

The logo for the Database Technology Conference China (DTCC) 2013. It features the letters 'DTCC' in a bold, orange, sans-serif font. The 'D' and 'T' are connected, and the 'C's are also connected. Below the letters is a thin, curved orange line.

2013中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2013

大数据 数据库架构与优化 数据治理与分析

SequeMedia
盛拓传媒

IT168.com

ITPUB

ChinaUnix

A list of database technologies and concepts arranged in a 3D, isometric style. The text is white and set against an orange background. The items include: Database, BDaaS, flowingdata, DB2, NoSQL, MySQL, Oracle, and Big Data. The text is arranged in a way that suggests depth and perspective, with some items appearing to be on different planes or levels.

谈数据库相关规范

梁敬彬 福富软件



梁敬彬： 网名：wabjtam123 ITPUB版主、ITPUB社区专家、福富软件数据库专家。

新浪微博：weibo.com/wabjtam123
@梁敬彬_wabjtam123

规范交流从小高故事开始

今天演讲的主题就是两个字：**规范**。

听起来或许很枯燥，而且具体的技术细节，都已经体现在《数据库设计与开发规范》里了。我们就换一个角度，从身边的小故事说起。故事的主人公叫小高，是一个系统集成项目组的新人，从事DBA相关的工作一两年。 **这个故事不止是说明设计与开发规范，也不止是规范制定的讨论。**

希望大家：收获，不止规范。

1. 天啦，总算可以开始分析了

故事从某天早上10点，小高接到一个紧急救助电话开始。

“小高，我是小陈，河南XX系统的数据库挂了，请你尽快帮忙处理。多谢！”

“哦，好的。”放下电话，小高准备开始处理问题了。

咦，好像哪里有些不对劲。感觉少了什么？（ 愣啥？快打电话啊！ ）

“小陈，你那环境怎么访问啊……”

“哦，是这样这样访问滴……”

“好的，那我连连看。”电话又放下了 。

五分钟过后。

“小陈，你提供的数据库密码咋不对哩？”

“那我又怎么可以连上哩？”

“那我怎么知道你为什么又可以连上哩？”

五分钟过后。

整明白了，小高想的是数据库用户和密码，而小陈给的是主机的用户及密码。

五分钟过后。

“小陈，我可以登数据库正常操作，没挂啊？”

“哦，我说的挂，指的是前台访问数据库很慢。”

“我勒个去了，原来最遥远的距离不是苹果5代与5袋苹果，而是你我心中的挂啊……”
放下电话后，小高开始尝试用收集AWR报表的方法进行性能诊断。

十分钟过后。

“小陈，VPN中断(pause)了n次，名不虚传！我人快挂了，你帮我取AWR吧。”

“我不会啊。”

“那我电话里教你吧。”

三十分钟过后。

小高总算取到了报表，他放下电话，打开邮箱，收到了小陈传来的AWR报表。
天啦，总算可以开始分析了。

2. 艾玛，小高到底还要问多少次

二十分钟过后。

“小陈，是一些SQL需要优化，你帮我取一下它们的执行计划吧。”

“什么叫执行计划？”

“哦，MY GOD，你上次不是跟我说过研究数据库好多年了吗？”

“我是说我看了好多年数据库的书。”

“你……算了，还是我来教你取执行计划吧。”

二十分钟过去了。

“小陈，我崩溃了，你又错了，取的文件还是空的。”

“小高，我崩溃+1，要不你用QQ远程桌面连到我这来操作吧。”

十五分钟过去了。

小高终于取到了对应SQL的执行计划了，看着看着，他又拿起电话…….

“小陈，你帮我执行这个脚本，查查T1表的COL1列是否有索引？”

“好的。”

五分钟过后。

“小陈，再帮我看看T2表记录有多大，好吗？”

“好的。”

五分钟过后。

“小陈，最后帮我看看T3表是不是分区表，好吗？”

“好的。”

五分钟过后。

“小陈，最最后帮我看看T4表是什么类型的，我怀疑是全局临时表。”

“好的。”

N个五分钟过后。

“小陈，最最最……后帮我看看……。好吗？”
“不好。”

“咋啦，对我有意见？”

“不是我对你有意见，是我肚子对你有意见。”

艾玛，小高到底还要问多少次？

3. 崩溃，干脆老天让我也挂了吧

吃完午饭，他们下午继续奋战。

一个小时过后。

“小陈，我刚加了一些索引。你观察看看，系统有没有变快。”

“现在点前台界面，快多了！”

“欧耶，太好了，胜利！”

“小高，刚同事跟我说，就中午我们去吃饭的那会儿开始，系统已经正常了。”

“太好了，太……”小高忽然又察觉到哪里有些不对劲。（哎呀，这是我的功劳吗？

“小高，有人反映中午大致有20分钟左右时间入库进程一直失败，现在正常。”“哦，20分钟，现在正常了。”咦，停！咋又觉得哪里不对劲。（我的天，这20分钟，不就是我建索引的时长吗？）

第二天早上10点

“小高，我是小陈，河南XX系统的数据库又挂了！”

“拜托别再说挂了，是前台又点很慢吧。”

“是的，和昨天的情景一模一样。”

“崩溃，干脆老天让我也挂了吧。”

4. 师傅，您真是我们的救命菩萨啊

“师傅，总算请到您出马了，您请这里坐。”

“小高啊，这系统昨天出问题之前，是否都是正常？”

“哦，这倒没问，不清楚啊。”

“没问？那是某个菜单查询界面慢，还是所有前台界面都慢呢？”

“师傅，应该是都慢吧，哦，我不确定，要不我问问？”

两分钟过后。

“师傅，昨天以前系统是正常的，此外就小陈操作的那个前台查询界面会时快时慢，其他菜单都一直正常。小陈说他自己也是才了解到的。”

“那昨天以前有啥动作吗，比如打过补丁？如果有，问问他们是否允许回退。”

“好的，我再问问。”

两分钟过后。

“师傅，真是有打过补丁啊！他们说这个补丁允许回退。”

“那就回退吧，在相同数据量的测试环境下去测试一下，看看问题在哪里吧。”
一周过后。

“小高，上次那事怎么样，还有没有人再说那个什么来着？”

“挂？”

“哦，对！”

“补丁回退后，那个前台查询就一直都是正常的。”

“OK！对了，如果你当初前先在主机层面top一把，也就不会走弯路了。”

“哦，您有说过解决问题要有步骤，先整体，再局部，我都忘了。”

“是啊，你这个案例显然是局部故障。”

“师傅，您真是我们的救命菩萨啊！”

5. 小高，快松手啊，别拦着为师！

“小高，那问题查出原因了吗？”

“知道是由什么SQL引发的了，您看这条SQL，执行10分钟才出结果。”

“你知道你当时为什么没查出这条SQL吗？”

“我……”

“那是因为断点没生成。系统默认1小时采样，比如10点20出问题，由于11点没到系统没有生成11点的断点，你的报表也只有取到10点前，可惜10点前系统是正常的。”

“哦，怪不得，那我应该怎么办？”

“你可以当时手工生成断点，如：exec dbms_workload_repository.create_snapshot

“哦，对啊。”

“对了，你把SQL优化好了吗？”

“没啊，您看这条SQL，层层相套，300多行，执行计划有N次的表连接和全表扫描。”

“小高，我们一般不允许写超过100行的SQL，而且嵌套写法也一般不允许超过3层。”

“师傅，要改写吗？”

“是的，要尽量拆分，否则这以后无论扩展还是交给他人维护都不方便。”
一周过后。

“师傅，求您救命啊！”

“怎么办呢？”

“上次写那个超长SQL逻辑的人离职了，没人读懂这段代码，也不知道如何优化，客户已经催着要上补丁里的新功能模块了。”

