【全国】04735《数据库系统原理》

考前资料

考试题型及分值分布

类别	题型	题量	分值/题	总分
选择题	单项选择题	15 题	2 分	30 分
非选择题	填空题	10 题	1 分	10 分
	设计题	5 题	4 分	20 分
	简答题	5 题	6 分	30 分
	综合题	1 题	10 分	10 分

注：本资料题型根据考试大纲或历年真题进行预测，仅供复习参考

第一章　数据库系统概述

考点 1：数据库基本概念

数据、数据库、数据库管理系统和数据库系统是数据库中最常用的四个基本概念。

（1）数据（Data）：是描述事物的符号记录，是指用物理符号记录下来的、可以鉴别的信息。数据有多种表现形式，可以是包括数字、字母、文字、特殊字符组成的文本数据，也可以是图形、图像、动画、影像、声音、语言等多媒体数据。各种形式的数据经过数字化处理后可存入计算机，便于进一步加工、处理、使用。数据是信息存在的一种形式，只有通过解释或处理的数据才能成为有用的信息。

（2）数据库（DB）：通俗地被称为存储数据的仓库。所谓数据库是指长期储存在计算机中的有组织的、可共享的数据集合，且数据库中的数据按一定的数据模型组织、描述和存储，具有较小的冗余度、较高的数据独立性，系统易于扩展，并可以被多个用户共享。数据库中存储的数据具有永久存储、有组织和可共享三个基本特点。

（3）数据库管理系统（DBMS）：专门用于建立和管理数据库的一套软件，介于应用程序和操作系统之间。它负责科学有效地组织和存储数据，并帮助数据库的使用者能够从大量的数据中快速地获取所需数据，以及提供必要的安全性和完整性等统一控制机制，实现对数据有效的管理与维护。其主要功能包括：①数据定义功能；②数据操纵功能；③数据库的运行管理功能；④数据库的建立和维护功能；⑤数据组织、存储和管理功能；⑥其他功能：主要包括与其他软件的网络通信功能、不同数据库管理系统之间的数据传输以及相互访问功能等。

（4）数据库系统（DBS）：是指在计算机中引入数据库技术之后的系统。通常，一个完整的数据库系统包括数据库、数据库管理系统及相关实用工具、应用程序、数据库管理员和用户。一般在不引起混淆的情况下，常常将数据库系统简称为数据库。

考点 2：数据管理技术的发展

数据管理可以从两个方面来理解：一是针对组织业务的管理，负责制定并执行整个组织中关于数据的定义、组织、保护与有效使用的策略、过程和计划；二是依靠技术，负责实现数据作为一种资源的集中控制管理。数据管理的任务就是进行数据的收集、组织、控制、存储、选取、维护，实现在适当的时刻、以适当的形式、给适当的人、提供适当的数据，它是数据处理的中心问题，而数据处理则是指对各种数据进行收集、存储、加工和传播的一系列活动的总和。

随着计算机技术的发展及应用，数据管理技术共经历了人工管理、文件系统和数据库系统三个阶段。

（1）人工管理阶段

此阶段数据管理的特点有：①数据不保存；②应用程序管理数据；③数据面向应用。

（2）文件系统阶段

20 世纪 50 年代后期到 60 年代中期，计算机软、硬件技术发展到了一定阶段。其中，硬件方面配置了

磁盘、磁鼓等直接存取存储设备；软件方面，特别是在操作系统中配备了专门的数据管理软件，即文件系统。相对于人工管理数据的方法，文件系统管理数据有了很大的改进，具有数据可长期保存和专门管理的特点，它提供了物理数据独立性，使应用程序与数据的具体物理存储结构分离，并通过数据的抽取、排序、合并等可以为应用提供新的文件，从而使数据共享成为可能。但是，在文件系统中，不能实现数据的普通共享，只能实现文件级的共享，而不能在记录或数据项级实现数据的共享。

（3）数据库系统阶段

与人工管理、文件系统两种数据管理方法相比较，数据库系统具有如下一些特点：

①数据集成：是数据库管理系统的主要目的。

②数据共享性高：在数据库中，一个数据可以为多个不同的用户共同使用。

③数据冗余小：并非所有的冗余都可以或者应该被消除。

④数据一致性：引起不一致的根源是数据冗余。

⑤数据独立性高：数据库系统提供了两层数据独立，分别是数据的逻辑独立、数据的物理独立。

⑥实施统一管理与控制：数据库管理系统具有对数据的统一管理和控制功能，主要包括数据的安全性、完整性、并发控制与故障恢复等，即数据库保护。

A.数据的安全性：是指保护数据，以防止不合法的使用而造成数据泄密和破坏，使每个用户只能按规定对某些数据以某些方式进行使用和处理，即保证只有赋予权限的用户才能访问数据库中的数据，防止对数据的非法使用。

B.数据的完整性：是对数据的正确性、有效性和相容性要求，即控制数据在一定的范围内有效或要求数据之间满足一定的关系，保证输入到数据库中的数据满足相应的约束条件，以确保数据有效、正确。

C.并发控制：是指当多个用户的并发进程同时存取、修改数据库时，可能会发生相互干扰而得到错误结果，并使得数据库的完整性遭到破坏，因而对多用户的并发操作加以控制和协调。

D.故障恢复：计算机产生的硬件故障、操作员的失误以及人为的破坏都会影响数据库中数据的正确性，甚至造成数据库部分或全部数据的丢失，DBMS 必须具有将数据库从错误状态恢复到某一已知的正确状态的功能，这就是数据库的故障恢复。

⑦减少应用程序开发与维护的工作量。

考点 3：数据库系统的三级模式结构

数据库系统的三级模式结构是指数据库系统是由模式，外模式和内模式三级构成的。

（1）模式：也称为概念模式或逻辑模式，它是数据库中全体数据的逻辑结构和特征的描述，是所有用户的公共数据视图。模式实际上是数据库数据在逻辑级上的视图，即概念视图，其形式要比数据的物理存储方式抽象。

（2）外模式：也称为子模式或用户模式，它是数据库用户（包括应用程序员和最终用户）能够看见和使用的局部数据的逻辑结构和特征的描述，是与某一应用有关的数据的逻辑表示。外模式实际上是用于满足不同数据库用户需求的数据视图，即用户视图，其通常是模式的子集，是对数据库整体数据结构的局部重构。

（3）内模式：也称为存储模式，它是对数据库中数据物理结构和存储方式的描述，是数据在数据库内部的表示形式。内模式实际上是整个数据库的最底层表示，它不同于物理层，是数据库管理员（DBA）所见到的，特定的 DBMS 所处理的数据库的内部结构，即内部视图或存储视图。

（4）三级模式结构的两层映像与数据独立性

构成数据库系统三级模式结构的三个模式分别是对数据的三级抽象，它们彼此间具有如下一些特点：

①一个数据库的整体逻辑结构和特征的描述（概念模式）是独立于数据库其他层次结构（内/外模式）的描述，其是数据库的核心，也是数据库设计的关键。

②一个数据库的内部存储模式依赖于概念模式，但存储模式独立于外部模式，也独立于具体的存储设备。

③用户逻辑结构（外模式）是在全局逻辑结构描述的基础上定义的，它面向具体的应用程序，独立于内部模式和存储设备。

④特定的应用程序是在外模式的逻辑结构上编写的，它依赖于特定的外模式，与数据库的模式和存储结构独立。

用户可以不必考虑数据的物理存储细节，而把数据的具体组织留给 DBMS 负责管理，同时为了有效支撑数据的三级抽象以及它们相互间的联系和转换，DBMS 通过在内部提供三级模式之间的两层映像来实现，即外模式/模式映像与模式/内模式映像。

①外模式/模式映像定义了各个外模式与概念模式之间的映像关系，这些映像定义通常在各自的外模式中加以描述。由于同一个模式可以有任意多个外模式，因此对于每一个外模式，数据库系统都会有一个外模式/模式映像。数据库系统的模式如若发生改变，例如增加新的关系、新的属性、改变属性的数据类型等，数据库管理员（DBA）通常会对各个外模式/模式的映像做出相应的改变，以使那些对用户可见的外模式保持不变，从而应用程序的编程人员就不必去修改那些依据数据的外模式所编写的应用程序，如此实现了外模式不受概念模式变化的影响，并保证了数据与程序的逻辑独立性。

②模式/内模式映像定义了数据库全局逻辑结构与物理存储之间的对应关系，这种映像定义通常是在模式中加以描述的。由于数据库中只有一个模式，且也只有一个内模式，所以模式/内模式映像是唯一的。数据库系统的物理存储如若发生改变，例如选用另外一种存储结构或更换另外一个存储位置，数据库管理员（DBA）通常也会对模式/内模式映像做出相应的调整，以使数据库系统的模式保持不变，从而也不必去修改应用程序，如此实现了概念模式不受内模式变化的影响，并保证了数据与程序的物理独立性。

考点 4：数据库系统的运行与应用结构

从数据库系统应用的用户角度来看，目前数据库系统常见的运行与应用结构有：客户/服务器结构、浏览器/服务器结构。

（1）客户/服务器结构：在这种结构中，命令行客户端、图形化界面管理工具、应用程序等称为“客户端”、“前台”或“表示层”，主要完成与数据库使用者的交互任务；而数据库管理系统则称为“服务器”、“后台”或“数据层”，其主要负责数据管理。这种操作数据库的模式称为客户/服务器（C/S）模式。

（2）浏览器/服务器结构：是一种基于 Web 应用的客户/服务器结构，也称为三层客户/服务器结构。在

数据库系统中，它将与数据库管理系统交互的客户端进一步细分为“表示层”和“处理层”。其中，“表示层”是数据库使用者的操作和展示界面，通常由用于上网的各种浏览器构成，由此减轻数据库系统中客户端的工作负担；而“处理层”也称为“中间层”，则主要负责处理数据库使用者的具体应用逻辑，它与后台的数据库管理系统共同组成功能更加丰富的“胖服务器”。数据库系统的这种工作模式就称为浏览器/服务器（B/S）模式。

