



工业控制数据库

课程总结



基本概念

(选择题, 单选或多选)

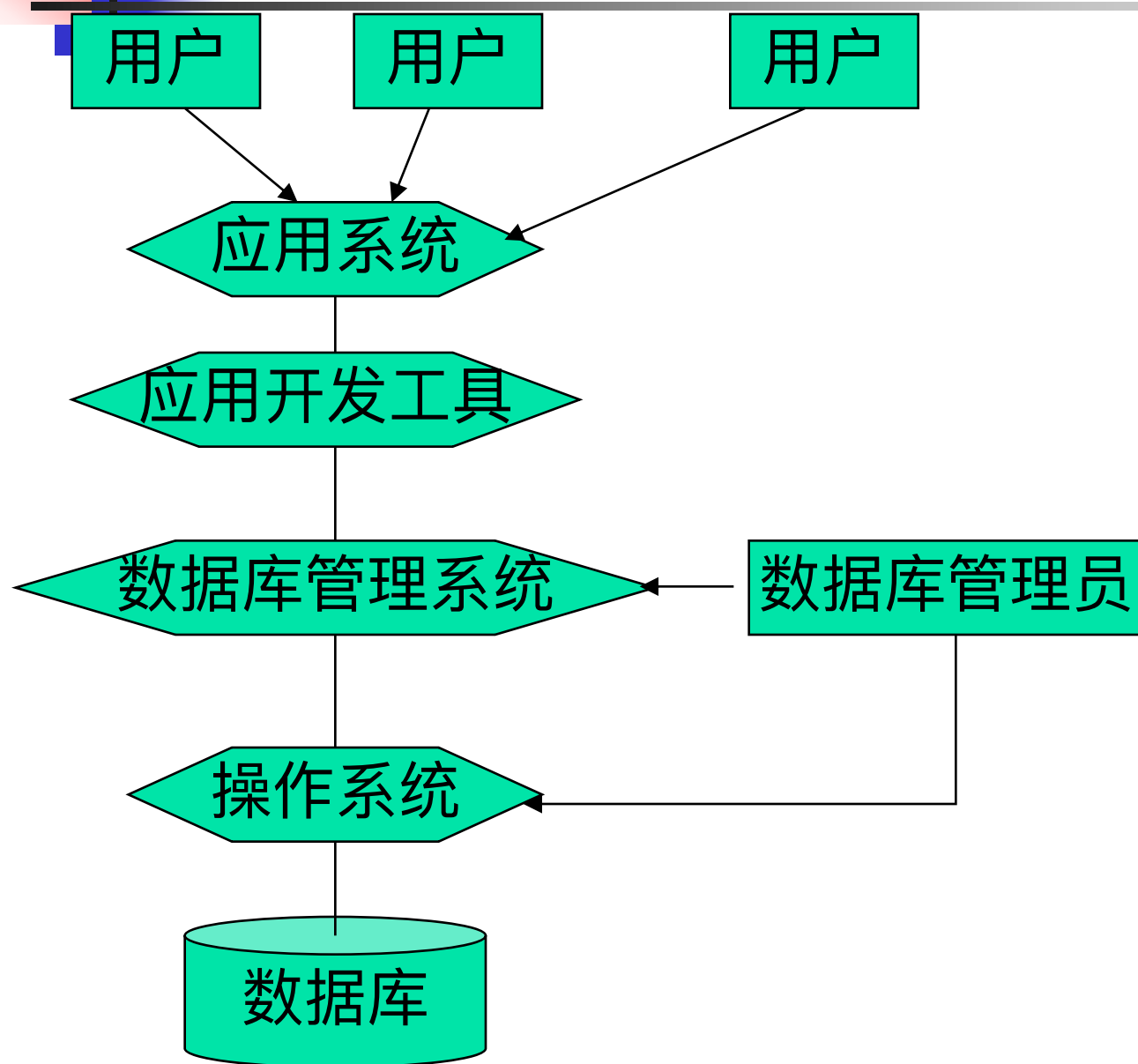
- 数据库系统的概念
- 作用
- 主要关键词:
 - 数据: 描述事物的符号记录, 有多种形式, 它们都可以经过数字化后存入计算机
 - 数据的语义: 数据的含义



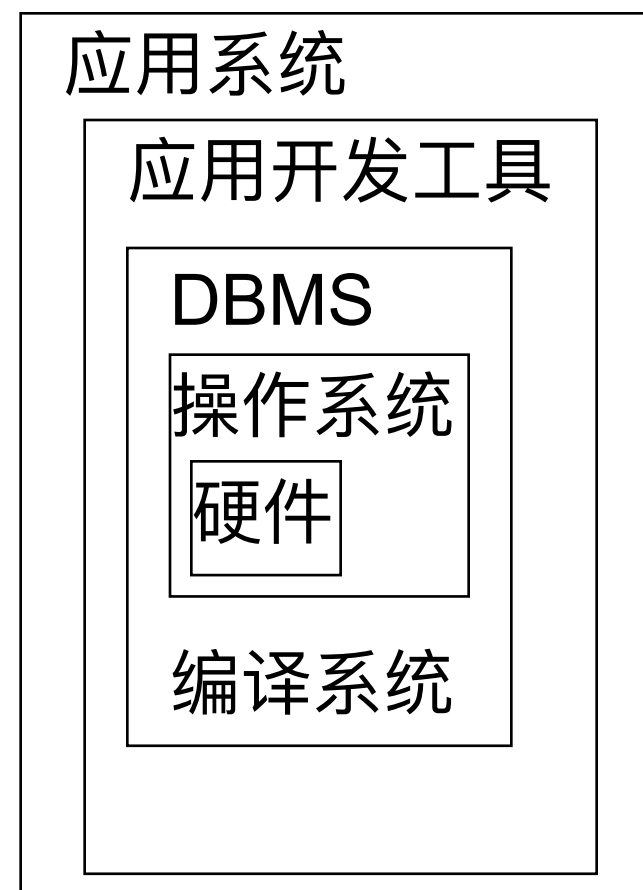
基本概念

- 数据库(DB): 长期存放在计算机内的有组织的可共享的数据集合
- 数据库管理系统(DBMS): 完成数据库的建立、使用和维护功能
- 数据库系统(DBS): 数据库(最底层)+数据库管理系统+应用系统+数据库管理员 +用户, 一般常说的数据库就是DBS
- 数据库管理员(DBA): 专门管理和维护数据库系统的人员

数据库系统 (看图说话)



数据库系统



数据库在计算机系统中的地位



主要的数据库产品

- 主流数据库系统

- Oracle: 美国Oracle公司产品
- Microsoft SQL Server: 美国Microsoft公司产品
- Sybase: 美国Sybase公司产品
- Informix: 美国Informix公司产品

