ROCESS MINING STEPS

1. **Extract the following tables/reports:**
   EKBE, EBAN, JQ Register (ZSD007), RFP Monitoring Report (Sl. No 5), JI Register, EKPO, EKKO, RBKP, and RSEG.
2. **Merge RFP Monitoring Report (left) with JI Register**
   - From Current Tender Status column from the RFP report remove Invalidated and cancelled evaluation
   - Join on: Project / Facility Ref and Project Number
   - Output: JI details
   - Save as: **DB1**
3. **Aggregate from JQ Register:**
   - Calculate SUM(Total Amount)
   - Concatenate unique JQ Numbers
4. **Merge JQ aggregation with DB1 (DB1 on left):**
   - Join on: Project / Facility and Project Definition
   - Save as: **DB2**
5. **Merge EBAN (right) with DB2 (left):**
   - Join on: Jaggaer Tender Reference and Tender #
   - Save as: **DB3**
6. **Join EKKO and EKPO on Purchasing Document:**
   - From EKKO, bring: Creation Date, Currency
   - Apply currency conversion function
   - Save as: **DB4**
7. **Merge DB3 (left) with DB4 (right):**
   - Join on: Purchasing Document and Item
   - Save as: **DB5**
8. **Filter EKBE where PO History Category = 'E'**
   - Merge DB5 (left) with filtered EKBE
   - Join on: Purchasing Document and Item
   - Save as: **DB6**
9. **Rename columns from RSEG:**
   - Document Number → Invoice Document No
   - Fiscal Year → Invoice Fiscal Year
   - Amount → Invoice Amount
   - Debit/Credit Indicator → Invoice Debit/Credit Indicator
10. **Rename columns from RBKP:**
    - Document Type → Invoice Document Type
    - Document Date → Invoice Document Date
    - Posting Date → Invoice Posting Date
    - Invoicing Party → Invoice Party
11. **Join RSEG and RBKP:**
    - Join on: invoice document number , RSEG on left
    - Save as: **DB7**
12. **Join DB6 (left) with DB7:**
    - Join on: Purchasing Document and Item
    - Final output contains full PR to invoice traceability

<u>Next steps –</u>

1. Rename the data with its type for example while combining EBAN Suffix columns with "_PR" and likewise.

2. Wherever need to split is written in my output , we need to bring those values in different rows in same column to capture individual vendors.

Control Gap	**Detection Logic**	**Tech logic**	**Dashboard Visual**
Common Vendors Always Receiving RFQs	Count vendors with 90%+ RFQ coverage across PRs	Groupby EOI Received to find the unique count of PR and then find the percentage from the total PR , by dividing the unique count of PO with the total PR.	Bar chart of vendor frequency
2. Dummy Vendors – Vendors always in RFQ list but never submit bids	RFQ shared ≠ Bid submitted across cycles	Vendor cases where RFP issued , but no tendor receive (count ratio) need to run a loop .	Table of vendors with RFQ:Bid ratio < 10%
3. Preferred Vendor Bias	Always winning vendor where others never respond	Cases where tendor received is from one unique vendor.	Highlight cases with winning vendor and all others with null bids
4. Employee-Vendor Nexus	Vendor address linked to employee address/contact		Cross-reference vendor master and HR master
5. Bid Iteration Loops	PR → RFQ → Bid → Resend → Rebid → Final	WIP	Process path with 3+ loops flagged
6. Fast-track PO Cycle (Suspicious Speed)	PR to GRN in < 50% of the avg lead time with same user-vendor pair	Capture the following : 1. PR requisition date , PR release date , PO creation date , PO release date , first GRN date (earliest date) for every PO – PR line item. 2.For every Vendor – user – material PO pair calculate the time from PR creation to first GRN Date . 3. Take the average of the column for that specific vendor user pair . 4. Compare every individual time with average lead time . highlight cases where the time is less than 50% of the average lead time	Heatmap of user-vendor + processing time
7. High Change Frequency	Frequent edits in PO, PR, RFQ by same user	1.Take the Pos against PR from EBAN . 2.Join CDPOS and CDHDR (PR) on the basis of object number and change NR .Suffix every column name with PR_ Change (Remove rows where Fname == KEY) 3. Now join the Object value with the PR in step 1 , and append the POs in the DB . DB1 4. Join CDPOS and CDHDR (PO) on the basis of object number and change NR .DB2 Suffix every column name with PO_ Change (Remove rows where Fname == KEY) 5. Based on the PO number join DB1 and DB2 and Bring all PO change fields. 6. If Uname PR_ Change == Uname PO_ Change , 7. Do the groupby basis Uname and do the unique count of PO and Unique count of PR and unique concatenate of PO_change field name and PR_change field name.	CDHDR/ CDPOS logs on EKKO/EBAN changes visualized
8. Vendor Cartelization	Same set of vendors bidding similar quotes repeatedly	WIP	Clustered bidding patterns , similarity scoring
9. Split POs for Same PR	One PR → Multiple POs → Same vendor	In EKPO , one PR having unique count of Pos and vendor code is same for that PO	Trace PO quantity & value allocation

10. Unapproved Invoices Being Parked	Invoice parked before GRN or PO approval	WIP	Timeline outlier detection
11. Transactions where PO Date is prior to PR		Check : PO Creation Vs PR creation PO Creation Vs PR Rel PO Release Vs PR creation PO Release Vs PR Release	
12. Instances where vendor code changed during invoice posting		WIP	
13. Instances where GRN happened beyond 50% delay from the required delivery date		WIP	

Objective

- **Define and detect the "happy path" in the EO procurement lifecycle.**
- **Quantify how many process instances follow the ideal sequence versus those with deviations.**

2. Ideal ("Happy") Path Definition

List each milestone in strict chronological order. Every case that respects this exact sequence is classified as a happy path.

1. **JI creation (Date of approval)**
2. **JQ creation**
3. **PR creation (requisition date)**
4. **PR release (Release date)**
5. **RAT creation (RAT Creation date)**
6. **RAT approval date (RAT approval date)**
7. **RFQ final**
8. **RAA release**
9. **Pre-award date**
10. **PO creation**
11. **PO release**
12. **GRN Date (First) (Posting date)**
13. **Invoice creation (First) (Posting date)**

3. Data Requirements

- **Case identifier (e.g., PO number, workflow ID)**
- **Event name (one of the twelve milestones above)**
- **Timestamp (date/time of each event)**

All timestamps must be in a date-format field to guarantee accurate sequencing.

4. Methodology

1. **Data Preparation**
   - **Import all event records into a single table or DataFrame.**
   - **Ensure uniform event naming and date formatting.**
2. **Sequence Construction**
   - **Group by case identifier.**
   - **Sort events chronologically within each case.**
   - **Concatenate the ordered event names into a single list or tuple.**
3. **Path Classification**
   - **Compare each case's event list to the predefined ideal sequence.**
   - **If they match exactly, label the case as happy_path.**
   - **Otherwise, label it as variation.**
4. **Aggregation & Reporting**
   - **Count the total number of cases following each distinct sequence.**
   - **Tabulate:**
     - **Path (the sequence of events)**
     - **Count (number of cases)**
     - **Label (happy_path vs. variation)**

5. Sample Output Table

Path Description	Case Count	Classification
JI → JQ → PR → PR release → … → Invoice creation	125	happy_path
JI → JQ → PR → RAT creation → …	42	variation
JI → PR → JQ → PO creation → …	18	variation

Use this as a template; adjust columns or formatting .

6. Extensions & Next Steps

- **Variation Analysis: Drill into the most common deviations to pinpoint process bottlenecks or rework loops.**