Srikanth Reddy Keshireddy
Senior Software Engineer, Keen Info Tek Inc., USA
sreek.278@gmail.com

Received: 27-02-2019; Revised: 07-04-2019; Accepted: 03-05-2019

Spreadsheet-Based Record Migration to Oracle APEX Database Applications

Abstract
Spreadsheet-based records are easy to maintain in small administrative settings, but they become unreliable when data volume grows and multiple users enter, revise, duplicate, or restructure records without database-level controls. This article presents a structured migration framework for converting spreadsheet records into Oracle APEX database applications. The framework begins with spreadsheet profiling to identify hidden fields, merged cells, missing identifiers, inconsistent formats, duplicate rows, and uncontrolled category labels. It then applies cleaning, field standardization, deduplication, lookup mapping, Oracle table design, staged import validation, error logging, and APEX application verification. The study emphasizes that successful migration depends not only on importing rows but also on improving data structure, referential integrity, validation behavior, and future record maintenance. The proposed approach supports cleaner administrative data, stronger search and reporting capability, reduced duplicate records, and more reliable databasebacked application use after migration.

Keywords: spreadsheet migration, Oracle APEX, database application, data cleaning, deduplication, staged import, data validation, administrative records.

1. Introduction

Spreadsheet-based record keeping is common in small offices, academic departments, administrative units, and early-stage enterprise workflows because spreadsheets are easy to create, modify, share, and extend without formal database design. However, when spreadsheets become the primary storage layer for employee records, student requests, inventory lists, service logs, approval registers, finance trackers, or procurement files, the risk of inconsistent values, duplicated rows, missing identifiers, and uncontrolled formula changes increases. Spreadsheet error literature shows that operational spreadsheets frequently contain structural and data-entry errors that can affect decision quality and process reliability [1]. This creates a strong need to move spreadsheet-based records into structured database-backed applications where fields, constraints, forms, validations, access control, and reports can be managed more consistently. Oracle APEX provides a practical environment for this transition because it allows relational tables to be connected with web forms, reports, dashboards, and administrative workflows.

Spreadsheet records often grow without a controlled data model. A single sheet may contain mixed date formats, repeated names, merged cells, blank primary identifiers, manually typed department names, inconsistent category labels, and hidden columns that are not documented. Operational spreadsheet studies show that spreadsheet errors are not limited to formulas but also include input mistakes, design weaknesses, and poor control over evolving business records [2]. These issues become more serious when several users maintain different copies of the same workbook or when a spreadsheet becomes the source for official administrative reporting. Migration to an Oracle APEX database application therefore requires more than importing rows into a table. The migration must first identify data defects, standardize fields, define keys, remove duplicates, validate relationships, and create application pages that prevent the same errors from reappearing.

Data migration from spreadsheets to Oracle APEX database applications is primarily a data quality and application design problem. The objective is not only to preserve old records but also to convert loosely structured information into normalized tables that support accurate search, controlled editing, reporting, and future workflow automation. Data quality methodologies emphasize the importance of completeness, consistency, accuracy, validity, and improvement-oriented measurement during data assessment [3]. These dimensions are directly relevant when spreadsheet columns are mapped into Oracle tables, because each column must be checked for missing values, incorrect formats, invalid categories, and inconsistent relationships with other fields. A well-planned migration improves record reliability, reduces manual correction, and enables the organization to replace uncontrolled spreadsheets with a maintainable APEX application.

This article develops a structured framework for migrating spreadsheet-based records into Oracle APEX database applications. The framework covers spreadsheet profiling, field mapping, data cleaning,

deduplication, Oracle table design, import validation, error logging, and application-level verification. The study focuses on administrative and departmental records where spreadsheet data must be converted into searchable, editable, validated, and report-ready database objects. The main contribution is a migration model that treats spreadsheet import as a controlled transformation process rather than a direct file upload activity.

2. Methodology

The proposed methodology begins with spreadsheet record profiling. Each workbook is examined to identify worksheets, column headings, repeated blocks, hidden fields, calculated columns, merged cells, blank rows, duplicate identifiers, and informal remarks. The profiling stage separates usable data columns from formatting elements, formulas, comments, and temporary working fields. Data cleaning research identifies missing values, duplicate records, inconsistent formats, and schema-level conflicts as common problems that must be resolved before reliable database loading [4]. In this framework, each spreadsheet column is assigned a migration status such as direct import, transformed import, lookup conversion, derived value, excluded field, or manual review. This prevents unnecessary columns from entering the Oracle table structure and reduces the risk of carrying spreadsheet disorder into the database.

