Spark SQL 的前身是 Shark，Shark 是伯克利实验室 Spark 生态环境的组件之一，它能运行在 Spark 引擎上，从而使得 SQL 查询的速度得到 10-100 倍的提升，但是，随着 Spark 的发展，由于 Shark 对于 Hive 的太多依赖（如采用 Hive 的语法解析器、查询优化器等等），制约了 Spark 的 One Stack Rule Them All 的既定方针，制约了 Spark 各个组件的相互集成，所以提出了 SparkSQL 项目。SparkSQL 抛弃了原有 Shark 的代码，汲取了 Shark 的一些优点，如内存列存储（In-MemoryColumnarStorage）、Hive 兼容性等，重新开发了 SparkSQL 代码；由于摆脱了对 Hive 的依赖性，SparkSQL 无论在数据兼容、性能优化、组件扩展方面都得到了极大的方便。SQLContext 具体的执行过程如下：

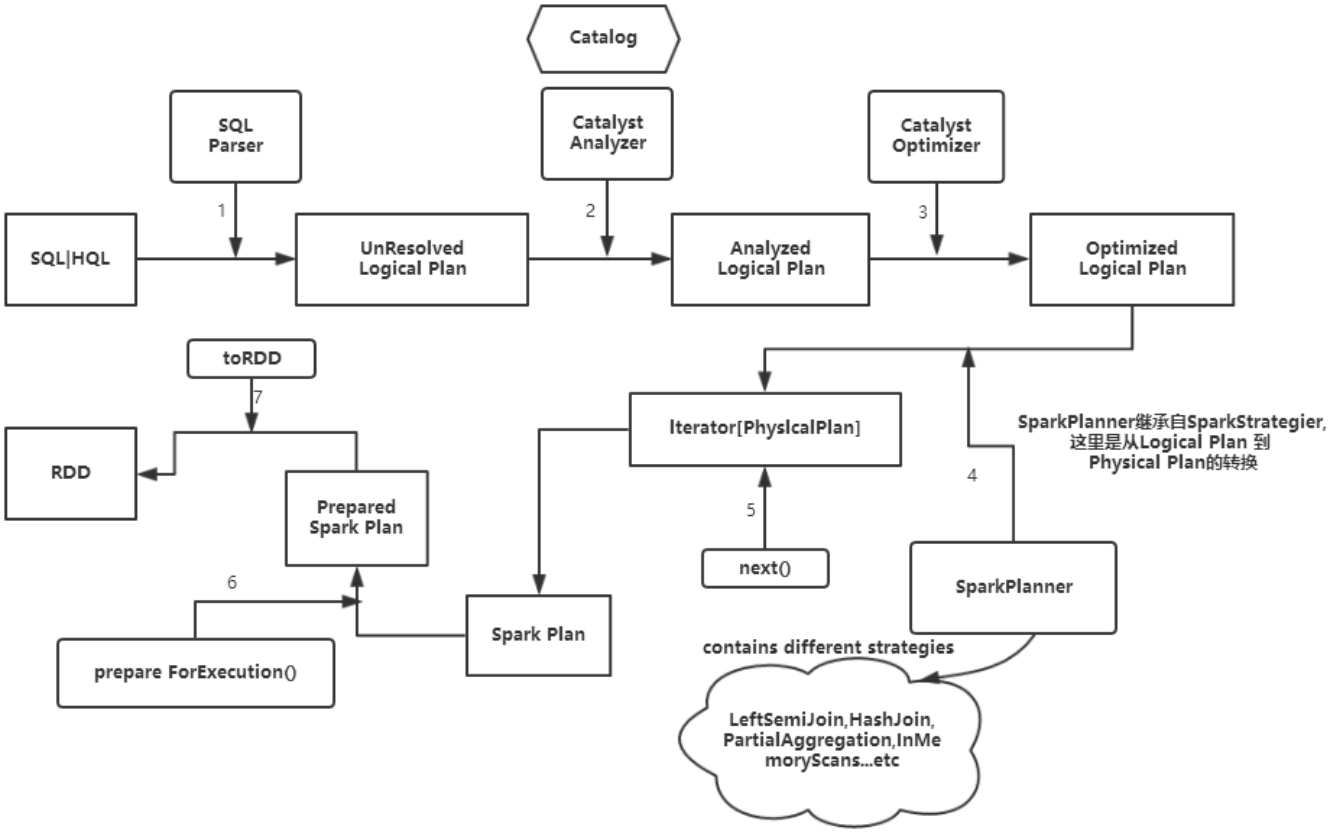

图 1: Spark SQL 执行流程

1) SQL | HQL 语句经过 SqlParse 解析成 UnresolvedLogicalPlan。

2) 使用 analyzer 结合数据字典（catalog）进行绑定，生成 resolvedLogicalPlan，在这个过程中，Catalog 提取出 SchemRDD，并注册类似 case class 的对象，然后把表注册进内存中。

3) Analyzed Logical Plan 经过 Catalyst Optimizer 优化器优化处理后，生成 Optimized Logical Plan，该过程完成以后，以下的部分在 Spark core 中完成。

4) Optimized Logical Plan 的结果交给 SparkPlanner，然后 SparkPlanner 处理后交给 PhysicalPlan，经过该过程后生成 Spark Plan。

5) 使用 SparkPlan 将 LogicalPlan 转换成 PhysicalPlan。

6) 使用 prepareForExecution() 将 PhysicalPlan 转换成可执行物理计划。

7) 使用 execute() 执行可执行物理计划。

8) 生成 DataFrame。

在整个运行过程中涉及到多个 SparkSQL 的组件，如 SqlParse、analyzer、optimizer、SparkPlan 等等。