第 1 题(本题2分)：声明式编程语言的优势有哪些？(1) 程序简单；(2) 程序运行效率高；(3) 程序运行过程可控。

A：(1) (2) (3) ✗

B：(1) (2) ✗

C：(1) ✓

D：都不是 ✗

不可控，因为只描述了想要什么数据，不管你怎么找到这样的数据

声明式编程语言的优势包括但不限于以下几点：

程序简单：声明式编程语言允许程序员描述他们想要的结果，而不需要详细说明如何达到这个结果。这意味着代码通常更简洁、更易于理解。

维护性和可读性：由于代码更加简洁，它通常更容易维护和理解。这有助于团队协作和代码的长期维护。

减少副作用：声明式编程往往强调不可变性和纯函数，这减少了因为状态变化而引起的复杂副作用。

可重用性：声明式代码因为其高级抽象，通常更容易被重用于不同的上下文中。

并行计算：在某些情况下，声明式代码更容易并行化，因为它不关心操作的顺序，这可以在现代多核处理器上提高效率。

关于选项中的其他点：

程序运行效率高：这并不总是正确的。虽然在某些场合声明式代码可以优化以提高运行效率，但是在其他场合，过度的抽象可能导致性能损失，尤其是当底层操作系统或硬件的细节对性能影响重大时。

程序运行过程可控：声明式语言通常更难以控制程序的具体执行过程，因为它抽象了执行细节。在需要精确控制性能和资源使用的应用中，命令式编程可能提供更多的可控性。

第 2 题(本题2分)：关于关系代数的描述，不正确的有：

A：关系代数的计算对象是集合 ✗

B：关系代数的计算结果是集合 ✗

C：一个关系也是一个集合 ✗

D：关系代数的计算符号有且仅有：选择、投影、连接 ✓

第 3 题(本题2分)：请思考，索引可以用于加速关系代数中的哪些计算?

- A：仅选择 ✗
- B：选择和投影 ✗
- C：选择和连接 ✗
- D：选择、投影和连接 ✓

索引是数据库系统中用于加速数据检索的数据结构。在关系数据库中，索引可以显著提高查询性能，尤其是对于选择、投影和连接操作：
选择（Selection）：
索引允许数据库快速定位满足特定条件的记录，而无需扫描整个表。例如，如果一个选择操作是基于某个被索引的列的值，数据库可以直接使用索引来快速找到满足条件的行，而不是逐行检查。
投影（Projection）：
索引可以包含一个或多个列的键值和指向实际存储记录的指针。因此，如果投影操作只包含索引中的列，数据库可以仅通过访问索引来检索必要的数据，而无需访问表中的行。
此外，对于包含排序或去重操作的投影，如果索引已经按照需要的顺序维护了数据，那么可以直接利用索引进行高效的排序或去重。
连接（Join）：
在执行连接操作时，如果参与连接的列被索引，数据库可以更有效地匹配来自不同表的行。例如，使用索引嵌套循环连接可以对一个表进行全表扫描，而对另一个表使用索引查找匹配的行。
索引还可以用于优化其他类型的连接，如 merge join 或 hash join。例如，在 merge join 中，如果两个表在连接列上都有排序的索引，那么连接操作可以非常迅速地进行，因为不需要额外的排序步骤。
总结来说，索引通过减少需要检查的数据量，减少磁盘 I/O 操作，提供排序和快速查找路径，从而加快了选择、投影和连接操作的处理速度。然而，值得注意的是，索引也有其维护成本，因为它们需要在数据插入、更新或删除时同步更新，这可能会影响写操作的性能。

第 4 题(本题2分)：关于在表A和表B上实施的连接操作，哪种说法不对?

- A：连接结果的元组个数可能比表A和表B的元组个数都多。 ✗
- B：连接结果的元组个数可能比表A和表B的元组个数都少。 ✗
- C：连接结果的元组个数不可能比表A和表B的元组个数都少。 ✓
- D：如果连接条件为A.x=B.x，且属性x在表A中没有重复取值，那么连接结果的元组个数不可能比表B的元组个数多。 ✗

第 4 题(本题2分)：关于在表A和表B上实施的连接操作，哪种说法不对?

A：连接结果的元组个数可能比表A和表B的元组个数都多。

B：连接结果的元组个数可能比表A和表B的元组个数都少。

C：连接结果的元组个数不可能比表A和表B的元组个数都少。

D：如果连接条件为A.x=B.x，且属性x在表A中没有重复取值，那么连接结果的元组个数不可能比表B的元组个数多。