考点 5：数据特征与数据模型组成要素

一般而言，数据具有静态和动态两种特征。其中，数据的静态特征包括数据的基本结构、数据间的联系以及对数据取值范围的约束；数据的动态特征是指对数据可以进行符合一定规则的操作。数据模型是用来描述数据的结构、定义在结构上的操纵以及数据间的约束的一组概念和定义。

数据模型通常由数据结构、数据操作和数据约束三个要素组成，分别如下：

（1）数据结构：描述的是系统的静态特性，即数据对象的数据类型、内容、属性以及数据对象之间的联系。由于数据结构反映了数据模型最基本的特征，因此人们常常按照数据结构的类型来命名数据模型。

（2）数据操作：描述的是系统的动态特性，是对各种对象的实例允许执行的操作的集合，包括操作及有关的操作规则。数据操作主要分为更新和检索两大类，其中更新包括插入、删除和修改。

（3）数据约束：描述数据结构中数据间的语法和语义关联，包括相互制约与依存关系以及数据动态变化规则，以保证数据的正确性、有效性与相容性。数据约束包括数据完整性约束、数据安全性约束以及并发控制约束，数据约束既刻画了数据静态特征，也表示了数据动态行为规则。

考点 6：数据模型的分类

数据模型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。

（1）概念层数据模型

概念层是数据抽象级别的最高层，其目的是按用户的观点来对世界建模。概念层数据模型，也称为数据的概念模型或信息模型，它用来描述现实世界的事物，与具体的计算机系统无关，且独立于任何 DBMS，但容易向 DBMS 所支持的逻辑数据模型转换。这类模型主要用于数据库的设计阶段，即在设计数据库时，通常用概念模型来抽象、表示现实世界的各种事物及其联系。通常，信息世界涉及的基本概念有：

①实体：客观存在并可相互区别的事物称为实体。

②属性：实体所具有的某种特性称为实体的属性。

③码或键：可唯一标识实体的属性集称为码或键。

④域：属性的取值范围称为该属性的域。

⑤实体型：用实体名与属性名集合来抽象和刻画同类实体，称为实体型。

⑥实体集：同型实体的集合称为实体集。

⑦联系：实体内部的联系通常是指实体各属性之间的联系。实体之间的联系是指不同实体之间的联系。

（2）逻辑层数据模型

逻辑层是数据抽象的中间层，描述数据整体的逻辑结构。这一层的数据抽象称为逻辑层数据模型，也

称为数据的逻辑模型，它是用户通过数据库管理系统看到的现实世界，是基于计算机系统的观点来对数据进行建模和表示。逻辑数据模型的基本概念有：

①层次模型：是数据库系统最早使用的一种数据模型，它的数据结构是一棵“有向树”，树的每个结点对应一个记录集，也就是现实世界的实体集。层次模型的特点是：有且仅有一个结点没有父结点，它称作根结点；其他结点有且仅有一个父结点。我们所熟悉的组织机构就是典型的层次结构。

②网状模型：以网状结构表示实体与实体之间的联系。网状模型是层次模型的扩展，允许结点有多于一个父结点，并可以有一个以上的结点没有父结点。

③关系模型：是用二维表结构来表示实体及实体间联系的模型，并以二维表格的形式组织数据库中的数据。它具有下列优点：

A.关系模型是建立在严格的数学概念的基础上的。

B.关系模型的概念单一，统一用关系来表示实体以及实体之间的联系，对数据的检索和更新结果同样也是用关系（即表）来表示。因而，关系模型的数据结构简单、清晰，用户易懂、易用。

C.关系模型的存取路径对用户透明，从而具有更高的数据独立性、更好的安全保密性，也简化了程序员的工作和数据库开发建立的工作。

④面向对象模型：面向对象方法与数据库相结合所构成的数据模型称为面向对象模型。面向对象模型既是概念模型又是逻辑模型。面向对象数据模型用面向对象观点来描述现实世界实体的逻辑组织、对象间的联系，其表达能力丰富，具有对象可复用、维护方便等优点。

（3）物理层数据模型

物理层数据模型，也称为数据的物理模型，其描述数据在存储介质上的组织结构，是逻辑模型的物理实现，即每一种逻辑模型在实现时都有与其相对应的物理模型。物理模型是数据库最底层的抽象，它确定数据的物理存储结构、数据存取路径以及调整、优化数据库的性能。物理模型的设计目标是提高数据库性能和有效利用存储空间。

第二章　关系数据库

考点 1：关系数据库概述

关系数据库是目前应用最广泛的数据库，它以关系模型作为数据的逻辑模型，采用关系作为数据的组织方式，其数据库操作建立在关系代数的基础上，具有坚实的数学基础。

关系数据库的基本特征是使用关系数据模型组织数据，这种思想源于数学。

1962 年 CODASYL 发表的“信息代数”就是最早将数学方法用于数据处理的。1970 年 IBM 公司的 E.F.Codd 在美国计算机学会会刊《Communications of the ACM》系统、严格地提出了关系模型，由此开创了数据库系统的新纪元。

20 世纪 70 年代末，关系方法的理论研究和软件系统的研制才取得了重大突破，其中最具代表性的是 IBM 公司的 SanJose 研究中心成功地在 IBM370 系列计算机上研制出关系数据库实验系统 SystemR，并于 1981 年宣布具有 SystemR 全部特征的数据库管理系统 SQL/DS 问世。

进入 20 世纪 80 年代后，在商用数据库管理系统中，关系模型逐渐取代早期的网状模型和层次模型，成为主流数据模型。

考点 2：关系数据模型的组成要素

关系数据库系统是支持关系模型的数据库系统。作为一种数据模型，关系模型同样包含三个组成要素，分别是关系数据结构、关系操作集合和关系完整性约束。

考点 3：关系数据结构

（1）表：也称为关系，是一个二维的数据结构，它由表名、构成表的各个列及若干行数据组成。每个表有一个唯一的表名，表中每一行数据描述一条具体的记录值。

（2）关系：一个关系逻辑上对应一张二维表，可以为每个关系取一个名称进行标识。关系可以有三种类型，即基本关系、查询表和视图表。其中，基本关系通常又称为基本表或基表，是实际存在的表，它是实际存储数据的逻辑表示；查询表是查询结果对应的表；视图表是由基本表或其他视图表导出的表，是虚表，不对应实际存储的数据。

（3）列：也称作字段或属性。表中每一列有一个名称，称为列名、字段名或属性名。每一列表示实体的一个属性，具有相同的数据类型。在一个数据库中，表名必须唯一；在表中，字段名必须唯一，不同表中可以出现相同的字段名；表和字段的命名应尽量有意义，并尽量简单。

（4）属性：表中的一列即为一个属性，给每一个属性起一个名称即属性名。与之同义的术语是“列”。表中属性的个数称为关系的元或度。列的值称为属性值；属性值的取值范围称为值域。

（5）行：表中的行也称作元组或记录。

（6）元组：表中的一行即为一个元组。

（7）分量：元组中的一个属性值，称为分量。

（8）码或键：如果在一个关系中，存在这样的属性（或属性组），使得在该关系的任何一个关系状态中的两个元组，在该属性（或属性组）上值的组合都不相同，即这些属性（或属性组）的值都能用来唯一标识该关系的元组，则称这些属性（或属性组）为该关系的码或键。

（9）超码或超键：如果在关系的一个码中移去某个属性，它仍然是这个关系的码，则称这样的码或键为该关系的超码或超键。

（10）候选码或候选键：如果在关系的一个码或键中，不能从中移去任何一个属性，否则它就不是这个关系的码或键，则称这样的码或键为该关系的候选码或候选键。

（11）主码或主键：在一个关系的若干个候选码或候选键中指定一个用来唯一标识关系的元组，则称这个被指定的候选码或候选键为该关系的主码或主键。

（12）全码或全键：一个关系模式的所有属性集合是这个关系的主码或主键，则称这样的主码或主键为全码或全键。

（13）主属性和非主属性：关系中包含在任何一个候选码中的属性称为主属性或码属性，不包含在任何一个候选码中的属性称为非主属性或非码属性。

（14）外码或外键：当关系中的某个属性（或属性组）不是这个关系的主码或候选码，而是另一关系的主码时，称该属性（或属性组）为这个关系的外码或外键。

（15）参照关系和被参照关系：参照关系也称为从关系，被参照关系也称为主关系，它们是指以外码相关联的两个关系。以外码作为主码的关系称为被参照关系；外码所在的关系称为参照关系。

（16）域：域表示属性的取值范围。

（17）数据类型：表中每个列都有相应的数据类型，它用于限制（或容许）该列中存储的数据。每个字段表示同一类信息，具有相同的数据类型。

（18）关系模式：在关系数据库中，关系模式是型，关系是值，即关系模式是对关系的描述。

（19）关系数据库：是以关系模型作为数据的逻辑模型，并采用关系作为数据组织方式的一类数据库，其数据库操作建立在关系代数的基础上。关系数据库对关系是有限定的，具体要求如下：

①每一个属性都是不可分解的。这是关系数据库对关系的最基本的一个限定，要求关系的每一个分量必须是一个不可分的数据项，也就是说，不允许表中有表。

②每一个关系仅仅有一种关系模式，即每一个关系模式中的属性的数据类型以及属性的个数是相对固定的。

③每一个关系模式中的属性必须命名，在同一个关系模式中，属性名必须是不同的。

④同一个关系中不允许出现候选码或候选键值完全相同的元组。

⑤在关系中元组的顺序（即行序）是无关紧要的，可以任意交换。

⑥在关系中属性的顺序（即列序）是无关紧要的，可以任意交换。

考点 4：关系操作集合

（1）基本的关系操作：关系模型中常用的关系操作包括查询操作和插入、删除、修改操作两大部分。

（2）关系数据语言的分类：分为关系代数语言、关系演算语言以及兼具两者双重特点的语言（例如

SQL）。它们的共同特点是：语言具有完备的表达能力，是非过程化的集合操作语言，功能强，能够独立使用也可以嵌入高级语言中使用。

（3）关系代数：是关系操作语言的一种传统表示方式，它是以集合代数为基础发展起来的。关系代数操作经过有限次复合的式子称为关系代数操作表达式，简称为关系代数表达式。按照运算符的不同，关系代数的操作可分为传统的集合运算与专门的关系运算。

