

第 1 题(本题2分): 假设有一个员工表 Employee(ID, Name, Address, Phone#) (其中ID为主键)。最初电话号码Phone#是一个单值属性, 即每个人只有一个电话。后来发现, 有少量员工有两个电话号码, 且都需要记录。你会如何对数据库的结构设计进行调整?

- ☐ A: 将Phone#的数据类型变成字符串, 将两个电话号码用分号隔开, 一起存在同一个字段中。✗
- ☐ B: 将主键变为 (ID, Phone#), 将有二个电话号码的人在表中存两份。✗
- ☐ C: 将模式修改为 Employee(ID, Name, Address), Phone(ID, Phone#)。✗
- ☒ D: 将模式修改为 Employee(ID, Name, Address, Phone#, Phone#Alt), 其中Phone#Alt用于记录另一个电话号码。✓

第 2 题(本题2分): 关于数据冗余的说法, 哪一项是不正确的?

- ☐ A: 冗余会增加数据更新的代价 ✗
- ☐ B: 冗余会增加数据删除的代价 ✗
- ☒ C: 冗余不会增加数据查询的代价 ✓
- ☐ D: 冗余会增加软件开发的复杂度 ✗

第 3 题(本题2分): 对一张属性很多的表 (通常称为“宽表”), 为了提升表访问的性能, 我们可以选择将表拆成两份或多份。比如, 将 Employee (ID, name, salary, tax, mgr#, dept#) 拆成 Employee (ID, name, salary) 和 Employee (ID, tax, mgr#, dept#)。请问做表拆分的原则应该是什么?

- ☒ A: 让同时被使用到的属性 (即出现在同一个SQL中的属性) 尽可能位于分拆后的同一张表中。✓
- ☐ B: 让经常被更新的属性尽可能位于分拆后的同一张表中。✗
- ☐ C: 让被索引的属性尽可能位于分拆后的同一张表中。✗
- ☐ D: 让可能为空的属性尽可能位于分拆后的同一张表中。✗

第 4 题(本题2分): 假设有一张表: 行程表 (旅客号码, 导游号码, 打卡景点), 记录了哪些旅客被哪些导游带领去了哪些地方。我们发现了如下形式的规律: 不同的旅客在同一位导游的带领下, 如果去了景点A, 那么必定也会去景点B。请问数据库中是否存在冗余信息, 可以如何处理?

- ☐ A: 没有冗余 ✗
- ☐ B: 有冗余, 但无法通过结构设计去除 ✗
- ☒ C: 有冗余, 适当改变结构设计后可去除 ✓
- ☐ D: 以上说法均不对 ✗

在这张行程表 (旅客号码, 导游号码, 打卡景点) 中, 如果存在规律, 即在同一位导游带领下, 去了景点 A 的旅客必定也会去景点 B, 那么数据库中就存在冗余信息。这是因为对于任何一位导游, 我们都可以推断出去景点 A 的旅客将去景点 B, 而不需要在表中为每个旅客重复这一信息。