# Gestok App Base De Datos

## Script Base de Datos

CREATE TABLE `company` (

`id` int PRIMARY KEY AUTO\_INCREMENT,

`company\_name` varchar(200),

`address` text,

`phone` varchar(20),

`owner` varchar(100),

`email` VARCHAR(255) UNIQUE NOT NULL,

`website` varchar(150),

`plan` enum("free","basic","premium"),

`deployment\_type` enum("saas","on\_premise"),

`status` enum("activa","suspendida","inactiva"),

`created\_at` datetime,

`comments` text

);

CREATE TABLE `stores` (

`id` int PRIMARY KEY AUTO\_INCREMENT,

`business\_name` varchar(150),

`company\_id` int,

`address` text,

`phone` varchar(20),

`manager` varchar(100),

`email` VARCHAR(255) UNIQUE NOT NULL,

`status` enum("activa","suspendida","inactiva"),

`created\_at` datetime,

`comments` text

);

CREATE TABLE `users` (

`id` int PRIMARY KEY AUTO\_INCREMENT,

`name` varchar(100),

`email` VARCHAR(255) UNIQUE NOT NULL,

`password` varchar(255),

`store\_id` int,

`rol\_id` int,

`status` boolean

);

CREATE TABLE `rols` (

`id` int PRIMARY KEY AUTO\_INCREMENT,

`name` varchar(50) UNIQUE

);

CREATE TABLE `permissions` (

`id` int PRIMARY KEY AUTO\_INCREMENT,

`name` varchar(100),

`code` varchar(100) UNIQUE

);

CREATE TABLE `permissions\_rols` (

`id` int PRIMARY KEY AUTO\_INCREMENT,

`rol\_id` int,

`permission\_id` int

);

CREATE TABLE `product\_type` (

`id` int PRIMARY KEY AUTO\_INCREMENT,

`name` varchar(100),

`price` float(2),

`type` varchar(100)

);

CREATE TABLE `detail\_sale` (

`id` int PRIMARY KEY AUTO\_INCREMENT,

`sale\_id` int,

`product\_type\_id` int,

`quantity` int,

`unit\_price` float,

`subtotal` float

);

CREATE TABLE `sale` (

`id` int PRIMARY KEY AUTO\_INCREMENT,

`date\_time` datetime,

`total` float(2),

`user\_id` int,

`method\_payment` enum("efectivo","tarjeta")

);

CREATE TABLE `inventory` (

`id` int PRIMARY KEY AUTO\_INCREMENT,

`name` varchar(100),

`quatity` varchar(100),

`provider\_id` int,

`store\_id` int

);

CREATE TABLE `provider` (

`id` int PRIMARY KEY AUTO\_INCREMENT,

`company\_name` varchar(100),

`contact\_name` varchar(100),

`phone` int,

`email` VARCHAR(255) UNIQUE NOT NULL,

`code` varchar(50)

);

CREATE TABLE `purchase` (

`id` int PRIMARY KEY AUTO\_INCREMENT,

`user\_id` int,

`store\_id` int,

`total` float,

`date\_time` date

);

CREATE TABLE `purchaseDetail` (

`id` int PRIMARY KEY AUTO\_INCREMENT,

`purchase\_id` int,

`inventory\_id` int,

`quantity` float,

`unit\_price` float,

`subtotal` float,

`date\_time` datetime

);

CREATE TABLE `reports` (

`id` int PRIMARY KEY AUTO\_INCREMENT,

`user\_id` int,

`title` varchar(50),

`description` varchar(200),

`created\_at` date,

`updated\_at` date

);

CREATE TABLE `files` (

`id` int PRIMARY KEY AUTO\_INCREMENT,

`file\_name` int,

`store\_id` int,

`type` varchar(50),

`url` text,

`upload\_date` datetime

);

CREATE TABLE `resource` (

`id` int PRIMARY KEY AUTO\_INCREMENT,

`type` varchar(50),

`reference\_id` int

);

CREATE TABLE `logs\_sistema` (

`id` int PRIMARY KEY AUTO\_INCREMENT,

`user\_id` int,

`resource\_id` int,

`accion` enum("view","modify","delete"),

`from\_ip` varchar(45),

`upload\_date` datetime

);

ALTER TABLE `stores` ADD FOREIGN KEY (`company\_id`) REFERENCES `company` (`id`);

