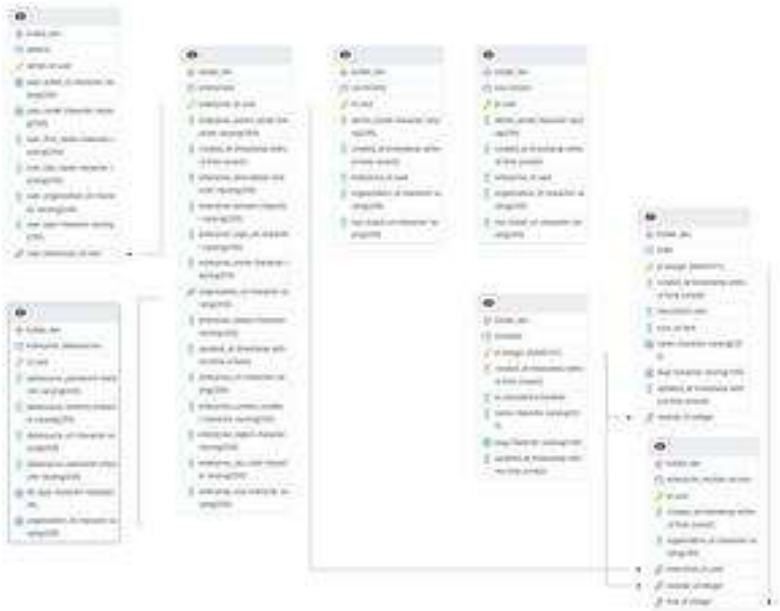
Schemas

Kolate Core Schema



Kolate Platform Database Documentation

Overview

The **Kolate platform database schema** (schema name: `kolate_dev`) manages enterprise onboarding, SSO integration, module access control, and tenant-specific configurations.
The schema is designed with multi-tenant principles — allowing each enterprise to have its own organization, authentication setup, and data provisioning through isolated configurations.

---

1. Table: `admins`

Purpose

Stores Kolate platform administrator details. These users manage enterprise onboarding, SSO configuration, and platform-level operations.

Columns

Column	Type	Description
id	uuid	Unique identifier for each admin record
user_auth0_id	varchar(255)	Auth0 user ID for authentication and identity management
user_email	varchar(255)	Admin's email address
user_first_name	varchar(255)	Admin's first name
user_last_name	varchar(255)	Admin's last name
user_organization_id	varchar(255)	Organization identifier (if applicable)
user_enterprise_id	uuid	Associated enterprise ID (links to enterprises.enterprise_id)

---

2. Table: `enterprises`

Purpose

Represents enterprise tenants onboarded to the Kolate platform.
Each enterprise corresponds to an organization created in Auth0 and may have separate configurations and data sources.

Columns

Column	Type	Description
enterprise_id	uuid	Primary key; unique enterprise identifier
enterprise_admin_email	varchar(255)	Admin email for the enterprise (Auth0 organization admin)
created_at	timestamp	Record creation timestamp
enterprise_description	varchar(255)	Short description of the enterprise
enterprise_logo_url	varchar(255)	Enterprise logo image URL
enterprise_name	varchar(255)	Name of the enterprise
organization_id	varchar(255)	Auth0 organization ID linked to the enterprise
enterprise_status	varchar(255)	Status (e.g., active, pending, suspended)
updated_at	timestamp	Last update timestamp
enterprise_url	varchar(255)	Enterprise-specific Kolate access URL

enterprise_contact_number	varchar(255)	Contact number
enterprise_region	varchar(255)	Geographic region
enterprise_zip_code	varchar(255)	Postal code
enterprise_size_character	varchar(255)	Size of the enterprise (e.g., small, medium, large)

---

3. Table: `sso_tickets`

Purpose

Stores SSO setup tickets that link enterprise onboarding to Auth0 SSO configuration.

Columns

Column	Type	Description
id	uuid	Unique SSO ticket identifier
admin_email	varchar(255)	Email of the Kolate admin who generated the SSO link
created_at	timestamp	Creation timestamp
enterprise_id	uuid	Linked enterprise identifier
organization_id	varchar(255)	Corresponding Auth0 organization ID
sso_ticket_url	varchar(255)	Auth0 SSO self-setup link sent to enterprise admin

Relationships

- Links to `enterprises.enterprise_id`
- Used during the onboarding workflow (Invitation + SSO Setup phases)

---

4. Table: `enterprise_datasources`

Purpose

Maintains configuration details for enterprise-specific databases or data connections.
Used during provisioning after successful SSO setup.

