数据库题

1、 数据库技术的发展分为哪几个阶段，各有什么特点？

（1） 人工管理阶段：数据的管理者是用户；面向的对象是某一应用程序；无共享，冗余度极大；不独立，完全依赖于程序；无结构；应用程序自己控制。
（2） 文件系统阶段：数据的管理者是文件系统；面向的对象是某一应用；共享性差，冗余度大；独立性差；记录内有结构、整体无结构；应用程序自己控制。
（3） 数据库系统阶段：数据的管理者是数据库管理系统；面向的对象是现实世界；共享性高，冗余度小；具有高度的物理独立性和一定的逻辑独立性；整体结构化，用数据模型描述；由数据库管理系统提供数据安全性、完整性、并发控制和恢复能力。
（4） 大数据阶段

2、 数据模型概念？数据模型作用？数据模型的组成的三个要素？

（1） 概念：数据模型是对现实世界数据特征的抽象，描述的是数据的共性内容。
（2） 作用：是模型化数据和信息的工具，也是数据库系统的核心和基础。
（3） 数据模型的三要素：
（4） 数据结构：描述的是数据库的组成对象以及对象之间的联系。
（5） 数据操作：是指对数据库中各种对象的实例允许执行的操作的集合，包括操作及有关的
（6） 数据的完整性约束条件：是一组完整性规则。

3、 关系代数的基本运算有哪些？

并、差、交、笛卡尔积、投影、选择、除运算。
（7） 选择 (σ)：从关系中选取满足给定条件的元组，生成新的关系。
（8） 投影 (π)：从关系中选取一部分属性，生成新的关系。
（9） 并 (∪)：将两个关系合并成一个不含重复元组的新关系。
（10）差 (-)：从一个关系中去掉另一个关系中的所有元组。
（11）笛卡尔积 (×)：两个关系之间的一种基本运算，将两个关系中的每个元组笛卡尔积作为新关系的元组，生成新的关系。

4、 函数依赖？1NF、2NF、3NF、BCNF 含义是什么，以及其相互间的关系？

1、函数依赖是一种数据属性之间的关系，其中一个或多个属性的值可以确定另一个属性的值。即根据给定的某个或某些属性可以确定其他属性的值。设 R(U)是属性集 U 上的关系模式，X、Y 是 U 的子集。若对于 R(U)的任意一个可能的关系 r，r 中不可能存在两个元组在 X 上的属性值相等，而在 Y 上的属性值不等，则称 X 函数确定 Y 或 Y 函数依赖于 X。
非平凡函数依赖、平凡函数依赖、完全函数依赖、部分函数依赖

- 第一正式形式（1NF）：每个属性不能再拆分成更小的属性，每个属性值都是原子的。关系模型必须是满足 1NF 的，以便于进行插入、更新、删除以及查询等操作。
- 第二正式形式（2NF）：在满足 1NF 的前提下，任何非主属性都必须完全依赖于关系的

主键属性。也就是说所有非主属性都必须与主键属性有一个仅与主键相关的关系。

- 第三正式形式（3NF）：在满足 2NF 的前提下，除了主键属性外的所有属性，其他非主属性间不应该相互依赖。
- 巴斯-科德范式（BCNF）：满足 3NF 的前提下，如果关系中的每个决策依赖都包含一个超码，则该关系就是符合 BCNF 的。它要求所有决策依赖项中，决策依赖项的关键码都只有一个属性。

这些范式之间的关系是，如果一个关系符合 BCNF 范式，那么它一定满足第三范式。同样地，如果一个关系符合第三范式，那么它一定符合第二范式和第一范式。实际上，设计关系模型时，通常应当优先满足 BCNF 范式，因为 BCNF 范式可以保证关系模型的数据冗余和数据插入异常最小。

3、1NF------(消除非主属性对码的部分函数依赖)------>2NF------(消除非主属性对码的传递函数依赖)------>3NF------(消除主属性对码的部分和传递函数依赖)------>BCNF------(消除非平凡且依赖的多值依赖)------->4NF

