MySQL 面试题还包括如下知识点或题型：

MySQL 高性能索引
SQL 语句
MySQL 查询优化
MySQL 高扩展高可用
MySQL 安全性

问题 1：char、varchar 的区别是什么？
varchar 是变长而 char 的长度是固定的。如果你的内容是固定大小的，你会得到更好的性能。

问题 2: TRUNCATE 和 DELETE 的区别是什么？
DELETE 命令从一个表中删除某一行，或多行，TRUNCATE 命令永久地从表中删除每一行。

问题 3：什么是触发器，MySQL 中都有哪些触发器？
触发器是指一段代码，当触发某个事件时，自动执行这些代码。在 MySQL 数据库中有如下六种触发器：

1、Before Insert
2、After Insert
3、Before Update
4、After Update
5、Before Delete
6、After Delete

问题 4：FLOAT 和 DOUBLE 的区别是什么？

FLOAT 类型数据可以存储至多 8 位十进制数，并在内存中占 4 字节。
DOUBLE 类型数据可以存储至多 18 位十进制数，并在内存中占 8 字节。

问题 5：如何在 MySQL 种获取当前日期？

```
SELECT CURRENT_DATE();
```

问题 6：如何查询第 n 高的工资？

```
SELECT DISTINCT(salary) from employee ORDER BY salary DESC LIMIT n-1,1
```

问题 7：请写出下面 MySQL 数据类型表达的意义（int(0)、char(16)、varchar(16)、datetime、text）

知识点分析

此题考察的是 MySQL 数据类型。MySQL 数据类型属于 MySQL 数据库基础，由此延伸出的知识点还包括如下内容：

MySQL 基础操作
MySQL 存储引擎
MySQL 锁机制
MySQL 事务处理、存储过程、触发器

数据类型考点：

1、整数类型，包括 TINYINT、SMALLINT、MEDIUMINT、INT、BIGINT，分别表示 1 字节、2 字节、3 字节、4 字节、8 字节整数。任何整数类型都可以加上 UNSIGNED 属性，表示数据是无符号的，即非负整数。
长度：整数类型可以被指定长度，例如：INT(11)表示长度为 11 的 INT 类型。长度在大多数场景是没有意义的，它不会限制值的合法范围，只会影响显示字符的个数，而且需要和 UNSIGNED ZEROFILL 属性配合使用才有意义。
例子，假定类型设定为 INT(5)，属性为 UNSIGNED ZEROFILL，如果用户插入的数据为 12 的话，那么数据库实际存储数据为 00012。

2、实数类型，包括 FLOAT、DOUBLE、DECIMAL。
DECIMAL 可以用于存储比 BIGINT 还大的整型，能存储精确的小数。
而 FLOAT 和 DOUBLE 是有取值范围的，并支持使用标准的浮点进行近似计算。
计算时 FLOAT 和 DOUBLE 相比 DECIMAL 效率更高一些，DECIMAL 你可以理解成是用字符串进行处理。

3、字符串类型，包括 VARCHAR、CHAR、TEXT、BLOB
VARCHAR 用于存储可变长字符串，它比定长类型更节省空间。
VARCHAR 使用额外 1 或 2 个字节存储字符串长度。列长度小于 255 字节时，使用 1 字节表示，否则使用 2 字节表示。
VARCHAR 存储的内容超出设置的长度时，内容会被截断。
CHAR 是定长的，根据定义的字符串长度分配足够的空间。
CHAR 会根据需要使用空格进行填充方便比较。
CHAR 适合存储很短的字符串，或者所有值都接近同一个长度。
CHAR 存储的内容超出设置的长度时，内容同样会被截断。

使用策略：

对于经常变更的数据来说，CHAR 比 VARCHAR 更好，因为 CHAR 不容易产生碎片。

对于非常短的列，CHAR 比 VARCHAR 在存储空间上更有效率。
使用时要注意只分配需要的空间，更长的列排序时会消耗更多内存。
尽量避免使用 TEXT/BLOB 类型，查询时会使用临时表，导致严重的性能开销。

