

Assignment: School Onboarding & Data Population via Excel

Primary Objective: Using the provided database schema and a set of Excel files as input, you are expected to configure and onboard a complete school setup into a **clean database** with blank tables. Your solution must include a script (in Python or SQL) that reads data from Excel files and populates all the corresponding tables while maintaining referential integrity.

✓ Part A: Database ETL Automation

- Read Excel sheets for: students, teachers, attendance, homework, class_diary, timetable, fees, salary, etc.
- Load into a clean relational database using scripts or tools
- Ensure 100% coverage of the functional requirements (school setup, homework status, attendance >80%, salaries, etc.)

1. School Setup

- Create **1 school** with:
 - **3 Grades** (e.g., Grade 6, 7, 8)
 - **2 Sections per grade** (e.g., A and B)

2. Students

- Each section should have **10 students**, totaling **60 students**.
- Ensure **overall attendance ≥ 80%** across the school for a month.
 - Insert attendance records accordingly from the sheet.

3. Teachers – Read the sheet and insert the data

- Create **8 teachers**.
- Each teacher must:
 - Be assigned to **at least 2 sections**
 - Have **subjects** assigned
 - Have **2 class diary entries**
 - Have **3 homework entries**:
 - 1 in Pending
 - 1 in Submitted
 - 1 in Completed
 - Have All the lesson plans created and HW are assigned to it as well:

4. Timetable – Read the sheet and insert the data

- Weekly timetable per section:
 - Clearly show **lunch breaks** and **free periods**
 - Reflect any teacher **on leave** in the timetable

5. Fees – Create your own insert script and insert in the relevant table

- Each student must:
 - Have at least **1 fee installment paid**
 - Payment must reflect in **school income** module

⚙ Your Task:

1. **Keep the database schema clean:**
 - Only Insert TABLE statements — **no hardcoded insert data.**
2. **Write a script** (preferably in Python using Pandas and SQLAlchemy or pgAdmin scripts) that:
 - Reads the provided Excel sheets
 - Validates basic fields and foreign keys
 - Inserts all relevant records into appropriate tables
3. You may write **individual scripts per sheet** or a **single loader** with modular functions.

Submission Format:

- create_tables.sql — All required tables (with audit columns)
- data_loader.py or java or sql — Excel to DB loader script
- sample_excels/ — Folder with expected Excel formats (with dummy rows)
- README.md — Describe:
 - Folder structure
 - How to run the loader
 - Any changes made
 - How all criteria are met (point-wise)

What You Can Change:

- You **may alter/add tables**, if:
 - Justified properly in README
 - Used in a clean, normalized, and efficient way

Reports to Verify:	Criteria	Weight
1. Attendance > 80% across school	Data completeness and correctness	25%
2. Homework per teacher: exactly 3 entries in required states	Excel-driven ETL process	25%
3. Class diary : 2 entries per teacher	Referential integrity maintained	15%
4. Fees collected → reflect in income module	Logical data setup & attendance/fees/salary coverage	15%
5. Salary payslip : for each teacher for June and July	Code quality, modularity & documentation	10%
6. Timetable shows breaks, leaves, and assignments clearly	Clarity in timetable and leave reflections	10%

All Table Descriptions (Full List)

1. **ss_t_schools**
Stores basic school metadata like name, type, contact details, and location.
2. **ss_t_grade**
Contains grades offered by each school with active flags and display order.
3. **ss_t_section**
Holds sections (e.g., A, B) assigned under specific grades.
4. **ss_t_subject**
Defines academic subjects taught in the school.
5. **ss_t_teacher**
Stores teacher profiles including contact info, gender, and qualifications.
6. **ss_t_teacher_section_map**
Maps teachers to specific sections, grades, and subjects.
7. **ss_t_class_timetable**
Stores class-wise period schedules including breaks and teacher assignments.
8. **ss_t_class_diary**
Records daily class logs by teachers including activities and topics covered.
9. **ss_t_homework_details**
Maintains homework assignment records with status and submission tracking.
10. **ss_t_student**
Contains complete student profiles including demographics and parental info.
11. **ss_t_student_academic_map**
Links students to their grade, section, and academic year.
12. **ss_t_lesson_plan**
Stores structured academic lesson plans per subject and class.
13. **ss_t_lesson_plan_topic_map**
Maps individual lesson plans to the specific topics taught.

