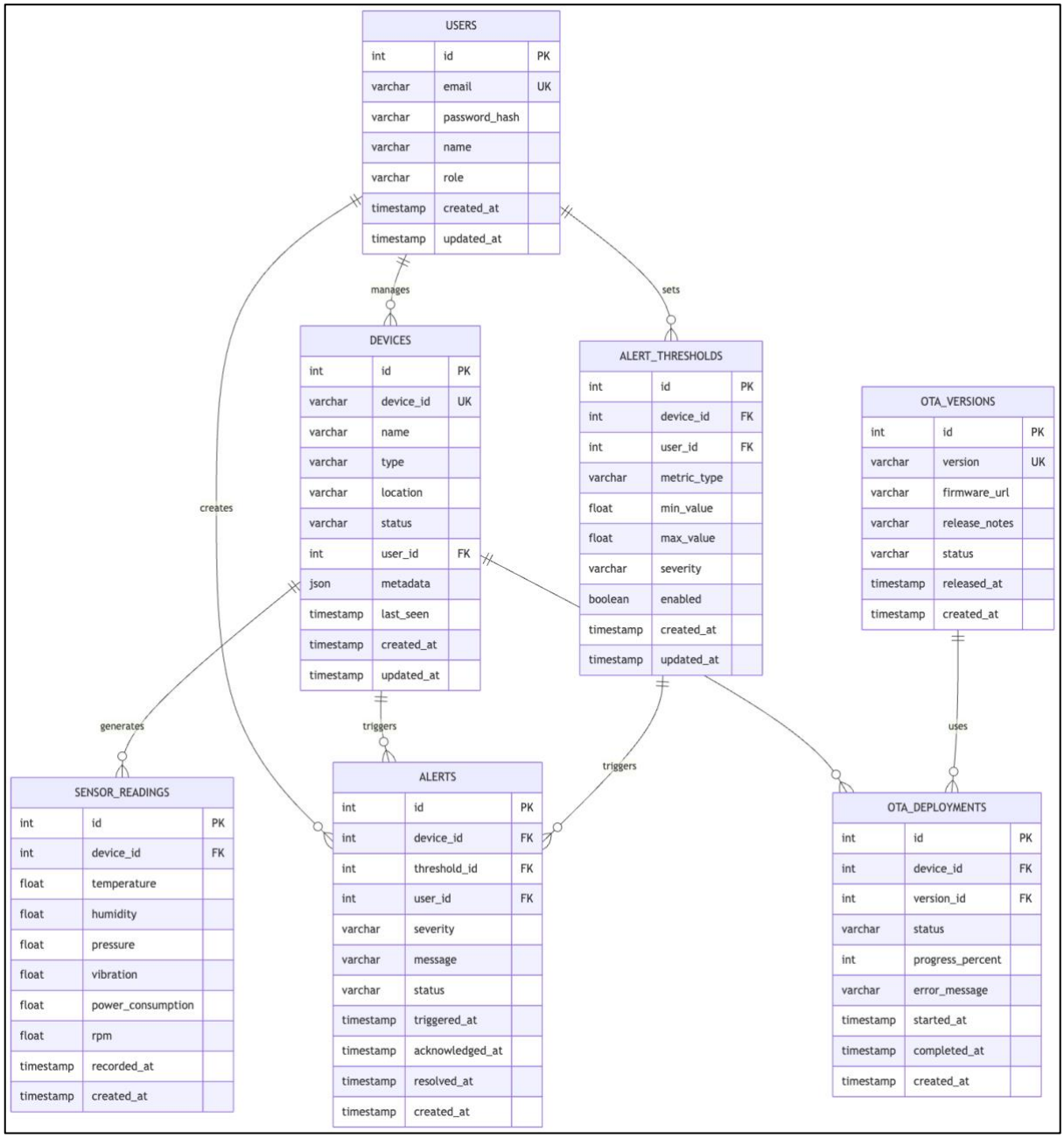
Smart Factory IoT - Database Schema

Entity Relationship Diagram



Core Tables

USERS - Stores user account information and authentication credentials. Integrated with Microsoft Entra ID for authentication.

```
CREATE TABLE users (
    id INT PRIMARY KEY AUTO_INCREMENT,
    email VARCHAR(255) UNIQUE NOT NULL,
    password_hash VARCHAR(255) NOT NULL,
    name VARCHAR(255) NOT NULL,
    role ENUM('admin', 'operator', 'user') NOT NULL,
    created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP
);
```

DEVICES - Represents IoT devices in the factory, registered via Azure IoT Hub.

```
CREATE TABLE devices (
    id INT PRIMARY KEY AUTO_INCREMENT,
    device_id VARCHAR(255) UNIQUE NOT NULL,
    name VARCHAR(255) NOT NULL,
    type VARCHAR(100) NOT NULL,
    location VARCHAR(255) NOT NULL,
    status ENUM('online', 'offline', 'error') NOT NULL,
    user_id INT,
    metadata JSON,
    last_seen TIMESTAMP,
    created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id)
);
```

SENSOR_READINGS - Time-series data from device sensors, ingested via Azure IoT Hub telemetry.

```
CREATE TABLE sensor_readings (
    id INT PRIMARY KEY AUTO_INCREMENT,
    device_id INT NOT NULL,
    temperature FLOAT,
    humidity FLOAT,
    pressure FLOAT,
    vibration FLOAT,
    power_consumption FLOAT,
    rpm FLOAT,
    recorded_at TIMESTAMP NOT NULL,
    created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (device_id) REFERENCES devices(id),
    INDEX idx_device_id_recorded_at (device_id, recorded_at),
    INDEX idx_recorded_at (recorded_at)
);
```