4、枚举类型（ENUM），把不重复的数据存储为一个预定义的集合。
有时可以使用 ENUM 代替常用的字符串类型。
ENUM 存储非常紧凑，会把列表值压缩到一个或两个字节。
ENUM 在内部存储时，其实存的是整数。
尽量避免使用数字作为 ENUM 枚举的常量，因为容易混乱。
排序是按照内部存储的整数

5、日期和时间类型，尽量使用 timestamp，空间效率高于 datetime，
用整数保存时间戳通常不方便处理。
如果需要存储微妙，可以使用 bigint 存储。
看到这里，这道真题是不是就比较容易回答了。

答：int(0)表示数据是 INT 类型，长度是 0、char(16)表示固定长度字符串，长度为 16、varchar(16)表示可变长度字符串，长度为 16、datetime 表示时间类型、text 表示字符串类型，能存储大字符串，最多存储 65535 字节数据）

MySQL 基础操作：

常见操作

MySQL 的连接和关闭：mysql -u -p -h -P

-u：指定用户名
-p：指定密码
-h：主机
-P：端口

进入 MySQL 命令行后：G、c、q、s、h、d

G：打印结果垂直显示
c：取消当前 MySQL 命令
q：退出 MySQL 连接
s：显示服务器状态
h：帮助信息
d：改变执行符

MySQL 存储引擎：

1、InnoDB 存储引擎，

默认事务型引擎，最重要最广泛的存储引擎，性能非常优秀。
数据存储在共享表空间，可以通过配置分开。也就是多个表和索引都存储在一个表空间中，可以通过配置文件改变此配置。
对主键查询的性能高于其他类型的存储引擎。
内部做了很多优化，从磁盘读取数据时会自动构建 hash 索引，插入数据时自动构建插入缓冲区。
通过一些机制和工具支持真正的热备份。
支持崩溃后的安全恢复。
支持行级锁。
支持外键。

2、MyISAM 存储引擎，

拥有全文索引、压缩、空间函数。
不支持事务和行级锁、不支持崩溃后的安全恢复。
表存储在两个文件，MYD 和 MYI。
设计简单，某些场景下性能很好，例如获取整个表有多少条数据，性能很高。
全文索引不是很常用，不如使用外部的 ElasticSearch 或 Lucene。

3、其他表引擎，
Archive、Blackhole、CSV、Memory

使用策略
在大多数场景下建议使用 InnoDB 存储引擎。

MySQL 锁机制

表锁是日常开发中的常见问题，因此也是面试当中最常见的考察点，当多个查询同一时刻进行数据修改时，就会产生并发控制的问题。共享锁和排他锁，就是读锁和写锁。

共享锁，不堵塞，多个用户可以同时读一个资源，互不干扰。

排他锁，一个写锁会阻塞其他的读锁和写锁，这样可以只允许一个用户进行写入，防止其他用户读取正在写入的资源。

锁的粒度

表锁，系统开销最小，会锁定整张表，MyIsam 使用表锁。
行锁，最大程度的支持并发处理，但是也带来了最大的锁开销，InnoDB 使用行锁。

MySQL 事务处理

MySQL 提供事务处理的表引擎，也就是 InnoDB。
服务器层不管理事务，由下层的引擎实现，所以同一个事务中，使用多种引擎是不靠谱的。
需要注意，在非事务表上执行事务操作，MySQL 不会发出提醒，也不会报错。

存储过程

为以后的使用保存的一条或多条 MySQL 语句的集合，因此也可以在存储过程中加入业务逻辑和流程。
可以在存储过程中创建表，更新数据，删除数据等等。

使用策略

可以通过把 SQL 语句封装在容易使用的单元中，简化复杂的操作
可以保证数据的一致性
可以简化对变动的管理

触发器

提供给程序员和数据分析员来保证数据完整性的一种方法，它是与表事件相关的特殊的存储过程。
使用场景

可以通过数据库中的相关表实现级联更改。
实时监控某张表中的某个字段的更改而需要做出相应的处理。
例如可以生成某些业务的编号。
注意不要滥用，否则会造成数据库及应用程序的维护困难。
大家需要牢记以上基础知识点，重点是理解数据类型 CHAR 和 VARCHAR 的差异，表存储引擎 InnoDB 和 MyISAM 的区别。

