

# PLAN DE PRUEBAS

## SQL Database Testing Project

Pruebas Automatizadas para Bases de Datos MySQL

**Versión:** 1.0

**Fecha:** 27 de November de 2025

**Autor:** Patricio

**Estado:** Aprobado

**Clasificación:** Interno

Basado en estándar IEEE 829  
Software Test Documentation

## Historial de Versiones

Versión	Fecha	Autor	Descripción
1.0	2025-11-27	Patricio	Versión inicial del plan de pruebas
1.1	2025-11-27	Patricio	Agregados tests de Sakila database

# Índice

1. Introducción .....	4
1.1 Propósito .....	4
1.2 Alcance .....	4
1.3 Referencias .....	4
1.4 Definiciones y Acrónimos .....	5
2. Elementos de Prueba .....	6
2.1 Base de Datos test_database .....	6
2.2 Base de Datos Sakila .....	7
3. Características a Probar .....	8
4. Características que No Se Probarán .....	9
5. Enfoque de Pruebas .....	10
5.1 Estrategia de Pruebas .....	10
5.2 Tipos de Pruebas .....	10
5.3 Técnicas de Diseño de Pruebas .....	11
6. Criterios de Aceptación .....	12
7. Criterios de Suspensión y Reanudación .....	13
8. Entregables de Prueba .....	14
9. Ambiente de Pruebas .....	15
10. Responsabilidades .....	16
11. Cronograma .....	17
12. Riesgos y Contingencias .....	18
13. Aprobaciones .....	19
14. Conclusiones .....	20
Anexo A: Resumen de Casos de Prueba .....	21
Anexo B: Métricas de Calidad .....	22

# 1. Introducción

## 1.1 Propósito

Este documento describe el plan de pruebas para el proyecto SQL Database Testing Project. El propósito principal es establecer el enfoque, estrategias, recursos y cronograma necesarios para verificar la correcta funcionalidad de las operaciones de base de datos MySQL, incluyendo operaciones CRUD, integridad de datos, validación de esquema y pruebas de rendimiento.

Este plan de pruebas está diseñado para garantizar que el framework de testing automatizado cumple con los requisitos de calidad establecidos y puede ser utilizado como herramienta confiable para la validación de bases de datos en entornos de desarrollo y producción.

## 1.2 Alcance

El alcance de este plan de pruebas incluye:

- Validación de operaciones CRUD (Create, Read, Update, Delete) en MySQL
- Verificación de integridad de datos y constraints (UNIQUE, NOT NULL, FK, ENUM)
- Validación de esquema de base de datos (tablas, columnas, tipos de datos)
- Pruebas de rendimiento para operaciones individuales y masivas
- Pruebas sobre la base de datos test\_database (creada para el proyecto)
- Pruebas sobre la base de datos Sakila (base de datos de ejemplo de MySQL)
- Automatización completa usando pytest con generación de reportes

## 1.3 Referencias

ID	Documento	Descripción
REF-001	IEEE 829-2008	Standard for Software Test Documentation
REF-002	ISO/IEC 25010	Systems and software Quality Requirements
REF-003	MySQL 8.0 Reference Manual	Documentación oficial de MySQL
REF-004	pytest Documentation	Framework de testing para Python
REF-005	Sakila Sample Database	Documentación de BD de ejemplo MySQL

## 1.4 Definiciones y Acrónimos

Término	Definición
<b>CRUD</b>	Create, Read, Update, Delete - Operaciones básicas de base de datos
<b>FK</b>	Foreign Key - Llave foránea para relaciones entre tablas
<b>PK</b>	Primary Key - Llave primaria identificadora única
<b>DDL</b>	Data Definition Language - Lenguaje de definición de datos
<b>DML</b>	Data Manipulation Language - Lenguaje de manipulación de datos
<b>pytest</b>	Framework de testing para Python
<b>Fixture</b>	Componente reutilizable para configuración de tests
<b>Marker</b>	Etiqueta para categorizar y filtrar tests
<b>CI/CD</b>	Continuous Integration / Continuous Deployment
<b>SUT</b>	System Under Test - Sistema bajo prueba

## 2. Elementos de Prueba

Los elementos bajo prueba en este proyecto son dos bases de datos MySQL con diferentes propósitos y características:

### 2.1 Base de Datos `test_database`

Base de datos creada específicamente para el proyecto, diseñada para validar operaciones CRUD completas y pruebas de integridad. Permite operaciones de escritura y modificación.

