

CPU 执行算术运算或者逻辑运算时，常将源操作数和结果暂存在(1)中。

- (1) A. 程序计数器 (PC)
B. 累加器 (AC)
C. 指令寄存器 (IR)
D. 地址寄存器 (AR)

【答案】B

【解析】

寄存器是 CPU 中的一个重要组成部分，它是 CPU 内部的临时存储单元。寄存器既可以用来存放数据和地址，也可以存放控制信息或 CPU 工作时的状态。在 CPU 中增加寄存器的数量，可以使 CPU 把执行程序时所需的数据尽可能地放在寄存器中，从而减少访问内存的次数，提高其运行速度。但是寄存器的数目也不能太多，除了增加成本外，由于寄存器地址编码增加也会增加指令的长度。CPU 中的寄存器通常分为存放数据的寄存器、存放地址的寄存器、存放控制信息的寄存器、存放状态信息的寄存器和其他寄存器等类型。

程序计数器用于存放指令的地址。令当程序顺序执行时，每取出一条指令，PC 内容自动增加一个值，指向下一条要取的指令。当程序出现转移时，则将转移地址送入 PC，然后由 PC 指向新的程序地址。

程序状态寄存器用于记录运算中产生的标志信息，典型的标志为有进位标志位、零标志位、符号标志位、溢出标志位、奇偶标志等。

地址寄存器包括程序计数器、堆栈指示器、变址寄存器、段地址寄存器等，用于记录各种内存地址。

累加寄存器通常简称为累加器，它是一个通用寄存器。其功能是当运算器的算术逻辑单元执行算术或逻辑运算时，为 ALU 提供一工作区。例如，在执行一个减法运算前，先将被减数取出放在累加器中，再从内存储器取出减数，然后同累加器的内容相减，所得的结果送回累加器中。累加器在运算过程中暂时存放被操作数和中间运算结果，累加器不能用于长时间地保存一个数据。

指令寄存器：一般用来保存当前正在执行的一条指令。

地址寄存器：一般用来保存当前 CPU 所访问的内存单元的地址，以方便对内存的读写操作。

要判断字长为 16 位的整数 a 的低四位是否全为 0，则 (2)

- (2) A. 将 a 与 0x000F 进行“逻辑与”运算，然后判断运算结果是否等于 0
B. 将 a 与 0x000F 进行“逻辑或”运算，然后判断运算结果是否等于 F

C. 将 a 与 0xFFF0 进行“逻辑异或”运算，然后判断运算结果是否等于 0

D. 将 a 与 0xFFF0 进行“逻辑与”运算，然后判断运算结果是否等于 F

【答案】A

【解析】

要判断数的最后四位是否都为 0，应该将最后四位与 1 进行逻辑与运算，其它数位与 0 做逻辑与运算，最后判定最终的结果是否为 0；因此得出与 a 进行逻辑与运算的数：前 12 位为 0 最后 4 位为 1，即 0x000F

逻辑或运算：0 或 0=0；1 或 0=1；0 或 1=1；1 或 1=1；

逻辑与运算：0 与 0=0；1 与 0=0；0 与 1=0；1 与 1=1；

计算机系统中常用的输入/输出控制方式有无条件传送、中断、程序查询和 DMA 方式等。
当采用(3)方式时，不需要 CPU 执行程序指令来传送数据。

(3)A. 中断

B. 程序查询

C. 无条件传送

D. DMA

【答案】D

【解析】

直接程序控制（无条件传送/程序查询方式）：

无条件传送：在此情况下，外设总是准备好的，它可以无条件地随时接收 CPU 发来的输出数据，也能够无条件地随时向 CPU 提供需要输入的数据

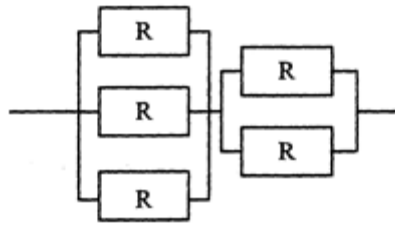
程序查询方式：在这种方式下，利用查询方式进行输入输出，就是通过 CPU 执行程序查询外设的状态，判断外设是否准备好接收数据或准备好了向 CPU 输入的数据，

中断方式：由程序控制 I/O 的方法，其主要缺点在于 CPU 必须等待 I/O 系统完成数据传输任务，在此期间 CPU 需要定期地查询 I/O 系统的状态，以确认传输是否完成。因此整个系统的性能严重下降。

直接主存存取（Direct Memory Access, DMA）是指数据在主存与 I/O 设备间的直接成块传送，即在主存与 I/O 设备间传送数据块的过程中，不需要 CPU 作任何干涉，只需在过程开始启动（即向设备发出传送一块数据的命令）与过程结束（CPU 通过轮询或中断得知过程是否结束和下次操作是否准备就绪）时由 CPU 进行处理，实际操作由 DMA 硬件直接完成，CPU 在传送过程中可做别的事情。

某系统由下图所示的冗余部件构成。若每个部件的千小时可靠度都为 R，则该系统的千

小时可靠度为(4)。



(4) A. $(1-R^3)(1-R^2)$

B. $(1-(1-R)^3)(1-(1-R)^2)$

C. $(1-R^3) + (1-R^2)$

D. $(1-(1-R)^3) + (1-(1-R)^2)$

【答案】B

【解析】本题考查系统可靠性。

计算机系统是一个复杂的系统，而且影响其可靠性的因素也非常繁复，很难直接对其进行可靠性分析。若采用串联方式，则系统可靠性为每个部件的乘积 $R=R_1 \times R_2 \times R_3 \times \cdots \times R_n$ ；若采用并联方式，则系统的可靠性为 $R=1-(1-R_1) \times (1-R_2) \times (1-R_3) \times \cdots \times (1-R_n)$ 。在本题中，既有并联又有串联，计算时首先我们要分别计算图中两个并联后的可靠度，它们分别为 $(1-(1-R)^3)$ 和 $(1-(1-R)^2)$ ，然后是两者串联，根据串联的计算公式，可得系统的可靠度为 $(1-(1-R)^3)(1-(1-R)^2)$ 。

已知数据信息为 16 位，最少应附加(5)位校验位，才能实现海明码纠错。

(5) A. 3

B. 4

C. 5

D. 6

【答案】C

【解析】

以下关于 Cache(高速缓冲存储器)的叙述中，不正确的是(6)。

(6) A. Cache 的设置扩大了主存的容量

B. Cache 的内容是主存部分内容的拷贝

C. Cache 的命中率并不随其容量增大线性地提高

D. Cache 位于主存与 CPU 之间

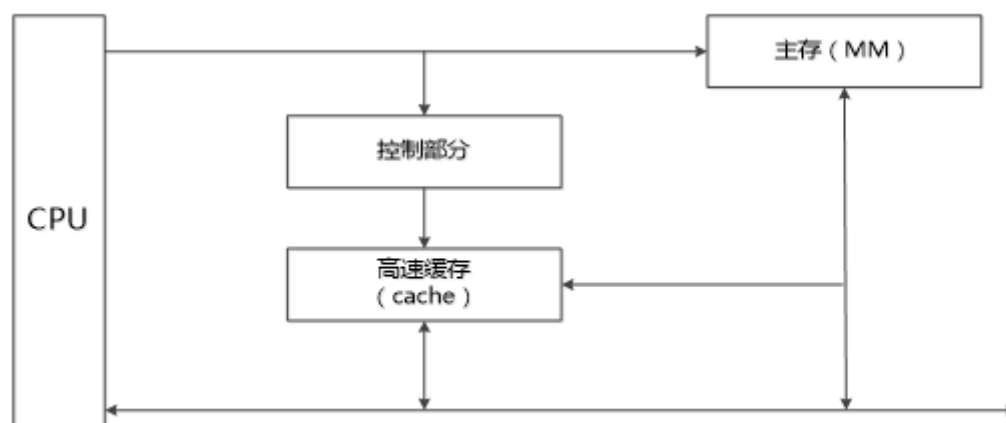
【答案】A

【解析】

高速缓存是用来存放当前最活跃的程序和数据的，作为主存局部域的副本，其特点是：

容量一般在几 KB 到几 MB 之间；速度一般比主存快 5 到 10 倍，由快速半导体存储器构成；其内容是主存局部域的副本，对程序员来说是透明的。

高速缓存的组成如下图所示：Cache 由两部分组成：控制部分和 cache 部分。Cache 部分用来存放主存的部分拷贝（副本）信息。控制部分的功能是：判断 CPU 要访问的信息是否在 cache 中，若在即为命中，若不在则没有命中。命中时直接对 cache 存储器寻址。未命中时，要按照替换原则，决定主存的一块信息放到 cache 的哪一块里面。



HTTPS 使用(7)协议对报文进行封装。

(7) A. SSH

B. SSL

C. SHA-1

D. SET

【答案】B

【解析】

HTTPS（全称：Hyper Text Transfer Protocol over Secure Socket Layer），是以安全为目标的 HTTP 通道，简单讲是 HTTP 的安全版。即 HTTP 下加入 SSL 层，HTTPS 的安全基础是 SSL，因此加密的详细内容就需要 SSL。

以下加密算法中适合对大量的明文消息进行加密传输的是(8)

(8) A. RSA

B. SHA-1

C. MD5

D. RC5

【答案】D

【解析】

加密技术中对称性加密技术的算法效率比较高，适合于大量数据的加密，在本题中属于对称性加密算法的只有 RC5。

(9) A、B 互换私钥
B. A、B 互换公钥
C. I1、I2 互换私钥
D. I1、I2 互换公钥

(12)A. 甲、乙作为共同申请人

- B. 甲或乙一方放弃权利并从另一方得到适当的补偿
- C. 甲、乙都不授予专利权
- D. 甲、乙都授予专利权

【答案】D

【解析】

软件专利权是指通过申请专利对软件的设计思想进行保护的一种方式,而非对软件本身进行的保护,我国在专利保护上,实行先申请制度,即谁申请在先,谁就享有该专利权。同时申请则协商归属,协商不成则同时驳回双方的专利申请。

数字语音的采样频率定义为 8kHz 这是因为(13)。

- (13)A. 语音信号定义的频率最高值为 4kHz
- B. 语音信号定义的频率最高值为 8kHz
- C. 数字语音传输线路的带宽只有 8kHz
- D. 一般声卡采样频率最高为每秒 8k 次

【答案】A

【解析】

音频数字化过程中采样频率应为声音最高频率 2 倍。本题采样频率为 8KHz, 所以其语音的最高频率应不超过 4KHz。

使用图像扫描仪以 300DPI 的分辨率扫描一幅 3x4 英寸的图片,可以得到(14)像素的数字图像。

- (14)A. 300X300 B. 300X400 C. 900X4 D. 900X1200

【答案】D

【解析】

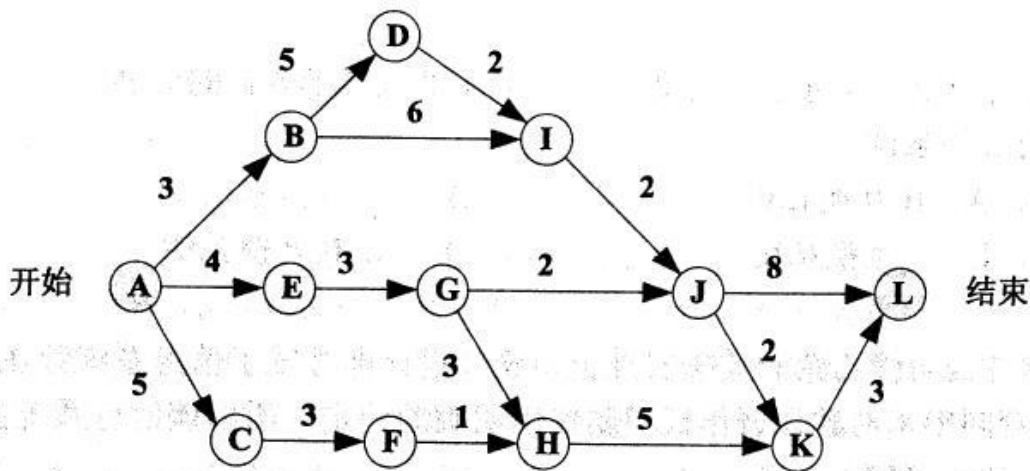
图像分辨率

是指组成一幅图像的像素密度;也是水平和垂直的像素表示;即用每英寸多少点(dpi)表示数字化图像的大小。

用 300dpi 来扫描一幅 3*4 英寸的彩色照片,那么得到一幅 900*1200 个像素点的图像

某软件项目的活动图如下图所示,其中顶点表示项目里程碑,连接顶点的边表示包含的

活动，边上的数字表示活动的持续时间(天)，则完成该项目的最少时间为(15)天。 活动 BD 和 HK 最早可以从第(16)天开始。(活动 AB 、AE 和 AC 最早从第 1 天开始)