问题 8：请说明 InnoDB 和 MyISAM 的区别

InnoDB 支持事务，MyISAM 不支持；
InnoDB 数据存储在共享表空间，MyISAM 数据存储在文件中；

InnoDB 支持行级锁，MyISAM 只支持表锁；
InnoDB 支持崩溃后的恢复，MyISAM 不支持；
InnoDB 支持外键，MyISAM 不支持；
InnoDB 不支持全文索引，MyISAM 支持全文索引；

问题 9：innodb 引擎的特性

插入缓冲（insert buffer)
二次写(double write)
自适应哈希索引(ahi)
预读(read ahead)

问题 10：请列举 3 个以上表引擎
InnoDB、MyISAM、Memory

问题 11：请说明 varchar 和 text 的区别

varchar 可指定字符数，text 不能指定，内部存储 varchar 是存入的实际字符数+1 个字节（n<=255）或 2 个字节(n>255)，text 是实际字符数+2 个字节。
text 类型不能有默认值。
varchar 可直接创建索引，text 创建索引要指定前多少个字符。varchar 查询速度快于 text,在都创建索引的情况下，text 的索引几乎不起作用。
查询 text 需要创建临时表。

问题 11：varchar(50)中 50 的含义
最多存放 50 个字符，varchar(50)和(200)存储 hello 所占空间一样，但后者在排序时会消耗更多内存，因为 order by col 采用 fixed_length 计算 col 长度(memory 引擎也一样)。

问题 12：int(20)中 20 的含义
是指显示字符的长度，不影响内部存储，只是当定义了 ZEROFILL 时，前面补多少个 0

问题 13：简单描述 MySQL 中，索引，主键，唯一索引，联合索引的区别，对数据库的性能有什么影响？

知识点分析

此真题主要考察的是 MySQL 索引的基础和类型，由此延伸出的知识点还包括如下内容：

MySQL 索引的创建原则
MySQL 索引的注意事项

MySQL 索引的原理

下面我们就来将这些知识一网打尽

索引的基础

索引类似于书籍的目录，要想找到一本数的某个特定主题，需要先查找书的目录，定位对应的页码
存储引擎使用类似的方式进行数据查询，先去索引当中找到对应的值，然后根据匹配的索引找到对应的数据行。

创建索引的语法：

首先创建一个表：create table t1 (id int primary key,username varchar(20),password varchar(20));
创建单个索引的语法：CREATE INDEX 索引名 on 表名（字段名）
索引名一般是：表名_字段名
给 id 创建索引：CREATE INDEX t1_id on t1(id);
创建联合索引的语法：CREATE INDEX 索引名 on 表名（字段名 1，字段名 2）
给 username 和 password 创建联合索引：CREATE index t1_username_password ON t1(username,password)
其中 index 还可以替换成 unique，primary key，分别代表唯一索引和主键索引
删除索引：DROP INDEX t1_username_password ON t1

索引对性能的影响：

大大减少服务器需要扫描的数据量。
帮助服务器避免排序和临时表。
将随机 I/O 变顺序 I/O。
大大提高查询速度。
降低写的速度（不良影响）。
磁盘占用（不良影响）。