ALTER TABLE `users` ADD FOREIGN KEY (`store\_id`) REFERENCES `stores` (`id`);

ALTER TABLE `users` ADD FOREIGN KEY (`rol\_id`) REFERENCES `rols` (`id`);

ALTER TABLE `permissions\_rols` ADD FOREIGN KEY (`rol\_id`) REFERENCES `rols` (`id`);

ALTER TABLE `permissions\_rols` ADD FOREIGN KEY (`permission\_id`) REFERENCES `permissions` (`id`);

ALTER TABLE `detail\_sale` ADD FOREIGN KEY (`sale\_id`) REFERENCES `sale` (`id`);

ALTER TABLE `detail\_sale` ADD FOREIGN KEY (`product\_type\_id`) REFERENCES `product\_type` (`id`);

ALTER TABLE `sale` ADD FOREIGN KEY (`user\_id`) REFERENCES `users` (`id`);

ALTER TABLE `inventory` ADD FOREIGN KEY (`provider\_id`) REFERENCES `provider` (`id`);

ALTER TABLE `purchase` ADD FOREIGN KEY (`user\_id`) REFERENCES `users` (`id`);

ALTER TABLE `purchase` ADD FOREIGN KEY (`store\_id`) REFERENCES `stores` (`id`);

ALTER TABLE `purchaseDetail` ADD FOREIGN KEY (`inventory\_id`) REFERENCES `inventory` (`id`);

ALTER TABLE `purchaseDetail` ADD FOREIGN KEY (`purchase\_id`) REFERENCES `purchase` (`id`);

ALTER TABLE `logs\_sistema` ADD CONSTRAINT `logs\_sistema\_ibfk\_1` FOREIGN KEY (`resource\_id`) REFERENCES `resource`(`id`);

ALTER TABLE `logs\_sistema` ADD FOREIGN KEY (`user\_id`) REFERENCES `users` (`id`);

ALTER TABLE `files` ADD FOREIGN KEY (`store\_id`) REFERENCES `stores` (`id`);

ALTER TABLE `reports` ADD FOREIGN KEY (`user\_id`) REFERENCES `users` (`id`);

## EER Base de Datos

A computer screen shot of a computer

AI-generated content may be incorrect.



# SQL QA Workbench - Reporte de Migración

------------------------------------------------------------------------------------  
MySQL Workbench Migration Wizard Report  
  
Date: Fri Aug 22 00:25:05 2025  
Source: MySQL 9.4.0  
Target: MySQL 9.4.0  
------------------------------------------------------------------------------------

# I. Migration

## 1. Summary

Number of migrated schemas: 1  
  
1. gestok\_app  
Source Schema: gestok\_app  
  
- Tables: 17  
- Triggers: 0  
- Views: 0  
- Stored Procedures: 0  
- Functions: 0

## 2. Migration Issues

No issues detected.

## 3. Object Creation Issues

No issues detected.

## 4. Migration Details

Se listan las tablas migradas con sus columnas, claves foráneas e índices.

* 4.1. company (empresa principal) - 12 columnas, PK en id, FK en company\_id en stores.
* 4.2. detail\_sale - relación con sale y product\_type.
* 4.3. files - relación con stores.
* 4.4. inventory - relación con provider.
* 4.5. logs\_sistema - relación con resource.
* 4.6. permissions - catálogo de permisos.
* 4.7. permissions\_rols - relación permisos/roles.
* 4.8. product\_type - catálogo de productos.
* 4.9. provider - proveedores.
* 4.10. purchase - relación con users y stores.
* 4.11. purchasedetail - relación con purchase e inventory.
* 4.12. reports - relación con users.
* 4.13. resource - recursos relacionados.
* 4.14. rols - catálogo de roles.
* 4.15. sale - relación con users.
* 4.16. stores - relación con company.
* 4.17. users - relación con stores y rols.

