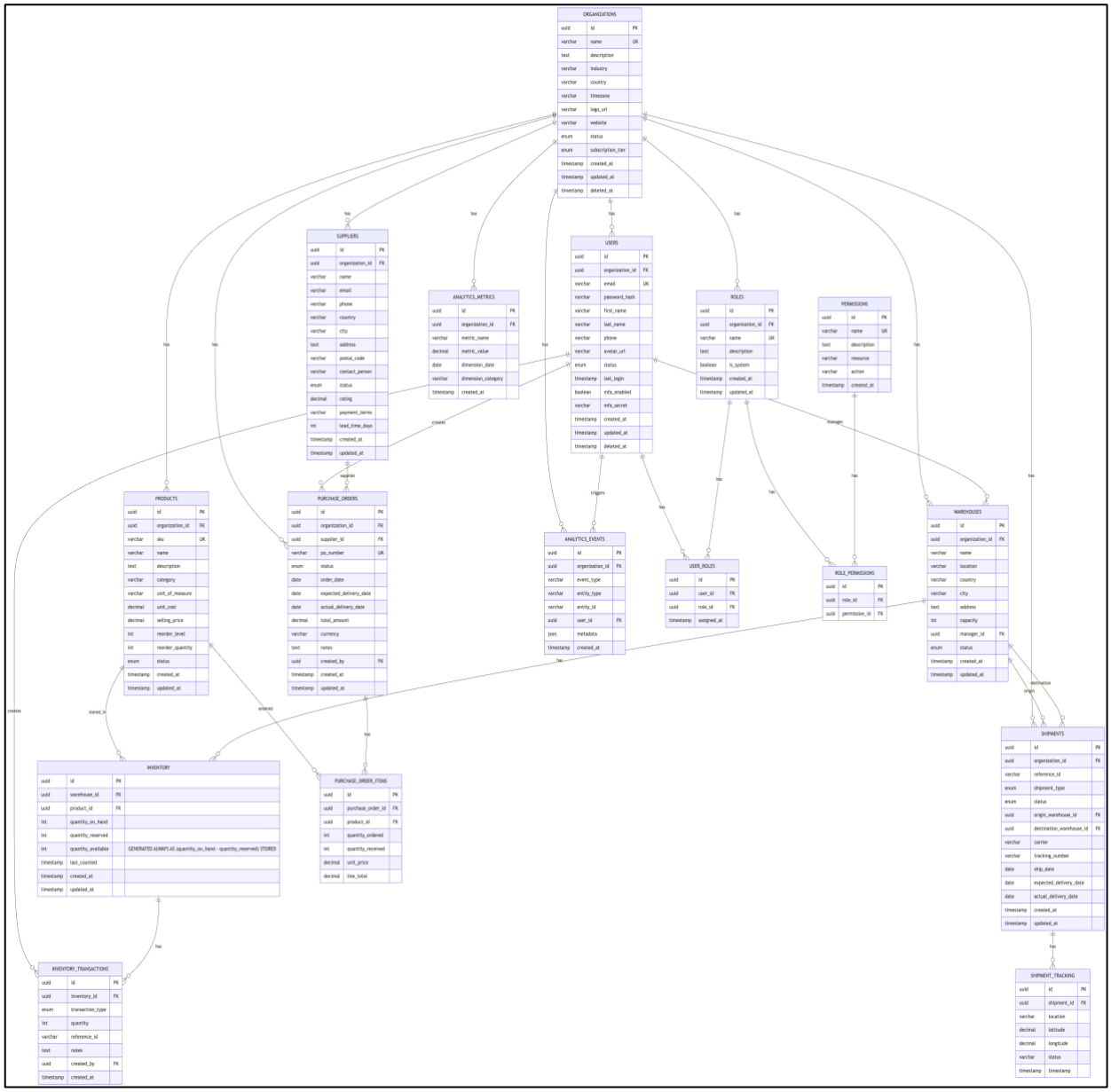
IMSOP - Database Schema

Entity Relationship Diagram



Core Tables

Organizations - Central organization management and multi-tenancy support.

```
CREATE TABLE organizations (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    name VARCHAR(255) NOT NULL UNIQUE,
    description TEXT,
    industry VARCHAR(100),
    country VARCHAR(100),
    timezone VARCHAR(50) DEFAULT 'UTC',
    logo_url VARCHAR(500),
    website VARCHAR(255),
    status ENUM('active', 'inactive', 'suspended') DEFAULT 'active',
    subscription_tier ENUM('starter', 'professional', 'enterprise') DEFAULT
'starter',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    deleted_at TIMESTAMP NULL,
    INDEX idx_status (status),
    INDEX idx_subscription_tier (subscription_tier),
    UNIQUE KEY unique_active_org (name, deleted_at)
);
```