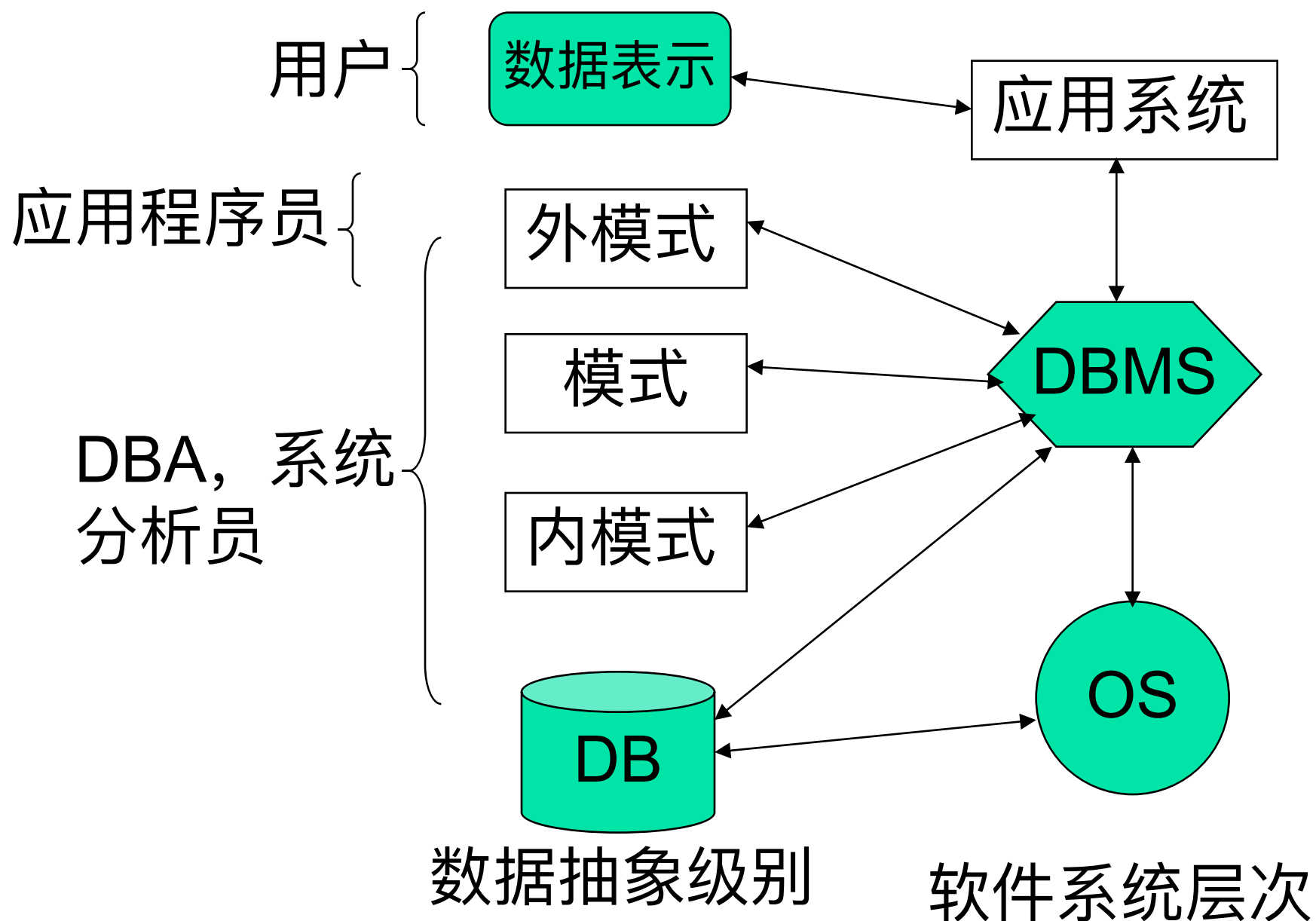
数据库系统结构

- 数据模型：是一组概念的集合，用于对现实世界数据特征进行抽象
- 按应用不同分为：概念模型和数据模型



- 数据库系统结构
 - 从数据库管理系统角度看：三级模式结构，二级映像功能。

数据库系统的组成

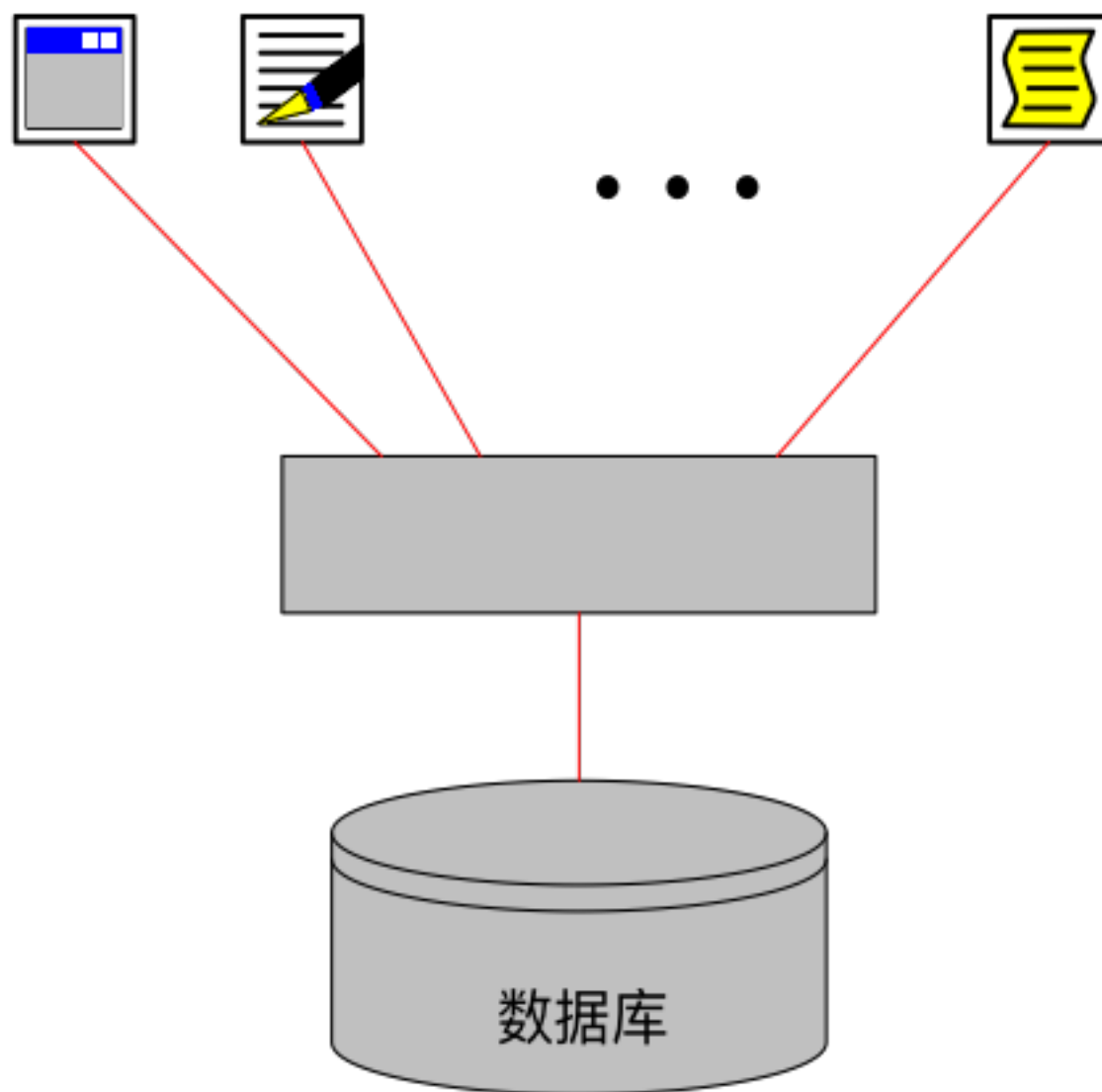


数据库的三级体系结构 (看图说话)

外模式
(单个用户视图)

概念模式
(公共用户视图)

内模式
(存储视图)





数据库技术的发展趋势

- 数据库技术的未来

- 分布式数据库
- 面向对象数据库
- 多媒体数据库
- 主动数据库
- 实时数据库
- **工业应用数据库及其数据挖掘** (应该成为过程全自动化、远程控制的基础)



关系型数据库理论

关系数据库是用表来表示数据的数据库，“关系”只是表的数学术语。关系数据库的数学基础是关系代数。

关系系统基于关系理论，即关系数据模型。体现在如下方面：

- 结构化方面：数据库中的数据对用户来说是表，并且只是表；
- 完整性方面：数据库中的这些表满足一定的完整性约束；
- 操纵性方面：用户可以使用用于表操作的操作符。



数据模型的基本概念

(1) 字段(Field)

对应于信息世界中的属性，也称数据项。字段的命名往往与属性名相同。

(2) 记录(Record)

字段的有序集合称为记录，它用来描述一个实体，是相应于这一实体的数据。例如，组成一个学生记录的字段(数据项)有学号、姓名、年龄和性别字段，这是记录的类型。



数据模型的基本概念

(3) 文件(File)

同一类记录的集合，例如所有的学生记录的集合就是一个学生文件。

(4) 数据模型(Data Model)

实体模型的数据化。

(5) 关键字(Key)