①传统集合运算是二目运算，它将关系看成元组的集合，其运算是从关系的“水平”方向，即行的角度来进行，具体有并、差、交、笛卡尔积 4 种运算。

②专门的关系运算不仅涉及行，而且涉及列，它可分为一元专门关系操作和二元专门关系操作。其中，一元专门关系操作包括对单个关系进行垂直分解的投影运算和进行水平分解选择运算；二元专门关系操作则是对两个关系进行操作，包括连接运算和除运算。

考点 5：关系的完整性约束

数据库的数据完整性是指数据库中数据的正确性、相容性和一致性。关系模型中有三类完整性约束，分别是实体完整性约束、参照完整性约束和用户定义完整性约束。其中，实体完整性和参照完整性是关系模型必须满足的完整性约束条件，被称作是关系的两个不变性，应该由关系数据库管理系统（RDBMS）自动支持；而用户定义完整性约束包括域完整性约束和其他约束，大多是指应用领域需要遵循的对属性值域约束条件和业务规则，体现了具体应用领域中的语义约束。

（1）实体完整性约束：是指关系的主属性，即主码的组成不能为空，也就是关系的主属性不能是空值。

（2）参照完整性约束：就是定义外码和主码之间的引用规则，它是对关系间引用数据的一种限制。这里描述参照完整性的定义：若属性（或属性组）F 是基本关系 R 的外码，它与基本关系 S 的主码 K 相对应，则对于 R 中每个元组在 F 上的值只允许两种可能，即要么取空值（F 的每个属性值均为空值），要么等于 S 中某个元组的主码值。

（3）用户定义的完整性约束：是针对某一应用环境的完整性约束条件，它反映了某一具体应用所涉及的数据应满足的要求。关系模型提供定义和检验这类完整性规则的机制，其目的是用统一的方式由系统来处理它们，不再由应用程序来完成这项工作。在实际系统中，这类完整性规则一般在建立数据库表的同时进行定义，但如果某些约束条件没有建立在库表一级，则应用编程人员应在各模块的具体编程中通过程序进行检查和控制。

（4）关系模型完整性约束的检验：①执行插入操作；②执行删除操作；③执行更新操作。

考点 6：关系模式中可能存在的冗余和异常问题

（1）数据冗余：是指同一数据被反复存储的情况。

（2）更新异常：数据冗余将导致存储空间的浪费和潜在数据不一致性及修改麻烦等问题。

（3）插入异常：是指应该插入到数据库中的数据不能执行插入操作的情形。

（4）删除异常：是指不应该删去的数据被删去的情形。

现在人们已经提出了许多种类型的数据依赖，其中最重要的是函数依赖和多值依赖。关系模式产生上

述问题的原因，以及消除这些问题的方法，都与数据依赖的概念密切相关。

考点 7：函数依赖与关键字

函数依赖是指关系中属性间的对应关系。设 R 为任一给定关系，如果对于 R 中属性 X 的每一个值，R 中的属性 Y 只有唯一值与之对应，则称 X 函数决定 Y 或称 Y 函数依赖于 X，记作 X→Y。其中，X 称为决定因素。函数依赖根据其不同性质可分为完全函数依赖、部分函数依赖和传递函数依赖。

（1）完全函数依赖：设 R 为任一给定关系，X、Y 为其属性集，若 X→Y，且对 X 中的任何真子集 X' 都有 X'→Y，则称 Y 完全函数依赖于 X。

（2）部分函数依赖：设 R 为任一给定关系，X、Y 为其属性集，若 X→Y，且 X 中存在一个真子集 X' 满足 X'→Y，则称 Y 部分函数依赖于 X。

（3）传递函数依赖：设 R 为任一给定关系，X、Y、Z 为其不同属性子集，若 X→Y，Y→X，Y→Z，则有 X→Z，称为 Z 传递函数依赖于 X。

在函数依赖的概念的基础上，可以给出更为严格的关键字的定义：设 R 为任一给定关系，U 为其所含的全部属性集合，X 为 U 的子集，若有完全函数依赖 X→U，则 X 为 R 的一个候选关键字。

作为候选关键字的属性集 X 唯一标识 R 中的元组，但该属性集的任何真子集不能唯一标识 R 中的元组。显然，一个关系 R 中可能存在多个候选关键字，通常选择其中之一作为主关键字，即主键。候选关键字中所含有的属性称为主属性，不包含在候选关键字中的属性称为非主属性。

考点 8：范式与关系规范化过程

关系数据库中的关系需要满足一定的要求，不同程度的要求称为不同的范式。满足最低要求的称为第一范式，称简 1NF，这是最基本的范式；在第一范式的基础上进一步满足一些新要求的称为第二范式（2NF）；以此类推，再进一步的范式是第三范式（3NF）及其改进形式 BCNF；当然，还有更进一步的高级范式，如第四范式（4NF）、第五范式（5NF）等。

一个低一级范式的关系模式通过模式分解可以转换为若干个高一级范式的关系模式的集合，这种过程就叫规范化。

（1）第一范式：设 R 为任一给定关系，如果 R 中每个列与行的交点处的取值都是不可再分的基本元素，则 R 为第一范式。第一范式是一个不含重复组的关系，其中不存在嵌套结构。不满足第一范式的关系为非规范关系。

（2）第二范式：设 R 为任一给定关系，若 R 为 1NF，且其所有非主属性都完全函数依赖于候选关键字，则 R 为第二范式。

（3）第三范式：设 R 为任一给定关系，若 R 为 2NF，且其每一个非主属性都不传递函数依赖于候选关键字，则 R 为第三范式。

（4）BCNF：设 R 为任一给定关系，X、Y 为其属性集，F 为其函数依赖集，若 R 为 3NF，且其 F 中所有函数依赖 X→Y（Y 不属于 X）中的 X 必包含候选关键字，则 R 为 BCNF。

考点 9：关系规范化理论的应用

关系规范化理论主要应用于数据库设计中的概念设计阶段，对所产生的概念设计，可用它来分析其实体划分是否适合，判断属性分配到哪个实体中更为合理。在实现设计中当将 E-R 图向关系模型转换时，还可以用它来分析并发现概念设计中可能存在的遗漏或不当之处，特别是联系实体是否不单独转换为一独立关系而集成到与之相联的基本实体中去处理时，规范化理论是最有效的评价准则。

第三章　数据库设计

考点 1：数据库设计概述

（1）数据库的生命周期：可分为两个阶段，分别是数据库分析与设计阶段、数据库实现与操作阶段。其中，数据库分析与设计阶段包括需求分析、概念设计、逻辑设计和物理设计四个环节；数据库实现与操作阶段包含数据库的实现、操作与监督、修改与调整三个子阶段。

（2）数据库设计的目标：满足应用功能需求和良好的数据库性能。

（3）数据库设计的内容

数据库设计是从用户对数据的需求出发，研究并构造数据库的过程，其包含两个方面的内容：

①数据库结构设计：是针对给定的应用环境进行数据库的模式或子模式的设计，包括数据库的概念结构设计、逻辑结构设计和物理结构设计。（静态）

②数据库行为设计：是确定数据库用户的行为和动作，而用户的行为和动作是对数据库的操作，它们通常是通过应用程序来实现的。（动态）

（4）数据库设计的方法

①直观设计法：是一类最原始的数据库设计方法，它利用设计者的经验和技巧来设计数据库模式。

②规范设计法：是一类较为普遍、常用的数据库设计方法。其中，常见有：新奥尔良设计方法、基于 E-R 模型的数据库设计方法、基于第三范式的设计方法。

③计算机辅助设计法：是指在数据库设计过程中，以领域专家的知识或经验为主导，模拟某一规范化设计的方法，通常通过人机交互的方式来完成设计的某些过程。目前，许多计算机辅助软件工程工具（俗称 CASE 工具），可以用来帮助数据库设计人员完成数据库设计的一些工作，如此可减轻数据库设计人员的工作量，加快数据库设计的进度。

（5）数据库设计的过程：将数据库设计分为：需求分析阶段；结构设计阶段，其包括概念结构设计、逻辑结构设计和物理结构设计；行为设计阶段，其包括功能设计、事务设计和程序设计；数据库实施阶段，其包括加载数据库数据和调试运行应用程序；数据库运行和维护阶段。

考点 2：数据库设计的基本步骤

（1）需求分析：需求分析是数据库设计的起点。需求分析一般可分为四个步骤，即确定数据库范围、分析数据应用过程、收集与分析数据和编写需求分析报告。

（2）概念结构设计：概念结构设计的任务是在需求分析中产生的需求分析报告的基础上，按照特定的方法设计满足应用需求的用户信息结构，该信息结构通常称为概念模型。概念结构设计的常用方法有实体分析法和属性综合法两种，它们也分别称为自顶向下法和自底向上法。

（3）逻辑结构设计：目标是将概念模型转换为等价的、并为特定 DBMS 所支持数据模型的结构。数据库逻辑模型一般由层次、网状、关系数据模型表示。

（4）物理设计：是指对于一个给定的数据库逻辑结构，研究并构造物理结构的过程，其具体任务主要是确定数据库在存储设备上的存储结构及存取方法，因 DBMS 的不同还可能包括建立索引和聚集，以及物理块大小、缓冲区个数和大小、数据压缩的选择等。

