

# 데이터베이스 설계서 (Database Design Description)

## 1. 개요 (Overview)

### 1.1 설계 원칙

- 아키텍처:** MSA (Microservices Architecture) 기반으로 서비스별 독립적인 스키마 구성
- 식별자 전략:** UUID v7을 사용하여 시간 순 정렬 성능과 고유성 보장
- 보안 전략:**
  - 서비스 간 참조는 Logical Reference (UUID 값만 저장) 원칙 준수 (FK 제거)
  - 민감 정보(PII)는 encrypted\_data 컬럼에 AES-256 암호화된 JSON으로 저장
- 확장성:** 대용량 로그 데이터에 대한 Partitioning 전략 적용

## 2. 공통 사항 (Common)

### 2.1 표준 컬럼 정의

모든 테이블에 공통적으로 적용되거나 자주 사용되는 컬럼 정의입니다.

컬럼명	데이터 타입	설명	비고
id	UUID	Primary Key	UUID v7 사용 (Time-ordered)
is_active	BOOLEAN	활성화 여부	논리적 삭제(Soft Delete) 지원
created_at	TIMESTAMPTZ	생성 일시	기본값 NOW()
updated_at	TIMESTAMPTZ	수정 일시	Trigger로 자동 갱신
organization_id	UUID	소속 조직 ID	Multi-tenancy 지원

## 3. 서비스별 스키마 상세

### 3.1 신원 및 접근 관리 서비스 (IAM Service)

역할: 계정 관리, 인증(AuthN), 인가(AuthZ)

#### 3.1.1 Users (사용자)

시스템 로그인 계정을 관리합니다.

- 주요 컬럼: `username` (Unique), `password_hash`, `encrypted_data` (이름/연락처 등)

### 3.1.2 Roles (역할)

권한의 집합(Role)을 정의합니다.

- 주요 컬럼: `role_type` (SYSTEM/ORGANIZATION), `name`

### 3.1.3 Permissions (권한)

API 또는 기능 단위의 접근 제어 요소입니다.

- 주요 컬럼: `permission_code` (예: `USER:READ`)

### 3.1.4 매팅 테이블

- `role_permissions` : 역할-권한 (N:M)
- `user_roles` : 사용자-역할 (N:M)

## 3.2 인사 및 조직 서비스 (HR Service)

역할: 조직 구조 및 임직원 마스터 데이터 관리

### 3.2.1 Organizations (조직)

최상위 고객사(Tenant) 정보입니다.

- 주요 컬럼: `organization_code`, `encrypted_business_registration_number`

### 3.2.2 Departments (부서)

트리 구조의 부서 정보를 관리합니다.

- 주요 컬럼: `parent_department_id` (Self-referencing), `department_code`

### 3.2.3 Employees (직원)

임직원 개인 프로필 정보입니다.

- 주요 컬럼: `user_id` (IAM 연동), `encrypted_data` (인적사항)

### 3.2.4 Employee Assignments (발령)

직원의 현재 소속/직무 상태를 관리합니다.

- 관계: Department, Job, Position, Grade 참조

### 3.3 공간 서비스 (Space Service)

역할: 물리적/논리적 공간 계층 및 지도 관리

#### 3.3.1 공간 계층 (Hierarchy)

- `sites` : 최상위 거점 (예: 본사)
- `facilities` : 건물/시설 (Type: BUILDING, OUTDOOR)
- `sectors` : 층/구획 (Type: FLOOR, SECTION)
- `areas` : 세부 구역 (Type: GENERAL, SECURITY 등)

#### 3.3.2 Access Points (출입 지점)

실제 출입 통제가 이루어지는 논리적 지점(문)입니다.

- 주요 컬럼: `access_point_type` (DOOR, GATE, etc), `from_area_id`, `to_area_id`

#### 3.3.3 Map Views (지도)

관제 모니터링을 위한 2D/3D 지도 정보입니다.

- 주요 컬럼: `map_type` (2D/3D), `image_url`, `model_url`

### 3.4 장비 서비스 (Device Service)

역할: 하드웨어 수명 주기 및 펌웨어 관리

#### 3.4.1 Devices (장비)

물리적 하드웨어 자산 정보입니다.

- 주요 컬럼: `device_type` (ACU, READER 등), `ip_address`, `mac_address`

#### 3.4.2 Access Point Devices (설치 매팅)

공간(문)과 장비의 연결 정보를 관리합니다.

- 주요 컬럼: `access_point_id` (Space SVC 참조), `device_role` (ENTRY/EXIT)

### 3.5 출입 제어 서비스 (Access Control Service)

역할: 출입 권한 정책 수립 및 판정 기준 관리

#### 3.5.1 권한 정책 요소

- `time_schedules` : 언제 출입 가능한가? (주간/상세 시간)

- `access_groups` : 어디를 출입 가능한가? (출입문 그룹핑)
- `operator_levels` : 출입자의 보안 등급
- `access_thresholds` : 출입문의 보안 임계값

### 3.5.2 Credentials (인증 수단)

- 주요 컬럼: `credential_type` (RFID, FACE, PIN 등), `credential_status` (ACTIVE, LOST)

### 3.5.3 Access Grants (권한 부여)

사람( `personnel` )에게 출입 규칙( `access_rule` )을 할당합니다.

- 구조: Personnel(1) : Grant(N) : Access Rule(1)

## 3.6 로그 서비스 (Log Service)

역할: 대용량 이벤트 기록 (Partitioning 적용)

### 3.6.1 주요 로그 테이블

- `access_logs` : 출입 인증 성공/실패 이력 (가장 데이터 많음)
- `system_alarms_logs` : 화재, 강제 개방 등 중요 이벤트
- `device_status_logs` : 장비 Online/Offline 상태 변경
- `system_audit_logs` : 관리자 조작 감사 로그

## 3.7 근태 서비스 (Attendance Service)

역할: 출퇴근 정책 및 휴가 관리

### 3.7.1 Attendance Policies (근태 정책)

근무 유형(유연/고정/교대) 및 코어 타임 등을 정의합니다.

- 주요 컬럼: `rules` (JSONB로 상세 규칙 저장)

### 3.7.2 Employee Leaves (휴가)

휴가 신청 및 승인 상태를 관리합니다.

- 주요 컬럼: `start_date`, `end_date` (로직 처리를 위해 평문 저장 권장), `encrypted_data` (사유)

## 3.8 자동화 서비스 (Automation Service)

역할: 이벤트 기반 자동 제어 (ECA: Event-Condition-Action)

### 3.8.1 Event Rules (자동화 규칙)

- `event_rules` : 규칙 헤더
- `event_rule_conditions` : 조건 (IF 화재 감지 시)
- `event_rule_actions` : 동작 (THEN 모든 문 개방)

## 4. 데이터 저장 및 보안 전략 상세

### 4.1 개인정보 처리 (PII Handling)

- **대상 컬럼:** 이름, 전화번호, 이메일, 주민번호, 사진 등
- **저장 방식:** `encrypted_data` (TEXT) 컬럼에 JSON 포맷으로 압축 후 **AES-256 암호화**하여 저장
- **검색 전략:** 암호화된 데이터는 DB에서 직접 검색(`LIKE`) 불가하므로, 복호화된 데이터를 **Elasticsearch**에 인덱싱하여 검색 기능 제공

### 4.2 데이터 무결성 및 유니크 체크

- **이메일/전화번호 중복 확인:**
  - DB Unique Index 사용 불가 (암호화됨)
  - **해결:** 데이터 생성/수정 시 **Redis** set/hash 구조를 활용하거나, 암호화된 값 (Deterministic Encryption)을 별도 해시 컬럼으로 저장하여 중복 체크

### 4.3 대용량 데이터 처리 (Partitioning)

- **대상 테이블:** `access_logs`, `device_status_logs` 등 로그성 테이블
- **전략:** `created_at` 기준 **Range Partitioning** 적용 (월별 또는 일별 파티션 생성)

## ▼ 도메인별 서비스 구성 (Service Decomposition)

### 1. 신원 및 접근 관리 서비스 (IAM Service)

**역할:** 시스템 사용자 계정 관리, 인증(Authentication), 인가(Authorization), 시스템 접근 제어(RBAC)를 담당합니다.

- `users` : 시스템 로그인 계정
- `roles` : 역할 정보 (시스템 관리자, 일반 사용자 등)
- `permissions` : 시스템 기능별 접근 권한

- `role_permissions` : 역할과 권한의 매핑 (N:M)
- `user_roles` : 사용자와 역할의 매핑 (N:M)

## 2. 인사 및 조직 서비스 (HR Service)

**역할:** 기업의 조직 구조(부서), 직무/직책/직급 체계, 직원 프로필 및 발령 정보를 관리합니다. 타 서비스의 기초 데이터가 됩니다.

- `organizations` : 최상위 조직/고객사 정보
- `departments` : 부서 구조
- `jobs` : 직무 정보
- `positions` : 직책 정보
- `grades` : 직급 정보
- `employees` : 직원 개인 프로필
- `employee_assignments` : 직원의 현재 소속 및 발령 상태
- `employee_assignment_histories` : 직원의 과거 발령 이력

## 3. 공간 서비스 (Space Service)

**역할:** 물리적인 공간(사이트, 건물, 층, 구역)과 논리적인 출입 지점(문, 게이트)을 계층적으로 관리하며, 관제용 지도(Map) 정보를 제공합니다.

- `sites` : 최상위 물리적 거점 (예: 판교 캠퍼스)
- `facilities` : 건물 또는 시설물
- `sectors` : 층(Floor) 또는 구획
- `areas` : 세부 구역 (방, 로비 등)
- `access_points` : 논리적 출입 지점 (문, 게이트 등)
- `map_views` : 2D/3D 도면 및 모델 뷰
- `map_items` : 지도에 배치되는 아이콘/객체 정보
- `map_view_map_items` : 맵 뷰와 아이템의 배치 좌표 및 설정

## 4. 장비 서비스 (Device Service)

**역할:** 출입 통제 장치(ACU, 리더기, 카메라 등)의 하드웨어 수명 주기, 네트워크 설정, 펌웨어를 관리합니다. 공간 서비스의 '출입 지점'에 물리적 장비를 연결합니다.

- `devices` : 물리적 장비 정보 (IP, MAC, 모델명 등)
- `access_point_devices` : 출입 지점(Space)과 장비의 연결 매핑
- `firmware_files` : 장비 업데이트용 펌웨어 파일 정보

## 5. 출입 제어 서비스 (Access Control Service)

역할: "누가(Who), 언제(When), 어디를(Where), 어떻게(How)" 출입할 수 있는지에 대한 핵심 보안 정책과 권한을 관리합니다.

- **시간 및 일정 (When)**
  - `time_schedules` : 출입 가능 시간 패턴
  - `time_schedule_items` : 요일별 상세 시간
  - `holidays` : 공휴일 및 휴무일 관리
- **보안 등급 및 정책 (How - Level)**
  - `operator_levels` : 출입자에게 부여되는 보안 등급
  - `access_thresholds` : 출입문에 설정되는 최소 요구 등급
- **출입 주체 및 인증 수단 (Who & How - Method)**
  - `personnels` : 출입증 발급 대상자 (직원, 방문객 등)
  - `authentication_rules` : 인증 방식 정의 (카드, 지문, 얼굴 등)
  - `credentials` : 개인이 소지한 인증 매체 (카드 번호, 생체 정보 등)
  - `credential_histories` : 인증 매체 발급/폐기 이력
- **출입문 속성 및 그룹 (Where)**
  - `access_point_configs` : 개별 출입문의 보안 설정 (입/출입 규칙)
  - `access_groups` : 편의를 위한 출입문 그룹핑
  - `access_group_items` : 그룹에 속한 출입문 목록
- **권한 규칙 및 부여 (Assignment)**
  - `access_rules` : 출입 규칙 정의 (어디를 + 언제)
  - `access_rule_items` : 규칙 상세 (공간 + 시간 조합)
  - `access_grants` : 사람에게 규칙을 부여하여 최종 권한 생성
  - `access_grant_histories` : 권한 부여/회수 이력

- **특수 제어**

- `interlocks` : 인터락(Mantrap) 등 특수 연동 제어 그룹
- `interlock_access_points` : 인터락에 포함된 출입문

## 6. 로그 서비스 (Log Service)

**역할:** 시스템에서 발생하는 모든 이벤트(출입, 시스템 알람, 장비 상태, 조작 이력)를 대용량으로 저장하고 관리합니다. (파티셔닝 적용)

- `access_logs` : 출입 인증 성공/실패 기록
- `meal_logs` : 식수 태깅 기록
- `device_status_logs` : 장비 상태 변경(Online/Offline 등) 기록
- `device_communication_logs` : 장비 통신 패킷 로그 (디버깅용)
- `system_alarms_logs` : 화재, 강제 개방 등 중요 알람
- `system_audit_logs` : 관리자 조작 감사 로그
- `automation_logs` : 자동화 규칙 수행 결과 로그

## 7. 통계 및 분석 서비스 (Analytics Service)

**역할:** 로그 데이터를 집계하여 대시보드나 리포트에서 사용할 요약(Summary) 데이터를 생성합니다.

- `access_daily_summaries` : 일별 개인 출입 통계
- `access_monthly_summaries` : 월별 개인 출입 통계
- `attendance_daily_summaries` : 일별 근태 마감 데이터
- `attendance_monthly_summaries` : 월별 근태 마감 데이터
- `meal_daily_summaries` : 일별 식수 통계
- `meal_monthly_summaries` : 월별 식수 통계

## 8. 근태 서비스 (Attendance Service)

**역할:** 출퇴근 정책을 정의하고, 휴가 신청 및 사용 내역을 관리합니다.

- `attendance_policies` : 근무 정책 (유연근무, 고정근무 등)
- `leave_types` : 휴가 종류 (연차, 반차, 병가 등)

- `employee_leaves` : 직원 휴가 신청 및 잔여 현황
- `employee_leave_histories` : 휴가 상태 변경 이력

