# Asset Management Database Schemas

## Overview

This document describes the asset management database schemas that extend the core asset tracking with comprehensive cost management, lifecycle tracking, operational costs, and risk mapping capabilities.

## Tables

### 1. asset\_cost\_management

Tracks all financial aspects of asset management including purchases, leases, maintenance, and operational costs.

#### Schema

CREATE TABLE asset\_cost\_management (  
 id SERIAL PRIMARY KEY,  
 cost\_type enum\_asset\_cost\_management\_cost\_type NOT NULL,  
 amount DECIMAL(15,2) NOT NULL,  
 currency VARCHAR(3) DEFAULT 'USD',  
 billing\_cycle enum\_asset\_cost\_management\_billing\_cycle DEFAULT 'one\_time',  
 start\_date TIMESTAMPTZ,  
 end\_date TIMESTAMPTZ,  
 vendor VARCHAR(255),  
 contract\_number VARCHAR(255),  
 purchase\_order VARCHAR(255),  
 invoice\_number VARCHAR(255),  
 cost\_center VARCHAR(255),  
 budget\_code VARCHAR(255),  
 notes TEXT,  
 attachments JSONB DEFAULT '[]',  
 metadata JSONB DEFAULT '{}',  
 created\_by INTEGER REFERENCES users(id),  
 last\_modified\_by INTEGER REFERENCES users(id),  
 created\_at TIMESTAMPTZ DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMPTZ DEFAULT CURRENT\_TIMESTAMP,  
 asset\_uuid UUID  
);

#### Enums

* **cost\_type**: purchase, lease, maintenance, support, license, subscription, upgrade, repair, insurance, other
* **billing\_cycle**: one\_time, monthly, quarterly, semi\_annual, annual, biennial

#### Usage Examples

// Track a server purchase  
await db.insert(assetCostManagement).values({  
 costType: 'purchase',  
 amount: '15000.00',  
 currency: 'USD',  
 billingCycle: 'one\_time',  
 vendor: 'Dell Technologies',  
 contractNumber: 'DELL-2025-001',  
 purchaseOrder: 'PO-2025-0123',  
 costCenter: 'IT-INFRASTRUCTURE',  
 budgetCode: 'CAPEX-2025-Q1',  
 assetUuid: '550e8400-e29b-41d4-a716-446655440000'  
});  
  
// Track monthly software license  
await db.insert(assetCostManagement).values({  
 costType: 'license',  
 amount: '299.99',  
 currency: 'USD',  
 billingCycle: 'monthly',  
 vendor: 'Microsoft',  
 startDate: new Date('2025-01-01'),  
 endDate: new Date('2025-12-31'),  
 assetUuid: '550e8400-e29b-41d4-a716-446655440000'  
});

### 2. asset\_lifecycle

Manages asset lifecycle from purchase through end-of-life, including warranty tracking and replacement planning.

#### Schema

CREATE TABLE asset\_lifecycle (  
 id SERIAL PRIMARY KEY,  
 purchase\_date DATE,  
 warranty\_end\_date DATE,  
 manufacturer\_eol\_date DATE,  
 internal\_eol\_date DATE,  
 replacement\_cycle\_months INTEGER,  
 estimated\_replacement\_cost DECIMAL(15,2),  
 replacement\_budget\_year INTEGER,  
 replacement\_budget\_quarter INTEGER,  
 replacement\_notes TEXT,  
 created\_at TIMESTAMPTZ DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMPTZ DEFAULT CURRENT\_TIMESTAMP,  
 asset\_uuid UUID  
);

#### Usage Examples

// Track server lifecycle  
await db.insert(assetLifecycle).values({  
 purchaseDate: '2025-01-15',  
 warrantyEndDate: '2028-01-15',  
 manufacturerEolDate: '2030-01-15',  
 internalEolDate: '2029-06-30',  
 replacementCycleMonths: 48,  
 estimatedReplacementCost: '18000.00',  
 replacementBudgetYear: 2029,  
 replacementBudgetQuarter: 2,  
 replacementNotes: 'Consider cloud migration before replacement',  
 assetUuid: '550e8400-e29b-41d4-a716-446655440000'  
});

### 3. asset\_operational\_costs

Tracks monthly operational costs broken down by category (power, space, network, storage, labor).

#### Schema

CREATE TABLE asset\_operational\_costs (  
 id SERIAL PRIMARY KEY,  
 year\_month DATE NOT NULL,  
 power\_cost DECIMAL(15,2),  
 space\_cost DECIMAL(15,2),  
 network\_cost DECIMAL(15,2),  
 storage\_cost DECIMAL(15,2),  
 labor\_cost DECIMAL(15,2),  
 other\_costs DECIMAL(15,2),  
 notes TEXT,  
 created\_at TIMESTAMPTZ DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMPTZ DEFAULT CURRENT\_TIMESTAMP,  
 asset\_uuid UUID  
);

#### Usage Examples

// Track January 2025 operational costs  
await db.insert(assetOperationalCosts).values({  
 yearMonth: '2025-01-01',  
 powerCost: '245.50',  
 spaceCost: '150.00',  
 networkCost: '89.99',  
 storageCost: '125.00',  
 laborCost: '500.00',  
 otherCosts: '25.00',  
 notes: 'Increased power costs due to higher utilization',  
 assetUuid: '550e8400-e29b-41d4-a716-446655440000'  
});

### 4. asset\_risk\_mapping

Links assets to risk models and cost centers with confidence scoring and verification tracking.

