



Scripts de Base de Datos - Zabala Gaietak

Esta carpeta contiene scripts SQL para gestionar la base de datos MySQL del portal de RRHH de Zabala Gaietak.



Índice de Scripts

Script	Propósito	¿Cuándo usar?
mysql_zabala_gaietak_fresh_install.sql	Instalación limpia	Base de datos nueva, primera vez
mysql_migration_fix_vacation_system.sql	Migración sin pérdida de datos	Base de datos existente, conservar usuarios
cleanup_vacation_db.sql	Limpieza manual	Corregir datos corruptos (obsoleto)
mysql_vacation_triggers.sql	Solo triggers	Añadir solo triggers (obsoleto)



Instalación Limpia (Recomendado para desarrollo)

Archivo: mysql_zabala_gaietak_fresh_install.sql

✓ Cuándo usar:

- Primera instalación
- Entorno de desarrollo local
- Quieres empezar desde cero
- No te importa perder datos existentes



Lo que incluye:

- 20 tablas completas con relaciones (FOREIGN KEYS)
- Datos de ejemplo (departamentos, usuario admin)
- 3 triggers automáticos para gestión de vacaciones
- Estructura optimizada para MySQL/MariaDB

- Soporte completo de UUIDs
- Usuario admin: admin@zabalagaietak.com / Admin123!

Cómo usar:

Opción 1: phpMyAdmin (InfinityFree)

1. Accede a phpMyAdmin en tu hosting
2. Selecciona la base de datos if0_40982238_zabala_gaietak
3. Ve a la pestaña **SQL**
4. Copia y pega el contenido del script
5. Haz clic en **Ejecutar**
6. ¡Listo! Ya puedes hacer login con admin@zabalagaietak.com

Opción 2: MySQL CLI

```
mysql -u tu_usuario -p if0_40982238_zabala_gaietak < mysql_zabala_gaietak_fresh_ins
```

ADVERTENCIA:

Este script **BORRA TODAS LAS TABLAS EXISTENTES**. Solo úsalo si estás seguro.

Migración con Conservación de Datos

Archivo: mysql_migration_fix_vacation_system.sql

Cuándo usar:

- Ya tienes datos en producción
- Quieres conservar usuarios y empleados existentes
- Necesitas arreglar el sistema de vacaciones sin perder datos
- Prefieres una actualización segura

Lo que hace:

1. Crea copias de seguridad automáticas (*_backup tables)
2. Corrige la tabla vacation_balances (elimina campo GENERATED)
3. Actualiza vacation_requests para soportar UUIDs
4. Recalcula pending_days y used_days desde datos existentes
5. Instala triggers automáticos

6. Muestra reporte de verificación al final

Cómo usar:

Opción 1: phpMyAdmin

1. **IMPORTANTE:** Haz backup manual primero (Exportar > Formato SQL)
2. Selecciona tu base de datos
3. Ve a la pestaña **SQL**
4. Copia y pega el contenido de `mysql_migration_fix_vacation_system.sql`
5. Haz clic en **Ejecutar**
6. Revisa el reporte de verificación al final

Opción 2: MySQL CLI

```
# Hacer backup primero
mysqldump -u tu_usuario -p if0_40982238_zabala_gaileta > backup_$(date +%Y%m%d_%H%M)

# Ejecutar migración
mysql -u tu_usuario -p if0_40982238_zabala_gaileta < mysql_migration_fix_vacation_sy
```

Verificación post-migración:

```
-- Ver balances de vacaciones
SELECT
    e.first_name,
    e.last_name,
    vb.total_days,
    vb.used_days,
    vb.pending_days,
    (vb.total_days - vb.used_days - vb.pending_days) AS available_days
FROM vacation_balances vb
JOIN employees e ON e.id = vb.employee_id;

-- Ver triggers instalados
SHOW TRIGGERS LIKE 'vacation_requests';
```

Limpiar backups (después de verificar):

```
DROP TABLE IF EXISTS vacation_balances_backup;
DROP TABLE IF EXISTS vacation_requests_backup;
```

Probar el Sistema de Vacaciones

Después de ejecutar cualquiera de los scripts:

1 Verifica el balance inicial:

```
SELECT * FROM vacation_balances WHERE employee_id = 'ff15d24e-fa89-11f0-9b20-fab8ad3a'
```

Resultado esperado:

```
total_days: 22
used_days: 0
pending_days: 0
```

2 Crea una solicitud de prueba desde la web:

- Login: admin@zabalagaietak.com / Admin123!
- Ve a **Vacaciones** → **Solicitar Vacaciones**
- Selecciona fechas (ej: 2026-03-01 a 2026-03-05)
- Envía la solicitud

3 Verifica que el trigger funcionó:

```
SELECT * FROM vacation_balances WHERE employee_id = 'ff15d24e-fa89-11f0-9b20-fab8ad3a'
```

Resultado esperado:

```
total_days: 22
used_days: 0
pending_days: 5 ← ¡Incrementado automáticamente!
```

4 Verifica la solicitud creada:

```
SELECT id, status, start_date, end_date, total_days
FROM vacation_requests
WHERE employee_id = 'ff15d24e-fa89-11f0-9b20-fab8ad3a19ce' ;
```

Solución de Problemas

Problema: “Duplicate entry ‘ for key ‘PRIMARY’”

Solución: El código PHP no está generando UUIDs. Verifica que `VacationService.php` tenga el método `generateUUID()`.