## 9. 식사 서비스 (Meal Service)

**역할:** 구내식당의 메뉴, 단가, 끼니 종류(조/중/석식)를 관리합니다.

- `meal_types` : 식사 구분 및 단가 정보
- `weekly_menus` : 일자별/식사별 식단 정보

## 10. 자동화 서비스 (Automation Service)

**역할:** "A가 발생하면 B를 실행하라"는 형태의 이벤트 기반 자동화 규칙(Rule Engine)을 관리합니다.

- `system_object_types` : 자동화 대상 객체 타입 정의 (메타데이터)
- `event_trigger_types` : 트리거 이벤트 타입 정의 (메타데이터)
- `event_action_types` : 수행 동작 타입 정의 (메타데이터)
- `event_rules` : 사용자 정의 자동화 규칙 헤더
- `event_rule_conditions` : 규칙 발동 조건 (IF)
- `event_rule_actions` : 규칙 수행 동작 (THEN)

## ▼ SQL문

### 공통 함수 및 트리거 (Functions & Triggers)

```
-- UUID v7 생성 함수
CREATE OR REPLACE FUNCTION uuid_generate_v7()
RETURNS UUID AS $$

DECLARE
    unix_ts_ms bytea;
    uuid_bytes bytea;

BEGIN
    unix_ts_ms := substring(int8send(floor(extract(epoch from clock_time
stamp()) * 1000)::bigint) from 3);
    uuid_bytes := unix_ts_ms || gen_random_bytes(10);
    uuid_bytes := set_byte(uuid_bytes, 6, (get_byte(uuid_bytes, 6) & 15) |
112);
    uuid_bytes := set_byte(uuid_bytes, 8, (get_byte(uuid_bytes, 8) & 63) |
128);
    RETURN encode(uuid_bytes, 'hex')::UUID;
END;
$$ LANGUAGE plpgsql;

-- updated_at 자동 갱신 트리거
CREATE OR REPLACE FUNCTION update_timestamp()
RETURNS TRIGGER AS $$
BEGIN
    IF NEW IS DISTINCT FROM OLD THEN
        NEW.updated_at := now();
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

### 신원 및 접근 관리 서비스(Iam Service)

```

-- =====
=====
-- 1. 시스템 사용자 (Users)
-- 역할: 시스템에 로그인할 수 있는 계정 정보
-- =====
=====

CREATE TABLE users (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)

    username TEXT NOT NULL UNIQUE, -- 로그인 아이디 (DB 레벨 유일성 보장)
    password_hash TEXT NOT NULL, -- 비밀번호 해시 (SHA-256 + Salt + Pepper)
    password_salt TEXT NOT NULL, -- 비밀번호 암호화에 사용된 솔트(Salt) 값

    -- [개인정보 보안 강화]
    -- name, email, phone_number는 암호화하여 통합 저장
    -- email, phone_number는 Elasticsearch + Redis 조합으로 애플리케이션 레벨에서 처리
    -- AES-256으로 암호화된 개인정보 JSON (예: {"name": "...", "email": "...", "phone_number": "..."})
    encrypted_data TEXT NOT NULL,

    is_active BOOLEAN NOT NULL DEFAULT TRUE, -- 활성화 여부
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
    updated_at TIMESTAMPTZ -- 레코드 수정 일시
);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_users
BEFORE UPDATE ON users
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

-- =====
=====
```

```

=====
-- 2. 역할 (Roles)
-- 역할: 권한의 집합이자 사용자에게 부여되는 자격
=====

CREATE TABLE roles (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)
    organization_id UUID, -- 소속 조직 ID

    name TEXT NOT NULL, -- 역할 이름 (예: SYSTEM_SUPER_ADMIN, SYSTEM_ADMIN, SYSTEM_VIEWER, USER)
    description TEXT, -- 역할 설명

    -- [역할 성격 구분]
    -- SYSTEM: 시스템 전체 관리용 (SYSTEM_SUPER_ADMIN, SYSTEM_VIEWER 등)
    -- ORGANIZATION: 조직 내 업무용 (ORG_ADMIN, USER, FACILITY_MANAGER 등)
    role_type TEXT NOT NULL DEFAULT 'ORGANIZATION' CHECK (role_type IN ('SYSTEM', 'ORGANIZATION')),

    is_active BOOLEAN NOT NULL DEFAULT TRUE, -- 활성화 여부
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
    updated_at TIMESTAMPTZ -- 레코드 수정 일시
);

-- [PARTIAL UNIQUE INDEX]
CREATE UNIQUE INDEX idx_roles_name_unique ON roles(name)
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_roles_organization_id_name_unique ON roles(organization_id, name)
    WHERE is_active = true AND organization_id IS NOT NULL;

-- [INDEX]
CREATE INDEX idx_roles_type ON roles(role_type);

```

```

-- [TRIGGER]
CREATE TRIGGER set_timestamp_roles
BEFORE UPDATE ON roles
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

=====
=====

-- 3. 권한 (Permissions)
-- 역할: 시스템 기능(API) 단위의 접근 제어 요소
=====

=====

CREATE TABLE permissions (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)

        name TEXT NOT NULL UNIQUE,          -- 권한명
        description TEXT,                 -- 권한 설명
        permission_code TEXT NOT NULL UNIQUE, -- 권한 코드 (예: USER:CREATE, DEVICE:READ). 시스템 로직과 매핑되므로 중복 불가.
        permission_type TEXT,             -- 권한 종류 (예: SYSTEM_MANAGEMENT)

        is_active BOOLEAN NOT NULL DEFAULT TRUE,   -- 활성화 여부
        created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
        updated_at TIMESTAMPTZ                -- 레코드 수정 일시
);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_permissions
BEFORE UPDATE ON permissions
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

=====
=====

-- 4. 역할 - 권한 매핑 (Role Permissions)

```

```

-- 역할: 역할이 어떤 권한들을 가지는지 정의 (N:M 관계)
-- =====
=====

CREATE TABLE role_permissions (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)

        role_id UUID NOT NULL REFERENCES roles(id) ON DELETE CASCADE, -- 대상 역할 ID
        permission_id UUID NOT NULL REFERENCES permissions(id) ON DELETE CASCADE, -- 부여할 권한 ID

        created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
        updated_at TIMESTAMPTZ, -- 레코드 수정 일시

        UNIQUE (role_id, permission_id)
);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_role_permissions
BEFORE UPDATE ON role_permissions
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

-- =====
=====

-- 5. 사용자-역할 매팅 (User Roles)
-- 역할: 사용자에게 역할을 부여 (누가, 어디서, 무엇을 하는가)
-- =====
=====

CREATE TABLE user_roles (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)
        organization_id UUID, -- 소속 조직 ID

        user_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE, -- 대상 사용자 ID

```

```

role_id UUID NOT NULL REFERENCES roles(id) ON DELETE CASCADE,
-- 부여할 역할 ID

    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
    updated_at TIMESTAMPTZ -- 레코드 수정 일시
);

-- [PARTIAL UNIQUE INDEX]
CREATE UNIQUE INDEX idx_user_roles_global_unique ON user_roles(user_id, role_id)
    WHERE organization_id IS NULL;
CREATE UNIQUE INDEX idx_user_roles_organization_unique ON user_roles(organization_id, user_id, role_id)
    WHERE organization_id IS NOT NULL;

-- [INDEX]
CREATE INDEX idx_user_roles_user_id ON user_roles(user_id);
CREATE INDEX idx_user_roles_org_id ON user_roles(organization_id);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_user_roles
BEFORE UPDATE ON user_roles
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

```

## 인사 및 조직 서비스(Hr Service)

```

=====
=====
-- 1. 조직 (Organizations)
-- 역할: 시스템을 사용하는 고객사(Tenant) 정보의 최상위 루트
=====
=====
CREATE TABLE organizations (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)

```

```

name TEXT NOT NULL,      -- 조직명 (예: 삼성전자)
description TEXT,        -- 조직 설명
organization_code TEXT, -- 조직 식별 코드 (예: SAMSUNG)

encrypted_business_registration_number TEXT, -- 암호화된 사업자등록
번호 (AES-256)

-- [외부 시스템 연동 메타데이터]
external_system_id TEXT,      -- 외부 시스템 조직 식별키 (예: SAP_ORG_
ID)
last_sync_at TIMESTAMPTZ,     -- 마지막 동기화 일시
is_synced BOOLEAN DEFAULT FALSE, -- 외부 시스템과 연동된 데이터인
지 여부

is_active BOOLEAN NOT NULL DEFAULT TRUE,    -- 활성화 여부
created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성
일시
updated_at TIMESTAMPTZ                  -- 레코드 수정 일시
);

-- [PARTIAL UNIQUE INDEX]
CREATE UNIQUE INDEX idx_org_name_unique ON organizations(name)
WHERE is_active = true;
CREATE UNIQUE INDEX idx_org_code_unique ON organizations(organiz
ation_code) WHERE is_active = true;

-- [TRIGGER]
CREATE TRIGGER set_timestamp_organizations
BEFORE UPDATE ON organizations
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

=====
=====

-- 2. 부서 (Departments)
-- 역할: 조직 내의 부서 계층 구조 관리
=====
```

```

=====
CREATE TABLE departments (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(),      -- 고유 식별자
    (UUID v7)
    organization_id UUID REFERENCES organizations(id), -- 소속 조직 ID
    parent_department_id UUID REFERENCES departments(id), -- 상위 부서 ID

    name TEXT NOT NULL,      -- 부서명 (예: 개발팀, 인사팀)
    description TEXT,        -- 부서 설명
    department_code TEXT,   -- 부서 코드

    -- [외부 시스템 연동 메타데이터]
    external_system_id TEXT, -- 외부 시스템 조직 식별키 (예: SAP_ORG_ID)
    last_sync_at TIMESTAMPTZ, -- 마지막 동기화 일시
    is_synced BOOLEAN DEFAULT FALSE, -- 외부 시스템과 연동된 데이터인지 여부

    is_active BOOLEAN NOT NULL DEFAULT TRUE, -- 활성화 여부
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
    updated_at TIMESTAMPTZ                      -- 레코드 수정 일시
);

-- [PARTIAL UNIQUE INDEX]
CREATE UNIQUE INDEX idx_departments_name_unique ON departments(name)
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_departments_organization_id_name_unique
    ON departments(organization_id, name)
    WHERE is_active = true AND organization_id IS NOT NULL;

CREATE UNIQUE INDEX idx_departments_department_code_unique ON
    departments(department_code)
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_departments_organization_id_department_code_unique
    ON departments(organization_id, department_code)

```

```

WHERE is_active = true AND organization_id IS NOT NULL;

-- [TRIGGER]
CREATE TRIGGER set_timestamp_departments
BEFORE UPDATE ON departments
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

=====
=====

-- 3. 직무/직책/직급
-- 역할: 인사 관리를 위한 기준 정보
=====

-- 3-1. 직무 (Jobs - 예: 개발, 영업, 인사)
CREATE TABLE jobs (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자
    organization_id UUID REFERENCES organizations(id), -- 소속 조직 ID

    name TEXT NOT NULL, -- 직무명
    description TEXT, -- 직무 설명
    job_code TEXT, -- 직무 코드

    -- [외부 시스템 연동 메타데이터]
    external_system_id TEXT, -- 외부 시스템 조직 식별키 (예: SAP_ORG_ID)
    last_sync_at TIMESTAMPTZ, -- 마지막 동기화 일시
    is_synced BOOLEAN DEFAULT FALSE, -- 외부 시스템과 연동된 데이터인지 여부

    is_active BOOLEAN NOT NULL DEFAULT TRUE, -- 활성화 여부
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
    updated_at TIMESTAMPTZ -- 레코드 수정 일시
);

```

```

-- [PARTIAL UNIQUE INDEX]
CREATE UNIQUE INDEX idx_jobs_name_unique ON jobs(name)
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_jobs_organization_id_nameunique ON jobs
    (organization_id, name)
    WHERE is_active = true AND organization_id IS NOT NULL;

CREATE UNIQUE INDEX idx_jobs_job_code_unique ON jobs(job_code)
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_jobs_organization_id_job_code_unique ON j
obs(organization_id, job_code)
    WHERE is_active = true AND organization_id IS NOT NULL;

-- [TRIGGER]
CREATE TRIGGER set_timestamp_jobs
BEFORE UPDATE ON jobs
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

-- 3-2. 직책 (positions - 예: 팀장, 본부장, 파트장)
CREATE TABLE positions (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자
    (UUID v7)
    organization_id UUID REFERENCES organizations(id), -- 소속 조직 ID

    name TEXT NOT NULL, -- 직책명
    description TEXT, -- 직책 설명
    position_code TEXT, -- 직책 코드

    -- [외부 시스템 연동 메타데이터]
    external_system_id TEXT, -- 외부 시스템 조직 식별키 (예: SAP_ORG_
ID)
    last_sync_at TIMESTAMPTZ, -- 마지막 동기화 일시
    is_synced BOOLEAN DEFAULT FALSE, -- 외부 시스템과 연동된 데이터인
지 여부

    is_active BOOLEAN NOT NULL DEFAULT TRUE, -- 활성화 여부
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성

```

```

일시
updated_at TIMESTAMPTZ          -- 레코드 수정 일시
);

-- [PARTIAL UNIQUE INDEX]
CREATE UNIQUE INDEX idx_positions_name_unique ON positions(name)
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_positions_organization_id_name_unique ON
positions(organization_id, name)
    WHERE is_active = true AND organization_id IS NOT NULL;

CREATE UNIQUE INDEX idx_positions_position_code_unique ON positio
ns(position_code)
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_positions_organization_id_position_code_un
ique ON positions(organization_id, position_code)
    WHERE is_active = true AND organization_id IS NOT NULL;

-- [TRIGGER]
CREATE TRIGGER set_timestamp_positions
BEFORE UPDATE ON positions
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

-- 3-3. 직급 (Grades - 예: 사원, 대리, 과장 / G1, G2)
CREATE TABLE grades (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(),    -- 고유 식별자
    (UUID v7)
    organization_id UUID REFERENCES organizations(id), -- 소속 조직 ID

    name TEXT NOT NULL,      -- 직급명
    description TEXT,        -- 직급 설명
    grade_code TEXT,         -- 직급 코드
    step INTEGER DEFAULT 0, -- 직급 서열 (높을수록 상위 직급, 예: 사원=1, 대
리=2)

-- [외부 시스템 연동 메타데이터]
    external_system_id TEXT,     -- 외부 시스템 조직 식별키 (예: SAP_ORG_

```

ID)