#### Estructura de Tablas:

**Tabla: users**

Columna	Tipo	Constraints	Descripción
<code>id</code>	INT	PK, AUTO_INCREMENT	Identificador único
<code>username</code>	VARCHAR(50)	UNIQUE, NOT NULL	Nombre de usuario
<code>email</code>	VARCHAR(100)	UNIQUE, NOT NULL	Correo electrónico
<code>password_hash</code>	VARCHAR(255)	NOT NULL	Hash de contraseña
<code>first_name</code>	VARCHAR(50)	NULL	Nombre
<code>last_name</code>	VARCHAR(50)	NULL	Apellido
<code>age</code>	INT	NULL	Edad
<code>is_active</code>	BOOLEAN	DEFAULT TRUE	Estado activo
<code>created_at</code>	TIMESTAMP	AUTO	Fecha creación
<code>updated_at</code>	TIMESTAMP	AUTO UPDATE	Fecha actualización

**Tabla: products**

Columna	Tipo	Constraints	Descripción
<code>id</code>	INT	PK, AUTO_INCREMENT	Identificador único
<code>name</code>	VARCHAR(100)	NOT NULL	Nombre del producto
<code>description</code>	TEXT	NULL	Descripción
<code>price</code>	DECIMAL(10,2)	NOT NULL	Precio
<code>stock</code>	INT	DEFAULT 0	Cantidad en stock
<code>category</code>	VARCHAR(50)	NULL	Categoría
<code>is_available</code>	BOOLEAN	DEFAULT TRUE	Disponibilidad
<code>created_at</code>	TIMESTAMP	AUTO	Fecha creación

**Tabla: orders**

Columna	Tipo	Constraints	Descripción
id	INT	PK, AUTO_INCREMENT	Identificador único
user_id	INT	FK → users(id) CASCADE	ID del usuario
product_id	INT	FK → products(id) CASCADE	ID del producto
quantity	INT	NOT NULL	Cantidad
total_price	DECIMAL(10,2)	NOT NULL	Precio total
status	ENUM	DEFAULT pending	Estado del pedido
order_date	TIMESTAMP	AUTO	Fecha del pedido

## 2.2 Base de Datos Sakila

Sakila es una base de datos de ejemplo proporcionada por MySQL que simula una tienda de alquiler de DVDs. Contiene un esquema completo con 16 tablas, 7 vistas, stored procedures, funciones y triggers. Es utilizada para validar operaciones de lectura y queries complejas.

Elemento	Cantidad	Descripción
Tablas	16	actor, film, customer, rental, payment, etc.
Vistas	7	customer_list, film_list, sales_by_category, etc.
Registros (actor)	200	Actores de películas
Registros (film)	1,000	Catálogo de películas
Registros (customer)	599	Clientes registrados
Registros (rental)	16,044	Historial de alquileres
Registros (payment)	16,049	Pagos realizados

### 3. Características a Probar

Las siguientes características serán probadas como parte de este plan:

ID	Característica	Descripción	Prioridad
F-001	Operaciones CREATE	INSERT simple y masivo de registros	Alta
F-002	Operaciones READ	SELECT con filtros, ordenamiento y límites	Alta
F-003	Operaciones UPDATE	Actualización simple y condicional	Alta
F-004	Operaciones DELETE	Eliminación simple y en cascada	Alta
F-005	Constraints UNIQUE	Validación de valores únicos	Alta
F-006	Constraints NOT NULL	Validación de campos requeridos	Alta
F-007	Constraints FK	Validación de llaves foráneas	Alta
F-008	Constraints ENUM	Validación de valores enumerados	Alta
F-009	Tipos de Datos	Validación de INT, VARCHAR, DECIMAL, etc.	Media
F-010	Timestamps	Generación automática de fechas	Media
F-011	Valores DEFAULT	Aplicación de valores por defecto	Media
F-012	CASCADE DELETE	Eliminación en cascada de registros hijos	Alta
F-013	JOINS	Consultas con múltiples tablas	Alta
F-014	Agregaciones	COUNT, SUM, AVG, MIN, MAX	Alta
F-015	Subqueries	Subconsultas y queries anidadas	Media
F-016	Vistas	Consulta de vistas predefinidas	Media
F-017	Performance	Tiempos de ejecución aceptables	Media
F-018	Conexión	Establecimiento y cierre de conexión	Alta

## 4. Características que No Se Probarán

Las siguientes características están fuera del alcance de este plan de pruebas:

- Stored Procedures y Functions de Sakila (solo se valida su existencia)
- Triggers de Sakila (solo se valida comportamiento indirecto)
- Pruebas de seguridad (SQL Injection, autenticación)
- Pruebas de carga con múltiples usuarios concurrentes
- Pruebas de recuperación ante desastres
- Pruebas de migración de datos
- Pruebas de replicación de base de datos
- Pruebas en bases de datos diferentes a MySQL
- Interfaz gráfica de usuario (no aplica)
- APIs REST o servicios web

**Justificación:** El alcance se limita a validar el framework de testing y las operaciones fundamentales de base de datos. Las características excluidas requieren herramientas especializadas y están fuera del objetivo principal del proyecto.