- (15) A. 17 B. 18 C. 19 D. 20
- (16) A. 3 和 10 B. 4 和 11 C. 3 和 9 D. 4 和 10

【答案】D B

【解析】

关键路径：在活动图中时间跨度最长的路径。其决定项目最少工期。

项目关键路径是路径最长的一条，在该图中路径：A→B→D→I→J→L=3+5+2+2+8=20，是路径最长的一条。

有任务 A→B 需要 3 天，所以 B→D 应在第 4 天开始；由于 H 点最迟出的时间为 10 后（A→E→G→H），所以 H→K 应在第 11 天开始

在采用结构化开发方法进行软件开发时，设计阶段接口设计主要依据需求分析阶段的(17)。接口设计的任务主要是(18)。

- (17) A. 数据流图 B. E-R 图 C. 状态-迁移图 D. 加工规格说明
- (18) A. 定义软件的主要结构元素及其之间的关系
- B. 确定软件涉及的文件系统的结构及数据库的表结构
- C. 描述软件与外部环境之间的交互关系，软件内模块之间的调用关系
- D. 确定软件各个模块内部的算法和数据结构

【答案】A C

【解析】

(17) 软件设计必须依据对软件的需求来进行, 结构化分析的结果为结构化设计提供了最基本的输入信息。从分析到设计往往经历以下流程:

- (1) 研究、分析和审查数据流图。根据穿越系统边界的信息流初步确定系统与外部接口。
- (2) 根据数据流图决定问题的类型。数据处理问题通常有两种类型: 变换型和事务型。针对两种不同的类型分别进行分析处理。
- (3) 由数据流图推导出系统的初始结构图。
- (4) 利用一些启发式原则来改进系统的初始结构图, 直到得到符合要求的结构图为止。
- (5) 根据分析模型中的实体关系图和数据字典进行数据设计, 包括数据库设计或数据文件的设计。
- (6) 在设计的基础上, 依旧分析模型中的加工规格说明、状态转换图进行过程设计。

所以接口设计应该为需求分析阶段的数据流图, 即选项 A

(18) 接口设计的主要任务为: 描述软件与外部环境之间的交互关系, 软件内模块之间的调用关系

在进行软件开发时, 采用无主程序员的开发小组, 成员之间相互平等; 而主程序员负责制的开发小组, 由一个主程序员和若干成员组成, 成员之间没有沟通。在一个由 8 名开发人员构成的小组中, 无主程序员组和主程序员组的沟通路径分别是 (19)

- (19) A. 32 和 8 B. 32 和 7 C. 28 和 8 D. 28 和 7

【答案】D

【解析】

程序设计小组的组织形式一般有主程序员组, 无主程序员组和层次式程序员组。其中无主程序员组中的成员之间相互平等, 工作目标和决策都由全体成员民主讨论。对于项目规模较小、开发人员少、采用新技术和确定性较小的项目比较合适, 而对大规模项目不适宜采用。所以其沟通路径的数量为 $(8 \times 7) / 2 = 28$

而主程序员制则有主程序员负责决策。其他成员与主程序员沟通即可。所以其沟通路径数量为 $8 - 1 = 7$

在高级语言源程序员, 常需要用户定义的标识符程序员的对象命名, 常见的命名对象有 (20)。

- ① 关键字 (或保留字) ② 变量 ③ 函数 ④ 数据类型 ⑤ 注释

(20) A. ①②③

B. ②③④

C. ①③⑤

D. ②④⑤

【答案】A

【解析】

在编写程序时需要命名的对象有关键字，变量，函数。

在仅由字符 a、b 构成的所有字符串中，其中以 b 结尾的字符串集合可用正规式表示为 (21)。

(21) A. $(b|ab)^*b$

B. $(ab)^*b$

C. a^*b^*b

D. $(a|b)^*b$

【答案】D

【解析】

正规式 $(a|b)^*$ 对应的正规集为 $\{\epsilon, a, b, aa, ab, \dots, \text{所有由 } a \text{ 和 } b \text{ 组成的字符串}\}$ ，结尾为 b。

在以阶段划分的编译中，判断程序语句的形式是否正确属于 (22) 阶段的工作。

(22) A. 词法分析

B. 语法分析

C. 语义分析

D. 代码生成

【答案】B

【解析】

词法分析阶段

是编译过程的第一阶段，其任务是对源程序从前到后（从左到右）逐个字符扫描，从中识别出一个个“单词”符号。

词法分析过程的依据是语言的词法规则，即描述“单词”结构的规则。

语法分析阶段

其任务是在词法分析的基础上，根据语言的语法规则将单词符号序列分解成各类语法单位。

通常语法分析是确定整个输入串是否构成一个语法上正确的程序。

一般来说，通过编译的程序，不存在语法上的错误。

语义分析阶段

其任务主要检查源程序是否包含静态语义错误，并收集类型信息供后面的代码生成阶段使用。

语义分析的一个主要工作是进行类型分析和检查。

中间代码生成

其任务是根据语义分析的输出生成中间代码。

目标代码生成

是编译器工作的最后一个阶段。其任务是中间代码转换成特定机器上的绝对指令代码、可重定位的指令代码或汇编指令代码。本阶段与具体机器密切相关。

某计算机系统页面大小为 4K，进程的页面变换表如下所示。若进程的逻辑地址为 2D16H 。该地址经过变换后，其物理地址应为(23)

页号	物理块号
0	1
1	3
2	4
3	6

- (23) A. 2048H B. 4096H C. 4D16H D. 6D16H

【答案】C

【解析】

页面大小为 4K 说明，页内地址为 12 位。也就是逻辑地址中 2D16H 中的 D16H 在页内（十六进制转二进制，每位十六进制的数对应四位二进制），其 2 表示的是页号，逻辑页号为 2 对应的物理块号为 4。所以该逻辑地址对应的物理地址应该为 4D16H。

某系统中有 3 个并发进程竞争资源 R，每个进程都需要 5 个 R，那么至少有(24) 个 R，才能保证系统不会发生死锁。

- (24) A. 12 B. 13 C. 14 D. 15

【答案】B

【解析】

给每个进程分配其所需的最大资源数少一个资源（本题 3*4 个），如果还有一个资源剩余，则不会发生死锁。因为将这个剩余资源分配给任意一个进程，该进程就会得到满足运行，其运行后，将其所释放的资源再分配给其他进程，这样所有的进程都可以执行完成。

以下关于 C/S（客户机/服务器）体系结构的优点的叙述中，不正确的是(25)。

- (25) A. 允许合理的划分三层的功能，使之在逻辑上保持相对独立
B. 允许各层灵活地选用平台和软件
C. 各层可以选择不同的开发语言进行并行开发

D. 系统安装、修改和维护均只在服务器端进行

【答案】D

【解析】

C/S（客户机/服务器）体系结构由于在客户端需要安装相关的客户端软件，当客户端软件需要安装、修改和维护时，需要到每个客户端进行维护操作。

在设计软件的模块结构时，(26)不能改进设计质量。

(26) A. 尽量减少高扇出结构

B. 模块的大小适中

C. 将具有相似功能的模块合并

D. 完善模块的功能

【答案】C

【解析】

将具有相似功能的模块合并，会导致模块的聚合程度变低，可维护性下降。

在结构化设计中，系统由多个逻辑上相对独立的模块组成，在模块划分时需要遵循如下原则：

(1) 模块的大小要适中。系统分解时需要考虑模块的规模，过大的模块可能导致系统分解不充分，其内部可能包括不同类型的功能，需要进一步划分，尽量使得各个模块的功能单一；过小的模块将导致系统的复杂度增加，模块之间的调用过于频繁，反而降低了模块的独立性。一般来说，一个模块的大小使其实现代码在 1~2 页纸之内，或者其实现代码行数在 50~200 行之间，这种规模的模块易于实现和维护。

(2) 模块的扇入和扇出要合理。一个模块的扇出是指该模块直接调用的下级模块的个数；扇出大表示模块的复杂度高，需要控制和协调过多的下级模块。扇出过大一般是因为缺乏中间层次，应该适当增加中间层次的控制模块；扇出太小时可以把下级模块进一步分解成若干个子功能模块，或者合并到它的上级模块中去。一个模块的扇入是指直接调用该模块的上级模块的个数；扇入大表示模块的复用程度高。设计良好的软件结构通常顶层扇出比较大，中间扇出较少，底层模块则有大扇入。一般来说，系统的平均扇入和扇出系数为 3 或 4，不应该超过 7，否则会增大出错的概率。

(3) 深度和宽度适当。深度表示软件结构中模块的层数，如果层数过多，则应考虑是否有些模块设计过于简单，看能否适当合并。宽度是软件结构中同一个层次上的模块总数的最大值，一般说来，宽度越大系统越复杂，对宽度影响最大的因素是模块的扇出。在系统设计时，需要权衡系统的深度和宽度，尽量降低系统的复杂性，减少实施过程的难度，提高开发和维护的效率。

在面向对象方法中，多态指的是(27)

(27) A. 客户类无需知道所调用方法的特定子类的实现

B. 对象动态地修改类

C. 一个对象对应多张数据库表

D. 子类只能够覆盖父类中非抽象的方法

【答案】D

【解析】

不同类的对象对同一消息作出不同的响应就叫做多态

多态存在的三个条件

1、有继承关系

2、子类重写父类方法

3、父类引用指向子类对象

在数据库系统运行维护阶段，通过重建视图能够实现(28)

(28) A. 程序的逻辑独立性

B. 程序的物理独立性

C. 数据的逻辑独立性

D. 数据的物理独立性

【答案】C

【解析】

外模式/模式：保证了数据与程序的逻辑独立性，简称数据的逻辑独立性。

模式/内模式：保证了数据与应用程序的物理独立性，简称数据的物理独立性。

外模式对应关系数据库的视图。

数据库概念结构设计阶段是在(29)的基础上，依照用户需求对信息进行分类、聚集和概括，建立概念模型。

(29) A. 逻辑设计

B. 需求分析

C. 物理设计

D. 运行维护

【答案】B

【解析】

数据库的设计阶段分为四个阶段：需求分析阶段、概念结构设计阶段、逻辑结构设计阶段、和物理结构设计阶段。数据库概念结构设计阶段是在需求分析的基础上，依照用户需求

对信息进行分类、聚集和概括，建立概念模型

数据模型通常由(30)三要素构成。

- (30) A. 网状模型、关系模型、面向对象模型 B. 数据结构、网状模型、关系模型
C. 数据结构、数据操纵、关系模型 D. 数据结构、数据操纵、完整性约束

【答案】D

【解析】

数据模型的三要素

数据结构：是所研究的对象类型的集合，是对系统静态特性的描述。

数据操作：对数据库中各种对象（型）的实例（值）允许执行的操作的集合，包括操作及操作规则。是对系统动态特性的描述。

数据的约束：是一组完整性规则的集合。也就是说，对于具体的应用数据必须遵循特定的语义约束条件，以保证数据的正确、有效、相容。

给定关系模式 $R\langle U, F \rangle$ ，其中 U 为关系 R 的属性集， F 是 U 上的一组函数依赖， X 、 Y 、 Z 、 W 是 U 上的属性组。下列结论正确的是(31)。