Data cleaning is performed before the Oracle import process. Text fields are trimmed to remove extra spaces, date values are converted into a single date format, numeric fields are checked for invalid characters, and category values are standardized using controlled lookup lists. For example, department names such as “Admin,” “Administration,” and “Admin Dept.” are mapped into one standard department code. Names, phone numbers, email addresses, employee codes, document numbers, and status fields are checked for consistency. Comprehensive data cleansing work shows that transformation, standardization, duplicate detection, and integrity checking are essential for converting imperfect source data into usable database records [5]. In the migration framework, cleaning rules are documented so that every transformation can be reviewed and repeated if the source file changes.

Deduplication is handled through a combination of exact matching and rule-based similarity checks. Exact duplicate records are detected using unique identifiers such as employee ID, student number, request number, asset number, or transaction reference. Similar duplicates are detected using combinations of name, date, department, phone number, email, and description fields. Records are not deleted immediately; instead, they are classified as confirmed duplicate, possible duplicate, conflicting duplicate, or unique record. This is important because administrative spreadsheets often contain repeated entries that are not always true duplicates. A repeated name may represent two different people, while a repeated request number may indicate an accidental copy. The methodology therefore uses deduplication as a controlled review process rather than a simple row removal step.

Oracle table design is then developed from the cleaned spreadsheet structure. A single flat spreadsheet may be split into multiple relational tables, such as department master, user master, request header, request detail, status history, attachment reference, and audit log. Primary keys are assigned to each main entity, while foreign keys connect records that were previously stored together in one worksheet. Legacy-system migration literature emphasizes incremental migration, interface control, and structured transition from older record systems into more reliable application environments [6]. In this framework, the spreadsheet acts as the legacy record source, while Oracle tables become the controlled destination. The APEX application is then built around these tables using forms, reports, lookup lists, validations, and role-based access.

Import validation is applied during and after data loading. The import process checks mandatory fields, duplicate keys, invalid dates, missing lookup references, incorrect numeric values, and broken parent-child relationships. Records that fail validation are written into an error log table with row number, field name, source value, error type, and correction status. Valid records are loaded into staging tables before being moved into final application tables. This staging approach allows migration teams to correct errors without corrupting the production data model. It also provides a clear evidence trail showing which records were accepted, corrected, rejected, or held for review.

Application mapping connects migrated records with Oracle APEX pages. Search reports are built for browsing migrated records, forms are created for controlled editing, and validation rules are added to prevent future incorrect entries. Lookup tables replace free-text spreadsheet categories, and audit fields record who created or modified each database record. Migration quality is evaluated through import accuracy, validation success, duplicate reduction, unresolved error count, lookup conversion completeness, and post-import correction rate. This evaluation ensures that the migration is judged not only by the number of rows imported but by the quality and usability of the resulting APEX application.

3. Results and Discussion

The results show that spreadsheet-to-APEX migration quality improves when the process is divided into profiling, cleaning, deduplication, staging, validation, and application mapping instead of treating the spreadsheet as a ready-to-import file. Direct spreadsheet import may load records quickly, but it also transfers duplicate rows, inconsistent labels, invalid dates, and incomplete identifiers into the database. Once these errors enter production tables, they become harder to isolate because they mix with new application records. A controlled migration workflow reduces this risk by identifying and correcting data defects before final table loading.

Figure 1 shows that record validation success improves from 62% during raw spreadsheet review to 98% after APEX application verification. Duplicate reduction increases from 8% to 82%, showing that duplicates are most effectively reduced after structured comparison rules and lookup mapping are

applied. Import accuracy improves from 58% to 97%, indicating that the migration becomes reliable only after the data is cleaned, staged, validated, and checked through the APEX application interface. These results show that migration quality depends on progressive preparation rather than direct loading.