Columns

Column	Type	Description
id	uuid	Unique identifier
datasource_url	varchar(255)	Connection URL of the data source
datasource_schema	varchar(255)	Schema or database name
datasource_username	varchar(255)	Username for connecting to the data source
datasource_password	varchar(255)	Encrypted password
db_type	varchar(255)	Type of database (e.g., Postgres, Neo4j, Redis, etc.)
organization_id	varchar(255)	Auth0 organization ID associated with the enterprise

Relationships

- Indirectly linked to `enterprises` via `organization_id`.

---

5. Table: `modules`

Purpose

Defines functional modules available within the Kolate platform.
Modules represent key product components or feature sets.

Columns

Column	Type	Description
id	integer	Unique module identifier
created_at	timestamp	Creation time
updated_at	timestamp	Last modified time
name	varchar(255)	Module name
slug	varchar(100)	Unique short identifier used in URLs or APIs
is_standalone	boolean	Indicates if module operates independently of others

---

6. Table: `trials`

Purpose

Stores details of trial configurations or demo modules available for enterprises during onboarding or testing phases.

Columns

Column	Type	Description
id	integer	Unique trial ID
created_at	timestamp	Creation timestamp
updated_at	timestamp	Last modified timestamp
description	text	Description of the trial
icon_url	text	Icon representing the trial
name	varchar(255)	Trial name
slug	varchar(100)	Slug identifier for internal references
module_id	integer	Linked module identifier

Relationships

- module_id → references modules.id

---

7. Table: enterprise_module_access

Purpose

Maps which enterprise has access to which module and trial.
Defines access permissions and associations for tenants.

Columns

Column	Type	Description
id	uuid	Unique access record identifier
created_at	timestamp	Creation timestamp
organization_id	varchar(255)	Organization identifier
enterprise_id	uuid	Linked enterprise identifier
module_id	integer	Linked module

trial_id	integer	Linked trial (optional)

Relationships

- enterprise_id → enterprises.enterprise_id
- module_id → modules.id
- trial_id → trials.id

---

8. Table Relationships (Summary)

From Table	Relationship	To Table	Type
admins.user_enterprise_id	→	enterprises.enterprise_id	One-to-Many
sso_tickets.enterprise_id	→	enterprises.enterprise_id	One-to-One
enterprise_datasources.organization_id	→	enterprises.organization_id	One-to-One
enterprise_module_access.enterprise_id	→	enterprises.enterprise_id	One-to-Many
enterprise_module_access.module_id	→	modules.id	Many-to-One
enterprise_module_access.trial_id	→	trials.id	Many-to-One
trials.module_id	→	modules.id	Many-to-One

---

9. Data Flow Alignment with Onboarding Process

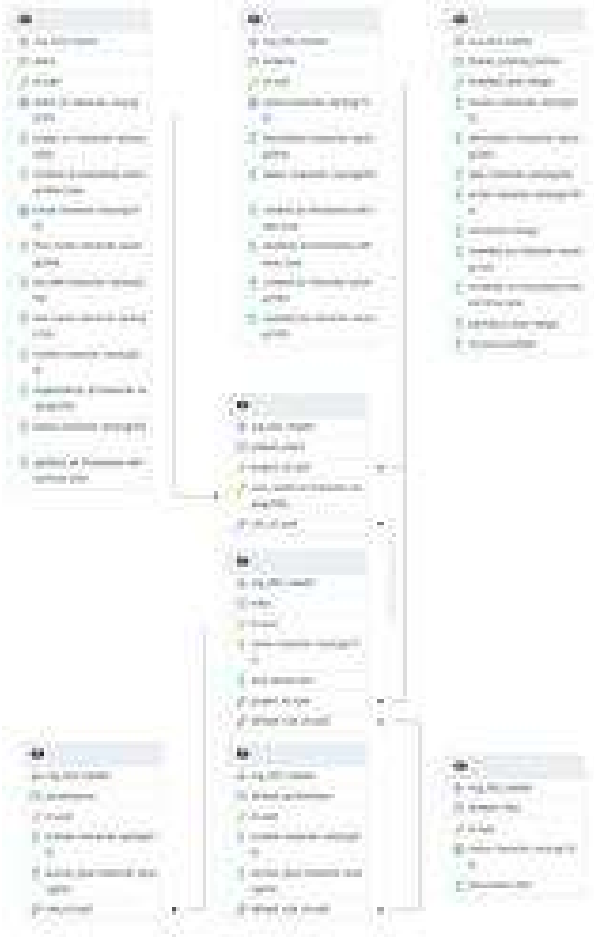
Onboarding Phase	Key Tables Used	Description
Invitation Phase	`admins`, `sso_tickets`, `enterprises`	Admin triggers organization creation and SSO ticket generation
SSO Setup Phase	`enterprises`, `sso_tickets`	Enterprise admin configures SSO; details stored for validation
Provisioning Phase	`enterprise_datasources`, `enterprise_module_access`	Database and module access are provisioned per enterprise
Completion Phase	`enterprises`, `enterprise_module_access`	Enterprise marked active; users gain platform access

---

10. Design Notes

- Schema supports **multi-tenant** architecture.
- All timestamps use **UTC with timezone**.
- Enterprise data isolation is enforced via:
  - Unique `enterprise_id` across all linked entities.
  - Dedicated datasource entries per tenant.
- Designed for **Auth0 SSO integration** and **modular access control**.

