1. 关系数据库让用户自行定义每一张表的Primary Key（主键），用于唯一识别表中的每一行数据。例如学生表student(sno, sname, birthday, gender)的主键可定义为为sno，宿舍表rooms(dorm\_no, room\_no, size, floor)的主键可定义为(dorm\_no, room\_no)（由宿舍号和房间号组成的复合主键）。在文档数据库中，用户无需定义主键，每一个文档都可以由系统自动产生的省缺ID进行识别。换句话说，文档数据库的ID属性起到了Primary Key的作用。请思考：关系数据库的Primary Key机制和文档数据库的ID机制有什么不同，各自的优缺点是什么？

**关系数据库的Primary Key机制与文档数据库的ID机制的不同，以及各自的优缺点：**

**不同：**

1、关系数据库的主键是由用户定义的，用于保证表中记录的唯一性。用户可以选择一个或多个列作为主键，而这些列的值必须是唯一的，且不可为空。

2、文档数据库中的ID通常是由系统自动生成的，保证每个文档的唯一性。这个ID是文档创建时自动分配的，用户无需事先定义。

**优缺点：**

**关系数据库主键：**

**优点：**用户定义的主键允许对数据模型有更严格的控制，可以根据业务规则选择合适的主键，这有助于确保数据的完整性和业务逻辑的一致性。

**缺点：**用户必须预先知道哪些字段可以唯一标识记录，这在某些情况下可能不是很明显。同时，主键的选择可能会对数据库性能产生重大影响，不当的选择可能导致性能问题。

**文档数据库ID：**

**优点：**系统自动生成的ID减少了用户的工作量，使得用户不必担心唯一性和主键的选择问题。这也使得文档数据库可以更灵活地处理不同结构的数据。

**缺点：**自动生成的ID可能缺乏业务意义，不如用户定义的主键直观，且在需要维护数据之间的关系时可能不如手动定义的主键那样灵活。

2. 关系数据库要求用户在使用一张表之前用DDL对表进行事先定义，并且给出表需满足的各种约束（比如主键、外键等）。文档数据库则不同，它通常不要求用户对文档集的结构做事先定义，甚至允许用户往文档集中插入任意结构的文档。请思考：关系数据库和文档数据库为什么使用了两种不同的功能设计？背后的原因是什么？

关系数据库通过DDL定义表的结构和约束，因为它们通常用于更复杂的查询和事务处理，其中数据的完整性、一致性和关系是非常重要的。这种设计支持强类型和结构化查询，有利于维护大型、复杂的数据系统。

文档数据库不要求预定义结构，因为它们旨在提供更高的灵活性和扩展性，尤其是在处理半结构化或非结构化数据时。这种设计允许快速开发和迭代，适用于快速变化的数据模型和需求。

背后的原因是关系数据库和文档数据库服务于不同的使用案例。关系数据库强调ACID属性，适合需要严格数据完整性和预定义模式的应用。而文档数据库更适合于需要处理大量变化数据、不需要事先定义固定模式的应用场景。

3. 如果我们在属性A上定义了Primary Key或Unique这样的约束，那么数据库将要求表中的任何一个元组在A上的取值都是唯一的。也就是说，当我们往表中重复插入A值时，数据库将立即感知到，并禁止我们这样做。请问：数据库是如何感知我们往表中重复插入A值的？

1、数据库通过在属性A上建立索引来感知重复值。当定义了主键或唯一约束时，数据库会自动为这些字段创建一个唯一索引。

2、当尝试插入新的记录时，数据库会在这个唯一索引中查找是否存在相同的A值。

3、如果索引查找表明已经存在一个相同的A值，数据库将拒绝新的插入操作并返回一个错误，通知用户违反了唯一性约束。

4、这种检查几乎是即时的，因为索引结构（如B树）是为快速检索和维护唯一性而优化的。

假设我们有三个关系（同课程中使用的例子一样），其模式分别为

Student(s\_no, s\_name, birthday, gender)

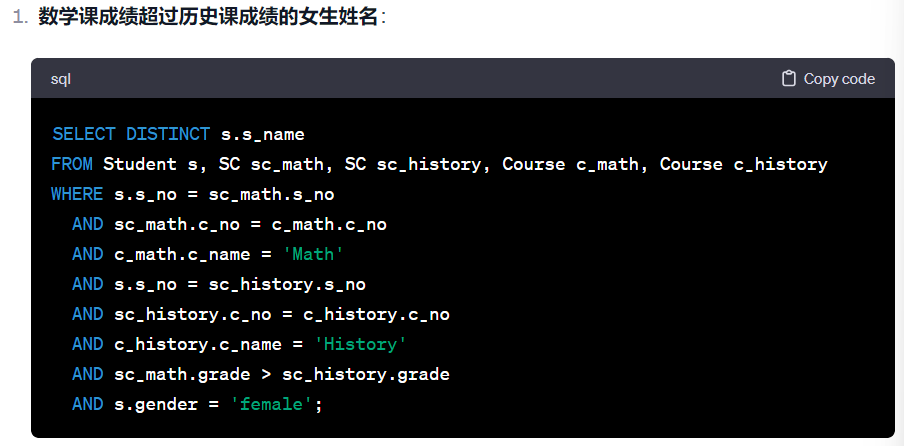
Course(c\_no, c\_name, credit)

SC(s\_no, c\_no, grade)请写出以下信息需求的SQL查询：

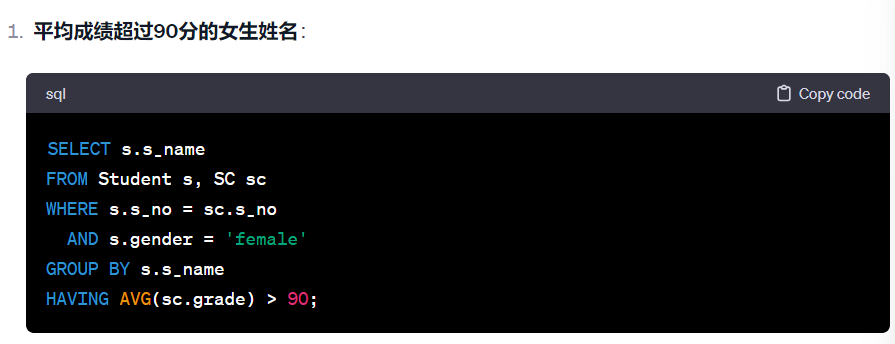
在数学课上成绩超过90分的男生姓名；



数学课成绩超过历史课成绩的女生姓名；



平均成绩超过90分的女生姓名。



假设我们有三个关系（同课程中使用的例子一样），其模式分别为

Student(s\_no, s\_name, birthday, gender)

Course(c\_no, c\_name, credit)

SC(s\_no, c\_no, grade)请写出满足以下访问需求的SQL语句：

删除所有平均成绩小于70分的课程；

找到在所有课程上的成绩都超过课程平均成绩的学生。

