MySQL的架构

应用层：负责和客户端，响应客户端请求，建立连接，返回数据。

逻辑层：包括SQK接口，解析器，优化器，Cache与buffer。

数据库引擎层：MyISAM,InnoDB…。

物理层：负责文件存储，日志等。

InnoDB引擎

InnoDB 是 MySQL 的默认事务型引擎，支持事务，表是基于聚簇索引建立的。支持表级锁和行级锁，支持外键，适合数据增删改查都频繁的情况。InnoDB 采用 MVCC 来支持高并发，并且实现了四个标准的隔离级别。其默认级别是 REPEATABLE READ，并通过间隙锁策略防止幻读，间隙锁使 InnoDB 不仅仅锁定查询涉及的行，还会对索引中的间隙进行锁定防止幻行的插入。

(InnoDB 可以通过show create table table\_name 来查看,也可以用来查看外键名)

MyISAM存储引擎

MySQL5.1及之前，MyISAM 是默认存储引擎。MyISAM不支持事务，Myisam支持表级锁，不支持行级锁，表不支持外键，该存储引擎存有表的行数，count运算会更快。适合查询频繁，不适合对于增删改要求高的情况

Memory 存储引擎

Memory存储引擎将所有数据都保存在内存，不需要磁盘 IO。支持哈希索引，因此查找速度极快。Memory 表使用表级锁，因此并发写入的性能较低

事务（ACID）满足如下几个特性：

原子性（Atomicity）: 一个事务中的所有操作要么全部完成，要么全部不完成。

一致性（Consistency）: 事务执行前后数据库的状态保存一致。如分布式环境数据状态，操作一致性。

隔离性（Isolation） 多个并发事务对数据库进行操作，事务间互不干扰。

持久性（Durability） 事务执行完毕，对数据的修改是永久的，即使系统故障也不会丢失

事务：

对于一组相关操作通常需要其全部成功或全部失败，在关系数据库中，这组相关操作成为事务。

事务是数据库系统中执行过程中的最小逻辑单元，保证数据的操作都能全部完成，从而保证数据的完整性。

\*如果取消auto commit， select选取快照。

脏读：脏读是指在一个事务处理过程里读取了另一个未提交的事务中的数据

幻读：在update, insert，delete时候，事务1无法更改中途事务2修改的数据

不可重复读：如果一个事务连续读取一个数值，但是另外一个事务对值进行了修改。

幻读和不可重复读都是读取了另一条已经提交的事务（这里和脏读不同，脏读是未提交的数据），不可重复读查询的是同一个数据项，幻读针对的是一批数据体。

共享锁（读锁）：select 命令将在所引用的表上加上此锁。通常，任何只读取表而不修改表的查询都会请求这种锁模式。

排它锁（写锁）：如果获得某种数据项的排他锁，可以执行对该数据项的写操作。

加入读锁以后，就不可以插入操作了。

死锁：在两个或者多个任务中，

加密：加密算法利用加密密钥将明文转换为密文

（对称加密：

非对称加密：

）

解密：解密算法利用解密密钥将密文还原为明文

（1）数据定义。（SQL DDL）用于定义SQL模式、基本表、视图和索引的创建和撤消操作。  
（2）数据操纵。（SQL DML）数据操纵分成数据查询和数据更新两类。数据更新又分成插入、删除、和修改三种操作。  
（3）数据控制。（DCL）包括对基本表和视图的授权，完整性规则的描述，事务控制等内容。

（4）嵌入式SQL的使用规定。（TCL）涉及到SQL语句嵌入在宿主语言程序(Java, Python)中使用的规则

SQL 重要用法：

聚合函数作为窗口函数

作用：聚合函数作为窗口函数，是起到"累加/累计"的效果，比如，就是截止到本行，最大值？最小值是多少

#查询成绩的累加

SELECT\*, SUM(成绩) over (ORDER BY 成绩 DESC) AS '求和'

FROM class;

TopN 问题：

select\*

from（select\*，row\_number() over (partition by 部门 order by salary desc) as ranking

from 表名) as a

where ranking<=N;

\*case 的用法

case column\_name

when value then xxx

case

when 满足条件then xxx

1、简单case函数判断条件只能是等于,而搜索case函数的条件可以是子查询,In,大于、等于等等。

2、如果只是使用简单的条件分组,可以选择普通case函数,如果需要判断更多的场景,则选择搜索case更好。

\*Case可用于行转列， max (case when xx then xx), max(case when xx then xx)

\*Case 课用于求行数 sum(case when xx then 1 else 0 end)

索引（index）：

对数据库表中一列或多列的值进行排序的一种结构，是一个单独的，存储在磁盘上的数据库结构，包含了对数据表里所有记录的引用指针。

Hash索引：

Hash索引用于处理简单的等于（’=’）比较。当一个索引的列只涉及到使用’=’操作进行比较时，考虑使用(create index <index\_name> on <table\_name> using hash(column))

聚集索引：指数据库表行中数据的物理顺序与键值的逻辑（索引）顺序相同。

在数据库中由主键作为聚集索引。

非聚集索引：该索引中索引的逻辑顺序与磁盘上 行 的物理存储顺序不同。

非聚集索引不指向数据，而是指向聚集索引。

复合索引: Mysql从左到右的使用索引中的字段，一个查询可以只使用索引中的一部份，但只能是最左侧部分。 例如索引是key index (a,b,c). 可以支持a | a,b| a,b,c 3种组合进行查找，但不支持 b,c进行查找 .当最左侧字段是常量引用时，索引就十分有效

优化：

1）提高磁盘I/O的读写速度

2) 提高高负荷情况下的负载能力

3) 优化表结构、索引、查询语句

存储过程 (Stored Procedure):

是一种在数据库中存储复杂程序，以便外部程序调用的一种数据库对象。