这些范式之间的关系是，如果一个关系符合 BCNF 范式，那么它一定满足第三范式。同样地，如果一个关系符合第三范式，那么它一定符合第二范式和第一范式。实际上，设计关系模型时，通常应当优先满足 BCNF 范式，因为 BCNF 范式可以保证关系模型的数据冗余和数据插入异常最小。

5、 试述 SOL 语言的特点？

（1） 综合统一。
（2） 高度非过程化。
（3） 面向集合的操作方法。
（4） 以同一种语法结构提供多种使用方法。
（5） 语言简洁，易学易用。
（6） 具有面向对象编程语言的特性：支持类、对象、封装、继承、多态等面向对象编程特性，SOL 语言允许使用面向对象的方式进行数据库操作和处理。
（7） 支持动态模型和静态模型：动态模型可以通过编程方式创建、修改和销毁数据类型、对象和属性。静态模型是一种在编写应用程序之前定义和管理数据模型的方式。
（8） 支持存储过程和触发器：存储过程是一种用于存储在数据库中的可重用查询和处理代码，可以在需要时对其进行调用。触发器是一种当数据库中的某些事件发生时自动触发执行的一种类型的存储过程。
（9） 简化了查询语句的编写过程：SOL 语言允许将嵌套查询合并为单个语句，从而简化了复杂查询的编写。它还支持 CTE（Common Table Expressions），这是一种能够创建临时表并在查询中引用的语法结构。
（10） 支持多种数据类型：SOL 语言支持多种常见数据类型，如整数、浮点数、字符串、日期等等，并在此基础上提供了很多方便的函数和操作符，以完成复杂的运算和处理。

6、 简述数据库的安全性和完整性？

（1） 数据库安全性是指保护数据库以防止不合法使用所造成的数据泄露，更改或破坏。
（2） 数据库的完整性是指数据库的正确性和相容性。
（3） 安全性：数据库的安全性可以分为授权、身份验证和审计等方面。授权可以限制用户访问数据的权限，身份验证可以保护用户的身份信息和密码的安全，审计可以跟踪和记录

用户的操作和行为。数据库的安全性还可以通过加密、数据备份和恢复等手段来实现数据的安全保护。

（4） 完整性：数据库的完整性主要包括实体完整性和参照完整性等方面。实体完整性可以保证每个表中的每个记录都是唯一的，即没有重复数据，参照完整性可以保证表与表之间的引用和关系的一致性，即引用的对象必须存在。完整性还可以通过检查约束、触发器等方式来保证数据的完整性。

7、 数据库恢复的基本技术有哪些？

数据库恢复是一种紧急的数据库管理技术，用于恢复数据库在发生故障或灾难性事件之后的状态。数据库恢复的基本技术包括以下几种：

（1） 日志恢复技术：数据库管理系统中的事务日志记录了数据库中的所有事务操作，包括插入、更新、删除数据等操作。当数据库系统出现故障时，它可以使用事务日志来恢复到上一个稳定的状态。常见的日志恢复技术包括前滚、后滚和回滚等。

（2） 冷备份和热备份：冷备份就是备份数据库文件，而不运行数据库实例。这样可以避免备份中的错误数据，但系统需停止服务。热备份则是运行数据库实例，将数据备份到另一个地方。这种备份方式可以确保系统不会因备份而停止服务。

（3） 数据库复制技术：数据库复制技术将原数据库的副本复制到另一个服务器上。当主数据库崩溃时，整个用户可以切换到副本数据库。这种备份方式能够提供较快的系统恢复速度，但是也要避免复制的数据库出现故障。

（4） RAID 技术：RAID 技术是通过应用磁盘阵列来提供数据冗余和恢复功能。RAID 技术可使系统免于单个磁盘故障所造成的数据丢失，虽然不能保证整个系统的稳健性，但是 RAID 技术作为一种磁盘组织结构的方案，已成为一种备份数据库的有效手段。

