# Changes from updated\_views.py to new\_updated\_views.py

## 1. Modularization: Code Split into Helper Functions

Before: All logic was in one block inside upload\_csv.  
After: Each model (e.g., SDKData, Target) now has a dedicated store\_\*() helper function.  
Why: Improves readability, maintainability, and debugging.

## 2. Correct ER Diagram Order

Before: Order was implicit and not enforced.  
After: Follows strict order: SDKData → SW\_Details → Target → RunDetails → Model → Config → ECC → Results.  
Why: Ensures foreign key relationships are respected.

## 3. Explicit Use of .save()

Before: Used create() or get\_or\_create(), sometimes skipping .save().  
After: All models are instantiated and explicitly saved using .save().  
Why: More control and clarity in object lifecycle.

## 4. JSON Parsing for Stats Fields

Before: JSON fields were passed as plain strings.  
After: Uses parse\_json() utility to safely load JSON fields.  
Why: Ensures proper use of Django's JSONField.

## 5. Validation of Foreign Key Lookups

Before: Minimal checking before inserting.  
After: Uses .filter().first() to check if object exists before saving.  
Why: Prevents duplicate records.

## 6. Improved Error Handling

Before: One large try-except block.  
After: Smaller scoped try blocks in helpers (if needed), easier to debug.  
Why: Better traceability of errors.

## 7. Cleaner View Logic

Before: View mixed business logic with file handling.  
After: View is a clear sequence of operations using helper functions.  
Why: Easier to read and maintain.

This document summarizes the key improvements made in the refactored view function, ensuring alignment with the ER diagram and best coding practices.