```
    last_sync_at TIMESTAMPTZ,      -- 마지막 동기화 일시
    is_synced BOOLEAN DEFAULT FALSE, -- 외부 시스템과 연동된 데이터인지 여부
```

```
    is_active BOOLEAN NOT NULL DEFAULT TRUE,      -- 활성화 여부
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
    updated_at TIMESTAMPTZ                      -- 레코드 수정 일시
);
```

-- [PARTIAL UNIQUE INDEX]

```
CREATE UNIQUE INDEX idx_grades_name_unique ON grades(name)
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_grades_organization_id_name_unique ON gr
ades(organization_id, name)
    WHERE is_active = true AND organization_id IS NOT NULL;
```

```
CREATE UNIQUE INDEX idx_grades_grade_code_unique ON grades(gra
de_code)
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_grades_organization_id_grade_code_unique
ON grades(organization_id, grade_code)
    WHERE is_active = true AND organization_id IS NOT NULL;
```

-- [TRIGGER]

```
CREATE TRIGGER set_timestamp_grades
BEFORE UPDATE ON grades
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();
```

=====

=====

-- 4. 직원 (Employees)

-- 역할: 조직 구성원의 프로필 정보 관리

=====

=====

CREATE TABLE employees (

```

id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자
(UUID v7)
    organization_id UUID REFERENCES organizations(id), -- 소속 조직 ID

    user_id UUID, -- 시스템 접속 계정 ID

    -- [개인정보 암호화]
    -- 이름, 주민번호, 이메일, 핸드폰 번호, 생년월일, 주소, 사번, 등 개인 식별 정보
    를 JSON으로 묶어 AES-256 암호화
    encrypted_data TEXT NOT NULL,

    -- [외부 시스템 연동 메타데이터]
    external_system_id TEXT, -- 외부 시스템 조직 식별키 (예: SAP_ORG_
ID)
    last_sync_at TIMESTAMPTZ, -- 마지막 동기화 일시
    is_synced BOOLEAN DEFAULT FALSE, -- 외부 시스템과 연동된 데이터인
    지 여부

    is_active BOOLEAN NOT NULL DEFAULT TRUE, -- 활성화 여부
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성
    일시
    updated_at TIMESTAMPTZ -- 레코드 수정 일시
);

-- [INDEX]
CREATE INDEX idx_employees_user_id ON employees(user_id);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_employees
BEFORE UPDATE ON employees
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

```

```

=====
=====
-- 5. 직원 발령 정보 (Assignment)
=====
=====
```

```

-- 5.1 현재 발령 상태 (employee_assignments)
CREATE TABLE employee_assignments (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 현재 발령 상태
    고유 식별자 (UUID v7)
        organization_id UUID NOT NULL REFERENCES organizations(id) ON D
        ELETE CASCADE, -- 소속 조직 ID(조직 삭제 시 함께 삭제)
        department_id UUID NOT NULL REFERENCES departments(id), -- 소
        속 부서 ID
        job_id UUID REFERENCES jobs(id), -- 수행 직무 ID
        position_id UUID REFERENCES positions(id), -- 맡은 직책 ID
        grade_id UUID REFERENCES grades(id), -- 현재 직급 ID
        employee_id UUID NOT NULL REFERENCES employees(id) ON DELET
        E CASCADE, -- 직원 ID

        -- [보안 강화] 발령 사유 등 텍스트 정보는 암호화하여 저장
        -- 예: { "reason": "정기 인사", "note": "...", "is_main": "FALSE" ... }
        encrypted_data TEXT NOT NULL,

        -- [외부 시스템 연동 메타데이터]
        external_system_id TEXT, -- 외부 시스템 조직 식별키 (예: SAP_ORG_
        ID)
        last_sync_at TIMESTAMPTZ, -- 마지막 동기화 일시
        is_synced BOOLEAN DEFAULT FALSE, -- 외부 시스템과 연동된 데이터인
        지 여부

        created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성
        일시
        updated_at TIMESTAMPTZ -- 레코드 수정 일시
);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_employee_assignments
BEFORE UPDATE ON employee_assignments
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

-- 5-2. 발령 이력 (Histories)

```

```

CREATE TABLE employee_assignment_histories (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 발령 이력 고유
    식별자 (UUID v7)
        organization_id UUID NOT NULL,           -- 소속 조직 ID
        department_id UUID NOT NULL,             -- 소속 부서 ID
        job_id UUID,                            -- 당시의 수행 직무 ID
        position_id UUID,                      -- 당시의 맡았던 직책 ID
        grade_id UUID,                         -- 당시의 직급 ID
        employee_id UUID NOT NULL,             -- 직원 ID

        -- [보안 강화] 스냅샷 데이터 암호화 (AES-256)
        -- 발령 사유(reason)와 당시의 부서명(Name), 직급명(Name) 등을 JSON으로
        통합하여 암호화 저장
        -- ID는 컬럼으로 분리했지만, 가독성을 위한 Name 정보는 여기에 스냅샷으로 저
        장
        -- 예: { "dept_name": "영업1팀", "job_name": "...", "reason": "전배" }
        encrypted_snapshot_data TEXT NOT NULL,

    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성
    일시
    updated_at TIMESTAMPTZ                         -- 레코드 수정 일시
);

-- [INDEX]
CREATE INDEX idx_assignment_histories_organization ON employee_as
    signment_histories(organization_id);
CREATE INDEX idx_assignment_histories_department ON employee_assi
    gnment_histories(department_id);
CREATE INDEX idx_assignment_histories_employee ON employee_assig
    nment_histories(employee_id);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_assignment_histories
BEFORE UPDATE ON employee_assignment_histories
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

```

# 공간 서비스(Space Service)

```
-- 계층 구조
-- [실내]
-- site → facility(building) → sector(floor) → area → access_point(door,
elevator, ...)
-- [실외]
-- site → facility(outdoor) → sector(section) → area → access_point(doo
r, gate, ...)

=====
=====

-- 1. 사이트 (Sites)
-- 역할: 최상위 물리적 거점 (예: 서울 본사, 부산 공장)
-- =====
=====

CREATE TABLE sites (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (U
    UUID v7)
    organization_id UUID, -- 소속 조직 ID

    name TEXT NOT NULL, -- 사이트명 (예: 판교 캠퍼스)
    description TEXT, -- 사이트 설명
    site_code TEXT, -- 사이트 식별 코드 (외부 연동 및 엑셀 업로드용)
    address TEXT, -- 주소 정보

    is_active BOOLEAN NOT NULL DEFAULT TRUE, -- 활성화 여부
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성
    일시
    updated_at TIMESTAMPTZ -- 레코드 수정 일시
);

-- [PARTIAL UNIQUE INDEX]
CREATE UNIQUE INDEX idx_sites_name ON sites(name)
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_sites_organization_id_name ON sites(organ
    ization_id, name)
    WHERE is_active = true AND organization_id IS NOT NULL;
```

```

CREATE UNIQUE INDEX idx_sites_site_code ON sites(site_code)
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_sites_organization_id_site_code ON sites(or
ganization_id, site_code)
    WHERE is_active = true AND organization_id IS NOT NULL;

-- [TRIGGER]
CREATE TRIGGER set_timestamp_sites
BEFORE UPDATE ON sites
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

=====
=====

-- 2. 시설물 (Facilities)
-- 역할: 사이트 내에 존재하는 물리적 구조물 (건물, 주차장, 운동장 등)
=====

CREATE TABLE facilities (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (U
UID v7)
    organization_id UUID, -- 소속 조직 ID
    site_id UUID REFERENCES sites(id), -- 소속 사이트 ID

    name TEXT NOT NULL, -- 시설명 (예: 본관, 제1주차장)
    description TEXT, -- 시설 설명
    facility_code TEXT, -- 시설 코드

    -- 건물(BUILDING)인지 야외 구역(OUTDOOR)인지 구분
    facility_type TEXT NOT NULL DEFAULT 'BUILDING' CHECK (facility_ty
pe IN ('BUILDING', 'OUTDOOR')),

    -- 지도 표시를 위한 경계 좌표 (GeoJSON Polygon 등)
    boundary_coordinates JSONB,

    is_active BOOLEAN NOT NULL DEFAULT TRUE, -- 활성화 여부
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성

```

```

일시
updated_at TIMESTAMPTZ          -- 레코드 수정 일시
);

-- [INDEX]
CREATE INDEX idx_facilities_organization ON facilities(organization_id);
CREATE INDEX idx_facilities_site ON facilities(site_id);

-- [PARTIAL UNIQUE INDEX]
CREATE UNIQUE INDEX idx_facilities_name ON facilities(name)
    WHERE is_active = true AND site_id IS NULL;
CREATE UNIQUE INDEX idx_facilities_site_name ON facilities(site_id,
name)
    WHERE is_active = true AND site_id IS NOT NULL;

CREATE UNIQUE INDEX idx_facilities_facility_code ON facilities(facility_c
ode)
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_facilities_organization_id_facility_code ON f
acilities(organization_id, facility_code)
    WHERE is_active = true AND organization_id IS NOT NULL;

=====
=====

-- 3. 구획 (Sectors)
-- 역할: 시설물을 수직(층) 또는 수평(구역)으로 나눈 1차 하위 공간
=====

CREATE TABLE sectors (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (U
UID v7)
    organization_id UUID,                      -- 소속 조직 ID

    facility_id UUID REFERENCES facilities(id),   -- 소속 시설물 ID

    name TEXT NOT NULL, -- 구획명 (예: 1F, B1, A구역)
    description TEXT,  -- 구획 설명
    sector_code TEXT, -- 구획 코드

```

```

-- 층(FLOOR)인지 평면 구역(SECTION)인지 구분
sector_type TEXT NOT NULL DEFAULT 'FLOOR' CHECK (sector_type IN ('FLOOR', 'SECTION')),

sort_order INTEGER DEFAULT 0, -- 정렬 순서 (층수의 경우 지하는 음수 사용 가능)

is_active BOOLEAN NOT NULL DEFAULT TRUE, -- 활성화 여부
created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
updated_at TIMESTAMPTZ -- 레코드 수정 일시
);

-- [INDEX]
CREATE INDEX idx_sectors_organization ON sectors(organization_id);
CREATE INDEX idx_sectors_facility ON sectors(facility_id);

-- [PARTIAL UNIQUE INDEX]
CREATE UNIQUE INDEX idx_sectors_name ON sectors(name)
WHERE is_active = true AND facility_id IS NULL;
CREATE UNIQUE INDEX idx_sectors_facility_id_name ON sectors(facility_id, name)
WHERE is_active = true AND facility_id IS NOT NULL;

CREATE UNIQUE INDEX idx_sectors_sector_code ON sectors(sector_code)
WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_sectors_organization_id_sector_code ON sectors(organization_id, sector_code)
WHERE is_active = true AND organization_id IS NOT NULL;

-- [TRIGGER]
CREATE TRIGGER set_timestamp_sectors
BEFORE UPDATE ON sectors
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

```

```

-- =====
=====

-- 4. 구역 (Areas)
-- 역할: 층(실내) 또는 섹션(실외) 내의 세부 공간 (예: 서버실, 야외 출입문, 로비, 1
번 주차면)
-- =====
=====

CREATE TABLE areas (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (U
    UUID v7)
    organization_id UUID, -- 소속 조직 ID
    sector_id UUID REFERENCES sectors(id), -- 소속 구획 ID
    name TEXT NOT NULL, -- 구역명 (예: 101호 회의실)
    description TEXT, -- 구역 설명
    area_code TEXT, -- 구역 코드
    -- 구역의 용도 및 보안 타입 구분
    area_type TEXT NOT NULL DEFAULT 'GENERAL' CHECK (area_type I
    N ('GENERAL', 'SECURITY', 'FIRE', 'RESTRICTED', 'PARKING')),
    coordinates JSONB, -- 구역의 물리적 범위/좌표 (Polygon)
    is_active BOOLEAN NOT NULL DEFAULT TRUE, -- 활성화 여부
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성
    일시
    updated_at TIMESTAMPTZ -- 레코드 수정 일시
);

-- [INDEX]
CREATE INDEX idx_areas_organization ON areas(organization_id);
CREATE INDEX idx_areas_sector ON areas(sector_id);

-- [PARTIAL UNIQUE INDEX]
CREATE UNIQUE INDEX idx_areas_name ON areas(name)
    WHERE is_active = true AND sector_id IS NULL;
CREATE UNIQUE INDEX idx_areas_sector_id_name ON areas(sector_id,

```

```

name)
    WHERE is_active = true AND sector_id IS NOT NULL;

CREATE UNIQUE INDEX idx_areas_area_code ON areas(area_code)
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_areas_organization_id_area_code ON areas
(organization_id, area_code)
    WHERE is_active = true AND organization_id IS NOT NULL;

-- [TRIGGER]
CREATE TRIGGER set_timestamp_areas
BEFORE UPDATE ON areas
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

=====
=====

-- 5. 출입 지점 (Access Points)
-- 역할: 구역과 구역을 연결하거나 진입하는 물리적/논리적 통제 지점
=====

CREATE TABLE access_points (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)
    organization_id UUID,                                -- 소속 조직 ID
    area_id UUID REFERENCES areas(id),     -- 소속 구역 ID
    name TEXT NOT NULL,      -- 출입 지점명(예: 정문 게이트, 101호 문)
    description TEXT,        -- 출입 지점 설명
    access_point_code TEXT, -- 출입 지점 코드
    -- 출입 타입 (예: 출입문, 엘리베이터, 차량용 게이트)
    access_point_type TEXT NOT NULL DEFAULT 'DOOR'
    CHECK (access_point_type IN (
        'DOOR', 'GATE_DOOR', 'TURNSTILE', 'ELEVATOR', 'SPEED_GATE',
        'BARRIER', 'SLIDING', 'SHUTTER', 'BOLLARD'
    )),
```

```

-- [이동 경로] 안티패스백 및 재실 관리용 (어디서 어디로 이동하는가)
from_area_id UUID REFERENCES areas(id), -- 진입 전 구역
to_area_id UUID REFERENCES areas(id), -- 진입 후 구역

config JSONB DEFAULT '{}'::JSONB, -- 개방 시간, 릴레이 설정 등 타입별
속성

    is_active BOOLEAN NOT NULL DEFAULT TRUE, -- 활성화 여부
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성
    일시
    updated_at TIMESTAMPTZ -- 레코드 수정 일시
);

