ERP SaaS Database Design Roadmap Phase ۲

Version ۲٫۰

Status: Architectural Decision Document

State: Database Planning Reference

Project: HamarehERP SaaS Platform

---

۱. Purpose

این سند مسیر رسمی طراحی پایگاه داده HamarehERP را مشخص می‌کند.

هدف:

- تعیین مرزهای دیتابیس
- تعیین مالکیت جداول
- جلوگیری از طراحی اشتباه Schema
- هماهنگ‌سازی Database با معماری Bounded Context
- آماده‌سازی برای Migration در Laravel
- حفظ قابلیت توسعه آینده

اصل پایه:

Databaseباید بازتاب معماری سیستم باشد، نه محدودیت طراحی آن.

---

۲. Database Architecture Model

معماری دیتابیس از دو بخش اصلی تشکیل می‌شود:

Database A

hamareh_saas_core

مالک:

Layer ۱

SaaS Platform

Layer ۲

Identity & Security Core

==========

Database B

hamareh_erp_tenants

مالک:

Layer ۳

ERP Foundation

Layer ۴

ERP Business Modules

---

۳. Database Ownership Rules

Rule ۱ - Single Table Ownership

هر جدول فقط یک Owner دارد.

مثال صحیح:

sales_orders

Owner:

Sales Module

مثال غلط:

sales_orders

Owner:

Sales + Inventory

Rule ۲ - Logical Database Separation

حتی اگر تمام جدول‌ها در یک PostgreSQL قرار داشته باشند:

هر Module باید Schema منطقی مستقل داشته باشد.

نمونه:

hamareh_erp_tenants

├── organization

├── master_data

├── accounting

├── inventory

├── sales

├── purchase

├── manufacturing

└── hr

Rule ۳ - Physical Foreign Key Policy

داخل یکModule

Physical FKمجاز است.

مثال:

accounting.journal

|

FK

accounting.journal_items

بینModule ها

Physical FKممنوع است.

مثال ممنوع:

sales.orders

FK

inventory.stock

ارتباط:

- UUID Reference
- API
- Event

---

۴. Database Layer Mapping

Layer ۱ - SaaS Platform Database

Database:

hamareh_saas_core

Schema:

saas

Tables:

Tenant Management

مالک:

tenants

tenant_domains

tenant_settings

tenant_status_history

Subscription

مالک:

plans

plan_features

subscriptions

subscription_history

invoices

billing_transactions

Platform Administration

مالک:

platform_users

system_settings

platform_audit_logs

---

۵. Layer ۲ - Identity Core Database

Database:

hamareh_saas_core

Schema:

identity

Tables:

users

credentials

profiles

roles

permissions

scopes

user_roles

user_permissions

tenant_memberships

---

۶. Layer ۳ - ERP Foundation Database

Database:

hamareh_erp_tenants

Schema:

foundation

Organization Module

Tables:

companies

branches

departments

organization_units

تمام جداول:

اجباری:

tenant_id

Master Data Module

Schema:

master_data

Tables:

currencies

units

countries

languages

tax_definitions

measurement_units

Workflow Module

Schema:

workflow

Tables:

workflow_definitions

workflow_steps

workflow_instances

workflow_approvals

workflow_history

Document Module

Schema:

documents

Tables:

documents

attachments

file_metadata

document_versions

---

۷. Layer ۴ - Business Database

Database:

hamareh_erp_tenants

هر Module دارای Schema مستقل:

accounting

inventory

sales

purchase

manufacturing

hr

projects

---

۸. Accounting Database Boundary

Schema:

accounting

مالک:

chart_of_accounts

accounts

journals

journal_entries

journal_lines

financial_periods

cost_centers

ارتباط با:

Sales

Purchase

Inventory

فقط از طریق:

- API

- Event

---

۹. Inventory Database Boundary

Schema:

inventory

مالک:

warehouses

locations

items

stock_balances

inventory_movements

stock_adjustments

---

۱۰. Sales Database Boundary

Schema:

sales

مالک:

customers

leads

opportunities

sales_orders

sales_order_items

sales_status_history

---

۱۱. Purchase Database Boundary

Schema:

purchase

مالک:

suppliers

purchase_requests

purchase_orders

purchase_order_items

purchase_receipts

---

۱۲. Manufacturing Database Boundary

Schema:

manufacturing

مالک:

bill_of_materials

production_orders

work_orders

production_transactions

---

۱۳. HR Database Boundary

Schema:

hr

مالک:

employees

attendance

payroll

hr_documents

---

۱۴. Tenant Isolation Database Rules

تمام جدول‌های Layer ۳ و Layer ۴:

الزامی:

tenant_id UUID NOT NULL

همراه با:

PostgreSQL Row Level Security

+

Application Tenant Context

---

۱۵. Standard Audit Columns

تمام جدول‌های عملیاتی:

حداقل:

id UUID

tenant_id UUID

created_at

updated_at

created_by

updated_by

در جداول حساس:

deleted_at

version

---

۱۶. Primary Key Strategy

تمامEntity های:Business

استفاده از:

UUID Primary Key

دلایل:

- SaaS Multi Tenant

- Distributed Future
- Microservice Migration

---

۱۷. Database Design Sequence

Phase ۱

Foundation Database

ایجاد:

۱. Tenant Context
۲. Organization
۳. Identity Integration
۴. Master Data
۵. Audit Structure

Phase ۲

First Business Module

Accounting:

- Chart Of Account
- Journal
- Ledger

Phase ۳

Inventory:

- Warehouse
- Stock
- Movement

Phase ۴

Operational Modules:

- Sales
- Purchase
- Manufacturing

---

۱۸. Final Database Principle

معماری نهایی:

Architecture

↓

Bounded Context

↓

Database Ownership

↓

Schema Design

↓

Tables

↓

Migration

هیچ جدول جدیدی بدون مشخص شدن:

- Owner
- Context
- Dependency
- Tenant Impact

ایجاد نخواهد شد.