（5）数据库实施：数据库实施阶段需要完成的工作包括：加载数据、应用程序设计和数据库试运行。

（6）数据库运行和维护：只有经过试运行之后，确认系统无故障或暂未发现故障时，系统才能投入到生产实际中运行。数据库系统投入实际运行标志着数据库设计和应用开发的基本完成，但绝不意味着设计和应用开发工作的终止。系统维护中最困难的工作是数据库重组与重构。

考点 3：关系数据库设计过程与各级模式

按照数据库设计的基本步骤，在关系数据库设计的不同阶段，会形成数据库的各级模式，即：

（1）在需求分析阶段，综合各个用户的应用需求；

（2）在概念结构设计阶段，形成独立于机器特点、独立于各个关系数据库管理系统产品的概念模式；

（3）在逻辑结构设计阶段，将 E-R 图转换成具体的数据库产品支持的关系数据模型，形成数据库逻辑模式，然后根据用户处理的要求、安全性的考虑，在基本表的基础上再建立必要的视图，形成数据的外模式；

（4）在物理结构的设计阶段，根据关系数据库管理系统的特点和处理的需要，进行物理存储安排，建立索引，形成数据库内模式。

考点 4：概念结构设计方法

（1）E-R 图的表示方法

概念结构设计就是将需求分析得到的用户需求抽象为信息结构（即概念模型）的过程，它通常使用 E-R 图来作为描述现实世界的建模工具。E-R 图提供了表示信息世界中实体、属性和联系的方法，具体如下：

①实体型，其用矩形表示，矩形框内写明实体的名称。

②属性，其用椭圆形表示，并用无向边将其与相应的实体连接起来。

③联系，其用菱形表示，菱形框内写明联系的名称，并用无向边分别与有关实体连接起来，同时在无向边旁标上联系的类型（1：1、1：N 或 M：N），如果一个联系具有属性，则这些属性也要用无向边与该联系连接起来。

（2）局部信息结构设计

根据需求分析报告中标明的不同用户视图范围所建立的满足该范围内用户需求的信息结构，称为局部信息结构。局部信息结构设计的步骤分为：确定局部范围；选择实体；选择实体关键字；确定实体间联系；确定实体的属性。

（3）全局信息结构设计

全局信息结构设计是将上述步骤中产生的所有局部信息结构合并成为一个全局信息结构。全局信息结构必须是所有局部信息结构的全面准确的映像，即合并前各局部信息结构所能实现的应用需求，合并形成的全局信息结构仍能实现。同时，全局信息结构中应努力避免或消除不同局部信息结构中因用户观点的不同所可能导致的数据不一致，并尽可能地增强数据的共享性，控制数据的冗余。合并的过程是一个不断发

现和解决冲突的过程。各局部 E-R 图之间的冲突主要表现在三个方面：属性冲突、命名冲突和结构冲突。

①属性冲突主要包含：A.属性域冲突（即属性值的类型、取值范围、取值集合的不同）。B.属性取值单位冲突。

②命名冲突主要包含：A.同名异义（即不同意义的实体类型名或联系类型名在不同的局部应用中具有相同的名字）。B.异名同义（即同一意义的实体类型名或联系类型名在不同的局部应用中具有不同的名字）。

③结构冲突主要包含：

A.同一对象在一个局部 E-R 图中作为实体，而在另一个局部 E-R 图中作为属性。

B.同一实体在不同的 E-R 图中属性个数和类型不同。

C.实体之间的联系在不同的 E-R 图中是不同的类型。

对于属性冲突和命名冲突，通常采用讨论、协商等行政手段解决，而对于结构冲突，则需要通过认真分析后，采用技术手段加以解决。

考点 5：逻辑结构设计方法

在关系数据库设计中，逻辑结构设计的任务就是把概念结构设计阶段已设计好的 E-R 图转换为关系数据库管理系统所支持的关系模型，其通常包括这样三项工作：将 E-R 图转换为关系模型、对关系数据模型进行优化、设计面向用户的外模式。

（1）E-R 图向关系模型的转换

将 E-R 图转换为关系模型实际上就是要将实体、实体的属性和实体间的联系转换为某种关系模式。这种转换一般遵守以下原则：

①一个实体型转换为一个关系模式。

②一个一对一（1：1）联系可以转换为一个独立的关系模式，也可以与任意一端对应的关系模式合并。

③一个一对多（1：N）联系可以转换为一个独立的关系模式，也可以与 N 端对应的关系模式合并。

④　一个多对多（M：N）联系转换为一个关系模式。

⑤三个或三个以上实体间的一个多元联系可以转换为一个关系模式。

⑥具有相同码的关系模式可合并。

（2）数据模型的优化

关系数据模型的优化通常以关系规范化理论为指导，其方法如下：

①确定各属性间的函数依赖关系。

②对于各个关系模式之间的数据依赖进行极小化处理，消除冗余的联系。

③判断每个关系模式的范式，根据实际需要确定最合适的范式。

④按照需求分析阶段得到的处理要求，分析这些模式对于这样的应用环境是否合适，确定是否要对某些模式进行合并或分解。

⑤对关系模式进行必要的分解，提高数据操作的效率和存储空间的利用率。

（3）设计用户子模式

定义数据库全局模式主要是从系统的时间效率、空间效率、易维护等角度出发。由于用户外模式与模

式是相对独立的，因此在定义用户外模式时可以注重考虑用户的习惯与方便。具体包括以下几个方面：

①可以通过视图机制在设计用户视图时，重新定义某些属性的别名，使其更符合用户的习惯，以方便使用。

②可以对不同级别的用户定义不同的视图，以保证系统的安全性。

③简化用户对系统的使用。

考点 6：物理设计方法

物理设计的任务主要是通过对关系建立索引和聚集来实现与应用相关数据的逻辑连接和物理聚集，以改善对数据库的存取效率。

（1）建立索引：索引的建立是通过 DBMS 提供的有关命令来实现的。建立索引的方式有静态和动态两种。

（2）建立聚集：聚集是将相关数据集中存放的物理存储技术，借以提高 I/O 的数据命中率而改善存取速度，其功能由具体的 DBMS 所提供。数据聚集结构的一种有效方式是块结构方式，块与块之间由指针连接，一个块对应于一个物理分区。数据聚集可在一个或多个关系上建立。

第四章　SQL 与关系数据库基本操作

考点 1：SQL 的发展

SQL 是于 1974 年由 Boyce 和 Chamberlin 提出的，并在 IBM 公司研制的关系数据库管理系统原型 SystemR 上实现。从 20 世纪 80 年代以来，SQL 一直是关系数据库管理系统（RDBMS）的标准语言。最早的 SQL 标准是 1986 年 10 月由美国国家标准局颁布的。随后，国际化标准组织于 1987 年 6 月也正式采纳它为国际标准，并在此基础上进行了补充，且于 1989 年 4 月 ISO 提出了具有完整性特征的 SQL，称之为 SQL-89。而后，在 SQL-89 的基础上，SQL 标准得到不断地丰富与修订，例如 ISO 和 ANSI 于 1992 年共同颁布的标准 SQL-92（或称为 SQL2），以及 1999 年颁布的 SQL-99（或称为 SQL3）。直至今日，SQL 成为了一个通用的、功能极强的关系数据库语言。

当然，目前没有一个数据库系统能够支持 SQL 标准的全部概念和特性。各个关系数据库管理系统产品在实现标准 SQL 时各有差别，与 SQL 标准的符合程度也不相同，但它们仍然遵循 SQL 标准，并以 SQL 标准为主体进行相应的扩展，提供一些执行特定操作的额外功能或简化方法。

考点 2：SQL 的特点

（1）SQL 不是某个特定数据库供应商专有的语言。

（2）SQL 简单易学。它的语句全都是由具有很强描述性的英语单词所组成，而且这些单词的数目不多。

（3）SQL 尽管看上去很简单，但它实际上是一种强有力的语言，灵活使用其语言元素，可以进行非常复杂和高级的数据库操作。

考点 3：SQL 的组成

SQL 集数据查询、数据定义、数据操纵和数据控制四大功能于一体，其核心主要包含有以下几个部分：

（1）数据定义语言

数据定义语言主要用于对数据库及数据库中的各种对象进行创建、删除、修改等操作。数据定义语言包括的主要 SQL 语句有以下三个：

①CREATE：用于创建数据库或数据库对象。

②ALTER：用于对数据库或数据库对象进行修改。

③DROP：用于删除数据库或数据库对象。

（2）数据操纵语言

数据操纵语言主要用于操纵数据库中各种对象，特别是检索和修改数据。数据操纵语言包括的主要 SQL 语句如下：

①SELECT：用于从表或视图中检索数据，其是数据库中使用最为频繁的 SQL 语句之一。

②INSERT：用于将数据插入到表或视图中。

③UPDATE：用于修改表或视图中的数据，其既可修改表或视图中一行数据，也可同时修改多行或全部数据。

④DELETE：用于从表或视图中删除数据，其中可根据条件删除指定的数据。

（3）数据控制语言

数据控制语言主要用于安全管理。数据控制语言包括的主要 SQL 语句如下：

①GRANT：用于授予权限，可把语句许可或对象许可的权限授予其他用户和角色。

②REVOKE：用于收回权限，其功能与 GRANT 相反，但不影响该用户或角色从其他角色中作为成员继承许可权限。

（4）嵌入式和动态 SQL 规则

嵌入式和动态 SQL 规则规定了 SQL 语句在高级程序设计语言中使用的规范方法，以便适应较为复杂的应用。

（5）SQL 调用和会话规则

SQL 调用包括 SQL 例程和调用规则，以便提高 SQL 的灵活性、有效性、共享性以及使 SQL 具有更多的高级语言的特征。SQL 会话规则则可使应用程序连接到多个 SQL 服务器中的某一个，并与之交互。

