

## 2021 年上半年数据库系统工程师（上午+下午）答案详解

如碰到错别字或者答案错误，请联系售后微信:ruankaopass 或者扫码

扫一扫，叫我微信号:ruankaopass



提供软考历年真题，视频

【1】A

【解析】

指令寄存器 IR:保存从存储器中读出的当前要执行的指令。

指令译码器 ID:用来对指令寄存器中的指令进行译码分析,指出指令的操作种类和寻址方式。

程序计数器 PC:用来存放下一条要执行指令的存储器地址,按此地址从对应存储单元取出的内容就是将要执行的指令。

【2】D

【解析】

	CISC	RISC
指令条数	多	只选取最常见的指令
指令复杂度	高	低
指令长度	变化	短、固定
指令执行周期	随指令变化大	大多在一个机器周期完成
指令格式	复杂	简单
寻址方式	多	少
涉及访问主存指令	多	极少,大部分只有存和取两条指令
通用寄存器数里	一般	大量
译码方式	微程序控制	硬件电路
对编译系统要求	低	高

【3】C

【解析】

DMA 是直接内存存取，传送数据的时间只与内存相关，与 CPU 的时间无关。

【4】C

【解析】

提高磁盘转速只能减少旋转延迟时间。

1、磁盘的访问时间

●寻道时间  $T_s$ :把磁臂从当前位置移到指定磁道上所经历的时间。

●旋转延迟时间  $T_r$ :指定扇区移动到磁头下面所经历的时间。

●传输时间  $T_t$ :数据从磁盘读出或向磁盘写入数据所经历的时间。

在访问时间中，寻道时间和旋转延迟时间，通常是占据了访问时间的大头。适当地集中数据（不要太零散）传输，将有利于提高传输效率。

【5】B

【解析】

栈是先进后出的数据结构，如果 5 最先出来，那么 1, 2, 3, 4 肯定都已经入栈，那 5, 4, 后面就不会是 1 先出来，再 3 出来。

【6】A

【解析】

排序方法	时间复杂度	辅助空间	稳定性
直接插入	$O(n^2)$	$O(1)$	稳定
简单选择	$O(n^2)$	$O(1)$	不稳定
冒泡排序	$O(n^2)$	$O(1)$	稳定
希尔排序	$O(n^{1.3})$	$O(1)$	不稳定
快速排序	$O(n\log n)$	$O(\log n)$	不稳定
堆排序	$O(n\log n)$	$O(1)$	不稳定
归并排序	$O(n\log n)$	$O(n)$	稳定
基数排序	$O(n\log n)$	$O(rd)$	稳定

**【7】B**

**【解析】**

队列是一种先进先出(FIFO)的线性表，它只允许在表的一端插入元素，而在表的另一端删除元素。在队列中，允许插入元素的一端称为队尾(rear)，允许删除元素的一端称为队头(front)。

**【8】C**

**【解析】**

结点数： $2^5-1=31$ ，第5层最多为 $2^{5-1}=2^4=16$

1、二叉树的性质

(1)二叉树第*i*层上的结点数最多为 $2^{i-1}(i \geq 1)$ 。

(2)深度为*k*的二叉树至多有 $2^k-1$ 个结点( $k \geq 1$ )。

(3)在任意一棵二叉树中，若终端结点数为 $n_0$ ，度为2的结点数为 $n_2$ ，则 $n_0=n_2+1$ 。

(4)具有*n*个结点的完全二叉树的深度为 $\lfloor \log_2 n \rfloor + 1$ 。

**【9】B**

**【解析】**

同第【8】题

**【10】D**

**【解析】**

希尔排序(Shell Sort)也称缩小增量排序，是直接插入排序算法的一种更高效的改进版本。先将整个待排元素序列分割成若干个子序列(由相隔某个“增量”的元素组成的)分别进行直接插入排序，然后依次缩减增量再进行排序，待整个序列中的元素基本有序(增量足够小)时，再对全体元素进行一次直接插入排序。

