# 个人贡献及工作内容

**作为开发人员，内容主要包括以下:**

-- 业务人员接受需求传导，并确定改修范围

-- 针对业务人员提出的问题，指导组员进行后台数据的调查(报告端和后台)

-- 参与需求实现方式的讨论，并决定解决方案的实现方式

-- 针对开发中所遇到的问题，自研开发了后台解析工具，极大的提高了项目组成员进行后台调查分析的效率

**作为Team Leader，负责团队管理，内容主要包括以下:**

-- 根据分配的任务，将任务细分，合理分配给组内成员，确保任务按时完成

-- 对组内成员，进行指导，主要包括业务知识的传授，技术工具的应用，及开发中遇到难题的解决

-- 根据数据仓库的设计规范，负责将需求转换为详细设计，并分配成员进行开发，同时代码review，进度监控，结果检验； 针对成员短板，进行短期培训补足

-- 带领组员，协助其他项目组，进行数据方面的协助，包括数据同步，数据清理等

**兼顾管理运维，内容主要包括以下:**

-- 上线操作，确保上线成功

-- 日常常规操作(job预警，数据库监控，报告提取，报告权限管理)

-- 后台ETL的临时紧急调整，障害原因调查

-- 与global第三方团队针对环境及配置问题，进行沟通协作，以及在service now系统提单

**工作涉及数仓相关业务，主要包括如下：**

--客户拜访相关：

主要负责了Coaching部分的开发及后期优化，团队销售逻辑调整，医生与坑号对应关系逻辑优化等，以及年度指标配置表的配置

--销量相关：

主要负责了销量整体ETL的性能优化调整工作，包括停更不必要的job，调整job顺序，以及针对复杂大量脚本的调优；

--奖金数据相关：

主要负责了奖金数据中，代表排名模块的开发

--会议数据相关：

主要负责了会议奖金报告后台ETL的开发，通过统计代表邀请医生参会数据，来计算不同的指标(包括各等级医生参会率，参会人数，参会人次等)

--组织架构数据相关：

主要负责架构表的调整及优化，根据上游架构数据系统的变更，随时调整数仓架构表逻辑，以及数据勘误

--报告权限数据相关：

主要负责了报告数据权限控制后台的调整及优化，比如数据权限的控制方式由代表ID调整为代表所在坑号

--主数据相关：

主要负责上游抽取过来的主数据的调整及优化，包括针对主数据追加时间限制来缩短加载时间，以及及时停更不再使用的主数据表

**工作中使用的技术及工具：**

--数据库： Oracle(精通，用于数仓),sqlserver(熟悉，用于外围系统)，postgresql(熟悉，用于数仓外围系统云端开发)

--编程语言： PL/SQL(精通)，python/java(熟悉，用于开发工具)

--ETL： Informatica(精通)

--报告端： Tableau(熟悉)，BO(熟悉)，Power BI(了解)

--AWS: glue/glue studio(熟悉，用于开发ETL)，cloudformation，cloudwatch，s3，lambda

--Linux: shell(熟悉)

# 自我介绍

My Name is Liye.Shi.

I graduated from Dalian University of Foreign Language in July, 2010.

My Major is information management and information system.

I studied Japanese and English in the college, I passed J-test in Japanese and CET-6 in English.

I received a bachelor's degree in management.

This is my education background.

I had worked in 2 companies since I graduated.

The first company was called Dalian Transcent Tecnology, I worked there for about 3 and half years on web development.

I worked as developer in team, my work includes system detail designing, Coding, Unit testing, documentation and also do some assistance work for project manager as subleader，like development progress managing and reporting.

I also worked as BSE to communicate with Japanese customers for requirement analysis and detail design.

I used java, html/CSS, JavaScript program language in the project.

The second company is where I am working at currently.

In this company I have joined 2 projects.

The first project is for Lilly Japan.

I worked as BSE, to communicate with Japanese customers.

I also worked on the data analysis, like data follow analysis, data impact analysis, data cleaning, data masking and some other data related task.

I used oracle SQL, procedure, informatica, Linux shell in the project.

In this period, I learned C# by myself and built a web application which used by business.

Also I learned and used data ming software, which called polyanalyst for advanced data processing.

The second project is for Lilly China, where I am working in currently.

I worked as ETL developer, and lead the SM team, the SM work includes reports/ETL job go online, job monitoring, performance issue resolving and some other SM task

Currently I work as the team leader, responsible for both ETL development and SM.

I am also responsible for ETL design and assign it to developers of my team.

My team had 3 members. One is for SM and the other 2 is for development.

I had confidence in Informatica, SQL, procedure, and familiar with BO, tableau, and I am learning power bi by myself now.

In this period, I developed 2 tools for the team.

One is for SM team for the operation automation, which developed in python.

The other tool is for development, it can analyze informatica workflow, which developed in java, this improves the work efficiency significantly, and I am proud of it.

Our DWH plan to go to AWS cloud this year, I am studying AWS related knowledge, and developed AWS glue for the upstream system of our DWH. I also familiar with other components of AWS, like s3, cloudwhatch, cloudformation and CLI.

