

## 填空题

1. 数据库系统一般由（数据库）、应用系统、（数据库管理系统）、（数据库管理员）和用户构成。
2. 数据模型通常由（数据结构）、（数据操作）、（数据的约束条件）三部分组成。
3. 实体之间的联系可抽象为三类，它们是（一对一）、（一对多）、（多对多）。
4. 在数据库设计中，（数据字典）是系统中各类数据描述的集合，是进行详细的数据收集和数据分析所获得的主要成果。
5. （事务）是用户定义的一个数据库操作序列，这些操作要么全做要么全不做，是一个不可分割的工作单位。
6. 数据库系统在运行过程中，可能会发生故障。故障主要有（事务故障）、系统故障、（介质故障）和计算机病毒四类。
7. 并发控制的主要方法是采用（封锁）机制，其类型有（共享锁）和（排它锁）两种。
8. 数据库的完整性是指数据的（正确性）和（相容性）。
9. 在 SQL 语言中，为了数据库的安全性，设置了对数据的存取进行控制的语句，对用户授权使用（GRANT）语句，收回所授的权限使用（REVOKE）语句。
10. （数据库）是长期存储在计算机内的、有组织、可共享的数据集合。
11. 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）、（内模式）三级构成的。
12. 在一个关系 R 中，若每个数据项都是不可再分割的，那么 R 一定属于（第一（1NF））范式。
- E-R 图一般在数据库设计的（概念结构设计）阶段使用。
13. （事务）是用户定义的一个数据库操作序列，是一个不可分割的工作单位，它具有（原子性）、（一致性）、（隔离性）、（持续性）四个特性。
14. 数据库系统中，系统故障通常称为（软故障），介质故障称为（硬故障）。
15. 关系操作的特点是（集合）操作。
16. 若事务 T 对数据对象 A 加了 S 锁，则其他事务只能对数据 A 再加（S）锁，不能加（X）锁，直到事务 T 释放 A 上的锁。
17. 信息处理的方式而言，在文件系统阶段，（程序设计）处于主导地位，（数据）只起着服从程序设计需要的作用；而在数据库方式下，（数据）占据了中心位置。
18. 现实世界进行第一层抽象的模型，称为（概念）模型；对现实世界进行第二层抽象的模型，称为（数据）模型。
19. 数据库的三级模式结构是对（数据）的三个抽象级别。
22. 关系模式的操作异常往往是由（数据冗余）引起的。
23. 消除了非主属性对候选键局部依赖的关系模式，称为（2NF）模式。
24. 消除了非主属性对候选键传递依赖的关系模式，称为（3NF）模式。
27. 如果对数据库的并发操作不加以控制，则会带来四类问题：（丢失更新）、（读“脏数据”）和（不可重复读）。
28. 事务的执行次序称为（调度）。
29. 判断一个并发调度是否正确，可以用（可串行化）概念来解决。
30. 封锁能避免错误的发生，但会引起（活锁和死锁）问题。
31. S 封锁增加了并发度，但缺点是（容易发生死锁）。
32. 两段式封锁是可串行化的（充分）条件。
33. 数据库的完整性是指数据的（正确性）和（相容性）。
34. SQL 中的安全性机制，主要有两个：（视图）和（授权）。
36. 在数据管理技术的发展过程中，经历了人工管理阶段，文件系统阶段和数据库系统阶段，在这几个阶段中，数据的独立性是（数据库系统）阶段实现的。
37. 数据库是在计算机系统中按照数据模型组织、储存和应用的（数据的集合），支持数据库各种操作的软件系统叫（数据库管理系统），由计算机、操作系统、DBMS、数据库应用程序及用户等组成的一个整体叫做（数据库系统）。
38. 数据库的基本特点是（数据可以共享（或数据结构化）、数据独立性、数据冗余度小，易扩充、统一管理和控制）。
39. （数据库）是存储在计算机内组织的结构化的相关数据的集合。

40. 在数据库中存储的是（数据以及数据之间的联系）。
42. 数据库系统的核心是（数据库管理系统）。
43. 数据库技术中采取分级方法将数据库的结构划分成多个层次，是为了提高数据库的（逻辑独立性）和（物理独立性）。
44. 数据库系统的特点是（数据共享），数据独立，减少数据剩余，避免数据不一致和加强了数据保护。
45. 数据库管理系统能实现对数据库中数据的查询，插入，修改和删除等操作，这种功能称为（数据操纵功能）
46. 在数据库的三级模式结构中，描述数据库中的全体数据的全局逻辑结构和特征的是（模式）
48. 数据管理技术经历了（人工管理），（文件管理）和（数据库系统）三个阶段
49. 数据库是长期存储在计算机内，有（组织）的，可（共享）的数据集合
50. DBMS 是指（数据库管理系统）它是位于（用户）和（操作系统）之间的一层管理软件
51. 数据库管理系统的主要功能有（数据定义功能），（数据操作功能），数据库的运行管理和数据库的建立以及维护等 4 个方面。
52. 数据独立性又可分为（逻辑数据独立性）和（物理数据独立性）。
53. 当数据的存储结构改变了，应用程序不变，而由 DBMS 处理这种改变。这是指数据的（物理独立性）
54. 根据数据模型的应用目的不同，数据模型可分为（概念模型）和（数据模型）
55. 数据模型是由（数据结构），（数据操作）和（完整性约束）三部分组成的
56. 按照数据结构的类型来命名，数据模型分为（层次模型），（网状模型）和（关系模型）。
57. 现实世界的事物反映到人的头脑中经过思维加工是数据，这一过程要经历三个领域，依次是（现实世界），（信息世界）和（计算机世界）。
58. 实体之间的联系可抽象为三类它们是（1: 1），（1: m）和（m: n）
59. 按所使用的数据模型来分，数据库可分为（层次）、（关系）和（网状）三种类型。
60. 关系数据库规范化是为解决关系数据库中（插入、删除和数据冗余）问题而引入的。
61. 关系规范中的删除操作异常是指（不该删除的数据被删除），插入操作异常是指（应该插入的数据未被插入）。
62. 设计性能较优的关系模式称为规范化，规范化主要的理论依据是（关系规范化理论）。
63. 规范化理论是关系数据库进行逻辑设计的理论依据。根据这个理论，关系数据库中的关系必须满足：其每一属性都是（不可分解的）。
64. 数据模型的三要素是（数据结构），（数据操作）和（数据约束条件）。
65. 对于数据库系统，负责定义数据库内容，决定存储结构和存储策略及安全授权等工作的是（数据库管理员）。
66. 关系模型是把实体之间的联系用（二维表格）表示。
67. 在关系数据库中，当关系的类型改变时，用户程序也可以不变，这是（数据的逻辑独立性）。
68. 当数据库的存储结构改变时，可相应修改（模式/内模式映像），从而保持模式不变。
69. 在数据库系统的三级模式体系结构中，描述数据在数据库中的物理结构或存储方式的是（内模式）。
70. 在数据库系统的三级组织结构中，DBA 视图是（概念模式），系统程序员视图是（内模式），用户视图是（外模式）。
71. 表示实体及其联系的方法为（E-R 图），基本图素包括（实体）、（属性）和（联系）。习惯上实体用（矩形）表示，实体属性用（椭圆形）表示，联系用（菱形）表示。
72. 层次模型是用（树型）结构表示记录类型及其联系，其中上一层记录型和下一层记录型的联系是（1: M），层次模型不能直接支持（M: N）联系。
73. 在数据库的三级模式体系结构中，模式与内模式之间的映象（模式/内模式），实现了数据库的（物理）独立性。
71. 在关系模型中，若属性 A 是关系 R 的主码，则在 R 的任何元组中，属性 A 的取值都不允许为空，这种约束称为（实体完整性）规则。
72. 描述概念模型常用的方法是（实体—联系方法）。
73. 消除了部分函数依赖的 1NF 的关系模式，必定是（2NF）。
74. 关系模式中 R 的属性全部是主属性，则 R 的最高范式必定是（3NF）。
75. 在关系 A (S, SN, D) 和 B (D, CN, NM) 中，A 的主键是 S，B 的主键是 D，则 D 在 S 中称为（外码）。
76. 对于非规范化的模式，经过（将每一个分量化分为不可分的数据项）转变为 1NF，将 1NF 经过（消除每一个