#### Schema

CREATE TABLE asset\_risk\_mapping (  
 id SERIAL PRIMARY KEY,  
 asset\_uuid UUID,  
 existing\_asset\_id INTEGER,  
 risk\_model\_id INTEGER,  
 cost\_center\_id INTEGER,  
 mapping\_confidence DECIMAL(3,2) DEFAULT 0.85,  
 mapping\_method VARCHAR(50) DEFAULT 'automatic',  
 mapping\_criteria JSONB,  
 verified\_by INTEGER REFERENCES users(id),  
 verified\_at TIMESTAMP,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 UNIQUE(asset\_uuid, existing\_asset\_id)  
);

#### Indexes

* idx\_asset\_risk\_mapping\_asset\_uuid on asset\_uuid
* idx\_asset\_risk\_mapping\_existing\_asset on existing\_asset\_id

#### Usage Examples

// Map asset to risk model  
await db.insert(assetRiskMapping).values({  
 assetUuid: '550e8400-e29b-41d4-a716-446655440000',  
 existingAssetId: 12345,  
 riskModelId: 1,  
 costCenterId: 100,  
 mappingConfidence: 0.95,  
 mappingMethod: 'manual',  
 mappingCriteria: {  
 criteria: ['hostname\_match', 'ip\_match', 'location\_match'],  
 scores: [0.98, 0.92, 0.95]  
 },  
 verifiedBy: 1,  
 verifiedAt: new Date()  
});

## Relationships

### Asset UUID Linking

All tables link to assets via asset\_uuid field: - Links to the main assets table - Enables comprehensive asset tracking across all dimensions - Supports asset lifecycle management

### User Tracking

Cost management and risk mapping tables track user actions: - created\_by - User who created the record - last\_modified\_by - User who last updated the record - verified\_by - User who verified risk mappings

## Common Queries

### Total Cost of Ownership (TCO)

-- Get total costs for an asset  
SELECT   
 a.hostname,  
 SUM(acm.amount) as total\_cost,  
 COUNT(acm.id) as cost\_entries  
FROM assets a  
LEFT JOIN asset\_cost\_management acm ON a.asset\_uuid = acm.asset\_uuid  
WHERE a.asset\_uuid = '550e8400-e29b-41d4-a716-446655440000'  
GROUP BY a.hostname;

### Assets Approaching EOL

-- Find assets approaching end-of-life  
SELECT   
 a.hostname,  
 al.warranty\_end\_date,  
 al.manufacturer\_eol\_date,  
 al.internal\_eol\_date,  
 al.estimated\_replacement\_cost  
FROM assets a  
JOIN asset\_lifecycle al ON a.asset\_uuid = al.asset\_uuid  
WHERE al.internal\_eol\_date <= CURRENT\_DATE + INTERVAL '6 months'  
ORDER BY al.internal\_eol\_date;

### Monthly Operational Costs

-- Get operational costs by month  
SELECT   
 year\_month,  
 SUM(power\_cost) as total\_power,  
 SUM(space\_cost) as total\_space,  
 SUM(network\_cost) as total\_network,  
 SUM(storage\_cost) as total\_storage,  
 SUM(labor\_cost) as total\_labor  
FROM asset\_operational\_costs  
WHERE year\_month >= '2025-01-01'  
GROUP BY year\_month  
ORDER BY year\_month;

### Risk Mapping Confidence

-- Find low-confidence risk mappings  
SELECT   
 a.hostname,  
 arm.mapping\_confidence,  
 arm.mapping\_method,  
 arm.verified\_at  
FROM assets a  
JOIN asset\_risk\_mapping arm ON a.asset\_uuid = arm.asset\_uuid  
WHERE arm.mapping\_confidence < 0.80  
 AND arm.verified\_at IS NULL  
ORDER BY arm.mapping\_confidence;

## Best Practices

### Cost Management

1. **Regular Updates**: Update operational costs monthly
2. **Cost Categories**: Use consistent cost types and billing cycles
3. **Documentation**: Include detailed notes and attachments
4. **Budget Tracking**: Link to cost centers and budget codes

### Lifecycle Management

1. **Proactive Planning**: Set internal EOL dates before manufacturer EOL
2. **Replacement Budgeting**: Plan replacement costs in advance
3. **Warranty Tracking**: Monitor warranty expiration dates
4. **Documentation**: Maintain detailed replacement notes

### Risk Mapping

1. **Confidence Scoring**: Use realistic confidence scores
2. **Verification**: Verify automated mappings manually
3. **Criteria Documentation**: Document mapping criteria in JSONB
4. **Regular Review**: Review and update mappings periodically

## Integration Points

### With Existing Systems

* **Assets Table**: Core asset information
* **Users Table**: User tracking and verification
* **Vulnerabilities**: Risk assessment integration
* **Controls**: Compliance and security controls

### With External Systems

* **Financial Systems**: Cost center and budget integration
* **CMDB**: Configuration management database
* **Risk Management**: Risk model integration
* **Procurement**: Purchase order and vendor management