-- [INDEX]
CREATE INDEX idx_access_points_organization ON access_points(organization_id);
CREATE INDEX idx_access_points ON access_points(area_id);

-- [PARTIAL UNIQUE INDEX]
CREATE UNIQUE INDEX idx_access_points_name ON access_points(name)
    WHERE is_active = true AND area_id IS NULL;
CREATE UNIQUE INDEX idx_access_points_area_id_name ON access_points(area_id, name)
    WHERE is_active = true AND area_id IS NOT NULL;

CREATE UNIQUE INDEX idx_access_points_access_point_code ON access_points(access_point_code)
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_access_points_organization_id_access_point_code ON access_points(organization_id, access_point_code)
    WHERE is_active = true AND organization_id IS NOT NULL;

-- [TRIGGER]
CREATE TRIGGER set_timestamp_access_points
BEFORE UPDATE ON access_points

```

```
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();
```

```
-- =====
=====

-- 6. 맵 뷰 (Map Views)
-- 역할: 관제 모니터링을 위한 도면 및 3D 모델 관리
-- =====
=====

CREATE TABLE map_views (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)
    organization_id UUID, -- 소속 조직 ID

    -- [계층별 맵 매핑] 어떤 공간 계층을 보여주는 도면이지 지정 (하나만 설정됨)
    target_site_id UUID REFERENCES sites(id),
    target_facility_id UUID REFERENCES facilities(id),
    target_sector_id UUID REFERENCES sectors(id),
    target_area_id UUID REFERENCES areas(id),

    name TEXT NOT NULL, -- 맵 뷰명 (예: 1층 전체 평면도)
    description TEXT, -- 맵 뷰 설명

    -- [3D 지원] 맵 타입 구분 및 리소스 경로
    map_type TEXT NOT NULL DEFAULT '2D' CHECK (map_type IN ('2D', '3D')),
    image_url TEXT, -- 2D 도면 이미지 파일 URL
    model_url TEXT, -- 3D 모델 파일 URL (예: .glb, .gltf)

    config JSONB DEFAULT '{}'::JSONB, -- 맵 설정(배율, 초기 좌표 등)

    is_active BOOLEAN NOT NULL DEFAULT TRUE, -- 활성화 여부
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
    updated_at TIMESTAMPTZ, -- 레코드 수정 일시

    -- 하나의 맵 뷰는 하나의 타겟만 가져야 함
    CONSTRAINT check_map_target CHECK (
```

```

        num_nonnauls(target_site_id, target_facility_id, target_sector_id, target_area_id) = 1
    )
);

-- [PARTIAL UNIQUE INDEX]
CREATE UNIQUE INDEX idx_map_views_name ON map_views(name)
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_map_views_organization_id_name ON map_views(organization_id, name)
    WHERE is_active = true AND organization_id IS NOT NULL;

-- [TRIGGER]
CREATE TRIGGER set_timestamp_map_views
BEFORE UPDATE ON map_views
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

=====
=====

-- 7. 맵 아이템 (Map Items)
-- 역할: 맵에 배치할 수 있는 아이템(오브젝트)의 메타 데이터
=====

CREATE TABLE map_items (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자
    (UUID v7)
    organization_id UUID, -- 소속 조직 ID

    target_id UUID NOT NULL, -- 지도에 표시할 대상 객체의 ID
    target_type TEXT NOT NULL CHECK (target_type IN ('ACCESS_POINT', 'DEVICE', 'AREA', 'CAMERA')), -- 대상 타입

    name TEXT NOT NULL, -- 맵 아이템명
    description TEXT, -- 맵 설명

    icon_style JSONB DEFAULT '{}'::JSONB, -- 아이콘 스타일 (색상, 라벨 표시 여부 등)

```

```

    is_active BOOLEAN NOT NULL DEFAULT TRUE,      -- 활성화 여부
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성
    일시
    updated_at TIMESTAMPTZ                      -- 레코드 수정 일시
);

-- [PARTIAL UNIQUE INDEX]
CREATE UNIQUE INDEX idx_map_items_name ON map_items(name)
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_map_items_organization_id_name ON map_i
tems(organization_id, name)
    WHERE is_active = true AND organization_id IS NOT NULL;

-- [TRIGGER]
CREATE TRIGGER set_timestamp_map_items
BEFORE UPDATE ON map_items
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

=====
=====

-- 8. 맵 뷰-맵 아이템 매핑 및 배치 (Map View Map Items)
-- 역할: 특정 맵 뷰에 아이템을 배치하고 위치/회전/크기를 정의 (N:M)
=====

CREATE TABLE map_view_map_items (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (U
    ID v7)

    map_view_id UUID NOT NULL REFERENCES map_views(id) ON DELET
    E CASCADE, -- 맵 뷰 ID
    map_item_id UUID NOT NULL REFERENCES map_items(id) ON DELET
    E CASCADE, -- 맵 아이템 ID

    -- [3D 좌표] 도면/모델 상의 위치 (Unity/Three.js 좌표계 대응)
    position_x NUMERIC(10, 2) NOT NULL DEFAULT 0,
    position_y NUMERIC(10, 2) NOT NULL DEFAULT 0,

```

```

position_z NUMERIC(10, 2) NOT NULL DEFAULT 0,
-- [3D 회전] 아이템의 방향 (예: CCTV가 바라보는 방향)
rotation_x NUMERIC(10, 2) NOT NULL DEFAULT 0,
rotation_y NUMERIC(10, 2) NOT NULL DEFAULT 0,
rotation_z NUMERIC(10, 2) NOT NULL DEFAULT 0,
-- [3D 크기] 아이템 스케일
scale_x NUMERIC(10, 2) NOT NULL DEFAULT 1.0,
scale_y NUMERIC(10, 2) NOT NULL DEFAULT 1.0,
scale_z NUMERIC(10, 2) NOT NULL DEFAULT 1.0,
-- 배치별 개별 스타일 오버라이드
override_style JSONB,
created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(),
updated_at TIMESTAMPTZ
);
-- [INDEX]
CREATE INDEX idx_map_view_map_items_view ON map_view_map_items(map_view_id);
CREATE INDEX idx_map_view_map_items_item ON map_view_map_items(map_item_id);
-- [TRIGGER]
CREATE TRIGGER set_timestamp_map_view_map_items
BEFORE UPDATE ON map_view_map_items
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

```

## 장비 서비스(Device Service)

```

=====
=====
-- 1. 장비 (Devices)
-- 역할: 물리적 하드웨어의 생명주기 및 기본 속성 관리

```

```

-- =====
=====
CREATE TABLE devices (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)
    organization_id UUID, -- 소속 조직 ID

    -- [상위 장비 연결]
    -- 리더기가 ACU에 연결된 경우, ACU의 ID를 참조
    parent_device_id UUID REFERENCES devices(id),

    name TEXT NOT NULL, -- 장비명 (예: 정문 입구 리더기 1번)
    description TEXT, -- 장비 설명
    device_code TEXT, -- 장비 코드

    -- [장비 유형]
    -- ACU: Access Control Unit (메인 컨트롤러)
    -- READER: 카드리더기, 지문인식기 등 (ACU에 연결되거나 독립형)
    -- CONTROLLER: 엘리베이터 컨트롤러, 릴레이 보드 등
    -- CAMERA: CCTV, LPR 카메라
    -- SENSOR: 화재 감지기, 문열림 센서 등
    device_type TEXT NOT NULL CHECK (device_type IN ('ACU', 'READER', 'CONTROLLER', 'CAMERA', 'SENSOR')),

    -- [네트워크 정보]
    ip_address TEXT, -- 고정 IP 사용 시
    mac_address TEXT, -- 물리적 주소
    port INTEGER, -- 통신 포트

    serial_number TEXT, -- 시리얼 번호
    model_name TEXT, -- 모델명 (펌웨어 매칭용)
    firmware_version TEXT, -- 현재 펌웨어 버전

    -- [상세 설정 - JSONB]
    -- 통신 속도(Baud rate), LED 색상, 볼륨, 타임아웃 등 제조사별/모델별 상이한
    설정값
    -- 예: { "baud_rate": 9600, "led_color": "BLUE", "beep_volume": 5 }
    config JSONB DEFAULT '{}'::JSONB, -- 장비별 하드웨어 설정

```

```

    is_active BOOLEAN NOT NULL DEFAULT TRUE,      -- 활성화 여부
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성
    일시
    updated_at TIMESTAMPTZ                      -- 레코드 수정 일시
);

-- [PARTIAL UNIQUE INDEX]
CREATE UNIQUE INDEX idx_devices_name ON devices(name)
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_devices_organization_id_name ON devices
(organization_id, name)
    WHERE is_active = true AND organization_id IS NOT NULL;

CREATE UNIQUE INDEX idx_devices_device_code ON devices(device_c
ode)
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_devices_organization_id_device_code ON d
evices(organization_id, device_code)
    WHERE is_active = true AND organization_id IS NOT NULL;

CREATE UNIQUE INDEX idx_devices_mac_address ON devices(mac_ad
dress)
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_devices_organization_id_mac_address ON d
evices(organization_id, mac_address)
    WHERE is_active = true AND organization_id IS NOT NULL;

CREATE UNIQUE INDEX idx_devices_serial_number ON devices(serial_n
umber)
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_devices_organization_id_serial_number ON
devices(organization_id, serial_number)
    WHERE is_active = true AND organization_id IS NOT NULL;

-- [TRIGGER]
CREATE TRIGGER set_timestamp_devices
BEFORE UPDATE ON devices

```

```

FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

=====
=====

-- 2. 출입 지점 - 장비 매팅 (Access Point Devices)
-- 역할: 물리적 장비가 논리적 출입 지점(문, 게이트)에 어떻게 설치되었는지 정의
=====

CREATE TABLE access_point_devices (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)

    access_point_id UUID NOT NULL, -- 출입 지점(문, 게이트) ID

    device_id UUID NOT NULL REFERENCES devices(id) ON DELETE CASCADE, -- 장비 ID

    -- [방향 및 역할 판별]
    -- ENTRY: 입실/입차용 리더기 (Outside → Inside)
    -- EXIT: 퇴실/출차용 리더기 (Inside → Outside)
    -- CONTROLLER: 락(Lock)이나 차단기를 제어하는 컨트롤러 (방향성 없음)
    -- SENSOR: 문열림 감지 센서 등
    device_role TEXT NOT NULL CHECK (device_role IN ('ENTRY', 'EXIT',
    'CONTROLLER', 'SENSOR')),

    -- [하드웨어 결선 정보]
    -- 컨트롤러의 몇 번 포트에 연결되었는가? (Wiegand Port, Relay Port 번호)
    port_index INTEGER,

    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
    updated_at TIMESTAMPTZ -- 레코드 수정 일시

    UNIQUE (access_point_id, device_id)
);

-- [INDEX]

```

```

CREATE INDEX idx_access_point_devices_point ON access_point_devices(access_point_id);
CREATE INDEX idx_access_point_devices_device ON access_point_devices(device_id);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_access_point_devices
BEFORE UPDATE ON access_point_devices
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

=====
=====

-- 4. 펌웨어 파일 (Firmware Files)
-- 역할 장비 업데이트를 위한 바이너리 파일 관리 (OTA)
=====

=====

CREATE TABLE firmware_files (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 파일 고유 식별자 (UUID v7)

    name TEXT NOT NULL, -- 펌웨어 파일명
    description TEXT, -- 펌웨어 파일 설명

    -- 펌웨어는 특정 기기(ID)가 아니라 특정 모델(Model) 전체에 적용되므로 model 명을 저장
    device_model TEXT NOT NULL, -- 적용 가능한 모델명 (예: ACU-1000)
    version TEXT NOT NULL, -- 버전 (예: 1.0.2)

    file_url TEXT NOT NULL UNIQUE, -- S3 등 바이너리 파일 저장 경로

    -- [무결성 검증] 파일 원본의 SHA-256 해시값 (64글자 문자열)
    -- 장비가 다운로드 후 이 값과 비교하여 변조 여부 확인
    checksum TEXT NOT NULL,

    release_date DATE NOT NULL DEFAULT CURRENT_DATE, -- 배포일

    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성

```

```

일시
updated_at TIMESTAMPTZ          -- 레코드 수정 일시
);

-- [PARTIAL UNIQUE INDEX]
CREATE UNIQUE INDEX idx_firmware_files_device_model_version_unique
ON firmware_files(device_model, version);

-- [INDEX]
CREATE INDEX idx_firmware_model ON firmware_files(device_model, ve
rsion DESC);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_firmware_files
BEFORE UPDATE ON firmware_files
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

```

## 출입 제어 서비스(Access Control Service)

```

=====

-- 1. 시간 정책 (Time Policies) - [언제 (When)]
-- 역할: 출입이 허용되는 시간대 정의
=====

-- 1-1. 시간 스케줄 (Time Schedules)
CREATE TABLE time_schedules (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (U
    UID v7)
    organization_id UUID,                      -- 소속 조직 ID

    name TEXT NOT NULL, -- 시간 스케줄명 (예: 평일 근무시간)
    description TEXT,  -- 시간 스케줄 설명
    schedule_code TEXT, -- 시간 스케줄 코드

```

```

    is_active BOOLEAN NOT NULL DEFAULT TRUE,      -- 활성화 여부
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성
    일시
    updated_at TIMESTAMPTZ                      -- 레코드 수정 일시
);

-- [PARTIAL UNIQUE INDEX]
CREATE UNIQUE INDEX idx_time_schedules_name ON time_schedules
(name)
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_time_schedules_organization_id_name ON ti
me_schedules(organization_id, name)
    WHERE is_active = true AND organization_id IS NOT NULL;

CREATE UNIQUE INDEX idx_time_schedules_schedule_code ON time_sc
hedules(schedule_code)
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_time_schedules_organization_id_schedule_c
ode ON time_schedules(organization_id, schedule_code)
    WHERE is_active = true AND organization_id IS NOT NULL;

-- [TRIGGER]
CREATE TRIGGER set_timestamp_time_schedules
BEFORE UPDATE ON time_schedules
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

