# SGestok 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

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;