（5） 综上所述，数据库恢复技术可以提高系统抗灾和恢复的能力，保证数据库系统的安全性和可靠性，需要在数据库设计和管理过程中及时采取必要的恢复技术和措施。

8、 试述数据库设计的特点及设计步骤

数据库设计是构建一种有组织的、易于访问和维护的数据库模式的过程。数据库设计的特点包括以下几个方面：

（1） 目标导向：数据库设计应该是以目标为导向的，根据系统的实际情况和用户需求进行设计，以满足用户的需求和提高系统的效率。

（2） 模块化设计：数据库设计要采用模块化设计的思想，根据不同的功能模块将数据分离，从而提高系统的可维护性、可伸缩性和可扩展性。

（3） 可扩展性：数据库设计应该具有良好的可扩展性，以适应系统未来的发展和变化。

（4） 数据的完整性和安全性：对数据的完整性和安全性要有意识的保护，从设计时就要考虑到这些方面。

（1） 需求分析：收集并分析用户的需求。在需求分析中，确定数据库中需要存储哪些数据、数据的结构、数据的关系等。

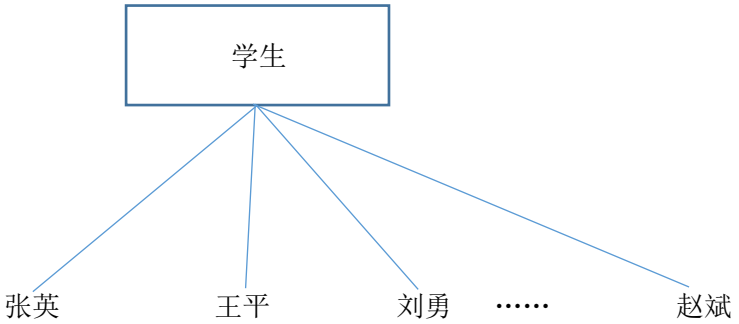
（2） 概念设计：在需求分析的基础上，建立数据库概念模型，包括实体-关系图（ER 图）和数据字典等，用于描述实体、属性和实体之间的关系。

（3） 逻辑设计：在概念设计的基础上，完成数据库逻辑设计，包括设计数据库中的表、字段、关系等，对数据进行规范化处理。

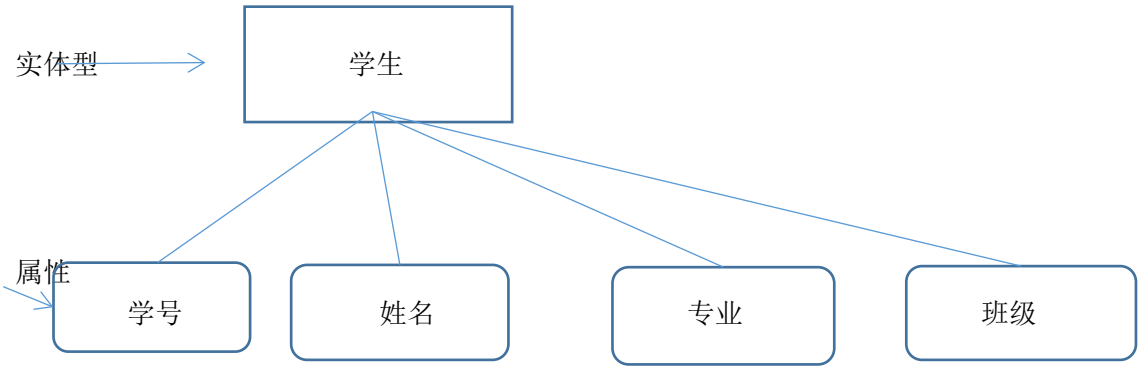
（4） 物理设计：在逻辑设计的基础上，考虑数据库的物理结构和存储方式，包括选择数据存储方式和索引设计等。
（5） 实施和运行：在物理设计的基础上，实施数据库系统，包括数据库的创建、表的创建、用户的创建等，并对系统进行测试、优化和维护。
（6） 维护和优化：对数据库系统进行维护和优化，包括备份和恢复、调整参数、优化查询等。并及时修复故障和缺陷。