I hope to be data architecture in the future.

# 如何设计数据仓库

## 数据仓库的整体架构：

files/ODS/上游系统DB 🡺 stg层 🡺 DW层 🡺 DM层 🡺 reporting/analysis/data mining

**stg：**原封不动保持原始数据，便于追溯

如果源系统性能有要求，stg层便于减少源系统的IO负载，提高用户体验

**DW层：**针对stg数据进行清洗，转换，stg数据来自不同的系统，比如命名，定义，格式会不统一，不规范，在DW层可以针对这些问题进行统一规范

还可以对一些脏数据进行清理，或更正

**DM层：** 面向主题对DW层数据进行进一步的处理，比如销量主题，CRM主题，会议主题

并且根据主题进行一些指标的计算，比如ytd，mtd，qtd计算，环比，同期比

## 从开发工程角度：

### 需求传导(概念模型)：

**内容**

确定需求，划清调整的范围和边界，通过会议等方式反复沟通、沟通技巧：引导客户，会问问题，问到点子上，通过客户的描述挖掘隐藏的需求

基本就是考虑where，when，what，how，

比如首先要考虑需求内容，改什么？涉及哪些报告，新建还是调整报告？涉及哪些业务模块？

加载是全量还是增量，刷新频次是每日，每周还是每月，还是手工刷新

此阶段应粗略估算出项目所需时间以及项目计划草案以及费用

**注意：**

* 注重全局的理解而非细节，过分注重细节会忽略大的方面的问题，会造成后续开发跑偏，只考虑需求，先不考虑具体细节的实现
* 需要对整体架构进行思考
* 搞好人际关系，可以避免相互扯皮
* 一定要使用业务术语，减少沟通障碍
* 确定业务用户间的逻辑保持一致

### 详细设计（逻辑模型）

**内容**

敲定具体的实现细节，涉及哪些ETL job？用ETL工具还是存储过程？改修涉及哪些DB实体？

**注意**

* 让高一级别的人做review
* 设计表结构一定基于真实数据，而不是自己模拟出来的数据
* 尽量使用既存模块成熟的模式
* 一定程度的弹性，为未来需求变更做准备
* 对表属性的描述要全面，约束条件，默认值（系统时间），值范围(虚拟医生 > 299999)
* 设计遵循命名规范，description字段就用dec缩写
* 总之，就是设计文档一定要尽可能的详细，做到让开发人员不要有理解上的偏差或者功能上的遗漏
* 文档的版本管理
* 评估数据量对性能的影响
* 建议使用3NF规范建模

