高级数据库技术课程

“做中学”选题清单和练功要诀

1) 基于数据特征的数据存储和索引：比如教务系统中的学生数据比较多，数据有层次性，例如学生数据（年级，学院，专业，班级），学生数据的操作有什么特征？如何在磁盘上存储？如何索引？如何自动识别层次关系？能否分析出各种层次关系的优劣？

2) 选课记录（学期，课程，老师，学生）的数据又有什么特征？怎样构建层次关系，就有利于数据访问性能？如何在磁盘上存储？如何索引？如何控制选课？

3) 有的层次更深，比如课程互动记录，层次包括：学期，课程，老师，学生，日期时间，主题，这些数据的操作特点又是什么？应该如何存储？如何索引？如何定期整理？

4) 基于数据库中表的类别和表之间的关系，以及表数据，如何自动识别某个表中的数据蕴含的层次关系。

5) 很多业务操作都涉及多个表之间的联接操作？如何可视化？如何简单化？实现 SQL 的自动合成，降低知识要求的门坎。

6) 数据库访问的审计记录，与数据库的日志，有什么联系和差异？能否结合起来？对于批量操作（update，delete， insert），它们分别可以分成几类？每类有什么特征？是否可以根据特征，进一步细化处理，以提高存储效率？

7) 有时出现错误删除操作，能否给管理员提供一个可视化检查的功能，以及回撤某些操作的功能。

8) 表中数据的更新（update，delete，insert）是如何标识的？如何来实现表中记录更新的标识？增量备份如何实现？

9) 数据库备份的自定义化，可视化操作，备份操作的跟踪管理？例如，表一级的增量备份，增量添加？如何来识别哪些表中的数据有更新，哪些表没有更新？

10) 做一个表一级的数据导入软件，实现重复记录的标识，以及重复记录的省略，或者覆盖。

11) 如何设置定时器，调用定时执行的数据库操作？

12) 审计记录，日志记录，的可视化展示和管理，包括统计。

13) 用户，对象，权限管理的可视化，层次化展示，和权限管理。

14) 数据库服务器中如何支持国际化，例如，报错的文字表达，表名和字段名的文字表达该

如何处理？报错信息的合成。

15) 数据库中表的关联关系的可视化展示。

16) 假定一个网站，提供一种同学聚会，会议之类的信息服务，主办方可自定义要收集的信息，平台自动生成网页，参加方可访问所提供的网页，填上自己的信息，也可修改，删除。给主办方提供邮件群发功能，文档资料的上载功能，自动生成文档资料的下载网页，以及给参加方的文档资料下载网页的邮件群发通知。会议数据的过期清除功能。

17) 自动排课算法的设计。

18) 自主选择的题目。

先了解现有技术，现有实现方法，然后审视现有技术和实现，琢磨能否改进，针对特定的需求，或者有附加条件的基础上做改进也是可行的。想清楚改进解决方案，然后加以实现，再进行测试，获得你所做的改进的实际效果的量化数据。不断地来对 MySQL 扩充完善，做出我们中国人的开源贡献。要求写出**完整、规范**的**技术报告**。做出**有见解，有干货**的**演讲 PPT**。

写技术报告和演讲 PPT，要求把你做的东东作为一个故事来讲，力求生动有趣。最好把它作为一个简单的故事来讲。故事只有简单才通俗易懂，别人听起来才轻松，否则就会云里雾里。一个故事，生动有趣是必备条件。要生动，就要渲染，就要有取舍，有加工。故事要简单，就要突出一个主题，说清在一个什么特定的场景下有个什么样的特定问题，如何如何关键，然后对这个问题你如何来解决（思路，结构轮廓，技术），由总到详，以树形结构和层次方式展开论述。其中，两点必须要注意：1）在引出问题上，要体现时代气息，也就是要从时代进步，说出问题是因新情况/新事物/新关系/新模式而引发出来的，才变得突出起来；2）在解决问题上，也要体现时代气息，交代清楚是因为有了新技术/新设备/新手段才具备了解决问题的可能，才具备了支撑条件，才变得可行。

讲故事，切忌面面俱到。面面俱到就会把听者弄得团团转，晕头转向，最终所以然，使读者感到混杂而难以把握，脑中一片茫然。另外，就是要注意逻辑衔接性，使得句与句之间，在逻辑上贯通，不至给读者“东一榔头，西一棒子”的错乱感觉。上下跳越太大，读者就跟不上，就接受不了，就觉得论文条理不清晰，就会低头打瞌睡。

故事的开场，就要把背景，领域，你对问题的观察角度，问题的选取，选题的意义交代清楚。接下来的是重头部分，讲清楚你是采取什么技术线路，按照什么原则，应用什么技术和工具，做了些什么事情。最后是收尾，交代你做的东东的实际价值到底体现在哪里？实际达到了一个什么样的具体水准？或者经验教训是什么？

另外，既然是研究生，做的工作理应该有探索性。因此，在表述上要从特定的角度去论述，去分析，在某个知识的应用上应该做到完备。论述与分析，自然是从问题入手，首

先交代别人在这个问题上是如何处理的，后续的改进有哪些？现有的方法具有什么特点？在什么样的应用场景下还会存在什么问题？