考点 4：MySQL 使用基础

目前，使用 MySQL 数据库管理系统构建各种信息管理系统或互联网网站的应用环境，主要有如下两种构架方式：

（1）LAMP：即使用 Linux 作为操作系统，Apache 作为 Web 服务器，MySQL 作为数据库管理系统，PHP、Perl 或 Python 语言作为服务器端脚本解释器。

（2）WAMP：即使用 Windows 作为操作系统，Apache 作为 Web 服务器，MySQL 作为数据库管理系统，PHP、Perl 或 Python 语言作为服务器端脚本解释器。

考点 5：MySQL 中的 SQL

MySQL 作为一种关系型数据库管理系统，遵循 SQL 标准，提供了对数据定义语言 DDL、数据操纵语言 DML、数据控制语言 DCL 的支持，并且同样支持关系数据库的三级模式结构。其外模式包括视图和部分基本表，数据库模式包括若干基本表，内模式则包括若干存储文件。基本表是本身独立存在的表，在 MySQL 中一个关系对应一个基本表，一个或多个基本表对应一个存储文件，一个表可以有若干索引，索引也存放在存储文件中，其中存储文件的逻辑结构组成了 MySQL 的内模式，并且存储文件的物理结构对最终用户是隐蔽的；而视图则是从一个或几个基本表导出的表，尽管它也是关系，但其本身不独立存储在数据库中，即数据库中只存储视图的定义而不存储视图对应的数据，这些数据仍存储在导出视图的基本表中，因此视图是一个虚表，并且用户可以在视图上再定义视图。

此外，为方便用户编程，MySQL 在 SQL 标准的基础上增加了部分扩展的语言要素。这些语言要素包括常量、变量、运算符、表达式、函数、流程控制语句和注解等。

（1）常量：是指在程序运行过程中值不变的量，也称为字面值或标量值。常量的使用格式取决于值的数据类型，可分为字符串常量、数值常量、十六进制常量、时间日期常量、位字段值、布尔值和 NULL 值。

（2）变量：用于临时存储数据，变量中的数据可以随着程序的运行而变化。变量有名字和数据类型两个属性。在 MySQL 中，变量分为用户变量和系统变量。

（3）运算符：MySQL 提供了如下几类编程语言中常用的运算符：①算术运算符；②位运算符；③比较运算符；④逻辑运算符。

（4）表达式：是常量、变量、列名、复杂计算、运算符和函数的组合。根据表达式的值的数据类型，表达式可分为字符型表达式、数值型表达式和日期表达式。

（5）内置函数：在编写 MySQL 数据库程序时，通常可直接调用系统提供的内置函数，来对据库表进行相关操作。MySQL 中包含了 100 多个函数，基本分为如下几类：数学函数、聚合函数、字符串函数、日期和时间函数、加密函数、控制流程函数、格式化函数、类型转换函数、系统信息函数。

考点 6：数据库模式定义

数据库模式的定义包含数据库的创建、选择、修改、删除、查看等操作。

（1）创建数据库：在 MySQL 中，可以使用 CREATE DATABASE 或 CREATE SCHEMA 语句创建数据库。

（2）选择数据库：在 MySQL 中，使用 USE 语句可以实现从一个数据库“跳转”到另一个数据库，在用 CREATE DATABASE 语句创建了数据库之后，该数据库不会自动成为当前数据库，需要用这条 USE 语句来指定。

（3）修改数据库：在 MySQL 中，可以使用 ALTER DATABASE 或 ALTER SCHEMA 语句，来修改已被创建的数据库的相关参数。

（4）删除数据库：在 MySQL 中，当需要删除已创建的数据库时，可使用 DROP DATABASE 或 DROPSCHEMA 语句。

（5）查看数据库：在 MySQL 中，可使用 SHOW DATABASES 或 SHOW SCHEMAS 语句查看可用数据库列表。

考点 7：表定义

数据表是关系数据库中最重要、最基本的数据对象，也是数据存储的基本单位。

数据表被定义为字段的集合，数据在表中是按照行和列的格式来存储的，每一行代表一条记录，每一列代表记录中一个字段的取值。

（1）创建表：在 MySQL 中，可以使用 CREATE TABLE 语句创建表。CREATE TABLE 语句的语法内容较多，主要由表创建定义、表选项和分区选项等内容所构成。

（2）更新表：为实现数据库中表规范化设计的目的，有时需要对之前已经创建的表做进一步的结构修改与调整。

（3）重命名表：除了使用前面的 ALTERTABLE 语句，还可以直接用 RENAMETABLE 语句来更改表的名字，并可同时重命名多个表。

（4）删除表：如若需要删除数据库中一个已存在的表，可以通过使用 DROPTABLE 语句来实现。

（5）查看表：在数据库中，查看表包括显示表的名称和显示表的结构两种情形。

考点 8：索引定义

所谓索引, 就是 DBMS 根据表中的一列或若干列按照一定顺序建立的列值与记录行之间的对应关系表，

因而索引实质上是一张描述索引列的列值与原表中记录行之间一一对应关系的有序表。索引是提高数据文件访问效率的有效方法。尽管索引可以大大加快数据查询的速度，但过多地使用索引也会增加系统的开销，这是因为索引存在着一些弊端：

（1）索引是以文件的形式存储的，DBMS 会将一个表的所有索引保存在同一个索引文件中，索引文件需要占用磁盘空间。如果有大量的索引，索引文件可能会比数据文件更快地达到最大的文件尺寸。特别是如果在一个大表上创建了多种组合索引，索引文件会膨胀得非常快。

（2）索引在提高查询速度的同时，却会降低更新表的速度。

考点 9：索引的类型

根据具体用途，索引在逻辑上通常包含有如下几类：

（1）普通索引：这是最基本的索引类型，它没有任何限制。创建普通索引时，通常使用的关键字是 INDEX 或 KEY。

（2）唯一性索引：这类索引和普通索引基本相同，只是有一点区别，即索引列中的所有值都只能出现一次，必须是唯一的。创建唯一性索引时，通常使用的关键字 UNIQUE。

（3）主键：是一种唯一性索引。创建主键时，必须指定关键字 PRIMARYKEY，且不能有空值。主键一般是在创建表的时候指定，也可以通过修改表的方式添加主键，并且每个表只能有一个主键。

考点 10：索引的创建、查看与删除

（1）索引的创建：①使用 CREATEINDEX 语句创建索引；②使用 CREATETABLE 语句创建索引；③使用 ALTERTABLE 语句创建索引。

（2）索引的查看：如果需要查看表中创建的索引的情况，可以使用 SHOWINDEX 语句。

（3）索引的删除：当一个索引不再需要时，可以使用 DROPINDEX 语句或 ALTERTABLE 语句来进行删除。

考点 11：数据更新

数据更新操作有三种：向表中添加若干行数据、修改表中的数据和删除表中的若干行数据。在 SQL 中有三类相应的语句，分别是插入数据、修改数据和删除数据。

（1）插入数据：在 MySQL 中，可以使用 NSERT 语句，向数据库中一个已有的表插入一行或多行元组数据。NSERT 语句有三种语法形式，分别对应的是 NSERT…VALUES 语句、INSERT…SET 语句和 NSERT…SELECT 语句。

（2）删除数据：在 MySQL 中，可以使用 DELETE 语句删除表中的一行或多行数据。

（3）修改数据：在 MySQL 中，可以使用 UPDATE 语句来修改更新一个表中的数据，实现对表中行的列数据进行修改。

考点 12：数据查询

数据查询是 SQL 语言的核心功能，也是数据库中使用得最多的操作，其用途是从数据库的一张或多张表（或视图）中检索出满足条件的数据信息。通常，查询的结果是由 0 行（没有满足条件的数据）或多行记录组成的一个记录集合，并允许选择一个或多个字段作为输出字段。SQL 是提供 SELECT 语句进行数据

查询，该功能强大、使用灵活，其数学理论基础是关系数据模型中对表对象的一组关系运算，即选择、投影和连接。

（1）SELECT 语句：使用 SELECT 语句可以在需要时从数据库中快捷方便地检索、统计或输出数据。该语句的执行过程是从数据库中选取匹配的特定行和列，并将这些数据组织成一个结果集，然后以一张临时表的形式返回。

（2）列的选择与指定：在 SELECT 语句中，语法项“select_expr”主要用于指定需要查询的内容，其指定方法通常有以下几种：①选择指定的列；②定义并使用列的别名；③替换查询结果集中的数据；④计算列值；⑤聚合函数。

（3）FROM 子句与多表连接查询：若一个查询同时涉及两个或两个以上的表，则称之为多表连接查询，也称多表查询或连接查询。多表连接查询是关系数据库中最主要的查询。通过在 FROM 子句中指定多个表时，SELECT 操作会使用“连接”运算将不同表中需要查询的数据组合到一个结果集中，并同样以一个临时表的形式返回，其连接方式主要包括交叉连接、内连接和外连接。

①交叉连接：又称笛卡尔积。在 MySQL 中，它是通过在 FROM 子句中使用关键字“CROSSJOIN”来连接两张表，从而实现一张表的每一行与另一张表的每一行的笛卡尔乘积，并返回两张表的每一行相乘的所有可能的搭配结果。

②内连接：是一种最常用的连接类型，它是通过在查询中设置连接条件的方式，来移除查询结果集中某些数据行之后的交叉连接。关于内连接的使用，通常有如下三种情形：等值连接、非等值连接、自连接。

③外连接：是首先将连接的两张表分为基表和参考表，然后再以基表为依据返回满足和不满足条件的记录。根据连接表的顺序，可分为左外连接和右外连接两种。

（4）WHERE 子句与条件查询：WHERE 子句中设置过滤条件的几个常用方法：

①比较运算：用于比较两个表达式的值。

②判定范围：在 WHERE 子句中，用于范围判定的关键字有“BETWEEN”和“IN”两个。

③判定空值：当需要判定一个表达式的值是否为空值时，可以使用关键字“ISNULL”来实现。

④子查询：通常，可以使用 SELECT 语句创建子查询，即可嵌套在其他 SELECT 查询中的 SELECT 查询。在 MySQL 中，区分如下四类子查询：表子查询、行子查询、列子查询、标量子查询。