能唯一标识文件中每一个记录的一个或多个字段的最小组合称为关键字。例如学生文件中，学号可以唯一地标识每个学生记录，所以学号是关键字。



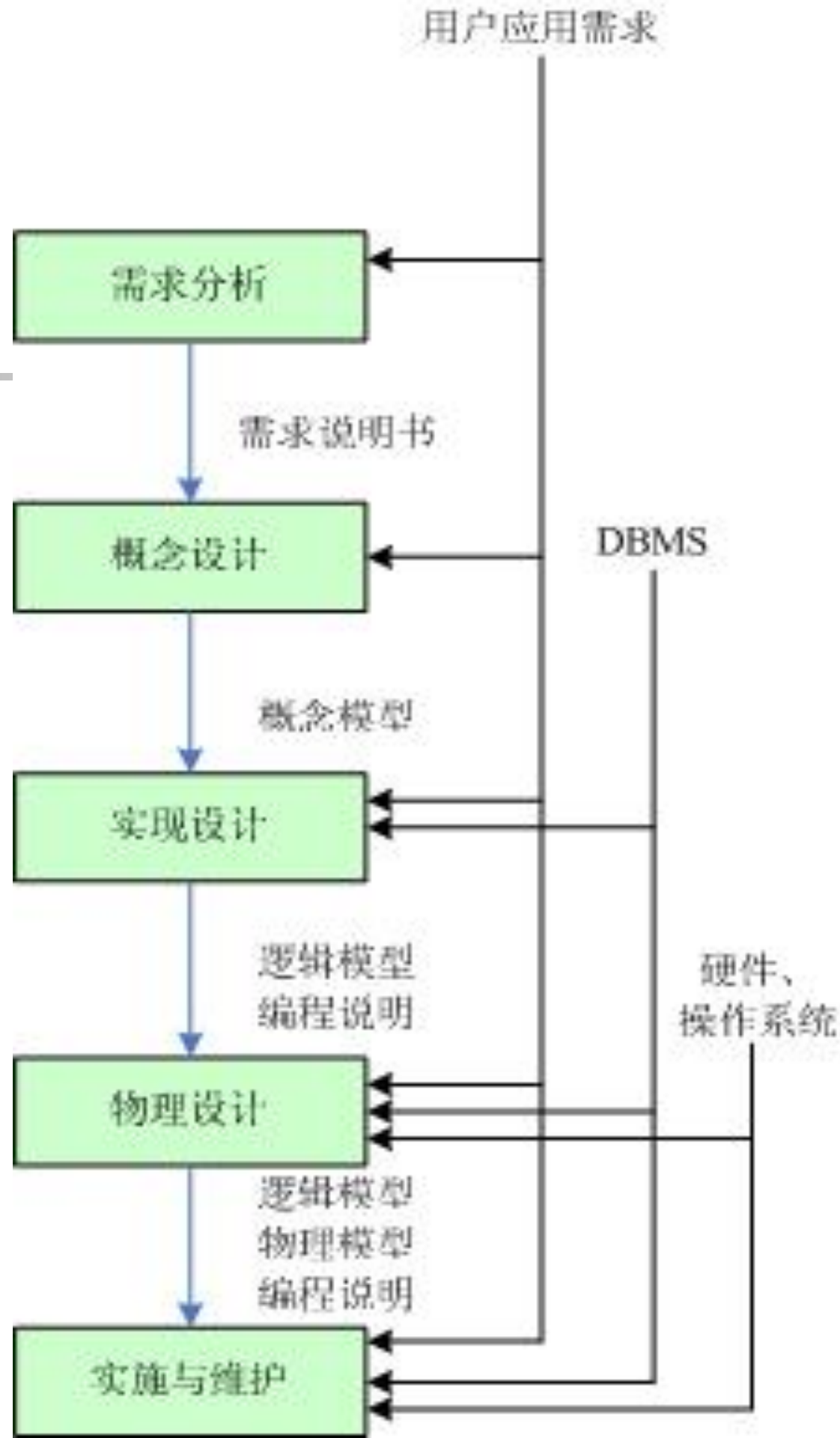
数据库设计方法概述 (结合实例)

分阶段设计方法，该方法遵循自顶向下、逐步求精的原则，将数据库设计过程分解为若干个相互依存的阶段。每一个阶段采用不同的技术与工具，解决不同的问题。

基本步骤：

- (1) 需求分析
- (2) 概念设计
- (3) 实现设计
- (4) 物理设计
- (5) 实施与维护

数据库设计步骤





SQL标准

- ▶ 70年代初 Codd提出了关系演算语言ALPHA，但一直未实现。
- ▶ 美国IBM公司研制了一个面向域的数据查询语言QBE (Query By Example)
- ▶ 美国加利福尼亚大学研制的面向元组的数据语言QUEL
- ▶ SQL (Structured Query Language) 是介于关系代数和关系演算之间的语言，Boyce和Chamberlin于1974年提出的SQUARE基础上发展而成的，目前已成功在大多数DBMS中实现。



SQL标准

- ▶ 1986年10月美国国家标准局（ANSI）批准了SQL作为关系数据库语言的美国标准。
- ▶ 自SQL成为国际标准语言以后，各个数据库厂家纷纷推出各自的SQL软件或与SQL的接口软件。这就使大多数数据库均用SQL作为共同的数据存取语言 and 标准接口，使不同数据库系统之间的互操作有了共同的基础。
- ▶ SQL成为国际标准，对数据库以外的领域也产生了很大影响，有不少软件产品将SQL语言的查询功能与图形功能、软件工程工具、软件开发工具、人工智能程序结合起来。



SQL标准

- ▶ SQL集数据定义、数据查询、数据更新和数据控制于一体，既可作为独立语言由终端用户以联机交互方式使用，也可作为数据子语言嵌入主语言中使用。SQL已经成为目前最主要的数据库操纵和控制语言，比较流程的程序设计工具基本上都支持嵌入式SQL语言。



SQL 语言主要动词含义

SQL功能	动词
■ 数据查询 _(表)	SELECT
■ 数据定义 _(表)	CREATE,DROP,ALTER
■ 数据操作 _(表中内容)	INSERT,UPDATE,DELETE
■ 数据控制 _(授权收回)	GRANT,REVOKE

单表查询

出现在where后面的运算符或者SQL定义的保留字

□ 查询满足条件的元组

常用查询条件

查询条件	谓词
比较	=, >, <, >=, <=, !=, <>, !>, !< not + 上述比较运算符
确定范围	between... and..., not between... and...
确定集合	in, not in
字符匹配	like, not like
空值	is null, is not null
多重条件	and, or



单表查询——一般查询

□ 一般查询

- 查询年龄在20岁以下的学生姓名及其年龄

```
select  Sname, Sage
from  Sduents
where  Sage < 20;
```

查询年龄在20~23岁之间的学生姓名、所在系，年龄

```
select  Sname, Sdept, Sage
from  Students
where  Sage between 20 and 23;
```



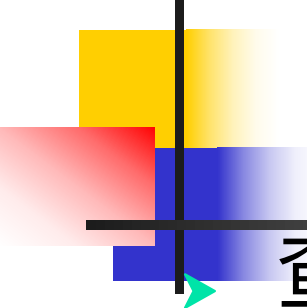
单表查询—‘in’, ‘%’查询

查询信息系、数学系和计算机系学生的姓名和性别

```
select Sname, Ssex  
from Students  
where Sdept in ('IS', 'MA', 'CS');
```

➤ 查询所有姓刘的学生的姓名、学号和性别

```
select Sname, Sno, Ssex  
from Students  
where Sname like '刘%';
```



单表查询—‘_’，条件

▶ 查询姓欧阳且全名为三个汉字的学生姓名

```
select Sname  
  from Students  
 where Sname like '欧阳__';
```

▶ 查询DB_Design课程号和学分

```
select Cno, Ccredit  
  from Course  
 where Cname like 'DB_ Design';
```

?



单表查询—‘_’, ‘is’条件

```
select Cno, Ccredit
```

```
from Course
```

```
where Cname like 'DB\_  Design' escape '\';
```

➤ 查询缺考学生的学号和课号

```
select Sno, Cno
```

```
from SC
```

```
where Grade is null;
```

这里is 可否用=代替?