Tenant Schema



tenant_core_schema_erd.pgerd

Kolate Enterprise Tenant Database Documentation

Schema Name: `org_003_master`

(Each enterprise/tenant will have its own schema, e.g., `org_001_master`, `org_002_master`, etc.)

Overview

This schema contains **organization-specific data** for enterprise users, roles, projects, and permissions.
It is created and maintained per enterprise during the **provisioning phase** after SSO setup.
While `kolate_dev` manages global onboarding and configuration, this schema stores **tenant-level operational data** — such as projects, users, roles, and access control.

---

1. Table: `users`

Purpose

Stores user profile and identity information for all users belonging to this enterprise organization.

Columns

Column	Type	Description
`id`	`uuid`	Primary key; unique identifier for each user
`user_auth0_id`	`varchar(255)`	User ID mapped from Auth0 for authentication
`email`	`varchar(255)`	User's email address
`first_name`	`varchar(255)`	User's first name
`last_name`	`varchar(255)`	User's last name
`job_title`	`varchar(255)`	Job title or designation
`mobile`	`varchar(20)`	Mobile number
`organization_id`	`varchar(255)`	Organization identifier; links back to global `kolate_dev.enterprises.organization_id`
`status`	`varchar(50)`	Account status (e.g., active, pending, disabled)
`created_at`	`timestamp`	Account creation timestamp
`updated_at`	`timestamp`	Last update timestamp

---

2. Table: `projects`

Purpose

Represents projects managed under the enterprise's workspace.
Each project can have assigned users and associated metadata.

Columns

Column	Type	Description
`id`	`uuid`	Unique identifier for each project
`name`	`varchar(100)`	Project name
`description`	`varchar(500)`	Brief description of the project
`status`	`varchar(50)`	Status (e.g., active, archived, draft)
`created_at`	`timestamp`	Creation timestamp
`updated_at`	`timestamp`	Last updated timestamp
`created_by`	`varchar(100)`	User ID or name of the creator
`updated_by`	`varchar(100)`	User ID or name of the last updater

---

3. Table: `project_users`

Purpose

Defines the association between users and projects.
Used for managing access and team assignments at the project level.

Columns

Column	Type	Description
`project_id`	`uuid`	Linked project identifier (references `projects.id`)
`user_auth0_id`	`varchar(255)`	Auth0 ID of the user assigned to the project
`role_id`	`uuid`	Role assigned to the user in this project (references `roles.id`)

Relationships

- `project_id` → `projects.id`
- `role_id` → `roles.id`

---

4. Table: `roles`

Purpose

Defines roles specific to this enterprise.
Roles determine access levels and permissions within projects and modules.

Columns

Column	Type	Description
`id`	`uuid`	Unique role identifier
`name`	`varchar(100)`	Role name (e.g., Admin, Developer, Viewer)
`description`	`text`	Description of the role
`project_id`	`uuid`	Optional link to a specific project (if role is project-level)
`default_role_id`	`uuid`	Reference to a default role (from `default_roles.id`) if this role is derived from a global template

Relationships

- `default_role_id` → `default_roles.id`
- `project_id` → `projects.id`

---

5. Table: `permissions`

Purpose

Defines fine-grained access control for custom roles within this enterprise.
Each record specifies which module and access type (e.g., read, write) a role has.

Columns

Column	Type	Description
id	uuid	Unique permission ID
module	varchar(50)	Module or functional area name
access_type	varchar(50)	Type of access (e.g., read, write, delete, execute)
role_id	uuid	Role associated with this permission

Relationships

- role_id → roles.id

---

6. Table: default_roles

Purpose

Provides a master list of predefined roles that can be copied into tenant-specific roles when an enterprise is provisioned.

Columns

Column	Type	Description
id	uuid	Unique default role ID
name	varchar(100)	Default role name
description	text	Description of the default role

---

7. Table: default_permissions

Purpose

Defines the default set of permissions associated with each default role.
Used as a baseline when creating enterprise-specific roles.

Columns

Column	Type	Description
id	uuid	Unique permission ID

module	varchar(50)	Module name
access_type	varchar(50)	Access type (e.g., read, write)
default_role_id	uuid	Default role to which this permission belongs

Relationships

- default_role_id → default_roles.id

---

8. Table: flyway_schema_history

Purpose

Internal Flyway table used for versioning and tracking schema migrations.
Ensures consistent database version control across all enterprise schemas.

