# Additional Asset Tables Documentation

## Overview

This document describes the four additional asset-related database tables that extend the core asset management functionality: asset\_operational\_costs, asset\_risk\_mapping, asset\_tags, and system\_assets.

## Database Tables

### 1. Asset Operational Costs (asset\_operational\_costs)

**Purpose**: Track monthly operational expenses for each asset across different cost categories.

**Schema**:

CREATE TABLE asset\_operational\_costs (  
 id SERIAL PRIMARY KEY,  
 year\_month VARCHAR(10) NOT NULL, -- Format: 'YYYY-MM-DD'  
 power\_cost NUMERIC(10,2) DEFAULT 0, -- Monthly power consumption costs  
 space\_cost NUMERIC(10,2) DEFAULT 0, -- Rack space, office space costs  
 network\_cost NUMERIC(10,2) DEFAULT 0, -- Network bandwidth, connectivity costs  
 storage\_cost NUMERIC(10,2) DEFAULT 0, -- Storage, backup costs  
 labor\_cost NUMERIC(10,2) DEFAULT 0, -- Maintenance, support labor costs  
 other\_costs NUMERIC(10,2) DEFAULT 0, -- Miscellaneous operational costs  
 notes TEXT, -- Additional notes  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 asset\_uuid UUID REFERENCES assets(asset\_uuid) ON DELETE CASCADE  
);

**Use Cases**: - Monthly cost tracking and budgeting - Total Cost of Ownership (TCO) calculations - Cost trend analysis and forecasting - Department cost allocation - Operational efficiency metrics

**Sample Data Generated**: - 12 months of historical data for each asset - Realistic cost ranges based on asset type: - Servers: $500-1,500/month total operational costs - Workstations: $150-400/month - Laptops: $50-150/month - Network equipment: $200-800/month

### 2. Asset Risk Mapping (asset\_risk\_mapping)

**Purpose**: Map assets to risk models and cost centers for risk assessment and financial tracking.

**Schema**:

CREATE TABLE asset\_risk\_mapping (  
 id SERIAL PRIMARY KEY,  
 asset\_uuid UUID REFERENCES assets(asset\_uuid) ON DELETE CASCADE,  
 existing\_asset\_id INTEGER, -- Reference to legacy asset ID  
 risk\_model\_id INTEGER, -- Risk model identifier  
 cost\_center\_id INTEGER, -- Cost center for financial tracking  
 mapping\_confidence NUMERIC(3,2), -- Confidence level (0.00-1.00)  
 mapping\_method VARCHAR(50), -- How mapping was determined  
 mapping\_criteria TEXT, -- Criteria used for mapping  
 verified\_by INTEGER, -- User ID who verified mapping  
 verified\_at TIMESTAMP, -- When mapping was verified  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);

**Mapping Methods**: - agent\_based: Determined by agent data - scan\_based: Determined by vulnerability scans - manual: Manually assigned by administrator - automatic: System-generated mapping - hybrid: Combination of methods

**Risk Model Assignment**: - High criticality assets: Risk models 101-105 - Medium criticality assets: Risk models 201-210 - Low criticality assets: Risk models 301-315 - Unknown criticality: Risk models 401-405

**Cost Center Assignment**: - SYS-001 (Infrastructure): Cost center 1001 - SYS-002 (Development): Cost center 1002 - SYS-003 (Production): Cost center 1003 - SYS-004 (Security): Cost center 1004 - SYS-005 (Cloud): Cost center 1005 - Default: Cost center 1000

### 3. Asset Tags (asset\_tags)

**Purpose**: Flexible tagging system for asset categorization, filtering, and management.

**Schema**:

CREATE TABLE asset\_tags (  
 id SERIAL PRIMARY KEY,  
 asset\_uuid UUID REFERENCES assets(asset\_uuid) ON DELETE CASCADE,  
 tag\_key VARCHAR(255) NOT NULL, -- Tag category/key  
 tag\_value VARCHAR(255) NOT NULL, -- Tag value  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);

**Standard Tag Categories Generated**:

1. **Environment Tags**:
   * infrastructure, development, production, security, cloud, general
2. **Asset Type Tags**:
   * server, workstation, laptop, printer, scanner, firewall, switch, ups, other
3. **Criticality Tags**:
   * high, medium, low, unknown
4. **Location Tags**:
   * datacenter-east, datacenter-west, office-hq, office-branch
5. **Department Tags**:
   * engineering, data, operations, general, network, infrastructure
6. **Compliance Tags**:
   * sox-compliant, pci-dss, fisma-moderate, standard

**Use Cases**: - Asset filtering and search - Compliance tracking - Automated policy application - Reporting and analytics - Inventory management