单表查询—多条件查询

▶ 查询计算机系年龄在20岁以下的学生的姓名

```
select Sname  
  from Students  
 where Sdept='CS' and Sage<20;
```

▶ 查询信息系、数学系和计算机系学生的姓名和性别

```
select Sname, Ssex  
  from Students  
 where Sdept='CS' or Sdept='IS' or Sdept='MA'
```




单表查询－查询结果排序

- 对查询结果排序

- 例子：查询选修了3号课程的学生学号和成绩，要求查询结果按成绩降序排列

```
select Sno, Grade  
from SC  
where Cno='3'  
order by grade desc;
```



单表查询—查询结果排序

例子：查询全体学生的情况，查询结果按系号升序排列，同一系的学生按年龄降序排列

```
select *  
  from SC  
 order by Sdept, Sage desc;
```

❑ 使用集函数

➤ 例子：查询学生总数

```
select count(*)  
  from Students;
```

单表查询—对查询结果分组

- 对查询结果分组: 将查询结果按一列或多列值分组, 目的是将**集函数**作用到组上(例如: 小计)
- 例子: 求每门课的课号及其选课人数

```
select Cno, count(Sno)
from SC
group by Cno;
```

- 例子: 求选课人数超过10人的课程号及其人数

```
select Cno, count(Sno)
from SC
group by Cno
having count(Sno)>10;
```

组内条件



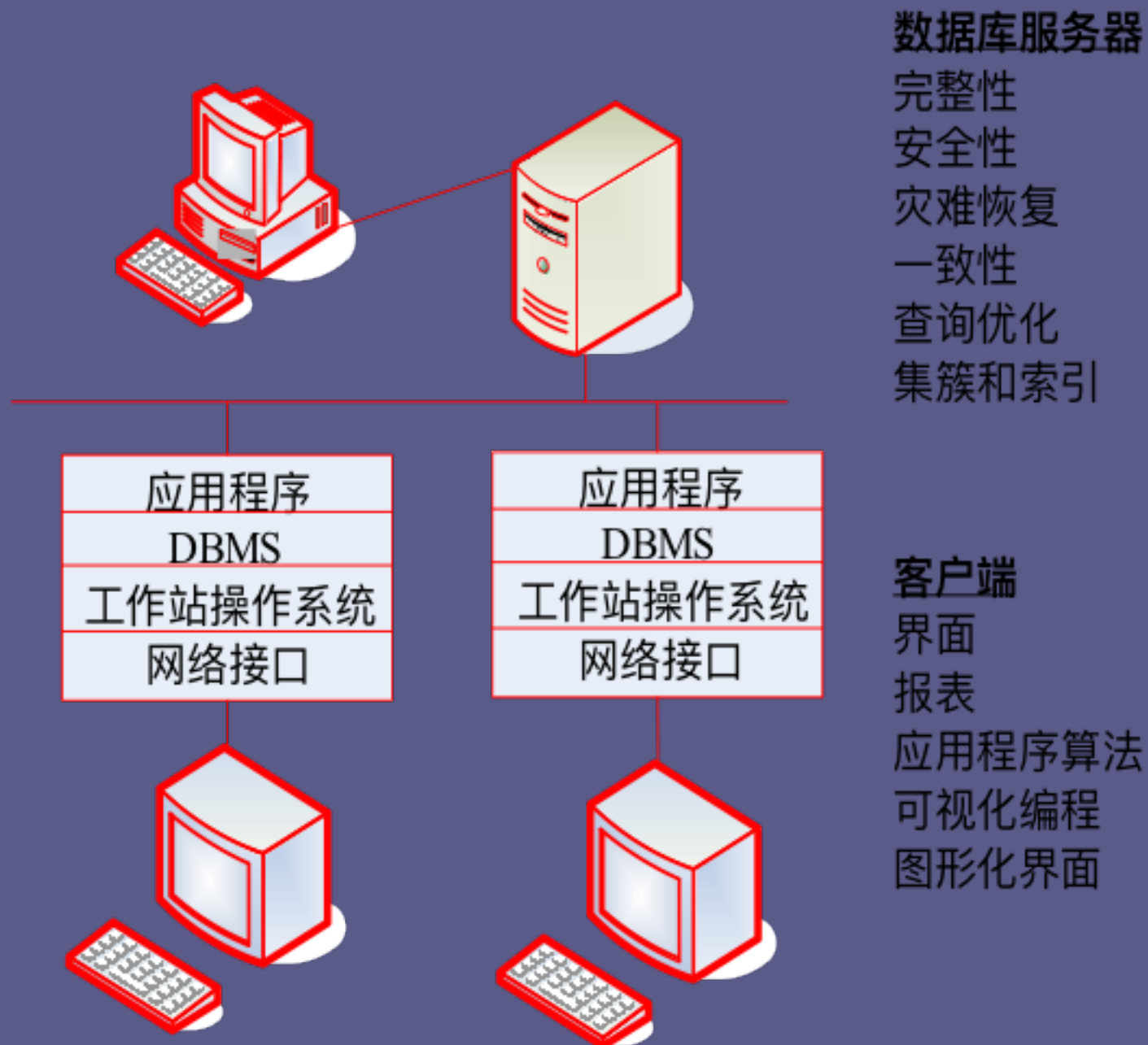
不相关子查询

❑ 查询选修了课程名为信息系统的学生学号和姓名

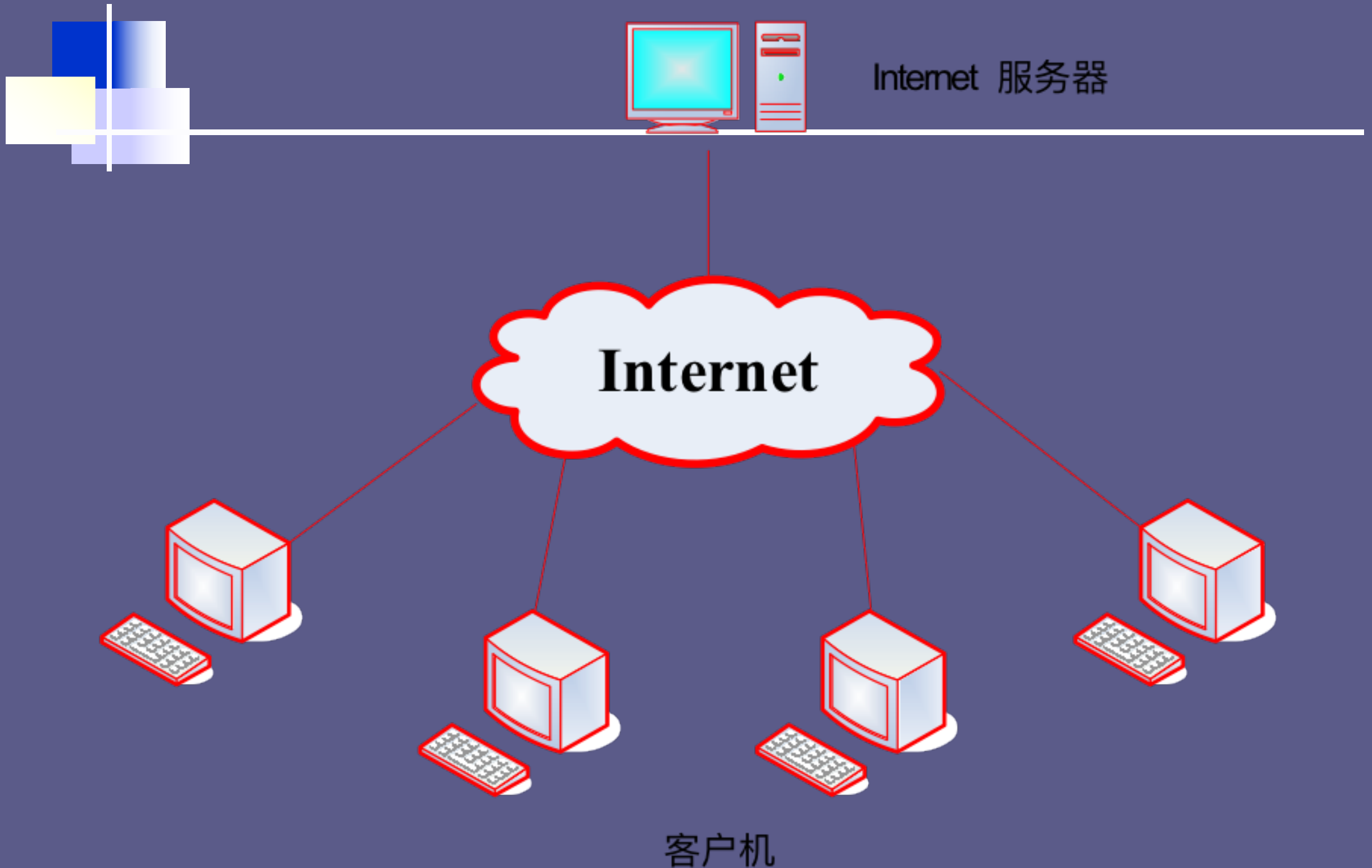
```
select Sno, Sname  
  from Students  
 where Sno in (select Sno  
                from SC  
               where Cno in (select Cno  
                              from Courses  
                              where Cname='信息系统'));
```