A.结合关键字“IN”使用的子查询：主要用于判定一个给定值是否存在于在子查询的结果集中。

B.结合比较运算符使用的子查询：主要用于将表达式的值与子查询的结果进行比较运算。

C.结合关键字“EXIST”使用的子查询：主要用于判定子查询的结果集是否为空。

（5）GROUPBY 子句与分组数据：ROUPBY 子句可指示 DBMS 分组数据，然后对每个组而不是对整个结果集进行聚合。

（6）HAVING 子句：在 SELECT 语句中，除了能使用 GROUPBY 子句分组数据之外，还可以使用 HAVING 子句来过滤分组，即在结果集中规定包含哪些分组和排除哪些分组。

（7）ORDERBY 子句：在 SELECT 语句中，可以使用 ORDERBY 子句将结果集中的数据行按一定的

顺序进行排列，否则结果集中数据行的顺序是不可预料的。

（8）LIMIT 子句：当使用 SELECT 语句返回的结果集中行数很多时，为了便于用户对结果数据的浏览和操作，可以使用 LIMIT 子句来限制被 SELECT 语句返回的行数。

考点 13：视图与基本表的异同

视图是数据库中的一个对象，它是数据库管理系统提供给用户的以多种角度观察数据库中数据的一种重要机制。视图是从一个或多个表或者其他视图中通过查询语句导出的表，它也包含一系列带有名称的数据列和若干条数据行，并有自己的视图名，由此可见视图与基本表十分类似。然而，视图仍不同于数据库中真实存在的基本表，它们存在以下区别：

（1）视图不是数据库中真实的表，而是一张虚拟表，其结构和数据是建立在对数据库中真实表的查询基础上的。

（2）视图的内容是由存储在数据库中进行查询操作的 SQL 语句来定义的，它的列数据与行数据均来自于定义视图的查询所引用的真实表，并且这些数据是在引用视图时动态生成的。

（3）视图不是以数据集的形式存储在数据库中，它所对应的数据实际上是存储在视图所引用的真实表（基本表）中。

（4）视图是用来查看存储在别处的数据的一种虚拟表，而其自身并不存储数据。

（5）集中分散数据。（6）简化查询语句。（7）重用 SQL 语句。（8）保护数据安全。

（9）共享所需数据。（10）更改数据格式。

考点 14：视图

（1）创建视图：在 MySQL 中，可以使用 CREATEVIEW 语句来创建视图。

（2）删除视图：在 MySQL 中，可以使用 DROPVIEW 语句来删除视图。使用 DROPVIEW 语句可以一次删除多个视图，但必须在每个视图上拥有 DROP 权限。同时，为防止因删除不存在的视图而出错，需要在 DROPVIEW 语句中添加关键字“IF EXISTS”。

（3）修改视图定义：在 MySQL 中，可以使用 ALTERVIEW 语句来对已有视图的定义（结构）进行修改。

（4）查看视图定义：在 MySQL 中，可以使用 SHOWCREATEVIEW 语句来查看已有视图的定义（结构）。

（5）更新视图数据：对于可更新的视图，需要该视图中的行和基本表中的行之间具有一对一的关系：

①使用 INSERT 语句通过视图向基本表插入数据。

②使用 UPDATE 语句通过视图修改基本表的数据。

③使用 DELETE 语句通过视图删除基本表的数据。

（6）查询视图数据：视图一经定义后，就可以如同查询数据库中的真实表一样，对视图进行数据查询检索，这也是对视图使用最多的一种操作。视图用于查询检索，主要体现在这样一些应用：利用视图简化复杂的表连接；使用视图重新格式化检索出的数据；使用视图过滤不想要的数据。

第五章　数据库编程

考点 1：存储过程的基本概念

存储过程是一组为了完成某项特定功能的 SQL 语句集，其实质上就是一段存储在数据库中的代码，它可以由声明式的 SQL 语句和过程式 SQL 语句组成。使用存储过程通常具有以下一些好处：

（1）可增强 SQL 语言的功能和灵活性。

（2）良好的封装性。

（3）高性能。

（4）可减少网络流量。

（5）存储过程可作为一种安全机制来确保数据库的安全性和数据的完整性。

考点 2：存储过程体

在 MySQL 数据库中用于构造存储过程体的常用语法元素：

（1）局部变量：在 MySQL 中，可以使用 DECLARE 语句来声明局部变量，并且同时还可以对该局部变量赋予一个初始值。需要注意的事项如下：

①局部变量只能在存储过程体的 BEGIN--END 语句块中声明。

②局部变量必须在存储过程体的开头处声明。

③局部变量的作用范围仅限于声明它的 BEGIN-END 语句块，其他语句块中的语句不可以使用它。

④局部变量不同于用户变量，两者间的区别是：局部变量声明时，在其前面没有使用@符号，并且它只能被声明它的 BEGIN-END 语句块中的语句所使用；而用户变量在声明时，会在其名称前面使用@符号，同时已声明的用户变量存在于整个会话之中。

（2）SET 语句：在 MySQE 中，可以使用 SET 语句为局部变量赋值。

（3）SELECT…INTO 语句：在 MySQL 中，可以使用 SELECT…INTO 语句把选定列的值直接存储到局部变量中。存储过程体中的 SELECT…INTO 语句返回的结果集只能有一行数据。

（4）流程控制语句：在 MySQL 中，可以在存储过程体中，使用条件判断语句和循环语句这样两类用于控制语句流程的过程式 SQL 语句。

（5）游标：游标是一个被 SELECT 语句检索出来的结果集。在存储了游标后，应用程序或用户就可以根据需要滚动或浏览其中的数据。在 MySQL 中，可以使用 DECLARECURSOR 语句创建游标。在 MySQL 中，使用游标的具体步骤如下：声明游标→打开游标→读取数据→关闭游标。

在使用游标的过程中，需要注意以下几点：

①游标只能用于存储过程或存储函数中，不能单独在查询操作中使用。

②在存储过程或存储函数中可以定义多个游标，但是在一个 BEGIN…END 语句块中每一个游标的名字必须是唯一的。

③游标不是一条 SELECT 语句，是被 SELECT 语句检索出来的结果集。

考点 3：存储过程的创建、调用、删除

（1）创建存储过程：在 MySQL 数据库中通过命令行的方式来创建存储过程时，经常会用到一个十分重要的命令，即 DELIMITER 命令。

（2）调用存储过程：创建好存储过程后，可以使用 CALL 语句在程序或者其他存储过程中调用它。

（3）删除存储过程：可以使用 DROPPROCEDURE 语句删除数据库中已创建的存储过程。

考点 4：存储函数与存储过程的异同

存储函数与存储过程一样，都是由 SQL 语句和过程式语句所组成的代码片断，并且可以被应用程序和其他 SQL 语句调用。然而，它们之间存在如下几点区别：

（1）存储函数不能拥有输出参数，这是因为存储函数自身就是输出参数；而存储过程可以拥有输出参数。

（2）可以直接对存储函数进行调用，且不需要使用 CALL 语句；而对存储过程的调用，需要使用 CALL 语句。

（3）存储函数中必须包含一条 RETURN 语句，而这条特殊的 SQL 语句不允许包含于存储过程中。

考点 5：存储函数的创建、调用、删除

（1）创建存储函数：在 MySQL 中，可以使用 CREATEFUNCTION 语句创建存储函数。

（2）调用存储函数：成功创建存储函数后，就可以如同调用系统内置函数一样，使用关键字 SELECT 对其进行调用。

（3）删除存储函数：在 MySQL 中，可以使用 DROPFUNCTION 语句来实现删除。

第六章　数据库安全与保护

考点 1：数据库完整性概述

数据库完整性是指数据库中数据的正确性和相容性。数据完整性约束是为了防止数据库中存在不符合语义的数据，为了维护数据的完整性，DBMS 必须提供一种机制来检查数据库中的数据，以判断其是否满足语义规定的条件。这些加在数据库数据之上的语义约束条件就是数据完整性约束，而 DBMS 检查数据是否满足完整性约束条件的机制就称为完整性检查。

考点 2：完整性约束条件的作用对象

完整性检查是围绕完整性约束条件进行的，因而完整性约束条件是完整性控制机制的核心。完整性约束条件的作用对象可以是列、元组和表。

（1）列级约束：主要指对列的类型、取值范围、精度等的约束，具体包括如下内容：

①对数据类型的约束，其包括数据类型、长度、精度等。

②对数据格式的约束。

③对取值范围或取值集合的约束。

④对空值的约束。

⑤对空值的约束。

（2）元组约束：指元组中各个字段之间的相互约束。

（3）表级约束：指若干元组之间、关系之间的联系的约束。

考点 3：定义与实现完整性约束

关系模型的完整性规则是对关系的某种约束条件，关系模型中可以有三类完整性约束，分别是实体完整性、参照完整性和用户定义的完整性。

（1）实体完整性：是通过主键约束和候选键约束来实现的。

（2）参照完整性：是通过在创建表或更新表的同时定义一个外键声明来实现的。

（3）用户定义的完整性：MySQL 支持几种用户自定义完整性约束，分别是非空约束、CHECK 约束和触发器。

考点 4：命名完整性约束

与数据库中的表和视图一样，可以对完整性约束进行添加、删除和修改等操作。其中，为了删除和修改完整性约束，首先需要在定义约束的同时对其进行命名。命名完整性约束的方法是在各种完整性约束的定义说明之前加上关键字“CONSTRAINT”和该约束的名字。只能给基于表的完整性约束指定名字，而无法给基于列的完整性约束指定名字。