- 非主属性对码的部分函数依赖,) 转变为 2NF, 将 2NF 经过 (消除非主属性对码的传递函数依赖) 转变为 3NF。
77. 关系模型中的关系模式至少是 (1NF)。
78. 关系模式中, 满足 2NF 的模式必定是 (1NF)。
79. 在关系模式 R 中, 若其函数依赖集中所有候选键都是决定因素, 则 R 最高范式是 (4NF)。
80. 候选关键字中的属性称为 (主属性)。
81. 在数据库设计中, 将 E-R 图转换成关系数据模型的过程属于 (逻辑设计阶段)。
82. 在数据库设计的 4 个阶段中, 每个阶段都有自己的设计内容。“在哪些表、在哪些属性上、建什么样的索引”, 这一内容应该属于 (物理) 设计阶段。
83. 在数据库设计的需求分析阶段, 数据字典通常由 (数据项)、(数据结构)、(数据流)、(数据存储) 和 (处理过程) 5 部分组成。
84. 在关系数据库设计中, 使每个关系达到 3NF, 这是 (逻辑设计阶段) 阶段完成的。
85. 设计概念结构时, 常用的数据抽象方法是 (聚集) 和 (概括)。
87. 就方法的特点而言, 需求分析阶段通常采用 (自顶向下逐步细化) 的分析方法; 概念设计阶段通常采用 (自底向上逐步总和) 的设计方法。
88. 在数据库设计中, 子类和超类存在着 (继承) 的联系。
89. 由分 E-R 图生成初步 E-R 图, 其主要任务是 (消除属性冲突)、(消除结构冲突) 和 (消除命名冲突)。
90. 由初步 E-R 图构成基本 E-R 图, 其主要任务是 (消除不必要的冗余)。
91. 假定一个 E-R 图包含有 A 实体和 B 实体, 并且从 A 到 B 存在着 1: N 的联系, 则转换成关系模后, 右以包含有 (1 或 2 ) 个关系模式。
92. 假定一个 E-R 图包含有 A 实体和 B 实体, 并且从 A 到 B 存在着 M: N 的联系, 则转换成关系模型后, 包含有 ( 3 ) 个关系模式。
93. (物理结构设计) 阶段与具体的 DBMS 密切相关。
94. 数据流图是数据库系统设计过程中 (概念结构设计) 阶段完成的。
95. 事物的所有操作要么全做, 要么全不做, 这是事物的 (原子) 特性。
96. 并发操作带来的问题主要有 (丢失修改), (不可重复读), (读脏数据)。
98. 若事物 T 对数据 R 已经加了 X 锁, 则其他事物对数据 R (不能加任何锁)。
99. 对数据对象施加封锁, 可能会引起活锁和死锁问题。预防死锁通常有 (一次封锁法) 和 (顺序封锁法) 两种方法。
100. 多个事物在某个调度下的执行是正确的, 是能保证数据库一致性的, 当且仅当该调度是 (可串行化的)。
101. 数据库管理系统通常提供授权功能来控制不同用户访问数据的权限, 这主要是为了实现数据库的 (安全性)。
103. 在关系 A (S, SN, D) 和 B (D, CN, NM) 中, A 的主键是 S, B 的主键是 D, 则 D 在 A 中称为 (外码)。
104. 在一个关系 R 中, 若每个数据项都是不在分割的, 那么 R 一定属于 (1NF)
105. 若关系为 1NF, 且它的每一个非主属性都 (完全函数依赖于或不部分函数依赖于) 码, 则该关系为 2NF。

## 二、选择题

1. 在数据管理技术的发展过程中, 经历了人工管理阶段、文件系统阶段和数据库系统阶段。在这几个阶段中, 数据独立性最高的是 ( A ) 阶段。
- A. 数据库系统      B. 文件系统      C. 人工管理      D. 数据项管理
2. 数据库的概念模型独立于 ( A )。
- A. 具体的机器和 DBMS      B. E-R 图      C. 信息世界      D. 现实世界
3. 在数据库的三级模式结构中, 描述数据库中全体数据的全局逻辑结构和特征的是 ( D )。
- A. 外模式      B. 内模式      C. 存储模式      D. 模式
4. 数据库的 **层次模型** 应满足的条件是 ( C )。
- A. 允许一个以上的结点无双亲, 也允许一个结点有多个双亲
- B. 必须有两个以上的结点
- C. 有且仅有一个结点无双亲, 其余结点都只有一个双亲
- D. 每个结点有且仅有一个双亲
9. 关系数据库中的码是指 ( D )。

- A. 能唯一决定关系的字段      B. 不可改动的专用保留字  
 C. 关键的很重要的字段      D. 能唯一标识元组的属性或属性集合
10. SQL 语言具有两种使用方式，分别为交互式 SQL 和（ C ）。
- A. 提示式 SQL      B. 多用户 SQL      C. 嵌入式 SQL      D. 解释式 SQL
11. 设计性能较优的关系模式称为规范化，规范化主要的理论依据是（ A ）。
- A. 关系规范化理论      B. 关系运算理论      C. 关系代数理论      D. 数理逻辑理论
12. 规范化过程主要为克服数据库逻辑结构中的插入异常，删除异常以及（ C ）缺陷。
- A. 数据的不一致性      B. 结构不合理      C. 冗余度大      D. 数据丢失
13. 当关系模式 R (A, B) 已属于 3NF，下列说法中（ B ）是正确的。
- A. 它一定消除了插入异常和删除异常      B. 仍存在一定的插入和删除异常  
 C. 一定属于 BCNF      D. A 和 C 都是
14. 消除了非主属性对码的部分函数依赖的 1NF 的关系模式，必定是（ B ）。
- A. 1NF      B. 2NF      C. 3NF      D. 4NF
15. E-R 图是数据库设计的工具之一，它适用于建立数据库的（ A ）。
- A. 概念模型      B. 逻辑模型      C. 结构模型      D. 物理模型
16. 在关系数据库设计中，设计关系模式是（ C ）的任务。
- A. 需求分析阶段      B. 概念设计阶段      C. 逻辑设计阶段      D. 物理设计阶段
17. 当局部 E-R 图合并成全局 E-R 图时可能出现冲突，不属于合并冲突的是（ B ）。
- A. 属性冲突      B. 语法冲突      C. 结构冲突      D. 命名冲突
18. DBMS 通常提供授权功能来控制不同用户访问数据的权限，这主要是为了实现数据库的（ D ）。
- A. 可靠性      B. 一致性      C. 完整性      D. 安全性
19. 设有两个事务 T1、T2，其并发操作如下图所示，下面评价正确的是（ D ）。
- A. 该操作不存在问题      B. 该操作丢失      C. 该操作不能重复读      D. 该操作读“脏”数据

