Central Multi-Application Database - Data Dictionary

Overview

This database serves as a single source of truth for multiple applications with centralized authentication, role-based access control, and flexible schema design using PostgreSQL's JSONB capabilities.

Table Descriptions

1. **users**

Purpose: Central user authentication and profile management

Column	Type	Constraints	Description
user_id	UUID	PRIMARY KEY	Unique identifier for each user
email	VARCHAR(255)	UNIQUE, NOT NULL	User's email address (used for login)
username	VARCHAR(100)	UNIQUE, NOT NULL	User's unique username
password_hash	VARCHAR(255)	NOT NULL	Bcrypt/Argon2 hashed password
first_name	VARCHAR(100)		User's first name
last_name	VARCHAR(100)		User's last name
phone_number	VARCHAR(20)		User's contact number
is_active	BOOLEAN	DEFAULT TRUE	Account active status
is_verified	BOOLEAN	DEFAULT FALSE	Email verification status
email_verified_at	TIMESTAMP		Timestamp of email verification
last_login_at	TIMESTAMP		Last successful login timestamp
metadata	JSONB	DEFAULT '{}'	**Flexible field** for custom attributes (e.g., profile picture, preferences, custom fields)

Column	Type	Constraints	Description
created_at	TIMESTAMP	DEFAULT NOW()	Record creation timestamp
updated_at	TIMESTAMP	DEFAULT NOW()	Last update timestamp

Indexes: email, username, is_active, metadata (GIN)

Notes:

- Use `metadata` JSONB column to add new user attributes without schema changes
- Example metadata: `{"profile_pic": "url", "language": "en", "timezone": "UTC"}`

---

2. **applications**

Purpose: Register and manage all applications using this database

Column	Type	Constraints	Description
app_id	UUID	PRIMARY KEY	Unique identifier for application
app_name	VARCHAR(100)	UNIQUE, NOT NULL	Human-readable application name
app_key	VARCHAR(255)	UNIQUE, NOT NULL	Public API key for application
app_secret	VARCHAR(255)	NOT NULL	Secret key for API authentication
description	TEXT		Application description
config	JSONB	DEFAULT '{}'	**Flexible configuration** (e.g., API endpoints, features enabled)
is_active	BOOLEAN	DEFAULT TRUE	Application active status
created_at	TIMESTAMP	DEFAULT NOW()	Record creation timestamp
updated_at	TIMESTAMP	DEFAULT NOW()	Last update timestamp

Indexes: app_key

Notes:

- Each app (App A, App B, etc.) gets one record
- Use `config` JSONB for app-specific settings without schema changes

---

3. **roles**

Purpose: Define user roles for Role-Based Access Control (RBAC)

Column	Type	Constraints	Description
role_id	UUID	PRIMARY KEY	Unique identifier for role
role_name	VARCHAR(100)	UNIQUE, NOT NULL	Role name (e.g., 'admin', 'user', 'moderator')
description	TEXT		Role description
is_system_role	BOOLEAN	DEFAULT FALSE	System-defined vs custom role
permissions_config	JSONB	DEFAULT '{}'	**Flexible permissions** configuration
created_at	TIMESTAMP	DEFAULT NOW()	Record creation timestamp
updated_at	TIMESTAMP	DEFAULT NOW()	Last update timestamp

Indexes: role_name

Default Roles: super_admin, admin, user, guest

---

4. **permissions**

Purpose: Define granular permissions for resources

Column	Type	Constraints	Description
permission_id	UUID	PRIMARY KEY	Unique identifier for permission
permission_name	VARCHAR(100)	UNIQUE, NOT NULL	Permission name (e.g., 'users.create')
resource_type	VARCHAR(100)		Resource this permission applies to
action	VARCHAR(50)		Action type (create, read, update, delete)
description	TEXT		Permission description
created_at	TIMESTAMP	DEFAULT NOW()	Record creation timestamp

Indexes: (resource_type, action)

Naming Convention: `{resource}.{action}` (e.g., users.create, documents.read)

---

5. **user_roles**

Purpose: Assign roles to users (many-to-many relationship)