14. **ss_t_behavior_log**
Logs student behavior entries categorized as positive, neutral, or negative.
15. **ss_t_feedback_category**
Lists feedback categories for various stakeholder inputs (e.g., academic, behavioral).
16. **ss_t_feedback_questions**
Stores structured feedback questions grouped under categories.
17. **ss_t_student_fee_structure**
Maintains assigned fee structure and components for each student.
18. **ss_t_fee_payment_installment**
Tracks student fee payments with installment, date, amount, and method.
19. **ss_t_school_income**
Stores income transactions for the school, often fed by fee collection.
20. **ss_t_teacher_salary_structure**
Defines the breakdown of salary components per teacher.
21. **ss_t_teacher_salary_payslip**
Records monthly salary payslips including gross, deductions, and net pay.
22. **ss_t_leave_type**
Defines types of leaves available to staff (e.g., casual, sick, paid).
23. **ss_t_teacher_leave_application**
Records teacher leave applications and approval status.
24. **ss_t_task_type**
Lists various task types (e.g., lesson planning, supervision).
25. **ss_t_teacher_tasks**
Stores individual task assignments with descriptions and status.
26. **ss_t_attendance_type**
Defines possible attendance states (e.g., Present, Absent, Late).
27. **ss_t_attendance_register**
Captures daily attendance entries per student with type and remarks.
28. **ss_t_student_guardian_map**
Links students to their guardians with relationship and contact info.
29. **ss_t_guardian**
Stores personal details of guardians including contact and profession.
30. **ss_t_event_type**
Classifies event types (e.g., Sports Day, Annual Function).
31. **ss_t_event_master**
Stores event-level details like title, date, venue, and type.
32. **ss_t_event_participants**
Lists students participating in school events.
33. **ss_t_transport_vehicle**
Stores data about school transport vehicles (bus number, capacity).
34. **ss_t_transport_route**
Stores route info with start, end points, and distance.
35. **ss_t_transport_stop**
Defines transport stops and sequence along a route.
36. **ss_t_transport_allocation**
Allocates students to specific transport routes and stops.
37. **ss_t_inventory_category**
Defines categories for school inventory (e.g., books, electronics).
38. **ss_t_inventory_item**
Lists inventory items with quantity and price.
39. **ss_t_inventory_issue_register**
Tracks inventory issued to students/staff.
40. **ss_t_inventory_stock_register**
Tracks inventory in/out stock movements.

41. **ss_t_asset_category**
Stores fixed asset categories (e.g., building, equipment).
42. **ss_t_asset_item**
Tracks specific asset items with code and purchase info.
43. **ss_t_asset_maintenance**
Logs maintenance schedules and records for school assets.
44. **ss_t_exam_type**
Lists types of exams (e.g., Unit Test, Final Exam).
45. **ss_t_exam_schedule**
Stores exam schedules per grade, subject, and section.
46. **ss_t_exam_marks_entry**
Contains student-wise marks entered for exams.
47. **ss_t_result_generation_log**
Tracks result generation activities and status.
48. **ss_t_student_report_card**
Stores final report card entries per student and term.
49. **ss_t_mcq_question_bank**
Stores MCQ-type questions with options and answers.
50. **ss_t_mcq_exam**
Contains metadata of MCQ exams (date, subject, class).
51. **ss_t_mcq_exam_question_map**
Maps questions to specific MCQ exams.
52. **ss_t_mcq_exam_submission**
Records student responses to MCQ exams.
53. **ss_t_mcq_exam_result**
Stores result summaries per student for MCQ exams.
54. **ss_t_announcement**
Contains announcements with visibility (staff/students).
55. **ss_t_student_disciplinary_action**
Stores disciplinary actions taken against students.
56. **ss_t_admission_enquiry**
Captures admission enquiries including status and follow-up.
57. **ss_t_admission_followup_log**
Tracks follow-ups on admission enquiries.
58. **ss_t_alumni_student**
Stores data of passed-out/alumni students.
59. **ss_t_staff_meeting**
Stores meeting schedules and agenda for school staff.
60. **ss_t_staff_meeting_minutes**
Captures meeting minutes and decisions taken.
61. **ss_t_library_book_category**
Categorizes library books by genre/type.
62. **ss_t_library_book**
Stores library book details including ISBN and shelf info.
63. **ss_t_library_transaction**
Records book issue and return transactions.
64. **ss_t_club**
Stores details of extracurricular clubs (e.g., Drama, Debate).
65. **ss_t_club_membership**
Lists student memberships in different clubs.
66. **ss_t_hostel_building**
Defines hostel buildings in the campus.
67. **ss_t_hostel_room**
Stores individual room info with capacity and type.