### 开发（物理模型）

按照详细设计进行开发，有问题随时沟通

可能会修改详细设计

### 总结

整个过程，沟通和业务理解最重要，其次是数据库知识

最好能做到业务，BA，设计，开发，QA所有人对改修内容达成一致，这样就能充分发挥团队的力量

## 规范化

**一个过程：1NF -> 2NF -> 3NF -> BCNF -> 4NF -> 5NF**

**减少冗余：同一含义的数据只存一份**

**规范化之前**

**列出实体：名词**

**勾勒出关系：动词**

**尽量消除多对多的关系**

**列出所有属性**

**第一范式**

**原子性，没有重复列，列不可以再分，也没有重复的行**

**第一范式结果：**

**数据规整成二维表格**

**确保每一列表达的同一含义的数据**

**去掉多值属性，拆成多列**

**去掉重复组，挪到新表中去(不能出现相同含义的多列)**

**确定主键**

**第二范式**

**非主键属性依赖于整个主键，而不是其中的一部分**

**第二范式结果：**

**首先满足第一范式**

**如果非主键属性只依赖于主键的一部分，则需要移出创建新表**

**如果只有一个主键，则无需考虑第二范式**

**第三范式**

**非主键属性只依赖于主键，不能依赖于其他非主键属性**

**第三范式结果：**

**首先满足第二范式**

**如果非主键属性还依赖于其他非主键属性，则需要移出创建新表**

# 最具有挑战的事情

# 业务模块具体说明

## 主数据（上游）

**数据模块：**

城市省份信息，医生信息，客户信息（分销商，零售商，医院，药房，其他），产品，架构，TD，THC，THP，RPC

## 销量（上游）

**销量数据类型：**

Actual sales：销售数量和销售金额两种计算方式

销售金额是通过价格和销售数量计算，销售金额是跟销售价格走的，因为价格会随时调整，药监局审核通过后，销售金额要重新计算，因此销售金额一般是通过最新价格计算的

Quota sales：销售计划值

**数据模块：**

客户信息，产品信息，销售组织架构信息，sleeve代表类型信息，时间信息，地理信息

**报告包括：**

Hospital Sales analysis，医院组的销量，按照年月，季度，年，还有大区，主管，代表这些粒度进行分析

Retail Sales analysis，零售组进行分析，目前停止更新了

Brand Sales analysis，按照产品大类，品牌进行分析

Cpex &Intentive sales analysis，大排名 奖金系统，根据销量进行奖金的分配

Assign/Unassign sales analysis，thc，thp里存在的就是assign，unassign就是销量没有指定到某个代表或坑号

## CRM（上游）

**数据模块：**

医生信息，医院信息，药房信息，产品信息，销售组织架构信息，时间信息

**上游系统：**

Veeva/NGMS

**关于拜访：**

代表针对某一种产品，对医生进行拜访

**关于团销：**

针对一群医生，和一组产品进行拜访销售，拜访医生是要至少3个医生，同时也会产生费用NGMS

**关于Coaching：**

主管带着代表，针对医生进行拜访，并进行指导

**报告包括：**

Call Activities Analysis，一次普通的拜访

Group Selling Activity Analysis，团销

Coaching Activity Analysis

KA Activity Analysis，大客户拜访，比较少，针对院长，副院长，主任级别客户

IVA Activity Analysis，幻灯片，视频播放

Call Related fees Analysis，费用明细

## 会议（上游）

**上游系统：**

Chief(China Integrated E-Form)

Veeva

Idoctor

**数据模块：**

参会者信息：来自idoctor，部分来自chief

讲者信息：来自chief

会议信息：来自chief，会议类型(面对面，网络会议)

产品信息

销售组织架构信息

时间信息

**报告包括：**

HCP program analysis: 关于医生的分析，program理解为会议，如场次数，参会人数

Meeting Fees Analysis：会议的费用，针对参会者和讲者，场地费，咨询费，餐费，差旅费，会议奖品

Speaker Activity Analysis：专门针对讲者进行分析，比如一年的费用多少，场次多少

Meeting Feedback Survey Analysis：会议结果的反馈状态，问卷调查，评价之类

## BenchMark（上游）

**平台介绍：**

主要提供了市场上，礼来同等药效的药品的竞品的数据，用于分析礼来产品的市场比重，及分析竞争对手

**上游系统：**

中国药协会/国药诚信：数据是到医院粒度的

IMS：数据是到制造商粒度的（礼来，辉瑞）

**模块：**

厂商信息，医院信息，产品信息，时间信息

**报告包括：**

Market Potential Analysis

* Competitor Analysis

## LCCP（上游）

**上游系统：**

LCCP(礼来糖尿病优行关爱项目 Lilly Connected Care Program)：

从第三方平台，如腾讯，丁香园抓取数据到系统，数据包括医生，患者，数据主要是糖尿病方面的

为医生提供卡片，然后医生将卡片给患者填写，卡片有一些表单，患者填写卡片后进行审核，如何通过则该患者入组，入组的意义可能就是患者买药，预约医生更方便些

我们分析数据主要是分析，医生患者入组情况，如果入组多，则说明医生活跃，该医生就会成为重点目标

一定时期内也会有有挂起，退组的情况，比如没有互动的话就挂起，挂起一定时间则退组

**数据模块：**

病人信息，医生信息，销售组织架构信息，时间维度信息

**报告：**

Patient Activity Analysis

Doctor Activity Analysis

Patient-doctor enrollment Analysis

## iDoctor（上游）

**上游系统：**

iDoctor：主要用于医生群体，医生通过该平台与代表进行互动，同时也可了解礼来相关动向，如新产品发布，组织活动，以及会议邀请等，

该平台也会跟其他系统进行同步及交互，现在也在微信平台可以使用了，包括扫码签到，扫码问答，问卷调查这些

idoctor里也包含了一个子平台，是礼信，主要针对代表，方便代表给医生推送文章，发送会议邀请之类的

中间会生成一些统计数据，比如代表推送文章次数，医生读取，打开文章次数，回应次数，收藏下载次数

该平台在国内比较领先，对于其他药企有很好的参照意义，其他药企还是传统的线下电话的方式

**数据模块：**

医生信息，销售组织架构信息，会议信息，产品信息，时间维度信息，content信息

**报告：**

医生认证激活分析

医生touchpoint分析，包括文章读取，收藏，下载，提问信息，页面点击信息，包括页面点击各个模块次数，

## 下游系统

**LCCI(lilly capability center india)：**

主数据，销量数据，会议数据，CRM，idoctor，lccp， m3

**Patient CRM（ipatient）：**

销售组织架构，代表类型，产品，THC，地理信息

主要针对患者，覆盖范围比lccp的患者更广

**ABC：**

销售组织架构，代表类型，产品，THP，TPP，Decile，销量数据

**INCENTIVE：**

销售组织架构，代表类型，医院组销量信息，客户信息

**SST：**

数据：销量信息，CRM，会议信息，CPA

Glue实现流程：

First, the cli should be installed on informatica server.

And then configure the communication channel between AWS and informatica server, this is not done by us, it is by global third party team.

This is the premise.

In the informatica workflow, we use cmd task to invoke cli by using linux shell scripts to upload files from informatica server to the specific location of s3.

Lamada function will be invoked once files had been uploaded to that specific location on s3, because we configured listeners there.

The lamada function will inovoke corresponding glue to fetch data from the files.

In glue, it will create dynamic data frame to read data from files.

And then it will do some small changes for the data, just like timestamp format conversion, deleting null value, filtering out dirty data.

We didn’t put complicated business logic here because we use glue just for straight load for staging