**工程介绍:**

2020年2月我来到一家公司进行实习，并接受了对一款报表工具进行测试的任务。该任务使用了foodmart这个数据库，由于字段全都是英文，所以在报表中参差不齐，看着很不美观。于是将数据库中的英文翻译成中文的想法油然而生。

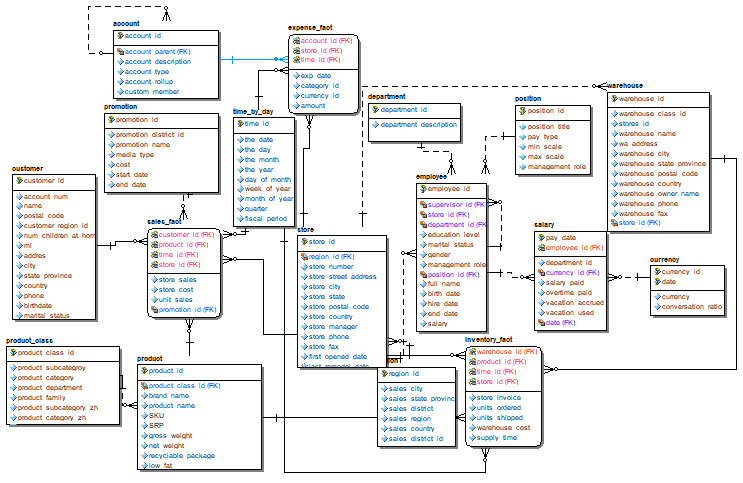
**思路：**

首先我决定用有道对foodmart数据库进行翻译，由于其中有大量的单词，所以我通过python脚本调用有道的URL（统一资源定位）接口。

有道对调用次数有一定限制，所以需要对数据库中的单词进行查重归并，这样就可以大大降低访问次数。

最后，为了方便其他人使用，python脚本必须是通用的程序。

**关于foodmart数据库：**

foodmart是旧版本的SQL Server所带的示例数据库，它模拟了一家大型的食品连锁店的经营业务所产生的数据，其中包括了客户管理数据、销售数据、分销数据和库存数据等。为了对其进行了解，下图是我绘制的er（实体关系）图。er图主要由实体、属性、关系三部分组成，通过一对一，一对多，多对多三种关系来表示表与表之间的联系。

如图：

* sales\_fact，expense\_fact，inventory\_fact表分别记录了该食品连锁店的销售状况，费用信息以及库存信息
* customer，product，store，time\_by\_day，promotion分别记录了顾客信息，产品信息，商店信息，时间信息以及促销活动的信息。account表记录了连锁店的账目信息。
* warehouse表记录了连锁店的仓库信息。product\_class表记录了食品的分类信息。
* Employee表记录了连锁店的雇员信息，
* department，position，salary表分别记录了连锁店的部门种类，每个职位的信息以及薪水情况。
* Currency表记录了每天货币的汇率。

每张表之间都有着决定性关系和非决定性关系，其中实线代表的是决定性关系，指主表的主键既是子表的外键也是子表的主键。虚线则表示非决定性关系，指主表的主键是子表的外键，且非空。从图中我们可以看出：

成决定性关系的有customer，product，store，time\_by\_day与sales\_fact；account，time\_by\_day，store与expense\_fact；store，time\_by\_day，warehouse，product与inventory\_fact。

成非决定性关系的有department，position，salary，store与employee；Currency与salary；promotion与sales\_fact；product\_class与product。

**使用说明：**

1. **应用环境准备**
   1. 安装python运行环境，版本3.7以上
   2. 然后在如下链接中下载代码压缩包，并解压

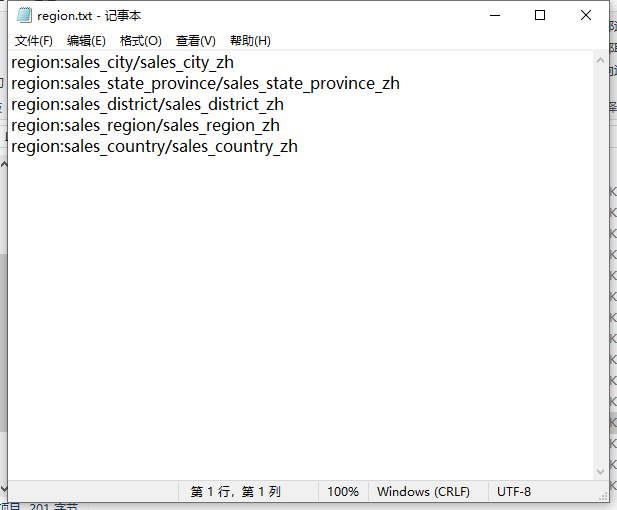
<https://github.com/zyf1999lj>

1. **运行程序**
   1. 修改region.txt文件，该文件记录了表的表名，数据库需要翻译的字段名（含待翻译的英文内容）以及翻译后的字段名（含翻译出的中文内容）。如图：

每行中“：”前的单词为表名，“：”后是一对字段名。

“/”前的单词为需要翻译的字段名，“/”后的单词为翻译后的字段名。

请根据你的实际情况替代这些内容。



* 1. 打开控制台或Windows PowerShell，进入trans\_region.py所在文件夹，输入python trans\_region.py mysql用户名 密码 region.txt地址 新建txt文件地址，运行程序后即可翻译数据库中的单词并将结果保存在txt文件中



* 1. 然后进入mysql\_zh.py所在文件夹，输入python mysql\_zh.py mysql用户名 密码 region.txt地址 之前生成的txt文件地址，运行程序即可将翻译结果填入数据库中



**小结：**

这个项目增强了我对数据库的理解，使我对python有了更深的体会。这里我将成果分享给大家，希望可以对大家有所帮助。

在此，我还要感谢给我实习机会的公司前辈。