T1	T2
①读 A=100 $A=A*2$ 写回 ② ③ROLLBACK 恢复 A=100	读 A=200

20. 若事务 T 对数据 R 已加了 X 锁，则其他事务对数据 R ( D )。
- A. 可以加 S 锁不能加 X 锁      B. 不能加 S 锁可以加 X 锁  
 C. 可以加 S 锁也可以加 X 锁      D. 不能加任何锁
21. 后备副本的用途是（ C ）。
- A. 安全性保障      B. 一致性控制      C. 故障后的恢复      D. 数据的转储
22. 用于数据库恢复的重要文件是（ C ）
- A. 数据库文件      B. 索引文件      C. 日志文件      D. 备注文件
23. 数据库系统与文件系统的主要区别是（ B ）。
- A. 数据库系统复杂，而文件系统简单。  
 B. 文件系统不能解决数据冗余和数据独立性问题，而数据库系统可以解决。  
 C. 文件系统只能管理程序文件，而数据库系统能够管理各种类型的文件。  
 D. 文件系统管理的数据量较少，而数据库系统可以管理庞大的数据量。
24. 在数据库中，下列说法（ A ）是不正确的。
- A. 数据库避免了一切数据的重复  
 B. 若系统是完全可以控制的，则系统可确保更新时的一致性  
 C. 数据库中的数据可以共享

- D. 数据库减少了数据冗余
26. 按所使用的数据模型来分，数据库可分为（ A ）三种类型。  
 A. 层次、关系和网状 B. 网状、环状和链状 C. 大型、中型和小型 D. 独享、共享和分时
29. SQL 语言是（ C ）语言。  
 A. 层次数据库 B. 网络数据库 C. 关系数据库 D. 非数据库
30. 关系数据库规范化是为解决关系数据库中（ A ）问题而引入的。  
 A. 插入、删除和数据冗余 B. 提高查询速度  
 C. 减少数据操作的复杂性 D. 保证数据的安全性和完整性
31. 关系模式中，满足 2NF 的模式，（ D ）。  
 A. 可能是 1NF B. 必定是 BCNF C. 必定是 3NF D. 必定是 1NF
32. 能消除多值依赖引起的冗余的是（ C ）。  
 A. 2NF B. 3NF C. 4NF D. BCNF
33. 从 E-R 模型向关系模型转换时，一个 M: N 联系转换为关系模式时，该关系模式的码是（ C ）。  
 A. M 端实体的码 B. N 端实体的码  
 C. M 端实体的码与 N 端实体的码的组合 D. 重新选取其他属性
34. 保护数据库，防止未经授权的或不合法的使用造成的数据泄露、更改破坏。这是指数据库的（ A ）。  
 A. 安全性 B. 完整性 C. 并发控制 D. 恢复
35. 设有两个事务 T1、T2，其并发操作如下图所示，下面评价正确的是（ C ）。  
 A. 该操作不存在问题 B. 该操作丢失 C. 该操作不能重复读 D. 该操作读“脏”数据

T1	T2
①读 A=10, B=5 求和 A+B=15 ② ③读 A=20, B=5 求和 25 验证错	读 A=10 A=A*2 写回

36. 若系统在运行过程中，由于某种原因，造成系统停止运行，致使事务在执行过程中以非正常方式终止，这时内存中的信息丢失，而存储在外存上的数据未受影响，这种情况称为（ B ）。
- A. 事务故障 B. 系统故障 C. 介质故障 D. 运行故障
37. 在数据库中存储的是（ C ）。  
 A. 数据 B. 数据模型 C. 数据以及数据之间的联系 D. 信息
38. 按所使用的数据模型来分，数据库可分为（ A ）三种类型。  
 A. 层次、关系和网状 B. 网状、环状和链状 C. 大型、中型和小型 D. 独享、共享和分时
39. 关系数据模型（ D ）。  
 A. 只能表示实体间的 1: 1 联系国 B. 只能表示实体间的 1: n 联系  
 C. 只能表示实体间的 m: n 联系 D. 可以表示实体间的上述三种联系
43. 候选码中的属性称为（ B ）。  
 A. 非主属性 B. 主属性 C. 复合属性 D. 关键属性
44. 数据库概念设计的 E-R 方法中，用属性描述实体的特征，属性在 E-R 图中，用（ D ）表示。  
 A. 矩形 B. 四边形 C. 菱形 D. 椭圆形
45. 数据库的（ B ）是指数据的正确性和相容性。  
 A. 安全性 B. 完整性 C. 并发控制 D. 恢复
46. 设有两个事务 T1、T2，其并发操作如下图所示，下面评价正确的是（ B ）。  
 A. 该操作不存在问题 B. 该操作丢失 C. 该操作不能重复读 D. 该操作读“脏”数据

T1	T2
①读 X=48 ② ③X←X+10 写回 X ④	读 X=48  X←X-2 写回 X

47. 若事务 T 对数据 R 已加了 S 锁，则其他事务对数据 R ( A )。  
 A. 可以加 S 锁不能加 X 锁      B. 不能加 S 锁可以加 X 锁  
 C. 可以加 S 锁也可以加 X 锁    D. 不能加任何锁
48. 事务 (Transaction) 是一个 ( C )  
 A. 程序      B. 进程      C. 操作序列      D. 完整性规则
49. 事务对 DB 的修改，应该在数据库中留下痕迹，永不消逝。这个性质称为事务的 ( A )  
 A. 持久性      B. 隔离性      C. 一致性      D. 原子性
50. 事务的执行次序称为 ( C )  
 A. 过程      B. 步骤      C. 调度      D. 优先级
51. 在事务依赖图中，如果两个事务的依赖关系形成一个循环，那么就会 ( B )  
 A. 出现活锁现象      B. 出现死锁现象      C. 事务执行成功      D. 事务执行失败

#### 四. 简答题

1. 什么是基本表？什么是视图？两者的区别和联系是什么？
2. 什么叫数据与程序的物理独立性？什么叫数据与程序的逻辑独立性？
3. 简述事务所具有的 ACID 特性。
4. DBMS 的完整性控制机制应具有哪些功能？
5. 什么是数据字典？数据字典的作用是什么？
6. 试叙述事务的 4 个性质，并结实每一个性质由 DBMS 的哪个子系统实现？每一个性质对 DBS 有什么益处？
7. 并发操作会产生几种不一致情况？用什么方法避免各种不一致的情况？
8. X 封锁与 S 封锁有什么区别？
9. 怎样进行系统故障的恢复？
10. 怎样进行介质故障的恢复？
11. 数据库在运行过程中可能产生的故障有哪几类？
12. 怎样进行事务故障的恢复？
13. 请给出预防死锁的若干方法。
14. 试述两段锁协议的概念。
15. 什么是数据库的完整性约束条件？

#### 五、应用题

3. 设有下列关系模式：

```

STUDENT (NO, NAME, SEX, BIRTHDAY, CLASS)
TEACHER (NO, NAME, SEX, BIRTHDAY, PROF, DEPART)
COURSE (CNO, CNAME, TNO)
SCORE (NO, CNO, DEGREE)

```

其中，学生信息表 STUDENT 由学号 (NO)、学生姓名 (NAME)、性别 (SEX)、出生年月日 (BIRTHDAY)、班级 (CLASS) 组成，记录学生的情况；教师信息表 TEACHER 由教师号 (NO)、教师姓名 (NAME)、性别 (SEX)、出生年月日 (BIRTHDAY)、职称 (PROF)、教师所在系 (DEPART) 组成，记录教师的情况；课程表 COURSE 由课程号 (CNO)、课程名称 (CNAME)、任课教师号 (TNO) 组成，记录所开课程及任课教师情况；成绩表 SCORE 由学生学号 (NO)、课程号 (CNO)、成绩 (DEGREE) 组成，记录学生选课情况及相应的成绩。用 SQL 语句实现以下操作