Columns

Column	Type	Description
installed_rank	integer	Sequential rank of the migration
version	varchar(50)	Version number of the migration
description	varchar(200)	Description of the migration script
type	varchar(20)	Type of migration (e.g., SQL, repeatable)
script	varchar(1000)	Script name or path
checksum	integer	Checksum of the migration file
installed_by	varchar(100)	Username or process that installed the migration
installed_on	timestamp	Timestamp when migration was applied
execution_time	integer	Time taken to execute the migration (ms)
success	boolean	Migration success flag

---

9. Relationships Summary

From Table	Relationship	To Table	Description

`project_users.project_id`	→	`projects.id`	Assigns users to specific projects
`project_users.role_id`	→	`roles.id`	Defines user's role in a project
`roles.default_role_id`	→	`default_roles.id`	Maps tenant roles to base templates
`permissions.role_id`	→	`roles.id`	Defines permissions for a role
`default_permissions.default_role_id`	→	`default_roles.id`	Links default permissions to base roles

---

10. Tenant Data Flow Summary

Phase	Activity	Affected Tables
Provisioning	Schema creation & Flyway migration	`flyway_schema_history`
User Onboarding (SSO)	New enterprise users added after Auth0 authentication	`users`
Role Initialization	Default roles & permissions copied to tenant	`default_roles`, `default_permissions`, `roles`, `permissions`
Project Setup	New projects created for enterprise users	`projects`
Access Control	Users assigned to projects and roles	`project_users`, `permissions`

---

11. Design Notes

- Each enterprise (tenant) has its **own isolated schema** for data security and compliance.
- Uses **Auth0 IDs** for consistent identity management across global and tenant databases.
- Roles and permissions follow a **template-based inheritance model**, simplifying onboarding.
- Flyway ensures schema migrations are version-controlled per tenant.

Mongo Schemas

`users` Collection Schema

```
{
  "$jsonSchema": {
    "bsonType": "object",
    "required": ["username", "email", "organizationId", "active",
"createdAt"],
    "properties": {
      "_id": { "bsonType": "string", "description": "Unique user ID" },
      "username": { "bsonType": "string", "description": "User's login or
display username" },
      "email": { "bsonType": "string", "pattern": "^.+@.+$", "description":
"User's email address" },
      "firstName": { "bsonType": "string", "description": "User's first
name" },
      "lastName": { "bsonType": "string", "description": "User's last
name" },
      "organizationId": { "bsonType": "string", "description": "Tenant or
organization identifier" },
      "active": { "bsonType": "bool", "description": "Indicates if user
account is active" },
      "createdAt": { "bsonType": "date", "description": "Creation
timestamp" },
      "updatedAt": { "bsonType": ["date", "null"], "description": "Last
update timestamp" },
      "department": { "bsonType": ["string", "null"], "description": "User's
department" },
      "role": { "bsonType": ["string", "null"], "description": "User's
assigned role" }
    }
  }
}
```

`patient_records_<projectId>_<trialSlug>` Collection Schema

```
{
  "$jsonSchema": {
    "bsonType": "object",
    "required": ["recordId", "patientData", "createdAt"],
    "properties": {
      "_id": { "bsonType": "string", "description": "Record ID (UUID or Mongo
ID)" },
      "recordId": { "bsonType": "string", "description": "Unique identifier
for the patient record" },
      "patientData": {
        "bsonType": "object",
        "description": "Flexible key-value map of patient attributes"
      },
      "createdAt": { "bsonType": "date", "description": "Record creation
timestamp" },
      "updatedAt": { "bsonType": ["date", "null"], "description": "Record
last update timestamp" }
```

```
    }
  }
}
```

`executionRecord` Collection Schema

```
{
  "$jsonSchema": {
    "bsonType": "object",
    "required": ["executionId", "userId", "basePatientData", "executedAt"],
    "properties": {
      "_id": { "bsonType": "string", "description": "Execution ID (primary
key)" },
      "executionId": { "bsonType": "string", "description": "Unique execution
identifier" },
      "userId": { "bsonType": "string", "description": "ID of user who
executed the job" },
      "basePatientData": {
        "bsonType": "object",
        "description": "Input patient data used during execution"
      },
      "basePrediction": {
        "bsonType": ["array", "null"],
        "items": { "bsonType": "object" },
        "description": "List of prediction outputs or results"
      },
      "executedBy": { "bsonType": ["string", "null"], "description":
"Executor identity" },
      "executedAt": { "bsonType": "date", "description": "Execution timestamp
(UTC)" },
      "updatedBy": { "bsonType": ["string", "null"], "description": "Last
updater's identity" },
      "updatedAt": { "bsonType": ["date", "null"], "description": "Last
updated timestamp (UTC)" }
    }
  }
}
```