68. **ss_t_hostel_allocation**
Tracks room allocations to students.
69. **ss_t_school_gallery**
Stores school photo gallery with tags and event references.
70. **ss_t_exam_grading_scheme**
Stores grade ranges for exams (e.g., A for 90–100).
71. **ss_t_student_certificate_request**
Captures student requests for certificates (e.g., TC, Bonafide).
72. **ss_t_certificate_template**
Stores formats/templates for printable certificates.
73. **ss_t_certificate_issue_log**
Tracks issued certificates with date and reference number.
74. **ss_t_transport_fees**
Defines transport-related fee structure.
75. **ss_t_hostel_fees**
Defines hostel-related fee structure and mapping.
76. **ss_t_user_login**
Stores login credentials and roles for users (teachers, admins, etc.).
77. **ss_t_audit_log**
Tracks user actions and changes made in the system.
78. **ss_t_notification_log**
Logs push/email/SMS notifications sent to users.
79. **ss_t_system_config**
Stores global configuration flags and key-value settings.

Key Relationships

Format: **Child Table.Column → Parent Table.Column**

School Structure

- `ss_t_grade.school_id → ss_t_schools.school_id`
 - `ss_t_section.grade_id → ss_t_grade.grade_id`
 - `ss_t_subject.school_id → ss_t_schools.school_id`
 - `ss_t_teacher.school_id → ss_t_schools.school_id`
 - `ss_t_teacher_section_map.teacher_id → ss_t_teacher.teacher_id`
 - `ss_t_teacher_section_map.grade_id → ss_t_grade.grade_id`
 - `ss_t_teacher_section_map.section_id → ss_t_section.section_id`
 - `ss_t_teacher_section_map.subject_id → ss_t_subject.subject_id`
-

Academic Records

- `ss_t_class_timetable.school_id → ss_t_schools.school_id`
- `ss_t_class_timetable.grade_id → ss_t_grade.grade_id`
- `ss_t_class_timetable.section_id → ss_t_section.section_id`
- `ss_t_class_timetable.subject_id → ss_t_subject.subject_id`
- `ss_t_class_timetable.teacher_id → ss_t_teacher.teacher_id`

- `ss_t_class_diary.grade_id` → `ss_t_grade.grade_id`
 - `ss_t_class_diary.section_id` → `ss_t_section.section_id`
 - `ss_t_class_diary.subject_id` → `ss_t_subject.subject_id`
 - `ss_t_class_diary.teacher_id` → `ss_t_teacher.teacher_id`
 - `ss_t_homework_details.school_id` → `ss_t_schools.school_id`
 - `ss_t_homework_details.grade_id` → `ss_t_grade.grade_id`
 - `ss_t_homework_details.section_id` → `ss_t_section.section_id`
 - `ss_t_homework_details.subject_id` → `ss_t_subject.subject_id`
 - `ss_t_homework_details.teacher_id` → `ss_t_teacher.teacher_id`
 - `ss_t_student_academic_map.student_id` → `ss_t_student.student_id`
 - `ss_t_student_academic_map.grade_id` → `ss_t_grade.grade_id`
 - `ss_t_student_academic_map.section_id` → `ss_t_section.section_id`
-

Lesson Planning

- `ss_t_lesson_plan.school_id` → `ss_t_schools.school_id`
 - `ss_t_lesson_plan.grade_id` → `ss_t_grade.grade_id`
 - `ss_t_lesson_plan.section_id` → `ss_t_section.section_id`
 - `ss_t_lesson_plan.subject_id` → `ss_t_subject.subject_id`
 - `ss_t_lesson_plan.teacher_id` → `ss_t_teacher.teacher_id`
 - `ss_t_lesson_plan_topic_map.lesson_plan_id` → `ss_t_lesson_plan.lesson_plan_id`
-

Student & Fees

- `ss_t_student.school_id` → `ss_t_schools.school_id`
 - `ss_t_student_fee_structure.student_id` → `ss_t_student.student_id`
 - `ss_t_fee_payment_installment.student_id` → `ss_t_student.student_id`
 - `ss_t_fee_payment_installment.fee_structure_id` → `ss_t_student_fee_structure.fee_structure_id`
 - `ss_t_school_income.fee_payment_id` → `ss_t_fee_payment_installment.fee_payment_id`
-

Staff & Payroll

- `ss_t_teacher_salary_structure.teacher_id` → `ss_t_teacher.teacher_id`
 - `ss_t_teacher_salary_payslip.teacher_id` → `ss_t_teacher.teacher_id`
-

Leave & Tasks

- `ss_t_teacher_leave_application.teacher_id` → `ss_t_teacher.teacher_id`
- `ss_t_teacher_leave_application.leave_type_id` → `ss_t_leave_type.leave_type_id`
- `ss_t_teacher_tasks.teacher_id` → `ss_t_teacher.teacher_id`
- `ss_t_teacher_tasks.task_type_id` → `ss_t_task_type.task_type_id`