“我忙不过来啊，可以推到下周帮你吗？”

一周过后。

“小高，昨天折腾我整一个通宵，理解该SQL的需求理解到快崩溃了，终于搞定了。你看：这里用CASE WHEN合并，减少了表扫描次数，那里我拆成另一条……现在速度没问题了，执行计划都很合理了。”

“哇，师傅，膜拜！”

“少来这套。对了，这里设计也有问题，T表有30GB这么大了，居然没建分区，你看居然还有3年前的数据，我了解到，只要保留3个月的记录够了。”

“师傅，那您的意思是？”

“建分区表，方便清理和转移数据，设计之初要对数据的大小及流向有规划。”

“感谢师傅，您就写个建议，我们到时候照您说的改造这个表结构。”

“对了，小高，我倒是有一个疑问？”

“您说。”

“为什么你们就早上10点击前台界面会慢，难道其他时刻点，不会调用这个SQL？”

“师傅，这里特殊，细节以后和您说。对了，有件事说出来，您要保持冷静哦。”

“啥事？”

“就是，写这个SQL的哥们，离职后昨天又杀回来了。”

“小高，快松手啊，别拦着为师！”

6. 拨云见天，您老就请出山吧！

一周过后。

“哦，小高，补丁上线没问题了啊，那就好，不过我觉的你们系统问题会很多。”

“都有什么问题啊？”

“你看你们那张有历史数据概念的30GB大的表，居然没建分区……”

“师傅，已经改造好了。”

“哦，你敢和为师打赌，再没第二张有类似情况的表了吗？”

“哎哟，那可不敢，我哪知道还有没有啊？”

“我还想跟你换个方式打赌，下个月开始，一年内不出现这情况你赢，你有把握吗？”

“有时间让我准备啊，说好了，赌一顿大餐。”

“没问题！”

“您可以把获取大小超过10GB，但是未建分区的表的脚本给我吗？”

“没问题！”

半年过后

“师傅，我输了。”

“知道输在哪里吗？”

“系统里符合条件的大表我之前都改造了，但是这是最近数据量增加上去。”

“我知道你一定要输。”

“为啥？”

“你应该让设计人员在设计之初就要根据业务情况，预估到数据量的未来发展趋势。”

另外你应该定时做巡检去发现问题。这里缺少了培训、宣贯和制度。”

“哦，师傅，受教了。”

“我觉得你们系统里的问题还多着呢。”

“哦，说的我怪紧张的。”

“难道数据库设计的问题中，就只有一个历史表分区改造的问题吗？难道开发中在的问题，就一个代码过长的问题吗？还有，你满意你上解决问题的次经历吗？”

“拨云见天，您老就请出山吧！”