Column	Type	Constraints	Description
user_role_id	UUID	PRIMARY KEY	Unique identifier
user_id	UUID	FOREIGN KEY (users)	Reference to user
role_id	UUID	FOREIGN KEY (roles)	Reference to role
assigned_at	TIMESTAMP	DEFAULT NOW()	Role assignment timestamp
expires_at	TIMESTAMP		Optional expiration date
assigned_by	UUID	FOREIGN KEY (users)	User who assigned this role

Indexes: user_id, role_id, expires_at

Constraints: UNIQUE(user_id, role_id)

---

6. **role_permissions**

Purpose: Assign permissions to roles (many-to-many relationship)

Column	Type	Constraints	Description
role_permission_id	UUID	PRIMARY KEY	Unique identifier
role_id	UUID	FOREIGN KEY (roles)	Reference to role
permission_id	UUID	FOREIGN KEY (permissions)	Reference to permission
constraints	JSONB	DEFAULT '{}'	**Flexible constraints** (e.g., time-based, IP-based)
created_at	TIMESTAMP	DEFAULT NOW()	Record creation timestamp

Indexes: role_id, permission_id

Constraints: UNIQUE(role_id, permission_id)

---

7. **user_sessions**

Purpose: Manage active user sessions across all applications

Column	Type	Constraints	Description
session_id	UUID	PRIMARY KEY	Unique session identifier
user_id	UUID	FOREIGN KEY (users)	Reference to user
app_id	UUID	FOREIGN KEY (applications)	Reference to application
session_token	VARCHAR(500)	UNIQUE, NOT NULL	JWT or session token
refresh_token	VARCHAR(500)		Refresh token for token renewal
ip_address	VARCHAR(45)		User's IP address
user_agent	TEXT		Browser/device information
expires_at	TIMESTAMP	NOT NULL	Session expiration timestamp
created_at	TIMESTAMP	DEFAULT NOW()	Session creation timestamp
last_activity_at	TIMESTAMP	DEFAULT NOW()	Last activity timestamp

Indexes: user_id, app_id, session_token, expires_at

Notes:

- Single Sign-On (SSO): User logs in once, sessions created for each app
- Automatic cleanup: Use `clean_expired_sessions()` function

---

8. **user_app_preferences**

Purpose: Store user-specific preferences and data for each application

Column	Type	Constraints	Description
preference_id	UUID	PRIMARY KEY	Unique identifier
user_id	UUID	FOREIGN KEY (users)	Reference to user
app_id	UUID	FOREIGN KEY (applications)	Reference to application
preferences	JSONB	DEFAULT '{}'	**UI preferences** (theme, language, layout)

Column	Type	Constraints	Description
app_specific_data	JSONB	DEFAULT '{}'	**Any app-specific data** without schema changes
last_accessed_at	TIMESTAMP		Last access timestamp
created_at	TIMESTAMP	DEFAULT NOW()	Record creation timestamp
updated_at	TIMESTAMP	DEFAULT NOW()	Last update timestamp

Indexes: user_id, app_id

Constraints: UNIQUE(user_id, app_id)

Examples:

- `preferences`: `{"theme": "dark", "language": "en", "notifications": true}`
- `app_specific_data`: `{"dashboard_widgets": [...], "saved_filters": {...}}`

---

9. **password_reset_tokens**

Purpose: Manage password reset requests securely

Column	Type	Constraints	Description
token_id	UUID	PRIMARY KEY	Unique identifier
user_id	UUID	FOREIGN KEY (users)	Reference to user
token	VARCHAR(500)	UNIQUE, NOT NULL	Reset token (hashed)
expires_at	TIMESTAMP	NOT NULL	Token expiration (typically 1 hour)
is_used	BOOLEAN	DEFAULT FALSE	Whether token has been used
created_at	TIMESTAMP	DEFAULT NOW()	Token creation timestamp

Indexes: user_id, token, expires_at

---

10. **app_resources**

Purpose: Store application-specific resources (documents, reports, files, etc.)