Users - User accounts and authentication.

```
CREATE TABLE users (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    organization_id UUID NOT NULL,
    email VARCHAR(255) NOT NULL,
    password_hash VARCHAR(255) NOT NULL,
    first_name VARCHAR(100),
    last_name VARCHAR(100),
    phone VARCHAR(20),
    avatar_url VARCHAR(500),
    status ENUM('active', 'inactive', 'suspended') DEFAULT 'active',
    last_login TIMESTAMP NULL,
    mfa_enabled BOOLEAN DEFAULT FALSE,
    mfa_secret VARCHAR(255),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    deleted_at TIMESTAMP NULL,
    FOREIGN KEY (organization_id) REFERENCES organizations(id),
    UNIQUE KEY unique_email_per_org (organization_id, email),
    INDEX idx_organization_id (organization_id),
    INDEX idx_status (status),
    UNIQUE KEY unique_active_user (email, deleted_at)
);
```

Roles & Permissions - Role-based access control.

```
CREATE TABLE roles (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    organization_id UUID NOT NULL,
    name VARCHAR(100) NOT NULL,
    description TEXT,
```

```
    is_system BOOLEAN DEFAULT FALSE,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    FOREIGN KEY (organization_id) REFERENCES organizations(id),
    UNIQUE KEY unique_role_per_org (organization_id, name),
    INDEX idx_organization_id (organization_id)
);
CREATE TABLE permissions (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    name VARCHAR(100) NOT NULL UNIQUE,
    description TEXT,
    resource VARCHAR(100) NOT NULL,
    action VARCHAR(50) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    UNIQUE KEY unique_resource_action (resource, action)
);
CREATE TABLE role_permissions (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    role_id UUID NOT NULL,
    permission_id UUID NOT NULL,
    FOREIGN KEY (role_id) REFERENCES roles(id) ON DELETE CASCADE,
    FOREIGN KEY (permission_id) REFERENCES permissions(id),
    UNIQUE KEY unique_role_permission (role_id, permission_id)
);
CREATE TABLE user_roles (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    user_id UUID NOT NULL,
    role_id UUID NOT NULL,
    assigned_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE,
    FOREIGN KEY (role_id) REFERENCES roles(id),
    UNIQUE KEY unique_user_role (user_id, role_id)
);
```

Suppliers - Supplier management and tracking.

```
CREATE TABLE suppliers (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    organization_id UUID NOT NULL,
    name VARCHAR(255) NOT NULL,
    email VARCHAR(255),
    phone VARCHAR(20),
    country VARCHAR(100),
    city VARCHAR(100),
    address TEXT,
    postal_code VARCHAR(20),
    contact_person VARCHAR(100),
    status ENUM('active', 'inactive', 'blocked') DEFAULT 'active',
    rating DECIMAL(3,2) DEFAULT 0,
    payment_terms VARCHAR(50),
    lead_time_days INT DEFAULT 0,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    FOREIGN KEY (organization_id) REFERENCES organizations(id),
    INDEX idx_organization_id (organization_id),
    INDEX idx_status (status),
    INDEX idx_rating (rating)
);
```

Products - Product catalog and specifications.

```
CREATE TABLE products (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    organization_id UUID NOT NULL,
    sku VARCHAR(100) NOT NULL,
    name VARCHAR(255) NOT NULL,
    description TEXT,
    category VARCHAR(100),
    unit_of_measure VARCHAR(20),
    unit_cost DECIMAL(12,4),
    selling_price DECIMAL(12,4),
    reorder_level INT DEFAULT 0,
    reorder_quantity INT DEFAULT 0,
    status ENUM('active', 'discontinued', 'archived') DEFAULT 'active',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    FOREIGN KEY (organization_id) REFERENCES organizations(id),
    UNIQUE KEY unique_sku_per_org (organization_id, sku),
    INDEX idx_organization_id (organization_id),
    INDEX idx_category (category),
    INDEX idx_status (status)
);
```

