1. 数据库维护/更新说明（中英双语维护手册模板）

你可以直接粘贴到 README 的“维护指南”或新建 维护手册\_Maintenance.md。

数据库维护与更新操作说明

Database Maintenance & Update Manual

1. 结构变更（如增减字段、改表结构）

Schema Changes (e.g., add/delete columns, change table structure)

* 修改结构请先备份数据。
  + Always backup your data before making schema changes.
* 推荐在 pgAdmin/psql 内执行 ALTER TABLE 语句。
  + Use ALTER TABLE SQL commands in pgAdmin or psql.
* 典型结构变更示例：
  + Add column:
  + ALTER TABLE enrollments ADD COLUMN refund\_amount NUMERIC(10,2) DEFAULT 0;
  + Drop column:
  + ALTER TABLE students DROP COLUMN dropout;
  + 改字段类型：
  + ALTER TABLE enrollments ALTER COLUMN status TYPE VARCHAR(30);
* 结构更新后需同步更新相关 Python 代码、数据导入模板等。
  + After schema change, update Python scripts and CSV templates accordingly.

2. 数据备份与恢复

Backup & Restore

* 定期用 pgAdmin 或 pg\_dump 备份数据：
  + Use pgAdmin export or pg\_dump:
  + pg\_dump -U youruser -d yourdb > yourdb\_backup.sql
* 恢复数据用：
* psql -U youruser -d yourdb < yourdb\_backup.sql

3. 数据批量导入和清空

Batch Import & Data Reset

* 导入前建议先清空表（如重复导入多次可能会导致数据重复）：
  + Before batch import, clear tables if necessary:
  + TRUNCATE TABLE enrollment\_tas RESTART IDENTITY CASCADE;
  + TRUNCATE TABLE enrollments RESTART IDENTITY CASCADE;
* 执行导入脚本（import\_data.py）：
* python import\_data.py

4. 常见问题排查

Common Troubleshooting

* 无法连接数据库：请检查 db\_config.py 配置与 Postgres 账号/端口/数据库名是否一致。
  + Check your db\_config.py settings (database, user, password, host, port).
* 导入报错：通常为数据格式问题、缺失字段或主键冲突。
  + Most errors are due to data format issues, missing columns, or duplicate primary keys.
* 分析结果人数/利润异常高：请确认未重复导入、用 COUNT(DISTINCT ...) 做聚合。

5. 新增数据/修改数据操作

Add/Modify Data

* 可用 crud.py 里的相关函数添加、删除、更新报名/学员/助教等数据，或直接在 pgAdmin 手工操作。
  + Use functions in crud.py or operate in pgAdmin.

6. 版本管理和团队协作

Version Control & Collaboration

* 上传到GitHub后，所有结构变动、导入代码更新，务必同步更新README或维护手册。
  + After pushing to GitHub, keep the documentation up to date with any changes.
* 建议每次大更新前新建分支、合并后测试分析无误再上线。
  + Create a new branch for major changes and test thoroughly before merging.

7. 项目升级建议

Upgrade Suggestions

* 可逐步升级为支持Web管理（如Streamlit/Flask），或自动定时分析报表、可视化。
* 数据库可迁移到云端托管（如 AWS RDS、阿里云、腾讯云等）。

推荐维护流程图

1. 备份 → 结构变更 → 导入新数据 → 测试分析 → 更新文档 → 推送GitHub

2. 上传到 GitHub 后的建议

How to manage after uploading to GitHub:

* 新人/同事/未来的你，只需 clone 仓库，按 README 步骤即可复现和维护系统。
* 每次数据库结构或主要代码更新，建议新建分支开发，合并后同步维护说明。
* 可以用 GitHub Issues 记录 Bug、优化点，Pull Request 做协作。

如果你的项目要给外部/第三方用，可以考虑加一份简单的视频/截图操作说明，让人上手更快！

如需\*\*.gitignore模板\*\*、数据库备份自动化脚本、升级/协作FAQ，也可以要！