索引的使用场景：

对于非常小的表，大部分情况下全表扫描效率更高。

中到大型表，索引非常有效。
特大型的表，建立和使用索引的代价会随之增大，可以使用分区技术来解决。

索引的类型：
索引很多种类型，是在 MySQL 的存储引擎实现的。

普通索引：最基本的索引，没有任何约束限制。
唯一索引：和普通索引类似，但是具有唯一性约束。
主键索引：特殊的唯一索引，不允许有空值。

索引的区别：

-一个表只能有一个主键索引，但是可以有多个唯一索引。
主键索引一定是唯一索引，唯一索引不是主键索引。
主键可以与外键构成参照完整性约束，防止数据不一致。
联合索引：将多个列组合在一起创建索引，可以覆盖多个列。（也叫复合索引，组合索引）
外键索引：只有 InnoDB 类型的表才可以使用外键索引，保证数据的一致性、完整性、和实现级联操作（基本不用）。
全文索引：MySQL 自带的全文索引只能用于 MyISAM，并且只能对英文进行全文检索 （基本不用）

MySQL 索引的创建原则

最适合创建索引的列是出现在 WHERE 或 ON 子句中的列，或连接子句中的列而不是出现在 SELECT 关键字后的列。
索引列的基数越大，数据区分度越高，索引的效果越好。
对于字符串进行索引，应该制定一个前缀长度，可以节省大量的索引空间。
根据情况创建联合索引，联合索引可以提高查询效率。
避免创建过多的索引，索引会额外占用磁盘空间，降低写操作效率。
主键尽可能选择较短的数据类型，可以有效减少索引的磁盘占用提高查询效率。

MySQL 索引的注意事项

1、联合索引遵循前缀原则

```
KEY(a,b,c)
WHERE a = 1 AND b = 2 AND c = 3
WHERE a = 1 AND b = 2
WHERE a = 1
#以上 SQL 语句可以用到索引
WHERE b = 2 AND c = 3
WHERE a = 1 AND c = 3
#以上 SQL 语句用不到索引
```

2、LIKE 查询，%不能在前

```
WHERE name LIKE "%wang%"
#以上语句用不到索引，可以用外部的 ElasticSearch、Lucene 等全文搜索引擎替代。
```

3、列值为空（NULL）时是可以使用索引的，但 MySQL 难以优化引用了可空列的查询,它会使索引、索引统计和值更加复杂。可空列需要更多的储存空间，还需要在 MySQL 内部进行特殊处理。

4、如果 MySQL 估计使用索引比全表扫描更慢，会放弃使用索引，例如：
表中只有 100 条数据左右。对于 SQL 语句 WHERE id > 1 AND id < 100，MySQL 会优先考虑全表扫描。

5、如果关键词 or 前面的条件中的列有索引，后面的没有，所有列的索引都不会被用到。

6、列类型是字符串，查询时一定要给值加引号，否则索引失效，例如：
列 name varchar(16)，存储了字符串"100"
WHERE name = 100;
以上 SQL 语句能搜到，但无法用到索引。

MySQL 索引的原理

MySQL 索引是用一种叫做聚簇索引的数据结构实现的，下面我们就来看一下什么是聚簇索引。

聚簇索引是一种数据存储方式，它实际上是在同一个结构中保存了 B+树索引和数据行，InnoDB 表是按照聚簇索引组织的（类似于 Oracle 的索引组织表）。

注：

B+ 树是一种树数据结构，是一个 n 叉排序树，每个节点通常有多个孩子，一棵 B+树包含根节点、内部节点和叶子节点。根节点可能是一个叶子节点，也可能是一个包含两个或两个以上孩子节点的节点。

B+ 树通常用于数据库和操作系统的文件系统中。NTFS，ReiserFS，NSS，XFS，JFS，ReFS 和 BFS 等文件系统都在使用 B+树作为元数据索引。B+ 树的特点是能够保持数据稳定有序，其插入与修改拥有较稳定的对数时间复杂度。B+ 树元素自底向上插入。

InnoDB 通过主键聚簇数据，如果没有定义主键，会选择一个唯一的非空索引代替，如果没有这样的索引，会隐式定义个主键作为聚簇索引。
下图形象说明了聚簇索引表(InnoDB)和普通的堆组织表(MyISAM)的区别：