-- 1-2. 시간 스케줄 상세 (Schedule Items)
CREATE TABLE time_schedule_items (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (U
    UUID v7)
    time_schedule_id UUID NOT NULL REFERENCES time_schedules(id)
    ON DELETE CASCADE,
    day_of_week INTEGER NOT NULL, -- 1:월 ~ 7:일, 8: 공휴일1, 9: 공휴일2,
    10: 공휴일3
    start_time TIME NOT NULL,    -- 시작 시간
    end_time TIME NOT NULL,     -- 종료 시간

```

```

    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성
    일시
    updated_at TIMESTAMPTZ          -- 레코드 수정 일시
);

-- [INDEX]
CREATE INDEX idx_schedule_items_parent ON time_schedule_items(tim
e_schedule_id);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_time_schedule_items
BEFORE UPDATE ON time_schedule_items
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

-- 1-3. 공휴일 (Holidays)
CREATE TABLE holidays (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (U
    UID v7)
    organization_id UUID,           -- 소속 조직 ID

    name TEXT NOT NULL,      -- 공휴일명 (예: 설날)
    description TEXT,        -- 공휴일 설명
    holiday_code TEXT,       -- 공휴일 코드
    holiday_date DATE NOT NULL, -- 공휴일 날짜 (예: YYYY-MM-DD)

    is_recurring BOOLEAN NOT NULL DEFAULT FALSE, -- 반복 여부
    is_active BOOLEAN NOT NULL DEFAULT TRUE,     -- 활성화 여부
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성
    일시
    updated_at TIMESTAMPTZ          -- 레코드 수정 일시
);

-- [Partial Unique Index]
CREATE UNIQUE INDEX idx_holidays_name ON holidays(name)
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_holidays_organization_id_name ON holidays

```

```

(organization_id, name)
    WHERE is_active = true AND organization_id IS NOT NULL;

CREATE UNIQUE INDEX idx_holidays_holiday_code ON holidays(holiday_
code)
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_holidays_organization_id_holiday_code ON
holidays(organization_id, holiday_code)
    WHERE is_active = true AND organization_id IS NOT NULL;

CREATE UNIQUE INDEX idx_holidays_holiday_date ON holidays(holiday_
date)
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_holidays_organization_id_holiday_date ON h
olidays(organization_id, holiday_date)
    WHERE is_active = true AND organization_id IS NOT NULL;

-- [TRIGGER]
CREATE TRIGGER set_timestamp_holidays
BEFORE UPDATE ON holidays
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

```

```

=====
=====
-- 2. 보안 레벨 및 임계값 (Security Levels) - [어떻게 (How) - 레벨]
-- 역할: 출입자와 출입문의 등급을 비교하여 출입 허용 여부를 결정
=====
=====

-- 2-1. 운영자 등급 (Operator Levels) -> [출입자에게 부여]
-- 역할: 사람이 가진 권한의 높이
CREATE TABLE operator_levels (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (U
    UUID v7)
        organization_id UUID, -- 소속 조직 ID
        name TEXT NOT NULL, -- 운영자 등급명 (예: 마스터(Master), 일반(Gen

```

```

al), 방문객(Visitor)
description TEXT,
-- [레벨 정의]
access_level INTEGER NOT NULL DEFAULT 0,      -- 출입 등급 (이 값 >
= 문의 임계값이면 통과)
antipassback_level INTEGER NOT NULL DEFAULT 0, -- 안티패스백 면제
등급
arming_level INTEGER NOT NULL DEFAULT 0,       -- 경비(Arming) 설
정/해제 가능 등급

is_active BOOLEAN NOT NULL DEFAULT TRUE,        -- 활성화 여부
created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성
일시
updated_at TIMESTAMPTZ                         -- 레코드 수정 일시
);

-- [PARTIAL UNIQUE INDEX]
CREATE UNIQUE INDEX idx_operator_levels_name ON operator_levels(n
ame)
WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_operator_levels_organization_id_name ON o
perator_levels(organization_id, name)
WHERE is_active = true AND organization_id IS NOT NULL;

-- [TRIGGER]
CREATE TRIGGER set_timestamp_operator_levels
BEFORE UPDATE ON operator_levels
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

-- 2-2. 출입 임계값 (Access Thresholds) -> [출입문에게 부여]
-- 역할: 문을 통과하기 위해 필요한 최소 조건
CREATE TABLE access_thresholds (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (U
UID v7)
    organization_id UUID,                            -- 소속 조직 ID

```

```

name TEXT NOT NULL, -- 출입 임계값명 (예: 보안구역 레벨, 일반구역 레벨)
description TEXT, -- 출입 임계값 설명

-- [요구 조건]
min_access_level INTEGER NOT NULL DEFAULT 0, -- 최소 출입 등급 (이 값보다 낮은 사람은 못 들어옴)
min_antipassback_level INTEGER NOT NULL DEFAULT 0, -- 안티패스백 면제 최소 등급
min_arming_level INTEGER NOT NULL DEFAULT 0, -- 경비(Arming) 설정/해제 가능 최소 등급

is_active BOOLEAN NOT NULL DEFAULT TRUE, -- 출입 임계값 등급 활성화 여부
created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
updated_at TIMESTAMPTZ -- 레코드 수정 일시
);

-- [PARTIAL UNIQUE INDEX]
CREATE UNIQUE INDEX idx_access_threshold_name ON access_thresholds(name)
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_access_threshold_organization_id_name ON access_thresholds(organization_id, name)
    WHERE is_active = true AND organization_id IS NOT NULL;

-- [TRIGGER]
CREATE TRIGGER set_timestamp_access_thresholds
BEFORE UPDATE ON access_thresholds
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

```

```

=====
=====
-- 3. 출입 주체 (Personnels) - [누가 (Who)]
-- 역할: 출입자 정보 및 개인별 보안 등급 할당
=====
=====
```

```

CREATE TABLE personnels (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)
    organization_id UUID, -- 소속 조직 ID

    employee_id UUID, -- 직원 ID

    -- [개인정보 암호화]
    -- 이름(name), 조직명(organization_name), 부서명(department), 사진(profile_image_url), 전화번호 등 모든 식별 정보는 암호화된 JSON으로 저장
    -- 방문 목적(visit_purpose), 접견자 정보(host_info) 등 민감할 수 있는 모든 정보를 암호화된 JSON으로 저장
    encrypted_data TEXT NOT NULL,

    -- [개인 보안 속성 할당]
    -- 이 사람이 가진 레벨(출입문의 access_threshold와 비교됨)
    operator_level_id UUID REFERENCES operator_levels(id),

    is_active BOOLEAN NOT NULL DEFAULT TRUE, -- 활성화 여부
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
    updated_at TIMESTAMPTZ -- 레코드 수정 일시
);

-- [PARTIAL UNIQUE INDEX]
CREATE INDEX idx_personnels_reference_id ON personnels(reference_id);

-- [INDEX]
CREATE INDEX idx_personnels_organization ON personnels(organization_id);
CREATE INDEX idx_personnels_reference ON personnels(reference_id);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_personnels
BEFORE UPDATE ON personnels
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

```

```

-- =====
=====

-- 4. 인증 규칙 및 수단 (Authentication) - [어떻게 (How) - 방법 및 수단]
-- 역할: 문을 열기 위한 규칙과 사용자가 가진 인증 매체 정의
-- =====
=====

-- 4-1. 인증 규칙 (Authentication Rules)
-- 역할: 문을 열기 위해 필요한 인증 수단의 조합 (예: 카드 + 비밀번호)
CREATE TABLE authentication_rules (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)
    organization_id UUID, -- 소속 조직 ID
    name TEXT NOT NULL, -- 인증 규칙명 (예: 카드 전용, 카드 + 지문(이중인증))
    description TEXT, -- 인증 규칙 설명
    -- [인증 모드]
    -- CARD, FINGERPRINT, FACE, PIN, MOBILE, QR 등 조합
    -- 예: 'CARD_OR_FACE', 'CARD_AND_PIN'
    auth_mode TEXT NOT NULL DEFAULT 'CARD_ONLY',
    is_antipassback BOOLEAN DEFAULT TRUE, -- 이 규칙 적용 시 안티패스백 검사 여부
    is_active BOOLEAN NOT NULL DEFAULT TRUE, -- 출입 주체 활성화 여부
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
    updated_at TIMESTAMPTZ -- 레코드 수정 일시
);

-- [PARTIAL UNIQUE INDEX]
CREATE UNIQUE INDEX idx_auth_rules_name ON authentication_rules(name)
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_auth_rules_organization_id_name ON auth

```

```

ntication_rules(organization_id, name)
WHERE is_active = true AND organization_id IS NOT NULL;

-- [TRIGGER]
CREATE TRIGGER set_timestamp_authentication_rules
BEFORE UPDATE ON authentication_rules
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

-- 4-2. 인증 수단 (Credentials)
-- 역할: 출입자가 가진 카드, 지문 등 물리적 매체 관리
CREATE TABLE credentials (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)
    organization_id UUID, -- 소속 조직 ID
    personnel_id UUID NOT NULL REFERENCES personnels(id),
    credential_type TEXT NOT NULL CHECK (credential_type IN ('RFID',
'FINGERPRINT', 'FACE', 'PIN', 'MOBILE', 'QR', 'LPR', 'IRIS')),
    encrypted_data TEXT NOT NULL,
    -- [상태 제한] ACTIVE, LOST(분실), EXPIRED(만료), SUSPENDED(정지)
    credential_status TEXT NOT NULL DEFAULT 'ACTIVE' CHECK (credential_status IN ('ACTIVE', 'LOST', 'EXPIRED', 'SUSPENDED')),

    is_active BOOLEAN NOT NULL DEFAULT TRUE, -- 활성화 여부
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
    updated_at TIMESTAMPTZ -- 레코드 수정 일시
);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_credentials
BEFORE UPDATE ON credentials
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

-- 4-3. 인증 수단 이력 (Credential Histories) - [이력 (Audit)]

```

```

CREATE TABLE credential_histories (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)
    organization_id UUID, -- 소속 조직 ID
    department_id UUID, -- 소속 부서 ID
    job_id UUID, -- 당시의 수행 직무 ID
    position_id UUID, -- 당시의 맡았던 직책 ID
    grade_id UUID, -- 당시의 직급 ID
    employee_id UUID, -- 직원 ID

    credential_id UUID NOT NULL, -- 원본 Credential ID
    personnel_id UUID NOT NULL, -- 소유자 ID

    -- [보안 강화] 통합 암호화 데이터 (AES-256)
    -- action_type(발급/폐기), action_reason(사유), actor_id(처리자), credential_type 등 상세 정보 통합
    -- 예: { "action": "REVOKE", "reason": "퇴사", "actor": "admin_uuid", "type": "RFID" }
    encrypted_snapshot_data JSONB DEFAULT '{}'::JSONB,
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
    updated_at TIMESTAMPTZ -- 레코드 수정 일시
);

-- [INDEX]
CREATE INDEX idx_credential_histories_credential_id ON credential_histories(credential_id);
CREATE INDEX idx_credential_histories_personnel_id ON credential_histories(personnel_id);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_credential_histories
BEFORE UPDATE ON credential_histories
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

```

```

-- =====
=====
-- 5. 출입문 설정 (Access Point Configs) - [문 속성 정의]
-- 역할: 각 출입문에 '보안 등급'과 '인증 규칙'을 할당 (문의 성격 정의)
-- =====
=====

CREATE TABLE access_point_configs (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)

    access_point_id UUID NOT NULL UNIQUE, -- 출입 포인트(출입문, 게이트 등) ID

    -- [입실(Entry) 설정: 들어올 때]
    entry_access_threshold_id UUID REFERENCES access_thresholds(id),
    -- 입실 시 필요한 최소 보안 등급
    entry_authentication_rule_id UUID REFERENCES authentication_rules(id), -- 입실 시 필요한 인증 방식

    -- [퇴실(Exit) 설정: 나갈 때]
    exit_access_threshold_id UUID REFERENCES access_thresholds(id),
    -- 퇴실 시 필요한 최소 보안 등급
    exit_authentication_rule_id UUID REFERENCES authentication_rules(id), -- 퇴실 시 필요한 인증 방식

    is_active BOOLEAN NOT NULL DEFAULT TRUE,      -- 활성화 여부
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
    updated_at TIMESTAMPTZ                      -- 레코드 수정 일시
);

-- [INDEX]
CREATE INDEX idx_access_point_configs_point ON access_point_configs(access_point_id);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_access_point_configs
BEFORE UPDATE ON access_point_configs

```

```
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();
```

```
-- =====
=====
-- 6. 공간 그룹 (Access Groups) - [어디를 (Where) - 그룹화]
-- 역할: 권한 부여 편의성을 위해 출입 포인트들을 그룹핑 (예: 1층 전체)
-- =====
=====
CREATE TABLE access_groups (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)
    organization_id UUID, -- 소속 조직 ID
    name TEXT NOT NULL, -- 출입 포인트(출입문, 게이트 등) 그룹명 (예: 서버 실 출입문 그룹)
    description TEXT, -- 출입 포인트(출입문, 게이트 등) 그룹 설명
    is_active BOOLEAN NOT NULL DEFAULT TRUE, -- 활성화 여부
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
    updated_at TIMESTAMPTZ -- 레코드 수정 일시
);
-- [PARTIAL UNIQUE INDEX]
CREATE UNIQUE INDEX idx_access_groups_name ON access_groups(organization_id, name) WHERE is_active = true;
-- [TRIGGER]
CREATE TRIGGER set_timestamp_access_groups
BEFORE UPDATE ON access_groups
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

CREATE TABLE access_group_items (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)
```

```

access_group_id UUID NOT NULL REFERENCES access_groups(id) O
N DELETE CASCADE,
access_point_id UUID NOT NULL, -- 출입 포인트(출입문, 게이트 등) ID

created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성
일시
updated_at TIMESTAMPTZ, -- 레코드 수정 일시

UNIQUE (access_group_id, access_point_id)
);

-- [INDEX]
CREATE INDEX idx_access_group_items_parent ON access_group_items
(access_group_id);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_access_group_items
BEFORE UPDATE ON access_group_items
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

```

```

=====
=====

-- 7. 출입 규칙 (Access Rules) - [정의 (Definition)]
-- 역할: 자주 사용하는 [어디를 + 언제] 조합 규칙 상세들을 하나로 묶은 규칙
-- 예: "신입사원 출입 규칙"
=====