- (31) A. 若 $wx \rightarrow y$ ， $y \rightarrow z$ 成立，则 $x \rightarrow z$ 成立
B. 若 $wx \rightarrow y$ ， $y \rightarrow z$ 成立，则 $w \rightarrow z$ 成立
C. 若 $x \rightarrow y$ ， $wy \rightarrow z$ 成立，则 $xw \rightarrow z$ 成立
D. 若 $x \rightarrow y$ ， $z \subseteq U$ 成立，则 $x \rightarrow yz$ 成立

【答案】C

【解析】

函数依赖的公理系统(Armstrong)

设关系模式 $R\langle U, F \rangle$ ， U 是关系模式 R 的属性全集， F 是关系模式 R 的一个函数依赖集。

对于 $R\langle U, F \rangle$ 来说有以下的：

自反律：若 $Y \subseteq X \subseteq U$ ，则 $X \rightarrow Y$ 为 F 所逻辑蕴含

增广律：若 $X \rightarrow Y$ 为 F 所逻辑蕴含，且 $Z \subseteq U$ ，则 $XZ \rightarrow YZ$ 为 F 所逻辑蕴含

传递律：若 $X \rightarrow Y$ 和 $Y \rightarrow Z$ 为 F 所逻辑蕴含，则 $X \rightarrow Z$ 为 F 所逻辑蕴含

合并规则：若 $X \rightarrow Y$ ， $X \rightarrow Z$ ，则 $X \rightarrow YZ$ 为 F 所蕴涵

伪传递率：若 $X \rightarrow Y$ ， $WY \rightarrow Z$ ，则 $XW \rightarrow Z$ 为 F 所蕴涵

分解规则：若 $X \rightarrow Y$ ， $Z \subseteq Y$ ，则 $X \rightarrow Z$ 为 F 所蕴涵

在关系 $R(A_1, A_2, A_3)$ 和 $S(A_2, A_3, A_4)$ 上进行

$\pi_{A_1, A_4}(\sigma_{A_2 < '2017' \wedge A_4 = '95'}(R \bowtie S))$ 关系运算，与该关系表达式等价的是 (32)

- (32) A. $\pi_{1,4}(\sigma_{2 < '2017' \vee 4 = '95'}(R \bowtie S))$
B. $\pi_{1,6}(\sigma_{2 < '2017'}(R) \times \sigma_{3 = '95'}(S))$
C. $\pi_{1,4}(\sigma_{2 < '2017'}(R) \times \sigma_{6 = '95'}(S))$
D. $\pi_{1,6}(\sigma_{2=4 \wedge 3=5}(\sigma_{2 < '2017'}(R) \times \sigma_{3 = '95'}(S)))$

【答案】D

【解析】

题干的关系代数运算的含义是 R 与 S 先进行自然连接运算，然后在自然连接的基础上进行选择运算，最后做投影运算。

自然连接运算，可以转化为 R 与 S 先进行笛卡儿积运算，在笛卡儿积运算的基础上，进行选择运算，选择运算的条件为：R.A2=S.A2 AND R.A3=S.A3，然后在选择运算的结果集上，进行投影运算，投影运算是消除重复的列。

将表达式综合起来，进行优化可以转换成选项 D 的表达式。

将该查询转换为等价的 SQL 语句如下：

SELECT DISTINCT A_1, A_4 FROM R, S WHERE $R.A_2 < '2017'$ (33) ;

- (33) A. OR $S.A_4 < '95'$ OR $R.A_2 = S.A_2$ OR $R.A_3 = S.A_3$
B. AND $S.A_4 < '95'$ OR $R.A_2 = S.A_2$ AND $R.A_3 = S.A_3$
C. AND $S.A_4 < '95'$ AND $R.A_2 = S.A_2$ AND $R.A_3 = S.A_3$
D. OR $S.A_4 < '95'$ AND $R.A_2 = S.A_2$ OR $R.A_3 = S.A_3$

【答案】C

【解析】

本题筛选条件 $A_2 < '2017'$ 已经给出，像关系连接的筛选条件 (R.A2=S.A2 AND R.A3=S.A3)

和 $A4 = '95'$ 的条件缺失，且这些条件应该是同时满足，应使用逻辑与运算。

给定关系模式 $R \langle U, F \rangle$ ， $U = \{A, B, C, D, E\}$ ， $F = \{B \rightarrow A, D \rightarrow A, A \rightarrow E, AC \rightarrow B\}$ ，则 R 的候选关键字为(34)，分解 $\rho = (R1 \langle ABCE \rangle, R2 \langle CD \rangle)$ (35)。

(34) A. CD B. ABD C. ACD D. ADE

(35) A. 具有无损连接性，且保持函数依赖

B. 不具有无损连接性，但保持函数依赖

C. 具有无损连接性，但不保持函数依赖

D. 不具有无损连接性，也不保持函数依赖

【答案】A D

【解析】

本题中由于 C 和 D 只出现在左边，必为候选码的成员。当选择属性 CD 时，由于 $D \rightarrow A$ ， $A \rightarrow E$ ；可以得出 $D \rightarrow AE$ ；由于 $D \rightarrow A$ ， $AC \rightarrow B$ 利用伪传递率得出 $CD \rightarrow B$ ；由于 $D \rightarrow AE$ 和 $CD \rightarrow B$ 利用增广率和合并率得出 $CD \rightarrow ABCDE$ 。因此 CD 属性为候选码。

利用无损连接性的判断定理：不存在 $R1 \cap R2 \rightarrow R1 - R2$ 或 $R1R2 \rightarrow R2 - R1$ 被 F 逻辑蕴含的情况，所以分解不具有无损连接性；同时 $F1 \cup F2 \neq F$ ，所以分解也不保持函数依赖。

并发执行的三个事务 $T1$ 、 $T2$ 和 $T3$ ，事务 $T1$ 对数据 $D1$ 加了共享锁，事务 $T2$ 、 $T3$ 分别对数据 $D2$ 、 $D3$ 加了排它锁，之后事务 $T1$ 对数据(36)；事务 $T2$ 对数据(37)。

(36) A. $D2$ 、 $D3$ 加排它锁都成功 B. $D2$ 、 $D3$ 加共享锁都成功

C. $D2$ 加共享锁成功， $D3$ 加排它锁失败 D. $D2$ 、 $D3$ 加排它锁和共享锁都失败

(37) A. $D1$ 、 $D3$ 加共享锁都失败

B. $D1$ 、 $D3$ 加共享锁都成功

C. $D1$ 加共享锁成功， $D3$ 加排它锁失败

D. $D1$ 加排它锁成功， $D3$ 加共享锁失败

【答案】D C

【解析】

并发事务如果对数据读写时不加以控制，会破坏事务的隔离性和一致性。控制的手段就是加锁，在事务执行时限制其他事务对数据的读取。在并发控制中引入两种锁：排它锁

(Exclusive Locks, 简称 X 锁)和共享锁(Share Locks, 简称 S 锁)。

排它锁又称为写锁,用于对数据进行写操作时进行锁定。如果事务 T 对数据 A 加上 X 锁后,就只允许事务 T 读取和修改数据 A , 其他事务对数据 A 不能再加任何锁,从而也不能读取和修改数据 A , 直到事务 T 释放 A 上的锁。

共享锁又称为读锁,用于对数据进行读操作时进行锁定。如果事务 T 对数据 A 加上了 S 锁后,事务 T 就只能读数据 A 但不可以修改,其他事务可以再对数据 A 加 S 锁来读取,只要数据 A 上有 S 锁,任何事务都只能再对其加 S 锁读取而不能加 X 锁修改。

数据库概念结构设计阶段的工作步骤依次为(38)。

- (38)A. 设计局部视图→抽象→修改重构消除冗余→合并取消冲突
B. 设计局部视图→抽象→合并取消冲突→修改重构消除冗余
C. 抽象→设计局部视图→修改重构消除冗余→合并取消冲突
D. 抽象→设计局部视图→合并取消冲突→修改重构消除冗余

【答案】D

【解析】

先划分好各个局部应用之后,使用抽象机制,确定局部应用中的实体、实体的属性、实体的标识符及实体间的联系及其类型,然后绘制局部 E-R 图,根据局部应用设计好各局部 E-R 图之后,就可以对各分 E-R 图进行合并。在合并过程中解决分 E-R 图中相互间存在的冲突,消除分 E-R 图之间存在的信息冗余使之成为能够被全系统所有用户共同理解和接受的统一的、精炼的全局概念模型。

在数据传输过程中,为了防止被窃取可以通过(39)来实现的。

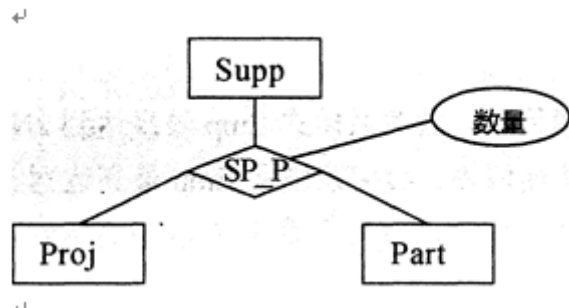
- (39)A. 用户标识与鉴别 B. 存取控制 C. 存取控制 D. 审计

【答案】C

【解析】

使用数据加密技术,可以保障数据在传输过程是机密的。

在某企业的工程项目管理数据库中供应商关系 Supp、项目关系 Proj 和零件关系 Part 的 E-R 模型和关系模式如下



Supp (供应商号, 供应商名, 地址, 电话) //供应商号唯一标识 Supp 中的每一个元组

Proj (项目号, 项目名, 负责人, 电话) //项目号唯一标识 Proj 中的每一个元组

Part (零件号, 零件名) //零件号唯一标识 Part 中的每一个元组

其中,每个供应商可以为多个项目供应多种零件,每个项目可以由多个供应商供应多种零件,每种零件可以由多个供应商供应给多个项目。SP_P 的联系类型为(40), (41)。

(40) A. *:*:~ B. 1:~ C. 1:1:~ D. 1:1:1

(41) A. 不需要生成一个独立的关系模式

B. 需要生成一个独立的关系模式, 该模式的主键为(项目号, 零件号, 数量)

C. 需要生成一个独立的关系模式, 该模式的主键为(供应商号, 数量)

D. 需要生成一个独立关系模式, 该模式的主键为(供应商号, 项目号, 零件号)

【答案】A D

【解析】

题干中:“每个供应商可以为多个项目供应多种零件, 每个项目可以由多个供应商供应多种零件, 每种零件可以由多个供应商供应给多个项目”, 说明三个实体间的联系类型应为: 多对多对多。

对于多对多的联系在转关系时, 应该转为一个独立的关系模式, 该关系的主键, 应为多方实体码的属性组成。

给定关系模式 SP_P (供应商号, 项目号, 零件号, 数量), 查询至少给 3 个(包含 3 个)不同项目供应了零件的供应商, 要求输出供应商号和供应零件数量的总和, 并按供应商号降序排列。

SELECT 供应商号, SUM (数量) FROM SP_P (42) (43) (44)。

(42) A. ORDERBY 供应商号

B. GROUPBY 供应商号

C. ORDERBY 供应商号 ASC

D. GROUPBY 供应商号 DESC

(43) A. WHERE 项目号 >2

B. WHERE COUNT(项目号)>2

C. HAVING (DISTINCT 项目号)> 2

D. HAVING COUNT(DISTINCT 项目号)>2

(44) A. ORPERBY 供应商号

B. GROUPBY 供应商号

C. ORDERBY 供应商号 DESC

D. GROUP BY 供应商号 DESC

【答案】C D C

【解析】

SELECT 语句的基本语法结构:

SELECT [ALL | DISTINCT] <列名>[, . . . n]

FROM <表名|视图名> [, . . . n]

[WHERE <条件表达式>]

[GROUP BY <列名> [HAVNG <条件表达式>]]

[ORDER BY <列名>[ASC|DESC] [, . . . n]]

本题中,需要进行分组,分组的依据为供应商号;同时在分组的基础上需要指定条件,这时需使用 HAVING 子句,统计项目的个数大于,由于项目可能重复,因此在统计之前应该消除重复的项目,需使用 DISTINCT 关键字。