**【11】B**

**【解析】**

系统的保护机制应该公开。

**【12】C**

**【解析】**

DoS 是 Denial of Service 的简称, 即拒绝服务, DoS 攻击是指故意的攻击网络协议实现的缺陷或耗尽被攻击对象的资源, 目的是让目标计算机或网络无法提供正常的服务或资源访问, 使目标系统服务系统停止响应甚至崩溃。这些服务资源包括网络带宽, 文件系统空间容量, 内存, 开放的进程或者允许的连接。

**【13】D**

**【解析】**

蜜罐技术是一种在互联网上运行的计算机系统, 它通过模拟一个或多个易受攻击的主机和服务器, 来吸引诱骗那些试图非法闯入他人计算机系统的人。

●主要功能: 对系统中所有的操作和行为进行监视和记录。通过对系统进行伪装, 使得攻击者在进入到蜜罐系统后并不会知晓其行为已经处于系统的监视之中, 然后根据它攻击行为, 分析攻击的方法和攻击企图。

**【14】D**

**【解析】**

1. 不使用字符串组合的方式执行 SQL 命令, 而使用参数式 SQL 命令 (PreparedStatement)。
2. 对输入参数的值进行过滤、净化
3. 避免使用数据库 Root 或高权限执行 SQL
4. SQL 出错信息不要发送到客户端, 在 Spring 框架中可以拦截统一处理
5. 密码不要明文存储, 可以 Hash 加密 (如 md5), 安全性要求较高, 最好使用使用加盐 Hash 加密。

**【15】B**

**【解析】**

重放攻击又称重播攻击、回放攻击, 是指攻击者发送一个目的主机已接收过的包来达到欺骗系统的目的。主要用于身份认证过程, 破坏认证的正确性。防范重放攻击可以加时间戳, 或者一次一密的方式。

**【16】D**

**【解析】**

对著作权的保护只是针对计算机软件和文档,并不包括开发软件所用的思想、处理过程、操作方法或数学概念等,并且著作权人还需在软件登记机构办理登记。

**【17】D**

**【解析】**

两个或者两个以上的申请人,在同一种商品或者类似商品上,分别以相同或者近似的商标在同一天申请注册的,各申请人应当自收到商标局通知之日起 30 日内提交其申请注册前在先使用该商标的证据。

●同日使用或者均未使用的,各申请人可以自收到商标局通知之日起 30 日内自行协商,并将书面协议报送商标局;

●不愿协商或者协商不成的,商标局通知各申请人以抽签的方式确定一个申请人,驳回其他人的注册申请;

●商标局已经通知但申请人未参加抽签的,视为放弃申请,商标局应当书面通知未参加抽签的申请人。

**【18】B**

**【解析】**

引入缓冲区的主要原因归结为以下几点:

●缓和 CPU 与 I/O 设备间速度不匹配的矛盾。

●减少对 CPU 的中断频率,放宽对 CPU 中断响应时间的限制。

●提高 CPU 和 I/O 设备之间的并行性。

**【19】C**

**【解析】**

先来先服务调度算法(FCFS):就是按照各个作业进入系统的自然次序来调度作业。这种调度算法的优点是实现简单,公平。其缺点是没有考虑到系统中各种资源的综合使用情况,往往使短作业的用户不满意,因为短作业等待处理的时间可能比实际运行时间长得多。

短作业优先调度算法(SPF):就是优先调度并处理短作业,所谓短是指作业的运行时间

短。而在作业未投入运行时，并不能知道它实际的运行时间的长短，因此需要用户在提交作业时同时提交作业运行时间的估计值。

时间片轮转调度算法：每个进程被分配一个时间段，称作它的时间片，即该进程允许运行的时间。如果在时间片结束时进程还在运行，则 CPU 将被剥夺并分配给另一个进程。如果进程在时间片结束前阻塞或结束，则 CPU 当即进行切换。调度程序所要做的就是维护一张就绪进程列表，当进程用完它的时间片后，它被移到队列的末尾。