# Pruebas Manuales

Insert Into Tables

Company

INSERT INTO company

VALUES(3, 'Supermercado La Central', 'Av. Siempre Viva 742', '22334455',

'María López', 'contacto@central.com', 'lacentral.com', 'Premium',

'SaaS', 1, NOW(), 'Cliente premium en prueba');

INSERT INTO company

VALUES(4, 'Hospital El Buen Samaritano', 'Km 15 Carretera a Sonsonate', '24567890',

'Dr. Carlos Ruiz', 'info@buenhospi.com', 'buenhospi.com', 'premium',

'on\_premise', 'suspendida', NOW(), 'Hospital con instalación local');

INSERT INTO company

VALUES(5, 'Panadería El Trigal', 'Col. Escalón, San Salvador', '76543210',

'José Martínez', 'ventas@trigal.com', 'trigal.com', 'basic',

'on\_premise', 'activa', NOW(), 'Negocio pequeño en plan básico local');

-- Cliente 4

INSERT INTO company

VALUES(6, 'Universidad Nacional', 'Ciudad Universitaria, San Salvador', '22001122',

'Rectora Ana Torres', 'rectoria@unacional.edu.sv', 'unacional.edu.sv', 'free',

'saas', 'activa', NOW(), 'Universidad en plan free SaaS');

-- Cliente 5

INSERT INTO company

VALUES(7, 'Farmacia Central', 'Calle Principal #45, Santa Tecla', '23456789',

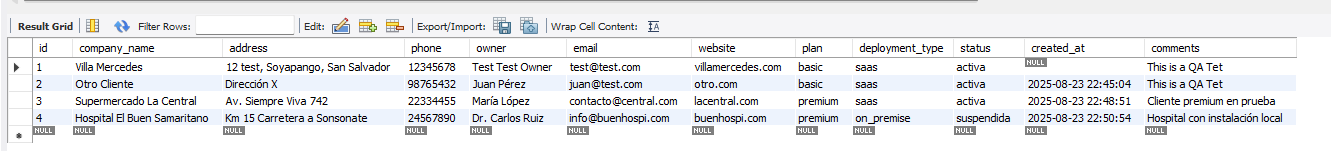
'Carlos Méndez', 'farmacia@central.com', 'farmaciacentral.com', 'basic',

'saas', 'inactiva', NOW(), 'Farmacia en plan básico SaaS');

Update Table

Company

Antes de Update



Despues de Update

-- Update Cliente 1: cambiar status a inactiva y descripción

UPDATE company

SET status = 'inactiva',

comments = 'Cliente premium pasó a inactivo'

WHERE id = 3;

-- Update Cliente 2: cambiar plan de premium a basic y tipo a saas

UPDATE company

SET plan = 'basic',

deployment\_type = 'saas',

comments = 'Hospital cambió de on\_premise a SaaS plan basic'

WHERE id = 4;

-- Update Cliente 3: suspender la panadería

UPDATE company

SET status = 'suspendida',

comments = 'Panadería suspendida por falta de pago'

WHERE id = 5;

-- Update Cliente 4: actualizar a premium

UPDATE company

SET plan = 'premium',

comments = 'Universidad subió al plan premium SaaS'

WHERE id = 6;

-- Update Cliente 5: reactivar la farmacia

UPDATE company

SET status = 'activa',

comments = 'Farmacia reactivada después de soporte'

WHERE id = 7;

A screenshot of a computer

AI-generated content may be incorrect.

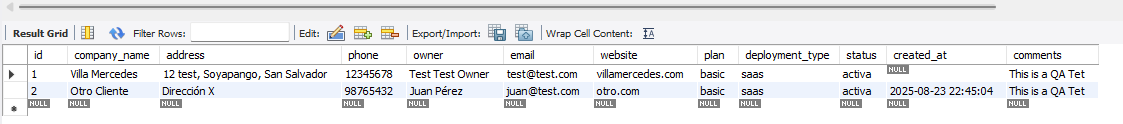
Delete Table

-- Eliminar un registro

DELETE FROM company WHERE id = 7;

-- Eliminar varios registros

DELETE FROM company WHERE id BETWEEN 3 AND 6;



Reinsertar datos eliminados

-- Reinserta registros para pruebas

INSERT INTO company

VALUES(3, 'Supermercado La Central', 'Av. Siempre Viva 742', '22334455',

'María López', 'contacto@central.com', 'lacentral.com', 'premium',

'saas', 'activa', NOW(), 'Cliente premium en prueba');

INSERT INTO company

VALUES(4, 'Hospital El Buen Samaritano', 'Km 15 Carretera a Sonsonate', '24567890',

'Dr. Carlos Ruiz', 'info@buenhospi.com', 'buenhospi.com', 'premium',

'on\_premise', 'suspendida', NOW(), 'Hospital con instalación local');