题干要求按供应商号进行降序排列,需使用 ORDER BY 子句和关键字 DESC

某企业的信息系统管理数据库中的员工关系模式为 Emp (员工号, 姓名, 部门号, 岗位, 联系地址, 薪资), 函数依赖集 $F=\{\text{员工号} \rightarrow (\text{姓名}, \text{部门号}, \text{岗位}, \text{联系地址}), \text{岗位} \rightarrow \text{薪资}\}$ 。Emp 关系的主键为(45), 函数依赖集 F(46)。

(45) A. 员工号, Emp 存在冗余以及插入异常和删除异常的问题

B. 员工号, Emp 不存在冗余以及插入异常和删除异常的问题

C. (员工号, 岗位), Emp 存在冗余以及插入异常和删除异常的问题

D. (员工号, 岗位), Emp 不存在冗余以及插入异常和删除异常的问题

(46) A. 存在传递依赖, 故关系模式 Emp 最高达到 1NF

B. 存在传递依赖, 故关系模式 Emp 最高达到 2NF

C. 不存在传递依赖, 故关系模式 Emp 最高达到 3NF

D. 不存在传递依赖, 故关系模式 Emp 最高达到 4NF

【答案】A B

【解析】

由于员工号 \rightarrow (姓名, 部门号, 岗位, 联系地址), 岗位 \rightarrow 薪资, 利用传递率可以得出员工号 \rightarrow (姓名, 部门号, 岗位, 联系地址, 薪资), 所以该关系的主码应该为员工号, 由于存在传递函数依赖, 所以不满足 3NF 的要求。

满足 BCNF 范式的关系(47)。

- (47) A. 允许属性对主键的部分依赖 B. 能够保证关系的实体完整性
C. 没有传递函数依赖 D. 可包含组合属性

【答案】C

【解析】

若关系模式 $R \in 1NF$, 若 $X \rightarrow Y$ 且 $Y \not\subseteq X$ 时, X 必含有码, 则关系模式 R 属于第 BC 范式, 记为: $R \in BCNF$

BCNF 是在 3NF 的基础要求消除键属性对码的部分和传递依赖。

数据的物理存储结构, 对于程序员来讲, (48)。

- (48) A. 采用数据库方式管理数据是可见的, 采用文件方式管理数据是不可见的
B. 采用数据库方式管理数据是不可见的, 采用文件方式管理数据是可见
C. 采用数据障方式管理数据建可见的, 采用文件方式管理数据是可见的
D. 采用数据库方式管理数据是不可见的, 采用文件方式管理数据是不可见的

【答案】B

【解析】

通过 DBMS 管理数据有较高的数据独立性, 数据独立性是指数据与程序独立, 将数据的定义从程序中分离出去, 由 DBMS 负责数据的存储, 应用程序关心的只是数据的逻辑结构, 无须了解数据在磁盘上的数据库中的存储形式, 从而简化应用程序, 大大减少应用程序编制的工作量。如果采用文件方式管理数据, 应用程序得明确数据的定义等操作, 也就是说程序员需要操作文件中的数据。

在 SQL 中, 用户(49)获取权限。

- (49) A. 只能通过数据库管理员授权 B. 可通过对象的所有者执行 GRANT 语句
C. 可通过自己执行 GRANT 语句 D. 可由任意用户授权

【答案】B

【解析】

在数据库中用户可以通过对象的所有者、拥有授予相关权限的权限的用户或者 DBA 执行 GRANT 语句获取对应的权限。

保证并发调度的可串行化，是为了确保事务的(50)。

- | | |
|-----------------|------------|
| (50) A. 原子性和一致性 | B. 原子性和持久性 |
| C. 隔离性和持久性 | D. 隔离性和一致性 |

【答案】D

【解析】

原子性：事务是原子的，要么做，要么都不做。

一致性：事务执行的结果必须保证数据库从一个一致性状态变到另一个一致性状态。

隔离性：事务相互隔离。当多个事务并发执行时，任一事务的更新操作直到其成功提交的整个过程，对其它事物都是不可见的。

持久性：一旦事务成功提交，即使数据库崩溃，其对数据库的更新操作也永久有效。

串行调度：多个事务依次串行执行，且只有当一个事务的所有操作都执行完后才执行另一个事务的所有操作

可串行化保证了事务并行调度时，相互不破坏，同时保证了数据从一个一致性状态到另一个一致性状态。

满足两阶段封锁协议的调度一定是(51)。

- | | |
|----------------|---------------|
| (51) A. 无死锁的调度 | B. 可串行化调度 |
| C. 可恢复调度 | D. 可避免级联回滚的调度 |

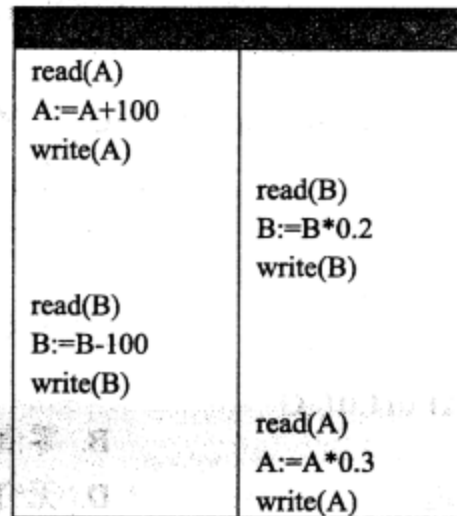
【答案】B

【解析】

两段锁协议是：对任何数据进行读写之前必须对该数据加锁，在释放了一个封锁之后，事务不再申请和获得任何其他封锁。这就缩短了持锁时间，提高了并发性，同时解决了数据的不一致性。

两段封锁协议可以保证可串行化，它把每个事务分解为加锁和解锁两段。

下图中两个事务的调度属于(52)。



(52) A. 可串行化调度

B. 串行调度

C. 非可串行化调度

D. 产生死锁的调度

【答案】C

【解析】

串行调度：非交错地依次执行给定事务集合中的每一个事务的全部动作。

可串行化，是指一个调度对数据库的状态的影响和某个串行调度相同，称为该调度具有可串行性。

以下对数据库故障的描述中，不正确的是(53)。

(53) A. 系统故障指软硬件错误导致的系统崩溃

B. 由于事务内部的逻辑错误造成该事务无法执行的故障属于事务故障

C. 可通过数据的异地备份来减少磁盘故障可能给数据库系统造成数据丢失

D. 系统故障一定会导致磁盘数据丢失

【答案】D

【解析】

数据库故障

主要分：事务故障、系统故障和介质故障

事务故障

是指事务在运行至正常终点前被终止，此时数据库可能出现不正确的状态

由于事务程序内部错误而引起的，有些可以预期，如金额不足等；有些不可以预期，如非法输入、运算溢出等

恢复过程

- ①反向（从后向前）扫描日志文件，查找该事务的更新操作
- ②对该事务的更新操作执行逆操作，也就是将日志记录更新前的值写入数据库。
- ③继续反向扫描日志文件，查找该事务的其他更新操作，并作同样处理。
- ④如此处理下去，直到读到了此事务的开始标记，事务故障恢复就完成了。

事务故障的恢复由系统自动完成，对用户是透明的

系统故障（通常称为软故障）

是指造成系统停止运转的任何事件，使得系统要重新启动

特定类型的硬件错误、操作系统故障、DBMS 代码错误、突然停电等

恢复过程

- ①正向（从头到尾）扫描日志文件，找出故障发生前已经提交的事务（这些事务既有 BEGIN TRANSACTION 记录，也有 COMMIT 记录），将其事务标识记入重做（REDO）队列。同时找出故障发生时尚未完成的事务（这些事务只有 BEGIN TRANSACTION 记录，无相应的 COMMIT 记录），将其事务标识记入撤销（UNDO）队列
- ②反向扫描日志文件，对每个 UNDO 事务的更新操作执行逆操作，也就是将日志记录中更新前的值写入数据库。
- ③正向扫描日志文件，对每个 REDO 事务重新执行日志文件登记的操作，也就是将日志记录中更新后的值写入数据库。

是在系统重启之后自动执行的。

介质故障（称为硬件故障）

是指外存故障，例如磁盘损坏、磁头碰撞，瞬时强磁场干扰等

这类故障将破坏数据库或部分数据库，并影响正在存取这部分数据的所有事务，日志文件也被破坏

恢复过程

- ①装入最新的数据库后备副本，使数据库恢复到最近一次转储时的一致性状态。
- ②转入相应的日志文件副本，重做已完成的事务

介质故障的恢复需要 DBA 的介入，具体的恢复操作仍由 DBMS 完成。

恢复过程

①DBA 只需要重装最近转储的数据库副本和有关的各日志文件副本。

②然后执行系统提供的恢复命令。

有两个关系模式 $R(A, B, C, D)$ 和 $S(A, C, E, G)$ ，则 $X=R \times S$ 的关系模式是(54)。

- (54) A. X(A, B, C, D, E, G) B. X(A, B, C, D)
- C. X(R.A, B, R.C, D, S.A, S.C, E, G) D. X(B, D, E, G)

【答案】C

【解析】

R 与 S 的笛卡儿积应该形成 M+N 元的关系，其中 M 表示来自关系 R 的列，N 表示来自关系 S 列，如果列名存在重复的情况，则需要带上关系名，表示该列来自哪个关系如：R.A

给定关系模式 $R\langle U, F \rangle$ ，其中属性集 $U=\{A, B, C, D, E, G, H\}$ 函数依赖集 $F=\{A \rightarrow B, AE \rightarrow H, BG \rightarrow DC, E \rightarrow C, H \rightarrow E\}$ ，下列函数依赖不成立的是 (55)

- (55) A. $A \rightarrow AB$ B. $H \rightarrow C$ C. $AEB \rightarrow C$ D. $A \rightarrow BH$

【答案】D

【解析】

由于 $A \rightarrow B$, 在加上 A 自身函数决定 A , 利用合并率, 得出 $A \rightarrow AB$;

由于 $H \rightarrow E$, $E \rightarrow C$, 利用传递率, 得出 $H \rightarrow C$:

由于 $E \rightarrow C$, 利用增广率和分解率, 得出 $ABE \rightarrow C$;

函数依赖的公理系统(Armstrong)

设关系模式 $R\langle U, F \rangle$ ， U 是关系模式 R 的属性全集， F 是关系模式 R 的一个函数依赖集。

对于 $R \langle U, F \rangle$ 来说有以下的:

自反律: 若 $Y \subseteq X \subseteq U$, 则 $X \rightarrow Y$ 为 F 所逻辑蕴含

增广律: 若 $X \rightarrow Y$ 为 F 所逻辑蕴含, 且 $Z \subseteq U$, 则 $XZ \rightarrow YZ$ 为 F 所逻辑蕴含

传递律: 若 $X \rightarrow Y$ 和 $Y \rightarrow Z$ 为 F 所逻辑蕴含, 则 $X \rightarrow Z$ 为 F 所逻辑蕴含

合并规则: 若 $X \rightarrow Y, X \rightarrow Z$, 则 $X \rightarrow YZ$ 为 F 所蕴涵

伪传递率：若 $X \rightarrow Y$, $WY \rightarrow Z$, 则 $XW \rightarrow Z$ 为 F 所蕴涵

分解规则：若 $X \rightarrow Y$, $Z \subseteq Y$, 则 $X \rightarrow Z$ 为 F 所蕴涵

在日志中加入检查点，可(56)。

(56) A. 减少并发冲突

B. 提高一并故障恢复的效率

C. 避免级联回滚

D. 避免死锁

【答案】B

【解析】

检查点将脏数据页从当前数据库的缓冲区高速缓存刷新到磁盘上。这最大限度地减少了数据库完整恢复时必须处理的活动日志部分。

某销售公司需开发数据库应用系统管理客户的商品购买信息。该系统需记录客户的姓名、出生日期、年龄和身份证号信息，记录客户每次购买的商品名称和购买时间等信息。如果在设计时将出生日期和年龄都设定为客户实体的属性，则年龄属于(57)，数据库中购买记录表中每条购买记录对应的客户必须在客户表中存在，这个约束属于(58)。