ALERTS - System alerts triggered by threshold violations, with notifications via Azure Logic Apps.

```
CREATE TABLE alerts (
    id INT PRIMARY KEY AUTO_INCREMENT,
    device_id INT NOT NULL,
    threshold_id INT,
    user_id INT,
    severity ENUM('critical', 'warning', 'info') NOT NULL,
    message TEXT NOT NULL,
    status ENUM('active', 'acknowledged', 'resolved') NOT NULL,
    triggered_at TIMESTAMP NOT NULL,
    acknowledged_at TIMESTAMP,
    resolved_at TIMESTAMP,
    created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (device_id) REFERENCES devices(id),
    FOREIGN KEY (threshold_id) REFERENCES alert_thresholds(id),
    FOREIGN KEY (user_id) REFERENCES users(id),
    INDEX idx_user_id_created_at (user_id, created_at),
    INDEX idx_status_severity (status, severity)
);
```

ALERT_THRESHOLDS - Configuration for alert triggers.

```
CREATE TABLE alert_thresholds (
    id INT PRIMARY KEY AUTO_INCREMENT,
    device_id INT NOT NULL,
    user_id INT NOT NULL,
    metric_type VARCHAR(100) NOT NULL,
    min_value FLOAT,
    max_value FLOAT,
    severity ENUM('critical', 'warning', 'info') NOT NULL,
    enabled BOOLEAN NOT NULL,
    created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    FOREIGN KEY (device_id) REFERENCES devices(id),
    FOREIGN KEY (user_id) REFERENCES users(id)
);
```

OTA_VERSIONS - Firmware version management, with URLs stored in Azure Blob Storage.

```
CREATE TABLE ota_versions (
    id INT PRIMARY KEY AUTO_INCREMENT,
    version VARCHAR(50) UNIQUE NOT NULL,
    firmware_url VARCHAR(500) NOT NULL,
    release_notes TEXT,
    status ENUM('draft', 'released', 'deprecated') NOT NULL,
    released_at TIMESTAMP,
    created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
);
```

OTA_DEPLOYMENTS - Tracks firmware update deployments to devices via Azure IoT Hub.

```
CREATE TABLE ota_deployments (
    id INT PRIMARY KEY AUTO_INCREMENT,
    device_id INT NOT NULL,
    version_id INT NOT NULL,
    status ENUM('pending', 'in_progress', 'completed', 'failed') NOT NULL,
    progress_percent INT,
    error_message TEXT,
    started_at TIMESTAMP,
    completed_at TIMESTAMP,
    created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (device_id) REFERENCES devices(id),
    FOREIGN KEY (version_id) REFERENCES ota_versions(id),
    INDEX idx_device_id_status (device_id, status),
    INDEX idx_version_id_status (version_id, status)
);
```

Indexing Strategy

Primary Indexes

- Primary keys on all tables
- Foreign key indexes for joins
- Composite indexes for common queries
- Timestamp indexes for time-range queries

Performance Indexes

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

Composite Indexes

- (device_id, recorded_at) for time-series queries
- (user_id, created_at) for user alert queries
- (status, severity) for alert filtering
- (device_id, status) for device deployment queries
- (version_id, status) for version deployment tracking

Query Patterns

Real-time Sensor Data

```
SELECT * FROM sensor_readings
WHERE device_id = ? AND recorded_at > NOW() - INTERVAL 1 HOUR
ORDER BY recorded_at DESC
LIMIT 100;
```

Active Alerts

```sql
SELECT a.*, d.name as device_name, u.email as user_email
FROM alerts a
JOIN devices d ON a.device_id = d.id
JOIN users u ON a.user_id = u.id
WHERE a.status = 'active'
ORDER BY a.severity DESC, a.triggered_at DESC;
```

Device Status Overview

```sql
SELECT
    d.id, d.name, d.status,
    COUNT(CASE WHEN a.status = 'active' THEN 1 END) as active_alerts,
    MAX(sr.recorded_at) as last_reading
FROM devices d
LEFT JOIN alerts a ON d.id = a.device_id
LEFT JOIN sensor_readings sr ON d.id = sr.device_id
GROUP BY d.id;
```

Performance Optimization

Query Optimization

- Use EXPLAIN to analyze queries
- Avoid N+1 queries with proper joins
- Batch operations for bulk updates
- Pagination for large result sets

Caching Strategy

- Cache frequently accessed thresholds using Azure Cache for Redis
- Cache device metadata
- Cache user roles and permissions via Microsoft Entra ID
- Invalidate cache on updates via Azure Functions

Data Retention

- Archive old sensor readings (>90 days)
- Archive old alerts (>1 year)
- Archive old OTA deployments (>1 year)
- Keep current devices and users indefinitely

Backup & Recovery

Backup Strategy

- Daily incremental backups via Azure Backup
- Weekly full backups
- Backup retention: 30 days
- Off-site backup storage in Azure

Recovery Procedures

- Identify recovery point
- Restore from Azure Backup
- Verify data integrity
- Sync with replicas using Azure SQL Geo-Replication
- Resume operations

Security Considerations

Data Protection

- Encrypt sensitive fields at rest using Azure SQL TDE
- Use parameterized queries to prevent SQL injection
- Implement row-level security for multi-tenant data
- Audit all data modifications via Azure SQL Auditing

Access Control

- Role-based access to data via RBAC and Microsoft Entra ID
- User can only see their own devices
- Admin can see all data
- Operator has read-only access

Compliance

- GDPR compliance for user data
- Data anonymization for deleted users
- Audit logs for all modifications
- Encryption in transit (HTTPS/TLS)