- 1) 显示 STUDENT 表中每个学生的姓名和出生日期。

SELECT NAME, BIRTHDAY FROM STUDENT;  
2) 显示 STUDENT 表中所有姓“王”的学生记录。

SELECT \* FROM STUDENT WHERE NAME LIKE ‘%王%’ ;

3) 显示成绩表中成绩在 60 分到 80 分之间的所有记录。

SELECT \* FROM SCORE WHERE DEGREE BETWEEN 60 AND 80;

4) 显示“男”教师及其所上的课程。(显示该教师的姓名和所上的课程名)

SELECT TEACHER.NAME, COURSE.CNAME FROM TEACHER, COURSE  
WHERE TEACHER.NO=COURSE.TNO AND SEX=‘男’ ;

5) 选出和“李军”同学同性别并同班的学生姓名。

SELECT NAME FROM STUDENT  
WHERE SEX= (SELECT SEX FROM STUDENT WHERE NAME=‘李军’) AND  
CLASS= (SELECT CLASS FROM STUDENT WHERE NAME=‘李军’);

6) 向 STUDENT 表中插入一条学生记录。(学号: 999, 姓名: 程功, 性别: 男, 出生年月日: 10/01/80, 班级: 95035)

INSERT INTO STUDENT VALUES(‘999’, ‘程功’, ‘男’, ‘10/01/80’, ‘95035’);

7) 在 STUDENT 表中将学号为“999”的学生的班号改为“95031”

UPDATE STUDENT SET=‘95031’ WHERE NO=‘999’ ;

8) 在 STUDENT 表中删除学号为“999”的学生记录。

DELETE FROM STUDENT WHERE NO=‘999’ ;

4. 设有下列四个关系模式:

S (SNO, SNAME, CITY)  
P (PNO, PNAME, COLOR, WEIGHT)  
J (JNO, JNAME, CITY)  
SPJ (SNO, PNO, JNO, QTY)

其中, 供应商表 S 由供应商号 (SNO)、供应商姓名 (SNAME)、供应商所在城市 (CITY) 组成, 记录各个供应商的情况; 零件表 P 由零件号 (PNO)、零件名称 (PNAME)、零件颜色 (COLOR)、零件重量 (WEIGHT) 组成, 记录各种零件的情况; 工程项目表 J 由项目号 (JNO)、项目名 (JNAME)、项目所在城市 (CITY) 组成, 记录各个工程项目的情况; 供应情况表 SPJ 由供应商号 (SNO)、零件号 (PNO)、项目号 (JNO)、供应数量 (QTY) 组成, 记录各供应商供应各种零件给各工程项目的数量。用 SQL 语言完成下列操作:

①找出工程项目 J2 使用的各种零件的名称及其数量

SELECT PNAME, QTY FROM SPJ, P WHERE SPJ.PNO=P.PNO AND JNO=‘J2’ ;

②找出所有零件的名称、颜色、重量

SELECT PNAME, COLOR, WEIGHT FROM P;

③找出上海厂商供应的所有零件号码

SELECT DISTINCT PNO FROM SPJ  
WHERE SNO IN (SELECT SNO FROM S WHERE CITY=‘上海’);

④从供应商关系中删除 S2 (供应商号) 的记录, 并从供应情况关系中删除相应的记录

DELETE FROM S WHERE SNO=‘S2’;

DELETE FROM SPJ WHERE SNO=‘S2’;

⑤请将 (S2, P4, J6, 200) 插入供应情况关系

INSERT INTO SPJ VALUES (‘S2’, ‘P4’, ‘J6’, 200);

6. 对于教学数据库的 3 个基本表

S (S#, SNAME, AGE, SEX)

SC (S#, C#, CNAME)

C (C#, CNAME, TEACHER) 试用关系代数表达式表示下列查询语句:

1) 检索 LIU 老师所授课程的课程号和课程名。

SELECT C#, CNAME FROM C WHERE TEACHER=‘LIU’ ;

2) 检索年龄大于 23 岁的男学生的学号和姓名。

```
SELECT S#, SNAME FROM S WHERE AGE>23 AND SEX= 'M';
```

3) 检索学号为 S3 学生所学课程的课程名与任课教师名。

```
SELECT CNAME, TEACHER FROM SC, C WHERE SC.C#=C.C# AND S#= 'S3' ;
```

4) 检索至少选修 LIU 老师所授课程中一门课的女学生姓名。

连接查询方式

```
SELECT SNAME FORM S, SC, C  
WHERE S.S#=SC.S# AND SC.C#=C.C# AND SEX= 'F' AND TEACHER= 'LIU' ;
```

嵌套查询方式

```
SELECT SNAME FORM S WHERE SEX = 'F' AND S# IN  
(SELECT S# FORM SC WHERE C# IN  
(SELECT C# FORM C WHERE TEACHER = 'LIU' ));
```

存在量词方式

```
SELECT SNAME FORM S WHERE SEX= 'F' AND EXISTS  
(SELECT * FORM SC WHERE SC.S#=S.S# AND EXISTS  
(SELECT * FORM C WHERE C.C#=SC.C# AND TEACHER= 'LIU' ));
```

5) 检索 WANG 同学不学的课程的课程名。

```
SELECT C# FROMC C WHERE NOT EXISTS  
(SELECT * FORM S, SC WHERE S.S#=SC.S# AND SC.C#=C.C# AND SNAME= 'WANG' );
```

6) 检索至少选修两门课的学生学号。

```
SELECT DISTINCT X.S# FROM SC X, SC Y WHERE X.S#=Y.S# AND X.C#!=Y.C#;
```

7) 检索全部学生都选修的课程的课程号与课程名。

```
SELECT C#, CNAME FROM C WHERE NOT EXISTS  
(SELECT * FROM S WHERE NOT EXISTS  
(SELECT * FROM SC WHERE S#=S.S# AND C#=C.C#));
```

用集合包含

```
SELECT C#, CNAME FROM C WHERE (SELECT S# FROM SC WHERE C#=C.C# )  
CONTAINS
```

```
SELECT S# FROM S;
```

或

```
SELECT C#, CNAME FROM C WHERE NOT EXISTS  
((SELECT S# FROM S) EXCEPT  
(SELECT S# FROM SC WHERE C#=C.C#));
```

8) 检索选修课程包含 LIU 老师所授课程的学生学号。

```
SELECT DISTINCT S# FROM SC X WHERE NOT EXISTS  
(SELECT * FROM C WHERE TEACHER= 'LIU' AND NOT EXISTS  
(SELECT * FROM SC Y WHERE Y.S#=X.S# AND X.C#=C.C#));
```

或: SELECT DISTINCT S# FROM SC X WHERE NOT EXISTS

```
((SELECT C# FROM C WHERE TRACHER= 'LIU' ) EXCEPT  
(SELECT C# FROM SC Y WHERE Y.S#=X.S#));
```

7. 试用 SQL 查询语句表达下列对上题中 3 个基本表 S、SC、C 的查询:

1) 统计有学生选修的课程门数。

```
SELECT COUNT (DISTINCT C#) FROM SC;
```

2) 求选修 C4 课程的女学生的平均年龄。

```
SELECT AVG(AGE) FROM S, SC WHERE S.S#=SC.S# AND C#= 'C' AND SEX= 'F' ;
```

3) 求 LIU 老师所授课程的每门课程的平均成绩。

```
SELECT C.C#, AVG(GRADE) FROM SC, C WHERE SC.C#=C.C# AND TEACHER= 'LIU'
```

GROUP BY C.C#;