(57) A. 派生属性

B. 多值属性

C. 主属性

D. 复合属性

(58) A. 参与约束

B. 参照完整性约束

C. 映射约束

D. 主键约束

【答案】A B

【解析】

简单属性：属性是原子的、不可再分的。

复合属性：可以细分为更小的部分。例如：职工实体集的通信地址。

单值属性：一个属性对应一个值。

多值属性：一个属性对应多个值。例如：职工实体集的职工的亲属姓名。

NULL 属性：表示无意义或不知道（属性没有值或属性值未知时）。

派生属性：可以从其它属性得来。例如：工龄可以从入职时间计算得出。

本题中年龄可以通过出生日期和系统时间计算出来。属于派生属性

实体完整性：规定基本关系 R 的主属性 A 不能取空。

用户自定义完整性：就是针对某一具体关系数据库的约束条件，反映某一具体应用所涉及的数据必须满足的语义要求，由应用的环境决定。如：年龄必须为大于 0 小于 150 的整数。

参照完整性/引用完整性：规定，若 F 是基本关系 R 的外码，它与基本关系 S 的主码 K，相对应（基本关系 R 和 S 不一定是不同的关系），则 R 中每个元组在 F 上的值必须为：或者取空值；或者等于 S 中某个元组的主码值。

NULL 值在数据库中表示(59)，逻辑运算 UNKNOWN OR TRUE 的结果是(60)。

(59) A. 空集 B. 零值 C. 不存在或不知道 D. 无穷大

(60) A. NULL B. UNKNOWN C. TRUE D. FALSE

【答案】C C

【解析】

NULL 属性：表示无意义或不知道（属性没有值或属性值未知时）。

逻辑运算 UNKNOWN OR TRUE 由于是逻辑或运算，OR 之前非布尔值，结果为 FALSE，OR 之后为 TRUE，所以逻辑运算的结果为 TRUE。

CAP 理论是 NoSql 理论的基础，下列性质不属于 CAP 的是(61)。

(61) A. 分区容错性 B. 原子性 C. 可用性 D. 一致性

【答案】B

【解析】

CAP 理论

CAP 简单来说：就是对一个分布式系统，一致性(Consistency)、可用性(Availability)和分区容忍性(Partition tolerance)三个特点最多只能三选二。

以下是平行数据库的四种体系结构，在(62)体系结构中 所有处理器共享一个公共的主存储器 and 磁盘。

(62) A. 共享内存 B. 共享磁盘 C. 无共享 D. 层次

【答案】A

【解析】

并行数据库体系结构

并行数据库要求尽可能的并行执行所有的数据库操作，从而在整体上提高数据库系统的性能。

根据所在的计算机的处理器 (Processor)、内存 (Memory) 及存储设备 (Storage) 的相互关系，并行数据库可以归纳为三种基本的体系结构(这也是并行计算的三种基本体系结构)，

即：

1. 共享内存结构 (Shared-Memory)、
2. 共享磁盘结构 (Shared-Disk)
3. 无共享资源结构 (Shared-Nothing)。

1、共享内存 (Shared-Memory) 结构

该结构包括多个处理器、一个全局共享的内存（主存储器）和多个磁盘存储，各个处理器通过高速通讯网络（InterconnectionNetwork）与共享内存连接，并均可直接访问系统中的一个、多个或全部的磁盘存储，在系统中，所有的内存和磁盘存储均由多个处理器共享。

（1）提供多个数据库服务的处理器通过全局共享内存来交换消息和数据，通讯效率很高，查询内部和查询间的并行性的实现也均不需要额外的开销；

（2）数据库中的数据存储在多个磁盘存储上，并可以为所有处理器访问；

（3）在数据库软件的编制方面与单处理机的情形区别也不大。

这种结构由于使用了共享的内存，所以可以基于系统的实际负荷来动态地给系统中的各个处理器分配任务，从而可以很好地实现负荷均衡。

2、共享磁盘（Shared-Disk）结构

该结构由多个具有独立内存（主存储器）的处理器和多个磁盘存储构成，各个处理器相互之间没有任何直接的信息和数据的交换，多个处理器和磁盘存储由高速通信网络连接，每个处理器都可以读写全部的磁盘存储。

这种结构常用于实现数据库集群，硬件成本低、可扩充性好、可用性强，且可很容易地从单处理器系统迁移，还可以容易地在多个处理器之间实现负载均衡。

3、无共享资源（Shared-Nothing）结构

该结构由多个完全独立的处理节点构成，每个处理节点具有自己独立的处理器、独立的内存（主存储器）和独立的磁盘存储，多个处理节点在处理器级由高速通信网络连接，系统中的各个处理器使用自己的内存独立地处理自己的数据。

这种结构中，每一个处理节点就是一个小型的数据库系统，多个节点一起构成整个的分布式的并行数据库系统。由于每个处理器使用自己的资源处理自己的数据，不存在内存和磁盘的争用，提高的整体性能。另外这种结构具有优良的可扩展性——只需增加额外的处理节点，就可以以接近线性的比例增加系统的处理能力。

这种结构中，由于数据是各个处理器私有的，因此系统中数据的分布就需要特殊的处理，以尽量保证系统中各个节点的负载基本平衡，但在目前的数据库领域，这个数据分布问题已经有比较合理的解决方案。

由于数据是分布在各个处理节点上的，因此，使用这种结构的并行数据库系统，在扩展时不可避免地会导致数据在整个系统范围内的重分布（Re-Distribution）问题。

数据仓库中的数据组织是基于(63)模型的。

- (63) A. 网状 B. 层次 C. 关系 D. 多维

【答案】D

【解析】

数据仓库是面向主题的；操作型数据库的数据组织面向事务处理任务，而数据仓库中的数据是按照一定的主题域进行组织。主题是指用户使用数据仓库进行决策时所关心的重点方面，一个主题通常与多个操作型信息系统相关。

主题是与传统数据库的面向应用相对应的，是一个抽象概念，是在较高层次上将企业信息系统中的数据综合、归类并进行分析利用的抽象。每一个主题对应一个宏观的分析领域。数据仓库排除对于决策无用的数据，提供特定主题的简明视图。

因此数据通常是多维数据，包括维属性和量度属性。即数据仓库中的数据组织是基于多维模型的。

数据挖掘中分类的典型应用不包括(64)。(65)可以用于数据挖掘的分类任务。

- (64) A. 识别社交网络中的社团结构，即连接稠密的子网络
B. 根据现有的客户信息，分析潜在客户
C. 分析数据，以确定哪些贷款申请是安全的，哪些是有风险的
D. 根据以往病人的特征，对新来的病人进行诊断

- (65) A. EM B. Apriori C. K-means D. SVM

【答案】A D

【解析】

分类(classification)：有指导的类别划分，在若干先验标准的指导下进行，效果好坏取决于标准选取的好坏。它找出描述并区分数据类或概念的模型(或函数)，以便能够使用模型预测类标记未知的对象类。分类分析在数据挖掘中是一项比较重要的任务，目前在商业上应用最多。

识别社交网络中的社团结构，即连接稠密的子网络一般采用社区分析算法 CNM。

在浏览器地址栏输入一个正确的网址后，本地主机将首先在(66)中查询该网址对应的 IP 地址。

- (66) A. 本地 DNS 缓存 B. 本机 hosts 文件
C. 本地 DNS 服务器 D. 根域名服务器

【答案】A

【解析】

本地主机进行 DNS 解析的时候，先查询本地的 DNS 缓存，如果没有再查询本地 DNS 服务器，如果没有再由本地 DNS 服务器进行迭代查询后，将查询结果回返给客户机。

下面关于 Linux 目录的描述中，正确的是(67)。

- (67)A. Linux 只有一个根目录，用“/root”表示
B. Linux 中有多个根目录，用“/”加相应目录名称表示
C. Linux 中只有一个根目录，用“/”表示
D. Linux 中有多个根目录，用相应目录名称表示

【答案】C

【解析】

Linux 文件系统只有一个根目录，使用“/”表示。

以下 IP 地址中，属于网络 10.110.12.29/255.255.255.224 的主机 IP 是(68)。

- (68)A. 10.110.12.0 B. 10.110.12.30 C. 10.110.12.31 D. 10.110.12.32

【答案】B

【解析】

子网掩码为 255.255.255.224，说明 IP 地址中有 27 位表示网络位，剩下 5 位表示主机位，5 位表示主机位，即每个子网一共有 $2^5-2=30$ 个可用 IP 地址，而本题的 IP 地址的网络号为：10.110.12.0，该网络中的可用 IP 地址范围是 10.110.12.0~10.110.12.31，其中 10.110.12.31 表示子网广播地址。

0000 1010 . 0110 1110 . 0000 1100 . 0001 1101	← 10.110.12.29
1111 1111 . 1111 1111 . 1111 1111 . 1110 0000	← 255.255.255.224
0000 1010 . 0110 1110 . 0000 1100 . 0000 0000	← 10.110.12.0
0000 1010 . 0110 1110 . 0000 1100 . 0001 1111	← 10.110.12.31

在异步通信中，每个字符包含 1 位起始位、7 位数据位和 2 位终止位，若每秒钟传送 500 个字符，则每秒有效数据速率为(69)。

- (69)A. 500b/s B. 700b/s C. 3500b/s D. 5000b/s

【答案】C

【解析】

由于每个字符能传送 7 为有效数据位，每秒能传送有效数据位为：500*7b/s=3500b/s

以下路由策略中，依据网络信息经常更新路由的是(70)。

(70)A. 静态路由 B. 洪泛式 C. 随机路由 D. 自适应路由

【答案】D

【解析】

静态路由路由信息是不进行路由信息更新的；

动态路由选择算法就是自适应路由选择算法，是依靠当前网络的状态信息进行决策，从而使路由选择结果在一定程度上适应网络拓扑结构和通信量的变化，需要依据网络信息经常更新路由。

随机路由使用前向代理来收集网络中的有限全局信息即当前结点至其源结点的旅行时间，并以此来更新结点的旅行时间表；算法根据结点旅行时间表所记录的历史信息和当前的链路状态来共同确定一个邻结点的路由质量，并以此为参考随机路由分组来均衡网络负载。洪泛（mflood）路由算法是一个简单有效的路由算法，其基本思想是每个节点都是用广播转发收到的数据分组，若收到重复分组则进行丢弃处理。

The beauty of software is in its function, in its internal structure, and in the way in which it is created by a team. To a user, a program with just the right features presented through an intuitive and (71) interface is beautiful. To a software designer, an internal structure that is partitioned in a simple and intuitive manner, and that minimizes internal coupling is beautiful. To developers and managers, a motivated team of developers making significant progress every week, and producing defect-free code, is beautiful. There is beauty on all these levels.

Our world needs software -- 10 years of software was something

that ran in a few big and expensive machines. 10 years ago it was something that ran in most companies and industrial settings. Now there is software running in our cell phones, watches, appliances, automobiles, toys, and tools. And need for new and better software never (72) As our civilization grows and expands, as developing

nations build their masuctures, as devel0ped nations strive@achieve ever greater efficiencies, the need for more and more software (73)to increase. It would be a great shame if, in all that sware, there was no beauty.

We know that sware can be uglWe know that it can be hard to use, unreliable , and carelessly structured. We know that there are software systems whose tangled and careless intemal structures make them expensive and difficult to change. We know that there are sware systems that present their features through an awkward and cumbersome interface.We know that there are software systems that crash and misbehave. These are (74) systems. Unfortunately , as a profession , sware developers tend to create more ugly systems than beautiful ones.

There is a secret at the best sofual rare developers know. Beauty is cheaper than ugliness. Beauty is faster than ugliness. A beautiful sware system can be built and maintained in less time, and for less money, than an ugly one. Novice sware developers don't understand is. They think that they have to do everything fast and quick.They think that beauty is (75). No! By doing things fast and quick, they make messes that make the software stiff and hard to understand. Beautiful systems are flexible and easy to understand. Building them and maintaining them is a joy. It is ugliness that is impractical.Ugliness will slow you down and make your software expensive and brittle. Beautiful systems cost the least to build and maintain , and are delivered soonest.