● C/S结构 (client/sever)

➤ C/S体系是随着人们对数据处理的要求应运而生的。目前，C/S体系是当今世界上最为流行的体系结构。



• B/S结构 (browser/server)

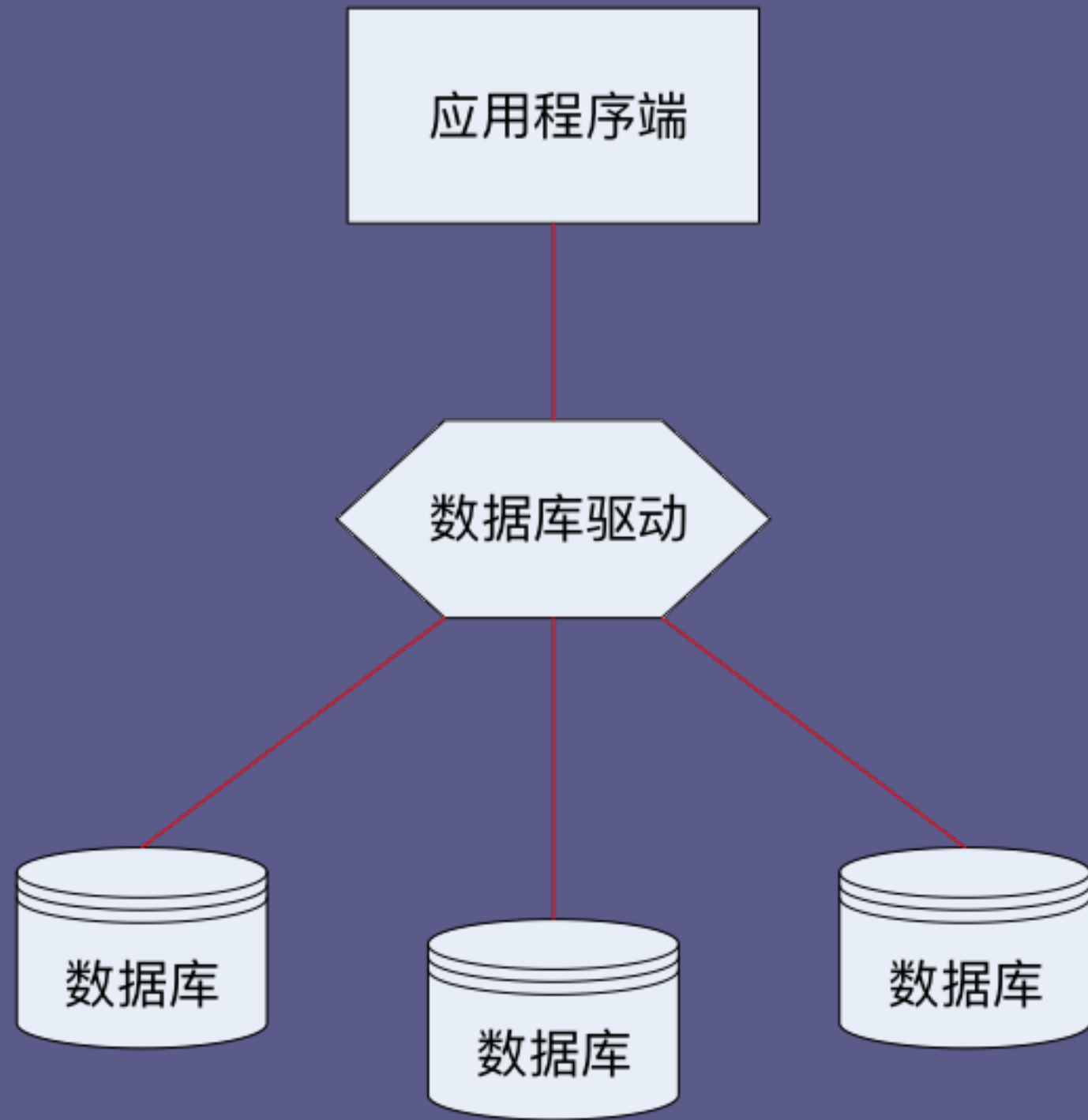


● B/S结构

- ▶ 促使Internet的飞速发展的原因之一就是它的简单性和方便性，人们只要通过一个通用的浏览器（如Microsoft公司的Internet Explorer，简称IE），就可以浏览网络任何共享资源。也就是人们常说的“瘦客户”，在客户机上没有必要在安装一些额外的客户端软件就可以工作。
- ▶ Internet网络上共享的信息和软件全部存储在Internet服务器上，软件和信息更新只要在Internet服务器上面进行更新之后，所有访问这台服务器的客户端都自动显示更新的内容。

- **三层体系结构**

- 软件人员为了使自己开发的软件系统与不同的DBMS连接，需要一种接口来使数据库开放，从而让开发人员编写不依赖于任何DBMS的应用程序，这种数据库接口通常被称为开放数据连接。





ODBC

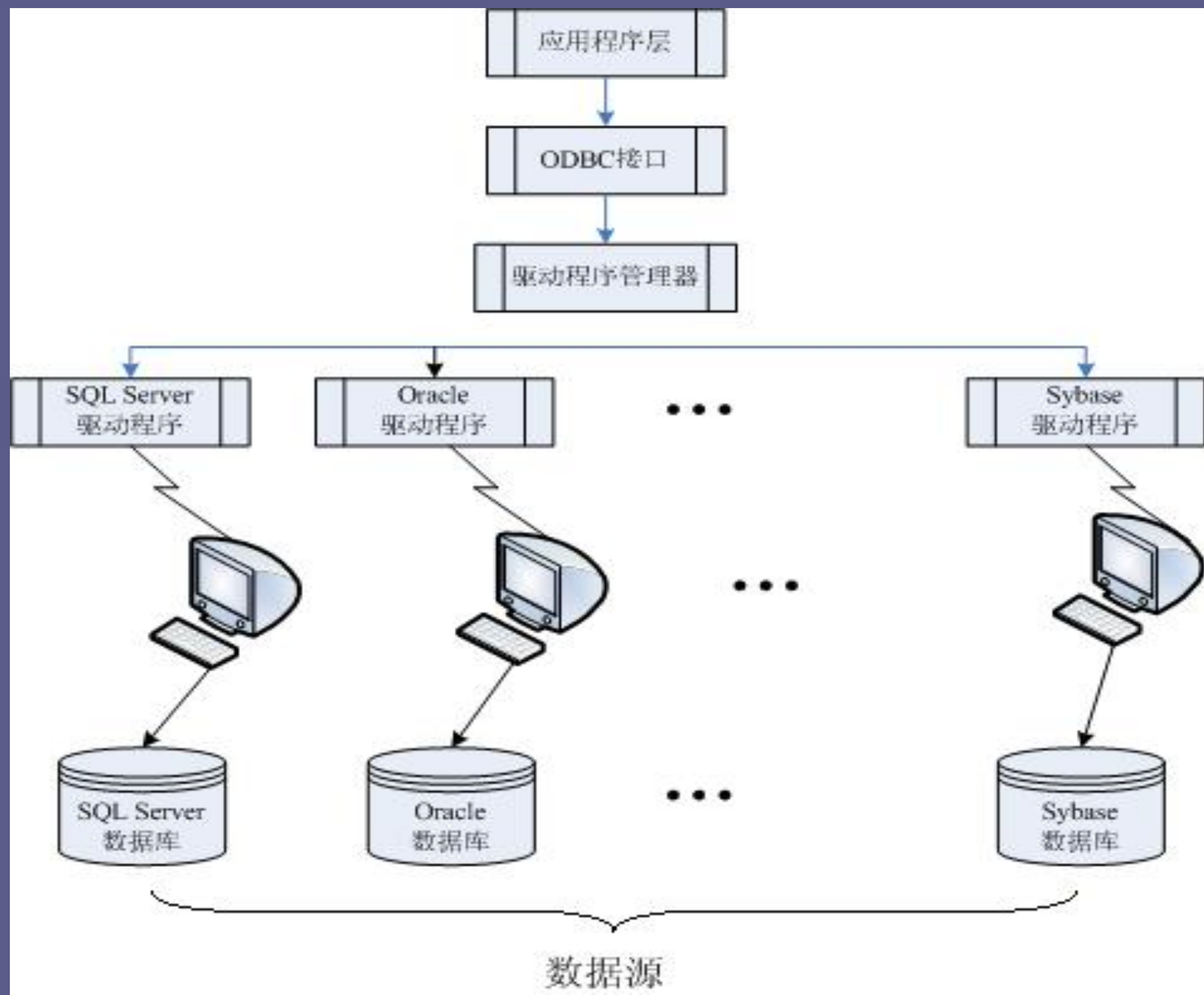
- ODBC(Open Database Connectivity, 开放数据库互连)是微软公司开放服务结构(WOSA, Windows Open Services Architecture)中有关数据库的一个组成部分, ODBC技术是PC领域内最流行的数据库接口之一, 并且正在逐渐地扩展到其它平台。
- 它建立了一组规范, 并提供了一组对数据库访问的标准API(应用程序编程接口)。这些API利用SQL来完成其大部分任务。ODBC本身也提供了对SQL语言的支持, 用户可以直接将SQL语句送给ODBC。