Problema: “pending_days se resetea solo”

Solución: La tabla antigua tenía pending_days como campo GENERATED. Ejecuta la migración para convertirlo en campo manual.

Problema: “available_days siempre es 0”

Solución:

1. Verifica que pending_days no sea GENERATED:

```
SHOW CREATE TABLE vacation_balances;
```

2. Si ves GENERATED ALWAYS AS, ejecuta la migración.

Problema: “Los triggers no se ejecutan”

Solución:

1. Verifica que existen:

```
SHOW TRIGGERS LIKE 'vacation_requests';
```

2. Si no aparecen, ejecuta solo la parte de triggers del script de migración.

Problema: “Foreign key constraint fails”

Solución:

- InfinityFree usa MariaDB que no soporta todas las constraints.
- El script de instalación limpia usa ENGINE=InnoDB para maximizar compatibilidad.
- Si falla, cambia a ENGINE=MyISAM (sin foreign keys):

```
-- Buscar y reemplazar en el script:  
ENGINE=InnoDB → ENGINE=MyISAM
```

Estructura de Datos Clave

vacation_balances

id	VARCHAR(36)	PRIMARY KEY (UUID)
employee_id	VARCHAR(36)	FK → employees.id
year	INT	Año fiscal (ej: 2026)
total_days	INT	Días totales asignados (22)
used_days	INT	Días ya usados (aprobados y tomados)

pending_days	INT	Días reservados (solicitudes pendientes)
carried_over	INT	Días arrastrados del año anterior

Fórmula:

```
available_days = total_days - used_days - pending_days
```

vacation_requests

id	VARCHAR(36)	PRIMARY KEY (UUID generado en PHP)
employee_id	VARCHAR(36)	FK → employees.id
start_date	DATE	Fecha inicio
end_date	DATE	Fecha fin
total_days	DECIMAL(5,2)	Días calculados (días laborables)
status	VARCHAR(20)	PENDING MANAGER_APPROVED APPROVED REJECTED
manager_approval_by	VARCHAR(36)	FK → users.id (jefe de sección)
hr_approval_by	VARCHAR(36)	FK → users.id (RRHH)

Estados del flujo:

1. PENDING → Solicitud enviada, esperando aprobación de jefe
2. MANAGER_APPROVED → Aprobada por jefe, esperando RRHH
3. APPROVED → Aprobada por RRHH, días movidos de pending → used
4. REJECTED → Rechazada, días liberados de pending

Seguridad y Backups

Antes de cualquier cambio en producción:

1. Exportar base de datos completa:

```
# phpMyAdmin: Exportar > Formato SQL > Guardar archivo
# O desde CLI:
mysqldump -u user -p database > backup_YYYY-MM-DD.sql
```

2. Probar en local primero:

- Usa XAMPP/MAMP/Laragon
- Crea una base de datos de prueba
- Ejecuta el script
- Verifica que todo funciona

3. Documentar cambios:

- Anota qué script ejecutaste
 - Guarda el archivo de backup
 - Registra fecha y hora
-

Notas de Compatibilidad

MySQL vs MariaDB vs PostgreSQL

Característica	PostgreSQL	MySQL	MariaDB	Script
UUID nativo	✓ uuid_generate_v4()	✗	✗	✓ Genera en PHP
RETURNING	✓	✗	✗	✓ Genera UUID primero
Triggers complejos	✓ PL/pgSQL	⚠ Limitado	⚠ Limitado	✓ Simplificado
FOREIGN KEYS	✓	✓ InnoDB	✓ InnoDB	✓ Incluidas
Generated columns	✓	✓	✓	⚠ Evitadas en balances

InfinityFree específico:

- Usa MariaDB 11.4.9
 - Soporta triggers (verificado ✓)
 - Soporta InnoDB con FOREIGN KEYS
 - **NO** soporta funciones custom (UUID)
 - **Solución:** Generamos UUIDs en PHP
-

Próximos Pasos

Después de ejecutar el script:

1. ✓ **Login:** admin@zabalagaileta.com / Admin123!
2. ✓ **Crear empleados:** Gestión → Empleados → Añadir
3. ✓ **Probar vacaciones:** Vacaciones → Solicitar
4. ✓ **Verificar triggers:** Comprobar que pending_days se actualiza
5. ✓ **Aprobar solicitudes:** Como admin, aprobar una solicitud de prueba

-
6.  **Verificar flujo completo:** PENDING → MANAGER_APPROVED → APPROVED
-

Referencias

- [ISO 27001:2022 Compliance](#)
 - [GDPR Documentation](#)
 - [IEC 62443 OT Security](#)
 - [API Documentation](#)
 - [Migration Plan](#)
-

Soporte

Si encuentras problemas:

1. **Revisa los logs de error:** phpMyAdmin → Variables → log_error
 2. **Verifica la estructura:** SHOW CREATE TABLE vacation_balances;
 3. **Comprueba triggers:** SHOW TRIGGERS LIKE 'vacation_requests';
 4. **Consulta la documentación:** Este archivo README
 5. **Restaura el backup** si algo falla gravemente
-

Última actualización: 2026-02-05

Versión: 2.0.0

Autor: Zabala Gaietak DevTeam

Licencia: Propietario - Uso interno únicamente