type
| Comment | Index_comment | Visible | Expression | +-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ | averia3 |
0 | PRIMARY | 1 | nparte | A | 1 | NULL | NULL | BTREE || YES | NULL | | averia3 | 1 | username_2 | 1 |
username | A | 1 | NULL | NULL | BTREE || YES | NULL | | averia3 | 1 | username_4 | 1 | username | A
| 1 | NULL | NULL | BTREE || YES | NULL | +-----+-----+-----+-----+-----+
---+-----+-----+-----+-----+-----+-----+-----+-----+
```

Perfecto. Ahora Sí está bien

Gracias por pasar el SHOW INDEX, con esto ya no hay duda.

Estado actual de averia3 (CORRECTO)

Lo que tienes ahora:

pgsql

```
PRIMARY      → nparte
username_2    → NO UNIQUE
username_4    → NO UNIQUE
```

Fíjate en esto

Non_unique = 1 en ambos índices de username

YA NO hay ningún UNIQUE sobre username

Eso significa:

- ✓ username **puede repetirse**
 - ✓ sigue siendo **foreign key**
 - ✓ relación **usuario (1) → averías (N)**
 - ✓ el error Duplicate entry 'pepe' **YA NO debería aparecer**
-

Siguiente paso (muy importante)

Prueba **exactamente el mismo INSERT** que antes, sin cambiar PHP:

php

```
$username = $_SESSION['user'];
```

Inserta **otra avería con el mismo usuario**.

Debe entrar sin error.

Detalle técnico (no es error, pero ojo)

Tienes **dos índices normales iguales**:

nginx

```
username_2
username_4
```

No rompe nada, pero **es redundante** y no tiene sentido mantener ambos.

(Opcional pero recomendable)

Quédate solo con uno:

sql

```
ALTER TABLE averia3 DROP INDEX username_4;
```

Con **un solo índice** basta para la foreign key y las búsquedas.

Resumen claro (para clase / examen)

- El error NO era PHP ✗
 - Era MySQL por:
 - índices UNIQUE duplicados ✗
 - foreign key bien definida ✓
 - Ahora:
 - PK → nparte
 - FK → username (NO UNIQUE)
 - diseño correcto ✓
-

Si ahora **aún fallara** (muy raro ya), dime **el error exacto** que salga al insertar.

Pero con lo que has pasado... esto ya está finiquitado 💪 😊

ChatGPT puede cometer errores. Considera verificar la información importante. Ver [preferencias de cookies](#).