探讨小高故事背后的问题



规范的讨论



制定规范



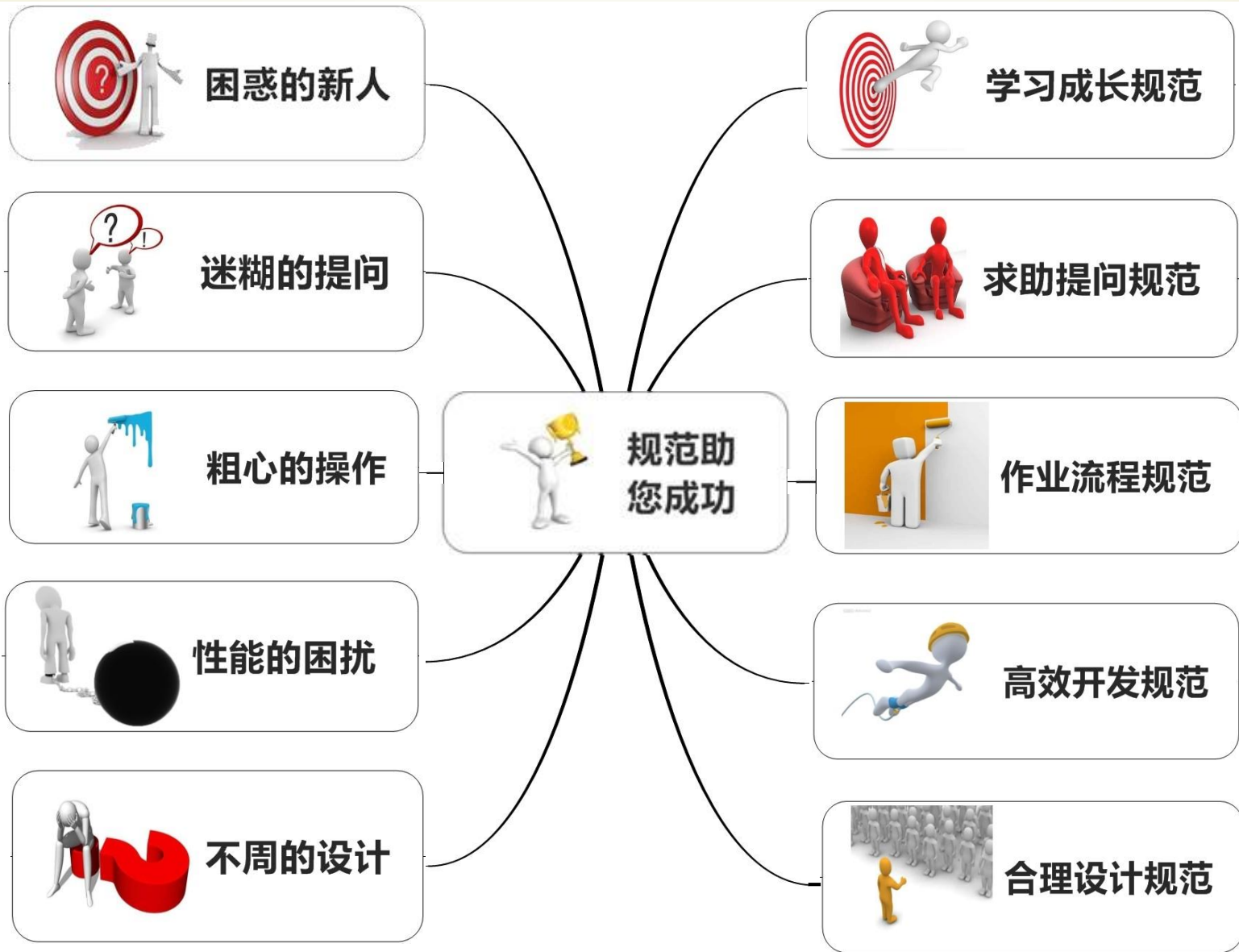
执行规范

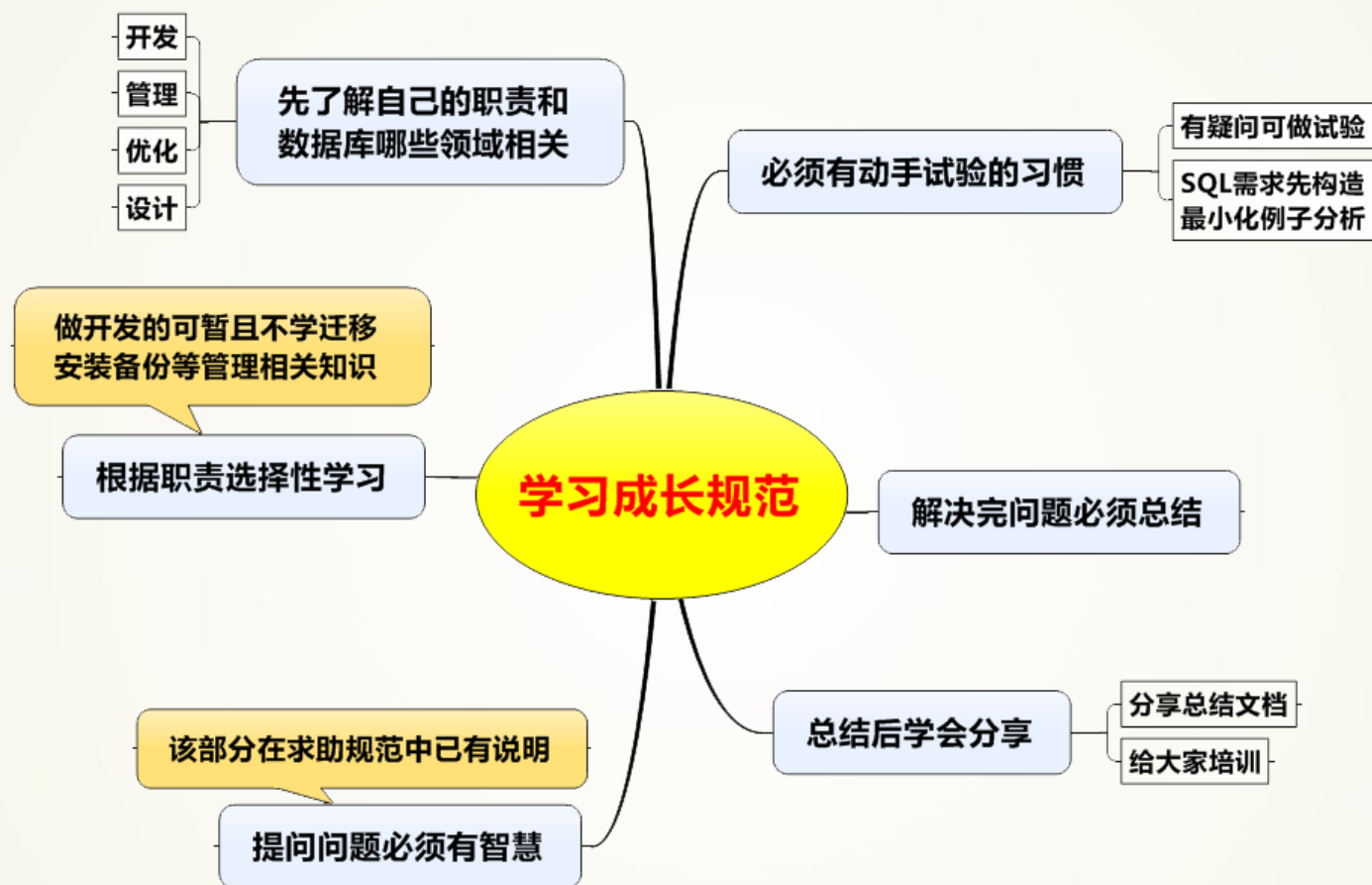


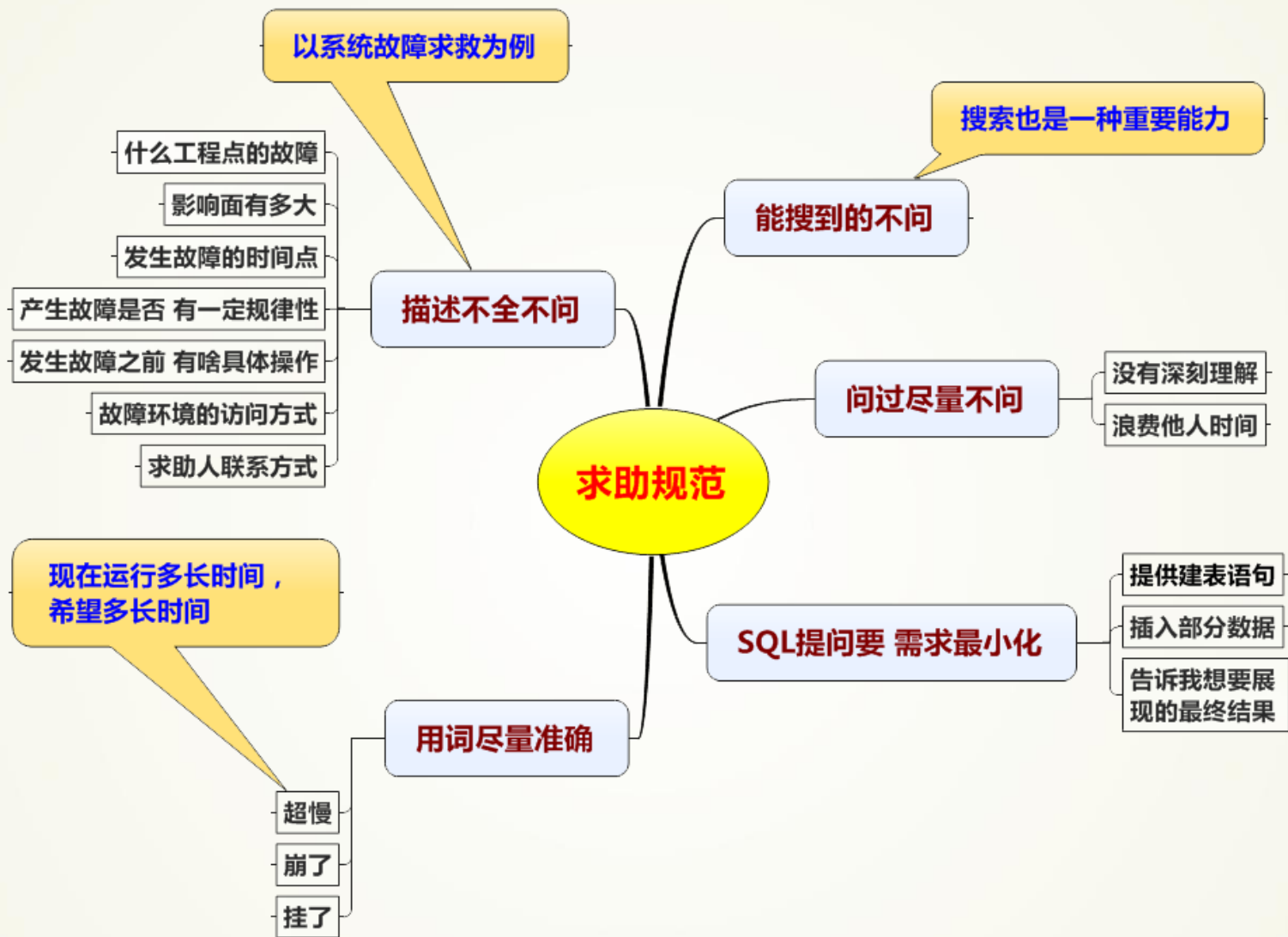