-- 7-1. 규칙 헤더 (Access Rules)
CREATE TABLE access_rules (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (U
    UID v7)
    organization_id UUID, -- 소속 조직 ID

    name TEXT NOT NULL, -- 출입 규칙명
    description TEXT, -- 출입 설명

```

```

is_active BOOLEAN NOT NULL DEFAULT TRUE,      -- 활성화 여부
created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성
일시
updated_at TIMESTAMPTZ                      -- 레코드 수정 일시
);

-- [PARTIAL UNIQUE INDEX]
CREATE UNIQUE INDEX idx_access_rules_name ON access_rules(name)
WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_access_rules_organization_id_name ON access_rules(organization_id, name)
WHERE is_active = true AND organization_id IS NOT NULL;

-- [TRIGGER]
CREATE TRIGGER set_timestamp_access_rules
BEFORE UPDATE ON access_rules
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

-- 7-2. 규칙 상세 항목 (Access Rule Items)
-- 역할: 규칙을 구성하는 세부 항목들 (Where + When 조합)
-- 예: "서버실 그룹(Where) + 24시간 스케줄(When)"
-- 예: "화장실(Where) + 24시간 스케줄(When)"
CREATE TABLE access_rule_items (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)

    access_rule_id UUID NOT NULL REFERENCES access_rules(id) ON DELETE CASCADE, -- 출입 규칙 ID

    -- [Where] 어디를? (그룹 또는 개별 문)
    access_group_id UUID REFERENCES access_groups(id), -- 그룹 단위
    access_point_id UUID,                                -- 개별 단위

    -- [When] 언제?
    time_schedule_id UUID NOT NULL REFERENCES time_schedules(id),

```

```

    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성
일시
    updated_at TIMESTAMPTZ,                      -- 레코드 수정 일시

    CONSTRAINT check_target_exists CHECK (access_group_id IS NOT N
ULL OR access_point_id IS NOT NULL)
);

--- [INDEX]
CREATE INDEX idx_access_rule_items_parent ON access_rule_items(ac
cess_rule_id);

--- [PARTIAL UNIQUE INDEX]
CREATE UNIQUE INDEX idx_rule_items_group_unique ON access_rule_it
ems(access_rule_id, access_group_id, time_schedule_id)
    WHERE access_group_id IS NOT NULL;

CREATE UNIQUE INDEX idx_rule_items_point_unique ON access_rule_ite
ms(access_rule_id, access_point_id, time_schedule_id)
    WHERE access_point_id IS NOT NULL;

-- [TRIGGER]
CREATE TRIGGER set_timestamp_access_rule_items
BEFORE UPDATE ON access_rule_items
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

```

```

=====
=====
-- 8. 권한 부여 (Access Grants) - [할당 (Assign)]
-- 역할: 출입자에게 '규칙'을 부여하여 최종 권한 완성
=====
=====
CREATE TABLE access_grants (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (U
UID v7)

-- [출입 주체]

```

```

personnel_id UUID NOT NULL REFERENCES personnels(id) ON DELETE CASCADE,
-- [규칙 할당]
access_rule_id UUID NOT NULL REFERENCES access_rules(id) ON DELETE CASCADE,
created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
updated_at TIMESTAMPTZ, -- 레코드 수정 일시
UNIQUE (personnel_id, access_rule_id)
);
-- [INDEX]
CREATE INDEX idx_access_grants_personnel ON access_grants(personnel_id);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_access_grants
BEFORE UPDATE ON access_grants
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

=====
=====

-- 9. 권한 부여 이력 (Access Grants Histories) - [이력 (Audit)]
-- 역할: 누가 언제 어떤 권한을 받았고, 언제 회수되었는지 추적
=====

CREATE TABLE access_grant_histories (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)
    organization_id UUID, -- 소속 조직 ID
    personnel_id UUID NOT NULL, -- 출입 주체 ID
    access_rule_id UUID NOT NULL, -- 출입 규칙 ID
-- [보안 강화] 통합 암호화 데이터 (AES-256)

```

```

-- 사용자 이름, 규칙 이름, 처리자, 사유, 액션 타입(GRANT/REVOKE) 등은 모두
이안에 암호화된 JSON으로 저장
encrypted_snapshot_data JSONB DEFAULT '{}'::JSONB,
created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성
일시
updated_at TIMESTAMPTZ -- 레코드 수정 일시
);

-- [INDEX]
CREATE INDEX idx_grant_histories_personnel ON access_grant_histories(personnel_id);
CREATE INDEX idx_grant_histories_rule ON access_grant_histories(access_rule_id);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_access_grant_histories
BEFORE UPDATE ON access_grant_histories
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

```

```

=====
=====
-- 10. 인터락 (Interlocks) - [출입문 연동 제어]
-- 역할: 보안 강화를 위해 두 개 이상의 출입문 동작을 연동
=====
=====
CREATE TABLE interlocks (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)
    organization_id UUID, -- 소속 조직 ID
    name TEXT NOT NULL, -- 인터락명
    description TEXT, -- 인터락 설명
    interlock_type TEXT NOT NULL DEFAULT 'MANTRAP', -- 인터락 종류
    is_active BOOLEAN NOT NULL DEFAULT TRUE, -- 활성화 여부

```

```

    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성
    일시
    updated_at TIMESTAMPTZ           -- 레코드 수정 일시
);

-- [PARTIAL UNIQUE INDEX]
CREATE UNIQUE INDEX idx_interlocks_name_unique ON interlocks(nam
e)
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_interlocks_organization_name_unique ON in
terlocks(organization_id, name)
    WHERE is_active = true AND organization_id IS NOT NULL;

-- [TRIGGER]
CREATE TRIGGER set_timestamp_interlocks
BEFORE UPDATE ON interlocks
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

CREATE TABLE interlock_access_points (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (U
    ID v7)

    interlock_id UUID NOT NULL REFERENCES interlocks(id) ON DELETE
    CASCADE, -- 인터락 ID
    access_point_id UUID NOT NULL, -- 출입 포인트(출입문, 게이트 등) ID

    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성
    일시
    updated_at TIMESTAMPTZ,           -- 레코드 수정 일시

    UNIQUE (interlock_id, access_point_id)
);

-- [INDEX]
CREATE INDEX idx_interlock_aps_parent ON interlock_access_points(int
erlock_id);

```

```
-- [TRIGGER]
CREATE TRIGGER set_timestamp_interlock_access_points
BEFORE UPDATE ON interlock_access_points
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();
```

## 로그 서비스(Log Service)

```
-- =====
=====
-- 1. 출입 로그 (Access Logs)
-- 역할: 누가, 언제, 어디를, 어떻게, 무엇을, 왜 출입했는지 기록
-- =====
=====
CREATE TABLE access_logs (
    id UUID DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)
    organization_id UUID,             -- 소속 조직 ID
    department_id UUID,              -- 소속 부서 ID
    job_id UUID,                   -- 당시의 수행 직무 ID
    position_id UUID,              -- 당시의 맡았던 직책 ID
    grade_id UUID,                 -- 당시의 직급 ID
    employee_id UUID,              -- 직원 ID

    -- [검색용 핵심 식별자]
    personnel_id UUID,            -- 출입 주체 ID
    access_point_id UUID,         -- 출입문 ID
    device_id UUID,               -- 장비 ID
    credential_id UUID,           -- 인증 수단 ID

    -- [통합 데이터] 스냅샷 데이터 암호화 (AES-256)
    -- 부서명, 직급명, 사용자 이름, 결과 사유 등은 모두 이 안에 암호화된 JSON으
    --로 저장
    -- 예: { "user_name": "홍길동", "department": "개발팀", "result": "DENIE
    D", "reason": "권한없음" ... }
    encrypted_snapshot_data TEXT NOT NULL,
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성
```

일시

updated\_at TIMESTAMPTZ, -- 레코드 수정 일시

PRIMARY KEY (id, created\_at)  
 ) PARTITION BY RANGE(created\_at);

-- 파티션 키 및 주요 ID 인덱스

CREATE INDEX idx\_access\_logs\_organization ON access\_logs(organization\_id, created\_at DESC);

CREATE INDEX idx\_access\_logs\_personnel ON access\_logs(personnel\_id, created\_at DESC);

CREATE INDEX idx\_access\_logs\_device ON access\_logs(device\_id, created\_at DESC);

-- [TRIGGER]

CREATE TRIGGER set\_timestamp\_access\_logs  
BEFORE UPDATE ON access\_logs  
FOR EACH ROW  
EXECUTE FUNCTION update\_timestamp();

-- =====

=====

-- 2. 식수 로그 (Meal Logs)

-- 역할: 식수 태깅 기록

-- =====

=====

CREATE TABLE meal\_logs (

id UUID DEFAULT uuid\_generate\_v7(), -- 고유 식별자 (UUID v7)

organization\_id UUID, -- 소속 조직 ID

department\_id UUID, -- 소속 부서 ID

job\_id UUID, -- 당시의 수행 직무 ID

position\_id UUID, -- 당시의 맡았던 직책 ID

grade\_id UUID, -- 당시의 직급 ID

employee\_id UUID, -- 직원 ID

-- [검색용 핵심 식별자]

employee\_id UUID,

meal\_type\_id UUID,

```

device_id UUID
    -- [통합 데이터] 스냅샷 데이터 암호화 (AES-256)
    -- 당시 단가, 식사 인원, 총액 등은 모두 이 안에 암호화된 JSON으로 저장
    -- 예: { "meal_count": 1, "unit_price": 5000, "total": 5000, "employee_name": "김철수" ... }
    encrypted_snapshot_data TEXT NOT NULL,
        created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
        updated_at TIMESTAMPTZ, -- 레코드 수정 일시
            PRIMARY KEY (id, created_at)
        ) PARTITION BY RANGE(created_at);

-- 파티션 키 및 주요 ID 인덱스
CREATE INDEX idx_meal_logs_employee ON meal_logs(employee_id, created_at DESC);
CREATE INDEX idx_meal_logs_organization ON meal_logs(organization_id, created_at DESC);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_meal_logs
BEFORE UPDATE ON meal_logs
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

=====
=====

-- 3. 장비 상태 로그 (Device Status Logs)
-- 역할: 장비의 상태 변경 이력 (Online/Offline, Battery 등)
=====

CREATE TABLE device_status_logs (
    id UUID DEFAULT uuid_generate_v7(), -- 장비 상태 로그 고유 식별자 (UUID v7)
    organization_id UUID, -- 소속 조직 ID

```

```

-- [검색용 핵심 식별자]
device_id UUID NOT NULL, -- 장비 ID

-- [통합 데이터] 스냅샷 데이터 암호화 (AES-256)
-- IP 주소, 장비 상태 등은 모두 이 안에 암호화된 JSON으로 저장
-- 예: { "status": "OFFLINE", "battery": 20, "signal": -80 ... }
encrypted_snapshot_data TEXT NOT NULL,

created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성
일시
updated_at TIMESTAMPTZ, -- 레코드 수정 일시

PRIMARY KEY (id, created_at)
) PARTITION BY RANGE(created_at);

-- 파티션 키 및 주요 ID 인덱스
CREATE INDEX idx_device_status_device ON device_status_logs(device
_id, created_at DESC);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_device_status_logs
BEFORE UPDATE ON device_status_logs
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

=====
=====

-- 4. 장비 통신 로그 (Device Communication Logs)
-- 역할: 로우 레벨 패킷 데이터 기록 (디버깅용)
=====

CREATE TABLE device_communication_logs (
    id UUID DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)
    organization_id UUID, -- 소속 조직 ID

-- [검색용 핵심 식별자]
device_id UUID NOT NULL, -- 장비 ID

```

```

-- [통합 데이터] 통신 패킷 암호화 (AES-256)
encrypted_payload TEXT NOT NULL, -- 원본 패킷
encrypted_parsed_data TEXT, -- 파싱된 데이터

created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
updated_at TIMESTAMPTZ, -- 레코드 수정 일시

PRIMARY KEY (id, created_at)
) PARTITION BY RANGE(created_at);

-- 파티션 키 및 주요 ID 인덱스
CREATE INDEX idx_communication_logs_device ON device_communication_logs(device_id, created_at DESC);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_device_communication_logs
BEFORE UPDATE ON device_communication_logs
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

=====
=====

-- 5. 시스템 알람 로그 (System Alarm Logs)
-- 역할: 화재, 강제 개방 등 중요 알람 발생 및 조치 이력
=====

CREATE TABLE system_alarms_logs (
    id UUID DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)
    organization_id UUID, -- 소속 조직 ID

    -- [검색용 핵심 식별자]
    access_point_id UUID, -- 출입 포인트(출입문, 게이트 등) ID
    device_id UUID, -- 장비 ID
    area_id UUID, -- 구역
    actor_id UUID, -- 조치자 ID

    -- [통합 데이터] 스냅샷 데이터 암호화 (AES-256)

```

```

-- 알람 타입, 메시지, 조치 메모 등은 모두 이 안에 암호화된 JSON으로 저장
-- 예: { "alarm_type": "FIRE", "message": "1층 화재 감지", "action_note":
"현장 확인 완료" ... }
encrypted_snapshot_data TEXT NOT NULL,

created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성
일시
updated_at TIMESTAMPTZ, -- 레코드 수정 일시

PRIMARY KEY (id, created_at)
) PARTITION BY RANGE(created_at);

-- 파티션 키 및 주요 ID 인덱스
CREATE INDEX idx_alarms_organization ON system_alarms_logs(organiz
ation_id, created_at DESC);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_system_alarms_logs
BEFORE UPDATE ON system_alarms_logs
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

=====
=====

-- 6. 시스템 감사 로그 (System Audit Logs)
-- 역할: 관리자의 조작 행위 추적
=====

CREATE TABLE system_audit_logs (
    id UUID DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)
    organization_id UUID, -- 소속 조직 ID

    -- [검색용 핵심 식별자]
    actor_user_id UUID, -- 조작한 관리자 ID
    target_id UUID NOT NULL, -- 변경된 대상 객체의 ID

    -- [통합 데이터] 스냅샷 데이터 암호화 (AES-256)
    -- 관리자가 조회한 개인정보나 변경한 설정값(Before/After)은 모두 이 안에 암

```