- 4) 统计每门课程的学生选修人数(超过10人的课程才统计)。要求显示课程号和人数,查询结果按人数降序排列,若人数相同,按课程号升序排列。

```
SELECT C#, COUNT(S#) FROM SC  
GROUP BY C# HAVING COUNT(*)>10  
ORDER BY 2 DESC, 1;
```

- 5) 检索学号比WANG同学大,而年龄比他小的学生姓名。

```
SELECT SNAME FROM S WHERE S#>ALL (SELECT S# S WHERE SNAME= 'WANG' )  
AND AGE<ALL (SELECT AGE FROM S WHERE SNAME= 'WANG' );
```

- 6) 在表SC中检索成绩为空值的学生学号和课程号。

```
SELECT S#, C# FROM SC WHERE GRADE IS NULL;
```

- 7) 检索姓名以L打头的所有学生的姓名和年龄。

```
SELECT SNAME, AGE FROM S WHERE SNAME LIKE 'L%';
```

- 8) 求年龄大于女同学平均年龄的男学生姓名和年龄。

```
SELECT SNAME, AGE FROM S WHERE SEX= 'M' AND  
AGE >(SELECT AVG(AGE) FROM S WHERE SEX= 'F' );
```

- 9) 求年龄大于所有女同学年龄的男学生姓名和年龄。

```
SELECT SNAME, AGE FROM S WHERE SEX= 'M' AND  
AGE >ALL(SELECT AGE FROM S WHERE SEX= 'F' );
```

8. 用SQL更新语句表达对上题教学数据库中关系S、SC、C作如下更新操作:

- 1) 往关系C中插一个课程元组。

```
INSERT INTO C VALUES( 'C8' , 'VC++' , 'BAO' );
```

- 2) SC中删除尚无成绩的选课元组。

```
DELETE FROM SC WHERE GRADE IS NULL;
```

- 3) 把选修LIU老师课程的女同学选课元组全部删去。

```
DELETE FROM SC WHERE S# IN (SELECT S# FROM S WHERE SEX = 'F' )  
AND C# IN(SELECT C# FROM C WHERE TEACHER= 'LIU' );
```

- 4) 把MATHS课不及格的成绩全改为60分。

```
UPDATE SC SET GRADE = 60  
WHERE GRADE<60 AND C# IN (SELECT C# FROM C WHERE CNAME= 'MATHS' );
```

- 5) 把低于所有课程总平均成绩的女同学成绩提高5%。

```
UPDATE SC SET GRADE = GRADE *1.05  
WHERE S# IN (SELECT S# FROM S WHERE SEX= 'F' ) AND GRADE<(SELECT AVG(GRADE) FROM SC);
```

- 6) 在表SC中修改C4课程的成绩,若成绩小于等于70分提高5%,若成绩大于70分提高4%(用两种方法实现,一种方法是用两个UPDATE语句实现,另一种方法是用带CASE操作的一个UPDATE语句实现)。

```
UPDATE SC SET GRADE=GRADE * 1.04  
WHERE C# = 'C4' AND GRADE>70;  
UPDATE SC SET GRADE =GRADE * 1.05  
WHERE C # = 'C4' AND GRADE<=70;
```

- 7) 在表SC中,当某个成绩低于全部课程的平均成绩时,提高5%。

```
UPDATE SC SET GRADE = GRADE * 1.05  
WHERE GRADE<(SELECT AVG(GRADE) FROM SC);
```

9. 设数据库中有3个关系:

职工表EMP(E#, ENAME, AGE, SEX, ECITY),其属性分别表示职工工号、姓名、年龄、性别和籍贯。

工作表WORKS(E#, C#, SALARY),其属性分别表示职工工号、工作的公司编号和工资。

公司表COMP(C#, CNAME, CITY),其属性分别表示公司编号、公司名称和公司所在城市。

试用SQL语言写出下列操作:

1) 检索超过 50 岁的男职工的工号和姓名。

```
SELECT E#, ENAME FROM EMP WHERE AGE>50 AND SEX= 'M' ;
```

2) 假设每个职工只能在一个公司工作，检索工资超过 1000 元的男性职工工号和姓名。

```
SELECT EMP.E#, ENAME FROM EMP, WORKS  
WHERE EMP.E#=WORKS.E# AND SALARY>1000;
```

3) 假设每个职工可在多个公司工作，检索在编号为 C4 和 C8 公司兼职的职工工号和姓名。

```
SELECT A.E#, A.ENAME FROM EMP A, WORKS B, WORKS C  
WHERE A.E#=B.E# AND B.E#=C.E# AND B.C#= 'C4' AND C.C#= 'C8' ;
```

4) 检索在“联华公司”工作、工资超过 1000 元的男性职工的工号和姓名。

```
SELECT A.E#, A.ENAME FROM EMP A, WORKS B, COMP C  
WHERE A.E#=B.E# AND B.C#=C.C# AND CNAME= '联华公司'  
AND SALARY>1000 AND SEX= 'M' ;
```

5) 假设每个职工可在多个公司工作，检索每个职工的兼职公司数目和工资总数，显示 (E#, NUM, SUM\_SALARY)，分别表示工号、公司数目和工资总数。

```
SELECT E#, COUNT(C#) AS NUM, SUM(SALARY) AS SUM_SALARY  
FROM WORKS GROUP BY E#;
```

6) 工号为 E6 的职工在多个公司工作，试检索至少在 E6 职工兼职的所有公司工作的职工工号。

```
SELECT X.E# FROM WORKS X WHERE NOT EXISTS  
(SELECT * FROM WORKS Y WHERE E#= 'E6' AND NOT EXISTS  
(SELECT * FROM WORKS Z WHERE Z.E#=X.E# AND Z.C#=Y.C#));
```

7) 检索联华公司中低于本公司平均工资的职工工号和姓名。

```
SELECT A.E#, A.ENAME FROM EMP A, WORKS B, COMP C  
WHERE A.E#=B.E# AND B.C#=C.C# AND CNAME = '联华公司' AND SALARY<
```

(SELECT AVG(SALARY) FROM WORKS, COMP WHERE WORKS.C#=COMP.C# AND CNAME = '联华公司')； 8) 在每个公司中为 50 岁以上职工加薪 100 元（若职工为多个公司工作，可重复加）。

```
UPDATE WORKS SET SALARY = SALARY+100  
WHERE E# IN (SELECT E# FROM EMP WHERE AGE>50);
```

10) 在 EMP 表和 WORKS 表中删除年龄大于 60 岁的职工有关元组。

```
DELETE FROM WORKS WHERE E# IN (SELECT E# FROM EMP WHERE AGE>60);  
DELETE FROM EMP WHERE AGE >60;
```

10. 设有关系模式：SB (SN, SNAME, CITY) 其中，SB 表示供应商，SN 为供应商代号，SNAME 为供应商名字，CITY 为供应商所在城市，主关键字为 SN。

PB (PN, PNAME, COLOR, WEIGHT) 其中 PB 表示零件，PN 为零件代号，PNAME 为零件名字，COLOR 为零件颜色，WEIGHT 为零件重量，主要字键为 PN。

JB (JN, JNAME, CITY) 其中，JB 表示工程，JN 为工程编号，JNAME 为工程名字，CITY 为工程所在城市，主关键字为 JN。

SPJB (SN, PN, CITY) 其中，SPJB 表示供应关系，SN 是为指定工程提供零件的供应商代号，PN 为所提供的零件代号，JN 为工程编号，OTY 表示提供的零件数量，主关键字为 SN, PN, JN，外关键字为 SN, PN, JN。