最常问的 MySQL 面试题三——每个开发人员都应该知道
对于普通的堆组织表来说（右图），表数据和索引是分别存储的，主键索引和二级索引存储上没有任何区别。
而对于聚簇索引表来说（左图），表数据是和主键一起存储的，主键索引的叶结点存储行数据，二级索引的叶结点存储行的主键值。
聚簇索引表最大限度地提高了 I/O 密集型应用的性能，但它也有以下几个限制：

1）插入速度严重依赖于插入顺序，按照主键的顺序插入是最快的方式，否则将会出现页分裂，严重影响性能。因此，对于 InnoDB 表，我们一般都会定义一个自增的 ID 列为主键。
2）更新主键的代价很高，因为将会导致被更新的行移动。因此，对于 InnoDB 表，我们一般定义主键为不可更新。
3）二级索引访问需要两次索引查找，第一次找到主键值，第二次根据主键值找到行数据。

二级索引的叶节点存储的是主键值，而不是行指针，这是为了减少当出现行移动或数据页分裂时二级索引的维护工作，但会让二级索引占用更多的空间。

解题方法

在一些 MySQL 索引基础考题中，我们可以轻松的通过索引基础和类型来解决此类问题，对于一些索引创建注意事项方面的考点，我们可以通过索引创建原则和注意事项来解决。

问题 14：创建 MySQL 联合索引应该注意什么？
需遵循前缀原则

问题 15：列值为 NULL 时，查询是否会用到索引？
在 MySQL 里 NULL 值的列也是走索引的。当然，如果计划对列进行索引，就要尽量避免把它设置为可空，MySQL 难以优化引用了可空列的查询,它会使索引、索引统计和值更加复杂。

*问题 16：以下语句是否会应用索引：SELECT FROM users WHERE YEAR(adddate) < 2007;**
不会，因为只要列涉及到运算，MySQL 就不会使用索引。

问题 17：MyISAM 索引实现？
MyISAM 存储引擎使用 B+Tree 作为索引结构，叶节点的 data 域存放的是数据记录的地址。MyISAM 的索引方式也叫做非聚簇索引的，之所以这么称呼是为了与 InnoDB 的聚簇索引区分。

问题 17：MyISAM 索引与 InnoDB 索引的区别？

InnoDB 索引是聚簇索引，MyISAM 索引是非聚簇索引。
InnoDB 的主键索引的叶子节点存储着行数据，因此主键索引非常高效。
MyISAM 索引的叶子节点存储的是行数据地址，需要再寻址一次才能得到数据。
InnoDB 非主键索引的叶子节点存储的是主键和其他带索引的列数据，因此查询时做到覆盖索引会非常高效。

问题 18：以下三条 sql 如何建索引，只建一条怎么建？

```
WHERE a=1 AND b=1
WHERE b=1
WHERE b=1 ORDER BY time DESC
```

以顺序 b,a,time 建立联合索引，CREATE INDEX table1_b_a_time ON index_test01(b,a,time)。因为最新 MySQL 版本会优化 WHERE 子句后面的列顺序，以匹配联合索引顺序。

问题 19：有 A(id,sex,par,c1,c2),B(id,age,c1,c2)两张表，其中 A.id 与 B.id 关联，现在要求写出一条 SQL 语句，将 B 中 age>50 的记录的 c1,c2 更新到 A 表中同一记录中的 c1,c2 字段中

考点分析

这道题主要考察的是 MySQL 的关联 UPDATE 语句

延伸考点：

MySQL 的关联查询语句
MySQL 的关联 UPDATE 语句

针对刚才这道题，答案可以是如下两种形式的写法：

```
UPDATE A,B SET A.c1 = B.c1, A.c2 = B.c2 WHERE A.id = B.id
UPDATE A INNER JOIN B ON A.id=B.id SET A.c1 = B.c1,A.c2=B.c2
```

再加上 B 中 age>50 的条件：

```
UPDATE A,B set A.c1 = B.c1, A.c2 = B.c2 WHERE A.id = B.id and B.age > 50;
UPDATE A INNER JOIN B ON A.id = B.id set A.c1 = B.c1,A.c2 = B.c2 WHERE B.age > 50
```