**1. Data Import Wizard**

**Purpose:**  
Simplify importing small datasets (up to 50,000 records) directly from the Salesforce UI.

**Use Case:**  
Used by Event Coordinators to upload Event, Venue, and Attendee data from spreadsheets.

**Steps:**

1. Navigate to **Setup → Data Import Wizard**.
2. Choose the object (Event\_\_c, Attendee\_\_c, Venue\_\_c).
3. Upload CSV file.
4. Map fields manually or automatically.
5. Run import and review results in the dashboard.

**Deliverables:**

* Documented import templates for each object.
* Mapped field lists.
* Import test results validation.

**2. Data Loader**

**Purpose:**  
Handle large-scale data operations (insert, update, upsert, delete, export) beyond the limits of the Data Import Wizard.

**Use Case:**  
Used by Admins to load bulk attendees, export historical events, or mass update statuses.

**Steps:**

1. Install Data Loader.
2. Login with OAuth credentials.
3. Select operation type (Insert, Update, Upsert, Delete).
4. Map CSV columns to Salesforce fields.
5. Run job and review success/error logs.

**Deliverables:**

* Reusable CSV templates for bulk imports/updates.
* Logs for each Data Loader operation.
* Bulk API configuration documentation.

**3. Duplicate Rules**

**Purpose:**  
Prevent creation of duplicate records for key objects like Event and Attendee.

**Example Rules:**

* Prevent duplicate Events with the same name and start date.
* Prevent duplicate Attendees with the same email for the same Event.

**Steps:**

1. Setup → Duplicate Rules → New Rule.
2. Define Matching Rule on Event\_\_c and Attendee\_\_c.
3. Choose “Allow but Report” (for soft duplicates) or “Block” (for strict prevention).
4. Activate and test.

**Deliverables:**

* Active Duplicate and Matching Rules.
* Duplicate Record Sets for monitoring.
* Periodic reports of duplicate entries.

**4. Data Export & Backup**

**Purpose:**  
Ensure data safety through regular backups of critical data.

**Tools:**

* **Data Export Service:** Schedule weekly or monthly full backups.
* **Data Loader Export:** For ad-hoc or selective backups.

**Frequency:**

* Full Export: Monthly
* Incremental Export: Weekly
* Before major releases or imports

**Deliverables:**

* Backup schedule document.
* Secure storage location (encrypted ZIPs).
* Verification logs post-export.

**5. Change Sets**

**Purpose:**  
Deploy metadata between environments (Sandbox → UAT → Production) declaratively.

**Steps:**

1. Create Outbound Change Set in Sandbox.
2. Add components (Objects, Fields, Apex Classes, Triggers, Flows, LWCs).
3. Upload to target org.
4. Validate, deploy, and verify.

**Deliverables:**

* Change Set deployment logs.
* Validation reports.
* Deployment checklist.

**6. Unmanaged vs Managed Packages**

**Purpose:**  
Distribute components effectively for testing, development, or external sharing.

| **Type** | **Description** | **Use Case** |
| --- | --- | --- |
| **Unmanaged Package** | Editable components post-installation | Used for sandbox testing or developer collaboration |
| **Managed Package** | Locked components with version control | Used for production deployment or AppExchange distribution |

**Deliverables:**

* Unmanaged package for testing in partner sandboxes.
* Managed package version for deployment.

**7. ANT Migration Tool**

**Purpose:**  
Automate metadata deployments via command-line scripts (Java-based).

**Use Case:**  
Used by developers for CI/CD integration or large, repetitive deployments.

**Steps:**

1. Install Apache ANT and Salesforce Migration Tool.
2. Create build.xml and package.xml.
3. Authenticate to Salesforce using secure credentials.
4. Run commands:
5. ant retrieve
6. ant deploy
7. ant validate
8. Review logs for success/failure.

**Deliverables:**

* Working ANT scripts (build.xml, package.xml).
* Deployment logs.
* Rollback procedures.

**8. VS Code & SFDX**

**Purpose:**  
Enable modern Salesforce development with CLI, source control, and automation.

**Setup:**

1. Install **VS Code**, **Salesforce Extensions Pack**, and **Salesforce CLI (SFDX)**.
2. Authenticate to org:
3. sfdx auth:web:login -a DevHub
4. Create or retrieve project:
5. sfdx force:project:create -n EventManagement
6. sfdx force:source:retrieve -u DevHub -m ApexClass,LightningComponentBundle
7. Deploy metadata:
8. sfdx force:source:deploy -p force-app/main/default/classes
9. Run tests:

sfdx force:apex:test:run --codecoverage