- | | | | |
|--------------------|--------------|-----------------|---------------|
| (71)A. simple | B. Hard | C. complex | D. duplicated |
| (72)A. happens | B. Exists | C. stops | D. starts |
| (73)A. starts | B. continues | C. appears | D. Stops |
| (74)A. practical | B. useful | C. beautiful | D. Ugly |
| (75)A. impractical | B. perfect | C. time-wasting | D. practical |

【答案】A A C D C

【解析】

软件的优点在于其功能，内部结构以及由团队创建的方式。对于用户来说，通过直观和____界面呈现的正确功能的程序是美丽的。对于软件设计师来说，分割的内部结构是一种简单而直观的方式，最小化内部耦合是美观的。对于开发人员和经理来说，一个积极的开发团

队每周都取得重大进展，并且生产无缺陷的代码是美丽的。所有这些级别都有美丽。

我们的世界需要大量软件。五十年前，软件是在大多数公司和工业环境中运行的。现在软件存在在我们的手机，手表，电器，汽车，玩具和工具中。并且对新的和更好的软件的需求永远不会____。随着我们文明的发展和壮大，随着发展中国家建设基础设施，发达国家努力实现更高的效率，越来越多的软件需求____增长。如果在所有的软件中没有美丽的话，这将是一个很大的耻辱。

我们知道软件可能是丑的。我们知道它可能很难使用，不可靠，粗心大意的结构。我们知道有一些软件系统的纠结和粗心的内部结构使得它们变得昂贵和难以改变。我们知道有一些软件系统通过尴尬和繁琐的界面来呈现其功能。我们知道有软件系统崩溃和行为不端。这些都是____系统。不幸的是，作为一个专业，软件开发人员倾向于创建丑陋的系统比美丽的系统更多。

这是最好的软件开发者的秘密。美丽的比丑陋的更便宜。美丽的比丑陋的更快。一个美丽的软件系统相当于一个丑陋的系统来说，建立和维护要花的时间与金钱会少得多。很多新手软件开发人员不明白这一点。他们认为做每一个事情必须快速，更快速。他们认为美是____。没有！通过快速，快速地做事情，他们使软件变得僵硬，难以理解。美观的系统灵活易懂。建立和维护他们是一种快乐。丑陋是不切实际的。丑陋会减慢你的速度，会使你的软件昂贵而脆弱。美观的系统成本最低，建立和维护成本最低，交货时间最短。

71 A simple (简单) B hard (困难) C complex (复杂) D duplicated (被复制)

72 A happens (发生) B exists (存在) C stops (停止) D starts (开始)

73 A starts (开始) B continues (持续) C appears (出现) D stops (停止)

74 A practical (实用的) B useful (有用的) C beautiful (美丽的) D ugly (丑陋的)

75 A impractical (不实用的) B perfect (完美的) C time-wasting (浪费时间) D practical (实用的)

试题一

阅读下列说明和图，回答问题 1 至问题 4，将解答填入答题纸的对应栏内。

【说明】

某医疗器械公司作为复杂医疗产品的集成商，必须保持高质量部件的及时供应。为了实现这一目标，该公司欲开发一采购系统。系统的主要功能如下：

1. 检查库存水平。采购部门每天检查部件库存量，当特定部件的库存量降至其订货店时，返回低存量部件及库存量。

2. 下达采购订单。采购部门针对低存量部件及库存量提交采购请求，向其供应商(通过供应商文件访问供应商数据)下达采购订单，并存储于采购订单文件中。

3. 交运部件。当供应商提交提单并交运部件时，运输和接收(S/R)部门通过执行以下三步过程接收货物：

(1) 验证装运部件。通过访问采购订单并将其与提单进行比较来验证装运的部件，并将提单信息发给 S/R 职员。如果收货部件项目出现在采购订单和提单上，则已验证的提单和收货部件项目将被送去检验。否则 S /R 职员提交的装运错误信息生成装运错误通知发送给供应商。

(2) 检验部件质量。通过访问质量标准来检查装运部件的质量，并将已验证的提单发给检验员。如果部件满足所有质量标准，则将其添加到接受的部件列表用于更新部件库存。如果部件未通过检查，则将检验员创建的缺陷装运信息生成缺陷装运通知发送给供应商。

(3) 更新部件库存。库管员根据收到的接受的部件列表添加本次采购数量，与原有库存量累加来更新库存部件中的库存量。标记订单采购完成。

现采用结构化方法对该采购系统进行分析与设计，获得如图 1-1 所示的上下文数据流图和图 1-2 所示的 0 层数据流图。

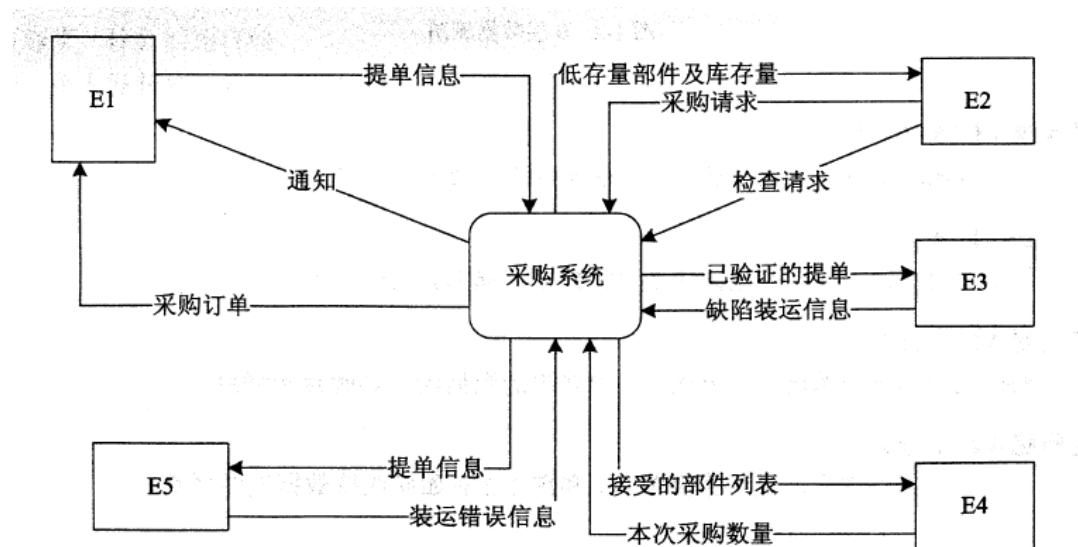
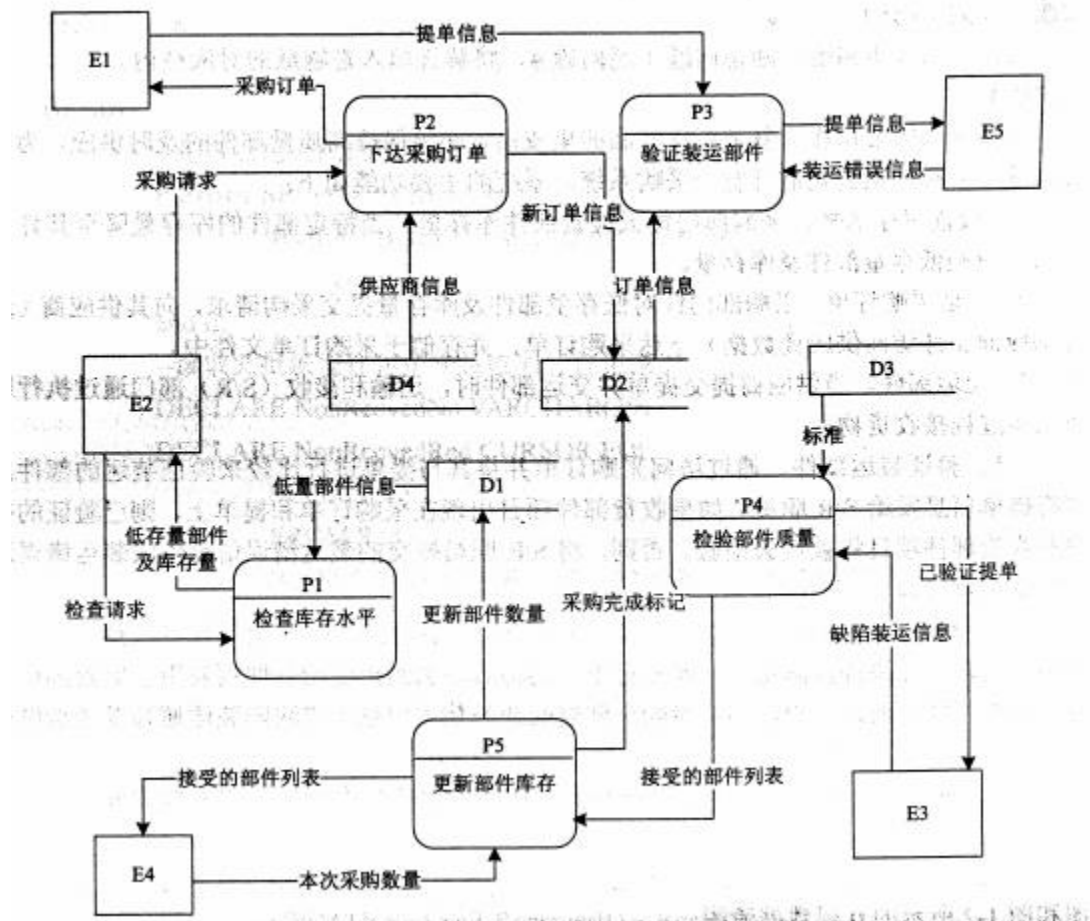


图 1-1 上下文数据流图



【问题 1】（5 分）

使用说明中的词语，给出图 1-1 中的实体 E1-E5 的名称。

E1: 供应商

E2: 采购部

E3: 检验员

E4: 库管员

E5: S/R 职员

依据题干中“下达采购订单。采购部门针对低存量部件及库存量提交采购请求，向其供应商（通过供应商文件访问供应商数据）下达采购订单，并存储于采购订单文件中。”可以判断出 E1 为供应商

依据题干中“检查库存水平。采购部门每天检查部件库存量，当特定部件的库存量降至其订货店时，返回低存量部件及库存量。”，可以判断出 E2 为采购部

依据题干中“通过访问质量标准来检查装运部件的质量，并将已验证的提单发给检验员。如果部件满足所有质量标准，则将其添加到接受的部件列表用于更新部件库存。”，可以判断出 E3 为检验员

依据题干中“库管员根据收到的接受的部件列表添加本次采购数量”，可以判断出 E4 为库管员

依据题干中“如果收货部件项目出现在采购订单和提单上，则已验证的提单和收货部件项目将被送去检验。否则 S/R 职员提交的装运错误信息生成装运错误通知发送给供应商。”，可以判断出 E5 为 S/R 职员

【问题 2】（4 分）

使用说明中的词语，给出图 1-2 中的数据存储 D1-D4 的名称。

D1: 库存表

D2: 采购订单表

D3: 质量标准表

D4: 供应商表

依据题干中“更新部件库存。库管员根据收到的接受的部件列表添加本次采购数量，与原有库存量累加来更新库存部件中的库存量。”，结合零层图的数据流，可以得出 D1 为库存表

依据题干中“下达采购订单。采购部门针对低存量部件及库存量提交采购请求，向其供应商（通过供应商文件访问供应商数据）下达采购订单，并存储于采购订单文件中。”和“更新部件库存。库管员根据收到的接受的部件列表添加本次采购数量，与原有库存量累加来更新

库存部件中的库存量。标记订单采购完成”，结合零层图的数据流，可以得出 D2 为采购订单表，D4 为采购订单表

依据题干中“检验部件质量。通过访问质量标准来检查装运部件的质量，并将已验证的提单发给检验员。如果部件满足所有质量标准，则将其添加到接受的部件列表用于更新部件库存。”，结合零层图的数据流，可以得出 D3 为质量标准表

【问题 3】（4 分）

根据说明和图中术语，补充图 1-2 中缺失的数据流及其起点和终点。

装运错误通知：P3（验证装运部件）-----E1（客户）

缺陷装运通知：P4（校验部件质量）-----E1（客户）

产品检验：P3（验证装运部件）----- P4（校验部件质量）

检查库存信息：P1（检查库存水平）-----D1（库存表）

依据题干中“（1）中：否则 S/R 职员提交的装运错误信息生成装运错误通知发送给供应商。”，结合零层图可以发现缺失数据流：装运错误通知：P3（验证装运部件）-----E1（客户）