监督规范



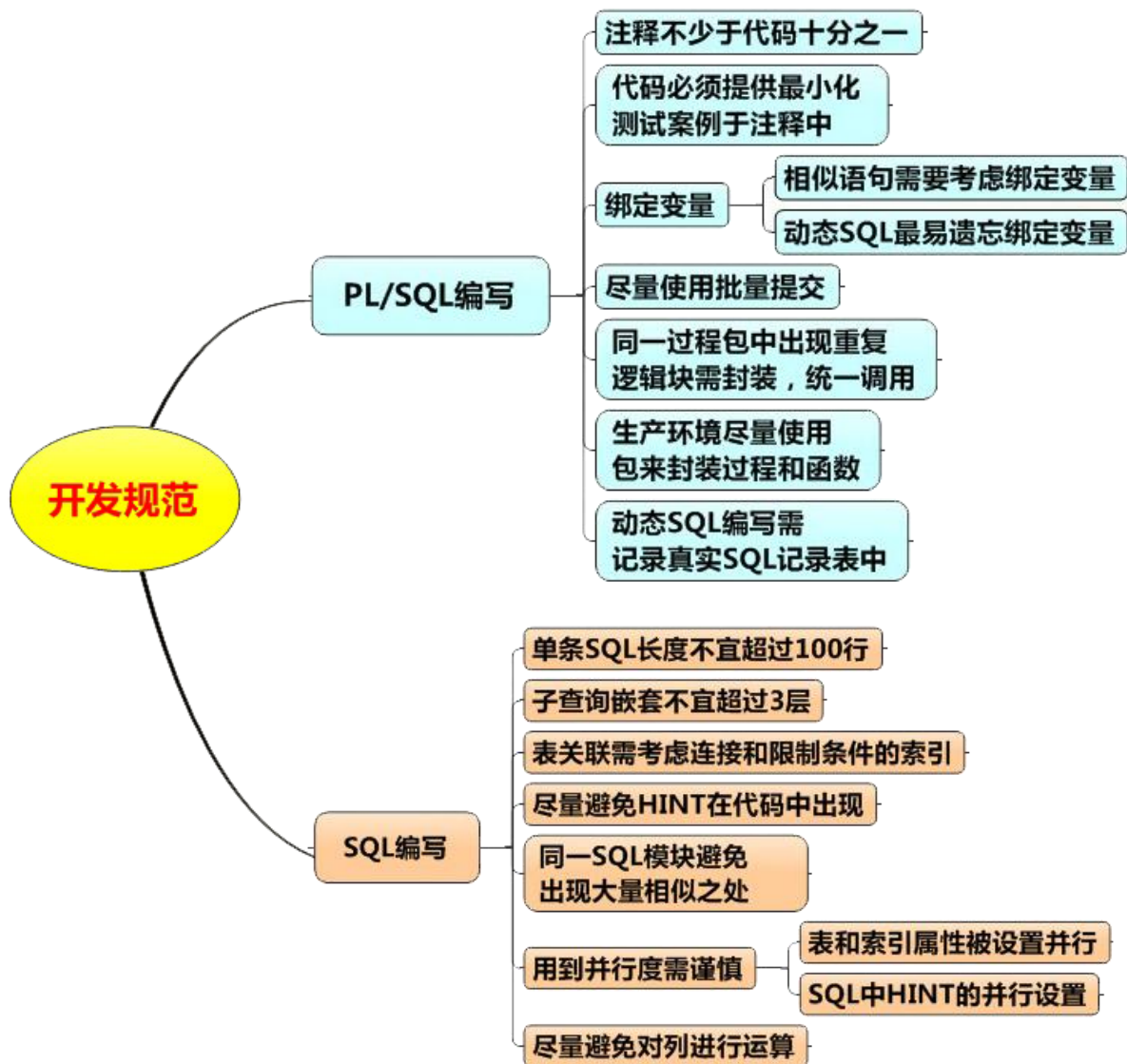
完善规范

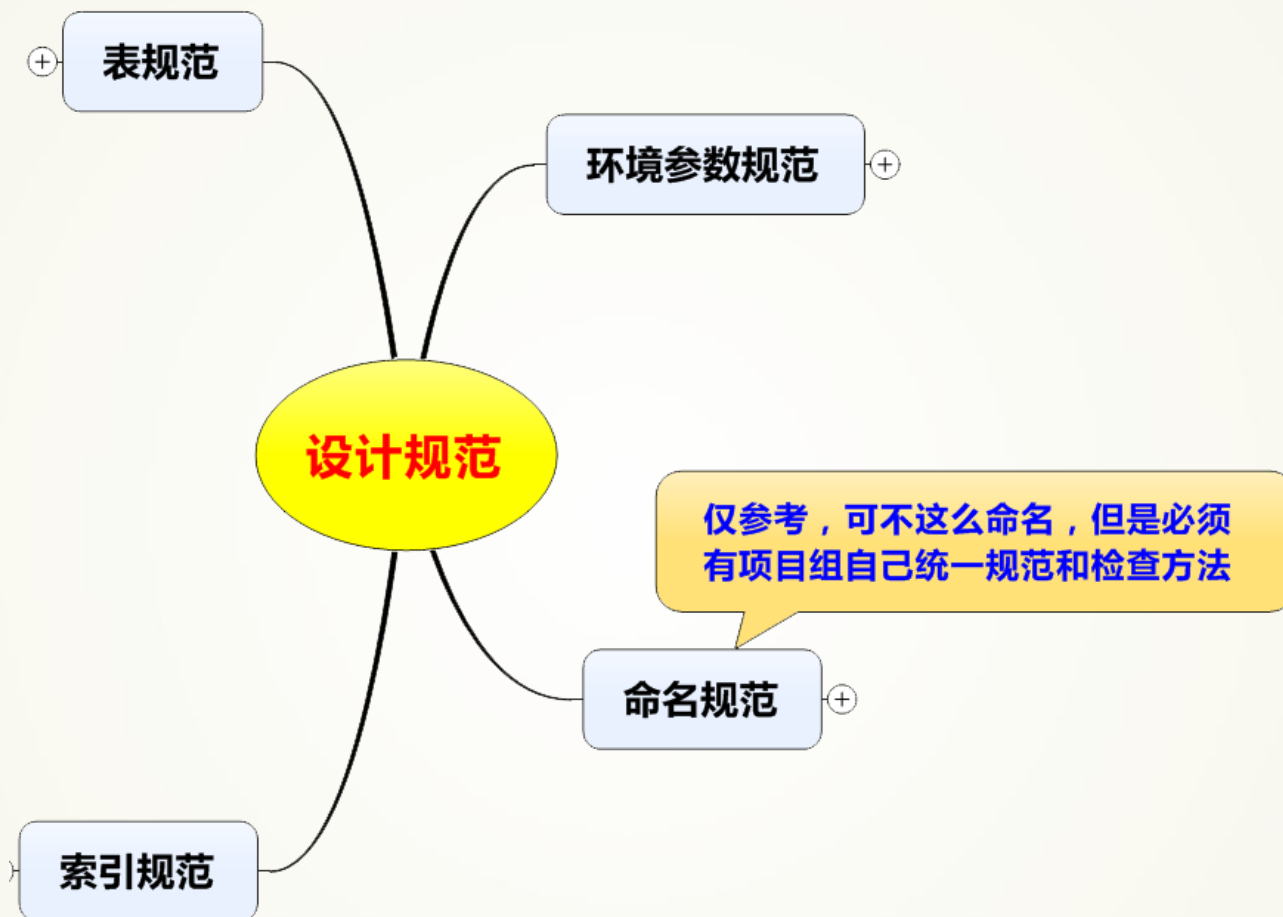


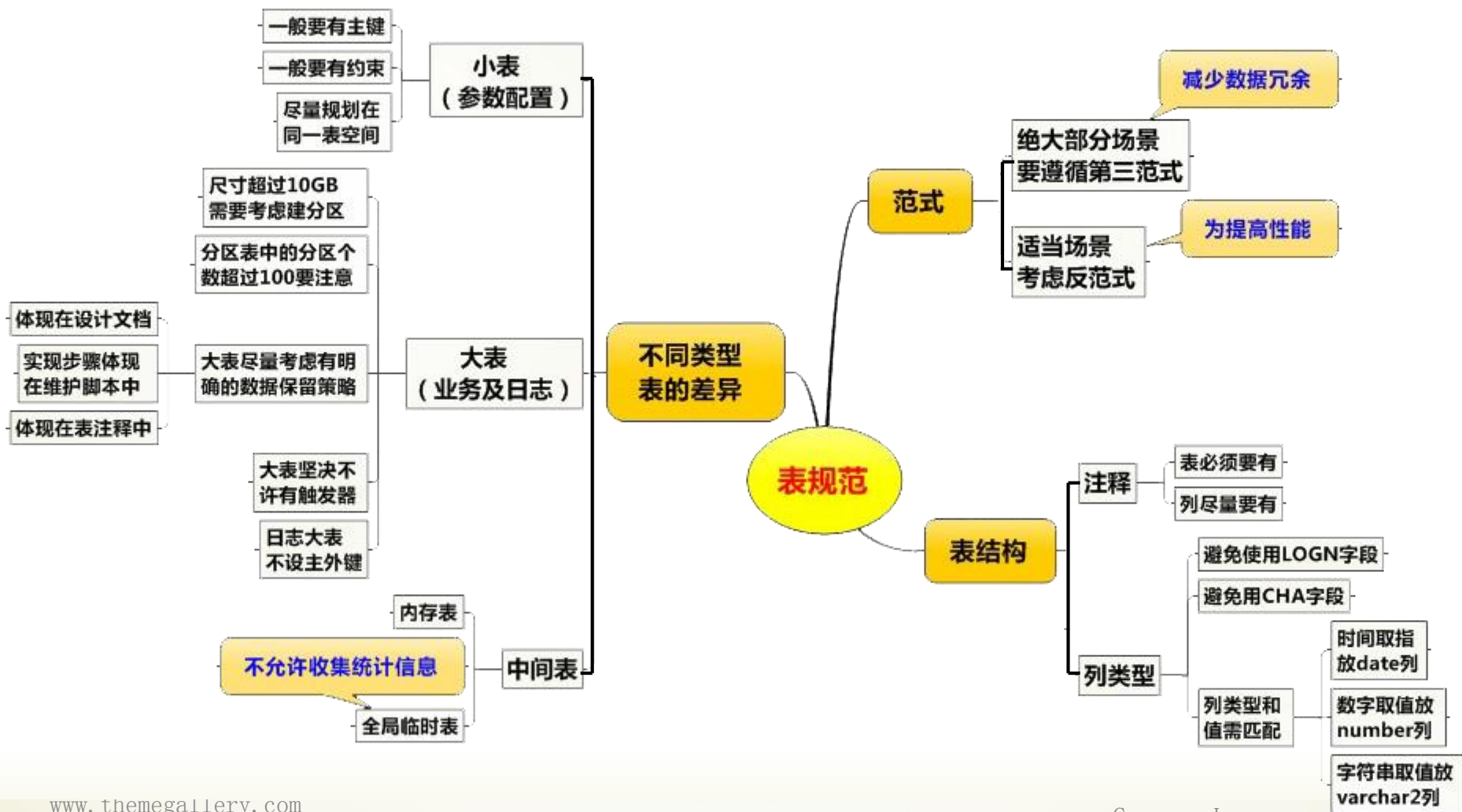