如图所示表示供应商 (S) - 零件 (P) - 工程 (J) 数据库表，写出实现以下各题功能的 SQL 语句：

SB

PB

SN	SNAME	CITY
S1	N1	上海
S2	N2	北京
S3	N3	北京
S4	N4	上海
S5	N5	南京

	PN	PNAME	COLOR	WEIGHT	
SPJB	P1	PN1	红	12	JB
	P2	PN2	绿	18	
	P3	PN3	蓝	20	
	P4	PN4	红	13	
	P5	PN5	蓝	11	
	P6	PN6	绿	15	

SN	PN	JN	PTY
S1	P1	J1	200
S1	P1	J4	700
S2	P3	J1	400
S2	P3	J2	200
S2	P3	J3	200
S2	P3	J4	500
S2	P3	J5	600
S2	P3	J6	400
S2	P3	J7	800
S2	P3	J2	100
S3	P3	J1	200
S3	P4	J2	500
S4	P6	J3	300
S4	P6	J7	300
S5	P2	J2	200
S5	P2	J4	100
S5	P5	J5	500
S5	P5	J7	100
S5	P6	J2	200
S5	P1	J4	1000
S5	P3	J4	1200
S5	P4	J4	800
S5	P5	J4	400
S5	P6	J4	500

JN	JNAME	CITY
J1	JN1	上海
J2	JN2	广州
J3	JN3	南京
J4	JN4	南京
J5	JN5	上海
J6	JN6	武汉
J7	JN7	上海

1) 取出为工程 J1 提供红色零件的供应商代号

```
SELECT DISTINCT SPJB.SN FROM SPJB, PB
WHERE PB.PN=SPJB.PN AND SPJB.JN='J1' AND PB.COLOR='红' ;
```

2) 取出为所在城市为上海的工程提供零件的供应商代号

```
SELECT DISTINCT SPJB.SN FROM SPJB, JB
WHERE SPJB.JN AND JB.CITY='上海' ;
```

3) 取出供应商与工程所在城市相同的供应商提供的零件代号

```
SELECT DISTINCT SPJB.PN FROM SB, JB, SPJB
WHERE SB.SN=SPJB.SN AND JB.JN=SPJB.JN AND SB.CITY=JB.CITY
```

## 六、综合题

1. 设有如下所示的关系 R (码为: 课程名)

问: 1) 该关系模式为第几范式? 为什么?

2) 是否存在删除操作异常? 若存在, 则说明在什么情况下发生的?

3) 将它分解为高一级范式, 分解后的关系是如何解决分解前可能存在的删除操作异常问题的?

关系 R

课程名	教师名	教师地址
C1	王小强	D1
C2	李鸿雁	D2
C3	王小强	D1
C4	张言	D1

答: (1)  $R \in 2NF$   
 $\because R$  的候选码为课程名, 存在课程名  $\rightarrow$  教师名,  
 $\text{教师名} \rightarrow \text{课程名}$ , 教师名  $\rightarrow$  教师地址  
 $\therefore \text{课程名} \rightarrow \text{教师地址}$   
 即存在非主属性对候选码的传递函数依赖关系  
 $\therefore R$  不属于 3NF  
 又  $\because$  不存在非主属性对候选码的部分函数依赖  
 $\therefore R \in 2NF$

(2) 存在删除操作异常, 当删除某门课程时, 教师的信息也被删除了。

(3) 分解 R

R1

课程名	教师名
C1	王小强
C2	李鸿雁
C3	王小强
C4	张言

R2

教师名	教师地址
王小强	D1
李鸿雁	D2
张言	D1

3. 设有如图所示关系 R

关系 R

课程名	教师名	教师地址
C1	马千里	D1
C2	于得水	D1
C3	余快	D2
C4	于得水	D1

- 1) 它为第几范式? 为什么?
- 2) 是否存在删除操作异常? 若存在, 则说明是在什么情况下发生的?
- 3) 将它分解为高一级范式, 分解后的关系是如何解决分角前可能存在的删除操作异常问题?

答: 1) 它是 2NF

$\because R$  的候选关键字为课程名, 而课程名  $\rightarrow$  教师名, 教师名  $\rightarrow$  教师地址  
 $\therefore$  课程名  $\rightarrow$  教师地址, 即存在非主属性教师地址对候选关键字课程名的传递函数依赖, 因此  $R$  不是 3NF。  
 $\therefore$  不存在非主属性对候选关键字是的部分函数依赖  
 $\therefore$  是 2NF。

- 2) 存在。当删除某门课程时会删除不该删除的教师的关信息。
- 3) 分解为高一级范式如图所示。

R1

课程名	教师名
C1	马千里
C2	于得水
C3	余快
C4	于得水

R2

教师名	教师地址
马千里	D1
于得水	D1
余快	D2

分解后, 若删除课程数据时, 仅对关系 R1 操作, 教师地址信息在关系 R2 中仍然保留, 不会丢失教师方面的信息。

4. 设有如图所示的关系 R:

职工号	职工名	年龄	性别	单位号	单位名
E1	ZHAO	20	F	D3	CCC
E2	QIAN	25	M	D1	AAA

E3	SEN	38	M	D3	CCC
E4	LI	25	F	D3	CCC

试问 R 属于第几范式?为什么?并如何规范化为 3NF?

答: R 不属于 3NF, 它是 2NF

$\because$  R 的候选关键字为职工号和职工名, 而:

职工号  $\rightarrow$  单位号, 单位号  $\rightarrow$  职工号, 单位号  $\rightarrow$  单位名

$\therefore$  职工号  $\longrightarrow$  单位名, 即存在非主属性单位名对候选关键职工号的传递函数依赖

规范化后的关系子模式为入图 4.25 所示的关系 R1 和 R2

R1

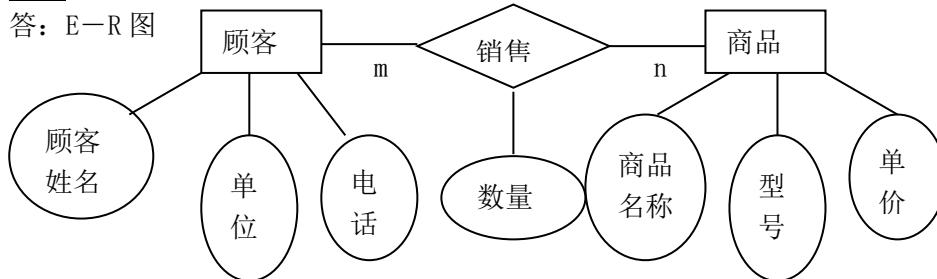
职工号	职工名	年龄	性别	单位号
E1	ZHAO	20	F	D3
E2	QIAN	25	M	D1
E3	SEN	38	M	D3
E4	LI	25	F	D3

R2

单位号	单位名
D3	CCC
D1	AAA

5. 设有商业销售记账数据库, 一个顾客(顾客姓名, 单位, 电话号码)可以买多种商品, 一种商品(商品名称, 型号, 单价)供应多个顾客。试画出对应的 E-R 图并将其转换为关系模式。

答: E-R 图



根据 E-R 图转换的关系模式为:

顾客(顾客姓名, 单位, 电话号码)

商品(商品名称, 型号, 单价)

销售(顾客姓名, 商品名称, 型号, 数量)

6. 某商业集团数据库中有 3 个实体集。一是“商店”实体集, 属性有商店编号、商店名、地址等; 二是“商品”实体集, 属性有商品号、商品名、规格、单价等; 三是“职工”实体集, 三是有职工编号、姓名、性别、业绩等。

商店与商品间存在“销售”关系, 每个商店可销售多种商品, 每种商品也可放在多个商店销售, 每个商店每销售一种商品, 有月销售量; 商店与职工间存在着“聘用”联系, 每个商店有许多职工, 每个职工只能在一个商店工作, 商店聘用职工有聘期和月薪。