依据题干中“（2）中：如果部件未通过检查，则将检验员创建的缺陷装运信息生成缺陷装运通知发送给供应商。”，结合零层图可以发现缺失数据流：缺陷装运通知：P4（校验部件质量）-----E1（客户）

依据题干中“（1）中：如果收货部件项目出现在采购订单和提单上，则已验证的提单和收货部件项目将被送去检验。”，结合零层图可以发现缺失数据流：产品检验：P3（验证装运部件）----- P4（校验部件质量）

依据题干中“1、检查库存水平。采购部门每天检查部件库存量，当特定部件的库存量降至其订货店时，返回低存量部件及库存量。”，结合零层图可以发现缺失数据流：检查库存信息：P1（检查库存水平）-----D1（库存表）

【问题 4】（2 分）

用 200 字以内文字，说明建模图 1-1 和图 1-2 时如何保持数据流图平衡。

父图中某个加工的输入输出数据流必须与其子图的输入输出数据流在数量上和名字上相同。父图的一个输入（或输出）数据流对应于子图中几个输入（或输出）数据流，而子图中组成的这些数据流的数据项全体正好是父图中的这一个数据流。

试题二

阅读下列说明，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

【说明】

某房屋租赁公司拟开发一个管理系统用于管理其持有的房屋、租客及员工信息。请根据下述需求描述完成系统的数据库设计。

【需求描述】

1. 公司拥有多幢公寓楼，每幢公寓楼有唯一的楼编号和地址。每幢公寓楼中有多套公寓，每套公寓在楼内有唯一的编号（不同公寓楼内的公寓号可相同）。系统需记录每套公寓的卧室数和卫生间数。

2. 员工和租客在系统中有唯一的编号（员主编号和租客编号）。

3. 对于每个租客，系统需记录姓名、多个联系电话、一个银行账号（方便自动扣房租）、一个紧急联系人的姓名及联系电话。

4. 系统需记录每个员工的姓名、类别、一个联系电话和月工资。员工类别可以经理或维修工，也可兼任。每个经理可以管理多幢公寓楼。每幢公寓楼必须由一个经理管理。系统需记录每个维修工的业务技能，如：水暖维修、电工、木工等。

5. 租客租赁公寓必须和公司签订租赁合同。一份租赁合同通常由一个或多个租客（合租）与该公寓楼的经理签订，一个租客也可租赁多套公寓。合同内容应包含签订日期、开始时间租期、押金和月租金。

【概念模型设计】

根据需求阶段收集的信息，设计的实体联系图（不完整）如图 2-1 所示。

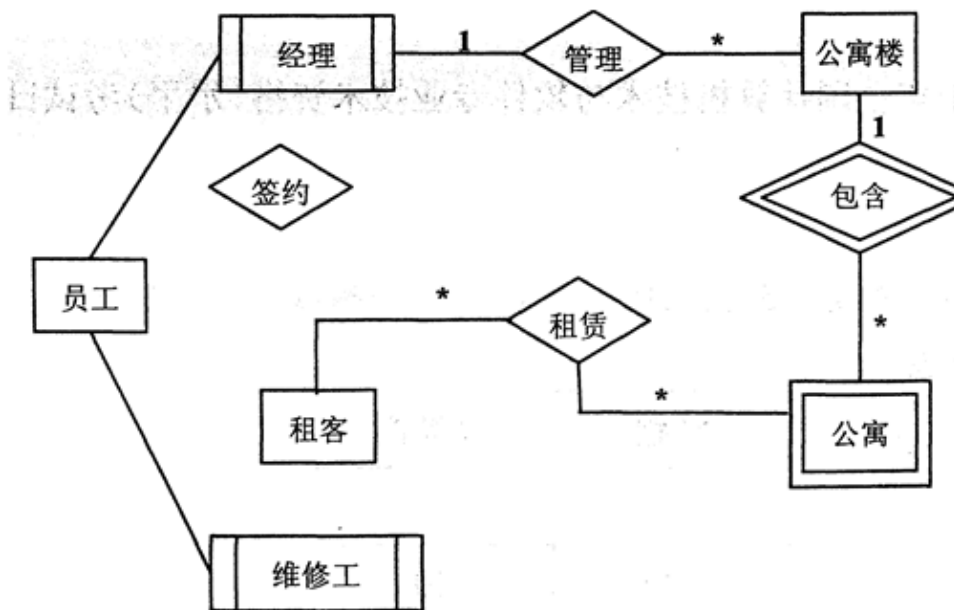


图 2-1 实体联系图

【逻辑结构设计】

根据概念模型设计阶段完成的实体联系图，得出如下关系模式(不完整)：

联系电话（电话号码，租客编号）

租客（租客编号，姓名，银行账号，联系人姓名，联系人电话）

员工（员工编号，姓名，联系电话，类别，月工资，（ a ））

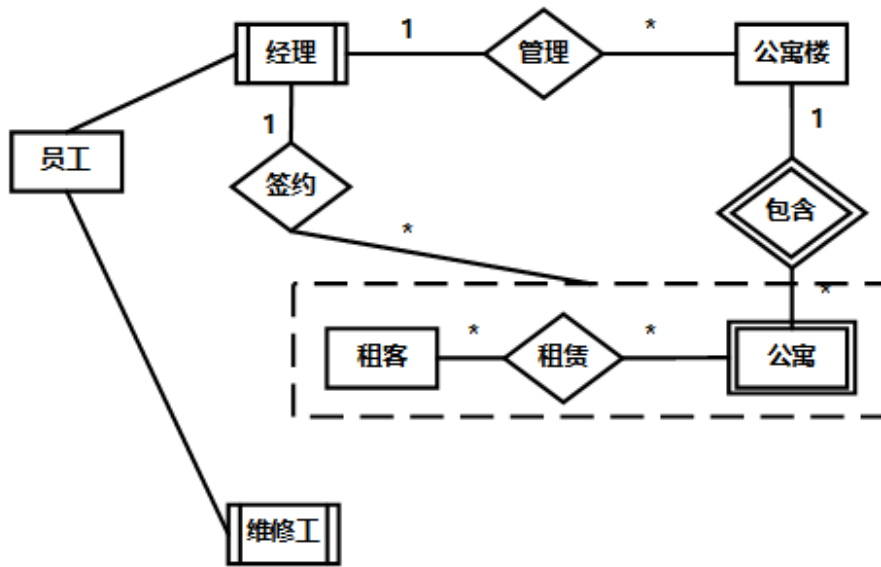
公寓楼（（ b ） ， 地址，经理编号）

公寓（楼编号，公寓号，卧室数，卫生间数）

合同（合同编号，租客编号，楼编号，公寓号，经理编号，签订日期，起始日期，租期，（ c ） ， 押金）

【问题 1】（4.5 分）

补充图 2-1 中的“签约”联系所关联的实体及联系类型。



依据题干中“租客租赁公寓必须和公司签订租赁合同。一份租赁合同通常由一个或多个租客（合租）与该公寓楼的经理签订，一个租客也可租赁多套公寓。合同内容应包含签订日期、开始时间租期、押金和月租金。”，说明签约应该是经理与租赁之间的，而已份租赁包括一位或多位租客，以及一个或多个公寓，所以可以考虑为：经理实体集与租赁（由租客和公寓组合成一个大的实体集）之间的联系

在结合题干中“每个经理管多个公寓楼，每个公寓楼由一个经理管理，和一个楼有多个公寓”的描述，可以判定联系的类型为 1：*

【问题 2】（4.5 分）

补充逻辑结构设计中的（a）（b）（c）三处空缺。

（a）业务技能

（b）楼编号

（c）月租金

从题干中“系统需记录每个员工的姓名、类别、一个联系电话和月工资。员工类别可以经理或维修工，也可兼任。每个经理可以管理多幢公寓楼。每幢公寓楼必须由一个经理管理。系统需记录每个维修工的业务技能，如：水暖维修、电工、木工等”说明需要记录的属性有：姓名、类别、一个联系电话、月工资和业务技能。因此（a）处应为：业务技能

题干中“每幢公寓楼有唯一的楼编号和地址以及每幢公寓楼必须由一个经理管理”同时管理联系没有转换成一个独立的关系，也就意味着管理联系被合并到了公寓楼的实体对应的关系中，因此，公寓楼实体对应的关系的属性应该有：楼编号、地址、经理编号；因此（b）处

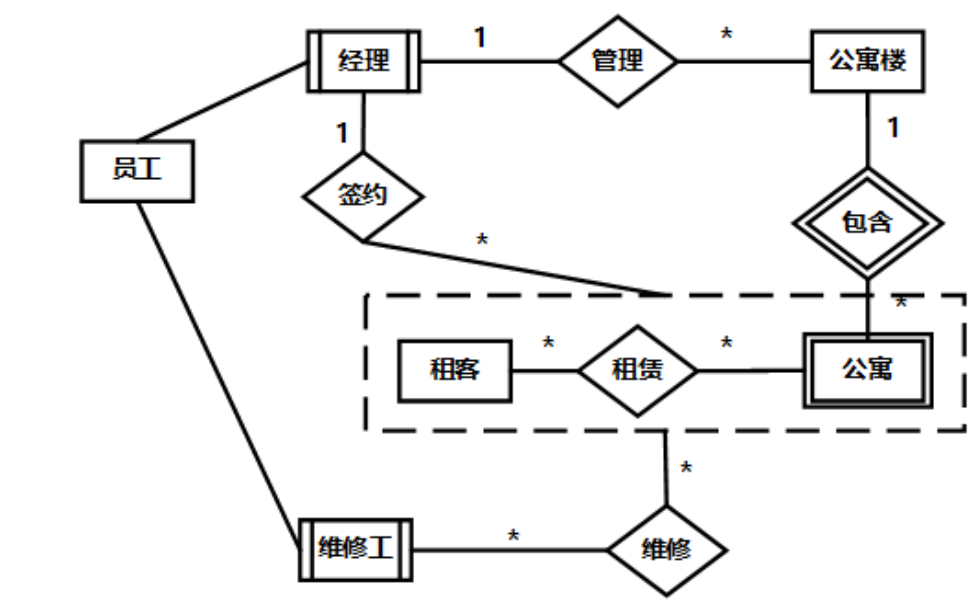
应该为：楼编号

依据题干中“合同内容应包含签订日期、开始时间租期、押金和月租金。”结合关系合同（合同编号，租客编号，楼编号，公寓号，经理编号，签订日期，起始日期，租期，(c)，押金），可以得出 (c) 处应该为：月租金

【问题3】(6分)

租期内，公寓内设施如出现问题，租客可在系统中进行故障登记，填写故障描述，每项故障由系统自动生成唯一的故障编号，由公司派维修工进行故障维修，系统需记录每次维修的维修日期和维修内容。请根据此需求，对图 2-1 进行补充，并将所补充的 ER 图内容转换为一个关系模式，请给出该关系模式。

维修（故障编号，维修工，维修日期，维修内容，楼编号，公寓号，租客编号）



题干中“租期内，公寓内设施如出现问题，租客可在系统中进行故障登记，填写故障描述，每项故障由系统自动生成唯一的故障编号，由公司派维修工进行故障维修，系统需记录每次维修的维修日期和维修内容”

说明，维修应该与租赁（由租客和公寓组合成一个大的实体集）之间存在多对多的联系，同时需要有自己的属性：故障编号、维修日期、维修内容。

维修（故障编号，维修工，维修日期，维修内容，楼编号，公寓号，租客编号）

试题三

【说明】

某社会救助基金会每年都会举办多项社会公益救助活动，需要建立一个信息系统，对其进行有效管理。

【需求描述】

1. 任何一个实名认证的个人或者公益机构都可以发起一项公益救助活动，基金会需要记录发起者的信息。如果发起者是个人，需要记录姓名、身份证号和一部电话号码；如果发起者是公益机构，需要记录机构名称、统一社会信用代码、一部电话号码、唯一的法人代表身份证号和法人代表姓名。一个自然人可以是多个机构的法人代表。

2. 公益救助活动需要提供详实的资料供基金会审核，包括被捐助人姓名、身份证号、一部电话号码、家庭住址。

3. 基金会审核并确认项目后，发起公益救助的个人或机构可以公开宣传并募捐，募捐得到的款项进入基金会账户。

4. 发起公益救助的个人或机构开展救助行动，基金会根据被捐助人所提供的医疗发票或其它信息，直接将所筹款项支付给被捐助者。

5. 救助发起者针对任一被捐助者的公益活动只能开展一次。

【逻辑结构设计】

根据上述需求，设计出如下关系模式：

公益活动(发起者编号，被捐助者身份证号，发起者电话号码，发起时间，结束时间，募捐金额)，其中对于个人发起者，发起者编号为身份证号；对于机构发起者，发起者编号为统一社会信用代码。

个人发起者(姓名，身份证号，电话号码)

机构发起者(机构名称，统一社会信用代码，电话号码，法人代表身份证号，法人代表姓名)

被捐助者(姓名，身份证号，电话号码，家庭住址)

【问题 1】(6 分)

对关系“机构发起者”，请回答以下问题：

- (1) 列举出所有候选键。
- (2) 它是否为 3NF，用 1100 字以内文字简要叙述理由。
- (3) 将其分解为 BC 范式，分解后的关系名依次为：机构发起者 1，机构发起者 2，...

