Explain+sql执行后，产生的表种每个字段的含义。

Id：决定表的读取顺序。有三种情况：

Id相同：执行顺序由上至下

Id不同，如果是子查询，id序号会递增，id值越大优先级就越高，越先被执行。所以子查询会比主查询先执行。

Id相同不同同时存在，如果id相同，可以认为是一组，从上往下顺序执行，在每一个组种，id值越大优先级越高，越先执行。

Select\_type：表示查询的类型，也就是数据读取的操作类型，可以分为5种情况：

Simple：简单的select查询，查询中不包含子查询或者union

Primary：查询中若包含任何复杂的子查询，最外层则会被标记位primary

Subquery：在select或者where列表中包含了子查询，则这些子查询回备标记为subquery

Derived：在from列表中包含的子查询被标记为DERIVED(衍生表),mysql会递归执行这些子查询,把结果放临时表中

Union：若第二个select出现union之后，则被标记为union，若union包含在from子句的子查询中，外层select将被标记为derived。

Union result：从union表中获取select的结果

Type**：**访问类型排序

显示了使用何种类型，从最好到最差依次是：system>const>eq\_ref>ref>range>index>all

System:表**只有一行**记录（相同于是系统表），这是const类型的特例，平时不会出现，所以可以忽略不计。

Const：表示通过索引一次就找到了，const用于比较primary key或者unique索引。因为只匹配一行记录，所以很快。如果将主键置于where列表中（where中的条件是主键），mysql就能将该查询出转化成一种常量。

Eq\_ref:唯一性索引扫描，对于每一个索引键，表中只有一条记录与之匹配，常用于主键或者唯一索引扫描。

Ref:非唯一性索引扫描，返回匹配某个单独值的所有行，本质上也是一种索引访问，它返回所有匹配某个单独值的行，然而，它可能会找到多个符合条件的行，所以它应该属于查找和扫描的混合体。

Range：只检索给定范围的行，使用一个索引来选择行，key列显示使用哪个索引，一般就是在你的where语句中出现了between，<，>，in等的查询，这种范围索引扫描比全表扫描要好，因为它只需要开始于索引的某一个点，结束于另一个点，不用扫描全部索引。

Index：index于all区别就是index类型只遍历索引树，这通常比all快，因为索引文件通常比数据文件小，也就是说虽然all和index都是读写表，但index是从索引中读取的，而all是从硬盘中读取的；

All：也就是全表扫描

备注：一般来说，得保证查询至少达到range级别，最好能达到ref

Possible\_keys：显示可能会被应用到这张表的索引，一个或者多个；查询涉及到的字段上若存在索引，则该索引将被列出，但不一定被查询实际用到。

Key：实际使用到的索引，如果为null，则没用使用索引；查询中若使用了覆盖索引，则该索引仅出现在key列表中。

Key\_len:表示索引使用的字节数，可通过该列计算查询中使用的索引的长度，在不损失精确性的情况下，长度越短越好；key\_len显示的值为索引字段的最大可能长度，并非实际使用长度，即key\_len是根据表定义计算而得到的，不是通过表内检索得出的。

Ref:显示索引的哪一列被使用了，如果可能的话，是一个常数，哪些列或常量被用于查找索引列的值上。

Rows：根据表统计信息及索引选用情况，大致估算出找到所需的记录所需要读取的行数。

Extra：包含不适合在其他列中显示，但十分重要的额外信息。

Using filesort（出现这个不好），说明mysql会对数据使用一个外部的索引排序，而不是按照表内的索引顺序进行读取的，mysql无法利用索引完成的排序操作称为“文件排序”

Using temporary：（出现这个东西更不好）：使用到了临时表保存中间结果，mysql在对查询结果排序时使用了临时表，常用于排序order by和分组查询group by。

Using index：表示响应的select操作中使用了覆盖索引 （covering index），避免了访问了表的数据行，效率不错。（select数据列从索引中就能取得）

Using where：使用了where

Using join buffer：使用了连接缓存：block nested loop，连接算法是块嵌套循环连接，Batched key access，连接算法是批量索引连接。

Impossible where：where子句的值总是false，不能用来获取任何元组

Select tables optimized away：在没有group by子句的情况下，基于索引优化min/max操作，或者对于myisam存储引擎优化count（\*）操作，不必等到执行阶段在进行计算，查询执行计划生成的阶段即完成优化。

Distinct：优化distinct操作，在找到第一匹配的元组后即停止找相同值的动作。