## 5. Enfoque de Pruebas

### 5.1 Estrategia de Pruebas

La estrategia de pruebas se basa en automatización completa utilizando pytest como framework principal. Se implementa un enfoque de pruebas en capas:

- **Pruebas Unitarias de BD:** Validación de operaciones individuales CRUD
- **Pruebas de Integración:** Validación de relaciones entre tablas y constraints
- **Pruebas de Schema:** Verificación de estructura de base de datos
- **Pruebas de Datos:** Validación de integridad y consistencia de datos
- **Pruebas de Performance:** Medición de tiempos de ejecución

### 5.2 Tipos de Pruebas

Tipo	Descripción	Cantidad	Archivos
CRUD	Operaciones básicas de BD	25	test_crud_operations.py
Integridad	Constraints y tipos de datos	26	test_data_integrity.py
Performance	Tiempos de ejecución	14	test_performance.py
Schema Sakila	Estructura de BD Sakila	35	test_sakila_schema.py
Datos Sakila	Validación de datos Sakila	22	test_sakila_data.py
Queries Sakila	Queries complejas	21	test_sakila_queries.py
Perf. Sakila	Performance en Sakila	13	test_sakila_performance.py
<b>TOTAL</b>		<b>156</b>	<b>7 archivos</b>

## 5.3 Técnicas de Diseño de Pruebas

### Partición de Equivalencia

Se identifican clases de equivalencia para cada campo de entrada:

Campo	Clases Válidas	Clases Inválidas
username	Alfanumérico 1-50 chars	NULL, vacío, >50 chars, duplicado
email	formato@dominio.ext	NULL, sin @, sin dominio, duplicado
price	Decimal positivo 0.00-99999999.99	NULL, negativo, overflow
quantity	Entero positivo $\geq 1$	NULL, 0, negativo
status	pending, processing, shipped, delivered, cancelled	otro string

### Análisis de Valores Límite

Se prueban los valores en los límites de cada rango:

Campo	Límite Inferior	Límite Superior	Fuera de Rango
username (VARCHAR 50)	1 char	50 chars	51 chars
age (INT)	0	2147483647	-1, 2147483648
price (DECIMAL 10,2)	0.00	99999999.99	100000000.00
stock (INT)	0	2147483647	-1

## 6. Criterios de Aceptación

Las pruebas se considerarán exitosas cuando se cumplan los siguientes criterios:

### Criterios de Entrada (Entry Criteria)

- ✓ MySQL Server 8.0+ instalado y en ejecución
- ✓ Base de datos test\_database creada
- ✓ Base de datos Sakila instalada con datos completos
- ✓ Python 3.10+ con todas las dependencias instaladas
- ✓ Archivo .env configurado con credenciales válidas
- ✓ Conexión a base de datos verificada

### Criterios de Salida (Exit Criteria)

- ✓ 100% de los casos de prueba ejecutados
- ✓ 95% o más de los casos de prueba pasados
- ✓ 0 defectos de severidad crítica o alta sin resolver
- ✓ Todos los tests de regresión pasados
- ✓ Documentación de pruebas completa
- ✓ Reporte de ejecución generado

### Criterios de Aceptación por Tipo de Prueba

Tipo de Prueba	Criterio de Aceptación	Métrica
CRUD	Todas las operaciones ejecutan sin error	100% pass
Integridad	Constraints funcionan correctamente	100% pass
Performance	Tiempos dentro de umbrales definidos	100% pass
Schema	Estructura coincide con especificación	100% pass
Datos	Datos consistentes e íntegros	100% pass

## 7. Criterios de Suspensión y Reanudación

### Criterios de Suspensión

La ejecución de pruebas será suspendida si ocurre alguna de las siguientes condiciones:

- Servidor MySQL no disponible o caído
- Más del 30% de los casos de prueba fallan en una ejecución
- Defecto crítico que bloquea la ejecución de otros tests
- Corrupción de datos en las bases de datos de prueba
- Falla en la conexión a base de datos persistente
- Ambiente de pruebas comprometido

