

CSP 语言与 SQL 数据库应用程序开发方法

王怀南 (本溪钢铁公司计算机中心)

摘要:本文从 CSP 语言的结构和开发方面入手,对如何开发 SQL 数据库应用程序进行了深入的分析 and 理论解释。

一、前言

通常,在开发一个新的应用程序时,往往会有这样的问题:原有的应用程序中的某些功能块、过程、图表、数据项等不需要修改,仍是新程序所需要的,但由于各单位具体情况不同,如使用的计算机类型不同,操作系统不同等,设计开发新的应用程序就意味着将这些部分要从新编写、组织,存在大量的重复劳动,造成许多不必要的资源浪费。而 CSP 语言 SQL 作为关系型数据库技术的标准已被 ISO9000 所认定,其简捷明了、易于理解的数据处理格式倍受程序员们的青睐。但是 SQL 数据库系统 SQL/DS 需通过 DBS 实用程序进行编程,或将编写的语句嵌入其它如 COBOL、FORTRAN、C 等语言中才可使用,很是不便。而 CSP 语言即可帮助程序员迅速地开发 SQL 应用程序而无需编写 SQL 语句。

二、CSP 语言简介及其结构特征

跨系统产品 CSP 是由美国 IBM 公司开发的用以交互式定义、测试和生产应用程序的非常优秀的第四代计算机语言,它的应用程序开发 CSP/AD 是一种联机、面向终端、项目单驱动的软件产品,可帮助用户高效率地建立应用程序,可以在 IBM 中小型机如 4300 系列、3000 系列、8100 系列及 MVS/TSO 等多种操作系统下运行,还可将在主机环境下开发的应用程序用 EZ-RREP 在 IBM PC 机上生成,用 EZ-RUN 解释执行应用程序。这种程序兼容性和可移植性为使用产品程序提供了很大的便利。

在 CSP 应用程序的开发中,可以分别对每个功能块、过程、图表、数据项和映象(即显示画面)进行定义,它们都存在于 CSP 的特殊数据集 MSL 中,它们都是 CSP 的成员。一个 CSP/AD 系统可以有需要数量的多个 MSL,而每个应用程序只能有一个 MSL,MSL 成员有不同类型,它们是记录、表、程序说明块(PSB)、数据项、映象、映象组、进程、语句组 and 应用程序。一个应用程序,就是把这些成员以适当的处理语句组装起来的框架,一旦组织好后,CSP/AD 就可以对这个程序进行测试,测试完毕后就可以生成一个产品程序,放在 ALF(应用程序装载文件,一个被索引了的 VSAM 文件)中供终端用户使用。同时,组成这个应用程序的成员仍然存在,所生成程序中的各个成员不过是它们副本,因此仍然可以用它们组成其他的应用程序。

CSP/AD 主要包括定义、测试和生成三方面内容,而定义又包括插数据定义、映象定义 and 应用程序定义。

CSP/AD 有七种功能,即程序列表处理、定义、测试、生成、实用程序及文件维护、指导阅读资料 and MSL 选择。在定义一个应用程序时,可以从功能选择项目单中选择功能并作为独立步骤来执行,独立地定义诸如逻辑、映象和数据等适用段,从而使应用程序具有模块化结构。

三、用 CSP 语言开发 SQL 应用程序

在有了对 CSP 语言大致的概念性了解后,下面,我们详细地探讨一个 CSP 是如何与 SQL 相结合进行应用程序开发的。

1. 数据定义

对于开发 SQL 应用程序来说,主要有 SQL 行记录定义 and 数据项定义。

SQL 行记录定义是关键的第一步,它将架起 CSP 与 SQL 数据库之间联系的桥梁,亦即二者间的接口。它明确了 SQL 建表者的标识、表名称 and 表标识。若是多表汇合,则在此定义多个表的名称并以不同的表标识加以区别。其次还有确定数据项定义,这些数据项代表的是关系数据库中的各列在 CSP 里的映射,而不是记录结构中的列,当然同时也要给出这种映射的对应关系。这样就建立起了 CSP 与 SQL 数据库之产数据传递的通道。至此,CSP 便会直接生成 SQL 行记录,即 CSP 所需要的 SQL 数据查询关系 SELECT 语句,我们可以在此对其中 WHERE 子句进行默认选择条件定义或修改定义。实际上,在对 SQL 应用程序定义的全过程中,贯穿始终的都是这种一、一多对应关系的映射和逻辑联结技术概念,这也正是 CSP 与 SQL 得以完美结合运用的理论基础。

