Q1：主键，外键，超键，候选键；

主键 对表中数据进行唯一标识的数据列的组合；依靠主键就能找到这条数据

外键 该列为另一表的主键；

Q2：数据库事务的四个特征及含义；

原子性 要么全部完成，要么不完成，若发生错误会进行回滚操作；

一致性 数据保持动态一致，如A给B转账100，那么A会减掉100，B加上100；

隔离性 事务与事务之间相隔离，串行化执行；

持久性 事务完成对数据的影响是永久的；

Q7：数据库范式

1NF数据库中每一个列不可再分 （不能大表套小表）

2NF 非主属性必须完全依赖于主属性（排除部分依赖）；

3NF 非主属性不能传递依赖于主属性（排除传递依赖）；

Q10：数据库隔离级别

读未提交：会读取到别的实物未提交的数据

读提交：只能读取已经提交的数据

隔离性： 并发访问数据库时，一个用户的事物不被其他事物所干扰，各并发事务之间数据库是独立的；

持久性: 一个事务被提交之后。它对数据库中数据的改变是持久的，即使数据库 发生故障也不应该对其有任何影响。

Q11：MYSQL的两种存储引擎

MYISAM 不支持事务，不可恢复，基于文件存储，查询效率比较高

INNODB 支持事务，外键，行锁，可恢复

Q12：MYSQL索引算法

HASH 适合等值查找，不适合范围，不能排序

BTREE 适合范围查找，无hash冲突

Q5：索引的工作原理和其种类；

索引分为B+树索引和哈希索引

B+树索引适合做区间查找，如查找id为100-200之间的数据

哈希索引适合做精确的查找，找具体的一条数据

按照使用场景有

唯一索引 不允许任何两行具相同值

主键索引 唯一索引的一种

聚集索引 行的物理顺序和键值的索引顺序相同

InnoDB 主要有B+树索引的哈希索引

Q13：聚集索引和非聚集索引

聚集索引 数据按索引顺序存储，节点存储的是真实数据

非聚集索引 节点存储的是指向真正数据的指针

Q14：索引的优缺点

优点 提高查询效率

缺点 降低了更新效率，每次更新数据的时候需要同时维护索引，导致性能下降，索引索引不是越多越好，需要适度

Q17：数据库连接池

数据库连接池用于管理数据库连接资源，先生成一些数据库连接对象，用的时候直接取出使用，有利于提高数据库连接的效率，并且重复使用数据库连接对象，节约内存资源。

Q3：视图的作用，可以更改吗；

视图是虚拟的表；只包含动态检索数据的查询，不包含数据；简化操作，隐藏细节，保护数据；对视图的更新会作用于基表，一般不更新；

Q6：连接的种类

内部联接（等值联接） INNER JOIN

外部联接（包含在相关表中没有关联行的行）

Q8：存储过程与触发器的区别

存储过程和触发器都是SQL语句集；触发器是在用户执行某些语句后自动调用；存储过程存在于数据库，并且经过编译，使用call调用，效率比较高

Q19：乐观锁和悲观锁

乐观锁 假定不会发生并发冲突，只在提交时检查，若有其他数据更新了数据，则回滚；使用数据版本标示数据（时间戳，版本号）

悲观锁 假定会发生并发冲突，屏蔽一切破坏数据库一致性的操作，主要用于数据争用激烈的环境，以及锁成本低于回滚成本时；排他锁；