INSERT INTO company

VALUES(5, 'Panadería El Trigal', 'Col. Escalón, San Salvador', '76543210',

'José Martínez', 'ventas@trigal.com', 'trigal.com', 'basic',

'on\_premise', 'activa', NOW(), 'Negocio pequeño en plan básico local');

INSERT INTO company

VALUES(6, 'Universidad Nacional', 'Ciudad Universitaria, San Salvador', '22001122',

'Rectora Ana Torres', 'rectoria@unacional.edu.sv', 'unacional.edu.sv', 'free',

'saas', 'activa', NOW(), 'Universidad en plan free SaaS');

INSERT INTO company

VALUES(7, 'Farmacia Central', 'Calle Principal #45, Santa Tecla', '23456789',

'Carlos Méndez', 'farmacia@central.com', 'farmaciacentral.com', 'basic',

'saas', 'inactiva', NOW(), 'Farmacia en plan básico SaaS');

A screenshot of a computer

AI-generated content may be incorrect.

Stores Table

Create

-- Enum "stores\_status\_enum" {"activa""suspendida""inactiva"}

-- Store 1

INSERT INTO stores

VALUES(1, 'test', 1, '12 test', '12345678', 'manager test', 'test@test.com', 1, NOW(), 'En proceso de migracion');

-- Store 2

INSERT INTO stores

VALUES(2, 'La Central', 3, 'Av. Siempre Viva 742', '22334455', 'María López', 'central@super.com', 2, NOW(), 'Activo');

-- Store 3

INSERT INTO stores

VALUES(3, 'Buen Samaritano', 4, 'Km 15 Carretera a Sonsonate', '24567890', 'Dr. Carlos Ruiz', 'info@hospital.com', 3, NOW(), 'En remodelación');

-- Store 4

INSERT INTO stores

VALUES(4, 'Panadería El Trigal', 5, 'Col. Escalón, San Salvador', '76543210', 'José Martínez', 'ventas@trigal.com', 1, NOW(), 'Abierto');

-- Store 5

INSERT INTO stores

VALUES(5, 'Farmacia Central', 7, 'Calle Principal #45, Santa Tecla', '23456789', 'Carlos Méndez', 'farmacia@central.com', 2, NOW(), 'Activo')

Update Stores

-- Store 1: cambiar estado y descripción

UPDATE stores

SET comments = 'Migración completada'

WHERE id = 1;

-- Store 2: cambiar manager y email

UPDATE stores

SET manager = 'María López Actualizada',

email = 'maria.lopez@central.com'

WHERE id = 2;

-- Store 6: cambiar company\_id a 3 para probar relación

UPDATE stores

SET company\_id = 3,

comments = 'Cambiada de company\_id para test'

WHERE id = 6;

-- Store 8: actualizar status/descripción

UPDATE stores

SET comments = 'Hospital Satélite activo'

WHERE id = 8;

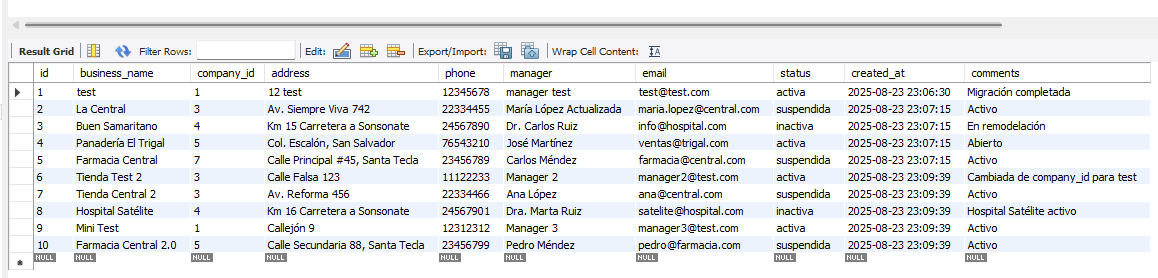
-- Store 10: cambiar nombre y teléfono

UPDATE stores

SET business\_name = 'Farmacia Central 2.0',

phone = '23456799'

WHERE id = 10;



Delete Stores

-- Eliminar una store específica

DELETE FROM stores WHERE id = 10;

-- Eliminar varias stores por company\_id

DELETE FROM stores WHERE company\_id = 1;

A screenshot of a computer

AI-generated content may be incorrect.