ODBC

- 一个基于ODBC的应用程序对数据库的操作不依赖任何DBMS，不直接与DBMS打交道，所有的数据库操作由对应的DBMS的ODBC驱动程序完成。也就是说，不论是Oracle、Access还是SQL Server数据库，均可用ODBC API进行访问。由此可见，ODBC的最大优点是能以统一的方式处理所有的数据库。

● ODBC的分层次结构



- ODBC主要部件

- ODBC API

- ODBC 驱动程序。是一些DLL，提供了ODBC和数据库之间的接口

- 数据源。数据源包含了数据库位置和数据库类型等信息，实际上是一种数据连接的抽象。

通用ODBC API应用结构

ODBC API是由一组函数调用组成的，其核心是SQL。ODBC函数的主要功能就是将SQL语句发送到目标数据库中，然后处理这些SQL语句返回的结果。使用ODBC的基本步骤：

- (1) 为ODBC分配环境句柄；
- (2) 分配一个连接句柄；
- (3) 连接到数据库；
- (4) 用SQL命令分配一个语句句柄；
- (5) 传送该命令；
- (6) 关闭连接；
- (7) 解除连接和环境句柄。

ODBC 初始化函数

SQLAllocHandle (ENV)
SQLSetEnvAttr
SQLAllocHandle (DBC)
SQLConnect
SQLSetConnectAttr
SQLGetInfo
SQLAllocHandle (STMT)
SQLSetStmtAttr

使用 ODBC API
检索数据

SQLExecDirect
OR SQLPrepare
SQLExecute

执行 SQL 命令

Select
OR Update/Delete/Insert

使用结果集

SQLNumResultCols
SQLBindCOL
SQLFetch
SQLGetData
SQLCloseCursor

释放空间

SQLFreeHandle (STMT)
SQLDisconnect
SQLFreeHandle (DBC)
SQLFreeHandle (ENV)

表 1.ODBC API 应用的结构

典型的ODBC步骤

1.分配环境和连接句柄

SQLAllocEnv SQLAllocConnect

2.数据库连接设置

SQLSetConnectOption

3.连接到数据库

SQLConnect

4.语句处理

SQLAllocStmt 分配语句句柄

SQLExecDirect 执行SQL语句

5.处理结果

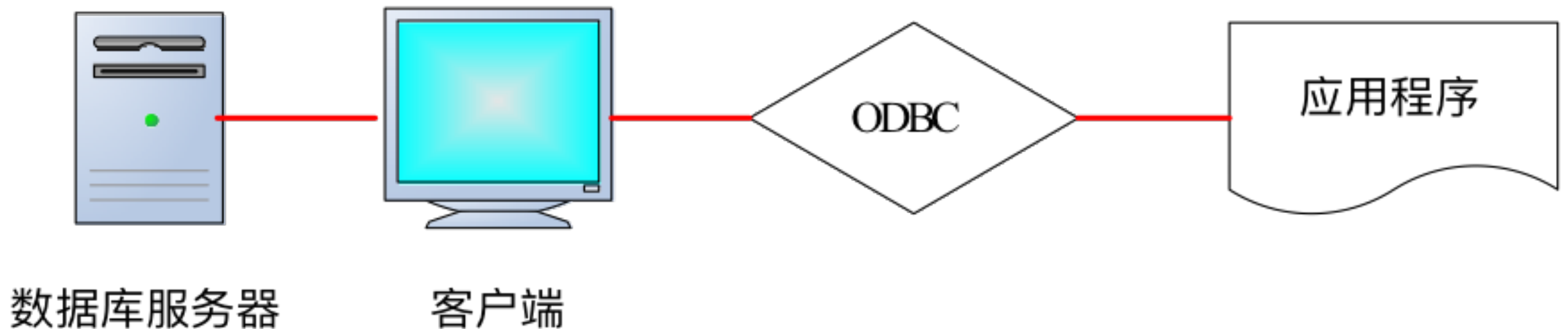
SQLBindColumn 绑定结果 SQLFetch 获取一个数据行

6.释放即清除

SQLFreeStmt SQLDisconnect SQLFreeConnect
SQLFreeEnv

ODBC的配置

- ▶ 安装数据库服务器
- ▶ 控制面板－管理工具－ODBC数据源
- ▶ 配置ODBC数据源





现代数据库的研究领域

- ▶ 整体系统方面：面向对象数据库、时态数据库、实时数据库、主动数据库
- ▶ 体系结构方面：内存数据库、分布式数据库、并行数据库、数据仓库
- ▶ 应用方面：工程数据库、科学与统计数据库、超文档数据库、演绎数据库等



