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 (Construction Project Management)

Projects Module

- **projects**: Core construction projects with timelines, budgets, and status (e.g., 'planning', 'completed').

Tasks Module

- **tasks**: Project tasks with dependencies and assignments.

Resources Module

- **resources**: Materials and equipment inventory with quantity tracking.

Change Orders Module

- **change_orders**: Modifications to projects with approval 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.
- **projects**: Construction projects with budget (CHECK positive), timeline, and custom specs in JSON.
- **tasks**: Tasks within projects, with priority, due dates, and status as VARCHAR.
- **resources**: Inventory for construction materials/equipment, with low stock checks.
- **change_orders**: Project change requests with impact on budget/time and approval workflow.

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 projects (tenant_id FK)
- tenants 1:M tasks (tenant_id FK)
- tenants 1:M resources (tenant_id FK)
- tenants 1:M change_orders (tenant_id FK)
- users 1:M user_roles (user_id FK)
- users 1:M audit_logs (user_id FK, nullable)
- users 1:M projects (assigned_user_id FK)
- users 1:M 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)
- projects 1:M tasks (project_id FK)
- projects 1:M change_orders (project_id FK)
- resources 1:M tasks (resource_id FK, nullable for allocation)
- 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)
```