并用下划线标示分解后的各关系模式的主键。

(1) 统一社会信用代码

(2) 否，存在传递依赖：统一社会信用代码→法人代表身份证号，法人代表身份证号→法人代表姓名

(3) 机构发起者 1（机构名称，统一社会信用代码，电话号码，人代表身份证号）主键：统一社会信用代码

机构发起者 2（人代表身份证号，法人代表姓名）主键：法人代表身份证号

【问题 2】(6 分)

对关系“公益摇动”，请回答以下问题：

(1) 列举出所有候选键。

(2) 它是否为 2NF，用 100 字以内文字简要叙述理由。

(3) 将其分解为 BC 范式，分解后的关系名依次为：公益活动 1，公益活动 2，...，并用下划线标示分解后的各关系模式的主键。

(1) 发起者编号+被捐助者身份证号

(2) 否：存在部分函数依赖：发起者编号→（发起者电话号码）对码（发起者编号，被捐助者身份证号）存在部分函数依赖

(3) 公益活动 1（发起者编号，发起者电话号码）主键：发起者编号

公益活动 2（发起者编号，被捐助者身份证号，发起时间，结束时间，募捐金额）主键：发起者编号+被捐助者身份证号

【问题 3】(3 分)

基金会根据被捐助人提供的医疗发票或其它信息，将所筹款项支付给被捐助者。可以存在分期多次支付的情况，为了统计所筹款项支付情况（详细金额和时间），试增加“支付记录”关系模式，用 100 字以文字简要叙述解决方案。

支付记录（支付编号，发起者编号，被捐助者身份证号，支付金额，支付时间，被捐助人的相关信息）（被捐助人的相关信息为医疗发票或其他信息），支付编号唯一标识每一次支付

试题四

阅读下列说明，回答问题 1 至问题 5，将解答填入答题纸的对应栏内。

【说明】

某公司要对其投放的自动售货机建立商品管理系统，其数据库的部分关系模式如下：

售货机：VEM (VEMno, Location)，各属性分别表示售货机编号、部署地点；

商品：GOODS (Gno, Brand, Price)，各属性分别表示商品编号、品牌名和价格；

销售单：SALES (Sno, VEMno, Gno, SDate, STime)，各属性分别表示销售号、售货机编号、商品编号、日期和时间。

缺货单：OOS (VEMno, Gno, SDate, STime)，各属性分别表示售货机编号、商品编号、日期和时间。

相关关系模式的属性及说明如下：(1) 售货机摆放固定种类的商品，售货机内每种商品最多可以储存 10 件。管理员在每天结束的时候将售货机中所有售出商品补全

(2) 每售出一件商品，就自动向销售单中添加一条销售记录。如果一天内某个售货机上某种商品的销售记录达到 10 条，则表明该售货机上该商品已售完，需要通知系统立即补货，通过自动向缺货单中添加一条缺货记录来实现。

根据以上描述，回答下列问题，将 SQL 语句的空缺部分补充完整。

【问题 1】(3 分)

请将下面创建销售单表的 SQL 语句补充完整，要求指定关系的主码和外码约束。

```
CREATE TABLE SALES (
```

```
Sno CHAR(8) (a)
```

```
VEMno CHAR(5) (b)
```

```
Gno CHAR(8) (c)
```

```
SDate DATE,
```

```
STime TIME);
```

(a) PRIMARY KEY

(b) REFERENCES VEM (VEMno)

(c) REFERENCES GOODS (Gno)

【问题 2】(4 分)

创建销售记录详单视图 SALES_Detail，要求按日期统计每个售货机上各种商品的销售数量，

属性有 VEMno、Location 、Gno、Brand 、Price 、amount 和 SDate。为方便实现，首先建立一个视图 SALES_Total ，然后利用 SALES_Total 完成视图 SALES_Detail 的定义。

```
CREATE VIEW SALES _Total (VEMno, Gno, SDate, amount) AS
SELECT VENno , Gno , SDate , count(*)
FROM SALES
GROUP BY (d);
CREATE VIWE (e) AS
SELECT VEM.VEMno, Location , GOODS.Gno , Brand, Price, amount, SDate
FROM VEM, GOODS, SALES_Total
WHERE (f) AND (g)
```

(d) VEMno, Gno, SDate

(e) SALES_Detail (VEMno, Location, Gno, Brand, Price, Amount, SDate)

(f) VEM.VEMno=SALES_Total. VEMno

(g) GOODS.Gno= SALES_Total. Gno

f 和 g 可以互换

【问题 3】 (3 分)

每售出一件商品，就自动向销售单中添加一条销售记录。如果一天内某个售货机上某种商品的销售记录达到 10 条，则自动向缺货单中添加一条缺货记录。需要用触发器来实现缺货单的自动维护。程序中的 GetTime () 获取当前时间。

```
CREAT(h) OOS_TRG AFTER (i) ON SALES
REFERENCING new row AS nrow
FOR EACH ROW
BEGIN
INSERT INTO OOS
SELECT SALES .VENno, (j) GetTime ( )
FROMSALES
WHERE SALES.VEMno = nrow.VEMno AND SALES.Gno = nrow.Gno
AND SALES.SDate = nrow.SDate
GROUP BY SALES.VEMno, SALES.Gno, SALES.SDate
```

HAVING count (*) > 0 AND mod(count (*), 10)=0;

END

(h) TRIGGER

(i) INSERT

(j) SALES.Gno, SALES.SDate

【问题4】 (3分)

查询当天销售最多的商品编号、品牌和数量。程序中的 GetDate() 获取当天日期。

SELECT GOODS.Gno , Brand, (k)

FROM GOODS, SALES

WHERE GOODS.Gno=SALES.GNO AND SDATE =GetDate()

GROUP BY (l)

HAVING(M) (SELECT count(*)

FROM SALELS

WHERE SDATE = GetDate()

GROUP BY Gno);

(k) COUNT (*) AS 数量

(l) GOODS.Gno, Brand

(m) COUNT (*) >=ALL

【问题5】 (2分)

查询一件都没有售出的所有商品编号和品牌。

SELECT Gno ,Brand

FROM GOODS

WHERE GNO(N)

SELECT DISTINCT GNO

FROM(o);

(n) NOT IN

(o) SALES

试题五

阅读下列说明，回答问题 1 和问题 2，将解答填入答题纸的对应栏内。

【说明】

某抢红包软件规定发红包人可以一次抛出多个红包，由多个人来抢。要求每个抢红包的人最多只能抢到同一批次中的一个红包，且存在多个人同时抢同一红包的情况。给定的红包关系模式如下：

Red(ID, BatchID, SenderID, Money, ReceiverID)

其中 ID 唯一标识每一个红包；BatchID 为发红包的批次，一个 BatchID 值可以对应多个 ID 值；SenderID 为发红包人的标识；Money 为红包中的钱数；ReceiverID 记录抢到红包的人的标识。

发红包人一次抛出多个红包，即向红包表中插入多条记录，每条记录表示一个红包，其 ReceiverID 值为空值。

抢某个红包时，需要判定该红包记录的 ReceiverID 值是否为空，不为空时表示该红包已被抢走，不能再抢，为空时抢红包人将自己的标识写入到 ReceiverID 字段中，即为抢到红包。

【问题 1】(9 分)

引入两个伪指令 $a = R(X)$ 和 $W(b, X)$ 。其中 $a = R(X)$ 表示读取当前红包记录的 ReceiverID 字段(记为数据项 X) 到变量 a 中， $W(b, X)$ 表示将抢红包人的唯一标识 b 的值写入到当前红包记录的 ReceiverID 字段(数据项 X) 中，变量 a 为空值时才会执行 $W(b, X)$ 操作。假设有多个人同时抢同一红包(即同时对同一记录进行操作)，用 $a_i = R_i(X)$ 和 $W_i(b_i, X)$ 表示系统依次响应的第 i 个人的抢红包操作。假设当前数据项 X 为空值，同时有三个人抢同一红包，则

(1) 如下的调度执行序列：

$a_1 = R_1, a_2 = R_2(X), W_1(b_1, X), W_2(b_2, X), a_3 = R_3(X)$

抢到红包的是第几人？并说明理由。

(2) 引入共享锁指令 SLock $_i(X)$ 、独占锁指令 XLock $_i(X)$ 和解锁指令 ULock $_i(X)$ ，其中下标 i 表示第 i 个抢红包人的指令。如下的调度执行序列：

SLock $_1(X), a_1 = R_1(X), SLock_2(X), a_2 = R_2(X), XLock_1(X), \dots$

是否会产生死锁？并说明理由。(3) 为了保证系统第一个响应的抢红包人为最终抢到红包的

人，请使用上述。)中引入的锁指令，对上述(1)中的调度执行序列进行修改，在满足 2PL 协议的前提下，给出一个不产生死锁的完整的调度执行序列。

(1) 第 2 人，并发操作出现了丢失更新的问题，第 2 个的更新覆盖了第 1 个的更新，原因是破坏了事物的隔离性。

(2) 会产生死锁，由于数据 X 同时被 1 和 2 加锁了 S 锁，再对方没有释放的时候，都无法加成功 X 锁，导致 1 和 2 一直都处于等待的状态

(3) XLock (X) a=R (X) W (b, X) UNLock (X)

【问题 2】(6 分)

下面是用 SQL 实现的抢红包程序的一部分，请补全空缺处的代码

```
CREATE PROCEDURE ScrambleRed (IN BatchNo VARCHAR(20) , --红包批号
```

```
(IN RecvrNo VARCHAR(20)) --接收红包者
```

```
BEGIN
```

```
--是否已抢过此批红包
```

```
if exists( SELECT * FROM Red
```

```
WHERE BatchID = BatchNo AND ReceiverID = RecvrNo) then
```

```
return -1;
```

```
end if;
```

```
---读取此批派发红包中未领取的红包记录 ID
```

```
DECLARE NonRecvedNo VARCHAR(30);
```

```
DECLARE NonRecvedNo CURSOR FOR
```

```
SELECT ID
```

```
FROM Red
```

```
WHERE BatchID = BatchNo AND ReceiverID IS NULL;
```

```
--打开游标
```

```
OPEN NonRecvedRed;
```

```
FETCH NonRecvedRed INTO NonRecvedNo;
```

```
while not error
```

```
--抢红包事务
```

```

BEGIN TRANSACTION;

//写入红包记录

UPDATE RED SET RECDIVER ID =RecvNo

WHERE ID = nonRECVED AND (a)

//执行状态判定

If <修改的记录数>= 1 THEN

COMMIT;

(b);

Return 1;

Else

ROLLBACK;

End if;

(c);

End while

--关闭游标

CLOSE NonRecved RD

Return 0;

END

```

(a) BatchID=BatchNo

(b) CLOSE NonRecvedRed

(c) END TRANSACTION