Warehouses & Inventory - Warehouse locations and inventory tracking.

```
CREATE TABLE warehouses (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    organization_id UUID NOT NULL,
    name VARCHAR(255) NOT NULL,
    location VARCHAR(255),
    country VARCHAR(100),
    city VARCHAR(100),
    address TEXT,
    capacity INT,
    manager_id UUID,
    status ENUM('active', 'inactive', 'maintenance') DEFAULT 'active',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    FOREIGN KEY (organization_id) REFERENCES organizations(id),
    FOREIGN KEY (manager_id) REFERENCES users(id),
    INDEX idx_organization_id (organization_id),
    INDEX idx_status (status)
);
CREATE TABLE inventory (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    warehouse_id UUID NOT NULL,
    product_id UUID NOT NULL,
    quantity_on_hand INT DEFAULT 0,
    quantity_reserved INT DEFAULT 0,
    quantity_available INT GENERATED ALWAYS AS (quantity_on_hand -
quantity_reserved) STORED,
    last_counted TIMESTAMP,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    FOREIGN KEY (warehouse_id) REFERENCES warehouses(id),
    FOREIGN KEY (product_id) REFERENCES products(id),
```

```
    UNIQUE KEY unique_warehouse_product (warehouse_id, product_id),
    INDEX idx_warehouse_id (warehouse_id),
    INDEX idx_product_id (product_id)
);
CREATE TABLE inventory_transactions (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    inventory_id UUID NOT NULL,
    transaction_type ENUM('receipt', 'shipment', 'adjustment', 'return')
NOT NULL,
    quantity INT NOT NULL,
    reference_id VARCHAR(100),
    notes TEXT,
    created_by UUID,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (inventory_id) REFERENCES inventory(id),
    FOREIGN KEY (created_by) REFERENCES users(id),
    INDEX idx_inventory_id (inventory_id),
    INDEX idx_created_at (created_at)
);
```

Purchase Orders - Purchase order management and tracking.

```
CREATE TABLE purchase_orders (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    organization_id UUID NOT NULL,
    supplier_id UUID NOT NULL,
    po_number VARCHAR(100) NOT NULL,
    status ENUM('draft', 'submitted', 'confirmed', 'shipped', 'received',
'cancelled') DEFAULT 'draft',
    order_date DATE NOT NULL,
    expected_delivery_date DATE,
    actual_delivery_date DATE,
    total_amount DECIMAL(14,2),
    currency VARCHAR(3) DEFAULT 'USD',
    notes TEXT,
    created_by UUID,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    FOREIGN KEY (organization_id) REFERENCES organizations(id),
    FOREIGN KEY (supplier_id) REFERENCES suppliers(id),
    FOREIGN KEY (created_by) REFERENCES users(id),
    UNIQUE KEY unique_po_per_org (organization_id, po_number),
    INDEX idx_organization_id (organization_id),
    INDEX idx_supplier_id (supplier_id),
    INDEX idx_status (status),
    INDEX idx_order_date (order_date)
);
CREATE TABLE purchase_order_items (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    purchase_order_id UUID NOT NULL,
    product_id UUID NOT NULL,
    quantity_ordered INT NOT NULL,
    quantity_received INT DEFAULT 0,
    unit_price DECIMAL(12,4),
    line_total DECIMAL(14,2),
    FOREIGN KEY (purchase_order_id) REFERENCES purchase_orders(id) ON
DELETE CASCADE,
    FOREIGN KEY (product_id) REFERENCES products(id),
    INDEX idx_purchase_order_id (purchase_order_id)
```

```
);
```

Shipments & Tracking - Shipment management and real-time tracking.

```
CREATE TABLE shipments (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    organization_id UUID NOT NULL,
    reference_id VARCHAR(100),
    shipment_type ENUM('inbound', 'outbound', 'transfer') NOT NULL,
    status ENUM('pending', 'in_transit', 'delivered', 'cancelled') DEFAULT
'pending',
    origin_warehouse_id UUID,
    destination_warehouse_id UUID,
    carrier VARCHAR(100),
    tracking_number VARCHAR(100),
    ship_date DATE,
    expected_delivery_date DATE,
    actual_delivery_date DATE,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    FOREIGN KEY (organization_id) REFERENCES organizations(id),
    FOREIGN KEY (origin_warehouse_id) REFERENCES warehouses(id),
    FOREIGN KEY (destination_warehouse_id) REFERENCES warehouses(id),
    INDEX idx_organization_id (organization_id),
    INDEX idx_status (status),
    INDEX idx_tracking_number (tracking_number)
);
CREATE TABLE shipment_tracking (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    shipment_id UUID NOT NULL,
    location VARCHAR(255),
    latitude DECIMAL(10,8),
    longitude DECIMAL(11,8),
    status VARCHAR(50),
    timestamp TIMESTAMP,
    FOREIGN KEY (shipment_id) REFERENCES shipments(id) ON DELETE CASCADE,
    INDEX idx_shipment_id (shipment_id),
    INDEX idx_timestamp (timestamp)
);
```