9、 数据抽象的三种方法（分类、聚集、概括）

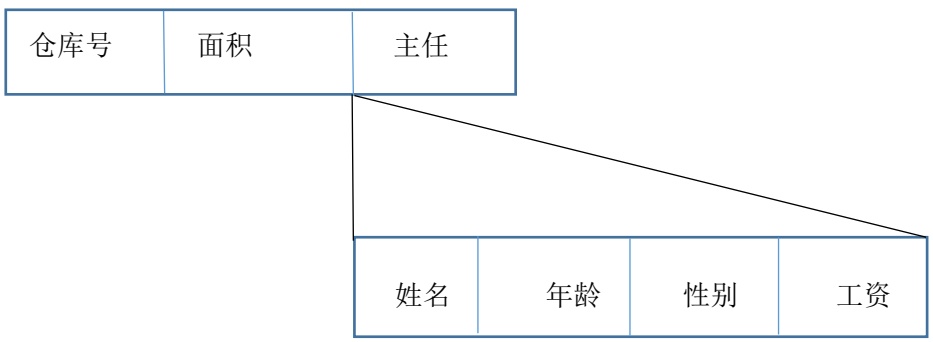
（1） 分类（Classification）：分类是指根据数据的一些特征将数据进行分类或分组的过程。比如，可以根据商品名称或价格将产品分类，这样就可以对大量的商品进行分类汇总，以便更好的进行管理。



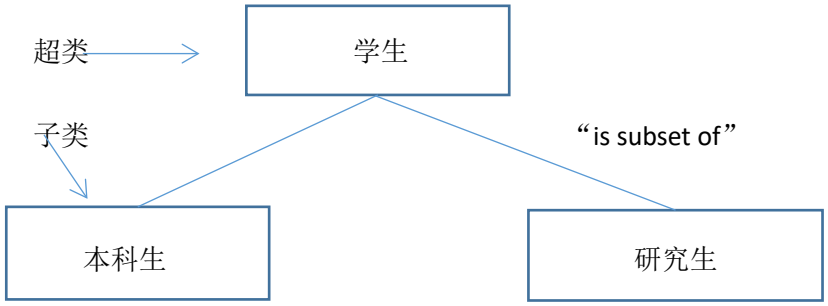
（2） 聚集是指将大量的数据进行聚合或汇总，以形成更高层次的总体或综合性知识。这种方法通常使用在大数据的处理中，比如根据某个时间段对销售额进行聚合，以形成一个汇总报表。



复杂的聚集，某一类型的成分仍是一个聚集



（3） 概括（Generalization）：概括是指将大量数据转换成一种更为简洁、易于理解的形式，以形成简洁、可理解的知识和信息。在数据可视化方面，概括是将一些复杂的数据视图用更简洁、易于理解的图表或示意图进行表示，使用户更容易理解数据。



10、 需求分析的任务、方法及意义

任务：通过详细调查现实世界要处理的对象（组织、部门、企业等），充分了解原系统的工资概况，明确用户的各种需求，然后在此基础上确定新系统的功能。新系统必须充分考虑今后可能的扩充和改变，不仅能按当前应用需求来设计数据库。

需求分析调查的重点是“数据”和“处理”，通过调查，收集与分析，获得用户对数据库的如下要求：

（1）信息要求。（2）处理要求。（3）安全性与完整性要求

需求分析的方法：

（1）跟班作业。（必写）（2）开调查会。（必写）（3）请专人介绍。（4）询问。（5）设计调查表请用户填写。（必写）（6）查阅记录。

需求分析的意义：需求分析报告包含数据字典。

11、 什么是基本表？什么是视图？两者的区别和联系是什么？简述视图（定义，查询，更新，作用）：

（1） 基本表：基本表是数据库设计中的一个基本概念，指的是数据库中的一个基本数据存储单元，是由一些列的列和一些行组成的。在关系型数据库中，基本表是用于存储数据的最基本的数据结构。所有的其他的数据库对象（如视图、存储过程等）都是基于基本表来创建的。

