

IT-Asset-Register

사내 IT 자산 대여·반납·검수·통합관리 프로세스 DB 설계

데이터 설계

1) 설계 목표

- 자산 대여/반납/검수/수리 흐름을 DB로 표준화
- due_date 경과 기반 반납관리(요구/경고/제재)를 로그로 남겨 누락/지연을 줄임
- “현재 상태(스냅샷)”와 “변경 근거(이력 로그)”를 분리해 추적/감사 가능하게 설계

2) 설계 핵심 원칙

- 상태 분리: 자산 업무 상태(assets) vs 정책/제재 상태(user_policy_state)
- 이력 우선: 상태 변경/정책 이벤트는 반드시 로그 테이블로 근거 고정
- 무결성 강화: 검수 없이 상태 복귀 금지, 근거 없이 제재 상승 금지
- 증빙 분리: 검수 증빙(파일/사진/메모)은 별도 테이블로 관리
- 용어 통일: 사용자=employees, 부서=departments, 정책/알림=policy_rules/notices

3) 핵심 엔티티(테이블 그룹)

- 조직/사용자
 - departments(부서), employees(직원/사용자)
- 자산(마스터/상태 이력)
 - asset_categories(자산 분류), assets(자산), asset_status_history(자산 상태 전이 이력)
- 대여/출고 및 변경 이력
 - assignments(대여/배정 계약), checkout_logs(출고 이력), checkin_logs(반납예정일 변경 등 이력)
- 반납/검수/수리
 - return_requests(반납 요청), inspections(검수), inspection_results(검수 결과 코드), inspection_evidences(검수 증빙), repairs(수리)
- 반납관리(정책/알림/누적/이력)
 - policy_rules(규칙), notices(알림), user_policy_state(누적 스냅샷), policy_event_logs(정책 이벤트 이력)

4) 상태 설계(업무 상태 vs 정책 상태)

- 자산 상태코드(assets.asset_status)
 - AVAILABLE → ASSIGNED → OUT → RETURN_REQUESTED → INSPECTING → (UNDER_REPAIR) → AVAILABLE
 - DISPOSED는 종결 상태
- 정책/제재 상태(user_policy_state)
 - overdue_count / warning_count / restriction_level 로 누적·제재만 관리
- 분리 이유
 - 자산의 사용 흐름(OUT 유지)과 사용자 제재(경고/제재)는 별개라 모델을 분리해야 운영/추적이 명확함

5) 이력/감사 로그 전략(재현성 확보)

- asset_status_history
 - asset_id, from_status, to_status, changed_by, changed_at 기반으로 상태 전이 근거 저장
- checkin_logs
 - assignment_id, before_due_date, after_due_date, employee_id, change_reason 기반으로 due_date 변경 근거 저장
- policy_event_logs
 - policy_id, employee_id, assignment_id, event_type, created_at, detail 기반으로 정책 판단/제재 근거 저장
- notices
 - notice_type, message, is_read로 요구/경고/제재 알림의 생성/확인 상태까지 기록

6) 무결성/감점 포인트 방지 규칙(설계 규칙)

- 상태 역전 방지
 - 검수(INSPECTING) 없이 AVAILABLE 복귀 금지
 - 수리 근거 없이 UNDER_REPAIR 전환 금지(검수 결과/수리 레코드 기반)
- 제재 근거 누락 방지
 - 요구/경고 로그 없이 restriction_level 상승 금지
 - 예외/수동 처리도 policy_event_logs에 사유/주체/시각을 남겨 근거 단절 방지
- 카디널리티 주의
 - inspection_evidences는 다건 첨부(1:N) 전제
 - return_requests 재요청/취소 정책에 따라 UNIQUE 제약 적용 여부를 조정

7) 업무 흐름 → 테이블 매핑(요약)

- 자산 등록/관리: asset_categories → assets
- 대여 시작: assignments + checkout_logs + assets.asset_status=OUT
- 반납 요청: return_requests +
assets.asset_status=RETURN_REQUESTED
- 검수/증빙/수리: inspections + inspection_evidences + (repairs)
- 반납관리 자동화: policy_rules 기준으로 due_date 경과 판단 →
notices 생성
→ user_policy_state 누적 갱신 → policy_event_logs로 이벤트 근거 고정