基于优先级调度算法 (HPF)：每一个作业规定一个表示该作业优先级别的整数，当需要将新的作业由输入井调入内存处理时，优先选择优先数最高的作业。

作业周转时间  $(T_i) = \text{完成时间}(T_{ei}) - \text{提交时间}(T_{si})$  作业平均周转时间  $(T) = \text{周转时间} / \text{作业个数}$

#### 【20】B

##### 【解析】

线程共享的内容包括：

进程代码段

进程的公有数据 (利用这些共享的数据，线程很容易的实现相互之间的通讯)、进程打开的文件描述符、信号的处理器、进程的当前目录和进程用户 ID 与进程组 ID

线程独有的内容包括：

线程 ID

寄存器组的值

线程的堆、栈

错误返回码

线程的信号屏蔽码

#### 【21】C

##### 【解析】

摒弃“不剥夺”条件

系统规定，进程是逐个地提出对资源的要求的。当一个已经保持了某些资源的进程，提出新的要求不被满足时必须释放它已经保持的所有资源，待以后需要时再重新申请。从而摒弃了“不剥夺”条件。

- 某一进程已经占有的资源，在运行过程中会被暂时释放掉，认为是被剥夺了。
- 实现起来比较复杂且付出很大代价。可能会前功尽弃，反复申请和释放等情况，延长了周转时间，增加系统开销。

**【22】C**

**【解析】**

一般地，程序设计语言的定义都涉及语法、语义和语用等方面。

语法是指由程序设计语言的基本符号组成程序中的各个语法成分(包括程序)的一组规则，其中由基本字符构成的符号(单词)书写规则称为词法规则，由符号构成语法成分的规则称为语法规则。程序设计语言的语法可用形式语言进行描述。

语义是程序设计语言中按语法规则构成的各个语法成分的含义，可分为静态语义和动态语义。静态语义指编译时可以确定的语法成分的含义，而运行时刻才能确定的含义是动态语义。一个程序的执行效果说明了该程序的语义，它取决于构成程序的各个组成部分的语义。

语用表示了构成语言的各个记号和使用者的关系，涉及符号的来源、使用和影响。

语言的实现则有个语境问题。语境是指理解和实现程序设计语言的环境，包括编译环境和运行环境。

**【23】D**

**【解析】**

同【22】解析

**【24】C**

**【解析】**

③语义分析阶段

语义分析阶段主要是检查源程序是否存在语义错误，并收集类型信息供后面的代码生成阶段使用，只有语法和语义都正确的源程序才能翻译成正确的目标代码。

语义分析的主要工作是进行各类型分析和检查。赋值语句的右端和左端的类型不匹配。表达式的除数是否为零等。

```
int arr[2],b;
```

```
b=arr* 10;
```

【25】B

【解析】

二、软件质量特性

ISO/IEC9126 软件质量模型是一种评价软件质量的通用模型，包括 3 个层次：

- 1、质量特性
- 2、质量子特性
- 3、度量指标

质量特性	功能性	可靠性	易用性	效率	维护性	可移植性
质量子特性	合适性	成熟型	易理解性	时间特性	易分析性	适应性
	准确性	容错性	易学性	资源利用性	易改变性	易安装性
	互操作性	易恢复性	易操作性		稳定性	共存性
	保密安全性		吸引性		易测试性	易替换性
	依从性	依从性	依从性	依从性	依从性	依从性

【26】B

【解析】

对象图可以看做是类图的实例，用来表达各个对象在某--时刻的状态。

【27】B

【解析】

将瀑布模型和演化模型相结合，综合了两者的优点，并增加了风险分析。它以原型为基础，沿着螺线自内向外旋转，每旋转一圈都要经过制订计划、风险分析、实施工程及客户评价等活动，并开发原型的一个新版本。经过若干次螺旋上升的过程，得到最终的系统。

【28】A

【解析】

初始级(Initial)。处于这个最低级的组织，基本\_上没有健全的软件工程管理制度。每件事情都以特殊的方法来做。软件开发的成功依赖软件开发人员个人能力。

可重复级(Repeatable)。有基本的软件项目的管理行为、设计和管理技术是基于相似产品中



的经验，故称为“可重复”。在这一级采取了一定措施，这些措施是实现一个完备过程所必不可少的第一步。典型的措施包括仔细地跟踪费用和进度。

已定义级(Defined)。已为软件生产的过程编制了完整的文档。软件过程的管理方面和技术方面都明确地做了定义，并按需要不断地改进过程，而且采用评审的办法来保证软件的质量。

