

Records Intake & Data Quality Review

A practical SQL and documentation exercise that reviews fictional intake records for completeness, duplicate identifiers, retention readiness, and workflow status before controlled processing.

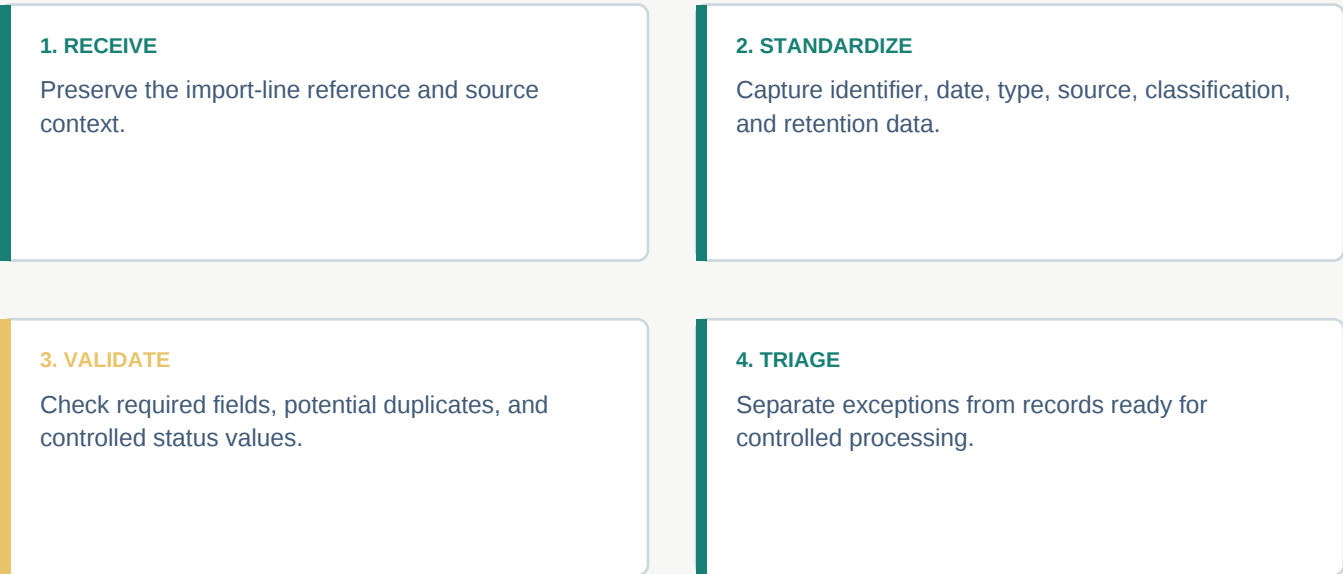
PROJECT SNAPSHOT

- 12 synthetic intake records and a documented field model
- Five repeatable SQL quality checks
- Completeness, duplicate-ID, retention, and status controls
- Downloadable source files and a generated exception register

WORKFLOW DESIGN

A controlled review before processing

The project treats record intake as a quality-control workflow. Each imported record is reviewed before it is indexed, reported on, or moved into a controlled processing state. The emphasis is traceability: each exception should be visible, explainable, and actionable.



Core fields and review rules

FIELD	PURPOSE	REVIEW RULE
Record ID	Unique record reference	Potential duplicates require review
Received date	Entry into the workflow	Required before controlled processing
Classification	Functional category	Required for organized handling
Retention code	Retention reference	Required before indexing
Workflow status	Handling state	Limited to controlled values

SQL QUALITY CONTROLS

Make exceptions visible and repeatable

The readiness query uses a common table expression to identify duplicate IDs, then applies a prioritized CASE expression to assign a review outcome. The logic turns a manual quality-review checklist into a repeatable SQL control.

```
WITH duplicate_ids AS (  
  SELECT record_id  
  FROM record_intake  
  GROUP BY record_id  
  HAVING COUNT(*) > 1  
)  
SELECT import_line, record_id, classification, retention_code,  
  CASE  
    WHEN record_id IN (SELECT record_id FROM duplicate_ids)  
      THEN 'Exception: potential duplicate'  
    WHEN TRIM(COALESCE(received_date, '')) = ''  
      THEN 'Exception: missing received date'  
    WHEN TRIM(COALESCE(classification, '')) = ''  
      THEN 'Exception: missing classification'  
    WHEN TRIM(COALESCE(retention_code, '')) = ''  
      THEN 'Exception: missing retention code'  
    WHEN workflow_status NOT IN ('Ready', 'Indexed', 'Pending Review')  
      THEN 'Exception: invalid workflow status'  
    ELSE 'Ready for controlled processing'  
  END AS review_outcome  
FROM record_intake  
ORDER BY import_line;
```

WHY THIS MATTERS

The query makes review decisions consistent and prevents incomplete or ambiguous records from appearing as ready.

WHAT IT DEMONSTRATES

CTEs, grouping, conditional logic, data validation, exception reporting, and documentation of review rules.

REVIEW OUTCOME

Evidence of a completed quality check

The synthetic sample contains 12 records. The readiness review produces six flagged conditions: missing classification, missing retention code, missing received date, two duplicate-ID entries, and an invalid workflow status. Records without an exception are marked ready for controlled processing.



Exception register excerpt

LINE	RECORD ID	ISSUE	RECOMMENDED ACTION
3	REC-2003	Missing classification	Assign an approved functional category.
4	REC-2004	Missing retention code	Apply the appropriate retention reference.
7	REC-2007	Missing received date	Confirm the intake date before indexing.
9	REC-2009	Potential duplicate identifier	Compare source rows and retain the authoritative record.
10	REC-2009	Potential duplicate identifier	Compare source rows and retain the authoritative record.
11	REC-2010	Invalid workflow status	Set the status to an approved controlled value.

Professional relevance

This project demonstrates an approach I can explain in an entry-level information-management, records, operations, or data-support context: define the fields, document the rules, make exceptions visible, and preserve a clear path from intake to controlled processing.