1) 试画出 ER 图, 并在图上注明属性、联系的类型。

2) 将 ER 图转换成关系模式集, 并指出每个关系模式主键和外键。

答: 1) E-R 图如图所示。

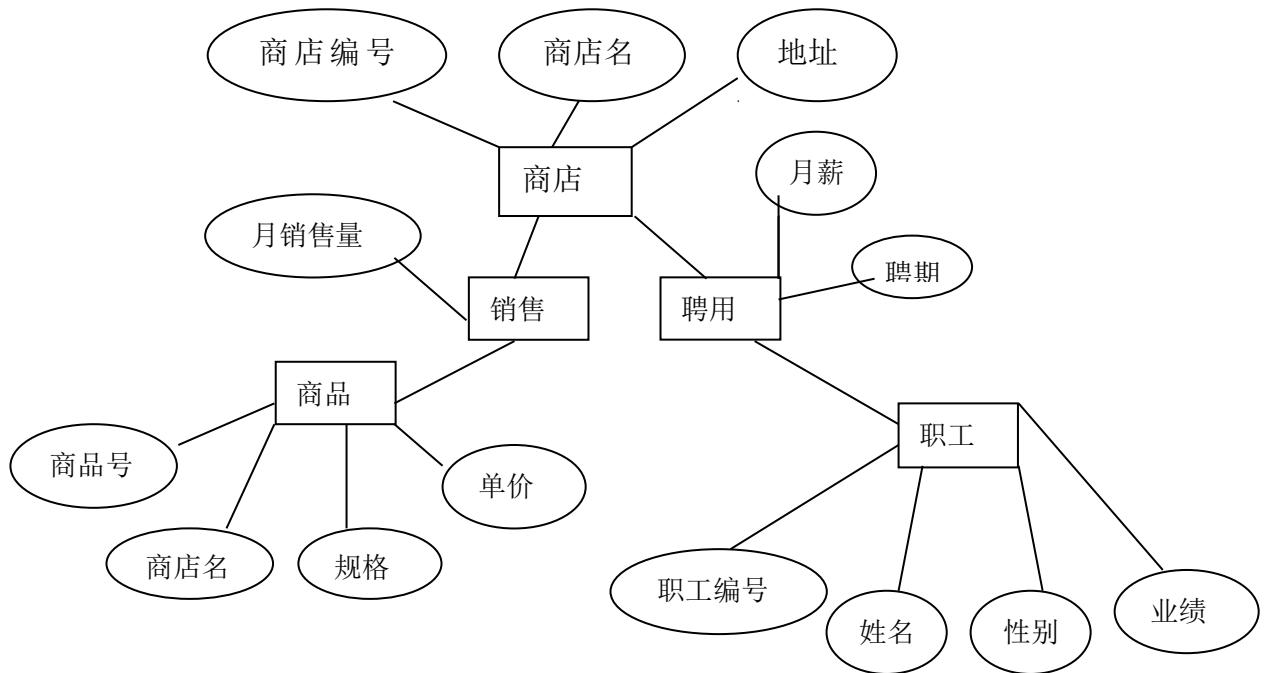
2) 这个 E-R 图可转换 4 个关系模式:

商店(商店编号, 商店名, 地址)

职工(职工编号, 姓名, 性别, 业绩, 商店编号, 聘期, 月薪)

商品(商品号, 商品名, 规格, 单价)

销售(商店编号, 商店号, 月销售量)



7. 设某

商业集团数据库中有 3 个实体集。一是“公司”实体集，属性有公司编号、公司名、地址等；二是“仓库”实体集，属性有仓库编号、仓库名、地址等；三是“职工”实体集，属性有职工编号、姓名、性别等。

公司与仓库间存在“隶属”联系，没个攻击管辖若干仓库，每个仓库只能属于一个公司管辖；仓库与职工间存在“聘用”联系，每个仓库可聘用多个职工，每个职工只能在一个仓库铬镍钢做，仓库聘用职工有聘期和工资。

1) 画出 ER 图，并在图上注明属性、联系的类型。

2) 将 ER 图转换成关系模式集，并指出每个关系迷失的主键和外键。

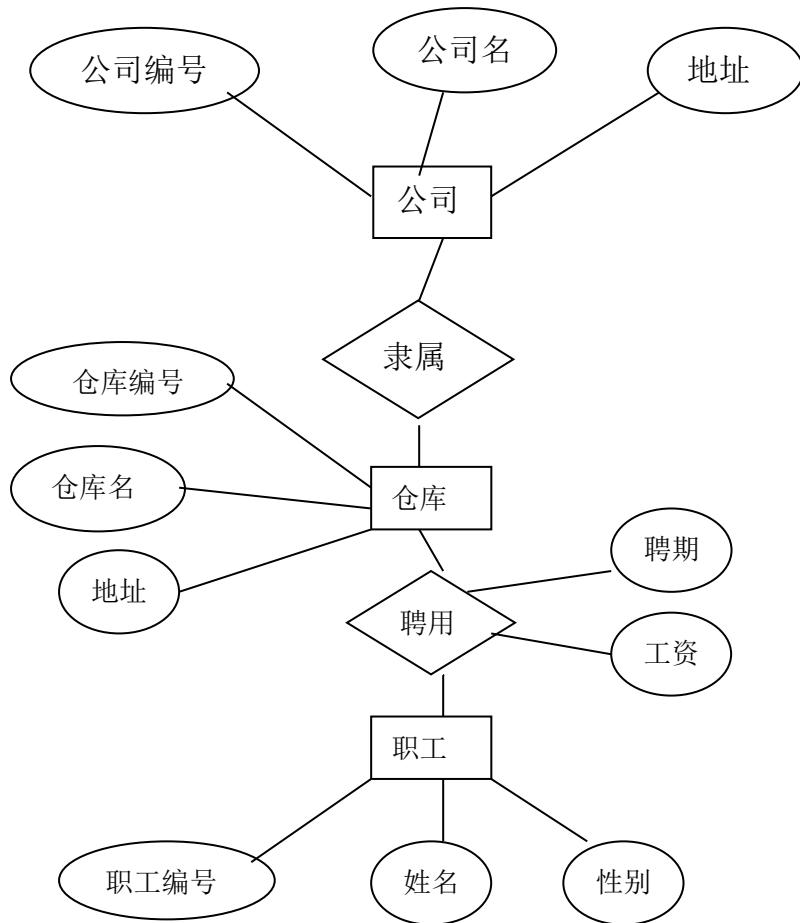
答：1) E-R 图所示。

2) 这个 E-R 图可转换 3 个关系模式：

公司 (公司编号, 公司名, 地址)

仓库 (仓库编号, 仓库名, 地址, 公司编号)

职工 (职工编号, 姓名, 性别, 仓库编号, 聘期, 工资)



8. 设某工商业集团数据库有 3 个实体集。一是“商品”实体集，属性有商品号、商品名、规格、单价等；二是“商店”实体集，属性有商店号、商店名、地址等；三是“供应商”实体集，属性有供应商编号、供应商名、地址等。