已管理级(Managed)。对每个项目都设定质量和生产目标。这两个量将被不断地测量，当偏离目标太多时，就采取行动来修正。利用统计质量控制，管理部门能区分出随机偏离和有深刻含义的质量或生产目标的偏离。

优化级(Optimizing)。组织的目标是连续地改进软件过程。这样的组织使用统计质量和过程控制技术作为指导。从各个方面中获得的知识将被运用在以后的项目中，从而使软件过程融入了正反馈循环，使生产率和质量得到稳步的改进。

**【29】A**

**【解析】**

默认的是：`http://www.abe.com` ---协议名称，服务器名，域名。

**【30】A**

**【解析】**

WLAN 是 Wireless Local Area Network (无线局域网)的简称，指应用无线通信技术将计算机设备互联起来，构成可以互相通信和实现资源共享的网络体系。

**【31】C**

**【解析】**

防火墙作为访问控制设备，主要工作在OSI 模型的三四层。防火墙主要基于IP 报文进行检测，对端口进行限制。还有一些防火墙具备一定的应用层防护能力，可以根据TCP 会话异常性及攻击特征阻止攻击，通过IP 拆分检测也能够判断隐藏在数据包中的攻击。

但是防火墙没有加密认证的功能。

**【32】C**

**【解析】**

Secure Shell (SSH) 是由 IETF 制定的建立在应用层基础上的安全网络协议。它是专为远程登录会话和其他网络服务提供安全性的协议，可有效弥补网络中的漏洞。

【33】B

【解析】

数据模型中，唯一标识实体的属性集称为码。

【34】A

【解析】

1、域

是一组具有相同数据类型的值的集合。

整数

实数

{ ‘男’，‘女’ }

.....

【35】B

【解析】

相比于文件系统，数据库管理系统具有数据独立性高的优势。

【36】D

【解析】

在用户观点下，关系模型中数据的逻辑结构是一张二维表，它由行和列组成。用表格结构表达实体集，用外键(外码)表示实体间联系。



学号	姓名	年龄	性别	系名	年级
2005004	王小明	19	女	社会系	2005
.....	.....	.....	.....	.....	.....

【37】C

【解析】

给定关系R(X, Y)和S(Y, Z), 其中X, Y, Z 为属性组。R 中的Y 与S 中的Y 可以有不同

的属性名, 但必须出自相同的域集。R 与S 的除运算得到一个新的关系P(X), P 是R 中满足下列条件的元组在X 属性列上的投影: 元组在X 上分量值x 的象集Y<sub>x</sub> 包含S 在Y 上投影的集合, 记作:  $R \div S = \{tr[X] | tr \in R \wedge \pi_Y(S) \subseteq Y_x\}$

Y<sub>x</sub> : x 在 R 中的象集,  $Y_x = \{tr[Y] | tr \in R \wedge tr[X] = x\}$

【38】D

【解析】

四、范式

如果关系模式 R 的每个关系 r 的属性值都是不可分的原子值, 那么称 R 是第一范式的模式, r 是规范化的关系。

姓名	工资			入职时间
	基本工资	岗位工资	月度奖金	
李明	2500	1500	300	2010.11

说明: 在任何一一个关系数据库中, 第一范式是对关系模式的基本要求, 不满足第一范式(1NF) 的数据库就不是关系数据库。

【39】C

【解析】

SQL 是一门ANSI 的标准计算机语言, 用来访问和操作数据库系统。SQL 可与数据库程序协同工作, 而关系型数据库管理系统使用的就是SQL 语言进行数据的操作。