考点 5：更新完整性约束

当对各种约束进行命名后，就可以使用 ALTER TABLE 语句来更新与列或表有关的各种约束。

（1）完整性约束不能直接被修改。若要修改某个约束，实际上是用 ALTER TABLE 语句先删除该约束，然后再增加一个与该约束同名的新约束。

（2）使用 ALTER TABLE 语句，可以独立地删除完整性约束，而不会删除表本身。若使用 DROP TABLE 语句删除一个表，则表中所有的完整性约束都会自动被删除。

考点 6：触发器的创建、删除、使用

触发器是用户定义在关系表上的一类由事件驱动的数据库对象，也是一种保证数据完整性的方法。其主要作用是实现主键和外键不能保证的复杂的参照完整性和数据的一致性，从而有效地保护表中的数据。

（1）创建触发器：在 MySQL 中，可以使用 CREATETRIGGER 语句创建触发器。每个表最多支持 6 个触发器，即每条 INSERT、UPDATE 和 DELETE 的“之前”与“之后”。单一触发器不能与多个事件或多个表关联。

（2）删除触发器：与其他数据库对象一样，同样可以使用 DROP 语句将触发器从数据库中删除。

（3）使用触发器：在实际使用中，MySQL 所支持的触发器有三种，分别是 INSERT 触发器、DELETE 触发器和 UPDATE 触发器。

①INSERT 触发器可在 INSERT 语句执行之前或之后执行。

②DELETE 触发器可在 DELETE 语句执行之前或之后执行。

③UPDATE 触发器在 UPDATE 语句执行之前或之后执行。

考点 7：安全性与访问控制

数据库的安全性是指保护数据库以防止不合法的使用而造成数据泄露、更改或破坏。在 MySQL 数据库中，数据库系统对数据的安全管理是使用身份验证、数据库用户权限确认等访问控制措施，来保护数据库中的信息资源，以防止这些数据遭受破坏。

（1）用户账号管理

①创建用户账号：可以使用 CREATEUSER 语句来创建一个或多个 MySQL 账户，并设置相应的口令。

②删除用户：为了删除一个或多个用户账号以及相关的权限，可以使用 DROPUSER 语句。

③修改用户账号：可以使用 RENAMEUSER 语句修改一个或多个已经存在的 MySQL 用户账号。

④修改用户口令：可以使用 SETPASSWORD 语句修改一个用户的登录口令。

（2）账户权限管理

成功创建用户账号后，需要为该用户分配访问适当的权限，因为新创建的用户账号没有访问权限，只能登录 MySQL 服务器，不能执行任何数据库操作。

①权限的授予：新建的 MySQL 用户必须被授权，可以使用 GRANT 语句来实现。

②权限的转移：可以通过在 GRANT 语句中使用 WITH 子句来实现。

③权限的撤销：当需要撤销一个用户的权限、而又不希望将该用户从系统中删除时，可以使用 REVOKE 语句来实现。

考点 8：事务的概念

所谓事务是用户定义的一个数据操作序列，这些操作可作为一个完整的工作单元，要么全部执行，要么全部不执行，是一个不可分割的工作单位。

事务与程序很相似，但它们是两个彼此相联而又不同的概念：程序是静止的，事务是动态的，是程序的执行而不是程序本身；同一程序的多个独立执行可以同时进行，而每一步执行则是一个不同的事务

考点 9：事务的特征

（1）原子性：事务的原子性保证事务包含的一组更新操作是原子不可分的，即事务是不可分割的最小工作单位，所包含的这些操作是一个整体。

（2）一致性：一致性要求事务必须满足数据库的完整性约束，且事务执行完毕后将数据库由一个一致性状态转变到另一个一致性状态。其中，数据库的一致性状态是一种以一致性规则为基础的逻辑属性。

（3）隔离性：隔离性要求事务是彼此独立的、隔离的，即一个事务的执行不能被其他事务所干扰，一个事务对数据库变更的结果必须在它 COMMIT 后，另一个事务才能存取。多个事务并发执行时，其结果应等价于它们的一种顺序执行的结果，就如同串行调度执行事务一般。这一特性也称为可串行性。

（4）持续性：也称为永久性，是指一个事务一旦提交，它对数据库中数据的改变就应该是永久性的，且接下来的其他操作或故障不应该对其执行结果有任何影响。

考点 10：并发操作问题

事务是并发控制的基本单位，保证事务的 ACID 特征是事务处理的重要任务，而事务的 ACID 特征可能遭到破坏的原因之一是多个事务对数据库的并发操作造成的。为了保证事务的隔离性和一致性，DBMS 需要对并发操作进行正确调度。其中，完整性检验可以保证一个事务单独执行时，若输入的数据库状态是正确的，则其输出的数据库状态也是正确的。但当多个事务交错执行时，可能出现不一致问题，这也称为并发操作问题，典型的有如下三种：丢失更新、不可重复读和读“脏”数据。

考点 11：封锁

封锁是最常用的并发控制技术，它的基本思想是：需要时，事务通过向系统请求对它所希望的数据对象（如数据库中的记录）加锁，以确保它不被非预期改变。

（1）锁：一个锁实质上就是允许或阻止一个事务对一个数据对象的存取特权。确切的控制由封锁的类型决定。基本的封锁类型有两种：排他锁和共享锁。一般地，写操作要求 X 锁，读操作要求 S 锁。

（2）用封锁进行并发控制：封锁的工作原理如下：

①若事务 T 对数据 D 加了 X 锁，则所有别的事务对数据 D 的锁请求都必须等待直到事务 T 释放锁。

②若事务 T 对数据 D 加了 S 锁，则别的事务还可对数据 D 请求 S 锁，而对数据 D 的 X 锁请求必须等待直到事务 T 释放锁。

③事务执行数据库操作时都要先请求相应的锁，即对读请求 S 锁，对更新（插入、删除、修改）请求 X 锁。

④事务一直占有获得的锁直到结束（COMMIT 或 ROLLBACK）时释放。

（3）封锁的粒度：通常以粒度来描述封锁的数据单元的大小。DBMS 可以决定不同粒度的锁。由最底层的数据元素到最高层的整个数据库，粒度越细，并发性就越大，但软件复杂性和系统开销也就越大。

（4）封锁的级别：又称为一致性级别或隔离度。由各种锁的类型与其封锁期限组合可形成不同的封锁级别：

①0 级封锁：封锁的事务不重写其他非 0 级封锁事务的未提交的更新数据。

②1 级封锁：被封锁的事务不允许重写未提交的更新数据。

③2 级封锁：被封锁的事务既不重写也不读未提交的更新数据。

④3 级封锁：被封锁的事务不读未提交的更新数据，不写任何（包括读操作的）未提交数据。

（5）活锁与死锁：封锁带来的一个重要问题是可能引起“活锁”与“死锁”。在并发事务处理过程中，由于锁会使一事务处于等待状态而调度其他事务处理，因而该事务可能会因优先级低而永远等待下去，这种现象称为“活锁”。两个以上事务循环等待被同组中另一事务锁住的数据单元的情形，称为“死锁”。预防死锁的办法主要有：①一次性锁请求；②锁请求排序；③序列化处理；④资源剥夺。

（6）可串行性：在数据库系统中，可串行性就是并发执行的正确性准则。

（7）两段封锁法：是一种最简单而有效的保障封锁其调度是可串行性的方法。两段封锁法是事务遵循两段锁协议的调度方法。

考点 12：备份与恢复

在数据库的实际使用过程中，存在着一些不可预估的因素，会造成数据库运行事务的异常中断，从而影响数据的正确性，甚至会破坏数据库，使数据库中的数据部分或全部丢失。这些因素可能是：计算机硬件故障、计算机软件故障、病毒、人为误操作、自然灾害、盗窃。面对这些可能的因素会造成数据丢失或被破坏的风险，数据库系统提供了备份和恢复策略来保证数据库中数据的可靠性和完整性。

数据库备份是指通过导出数据或者复制表文件的方式来制作数据库的复本；数据库恢复则是当数据库出现故障或遭到破坏时，将备份的数据库加载到系统，从而使数据库从错误状态恢复到备份时的正确状态。

数据库的恢复是以备份为基础的，它是与备份相对应的系统维护和管理操作。系统进行恢复操作时，先执行一些系统安全性的检查，包括检查所要恢复的数据库是否存在、数据库是否变化及数据库文件是否兼容等，然后根据所采用的数据库备份类型采取相应的恢复措施。另外，通过备份和恢复数据库，也可以实现将数据库从一个服务器移动或复制到另一个服务器的目的。

在 MySQL 中使用 SQL 语句备份与恢复数据库中表数据的方法：

（1）使用 SELECTINTO…OUTFILE 语句备份数据。

（2）使用 LOADDATA…INFILE 语句恢复数据。

另外需要注意的是，在多个用户同时使用 MySQL 数据库的情况下，为了得到一个一致的备份，需要在指定的表上使用 LOCKTABLEStable_nameREAD 语句做一个读锁定，以防止在备份过程中表被其他用户更新；而当恢复数据时，则需要使用 LOCKTABLEStable_nameWRITE 语句做一个写锁定，以避免发生数据冲突。在数据库备份或恢复完毕之后需要使用 UNLOCKTABLES 语句对该表进行解锁。

第七章　数据库应用设计与开发实例

考点 1：功能性需求

经过调研，得知选课系统的用户类型有教务管理员、学生和教师。下面，可将系统的功能依据用户类型进行模块化划分。

（1）管理员后台模块：是专门为教务管理员使用的，主要用于系统的数据管理，包括学生管理、教师管理、班级管理和课程管理，其具体的功能需求有：①学生信息管理；②教师信息管理；③课程信息管理；④班级信息管理。