Analytics & Metrics - Analytics and performance metrics.

```
CREATE TABLE analytics_events (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    organization_id UUID NOT NULL,
    event_type VARCHAR(100) NOT NULL,
    entity_type VARCHAR(100),
    entity_id VARCHAR(100),
    user_id UUID,
    metadata JSON,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (organization_id) REFERENCES organizations(id),
    FOREIGN KEY (user_id) REFERENCES users(id),
    INDEX idx_organization_id (organization_id),
    INDEX idx_event_type (event_type),
    INDEX idx_created_at (created_at)
);
CREATE TABLE analytics_metrics (
```

```
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    organization_id UUID NOT NULL,
    metric_name VARCHAR(100) NOT NULL,
    metric_value DECIMAL(14,4),
    dimension_date DATE,
    dimension_category VARCHAR(100),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (organization_id) REFERENCES organizations(id),
    INDEX idx_organization_id (organization_id),
    INDEX idx_metric_name (metric_name),
    INDEX idx_dimension_date (dimension_date)
);
```

Indexing Strategy

Primary Indexes

- All primary keys indexed automatically
- Foreign keys indexed for join performance
- Organization ID indexed for multi-tenancy filtering

Performance Indexes

- Status fields indexed for filtering
- Date fields indexed for range queries
- Frequently searched fields indexed

Composite Indexes

- (organization_id, status) for filtered queries
- (warehouse_id, product_id) for inventory lookups
- (created_at, organization_id) for time-series queries

Query Patterns

Get Inventory by Product

```
SELECT i.*, p.name, w.name as warehouse_name
FROM inventory i
JOIN products p ON i.product_id = p.id
JOIN warehouses w ON i.warehouse_id = w.id
WHERE p.organization_id = ? AND p.sku = ?
ORDER BY w.name;
```

Get Purchase Order Status

```
SELECT po.*, GROUP_CONCAT(p.name) as products
FROM purchase_orders po
LEFT JOIN purchase_order_items poi ON po.id = poi.purchase_order_id
LEFT JOIN products p ON poi.product_id = p.id
WHERE po.organization_id = ? AND po.po_number = ?
GROUP BY po.id;
```

Real-time Shipment Tracking

```
SELECT s.*, st.location, st.latitude, st.longitude, st.timestamp
FROM shipments s
LEFT JOIN shipment_tracking st ON s.id = st.shipment_id
WHERE s.organization_id = ? AND s.tracking_number = ?
ORDER BY st.timestamp DESC
LIMIT 1;
```

Performance Optimization

Query Optimization

- Use EXPLAIN ANALYZE for query planning
- Avoid SELECT * queries
- Use appropriate JOIN types
- Implement pagination for large result sets

Caching Strategy

- Cache frequently accessed products
- Cache organization settings
- Cache user permissions
- Cache warehouse locations

Data Retention

- Archive old purchase orders (>2 years)
- Archive old shipment tracking (>1 year)
- Archive old analytics events (>6 months)
- Keep current inventory indefinitely

Backup & Recovery

Backup Strategy

- Daily full backups (Aiven automated)
- Hourly incremental backups
- Point-in-time recovery enabled
- Backup retention: 30 days

Recovery Procedures

- Test recovery procedures monthly
- Document recovery time objectives (RTO)
- Document recovery point objectives (RPO)
- Maintain backup verification logs

Security Considerations

Data Protection

- Encrypt sensitive data at rest (Aiven PostgreSQL encryption)
- Use TLS for data in transit
- Hash passwords with bcrypt
- Mask PII in logs

Access Control

- Row-level security for multi-tenancy
- Column-level encryption for sensitive data
- Audit trail for all modifications
- Regular security audits