（2） 视图：视图是数据库中虚拟的表，是通过对一个或多个基本表进行查询而获得的。视图并不在数据库中实际储存数据，它仅仅是一个虚拟的表，它的数据是从所对应的基本表中提取的。视图可以隐藏实际数据表中的部分信息，以便用户可以只能看到他们需要的数据，还可以使复杂的查询更简单，从而提高了查询的效率。

（3） 基本表和视图之间的区别在于，基本表是实际储存数据的，而视图是虚拟的，不储存具体的数据，它仅仅是基于对其他表的查询来获取数据。基本表的更新可以直接更改数据，而视图的更新通常需要通过更改基本表中的数据来实现。

（4） 两者之间的联系在于，视图是基于基本表进行创建，视图通过在基本表上查询数据来返回一个新的表，它提供了基于不同的角度和条件查询基本表的便捷方式。在某些情况下，为了满足业务需求而需要从基本表中隐藏某些信息，或基于多个基本表中的数据创建虚拟表，这时就需要通过创建视图来满足业务需求。

简述视图的定义、查询、更新和作用：

（1） 视图的定义：视图是一种虚拟的数据表，它并不实际存在于数据库中，它是从一个或多个基本表通过查询操作获得的结果集。视图定义了一种视角，使得用户可以根据自己的需要访问数据。

（2） 视图的查询：视图的查询与普通的基本表查询相似，都需要使用 SELECT 命令进行查询。查询时，可以对视图进行条件过滤、排序、分组、连接等操作，返回结果与基本表类似，但是仅包含视图中查询语句所选取的数据。

（3） 视图的更新：视图的更新操作需要注意，它可以修改视图所对应的基本表，也可以修改视图本身（前提是它所对应的基本表不是分组聚集的结果，不包含聚集函数）。

（4） 视图的作用：视图可以隐藏数据表中某些字段，删除某些行，也可以通过关联多个表来整合数据，从而简化用户的查询操作。视图还可以实现使用权限的控制，通过授权的方式来限制用户对某些数据的访问。同时，在特定的情况下视图也可以用于查询优化，提高查询性能，避免数据重复。

12、 什么是 E-R 图？构成 E-R 图的基本元素是什么？举例说明。叙述集成 E-R 图的基本步骤。

答：含义：E-R 图是指实体-关系图（Entity-Relationship Diagram），它是用于描述现实世界中各种对象之间的关系和联系的一种图形化数据建模工具。它采用图形化的形式来表示现实世界中各种实体和它们之间的关系，以便更好地理解和分析系统的需求和功能。

E-R 图由三种基本元素构成：实体、属性和关系。

（1） 实体：实体指现实世界中的某个具体的对象或事物，如“学生”、“教师”、“订单”等。

（2） 属性：属性指实体所具有的特征或描述，如“学生”实体可以包含“姓名”、“性别”、“年龄”等属性。

（3） 关系：关系指实体之间的交互作用和联系，如“学生”与“教师”之间的“选课”关系。

集成 E-R 图的基本步骤包括以下几个方面：

（1） 确定实体集：确定系统中所有的实体集并用矩形表示，如“学生”、“教师”等。

（2） 确定实体间的关系：确定各实体之间的联系和关系，如“选课”、“授课”等。用菱形表示关系，用线条表示实体和关系之间的联系。具体的关系类型可以分为一对一、一对多和多对多。