Table Users

-- =========================================

-- INSERT: Crear usuarios de prueba

-- =========================================

INSERT INTO users VALUES(1,'test', 'test@test.com','password2025',3,2,1,1,NOW(),NOW());

INSERT INTO users VALUES(2,'Juan Pérez', 'juan@test.com','password123',1,1,1,1,NOW(),NOW());

INSERT INTO users VALUES(3,'María López', 'maria@test.com','password456',2,3,1,1,NOW(),NOW());

INSERT INTO users VALUES(4,'Carlos Méndez', 'carlos@test.com','password789',5,4,1,1,NOW(),NOW());

INSERT INTO users VALUES(5,'Ana Torres', 'ana@test.com','passwordabc',6,2,1,1,NOW(),NOW());

-- =========================================

-- SELECT: Ver todos los usuarios

-- =========================================

SELECT \* FROM users;

-- SELECT: Filtrar usuarios por rol\_id = 2

SELECT \* FROM users WHERE rol\_id = 2;

-- SELECT: Filtrar usuarios por company\_id = 1

SELECT \* FROM users WHERE company\_id = 1;

-- =========================================

-- UPDATE: Modificar datos de prueba

-- =========================================

-- Cambiar email y nombre del usuario 1

UPDATE users

SET nombre = 'Usuario Test Actualizado', email = 'test.update@test.com'

WHERE id = 1;

-- Cambiar rol y company del usuario 2

UPDATE users

SET rol\_id = 3, company\_id = 3

WHERE id = 2;

-- Cambiar contraseña del usuario 3

UPDATE users

SET password = 'newpassword456'

WHERE id = 3;

-- =========================================

-- DELETE: Eliminar usuarios de prueba

-- =========================================

-- Eliminar un usuario específico

DELETE FROM users WHERE id = 5;

-- Eliminar varios usuarios por company\_id

DELETE FROM users WHERE company\_id = 1;

-- Limpiar toda la tabla de prueba

DELETE FROM users;

1. Conteo de tablas migradas:  
```sql  
SELECT COUNT(\*) FROM information\_schema.tables   
WHERE table\_schema = 'gestok\_app';  
```  
  
2. Conteo de registros en tablas principales:  
```sql  
SELECT COUNT(\*) FROM users;  
SELECT COUNT(\*) FROM stores;  
SELECT COUNT(\*) FROM company;  
```  
  
3. Validación de integridad referencial:  
```sql  
SELECT table\_name, constraint\_name   
FROM information\_schema.key\_column\_usage   
WHERE referenced\_table\_schema = 'gestok\_app';  
```

# Inserciones y Consultas de Prueba

-- Inserciones de prueba  
INSERT INTO company (id, company\_name, address, phone, owner, email, website, plan, deployment\_type, status, created\_at, comments)  
VALUES (999, 'QA Company', 'San Salvador', '7777-0000', 'Tester', 'qa@test.com', 'www.qacompany.com', 'basic', 'saas', 'activa', NOW(), 'Registro de prueba');  
  
INSERT INTO stores (id, business\_name, company\_id, address, phone, manager, email, status, created\_at, comments)  
VALUES (999, 'QA Store', 999, 'Santa Tecla', '2222-0000', 'Manager QA', 'store@test.com', 'activa', NOW(), 'Registro de prueba');  
  
INSERT INTO users (id, name, email, password, store\_id, rol\_id, status)  
VALUES (999, 'Usuario QA', 'userqa@test.com', 'Password@123', 999, 1, 1);  
  
-- Consultas de validación  
SELECT \* FROM company WHERE id = 999;  
  
SELECT s.id, s.business\_name AS store, c.company\_name AS company  
FROM stores s  
JOIN company c ON s.company\_id = c.id  
WHERE s.id = 999;  
  
SELECT u.id, u.name AS usuario, s.business\_name AS store, c.company\_name AS company  
FROM users u  
JOIN stores s ON u.store\_id = s.id  
JOIN company c ON s.company\_id = c.id  
WHERE u.id = 999;  
  
-- Limpieza (rollback de datos de prueba)  
DELETE FROM users WHERE id = 999;  
DELETE FROM stores WHERE id = 999;  
DELETE FROM company WHERE id = 999;