面试问题整理

数据库篇

乐观锁

- 只在提交操作时检查,先判断此刻 version 的值是否与刚刚查询出来时的 version 的值相等,对那条记录进行操作(更新)
- need framework

悲观锁

- 每次操作时都要通过获取锁才能进行对相同数据的操作，这点跟 java 中的 synchronized 很相似
- built-in

超键、候选键、主键、外键分别是什么？

- 学生（学号，姓名，性别，身份证号，教师编号） 教师（教师编号，姓名，工资）
- 超键 (super key)：学生表中含有**学号**或者**身份证号**的任意组合都为此表的超键。如：（学号）、（学号，姓名）、（身份证号，性别）等。
- 候选键 (candidate key) ： 候选键属于超键，就是说如果再去掉候选键中的任何一个属性它就不再是超键了。学生表中的候选键为：（学号）、（身份证号）。
- 主键 (primary key) ： 主键就是候选键里面的一个，是人为规定的，例如学生表中，“学号”做主键，教师表中“教师编号”做主键。
- 外键 (foreign key) ： 学生表中的外键就是“教师编号”。外键主要是用来描述两个表的关系。

1. 超键：在关系中能唯一标识元组的属性集称为关系模式的超键。一个属性可以为作为一个超键，多个属性组合在一起也可以作为一个超键。超键包含候选键和主键。
2. 候选键：是最小超键，即没有冗余元素的超键。
3. 主键：数据库表中对储存数据对象予以唯一和完整标识的数据列或属性的组合。一个数据列只能有一个主键，且主键的取值不能缺失，即不能为空值（Null）。
4. 外键：在一个表中存在的另一个表的主键称此表的外键。

三个范式(https://www.iteye.com/blog/aijuans-1629645)

1. 第一范式（1NF）：是原子性，字段不可再分割；
2. 第二范式（2NF）：完全依赖，没有部分依赖(No unnecessary input key)；
3. 第三范式（3NF）：在第二范式的基础上，数据表中如果不存在非关键字段对任一候选关键字段的传递函数依赖则符合第三范式。所谓传递函数依赖，指的是如果存在"A → B → C"的决定关系，则 C 传递函数依赖于 A。因此，满足第三范式的数据库表应该不存在如下依赖关系： 关键字段 → 非关键字段 x → 非关键字段 y

MySQL 的索引要使用 B+树?

因为 B 树不管叶子节点还是非叶子节点，都会保存数据，这样导致在非叶子节点中能保存的指针数量变少（有些资料也称为扇出）

指针少的情况下要保存大量数据，只能增加树的高度，导致 IO 操作变多，查询性能变低；

Python

https://zhuanlan.zhihu.com/p/23526961?refer=passer

垃圾回收

- 引用计数：Python 在内存中存储每个对象的引用计数，如果计数变成 0，该对象就会消失，分配给该对象的内存就会释放出来。
- 标记-清除：容器对象(list、dict、instance)可能会出现引用循环，，垃圾回收器会定时回收这些循环（对象互相引用（指针），构成一个有向图，对象:有向图的节点，引用关系:有向图的边）,根对象就是全局变量、调用栈、寄存器,可达的（reachable）对象标记为活动对象，不可达的对象就是要被清除的非活动对象。
- 分代回收：创建>释放数量，当创建数与释放数量的差值达到规定的阈值，Python 把内存根据对象存活时间划分为三代，对象创建之后，垃圾回收器会分配它们所属的代，而被分配更年轻的代是被优先处理的，因此越晚创建的对象越容易被回收。

java

设计模式

单例模式

```
饿汉模式
public class Singleton{
    private static Singleton instance = new Singleton();
    private Singleton(){}
    public static Singleton newInstance(){
        return instance;
    }
}
```

```
懒汉模式
public class Singleton{
    private static Singleton instance = null;
    private Singleton(){}
    public static Singleton newInstance(){
        if(null == instance){
            instance = new Singleton();
        }
        return instance;
    }
}
```

```
双重校验锁
public class Singleton {
    private static Singleton instance = null;
    private Singleton(){}
    public static Singleton getInstance() {
        if (instance == null) {   // Single Checked
            synchronized (Singleton.class) {
                if (instance == null) { // Double checked
                    instance = new Singleton();
                }
            }
        }
        return instance;
    }
}
```

- 不过还需要考虑一种情况，假如两个线程 A、B， A 执行了 if (instance == null)语句，它会认为单例对象没有创建，此时线程切到 B 也执行了同样的语句，B 也认为单例对象没有创建，然后两个线程依次执行同步代码块，并分别创建了一个单例对象。为了解决这个问题，还需要在同步代码块中增加 if (instance == null)语句，也就是上面看到的代码中的校验 2。

公司信息

华为是信息与通信技术解决方案供应商，覆盖手机、移动宽带终端、终端云等。就责任而言，它担起了一个通信领域的大公司应该承担的责任。让中国老百姓用起了便宜质量好的手机，而且改变了中国电子产品和核心的芯片都需要进口的局面，甚至还做到了出口，是民族骄傲。如果从华为职工角度来说的话，在华为虽然辛苦，但是付出和收获一定是成正比的，很少听到华为员工抱怨工资低的。