

# Important Requirements & Notes

## Minimum MySQL Version

- Required: **MySQL 8.0 or higher**
- Recommended: **MySQL 8.4 LTS** (long-term support as of 2026)

This schema is built for modern MySQL features including:

- Native JSON columns for flexible data
- CHECK constraints
- Generated columns
- Partitioning by tenant\_id for scale

## Key Features & Best Practices

- **Multi-tenant isolation** via tenant\_id in all relevant tables
- **Row-level security** recommended: Implement via views or application-level checks (e.g., WHERE tenant\_id = SESSION\_TENANT\_ID())
- **Soft deletes** using deleted\_at TIMESTAMP column
- **Timestamps** (created\_at, updated\_at) with automatic updates
- **Compliance-ready** fields for GDPR/LGPD (consent, erase requests, retention period)
- **Audit trail** in audit\_logs table (partitioned for performance)
- **RBAC** fully implemented with roles, permissions, and many-to-many mappings

## Quick Start

1. Execute the full CREATE TABLE script in the exact order provided (dependencies are respected)
2. Run the seed data (optional, for testing)
3. Customize for your niche:
  - Add industry-specific tables
  - Extend existing tables via ALTER TABLE
  - Implement row-level security in your application

## Migration & Versioning Tips

Use the provided ALTER TABLE examples in the "Migration Hints" section at the end of the schema script. Always test migrations in staging before production.

## Security Recommendations

- Never store plain-text sensitive data (e.g., passwords, PII, health info)
- Use app-level encryption for sensitive fields (e.g., AES\_ENCRYPT for backups)
- Hash passwords with bcrypt/argon2
- Implement consent checks before processing personal data

## Support & Customization

This schema is designed to be white-label and reusable.

Need a custom niche extension, no-JSON fallback, or help with migrations?

Contact us or open an issue.

## Core Modules

### Tenants Module

- **tenants**: Stores information about each tenant in the multi-tenant system, including name and domain for white-label support. Includes compliance fields for data retention and erasure.
- **branding\_settings**: Holds customizable branding details per tenant, such as logos and color schemes stored in JSON for flexibility.

### Users Module

- **users**: Manages user accounts with multi-tenant isolation, authentication details, and GDPR/LGPD compliance fields like consent and erasure requests.
- **user\_roles**: Associates users with roles for RBAC (Role-Based Access Control).

### Roles Module

- **roles**: Defines roles that can be system-wide (tenant\_id NULL) or tenant-specific, with names unique per tenant.
- **permissions**: Lists atomic permissions like 'read\_users' that can be assigned to roles.
- **role\_permissions**: Many-to-many mapping between roles and permissions.

### Plans Module

- **plans**: Describes available subscription plans with features stored in JSON for extensibility.

### Subscriptions Module

- **subscriptions**: Tracks subscriptions for tenants, including lifecycle status (e.g., 'active', 'cancelled') and billing details.

### Audit Logs Module

- **audit\_logs**: Records all changes for auditing and compliance, with changes stored in JSON. Partitioned by tenant\_id for performance at scale.

## Startup Niche Modules (Legal & Compliance)

### Clients Module

- **clients**: Represents clients or parties involved in legal matters, with contact details (encryption hint: encrypt email/phone in app) and consent flags.

### Matters Module

- **matters**: Legal cases or matters with type, status (e.g., 'open', 'closed'), and details in JSON.

## Documents Module

- **documents:** Legal documents linked to matters, with file metadata in JSON and approval status.

## Tasks Module

- **compliance\_tasks:** Compliance activities or audits with due dates and completion status.

## Brief Explanation Per Table

- **tenants:** Core table for multi-tenant isolation. Each tenant has a unique domain for white-labeling. Includes soft delete and timestamps. Compliance: data\_retention\_period for GDPR/LGPD.
- **branding\_settings:** Extends tenants with JSON-based flexible settings for colors, fonts, etc., enabling easy white-label customization without schema changes.
- **users:** User accounts with hashed passwords (encryption hint: use bcrypt/argon2). Multi-tenant via tenant\_id. Compliance: consent\_given\_at, erase\_requested\_at for GDPR Art. 17. Unique email per tenant.
- **roles:** Roles for RBAC, supporting both global and per-tenant custom roles. Unique name per tenant.
- **permissions:** Granular permissions for actions across the system.
- **user\_roles:** Links users to multiple roles.
- **role\_permissions:** Assigns permissions to roles.
- **plans:** Subscription plans with JSON features for flexibility. CHECK constraint ensures positive pricing.
- **subscriptions:** Manages tenant subscriptions with lifecycle status as VARCHAR. Includes billing dates for operational use.
- **audit\_logs:** Logs actions for security and compliance. JSON for changes allows flexible auditing. Partitioned for scalability.
- **clients:** Client records with type (e.g., 'individual', 'corporate') and marketing consent for compliance.
- **matters:** Legal matters with assigned users, priority, and custom fields in JSON.
- **documents:** Documents attached to matters, with version control via JSON and encryption hints for sensitive content.
- **compliance\_tasks:** Tasks for regulatory compliance, linked to matters or standalone, with reminders.

## Relationship Summary

- tenants 1:M branding\_settings (tenant\_id FK)
- tenants 1:M users (tenant\_id FK)
- tenants 1:M roles (tenant\_id FK, nullable for system roles)
- tenants 1:M subscriptions (tenant\_id FK)
- tenants 1:M audit\_logs (tenant\_id FK)
- tenants 1:M clients (tenant\_id FK)
- tenants 1:M matters (tenant\_id FK)
- tenants 1:M documents (tenant\_id FK)

- tenants 1:M compliance\_tasks (tenant\_id FK)
- users 1:M user\_roles (user\_id FK)
- users 1:M audit\_logs (user\_id FK, nullable)
- users 1:M matters (assigned\_user\_id FK)
- users 1:M compliance\_tasks (assigned\_user\_id FK)
- roles 1:M user\_roles (role\_id FK)
- roles 1:M role\_permissions (role\_id FK)
- permissions 1:M role\_permissions (permission\_id FK)
- plans 1:M subscriptions (plan\_id FK)
- clients 1:M matters (client\_id FK)
- matters 1:M documents (matter\_id FK)
- matters 1:M compliance\_tasks (matter\_id FK, nullable)
- All relationships enforced with FOREIGN KEY constraints, ON DELETE CASCADE/RESTRICT/SET NULL as appropriate. No circular dependencies. Multi-tenant: tenant\_id in all niche tables for row-level security (via views). Audit logs reference users and tenants.

## Important Requirements for JSON Columns

This schema uses **native JSON columns** (e.g., `vitals` JSON, `data` JSON, `goals` JSON, `specs` JSON, etc.), which are fully supported and performant in **MySQL 8.0+** (strongly recommended: MySQL 8.4 LTS as of 2026).

### Minimum MySQL Version

- Required: **MySQL 8.0 or higher**
- Recommended: **MySQL 8.4 LTS** (long-term support in 2026)

If you see errors like:

- This version of MySQL doesn't yet support 'JSON' column type
- Invalid JSON text in argument 1 to function json\_valid
- JSON documents may not contain data outside of the document

It is almost always caused by:

1. Using MySQL < 8.0 (upgrade required)
2. Inserting invalid JSON strings (missing quotes, malformed syntax, etc.)

### How to Validate JSON Before Inserting

Always ensure the value is valid JSON. You can test with:

```
SELECT JSON_VALID('{"height": 170, "weight": 70}');      -- Should return 1 (valid)
SELECT JSON_VALID('invalid json here');                 -- Returns 0 (invalid)
```