```

호화된 JSON으로 저장
-- JSON 구조: { "action_type": "CREATE", "ip_address": "192.168.10.1",
"changes": "..." ... }
encrypted_snapshot_data TEXT NOT NULL,
created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성
일시
updated_at TIMESTAMPTZ, -- 레코드 수정 일시
PRIMARY KEY (id, created_at)
) PARTITION BY RANGE(created_at);

-- 파티션 키 및 주요 ID 인덱스
CREATE INDEX idx_audit_actor ON system_audit_logs(actor_user_id, cre
ated_at DESC);
CREATE INDEX idx_audit_target ON system_audit_logs(target_id);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_system_audit_logs
BEFORE UPDATE ON system_audit_logs
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

=====
=====

-- 7. 자동화 실행 로그 (Automation Logs)
-- 역할: 자동화 규칙(Rule Engine)의 실행 이력 및 결과 추적
=====

CREATE TABLE automation_logs (
    id UUID DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)
    organization_id UUID, -- 소속 조직 ID

    -- [검색용 핵심 식별자]
    event_rule_id UUID NOT NULL, -- 실행된 규칙 ID
    trigger_source_id UUID, -- 트리거를 유발한 원본 객체 ID (예: 센서 ID, 문
ID)

```

```

-- [통합 데이터]
-- trigger_event(무슨 이벤트로), action_results(수행 결과 목록), error_message,
execution_time 등은 모두 이 안에 암호화된 JSON으로 저장
-- 예: { "trigger": "FIRE_ALARM", "action": [{"type": "OPEN_DOOR", "status": "SUCCESS"}]}
encrypted_snapshot_data TEXT NOT NULL,

created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
updated_at TIMESTAMPTZ, -- 레코드 수정 일시

PRIMARY KEY (id, created_at)
) PARTITION BY RANGE(created_at);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_automation_logs
BEFORE UPDATE ON automation_logs
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

```

## 통계 및 분석 서비스(Aalytics Service)

```

=====
=====

-- 1. 출입 통계 (Access Statistics)
-- 역할: 출입 로그를 기반으로 한 개인별/조직별 출입 현황 요약
=====

-- 1-1. 일별 출입 요약 (Access Daily Summaries)
CREATE TABLE access_daily_summaries (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)
    organization_id UUID, -- 소속 조직 ID
    personnel_id UUID NOT NULL, -- 출입 주체 ID
    summary_date DATE NOT NULL, -- 요약 기준일 (YYYY-MM-DD)

```

```

-- [통합 통계 데이터]
-- 예: {
--   "total_count": 10, "success_count": 9, "failure_count": 1,
--   "first_access_at": "2023-10-01T08:50:00Z", "last_access_at": "202
3-10-01T18:10:00Z",
--   "door_stats": {"DOOR_01": 5, "GATE_02": 2},
--   "hourly_stats": {"09": 1, "18": 1}
-- }
details JSONB DEFAULT '{}'::JSONB,

created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성
일시
updated_at TIMESTAMPTZ, -- 레코드 수정 일시

UNIQUE (organization_id, personnel_id, summary_date)
);
-- [INDEX]
CREATE INDEX idx_access_daily_date ON access_daily_summaries(organization_id, summary_date);

-- [GIN INDEX]
CREATE INDEX idx_access_daily_summaries_details ON access_daily_su
mmaries USING GIN (details);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_access_daily_summaries
BEFORE UPDATE ON access_daily_summaries
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

-- 1-2. 월별 출입 요약 (Access Monthly Summaries)
CREATE TABLE access_monthly_summaries (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (U
UID v7)
    organization_id UUID, -- 소속 조직 ID
    personnel_id UUID NOT NULL, -- 출입 주체 ID

```

```

summary_month TEXT NOT NULL, -- 요약 기준월 (YYYY-MM)

-- [통합 통계 데이터]
-- 예: {
--   "total_days_visited": 20,
--   "total_access_count": 150,
--   "total_failure_count": 3,
--   "daily_trend": [10, 12, 11, ...]
-- }
details JSONB DEFAULT '{}'::JSONB,

created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
updated_at TIMESTAMPTZ, -- 레코드 수정 일시

UNIQUE (organization_id, personnel_id, summary_month)
);

-- [INDEX]
CREATE INDEX idx_access_monthly_date ON access_monthly_summaries(organization_id, summary_month);

-- [GIN INDEX]
CREATE INDEX idx_access_monthly_summaries_details ON access_monthly_summaries USING GIN (details);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_access_monthly_summaries
BEFORE UPDATE ON access_monthly_summaries
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

=====
=====

-- 2. 근태 통계 (Attendance Statistics)
-- 역할: 출입 기록을 가공하여 근무 시간, 지각, 조퇴 등을 판별한 결과
=====
```

```

-- 2-1. 일별 근태 요약 (Attendance Daily Summaries)
CREATE TABLE attendance_daily_summaries (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)
    organization_id UUID, -- 소속 조직 ID

    employee_id UUID NOT NULL, -- 직원 ID
    summary_date DATE NOT NULL, -- 요약 기준일 (YYYY-MM-DD)

    -- [통합 통계 데이터]
    -- 예: {
    --   "work_type": "FIXED", "attendance_status": "LATE",
    --   "check_in_at": "09:10:00", "check_out_at": "18:00:00",
    --   "work_seconds": 28800, "overtime_seconds": 0, "late_minutes": 1
    -- },
    --   "is_late": true, "is_early_leave": false,
    --   "time_segments": [{"type": "WORK", "start": "...", "end": "..."}]
    -- }
    details JSONB DEFAULT '{}'::JSONB,

    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
    updated_at TIMESTAMPTZ, -- 레코드 수정 일시

    UNIQUE (organization_id, employee_id, summary_date)
);

-- [INDEX]
CREATE INDEX idx_attendance_daily_date ON attendance_daily_summaries(organization_id, summary_date);

-- [GIN INDEX]
CREATE INDEX idx_attendance_daily_summaries_details ON attendance_daily_summaries USING GIN (details);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_attendance_daily_summaries

```

```

BEFORE UPDATE ON attendance_daily_summaries
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

-- 2-2. 월별 근태 요약 (Attendance Monthly Summaries)
CREATE TABLE attendance_monthly_summaries (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)
    organization_id UUID,                                -- 소속 조직 ID

    employee_id UUID NOT NULL, -- 직원 ID
    summary_month TEXT NOT NULL, -- 요약 기준월 (YYYY-MM)

    -- [통합 통계 데이터]
    -- 예: {
    --   "total_work_days": 20, "total_paid_days": 22.0,
    --   "total_work_seconds": 500000,
    --   "late_count": 1, "early_leave_count": 0, "absent_count": 0,
    --   "weekly_stats": [...]
    -- }
    details JSONB DEFAULT '{}'::JSONB,

    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
    updated_at TIMESTAMPTZ,                         -- 레코드 수정 일시

    UNIQUE (organization_id, employee_id, summary_month)
);

-- [INDEX]
CREATE INDEX idx_attendance_monthly_date ON attendance_monthly_summaries(organization_id, summary_month);

-- [GIN INDEX]
CREATE INDEX idx_attendance_monthly_summaries_details ON attendance_monthly_summaries USING GIN (details);

-- [TRIGGER]

```

```

CREATE TRIGGER set_timestamp_attendance_monthly_summaries
BEFORE UPDATE ON attendance_monthly_summaries
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

=====
-- 3. 식수 통계
-- 역할: 구내식당 이용 현황 및 정산 데이터
=====

-- 3-1. 일별 식수 요약 (Meal Daily Summaries)
CREATE TABLE meal_daily_summaries (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)
    organization_id UUID,                                -- 소속 조직 ID
    employee_id UUID NOT NULL, -- 직원 ID
    summary_date DATE NOT NULL, -- 요약 기준일 (YYYY-MM-DD)

    -- [통합 통계 데이터]
    -- 예: {
    --   "total_count": 2,
    --   "total_amount": 11000,
    --   "breakdown": {"LUNCH": 1, "DINNER": 1},
    --   "menu_names": ["제육볶음", "라면"]
    -- }
    details JSONB DEFAULT '{}'::JSONB,
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
    updated_at TIMESTAMPTZ,                          -- 레코드 수정 일시
    UNIQUE (organization_id, employee_id, summary_date)
);

-- [INDEX]

```

```

CREATE INDEX idx_meal_daily_date ON meal_daily_summaries(organization_id, summary_date);

-- [GIN INDEX]
CREATE INDEX idx_meal_daily_summaries_details ON meal_daily_summaries USING GIN (details);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_meal_daily_summaries
BEFORE UPDATE ON meal_daily_summaries
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

-- 3-2. 월별 식수 요약 (Meal Monthly Summaries)
CREATE TABLE meal_monthly_summaries (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)
    organization_id UUID,                            -- 소속 조직 ID
    employee_id UUID NOT NULL, -- 직원 ID
    summary_month TEXT NOT NULL, -- 요약 기준월 (YYYY-MM)

    -- [통합 통계 데이터]
    -- 예: { "total_count": 40, "total_amount": 220000, "daily_stats": [...] }
    details JSONB DEFAULT '{}'::JSONB,
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
    updated_at TIMESTAMPTZ,                         -- 레코드 수정 일시
    UNIQUE (organization_id, employee_id, summary_month)
);

-- [INDEX]
CREATE INDEX idx_meal_monthly_date ON meal_monthly_summaries(organization_id, summary_month);

-- [GIN INDEX]

```

```
CREATE INDEX idx_meal_monthly_summaries_details ON meal_monthly_
summaries USING GIN (details);
```

```
-- [TRIGGER]
CREATE TRIGGER set_timestamp_meal_monthly_summaries
BEFORE UPDATE ON meal_monthly_summaries
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();
```

## 근태 서비스(Attendance Service)

```
-- =====
=====
-- 1. 근태 정책 (Attendance Policies)
-- 역할: 조직별 근무 형태 및 규칙 정의 (예: 시차출퇴근제, 고정근무제)
-- =====
=====
CREATE TABLE attendance_policies (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (U
    ID v7)
    organization_id UUID, -- 소속 조직 ID
    name TEXT NOT NULL, -- 정책명 (예: 일반직군 표준근무, 연구직 탄력
    근무)
    description TEXT, -- 정책 설명
    attendance_policy_code TEXT, -- 정책 코드

    -- [근무 유형]
    -- FIXED: 고정 출퇴근 (09:00 ~ 18:00)
    -- FLEXIBLE: 유연 근무 (코어타임 존재)
    -- SHIFT: 교대 근무
    work_type TEXT NOT NULL DEFAULT 'FIXED' CHECK (work_type IN
    ('FIXED', 'FLEXIBLE', 'SHIFT')),

    -- [상세 규칙 - JSONB]
    -- 예: {
    --   "work_start_time": "09:00",
    --   "work_end_time": "18:00",
    --   "core_time": "09:00-13:00",
    --   "shift": "A"
    -- }
```

```

-- "work_end_time": "18:00",
-- "core_time_start": "10:00",
-- "core_time_end": "16:00",
-- "lunch_break_minutes": 60,
-- "late_grace_minutes": 10 (지각 유예 시간)
-- }
rules JSONB DEFAULT '{}'::JSONB,

is_default BOOLEAN DEFAULT FALSE,          -- 기본 정책 여부
is_active BOOLEAN NOT NULL DEFAULT TRUE,   -- 활성화 여부
created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
updated_at TIMESTAMPTZ                      -- 레코드 수정 일시
);

-- [PARTIAL UNIQUE INDEX]
CREATE UNIQUE INDEX idx_attendance_policies_name ON attendance_policies(name)
WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_attendance_policies_organization_id_name
ON attendance_policies(organization_id, name)
WHERE is_active = true AND organization_id IS NOT NULL;

CREATE UNIQUE INDEX idx_attendance_policies_default_unique ON attendance_policies((1))
WHERE is_default = true AND is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_attendance_policies_organization_default_unique
ON attendance_policies(organization_id)
WHERE is_default = true AND is_active = true AND organization_id IS NOT NULL;

-- [GIN INDEX]
CREATE INDEX idx_attendance_policies_rules ON attendance_policies USING GIN (rules);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_attendance_policies

```

```

BEFORE UPDATE ON attendance_policies
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

=====
=====

-- 2. 휴가 유형 (Leave Types)
-- 역할: 연차, 반차, 병가, 경조사 등 휴가 종류 정의
=====

=====

CREATE TABLE leave_types (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)
    organization_id UUID,                                -- 소속 조직 ID

    name TEXT NOT NULL,     -- 휴가명 (예: 연차, 오전반차)
    description TEXT,      -- 휴가 설명
    leave_type_code TEXT, -- 휴가 타입 코드

    -- [휴가 속성 통합 - JSONB]
    -- 유급 여부, 차감 단위, 승인 필요 여부, 이월 가능 여부 등을 유연하게 저장
    -- 예: {
        -- "is_paid": true,
        -- "deduction_day": 0.5,
        -- "requires_approval": true,
        -- "allow_negative_balance": false
    -- }
    config JSONB DEFAULT '{}'::JSONB,

    is_active BOOLEAN NOT NULL DEFAULT TRUE,      -- 활성화 여부
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
    updated_at TIMESTAMPTZ                         -- 레코드 수정 일시
);

-- [PARTIAL UNIQUE INDEX]
CREATE UNIQUE INDEX idx_leave_type_name ON leave_types(name)
    WHERE is_active = true AND organization_id IS NULL;

```

```

CREATE UNIQUE INDEX idx_leave_type_organization_id_name ON leave
_types(organization_id, name)
    WHERE is_active = true AND organization_id IS NOT NULL;

-- [GIN INDEX]
CREATE INDEX idx_leave_types_config ON leave_types USING GIN (con
fig);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_leave_types
BEFORE UPDATE ON leave_types
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

=====
=====

-- 3. 직원 휴가 내역 (Employee Leaves)
-- 역할: 휴가 신청의 '현재 상태' 관리
=====

CREATE TABLE employee_leaves (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (U
    UUID v7)
    organization_id UUID,                                -- 소속 조직 ID

    -- [직원 참조]
    employee_id UUID NOT NULL,
    -- [휴가 유형 참조]
    leave_type_id UUID NOT NULL REFERENCES leave_types(id),