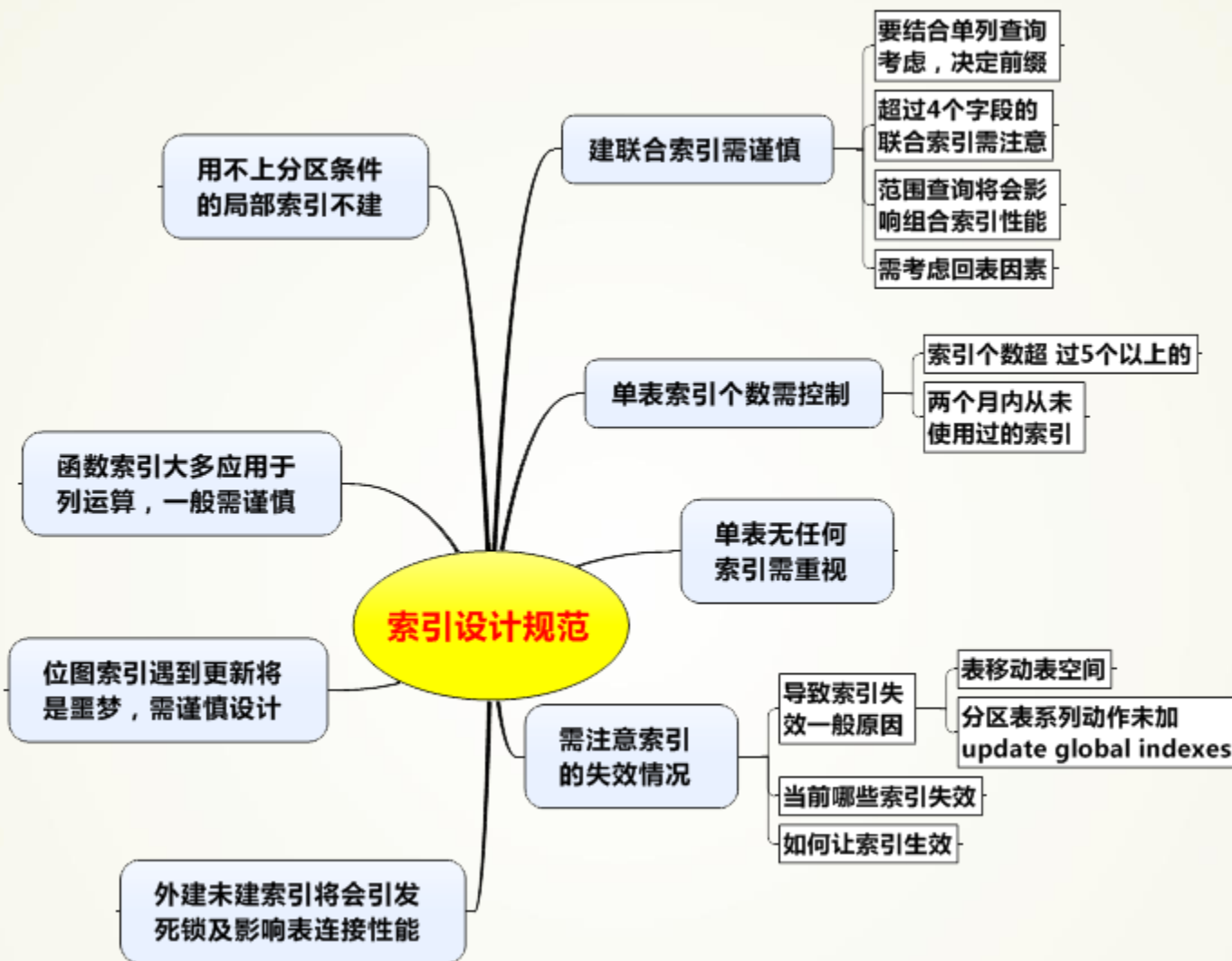


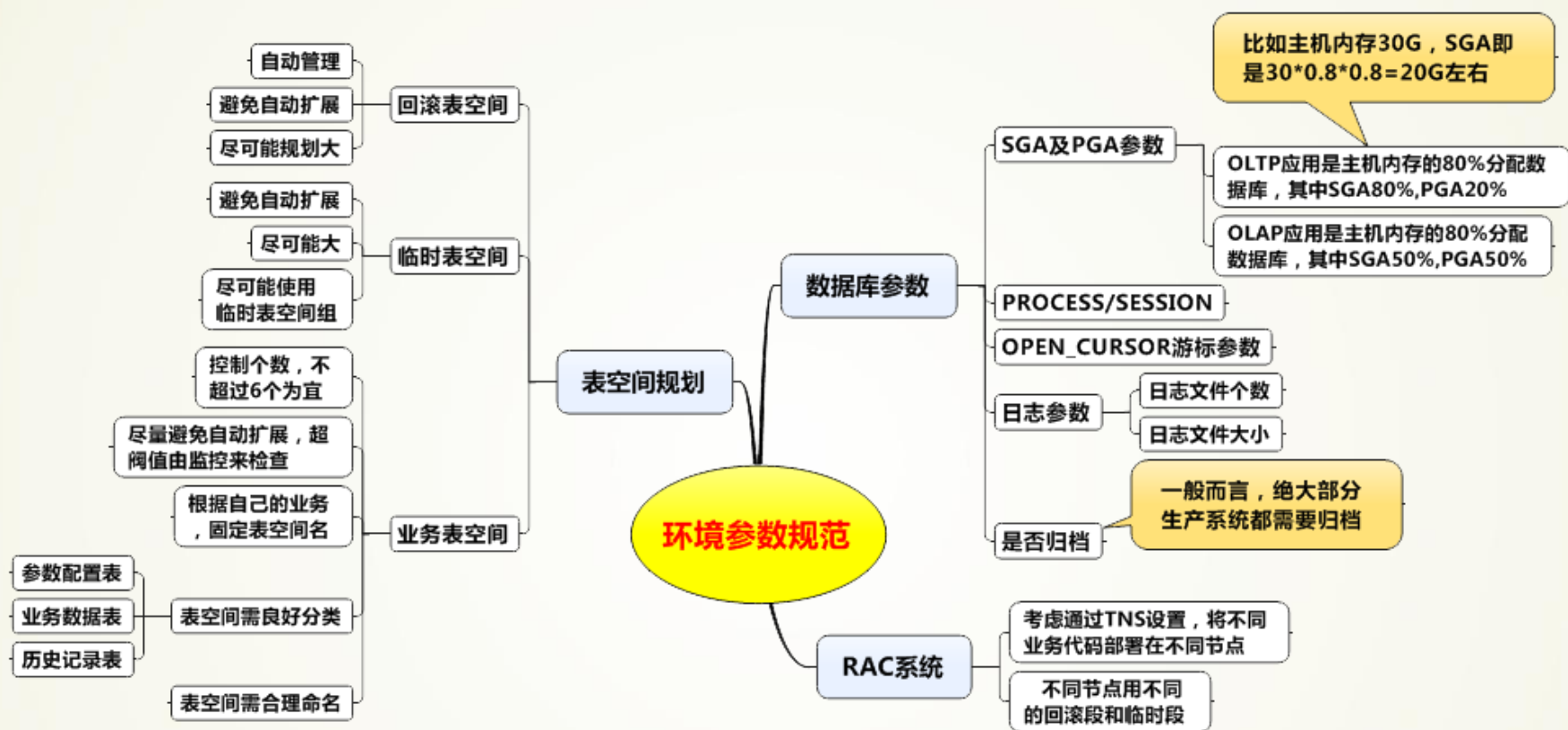


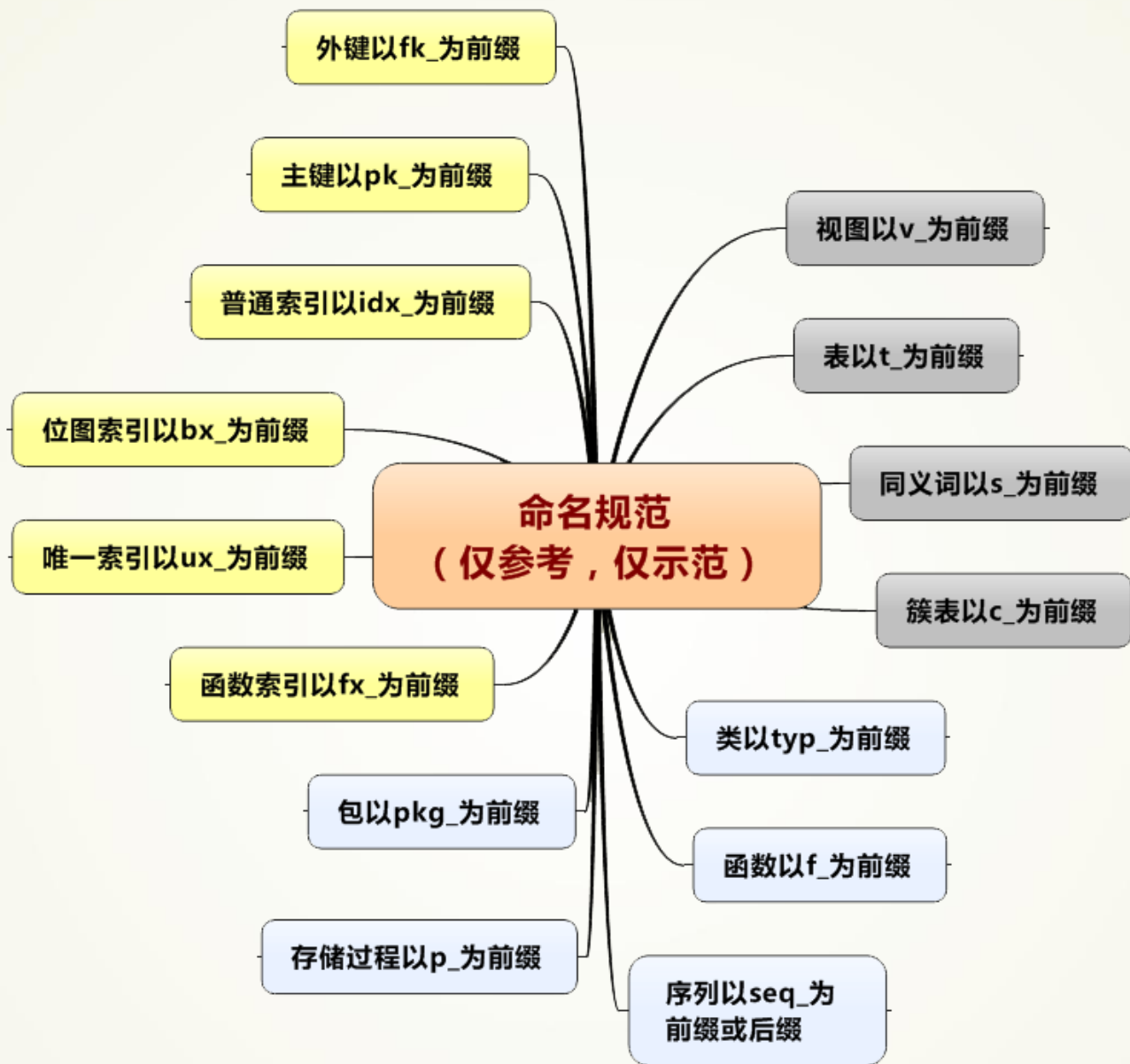


www.themegallery.com

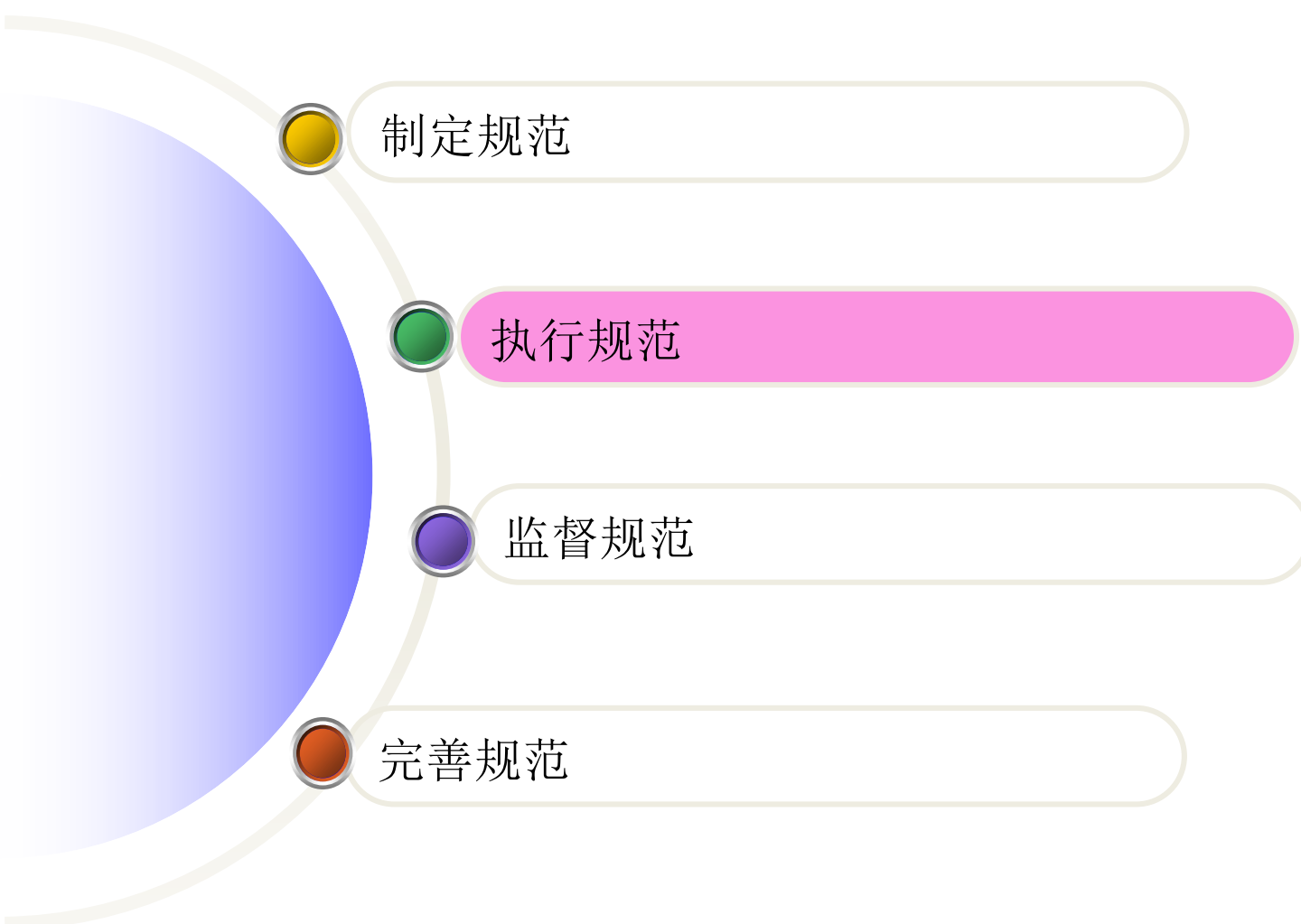