Column	Type	Constraints	Description
resource_id	UUID	PRIMARY KEY	Unique identifier
app_id	UUID	FOREIGN KEY (applications)	Reference to application
resource_type	VARCHAR(100)	NOT NULL	Resource type (document, report, file)
resource_name	VARCHAR(255)	NOT NULL	Resource name/title
resource_data	JSONB	DEFAULT '{}'	**Flexible resource data**
access_rules	JSONB	DEFAULT '{}'	**Access control rules**
created_at	TIMESTAMP	DEFAULT NOW()	Record creation timestamp
updated_at	TIMESTAMP	DEFAULT NOW()	Last update timestamp

Indexes: app_id, resource_type, resource_data (GIN)

Examples:

- Document: `{"title": "Report Q4", "url": "...", "size": 1024}`
- Access rules: `{"roles": ["admin"], "users": ["uuid1", "uuid2"]}`

---

11. audit_logs

Purpose: Complete audit trail of all user actions

Column	Type	Constraints	Description
log_id	UUID	PRIMARY KEY	Unique identifier
user_id	UUID	FOREIGN KEY (users)	Reference to user
app_id	UUID	FOREIGN KEY (applications)	Reference to application
action	VARCHAR(100)	NOT NULL	Action performed (login, create, update, delete)
resource_type	VARCHAR(100)		Type of resource affected
resource_id	VARCHAR(255)		ID of resource affected
old_values	JSONB		Previous values (for updates)
new_values	JSONB		New values (for updates)

Column	Type	Constraints	Description
ip_address	VARCHAR(45)		User's IP address
created_at	TIMESTAMP	DEFAULT NOW()	Action timestamp

Indexes: user_id, app_id, action, created_at

Notes: Immutable table - never delete, only insert

---

12. **resource_access_logs**

Purpose: Track access to specific resources

Column	Type	Constraints	Description
access_log_id	UUID	PRIMARY KEY	Unique identifier
resource_id	UUID	FOREIGN KEY (app_resources)	Reference to resource
user_id	UUID	FOREIGN KEY (users)	Reference to user
action	VARCHAR(50)	NOT NULL	Action attempted (view, download, edit)
is_allowed	BOOLEAN	NOT NULL	Whether access was granted
denial_reason	VARCHAR(255)		Reason for denial (if applicable)
accessed_at	TIMESTAMP	DEFAULT NOW()	Access timestamp

Indexes: resource_id, user_id, accessed_at

Views

v_user_roles

Combines user information with their assigned roles for easy querying.

v_active_sessions

Shows all currently active sessions across all applications.

Functions

user_has_permission(user_id UUID, permission_name VARCHAR)

Returns: BOOLEAN

Purpose: Check if a user has a specific permission

clean_expired_sessions()

Returns: INTEGER

Purpose: Remove expired sessions (run periodically via cron)

Extensibility Strategy

Adding New User Fields

```
-- No schema change needed!
UPDATE users
SET metadata = metadata || '{"department": "Engineering", "employee_id": "E
123"}'
WHERE user_id = 'xxx';
```

Adding New Application Config

```
-- No schema change needed!
UPDATE applications
SET config = config || '{"max_file_size": 10485760, "allowed_formats": ["pdf",
"docx"]}'
WHERE app_id = 'yyy';
```

Adding New Resource Types

```
-- No schema change needed!
INSERT INTO app_resources (app_id, resource_type, resource_name, resourc
e_data)
```

```
VALUES ('app-id', 'video', 'Training Video 1',
        '{"url": "...", "duration": 300, "resolution": "1080p"}');
```

Security Best Practices

1. **Password Storage:** Always use bcrypt or Argon2 (never plain text)
2. **Session Tokens:** Use JWT with short expiration (15-30 minutes)
3. **Refresh Tokens:** Longer expiration (7-30 days), rotated on use
4. **API Keys:** Store `app_secret` hashed, never expose in client code
5. **Row-Level Security:** Implement PostgreSQL RLS for multi-tenancy
6. **Audit Everything:** Log all sensitive operations to `audit_logs`

Migration & Maintenance

Regular Maintenance Tasks

1. Run `clean_expired_sessions()` daily
2. Archive old `audit_logs` quarterly
3. Monitor JSONB column sizes
4. Update statistics: `ANALYZE;`

Backup Strategy

- Full backup: Daily
- Incremental backup: Every 6 hours
- Point-in-time recovery enabled