【40】A

【解析】

缺省为 NULL

【41】C

【解析】

事务四种特性: 原子性、一致性、隔离性、持久性。

原子性是指事务包含的所有操作要么全部成功, 要么全部失败回滚, 因此事务的操作如果成

功就必须完全应用到数据库，如果操作失败则不能对数据库有任何影响。

一致性是指事务必须使数据库从一个一致性状态变换到另一个一致性状态，也就是说一个事务执行之前和执行之后都必须处于一致性状态。

隔离性是当多个用户并发访问数据库时，比如同时操作同一张表时，数据库为每一个用户开启的事务，不能被其他事务的操作所干扰，多个并发事务之间要相互隔离。

持久性是指一个事务一旦被提交了，那么对数据库中的数据的改变就是永久性的，即便是在数据库系统遇到故障的情况下也不会丢失提交事务的操作。

【42】C

【解析】

is\_valid=1 表示社保有效，is\_local=0 表示不在当地。所以 where 的条件应该是 insurance.is\_valid=1 and employee.is\_local=0 and insurance.id= employee.id

【43】A

【解析】

一、授权 (GRANT) 与销权 (REVOKE)

1. 授权语句格式

GRANT<权限>[<权限>]... [ON<对象类型><对象名>]TO<用户>[, <用户>]... [WITH GRANT OPTION];

对象	对象类型	操作权限
属性列	TABLE	SELECT. INSERT, UPDATE. DELETE. ALL PRIVILEGES(5种权限的总和)
视图	TABLE	SELECT. INSERT, UPDATE. DELETE. ALL PRIVILEGES(5种权限的总和)
基本表	TABLE	SELECT. INSERT. UPDATE, DELETE. ALTER. INDEX ALLPRIVILEGES(6种权限的总和)
数据库	DATABASE	CREATETAB建立表的权限,可由DBA授予普通用户

【44】B

【解析】

DELETE 是删除数据表里的数据。

DROP 是删除数据表。

ALTER TABLE 用于在已有的表中添加、修改或删除列。

**【45】C**

**【解析】**

select 列名 from 表名 where 条件 group by 分组字段 having 条件 order by 排序字段。

**【46】A**

**【解析】**

根据函数依赖集可以得到选课的候选键为：(学号，课程号)和(学号，课程名)。由于存在课程号→课程名，这个决定因数并不包含码。所以每个非平凡函数依赖的决定因素都包含码是错误的。

**【47】D**

**【解析】**

BCNF 是 3NF 的改进形式

一个满足 BCNF 的关系模式的条件：

1. 所有非主属性对每一个码都是完全函数依赖。
2. 所有的主属性对每一个不包含它的码, 也是完全函数依赖。
3. 没有任何属性完全函数依赖于非码的任何一组属性。

所以答案是 BCNF 和 BCNF 。

**【48】D**

**【解析】**

(学号，课程号)→课程号

平凡函数依赖的定义。当关系中属性集合 Y 是属性集合 X 的子集时 ( $Y \subset X$ )，存在函数依赖  $X \rightarrow Y$ ，即一组属性函数决定它的所有子集，这种函数依赖称为平凡函数依赖。

根据平凡函数依赖的定义，正确答案是平凡的函数依赖。

**【49】D**

**【解析】**

数据库事务(transaction)是访问并可能操作各种数据项的一个数据库操作序列, 这些操作要么全部执行, 要么全部不执行, 是一个不可分割的工作单位。事务由事务开始与事务结束之间执行的全部数据库操作组成。所以, 事务是可以包含一组 SQL 语句, 一个数据库应用程序只能包含一个数据库事务, 一个数据库事务仅包含条 SQL 语句, 一个数据库事务仅包含一个存储过程是错误的。

**【50】A**

**【解析】**

等价的 3 种情况: 分解具有无损连接性; 分解要保持函数依赖; 分解既要无损连接性, 又要保持函数依赖。

**【51】C**

**【解析】**

$F^+$  为依赖集  $F$  的闭包。 $(CE)^+$  为求  $(CE)$  属性 的闭包。 $C \rightarrow D$ ,  $E$  不能决定任何, 所以  $(CE)^+ = CED$