分布式数据库技术

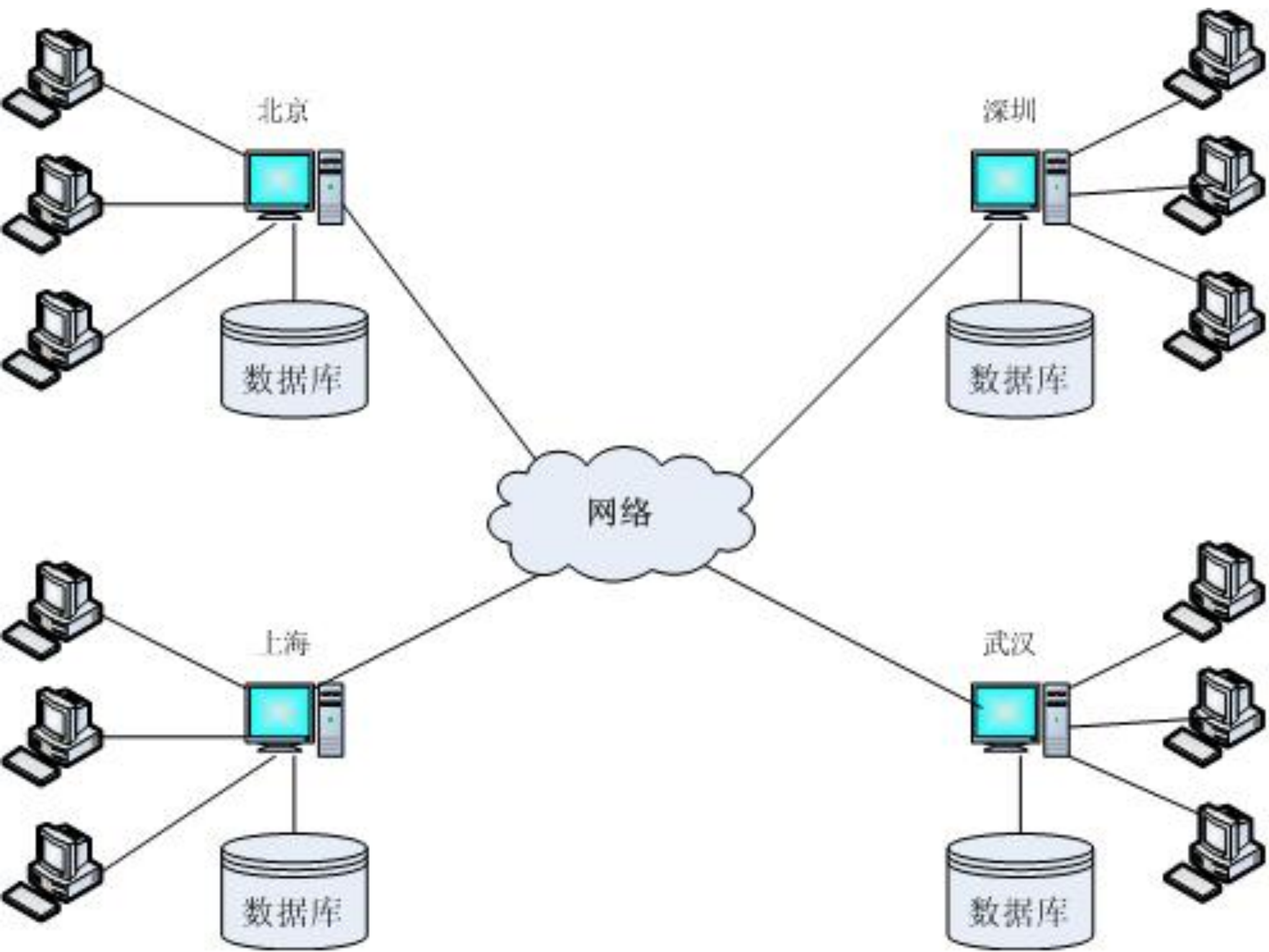
- 分布式数据库自20世纪80年代以来一直是数据库与信息系统中的—一个重要而热门领域，是数据库和网络技术相互渗透有机结合的结果。克服了集中式数据库的许多缺点，适合于—般企业的分散型结构。
- —个分布式数据库由—系列的场地组成，通过某种通信网络连接在—起
- 每个场地自身有—个完全的数据库系统
- 所有场地都可以协同工作



分布式数据库技术

- 分布式数据库系统是数据库技术和网络技术两者相互渗透和有机结合的结果。
- **概念**：分布式数据库由一组数据组成，这些数据物理上分布在计算机网络的不同结点上，逻辑上属于同一个系统。
- **特点**：分布性、逻辑整体性

分布式数据库技术



- 所谓的“分布式数据库”实际上是一种虚拟的数据库，它的各个组成部分物理地存储在许多不同场地上的不同数据库中。



为什么要实施分布式数据库

- 企业自身经常就是分布式的，一个企业在逻辑上和地理上被分为公司、部门、工作组等等。
- 每个部门都会有自己的数据。这样，企业的整个信息资源就被分裂成信息孤岛。
- 分布式数据库系统正是为把这些孤岛联系在一起的桥梁。
- 使数据库的结构能够反映企业的结构。



实时数据库技术

- 现实世界中，有许多应用包含了对数据的“定时”存取和对“短暂有效”数据的存取。这些应用一方面需要维护大量共享数据和控制知识；另一方面其应用活动有很强的时间性，要求在一定的时刻和一定的时间期限内从外部采集数据，按比促之间的联系存取已获得的数据和处理采集的数据，再及时作出响应。
- 同时需要数据库技术和实时数据处理技术



实时数据库技术

- 实时数据库系统就是其事务和数据都可以具有定时特性或显式的定时限制的数据库系统。
- 数据库研究工作的动机在于利用数据库技术的特点或优点来解决实时系统中的数据管理问题；实时系统研究工作则为给实时数据库系统提供时间驱动调度和资源分配算法。



当前主流的实时数据库产品

- OSI PI
- Honeywell PHD
- AspenTech Infoplus.21
- Wonderware InSQL



衡量一个实时数据库产品的标准

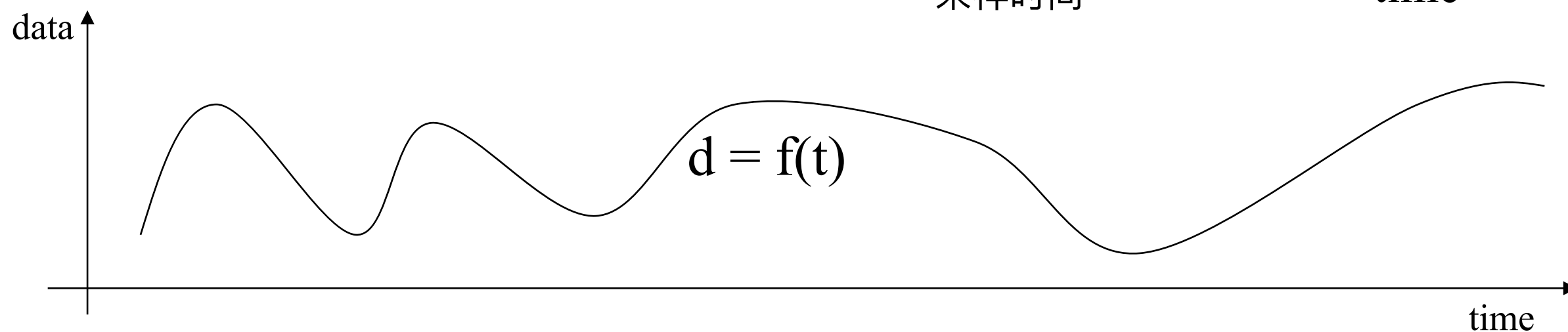
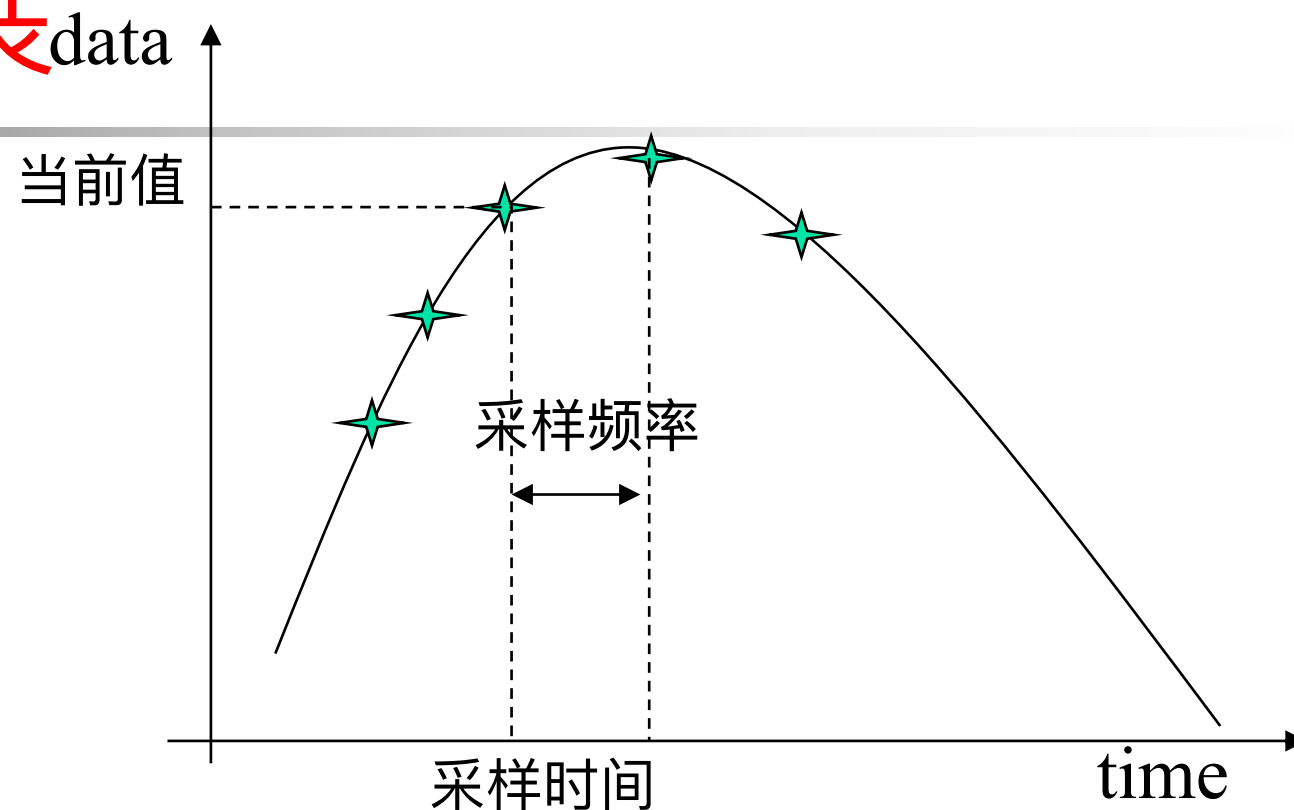
- 性能和稳定性
- 服务器端管理工具
- 客户端工具
- 数据收集接口
- 与关系型数据库和管理软件的连接
- 支持的数据类型