### Criterios de Reanudación

La ejecución de pruebas será reanudada cuando:

- ✓ El servidor MySQL está operativo y estable
- ✓ Los defectos bloqueantes han sido resueltos
- ✓ El ambiente de pruebas ha sido restaurado
- ✓ Las bases de datos de prueba han sido reinicializadas
- ✓ La conexión a base de datos es estable
- ✓ Se ha verificado la integridad del ambiente

## 8. Entregables de Prueba

Los siguientes entregables serán producidos como resultado del proceso de pruebas:

Entregable	Descripción	Formato
Plan de Pruebas	Este documento	PDF
Casos de Prueba	Detalle de todos los casos de prueba	CSV/Google Sheets
Scripts de Prueba	Código automatizado de pruebas	Python (.py)
Reporte de Ejecución	Resultados de ejecución de pytest	HTML
Clases de Equivalencia	Análisis de particiones	CSV/Google Sheets
Valores Límite	Análisis de boundaries	CSV/Google Sheets
Checklist de Verificación	Lista de verificación	CSV/Google Sheets
Matriz de Trazabilidad	Mapeo requisitos-tests	CSV/Google Sheets
Código Fuente	Repositorio completo	GitHub
README	Documentación del proyecto	Markdown

## 9. Ambiente de Pruebas

### Requisitos de Hardware

Componente	Requisito Mínimo	Recomendado
Procesador	Intel Core i3 / Apple M1	Intel Core i5+ / Apple M1+
Memoria RAM	4 GB	8 GB+
Almacenamiento	1 GB libre	5 GB+ libre
Conectividad	Localhost	Localhost

### Requisitos de Software

Software	Versión	Propósito
Sistema Operativo	macOS / Linux / Windows	Plataforma base
Python	3.10+	Lenguaje de programación
MySQL Server	8.0+	Motor de base de datos
pytest	Última versión	Framework de testing
mysql-connector-python	Última versión	Conector MySQL
Faker	Última versión	Generación de datos
PyCharm	2023+	IDE (recomendado)

### Configuración del Ambiente

- Archivo .env con credenciales de MySQL configuradas
- Base de datos test\_database creada y accesible
- Base de datos Sakila instalada con schema y datos
- Entorno virtual de Python con dependencias instaladas
- pytest configurado como test runner en IDE

## 10. Responsabilidades

Rol	Responsabilidades
<b>QA Engineer</b>	<ul style="list-style-type: none"><li>• Diseño y desarrollo de casos de prueba</li><li>• Implementación de scripts automatizados</li><li>• Ejecución de pruebas</li><li>• Reporte de defectos</li><li>• Generación de reportes</li></ul>
<b>Developer</b>	<ul style="list-style-type: none"><li>• Corrección de defectos reportados</li><li>• Revisión de código de pruebas</li><li>• Soporte técnico para ambiente</li></ul>
<b>Tech Lead</b>	<ul style="list-style-type: none"><li>• Revisión y aprobación del plan de pruebas</li><li>• Asignación de recursos</li><li>• Toma de decisiones sobre criterios</li></ul>
<b>DBA</b>	<ul style="list-style-type: none"><li>• Configuración de bases de datos</li><li>• Mantenimiento del ambiente</li><li>• Respaldo y restauración de datos</li></ul>

## 11. Cronograma

Fase	Actividades	Duración	Estado
Planificación	Diseño del plan de pruebas, definición de alcance	2 días	Completado ✓
Diseño	Diseño de casos de prueba, clases de equivalencia	3 días	Completado ✓
Implementación	Desarrollo de scripts automatizados	5 días	Completado ✓
Configuración	Setup de ambiente, bases de datos	1 día	Completado ✓
Ejecución	Ejecución de suite de pruebas	1 día	Completado ✓
Reporte	Generación de reportes y documentación	1 día	Completado ✓
<b>Total</b>		<b>13 días</b>	

## 12. Riesgos y Contingencias

ID	Riesgo	Probabilidad	Impacto	Mitigación
R-001	MySQL Server no disponible	Baja	Alto	Verificar servicio antes de ejecutar
R-002	Datos de Sakila incompletos	Media	Alto	Verificar COUNTs antes de pruebas
R-003	Credenciales incorrectas	Baja	Medio	Validar conexión en setup
R-004	Timeout en pruebas de performance	Media	Bajo	Ajustar umbrales si es necesario
R-005	Conflictos de versión de dependencias	Baja	Medio	Usar versiones flexibles
R-006	Espacio en disco insuficiente	Baja	Medio	Monitorear espacio disponible
R-007	Cambios en schema de Sakila	Muy Baja	Alto	Usar versión específica de Sakila