**【52】B**

**【解析】**

COMMIT 表示数据库正常提交, ROLLBACK 表示数据库撤销提交, 都表示数据库事务的结束。显式事务: 显式事务又称自定义事务, 是指用显式的方式定义其开始和结束的事务, 当使用 start transaction 和 commit 语句时则表示发生显式事务。

隐式事务: 隐式事务是指每一条数据操作语句都自动地成为一个事务, 事务的开始是隐式的, 事务的结束有明确的标记。即当用户进行数据操作时, 系统自动开启一个事务, 事务的结束则需手动调用 COMMIT 或 ROLLBACK 语句来结束当前事务。

**【53】A**

**【解析】**

(3) 三级封锁协议

一级封锁协议加上事务  $T$  在读取数据  $R$  之前先对其加  $S$  锁, 直到事务结束才释放。三级封锁协议可防止丢失修改、防止读“脏”数据, 还进一步防止了不可重复读。

**【54】C**

**【解析】**

两段锁可以保证多个事务的并发执行是可串行化的调度。所以答案是两段锁协议。

**【55】D**

**【解析】**

两段锁是不能保证死锁的，没有发生是因为这次的调度刚好没有出现而已。所以答案是偶然的调度。

**【56】A**

**【解析】**

触发器也可用于强制引用完整性, 以便在多个表中添加、更新或删除行时，保留在这些表之间所定义的关系。触发器是不能被应用程序显示调用，所以也是不能带参数的。

**【57】A**

**【解析】**

```
CREATE PROCEDURE (参数列表){  
过程体;  
}
```

**【58】C**

**【解析】**

REDO 是重做操作； UNDO 是撤销操作。

**【59】A**

**【解析】**

(1) 事务故障及恢复

事务故障:是指事务运行过程中由于种种原因使事务未运行至正常终止点而夭折的情况。如:

运算溢出、因死锁而被选中撤销该事务

事务故障的恢复:

(1) 反向扫描文件日志，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。

(3) 重复(1)、(2)直到遇此事务的开始标记。

**【60】B**

**【解析】**

并发调度的结果与某一种串行调度结构相同，是并发调度正确的必要条件。这个是正确的，并发调度的结果只要跟某个串行调度结果相同，就代表是正确的。所以答案是并发调度结果与某一种串行调度结果相同，是并发调度正确的必要条件。

**【61】D**

**【解析】**

三、数据库设计的步骤

●用户需求分析

●概念结构设计

●逻辑结构设计

●物理结构设计

**【62】C**

**【解析】**

存储记录的布局，就是确定数据的存放位置。存储记录作为一个整体，如何分布在物理区域上，是数据库物理结构设计的重要一环。聚簇功能可以大大提高按聚簇码进行查询的效率。

**【63】A**

**【解析】**

数据库系统的三级模式, 数据库系统可以分为:

●外模式(子模式、用户模式)

●模式(概念模式、逻辑模式)

●内模式(存储模式)



【64】D

【解析】

设 $R(U)$ 是属性集 $U$ 上的一个关系模式。 $X$ ， $Y$ ， $Z$ 是 $U$ 的子集，并且 $Z=U-X-Y$ 。关系模式 $R(U)$ 中多值依赖 $X \twoheadrightarrow Y$ 成立，当且仅当对 $R(U)$ 的任一关系 $r$ ，给定的一对 $(x, z)$ 值有一组 $Y$ 的值，这组值仅仅决定于 $x$ 值而与 $z$ 值无关。

平凡的多值依赖与非平凡的多值依赖：若 $X \twoheadrightarrow Y$ ，而 $Z$ 为空集，则称 $X \twoheadrightarrow Y$ 为平凡的多值依赖；若 $Z$ 不为空，则称其为非平凡的多值依赖。所以，4NF要求关系模式的属性之间不允许有非平凡且非函数依赖的多值依赖。

【65】A

【解析】

(3) 介质故障(硬故障)

介质故障:是指外存设备故障。如:磁盘损坏、强磁场干扰

介质故障的恢复:

(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) 装入相应的日志文件副本，重做已完成的事务。

【66】D

【解析】

介质故障即外存故障，如磁盘损坏、强磁场干扰等。这类故障发生的可能性较小，但破坏性很强。它使数据库受到破坏，并影响正在存取数据的事务。

【67】D

【解析】

数据库重组:将数据库的相关信息重新组织。数据库使用较长一段时间后，因为增、删、改等操作，使得数据的分布索引及相关数据会变得比较凌乱，从而影响数据库的查询效率。数据库重组即是将数据库的相关信息(索引、单表、表空间)重新组织，即删除原有的表或索引，重建空的表或索引后，再把数据导入新表或索引中。数据库重构:对表结构、数据、存储过程和触发器重新设计，在不改变软件表现形式的前提下，提高其可理解性，降低其维护成本。

【68】D

【解析】

分布式数据库的存取结构复杂。

【69】C

【解析】

Cassandra 是列存储型数据库；MongoDB 是文档型数据库；PostgreSQL 是一种特性非常齐全的自由软件的对象-关系型数据库管理系统 (ORDBMS)；Neo4 是图存储型数据库。

【70】B

【解析】

CAP 指的是：一致性 (Consistency)、可用性 (Availability)、分区容错性 (Partition tolerance)。

【71-75】A C B B B

【解析】

试题一

【问题 1】

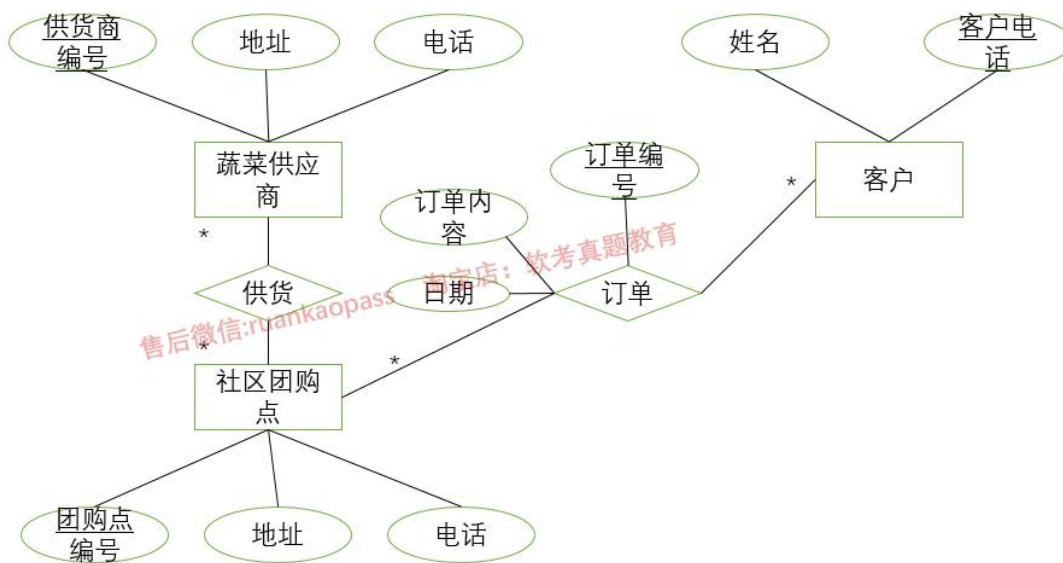


图1-1 实体联系图

【问题 2】

(a)：团购点编号。主键为：(供货商编号，团购点编号)，外键为：供货商编号，团购点编号。

(b)：客户电话。主键为:订单编号, 外键为:团购点编号, 客户电话。

【问题 3】

若社区蔬菜团购网站还兼有代收快递的业务, 请增加新的“快递”实体, 并给出客户实体和快递实体之间的“收取”联系, 对图 1 进行补充。“快递”关系模式包括快递编号、客户电话和日期。

