SQL Server Agent Automation Guide for Data Warehouse Operations

Below is a step-by-step guide to automate your entire data warehouse pipeline using SQL Server Agent. This covers all layers from Bronze to Gold, including the `PlaceOrder` and `UpdateInventory` procedures.

Step 1: Create SQL Server Agent Jobs

Job 1: `DW_Load_Bronze_Layer`

Purpose: Load raw data from CSV files into Bronze tables

```
-- 1. In SSMS, connect to your SQL Server instance
-- 2. Expand SQL Server Agent -> Jobs
-- 3. Right-click -> New Job
-- 4. Name: "DW_Load_Bronze_Layer"
-- 5. Add a new step with this T-SQL:

USE Team1;
GO
EXEC bronze.load_bronze;
GO

-- 6. Set schedule: Daily at 1:00 AM
```

Job 2: `DW_Transform_Silver_Layer`

Purpose: Transform Bronze data into Silver layer

```
-- 1. New Job named "DW_Transform_Silver_Layer"
-- 2. Add step 1:

USE Team1;
GO
```

```
EXEC silver.load_silver;
GO

-- 3. Add step 2 (PlaceOrder validation):
USE Team1;
GO
-- This would actually be triggered by application, not scheduled
-- For demo purposes, we'll create a validation step
PRINT 'PlaceOrder procedure available for application calls';
GO

-- 4. Add step 3 (Inventory maintenance):
USE Team1;
GO
-- Example inventory update (would normally be called by application)
DECLARE @Status NVARCHAR(50);
EXEC silver.UpdateInventory
    @ProductID = 5,
    @NewQuantity = 100,
    @UpdateStatus = @Status OUTPUT;
PRINT @Status;
GO

-- 5. Set schedule: Daily at 2:00 AM (after Bronze load completes)
-- 6. Add dependency: Only run if Bronze job succeeded
```

Job 3: DW_Refresh_Gold_Layer

Purpose: Refresh Gold layer views

```
-- 1. New Job named "DW_Refresh_Gold_Layer"
-- 2. Add step:

USE Team1;
GO
-- Views automatically refresh when queried, but we can:
-- 1) Update statistics
-- 2) Rebuild indexes if needed
-- 3) Run sample queries to pre-cache

-- Example: Refresh stats on key tables
UPDATE STATISTICS silver.orders WITH FULLSCAN;
UPDATE STATISTICS silver.products WITH FULLSCAN;
```

```
UPDATE STATISTICS silver.customers WITH FULLSCAN;
GO

-- 3. Set schedule: Daily at 3:00 AM
-- 4. Add dependency: Only run if Silver job succeeded
```

Step 2: Configure Job Dependencies

1. Right-click `DW_Transform_Silver_Layer` -> Properties
2. Go to "Notifications" page
3. Check "Email" and select your operator
4. Set to email on "Failure"
5. Repeat for all jobs

Step 3: Create Operator for Notifications

```
USE msdb;
GO
EXEC dbo.sp_add_operator
    @name = N'DataWarehouseAdmin',
    @email_address = N'your.email@company.com';
GO
```

Step 4: Error Handling Configuration

Add this to each stored procedure (if not already present):

```
-- Example addition to silver.load_silver
BEGIN TRY
    -- Existing procedure code
END TRY
BEGIN CATCH
    DECLARE @ErrorMessage NVARCHAR(4000) = ERROR_MESSAGE();
    DECLARE @ErrorSeverity INT = ERROR_SEVERITY();
    DECLARE @ErrorState INT = ERROR_STATE();

    -- Log error
    INSERT INTO silver.error_log (error_time, error_message, procedure_na
    VALUES (GETDATE(), @ErrorMessage, 'silver.load_silver');
```

```
    -- Re-throw error to SQL Agent
    RAISERROR(@ErrorMessage, @ErrorSeverity, @ErrorState);
END CATCH
```

Step 5: Create Supporting Tables

```
-- Error logging table
CREATE TABLE silver.error_log (
    log_id INT IDENTITY(1,1) PRIMARY KEY,
    error_time DATETIME2 NOT NULL,
    error_message NVARCHAR(4000),
    procedure_name NVARCHAR(255),
    batch_id UNIQUEIDENTIFIER
);

-- Job control table
CREATE TABLE silver.job_control (
    job_name NVARCHAR(255) PRIMARY KEY,
    last_run_time DATETIME2,
    status NVARCHAR(50),
    records_processed INT
);
```

Step 6: Final Automation Workflow

```
graph TD
    A[1:00 AM] -->|SQL Agent| B[Load Bronze]
    B --> C{Success?}
    C -->|Yes| D[2:00 AM Transform Silver]
    C -->|No| E[Send Alert]
    D --> F{Success?}
    F -->|Yes| G[3:00 AM Refresh Gold]
    F -->|No| E
    G --> H{Success?}
    H -->|No| E
```

Verification Steps

1. **Test Bronze Load**:

```
EXEC msdb.dbo.sp_start_job N'DW_Load_Bronze_Layer';
-- Verify data in bronze tables
SELECT COUNT(*) FROM bronze.customers;
```

2. **Test Silver Transformation**:

```
EXEC msdb.dbo.sp_start_job N'DW_Transform_Silver_Layer';
-- Check for errors
SELECT * FROM silver.error_log;
-- Verify inventory update
SELECT * FROM silver.inventory WHERE ProductID = 5;
```

3. **Test Gold Refresh**:

```
EXEC msdb.dbo.sp_start_job N'DW_Refresh_Gold_Layer';
-- Query gold views
SELECT TOP 10 * FROM gold.fact_SalesPerformance;
```

Maintenance Recommendations

1. **Weekly Maintenance Job**:
    - Rebuild indexes on Silver tables
    - Update statistics on all layers
    - Cleanup old error logs
2. **Monitoring**:

```
-- Check job history
SELECT * FROM msdb.dbo.sysjobhistory
WHERE job_id IN (
    SELECT job_id FROM msdb.dbo.sysjobs
    WHERE name LIKE 'DW_%'
);
```

This complete automation setup ensures your data warehouse runs on a daily schedule with proper error handling and notifications. The jobs will execute in sequence with dependencies, and you'll be alerted if any step fails.