技术的发展

dv : 当前值

dt : 采样时间

de : 有效期 (采集频率)





实时数据库的产生

RealTime Database \neq

Database

+

RealTime Operating System



实时数据库的特点

- 数据的时效性

- 一个数据对象有三个主要属性：当前值、采样时间、采样频率

- 事务的定时限制

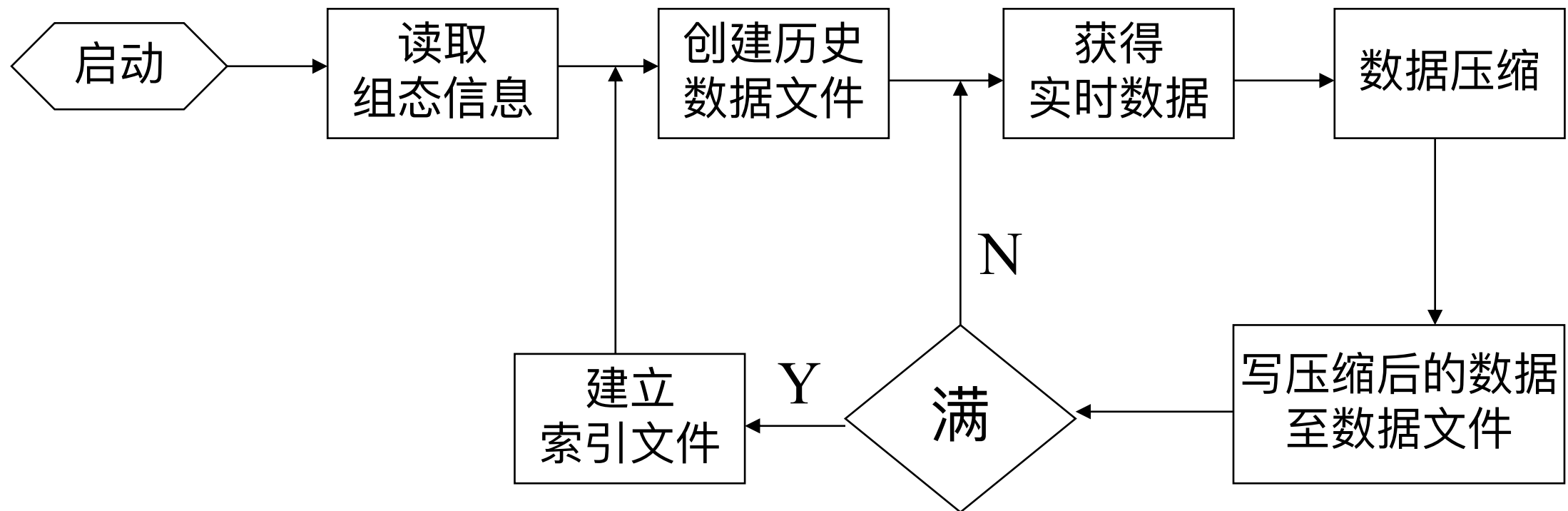
- 实时数据处理（即实时事务）有定时（定时可以是绝对、相对或周期时间）限制



内存中数据的管理

- 内存中的数据
 - 静态数据
 - 动态数据
- 内存管理
 - 数据缓存
 - 程序缓存
- 内存和磁盘数据的交换

历史数据处理过程





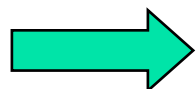
数据压缩



■ 有规律数据

逻辑压缩

- 在一段时间内，相邻的数据在精度允许范围内，常常会满足某种线性规律，如果我们利用这个规律，就可以对采集来的数据进行压缩处理，不必要逐点保存。



■ 无规律数据

物理压缩

- 数据的变化是随机的，无规律可寻。如设备发生故障时产生的数据。

数据源（设备）



- DCS
- PLC
- SCADA、HMI
- 常规仪表、智能仪表
- 在线或离线分析仪器
- 巡检仪



几种常见的数据接口

- OPC
- DDE、NETDDE
- RS-232 422 485
- TCP
- UDP



工业过程的分类

- 工业过程大致可以分为：
 - ▶ **连续过程工业**：主要包括石油化工、炼油、电力、冶金、化工、建材、制药、造纸等。连续流动着的物料，可测量，但不可点数，有误差
 - ▶ **离散过程工业**：机械制造业、汽车装配等。单独的零部件组成最终产成品的方式。因为其产成品的最终形成是以零部件的拼装为主要工序，所以装配自然就成了重点
 - ▶ **间歇过程工业**：批量生产过程



存储过程的概念

- 存储过程Stored Procedure 是一组为了完成特定功能的SQL 语句集，经编译后存储在数据库中。用户通过指定存储过程的名字并给出参数（如果该存储过程带有参数）来执行它。



存储过程的概念

- 在SQL Server 的系列版本中存储过程分为两类：系统提供的存储过程和用户自定义存储过程。

- 系统过程主要存储在master 数据库中并以sp_ 为前缀，并且系统存储过程主要是从系统表中获取信息，从而为系统管理员管理SQL Server 提供支持。通过系统存储过程，MS SQL Server 中的许多管理性或信息性的活动（如了解数据库对象、数据库信息）都可以被顺利有效地完成。

- 系统存储过程也可以在其它数据库进行调用，当创建一个新数据库时，一些系统存储过程会在新数据库中被自动创建。

- 用户自定义存储过程是由用户创建并能完成某一特定功能（如查询用户所需数据信息）的存储过程。



2. 举例：带参数的存储过程

- *Use pubs*

- *go*

- *create procedure test_liuxg1*

- *@name_liu varchar(20)*

- *as*

- *select **

- *From publishers*

- *where pub_name=@name*

- *Go*

- 执行：

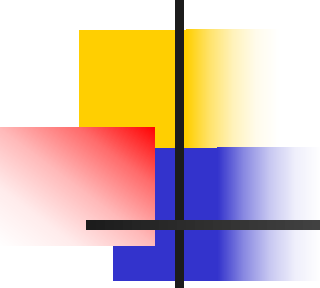
- *Exec student_infor @name='GGG&G'*

触发器

触发器概念及作用

- 触发器是一种特殊类型的存储过程。
- 两者的区别是：触发器主要是通过事件进行触发而被执行，而存储过程可以通过存储过程名称而被直接调用。
- 触发器的主要作用是它能够实现由主键和外键所不能保证的复杂的参照完整性和数据的一致性。

触发器主要优点如下：

- 
- ❖ 触发器是自动的：当对表中的数据作了任何修改（比如手工输入或者应用程序采取的操作）之后立即被激活。
 - ❖ 触发器可以通过数据库中的相关表进行层叠更改。
 - ❖ 触发器可以强制限制，这些限制比用CHECK 约束所定义的更复杂，功能更强大。



SQL Server数据库管理系统

- 工作环境：
- 建库、建表、权限管理、服务器管理、分离与附加、备份与还原、存储过程管理
- SQL语句、存储过程



SQL Server数据库管理系统

- 通过第三方平台来连接数据库系统
- 数据环境设计器
- 数据控件
- 数据报表设计器



SQL Server数据库应用

- 简单描述设计和开发一个工厂设备管理软件应考虑哪些方面？
- 注意从软件工程、三层体系结构出发，将实际工程背景结合进行讨论