Company Logo







规范的讨论



制定规范

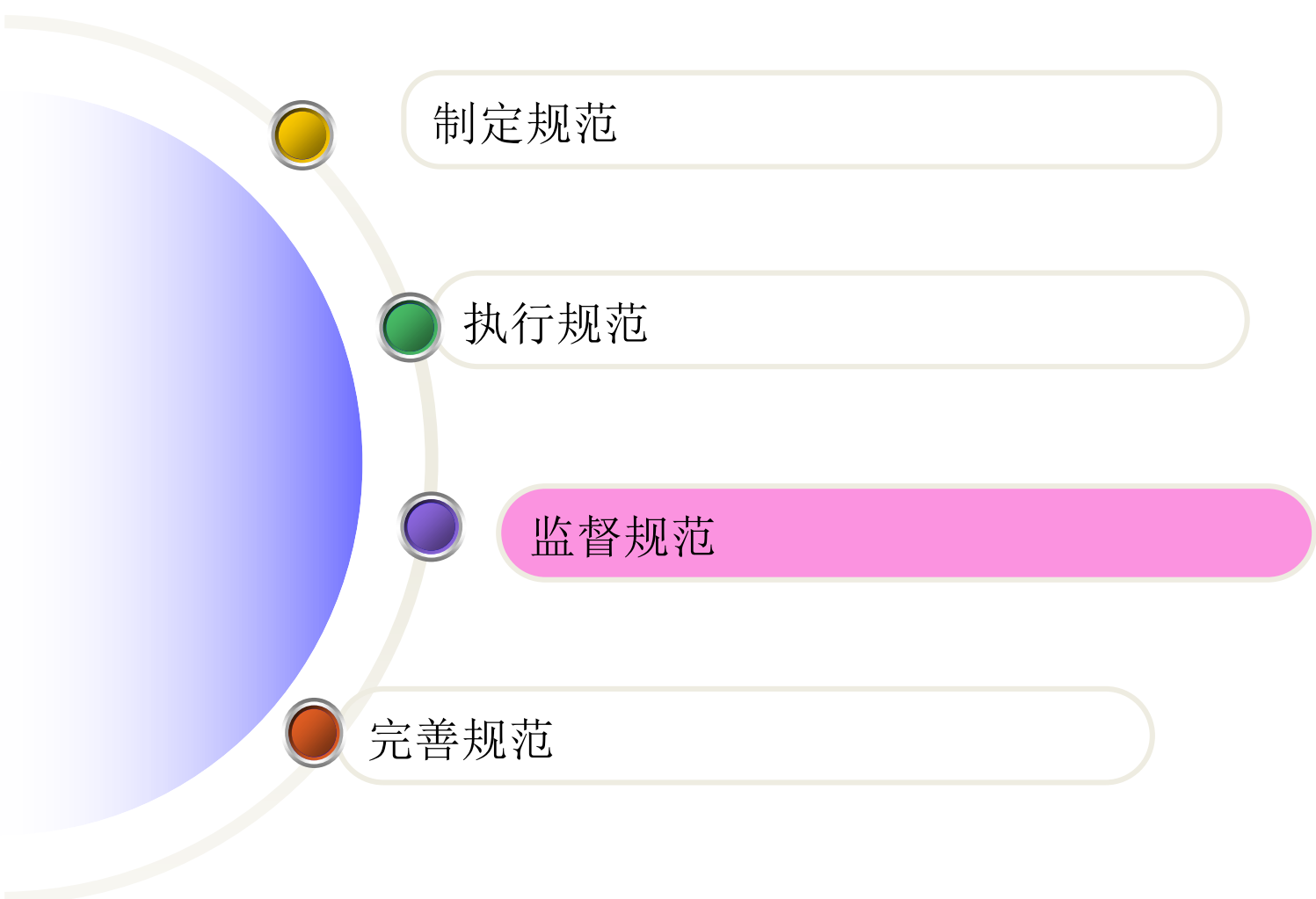
执行规范

监督规范

完善规范

1. 有脚本可以检验
2. 有人给予培训
3. 有制度给予奖惩

规范的讨论



制定规范