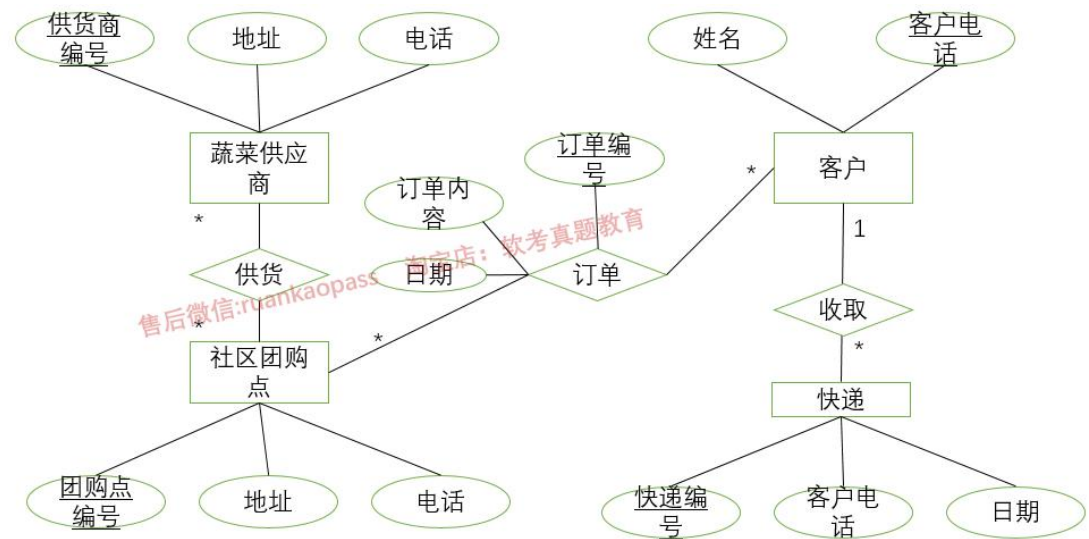


图1-1 实体联系图

试题二

【问题 1】

左侧只有一个属性的函数依赖关系有 2 个, 分别为:

车牌号->行驶路线

消毒人员工号->消毒人员姓名

不满足 2NF, 根据题干可以得到消毒记录表的候选键为(日期, 车牌号), 由于存在车牌号>行驶路线, 属于部分依赖。根据 2NF 的要求, 不存在非主数据对主属性的部分依赖, 所以不满足 2NF。

【问题 2】

由于消毒记录表不满足 2NF, 需要将消毒记录表进行拆分成多个表, 拆分的表如下:

消毒记录 1 (车牌号, 行驶路线)

消毒记录 2 (消毒人员工号, 消毒人员姓名)

消毒记录 3 (车牌号, 日期, 消毒人员工号)

由于拆分后的三个表都不存在非主属性对主属性的部分和传递函数依赖，所以都满足 3NF。

**【问题 3】**

此题目目前暂无答案，后续如果有解析出来再进行补充。可以加微信：ruankaopass

**试题三**

**【问题 1】**

(a) : PLAYER

(b) : CONTEST

(C) : NOT NULL

(d) : CHECK(Rank in('一','二','三','无'))

(e) : PRIMARY KEY (Pno,Cno)

**【问题 2】**

f: PLAYER

g: not in

h: pno

i pc.cno=contest.cno

j and

k: order by

**【问题 3】**

l: delete

m: PC

n: delete

o: CONTEST

**试题四**

**【问题 1】**

a:sum

b:bno

c:10000

d:1

e. commit

**【问题 2】**

f: after

g: all\_nums

h: for each row

i NEW.all\_nums

j OLD.all\_nums

k: bookorders

**【问题 3】**

说明 T4 时刻，用户 2 事务读取到的 all\_nums 数值是 498。

在 T8 时刻，all\_nums 数据会出现不一致性的问题，由于户 2 事务读到了用户 1 修改过的 all\_nums，然后在 T7 时刻用户 1 事务回滚了之前对 all\_nums 的修改，把 all\_nums 恢复到了 500。最终用户 2 事务读到的数据是 498，读到的是脏数据。所以是属于读脏数据的不一致。

**试题五**

**【问题 1】**

1. J. K 的数值会恢复为 1=3, J=5, K=9

**【问题 2】**

Redo: T2

Undo: T3

**【问题 3】**

由于掉电造成磁盘介质损坏，数据库已经无法启动，恢复过程的步骤如下所示：

- (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。

