# OpenStore设计思想

## 系统目标和开发模式的颠覆

引言：当前软件开发的系统，数据存储占据着相当重要的部分。从几十年前一统天下的关系数据库存储数据，到如今大数据、人工智能大背景之下的文档型数据库（以mongodb为首）、KV数据库（以redis为首）、列式存储（hbase，rcfile文件等）、图形存储（neo4j、cayley等）。每一个数据存储方式，都直接决定了软件系统的设计架构、开发工具、开发模式、甚至直接影响了程序猿们的“开发习俗”（注：用“习俗”而不用“习惯”，一字之差，却能真实表达这个事实，习惯代表了个个人的常态，而习俗代表了一个团体，一个社区的约定俗成，虽然这个约定俗成在现在看来，并非最好或最正确）。

而在这些底层存储机制之上，又不断涌现出各种开发框架和平台，例如：关系数据库的各类数据库管理系统：sqlserver系列\mysql等；graphsql图形计算平台等。这些工具的推广和成型，直接导致了开发者，既享受了这些工具和平台带来的关于“数据存储“方面的操作便捷性，统一性。又不得不陷入到这些平台工具的”习俗“之中。

### 传统 vs 当下

关于数据存储，正如引言中谈到。我们可以在这里将“传统“缩小一定的范围，将其只限定于”企业应用“范围。对企业应用来说，”传统“的数据存储机制，毫无疑问，当属《关系型数据库》，没有之一。这除了历史原因决定外，还是由“关系数据库”本身的特性所决定的：

1. 对企业应用来说，大部分的业务数据都习惯性的按照“表格”方式组织，企业中每个部门，每个员工在执行日常事务时，都习惯了这种描述数据，存储数据的方式。
2. 大部分的企业应用中的“表格”式数据之间，业务上都是有“关联”的，并且这种关联，在企业以及企业员工的“思维定式”中，都是自动具有“强制性关联“的。例如：购物车id一定关联着购物车中的商品id
3. 传统技术的“教育“，在设计企业应用时，为了数据所谓的”完整性“、”一致性“等这些数据”约束“，关系数据库将这些概念植入了设计和开发人员的骨髓，数据的schema模式、事务、一对多，一对一等概念已经成为了”数据设计“的”基本常识“
4. 由于DBMS管理系统的普及和强大，以及操作方式SQL语言的统一便捷，使得开发企业应用数据存储部分的开发人员，不必费时费力去思考特定环境下“数据”的各种存储、查询等特性。直接一条简单的sql语句，即能完成相对比较复杂的企业应用中数据的新增、搜索、排序等操作

而当下，随着大数据和人工智能的流行，企业应用中的各种功能、业务、逻辑也面临着各种挑战和改变。同时，导致了企业应用中的《数据存储》部分，也有了翻天覆地的变化，主要表现在如下几方面：

1. 企业组织机构的复杂以及业务的延伸，导致企业每天产生的数据量呈几何级数增长，这导致了之前关系数据库中很难得出现的分库、分表、集群等“小场景“应用功能提上了日程。而关系库中的分库、分表、集群等操作，又是及其复杂的