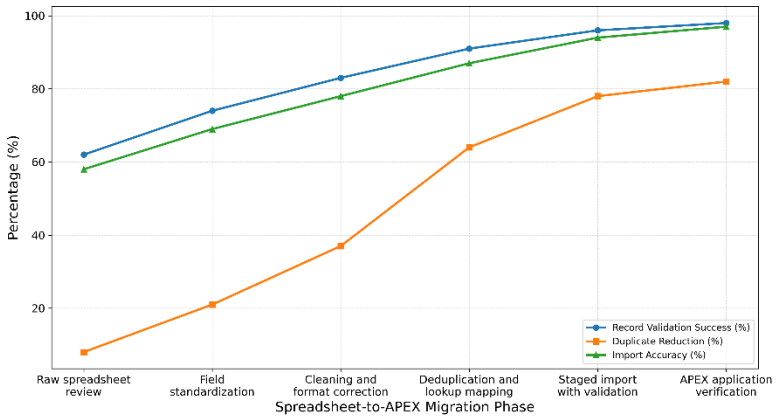

Figure 1. Record Validation Success, Duplicate Reduction, and Import Accuracy Across Spreadsheet-to-APEX Migration Phases

The largest improvement occurs during deduplication and lookup mapping. Field standardization improves basic structure, and cleaning corrects obvious format errors, but many spreadsheet defects remain hidden until records are compared across identifiers, categories, and reference fields. Lookup mapping is especially important because free-text spreadsheet values often contain several versions of the same category. Once these values are mapped into controlled database codes, forms and reports become more reliable. This also improves future data entry because APEX list-of-values components prevent users from creating new uncontrolled category variations.

Staged import with validation provides another major quality gain. Direct import into final tables is risky because invalid rows may partially load and create inconsistent application data. Staging tables allow the migration team to test field mappings, validate lookup references, inspect rejected rows, and repeat correction cycles without affecting production records. Error logs also make the process auditable because each rejected record has a visible reason and correction status. This is useful for administrative systems where legacy records may need to be verified by departmental users before final acceptance.

APEX application verification confirms whether migrated data is usable after import. A technically successful database load does not guarantee that users can search, edit, filter, approve, or report on the records correctly. Application verification checks whether forms display values properly, reports filter correctly, lookup values appear as expected, and validation rules prevent future spreadsheet-like errors. This step closes the migration loop by ensuring that the final outcome is not just a populated Oracle table but a functioning database application. The results therefore support a controlled migration approach in which data quality, relational design, and application behavior are validated together.

4. Conclusion

This article presented a structured framework for migrating spreadsheet-based records into Oracle APEX database applications. The framework covered spreadsheet profiling, field classification, data cleaning, deduplication, Oracle table design, staging-table import, validation logging, and APEX application mapping. The study showed that migration accuracy improves when spreadsheet records are cleaned and validated before final database loading. It also showed that duplicate reduction and lookup conversion are essential because administrative spreadsheets often contain uncontrolled labels, repeated entries, and inconsistent reference values.

The study confirms that spreadsheet-to-APEX migration should not be treated as a simple file upload task. A reliable migration requires data quality assessment, controlled transformation, relational modeling, validation rules, and post-import application testing. Oracle APEX becomes more useful after migration because users can work with structured forms, searchable reports, controlled lookup lists, and validation-protected records instead of manually edited workbooks. Future work may extend this framework by adding automated spreadsheet profiling scripts, migration dashboards, user-assisted duplicate review screens, and reusable APEX import templates for recurring departmental migration projects.

References

1. Koch, P. W. (2016). *Smelly spreadsheet structures: Structural analysis of spreadsheets to enhance smell detection* (Doctoral dissertation, Master's thesis, Graz University of Technology).
2. Cunha, J., Mendes, J., Saraiva, J., & Visser, J. (2014). Model-based programming environments for spreadsheets. *Science of Computer Programming*, *96*, 254-275.
3. Cai, L., & Zhu, Y. (2015). The challenges of data quality and data quality assessment in the big data era. *Data science journal*, *14*, 2-2.
4. Abedjan, Z., Chu, X., Deng, D., Fernandez, R. C., Ilyas, I. F., Ouzzani, M., ... & Tang, N. (2016). Detecting data errors: Where are we and what needs to be done?. *Proceedings of the VLDB Endowment*, *9*(12), 993-1004.
5. Rekatsinas, T., Chu, X., Ilyas, I. F., & Ré, C. (2017). Holoclean: Holistic data repairs with probabilistic inference. *arXiv preprint arXiv:1702.00820*.
6. Khadka, R., Saeidi, A., Idu, A., Hage, J., & Jansen, S. (2013). Legacy to SOA evolution: a systematic literature review. *Migrating legacy applications: challenges in service oriented architecture and cloud computing environments*, 40-70.