### 4. System Assets (system\_assets)

**Purpose**: Many-to-many relationship between systems and assets, allowing assets to belong to multiple systems.

**Schema**:

CREATE TABLE system\_assets (  
 id SERIAL PRIMARY KEY,  
 system\_id VARCHAR(50) NOT NULL, -- System identifier  
 asset\_uuid UUID REFERENCES assets(asset\_uuid) ON DELETE CASCADE,  
 assignment\_type VARCHAR(20) DEFAULT 'direct', -- Type of assignment  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);

**Assignment Types**: - primary: Asset’s main system assignment - secondary: Asset supports additional systems - shared: Asset is shared across multiple systems - direct: Direct assignment (default)

**Assignment Logic**: 1. **Primary Assignments**: Each asset gets assigned to its main system 2. **Secondary Assignments**: Database servers may support multiple applications 3. **Shared Assignments**: Network infrastructure (firewalls, switches, DNS, DHCP) shared across systems

## Data Population

### Running the Seed Script

Execute the SQL script to populate all tables:

# In DBeaver or your PostgreSQL client  
-- Run the script: api/scripts/seed-additional-asset-tables.sql

### Expected Data Volume

After running the script, you should have: - **Asset Operational Costs**: ~300-500 records (12 months × 25-40 assets) - **Asset Risk Mapping**: ~25-40 records (one per asset) - **Asset Tags**: ~150-240 records (6 tags per asset) - **System Assets**: ~40-80 records (primary + secondary + shared assignments)

## API Integration

### Drizzle Schema Integration

The schemas are defined in api/src/db/schema/assetManagement.js:

// Import the new schemas  
const {   
 assetOperationalCosts,   
 assetRiskMapping,   
 assetTags,   
 systemAssets   
} = require('./db/schema/assetManagement');

### Recommended API Endpoints

1. **Asset Operational Costs**:
   * GET /api/v1/asset-management/operational-costs?assetUuid={uuid}
   * POST /api/v1/asset-management/operational-costs
   * PUT /api/v1/asset-management/operational-costs/{id}
   * DELETE /api/v1/asset-management/operational-costs/{id}
2. **Asset Risk Mapping**:
   * GET /api/v1/asset-management/risk-mapping?assetUuid={uuid}
   * POST /api/v1/asset-management/risk-mapping
   * PUT /api/v1/asset-management/risk-mapping/{id}
3. **Asset Tags**:
   * GET /api/v1/asset-management/tags?assetUuid={uuid}
   * POST /api/v1/asset-management/tags
   * DELETE /api/v1/asset-management/tags/{id}
   * GET /api/v1/asset-management/tags/keys (get all tag keys)
   * GET /api/v1/asset-management/tags/values?key={tagKey} (get values for a key)
4. **System Assets**:
   * GET /api/v1/asset-management/system-assets?systemId={id}
   * GET /api/v1/asset-management/system-assets?assetUuid={uuid}
   * POST /api/v1/asset-management/system-assets
   * DELETE /api/v1/asset-management/system-assets/{id}

## Frontend Integration

### Asset Inventory Enhancements

1. **Operational Costs Panel**: Similar to Cost Management panel
2. **Risk Information Display**: Show risk model and confidence
3. **Tag Management**: Add/remove tags, filter by tags
4. **System Relationships**: Show which systems an asset belongs to

### New UI Components Needed

1. **Asset Operational Costs Panel**:
   * Monthly cost breakdown charts
   * Cost trend analysis
   * Budget vs actual comparisons
2. **Asset Tags Component**:
   * Tag cloud display
   * Tag filtering interface
   * Bulk tag operations
3. **Risk Assessment Panel**:
   * Risk score visualization
   * Confidence indicators
   * Risk model details
4. **System Relationships Panel**:
   * System assignment visualization
   * Assignment type indicators
   * Multi-system dependencies

## Analytics and Reporting

### Cost Analytics

* Monthly operational cost trends
* Cost per asset type analysis
* Department cost allocation
* Budget variance reporting

### Risk Analytics

* Risk distribution across assets
* Confidence level analysis
* Risk model effectiveness
* Compliance risk assessment

### Tag Analytics

* Tag usage statistics
* Asset categorization reports
* Compliance tag tracking
* Location-based analysis

### System Analytics

* Asset distribution across systems
* System dependency mapping
* Shared resource utilization
* System-specific cost analysis

## Next Steps

1. **Run the SQL script** to populate the tables with sample data
2. **Implement API endpoints** for each table
3. **Create frontend panels** for data visualization and management
4. **Add filtering and search** capabilities using tags
5. **Implement analytics dashboards** for cost and risk analysis
6. **Add bulk operations** for tag management and system assignments

This foundation provides comprehensive asset management capabilities with operational cost tracking, risk assessment, flexible tagging, and system relationship management.