    -- [보안 강화] 통합 암호화 데이터 (AES-256)
    -- 날짜(start_date, end_date), 상태(status), 사유(reason),
    -- 반차여부(is_half_day), 승인자(approver_id), 승인일시(approved_at) 등
    -- 식별자를 제외한 모든 상세 정보를 JSON으로 묶어 암호화 저장
    encrypted_data TEXT NOT NULL,

    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성
    일시

```

```

    updated_at TIMESTAMPTZ          -- 레코드 수정 일시
);

-- [INDEX]
CREATE INDEX idx_employee_leaves_employee ON employee_leaves(employee_id);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_employee_leaves
BEFORE UPDATE ON employee_leaves
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

=====
=====

-- 4. 직원 휴가 이력 (Employee Leave Histories) - [이력 (Audit)]
-- 역할: 휴가 신청, 승인, 반려, 취소 등의 상태 변경 이력 추적
=====

CREATE TABLE employee_leave_histories (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)
    organization_id UUID,                            -- 소속 조직 ID
    department_id UUID,                            -- 소속 부서 ID
    job_id UUID,                                 -- 당시의 수행 직무 ID
    position_id UUID,                            -- 당시의 맡았던 직책 ID
    grade_id UUID,                             -- 당시의 직급 ID
    employee_id UUID NOT NULL,                  -- 직원 ID
    employee_leave_id UUID NOT NULL,            -- 원본 휴가 신청 ID

    -- [보안 강화] 스냅샷 데이터 암호화 (AES-256)
    -- 휴가 종류, 기간, 사유 상태 등 모든 정보를 JSON으로 묶어 암호화
    encrypted_snapshot_data TEXT NOT NULL,
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
    updated_at TIMESTAMPTZ          -- 레코드 수정 일시
);

```

```
-- [INDEX]
CREATE INDEX idx_leave_histories_leave ON employee_leave_histories
(employee_leave_id);
CREATE INDEX idx_leave_histories_employee ON employee_leave_histories
(employee_id);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_employee_leave_histories
BEFORE UPDATE ON employee_leave_histories
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();
```

## 식사 서비스(Meal Service)

```
-- =====
=====

-- 1. 식사 구분 (Meal Types)
-- 역할: 식사의 종류, 단가, 운영 시간 정의 (예: 조식, 중식A, 중식B)
-- =====
=====

CREATE TABLE meal_types (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)
    organization_id UUID, -- 소속 조직 ID
    name TEXT NOT NULL, -- 식사 종류명 (예: 조식, 중식-한식, 중식-일품, 석식)
    description TEXT, -- 식사 종류 설명
    meal_type_code TEXT, -- 식사 종류 코드

    -- [단가 정보]
    unit_price NUMERIC(19, 4) NOT NULL DEFAULT 0, -- 식대 단가
    currency TEXT DEFAULT 'KRW', -- 통화

    -- [운영 시간]
    -- 해당시간에만 태깅 허용 또는 해당 시간 태깅 시 이 식사 태입으로 자동 분류
```

```

serve_start_time TIME NOT NULL, -- 배식 시작 (예: 11:30)
serve_end_time TIME NOT NULL, -- 배식 종료 (예: 13:30)

is_active BOOLEAN NOT NULL DEFAULT TRUE, -- 활성화 여부
created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
updated_at TIMESTAMPTZ -- 레코드 수정 일시
);

-- [PARTIAL UNIQUE INDEX]
CREATE UNIQUE INDEX idx_meal_types_name ON meal_types(name)
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_meal_types_organization_id_name ON meal_types(organization_id, name)
    WHERE is_active = true AND organization_id IS NOT NULL;

CREATE UNIQUE INDEX idx_meal_types_meal_type_code ON meal_types(meal_type_code )
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_meal_types_organization_id_meal_type_code ON meal_types(organization_id, meal_type_code)
    WHERE is_active = true AND organization_id IS NOT NULL;

-- [TRIGGER]
CREATE TRIGGER set_timestamp_meal_types
BEFORE UPDATE ON meal_types
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

=====

-- 2. 식단표 (Weekly Memus)
-- 역할: 날짜별, 식사 종류별 메뉴 정보 제공
=====

CREATE TABLE weekly_menus (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)

```

```

organization_id UUID,          -- 소속 조직 ID

-- [식사 참조]
meal_type_id UUID NOT NULL REFERENCES meal_types(id),

serve_date DATE NOT NULL, -- 제공 일자

-- [메뉴 정보 통합 - JSONB]
-- 메뉴명, 반찬 구성, 칼로리, 이미지, 알러지 정보 등을 유연하게 저장
-- 예: {
--   "main_dish": "제육볶음",
--   "side_dishes": ["쌀밥", "미역국", "계란말이", "김치"],
--   "calories": 850,
--   "image_url": "https://s3...",
--   "allergens": ["PORK", "SOYBEAN"]
-- }
menu_info JSONB DEFAULT '{}'::JSONB,

is_active BOOLEAN NOT NULL DEFAULT TRUE,    -- 활성화 여부
created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성
일시
updated_at TIMESTAMPTZ           -- 레코드 수정 일시
);

-- [PARTIAL UNIQUE INDEX]
CREATE UNIQUE INDEX idx_weekly_menus_meal ON weekly_menus(meal_type_id, serve_date)
WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_weekly_menus_organization_meal ON weekly_menus(organization_id, meal_type_id, serve_date)
WHERE is_active = true AND organization_id IS NOT NULL;

-- [GIN INDEX]
CREATE INDEX idx_weekly_menus_menu_info ON weekly_menus USING
GIN (menu_info);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_weekly_menus

```

```
BEFORE UPDATE ON weekly_menus
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();
```

## 자동화 서비스(Automation Service)

```
-- =====
=====
-- [메타 데이터] 시스템 정의 타입 (System Definitions Types)
-- 역할: 자동화 규칙을 구성할 때 사용할 수 있는 '블록'들을 정의합니다.
-- 이 테이블들은 주로 시스템 초기화 시 Seed Data로 들어갑니다.
-- =====
=====

-- 1. 시스템 객체 타입 (System Objects Types)
-- 역할: 자동화의 대상이 되는 객체의 종류 (예: DOOR, USER, DEVICE, ZONE,
TIME_SCHEDULE)
CREATE TABLE system_object_types (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)

        name TEXT NOT NULL UNIQUE,           -- 시스템 객체 탑입명 (예: 출
        입 지점, 사용자)
        description TEXT,                  -- 시스템 객체 탑입 설명
        system_object_type_code TEXT UNIQUE, -- 시스템 객체 탑입 코드
        (예: ACCESS_POINT, USER, DEVICE)

        is_active BOOLEAN NOT NULL DEFAULT TRUE,   -- 활성화 여부
        created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성
        일시
        updated_at TIMESTAMPTZ                 -- 레코드 수정 일시
);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_system_object_types
BEFORE UPDATE ON system_object_types
FOR EACH ROW
```

```

EXECUTE FUNCTION update_timestamp();

-- 2. 이벤트 트리거 타입 (Event Triggers Types)
-- 역할: "무슨 일이 일어났을 때?"에 해당하는 이벤트 종류 (예: ACCESS_DENIED, FIRE_ALARM)
CREATE TABLE event_trigger_types (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)

        name TEXT NOT NULL UNIQUE,           -- 이벤트 트리거 타입명 (예: 강제 개방, 화재 감지)
        description TEXT,                  -- 이벤트 트리거 타입 설명
        event_trigger_type_code TEXT UNIQUE, -- 이벤트 트리거 타입 코드 (예: ACCESS_DENIED, FIRE_ALARM)

        is_active BOOLEAN NOT NULL DEFAULT TRUE,   -- 활성화 여부
        created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
        updated_at TIMESTAMPTZ                 -- 레코드 수정 일시
);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_event_trigger_types
BEFORE UPDATE ON event_trigger_types
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

-- 3. 이벤트 액션 (Event Action Types)
-- 역할: "무엇을 할 것인가?"에 해당하는 동작 종류 (예: OPEN_DOOR, SEND_EMAIL, LOCK_DOWN)
CREATE TABLE event_action_types (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)

        name TEXT NOT NULL UNIQUE,           -- 이벤트 액션 타입명 (예: 장비 제어, 사용자 알림)
        description TEXT,                  -- 이벤트 액션 타입 설명
        event_action_type_code TEXT UNIQUE, -- 이벤트 액션 타입 코드

```

(예: CONTROL\_DEVICE, NOTIFY\_USER)

```
-- [파라미터 스키마]
-- 예:
-- {
--   "type": "object",
--   "properties": {
--     "message": { "type": "string", "title": "알림 메시지" },
--     "priority": { "type": "string", "enum": ["HIGH", "LOW"], "default": "LOW" }
--   },
--   "required": ["message"]
-- }
parameter_schema JSONB DEFAULT '{}'::JSONB,  
  
    is_active BOOLEAN NOT NULL DEFAULT TRUE,      -- 활성화 여부
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성
    일시
    updated_at TIMESTAMPTZ                      -- 레코드 수정 일시
);  
  
-- [GIN INDEX]
CREATE INDEX idx_event_action_types_parameter_schema ON event_ac
tion_types USING GIN (parameter_schema);  
  
-- [TRIGGER]
CREATE TRIGGER set_timestamp_event_action_types
BEFORE UPDATE ON event_action_types
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();  
  
=====  
=====  
-- [규칙 데이터] 사용자 정의 자동화 (User Defined Rules)
-- 역할: 각 조직(Tenant)이 설정한 자동화 시나리오
=====
```

```

-- 4. 자동화 규칙 헤더 (Event Rules)
-- 역할: 하나의 자동화 시나리오 정의 (예: "화재 발생 시 비상 개방")
CREATE TABLE event_rules (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)
    organization_id UUID, -- 소속 조직 ID

    name TEXT NOT NULL, -- 이벤트 규칙명
    description TEXT, -- 이벤트 규칙 설명

    -- [실행 조건: 언제?]
    -- 이 스케줄에 해당하는 시간에만 규칙이 활성화됨 (NULL이면 항상 활성)
    time_schedule_id UUID,

    priority INTEGER DEFAULT 0, -- 우선순위 (여러 규칙 충돌 시)
    is_enabled BOOLEAN DEFAULT TRUE, -- 규칙 사용 여부 (일시 정지 등)

    is_active BOOLEAN NOT NULL DEFAULT TRUE, -- 활성화 여부
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
    updated_at TIMESTAMPTZ -- 레코드 수정 일시
);

-- [PARTIAL UNIQUE INDEX]
CREATE UNIQUE INDEX idx_event_rules_name ON event_rules(name)
    WHERE is_active = true AND organization_id IS NULL;
CREATE UNIQUE INDEX idx_event_rules_organization_id_name ON event_rules(organization_id, name)
    WHERE is_active = true AND organization_id IS NOT NULL;

-- [TRIGGER]
CREATE TRIGGER set_timestamp_event_rules
BEFORE UPDATE ON event_rules
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

-- 5. 규칙 조건 (Rule Conditions) - [IF]
-- 역할: 규칙이 발동되기 위한 조건 정의 (AND 조건으로 연결됨)

```

```

-- 예: "본관(Source)에서" + "화재(Trigger)가 발생하면"
CREATE TABLE event_rule_conditions (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)
    event_rule_id UUID NOT NULL REFERENCES event_rules(id),
    -- [무엇에서?] 이벤트 발생원 (Source)
    trigger_source_type_id UUID NOT NULL REFERENCES system_object_types(id), -- 시스템 객체 타입 ID
    trigger_source_id UUID, -- 특정 객체 ID (NULL이면 해당 타입 전체)
    -- [무슨 일이?] 이벤트 종류
    trigger_event_type_id UUID NOT NULL REFERENCES event_trigger_types(id), -- 이벤트 트리거 타입 ID
    -- [상세 조건]
    condition_operator TEXT DEFAULT 'EQUALS' CHECK (condition_operator IN ('EQUALS', 'NOT_EQUALS', 'GT', 'LT', 'CONTAINS')),
    -- 예: { "current_temp": 55, "humidity": 30 }
    condition_value JSONB DEFAULT '{}'::JSONB,
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
    updated_at TIMESTAMPTZ -- 레코드 수정 일시
);

-- [INDEX]
CREATE INDEX idx_rule_conditions_rule ON event_rule_conditions(event_rule_id);

-- [GIN INDEX]
CREATE INDEX idx_event_rule_conditions_condition_value ON event_rule_conditions USING GIN (condition_value);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_event_rule_conditions
BEFORE UPDATE ON event_rule_conditions

```

```

FOR EACH ROW
EXECUTE FUNCTION update_timestamp();

-- 규칙 실행 (Rule Actions) - [THEN]
-- 역할: 조건 만족 시 수행할 동작 정의 (순차 실행)
-- 예: "모든 문을 열고(Action1)" + "관리자에게 알림 전송(Action2)"
CREATE TABLE event_rule_actions (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v7(), -- 고유 식별자 (UUID v7)
    event_rule_id UUID NOT NULL REFERENCES event_rules(id),
    -- [어디에/누구에게?] 대상 (Target)
    action_target_type_id UUID NOT NULL REFERENCES system_object_types(id), -- 시스템 객체 타입 ID
    action_target_id UUID, -- NULL이면 해당 타입 전체 또는 payload에서 지정
    -- [무엇을?] 동작 종류
    action_type_id UUID NOT NULL REFERENCES event_action_types(id), -- 이벤트 액션 타입 ID
    -- [어떻게?] 파라미터
    -- 예: { "duration": 10, "message": "Fire Detected!" }
    action_metadata JSONB DEFAULT '{}'::JSONB,
    sort_order INTEGER DEFAULT 0, -- 실행 순서
    delay_seconds INTEGER DEFAULT 0, -- 지연 실행 (예: 5초 후 실행)
    created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(), -- 레코드 생성 일시
    updated_at TIMESTAMPTZ -- 레코드 수정 일시
);

-- [INDEX]
CREATE INDEX idx_rule_actions_rule ON event_rule_actions(event_rule_id);

-- [GIN INDEX]
CREATE INDEX idx_event_rule_actions_action_metadata ON event_rule_

```

```
actions USING GIN (action_metadata);

-- [TRIGGER]
CREATE TRIGGER set_timestamp_event_rule_actions
BEFORE UPDATE ON event_rule_actions
FOR EACH ROW
EXECUTE FUNCTION update_timestamp();
```