（2）学生使用模块：学生使用模块包含三个功能，具体功能需求有：①查询课程；②浏览所选课程；③查询成绩。

（3）教师使用模块：此模块主要完成教师登分和查询本人开设课程的操作。教师使用模块有两个功能，其功能需求有：①我的课程；②登分。

考点 2：非功能性需求

作为一个基于网络的在线选课系统，本系统采用浏览器/服务器（B/S）结构，即一种基于 Web 应用的客户/服务器结构，因而对于客户端和服务器端，分别有不同的软、硬件环境需求。客户端要能在支持 IE 的浏览器上运行，并在客户机上安装了网卡；服务器要求部署 WAMP 环境，即使用 Windows 作为操作系统，Apache 作为 Web 服务器，MySQL 作为数据库管理系统，PHP 语言作为服务器端脚本解释器，同时要求具有 8GB 内存、2TB 磁盘容量、千兆带宽。此外，还需要考虑质量要求。

考点 3：系统设计

（1）功能模块设计：①登录验证模块；②管理员后台模块；③学生使用模块；④教师使用模块。

（2）数据库设计：①确定实体；②局部信息结构；③全局信息结构；④逻辑结构与规范化设计。

根据系统的功能模块设计结果，以及前期的需求分析，可首先明确本系统的数据库范围，然后可通过使用 E-R 图作为数据库概念设计的描述工具，建立本系统所涉及的局部信息结构，再将各个局部信息结构合并成为一个优化的全局信息结构，最后将全局信息结构的 E-R 图转换为关系模型，并依据关系数据库规范化理论进行优化。

考点 4：系统实现

（1）数据库的实现

①首先，通过使用 MySQL 数据库的 CREATEDATABASE 命令，创建本数据库，并命名为 db_xuanke。

②然后，根据本应用数据库逻辑结构设计所得出的关系模式，通过使用 MySQL 数据库的 CREATETABLE 命名，在数据库 db_xuanke 中创建 12 张数据表，并建立各表的主键，从而构成主键索引。

（2）系统功能的实现

完成数据库及其基本数据表的创建之后，就可以根据系统业务功能分析的结果，开展系统功能实现的

编码工作了，这其中包括实现必要的数据库行为和应用软件的业务逻辑。

需要注意的是：在 MySQL 中，只有使用了 InnoDB 引擎的数据库或表才支持事务。

考点 5：系统测试与维护

完成系统的实现工作之后，在正式交付用户使用之前，需要对所开发的系统进行必要的测试，验证其是否满足用户的功能要求，并根据测试的结果，以及用户的反馈意见，对该系统进行进一步的修改、完善和维护工作。

（1）登录验证功能测试

（2）管理员后台主要功能测试：①学生信息管理功能；②课程信息管理功能；③学生使用模块功能测试；④教师使用模块功能测试。

第八章　数据管理技术的发展

考点 1：数据库技术发展概述

数据模型是数据库系统的核心和基础。以数据模型的发展为主线，数据库技术可以相应地分为三个发展阶段，即第一代的网状、层次数据库系统，第二代的关系数据库系统，以及新一代的数据库系统。

（1）第一代数据库系统

层次数据库系统和网状数据库系统的数据模型虽然分别为层次模型和网状模型，但实质上层次模型是网状模型的特例。它们都是格式化模型，它们从体系结构、数据库语言到数据存储管理均具有共同特征，是第一代数据库系统。第一代数据库系统有如下两类代表：

①1969 年 IBM 公司研制的层次模型数据库管理系统 IMS。

②DBTG 所提议的方法是基于网状结构的，是网状模型数据库系统的典型代表。

这两类数据库系统具有以下几个共同特点：

①支持三级模式（外模式、模式、内模式）的体系结构。

②用存取路径来表示数据之间的联系。

③独立的数据定义语言。

④导航的数据操纵语言。

（2）第二代数据库系统

支持关系数据模型的关系数据库系统是第二代数据库系统。第二代关系数据库系统具有模型简单清晰、理论基础好、数据独立性强、数据库语言非过程化和标准化等特点。

（3）新一代数据库系统

1990 年高级 DBMS 功能委员会发表了《第三代数据库系统宣言》的文章（以下简称《宣言》），提出了第三代数据库系统应具有的三个基本特征，《宣言》中称为三条基本原则。这三个基本特征如下：

①第三代数据库系统应支持数据管理、对象管理和知识管理。

②第三代数据库系统必须保持或继承第二代数据库系统的技术。

③第三代数据库系统必须对其他系统开放。

考点 2：从数据库到数据仓库

计算机系统中存在着两类不同的数据处理工作：

（1）一类是操作型处理，也称为联机事务处理，它是针对具体业务在数据库联机的日常操作，通常对少数记录进行查询和修改，用户较为关心操作的响应时间、数据的安全性、完整性和并发支持的用户数等问题，传统的数据库系统作为数据管理的主要手段，主要用于操作型处理；

（2）另一类是分析型处理，也称为联机分析处理，一般针对某些主题的历史数据进行分析，支持管理决策，它通常是对海量的历史数据查询和分析，如金融风险预测预警系统、证券股市违规分析系统等，这

些系统要访问的数据量非常大，查询和分析的操作十分复杂。

数据仓库的建立将操作型处理和分析型处理区分开来。1992 年数据仓库概念的创始人 W.H.Inmon 在其《BuildingtheDataWarehouse》一书中定义了数据仓库的概念：数据仓库是面向主题的、集成的、稳定的、随时间变化的数据集合，用以支持管理决策的过程。数据仓库不是可以买到的产品，而是一种面向分析的数据存储方案。数据仓库主要有以下特征：①面向主题；②集成性；③数据的非易失性；④数据的时变性。

一般地，数据仓库具有三个常用的重要概念，即粒度、分割和维。

（1）粒度：是指数据仓库的数据单位中保存数据的细化或综合程度的级别，细化程度越高，粒度级就越小，相反地，细化程度越低，粒度级就越大。在数据仓库环境中粒度之所以是主要的设计问题，是因为它深深地影响存储在数据仓库中的数据量的大小，同时影响数据仓库所能回的查询类型。

（2）分割：是将数据分散到各自的物理单元中，以便能分别处理，以提高数据处理的效率。数据分割后的单元称为切片。

（3）维：是人们观察数据的特定角度，是考虑问题时的一类属性。此类属性的集合构成一个维度。

此外，数据仓库有时也称为企业仓库。人们提出了数据集市体系结构的数据仓库概念。数据集市的基本思想是自下而上的数据仓库的开发方法。数据集市结构的数据仓库，又称为主题结构数据仓库，是按照主题进行构思所形成的数据仓库。一般可以将数据集市分为独立的数据集市和从属的数据集市或这两种数据集市的混合。

考点 3：数据挖掘技术

数据挖掘是从大量的、不完全的、有噪声的、模糊的、随机的实际应用数据中发现并提取隐藏在其中的、人们事先不知道的、但又是潜在有用的信息和知识的一种技术。它又被称为数据库中的知识发现。

数据挖掘具备下列几种功能：

（1）概念描述：通过数据挖掘技术，可以归纳总结出数据的某些特征。

（2）关联分析：常见的关联分析算法有 Apriori、FP-Growth 等。

（3）分类与预测：常见的分类模型及算法有决策树模型、神经网络模型、线性回归模型等。

（4）聚类：聚类是把数据按照相似性归纳成若干类别，同一类中的数据彼此相似，不同类中的数据相异。常用的聚类算法有 K-Means、GMM 等。

（5）孤立点检测：孤立点是指数据中与整体表现行为不一致的数据集合。

（6）趋势和演变分析：在实际使用中，数据挖掘的过程通常由以下六个步骤构成：

①确定业务对象；②数据的选择；③数据的预处理；④建模；⑤模型评估；⑥模型部署。

考点 4：大数据的定义

目前大数据尚无统一的定义，通常被认为是数据量很大、数据形式多样化的数据。

IBM 则是把大数据概括为 4 个 V，即大量化、多样化、快速化和真实性，强调大数据呈现价值稀疏性的特点。

一般意义上，大数据是指无法在可容忍的时间内用现有信息技术和软、硬件工具对其进行感知、获取、

管理、处理的服务的数据集合，且其具有如下特征：

（1）数据量巨大，即大量化。　　（2）数据种类繁多，即多样化。

（3）处理速度快，即快速化。　　（4）价值密度低。

考点 5：大数据管理技术典型代表

（1）大数据存储：分布式文件系统用于统一管理这些服务器节点上存储的数据，典型案例是 Hadoop 开源架构下的分布式文件系统（HDFS）。

（2）NoSQL 数据管理系统：NoSQL 是以互联网大数据应用为背景发展起来的分布式数据管理系统。该系统对数据进行划分，对各个数据分区进行备份，以应对结点可能的失败，提高系统可用性；通过大量结点的并行处理获得高性能，采用的是横向扩展的方式。它弥补了传统数据库由于事务等机制而带来的对海量数据高并发请求处理性能上的欠缺，采用一种非关系的方式来解决大数据存储和管理的问题。NoSQL 系统支持的数据存储模型通常有键值模型（NoSQL 数据库采用最多的数据存储方式）、文档模型、列模型和图模型等。

①键值存储：是 NoSQL 数据库采用最多的数据存储方式，它的数据是以 Key-Value 的形式存储的。

②文档存储：文档存储的存储格式可以多样化，适合存储系统日志等非结构化数据。常见的文档型数据库有 CouchDB、MongoDB 等。

③列存储：是以列为单位来存储数据的，擅长以列为单位读入数据，比较适合对某一列进行随机查询处理。

④图存储：图存储数据库是基于图理论构建的，使用结点、属性和边的概念。结点代表实体，属性保存与结点相关的信息，而边用来连接结点，表示两者关系。图数据库存储某些数据集非常快，可以把图直接映射到面向对象应用程序中。

（3）MapReduce 技术：是 Google 公司于 2004 年提出的大规模并行计算解决方案，主要应用于大规模廉价集群上的大数据并行处理。MapReduce 是一种并行编程模型。它把计算过程分解为两个阶段，即 Map 阶段和 Reduce 阶段。