（3） 定义属性：对实体集的属性进行定义，如“学生”实体包含“姓名”、“年龄”等属性。用椭圆形表示属性。

（4） 确定参照完整性：确定实体之间的关系和联系的参照完整性，以避免数据冗余和不一致性。

（5） 检查和输出：对 E-R 图进行检查和修改，并将构建出来的 E-R 图输出到文档或其他工具中，以供后续的软件开发工作。

13、 试述事务的概念及事务的四个特性，事务故障的恢复步骤。

事务是指指一组数据库操作，这些操作组成一个有逻辑意义的工作单元，要么全部执行，要么全部不执行。在关系型数据库中，事务是用来保证数据库一致性的重要机制之一。通常情况下，事务应该具有“原子性”、“一致性”、“隔离性”和“持久性”等四个特性，这些特性也被称为 ACID 特性。

四个事务特性

（1） 原子性（Atomicity）：事务中所有操作要么全部执行成功，要么全部撤销回滚操作。

（2） 一致性（Consistency）：事务的执行不应该破坏数据库的一致性，数据库在事务执行前后应该保持一致性状态。

（3） 隔离性（Isolation）：不同的事务之间应该相互隔离，不应该相互干扰。即一个事务内部的操作与其他事务的操作应该是相互独立的。

（4） 持久性（Durability）：一旦事务提交后，它对数据库所做的更改就应该永久保存在数据库中，并且对所有其他操作可见。

事务故障的恢复步骤

当数据库中的事务发生故障时，需要进行故障恢复以保证数据库的一致性和完整性。事务故障的恢复步骤如下：

（1） 记录所有未提交的事务：对于未提交的事务，需要记录下其所做的更改操作，以便在后续操作中进行恢复。

（2） 撤销所有未提交的事务：对于已经记录下来的未提交的事务，需要逆向其所做的更改操作，即进行回滚操作，把已经执行的操作进行撤销。

（3） 重做所有已提交的事务：对于已经提交的事务，需要重新执行其所做的更改操作，以确保数据库的所有数据都正确地重新恢复到最新的状态。

（4） 清除日志信息：删除所有记录未提交事务的日志信息，释放其占用的系统资源。

14、 如何用封锁机制保证数据的一致性。简述三级封锁协议，两段锁协议

封锁机制保证数据的一致性

封锁机制是一种用于保护数据库一致性的机制，通过对数据库中的资源进行锁定和解锁来保证多个用户操作的数据一致性。当一个用户访问一个数据库资源时，会发送一个获得锁的请求，如果锁没有被占用，就可以获得锁并开始对资源进行操作，当事务完成后，就会释放所占用的锁，以便其他用户可以访问这个资源。

三级封锁协议

（1） 第一级封锁协议：即独占锁（X 锁），一旦一个事务对一个数据对象获得了一个 X 锁，那么其他事务就不能再获得该数据对象上的任何锁，只有当前的事务提交或回滚后，其他事务才能获得该数据对象上的锁。

（2） 第二级封锁协议：即共享锁（S 锁），当一个事务对某个数据对象执行了 S 锁操作后，其他事务可以再对该数据对象进行 S 锁操作，但是不能进行 X 锁操作，而且其他事务无法获得该数据对象的 X 锁。

（3） 第三级封锁协议：即增量锁（IX 锁）和意向共享锁（IS 锁），增量锁允许一个事务在一个数据对象上执行 IX 锁和 S 锁操作，而其他事务无法获得 X 锁和 IX 锁，意向共享锁则表示一个事务准备对某个数据对象进行 S 锁操作，避免其他事务在该数据对象上执行 X 锁操作。

两段锁协议

两段锁协议是指事务在运行期间的某一时刻开始，把所需的所有数据对象加锁，直到事务结束为止。它分为两个阶段：

（1） 扩展期：在该阶段中，事务可以获得其所需的所有数据对象的锁，但不能释放锁。

（2） 收缩期：在该阶段中，事务释放所占用的所有锁，但不能再获取任何新的锁。

两段锁协议使可以通过合理的加锁和释放锁来保证数据库的一致性，避免了死锁的发生和减少锁粒度，提高了并发性