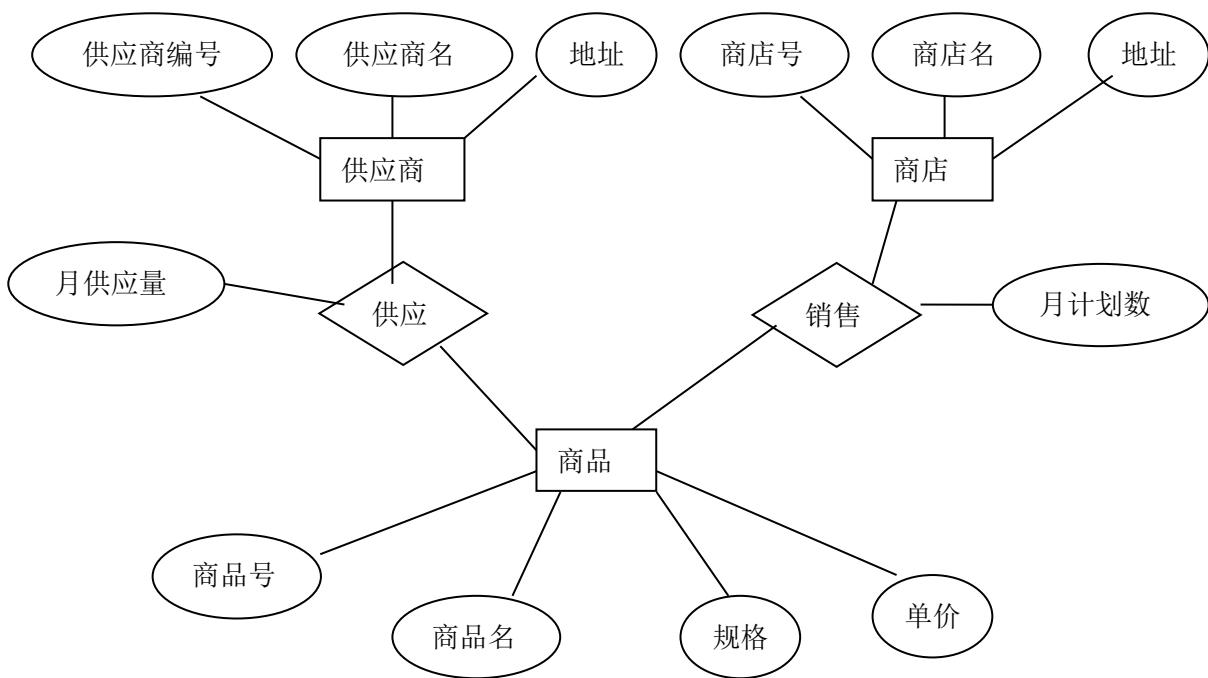
供应商与商品之间存在“供应”关系，每个供应商可供应多种商品，每种商品可向多个供应商订购，每个供应商供应每种商品有月供应量；商店与商品间存在“销售”联系，每个商店可销售多种商品，每种商品可在多个商店销售，没个商店销售每种商品有月计划数。

- 1) 试画出 ER 图，并在图上注明属性、联系的类型。
- 2) 将 ER 图转换成关系模式集，并指出每个关系模式的主键和外键。

答：1) E-R 图所示。

2) 这个 E-R 图可转换 5 个关系模式：

供应商（供应商编号，供应商名，地址）  
 商店（商店号，商店名，地址）  
 商品（商品号，商品名，规格，单价）  
 供应（供应商编号，商品号，月供应量）  
 销售（商店号，商品号，月计划数）



9. 某医院病房计算机管理中需要如下信息：

科室：科名，科地址，科电话，医生姓名

病房：病房号，床位号，所属科室号

医生：姓名，职称，所属科室名，年龄，工作证号

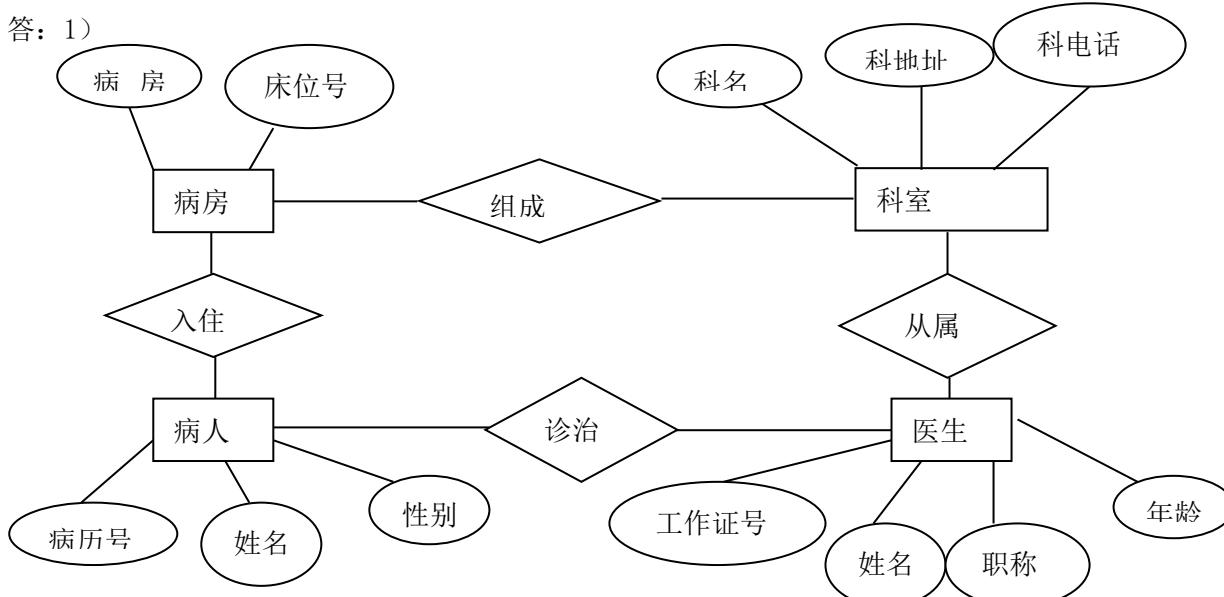
病人：病历号，姓名，性别，诊断，主管医生，病房号

其中，一个科室有多个病房，多个医生。一个病房只能属于一个科室，一个医生只属于一个科室，但可负责多个病人的诊治，一个病人的主管医生只有一个。

完成如下设计：

- 1) 设计核算计算机管理系统的 E-R 图；
- 2) 将该 E-R 图转换为关系模型结构；
- 3) 指出转换结果中每个关系模式的候选码

答：1)



2) 科室(科名, 科地址, 科电话)

病房(病房号, 床位号, 科室号)

医生(工作证号, 姓名, 职称, 科室名, 年龄)

病人(病历号, 姓名, 性别, 诊治, 主管医生, 病房号)

3) 每个关系模式的候选码如下：

科室的候选码是科名

病房的候选码是科室名+病房号

医生的候选码是工作证号

病人的候选码是病历号

10. 一个图书馆借阅管理数据库要求提供下述服务：

1) 可随时查询书库中现有书籍品种，数量与存放位置，所有各类书记均可由书号惟一标识。

2) 可随时查询书记借还情况，包括借书人单位，姓名，借书证号，借书日期和还书日期。

我们约定：任何人可借多种书，任何一种书可为多人所借，借书证号具有唯一性。

1) 当需要时，可通过数据库中保存的出版社的电报编号，电话，邮编及地址等信息向有关书籍的出版社增购有关书籍。我们约定，一个出版社可出版多种书籍，同一本书仅为一个出版社出版，出版社具有惟一性。

根据以上情况和假设，试作如下设计：

1) 构造 E-R 图；

2) 转换为等价的关系模式

答：1) E-R 图

2) 关系模式：

借书人(借书证号, 姓名单位)

图书(书号, 书名, 数量, 位置, 出版社名)

出版社(出版社名, 电报, 电话, 邮编, 地址)

借阅(借书证号, 书号, 借书日期, 还书日期)