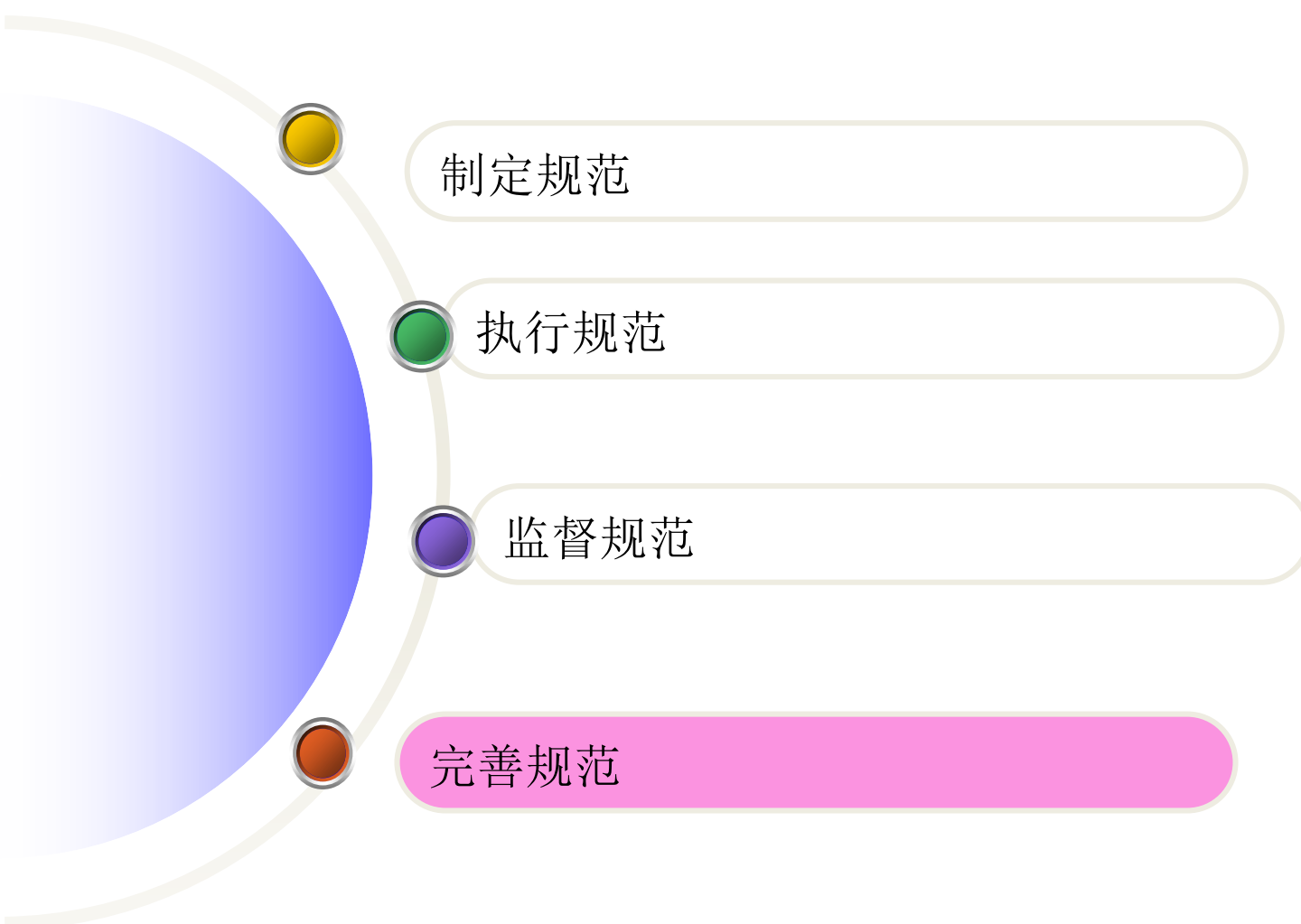
执行规范

监督规范

完善规范

1. 有专人审核规范制度合理性
2. 有专人检查规范脚本准确性
3. 有专人对相关人员的操作结果进行检查

规范的讨论



制定规范

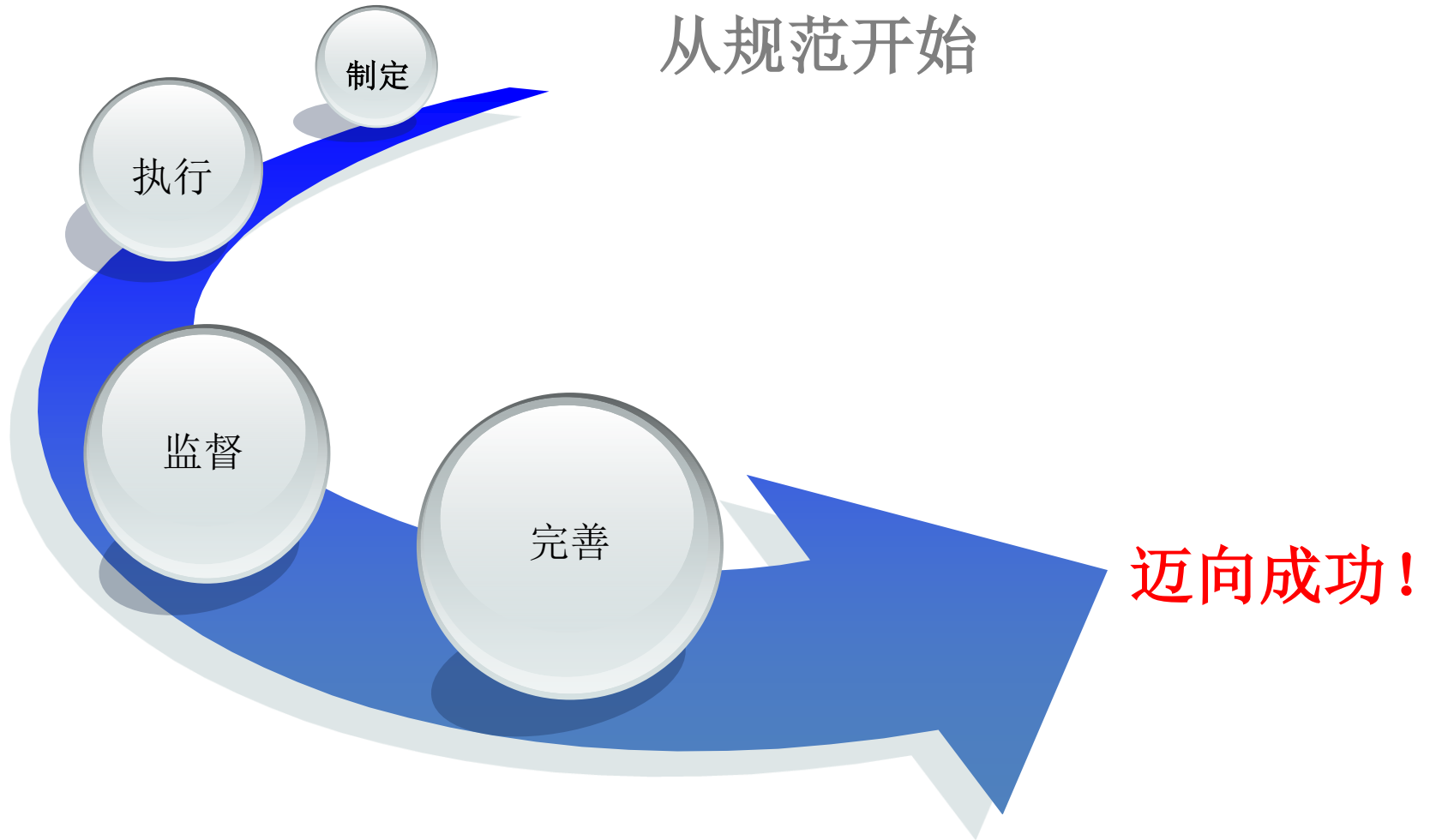
执行规范

监督规范

完善规范

1. 规范程序化，比如要能完成类似自动巡检的功能
2. 规范要具备校验的功能，比如操作完后的校验程序
3. 规范能为自动故障诊断系统做贡献

从规范开始



欢迎莅临

2013中国数据库技术大会

Database
BDaaS
flowingdata
DB2
NoSQL MySQL
Oracle Big Data