2. 进程定义

包括进程选择、进程目标定义、SQL 语句定义 and 进程语句定义。

进程选择共有 14 项,每项都代表着进程完成的单一 I/O 功能,我们可以根据每个进程的不同任务进行选项,例如某进程的任务是从 SQL 数据库中查找出满足条件的一条记录,则进程选择为 INQUIRY。而进程目标便是该进程与 SQL 数据库进行数据交流的某个已定义了的 SQL 行记录。进程选择与目标一经确定,实际上 CSP 就已经自动生成了默认的 SQL 语句定义,我们可以对其进行修改 and 再定义,或者也可为一个 SQLEXEC 进程自定义 SQL 语句,以达到某种特殊目的,比如建立一个新的 SQL 数据库结构等等。一个应用程序是由各处理块的从上向下的结构组成,通过进程语句把各处理块逻辑地组织起来。CSP 进程语句定义分为进程选择前预处理语句 and 进程选择后执行语句两部分,前者赋予目标以参数,后者进行数据处理或控制进程流。

3. 应用程序定义

这是定义的最后步骤,进行应用程序的说明及进程列表定义。这将产生一个新的成员作为整个应用程序的总控制模块,其中包括该应用程序是否为主事务或主批处理的类型说明、工作存储区说明、映象组说明等等,以及所包括的所有进程和语句组的列表、主进程的次序定义。

4. 测试

测试是一项在应用程序定义完成后的非常(转第 49 页)

(接第 42 页)重要的功能,它帮我们检查语法和任何用 CSP/AD 定义的应用程序的逻辑设计以及模拟执行应用程序,并可观察每个语句的处理过程和结果。对于 SQL 应用程序而言,它还要检查用户是否取得对所用的 SQL 数据库的授权以及所授权限制的级别。这里有一个问题需要特别指出:SQL 应用程序测试功能不能模拟分段执行方式。分段方式会影响 SQL 处理,因为在第一段的结尾,所有对数据库的更改都会自动落实到数据库中,这种落实也会释放所有封锁和失去 SCAN(扫描)位置。为了解决这个问题,我们可以通过在应用程序顶部(即主进程语句定义的预处理语句部分)把 EZEENVCN 字设置为 1(这个字是用来在非 DPPX/SP 系统上控制每个 CONVERSE 对话的数据自动落实),强制模拟分段方式的作用,以要求测试功能在第一个 CONVERSE 进程期间落实 SQL 资源。

5. 生成应用程序

这是建立 SQL 应用程序的最后一个步骤。当 SQL 应用程序经反复修改、测试,确认无误后,便可生成我们需要的应用程序执行代码文件,并通过填写应用装载文件名称,将所生成的应用程序装载到指定的 ALF 中去,供用户使用。在生成输出的模块中,除了一般用户程序所具有的用户装载模块(若未建立 CSP 用户表则无此模块)、映象装载模块、应用程序映象模块、数据装载模块、处理装载模块和数据特性半载模块之外,还包括 SQL 语句准备装载模块和 SQL 语句执行装载模块。这两大模块分别给出了识表(SSIT)和缩短了的准备 SQL 进程表。

如果用户是 PC 机用户,那么切勿在主机上进行生成,而要将 MSL 中的源程序取出(EXPORT),经传递接收(RECEIVE)至 PC 硬盘上,然后在 PC 机上启动 EZ-PREP 软件生成一个 EZ-RUN 应用程序,方可在 PC 机上运行。关于 EZPEP 和 EZRUN 的工作原理和方法,在此就不再赘叙。