**Student Data Import Script Documentation**

**Overview:**

This Python script processes student data from an Excel file and imports it into a MySQL database. It handles student information, course registrations, transactions, and user account creation while providing detailed logging of skipped records and progress tracking.

**Requirements:**

- Python 3.6+

- Required packages:

- pandas

- pymysql

- openpyxl

- tqdm

**Configuration:**

**File Paths:**

- `excel\_file\_path`: Path to the input Excel file ('studentdata.xlsx')

- `sheet\_name`: Worksheet name to read ('Sheet1')

- `table\_name`: Target table for student courses ('students\_courses')

- `skipped\_output\_file`: CSV file for skipped student records ('skipped\_students.csv')

**Database Connection:**

connection = pymysql.connect(

host='127.0.0.1',

user='xxxxx',

password='xxxxxuixxxxx',

database='xxxxx'

)

**Main Functionality:**

**1. Data Loading and Preparation**

- Reads Excel data into a pandas DataFrame

- Cleans column names (removes spaces, converts to lowercase)

- Handles null/NaN values

**2. Database Setup**

- Creates the `students\_courses` table if it doesn't exist

- Fetches reference data from database:

- Faculty mappings

- Department mappings

- Course code to ID mappings

**3. Data Processing**

- Processes each student record with progress bar visualization

- Extracts and normalizes:

- Student names (surname, firstname, middlename)

- Level and mode (regular/DE)

- Gender

- Semester information

- Course status

**4. Database Operations**

**Student Course Records**

- Inserts processed records into `students\_courses` table

- Tracks skipped records due to missing or invalid data

**Student Master Records**

- Creates new student records in `tbl\_students\_master\_test` if they don't exist

- Uses faculty and department mappings to ensure referential integrity

**Transaction Records**

- Creates registration transactions in `tbl\_students\_transactions\_test`

- Generates unique keys to prevent duplicates

**User Accounts**

- Creates student user accounts in `student\_users\_test`

- Uses surname as default password (falls back to matric number if surname missing)

**Course Registrations**

- Registers students for courses in `tbl\_course\_registered\_test`

- Handles course units and department mappings

**5. Logging and Reporting**

- Collects skipped records throughout the process

- Compiles all skipped records into a single Excel file (`import\_logs.xlsx`) with separate sheets for:

- Skipped students

- Skipped transactions

- Skipped user accounts

- Skipped course registrations

- Provides a detailed summary report including:

- Total records processed

- Successful inserts

- Skipped records count

- Output file location

**Usage:**

1. Run the script:

2. When prompted, enter the session\_id\_fk (foreign key for the academic session)

3. The script will:

- Show progress bars for each processing stage

- Display a summary report upon completion

- Generate an Excel log file with all skipped records

**Error Handling**

The script includes comprehensive error handling that:

- Skips invalid records while preserving the data

- Logs all skipped records with reasons

- Continues processing after non-critical errors

- Provides clear feedback about any issues encountered

**Customization Points**

1. Database connection parameters

2. Input file path and sheet name

3. Output file names and locations

4. Field mappings (status\_map, level\_map)

5. Table names for target databases

**Performance Notes**

- Uses tqdm for progress visualization

- Processes records in batches where possible

- Maintains database connections efficiently

- Optimizes queries with appropriate indexes

**Output Files**

- `import\_logs.xlsx`: Consolidated log of all skipped records

- Individual CSV files for each skip category (deleted after consolidation)