## 13. Aprobaciones

Este plan de pruebas ha sido revisado y aprobado por las siguientes personas:

Rol	Nombre	Firma	Fecha
QA Engineer	Patricio	<hr/>	2025-11-27
Tech Lead	<hr/>	<hr/>	<hr/>
Project Manager	<hr/>	<hr/>	<hr/>

## 14. Conclusiones

Este plan de pruebas establece un framework completo y profesional para la validación de operaciones de base de datos MySQL. Los principales logros y conclusiones son:

- Cobertura Completa:** Se han diseñado 156 casos de prueba que cubren operaciones CRUD, integridad de datos, validación de schema, queries complejas y pruebas de rendimiento.
- Automatización Total:** Todas las pruebas están automatizadas usando pytest, permitiendo ejecución repetible y consistente.
- Dos Bases de Datos:** El proyecto valida tanto una base de datos personalizada (test\_database) como una base de datos de referencia (Sakila).
- Estándares Profesionales:** El plan sigue el estándar IEEE 829 para documentación de pruebas de software.
- Documentación Completa:** Se incluyen casos de prueba detallados, clases de equivalencia, valores límite, checklist y matriz de trazabilidad.
- Mantenibilidad:** El código está organizado siguiendo patrones de diseño (Page Object Model adaptado a BD) y mejores prácticas.

## Resultados de Ejecución

Métrica	Valor
Total de Tests	156
Tests Pasados	156 (100%)
Tests Fallidos	0 (0%)
Tiempo de Ejecución	< 15 segundos
Cobertura de Requisitos	100%

El proyecto cumple con todos los criterios de aceptación establecidos y está listo para ser utilizado como framework de referencia para pruebas de bases de datos MySQL.

## Anexo A: Resumen de Casos de Prueba

Archivo	Clase de Test	Cantidad	Categoría
test_crud_operations.py	TestCreateOperations	7	CRUD
test_crud_operations.py	TestReadOperations	8	CRUD
test_crud_operations.py	TestUpdateOperations	5	CRUD
test_crud_operations.py	TestDeleteOperations	5	CRUD
test_data_integrity.py	TestSchemaIntegrity	6	Integridad
test_data_integrity.py	TestConstraints	8	Integridad
test_data_integrity.py	TestDataTypes	5	Integridad
test_data_integrity.py	TestReferentialIntegrity	4	Integridad
test_data_integrity.py	TestDataConsistency	3	Integridad
test_performance.py	TestQueryPerformance	5	Performance
test_performance.py	TestBulkOperationPerformance	4	Performance
test_performance.py	TestStressTests	3	Performance
test_performance.py	TestConnectionPerformance	2	Performance
test_sakila_schema.py	TestSakilaSchemaExists	23	Schema
test_sakila_schema.py	TestSakilaTableColumns	5	Schema
test_sakila_schema.py	TestSakilaConstraints	4	Schema
test_sakila_data.py	TestSakilaRecordCounts	9	Datos
test_sakila_data.py	TestSakilaDataValues	7	Datos
test_sakila_data.py	TestSakilaDataIntegrity	6	Datos
test_sakila_queries.py	TestSakilaBasicQueries	5	Queries
test_sakila_queries.py	TestSakilaJoinQueries	5	Queries
test_sakila_queries.py	TestSakilaAggregationQueries	8	Queries
test_sakila_queries.py	TestSakilaSubqueries	3	Queries
test_sakila_performance.py	TestSakilaQueryPerformance	10	Performance
test_sakila_performance.py	TestSakilaViewPerformance	3	Performance

## Anexo B: Métricas de Calidad

### Métricas de Cobertura

Métrica	Valor	Objetivo	Estado
Cobertura de Requisitos	100%	100%	✓ Cumple
Cobertura de Operaciones CRUD	100%	100%	✓ Cumple
Cobertura de Constraints	100%	100%	✓ Cumple
Cobertura de Tipos de Datos	100%	95%	✓ Cumple
Cobertura de Tablas Sakila	100%	100%	✓ Cumple
Cobertura de Vistas Sakila	100%	100%	✓ Cumple

### Métricas de Ejecución

Métrica	Valor
Tests Totales	156
Tests Pasados	156
Tests Fallidos	0
Tests Omitidos	0
Tasa de Éxito	100%
Tiempo Promedio por Test	~0.1 segundos
Tiempo Total de Suite	< 15 segundos