

数据库系统原理与设计

(第 3 版)

第 1 章 数据库系统概论

目 录



数据库系统的作用



数据模型



数据抽象与数据库三级模式



数据库系统

- 1.1.1 数据与数据管理
 - 1.1.2 数据管理技术的产生和发展
 - 1.1.3 数据库应用
-

1.1.1 数据与数据管理

■ 数据：描述事物的符号记录。

- 无结构的文本形式描述：李天乐，男，1968年9月出生，江西南昌人，现工作于江西财经大学信息学院，教授，主要研究兴趣包括XML数据库、Web数据管理。

➤ 显然，数据的表现形式不能完全表达其内容，其含义即语义需要经过解释才能被正确理解，因此数据和关于数据的解释是不可分的。

- 表格形式（有结构的记录形式）描述：

姓名	性别	出生年月	籍贯	工作单位 / 部门	职称	研究方向
李天乐	男	1968年9月	江西南昌	江西财经大学 信息学院	教授	XML数据库、Web数据管理
...						

➤ 表中一行数据组织在一起便构成一条记录，其数据的语义已由其所在列的表头栏目名解释，因此表格描述的数据称为结构化数据。

1.1.1 数据与数据管理

- 数据处理：从大量的、可能是杂乱无章的、难以理解的数据中**抽取并推导出**对于某些特定的人们来说有价值、有意义的数据。
- 数据管理：对数据进行有效的**分类、组织、编码、存储、检索、维护和应用**——**数据处理的中心问题**。
- **数据管理技术**的发展及其应用的广度和深度，极大地影响着人类社会发展的进程。

1.1.2 数据库技术的产生与发展

■ 人工管理阶段

- 20 世纪 50

- 计算机还仅用于科学计算。

- 数据是面向应用程序，程序与数据之间没有分离。

- 数据需要由应用程序专门负责维护。

- 当多个应用程序分时运行时，应用程序之间

此存在大量冗余数据。



图 1-1 人工管理阶段应用程序与数据之间的对应关系

1.1.2 数据库技术的产生与发展

■ 文件系统阶段

- 20 世纪 50 年代
- 计算机除了能进行数值运算外，还能进行文字、图形、声音、图像等非数值数据的处理。
- 在操作系统中，数据由专门的数据存取方法存取。
- 对于一个特定的应用，程序员需要编写特定的应用程序，以实现对文件的存取。
- 文件系统阶段程序与数据之间的关系如图 1-2 所示。

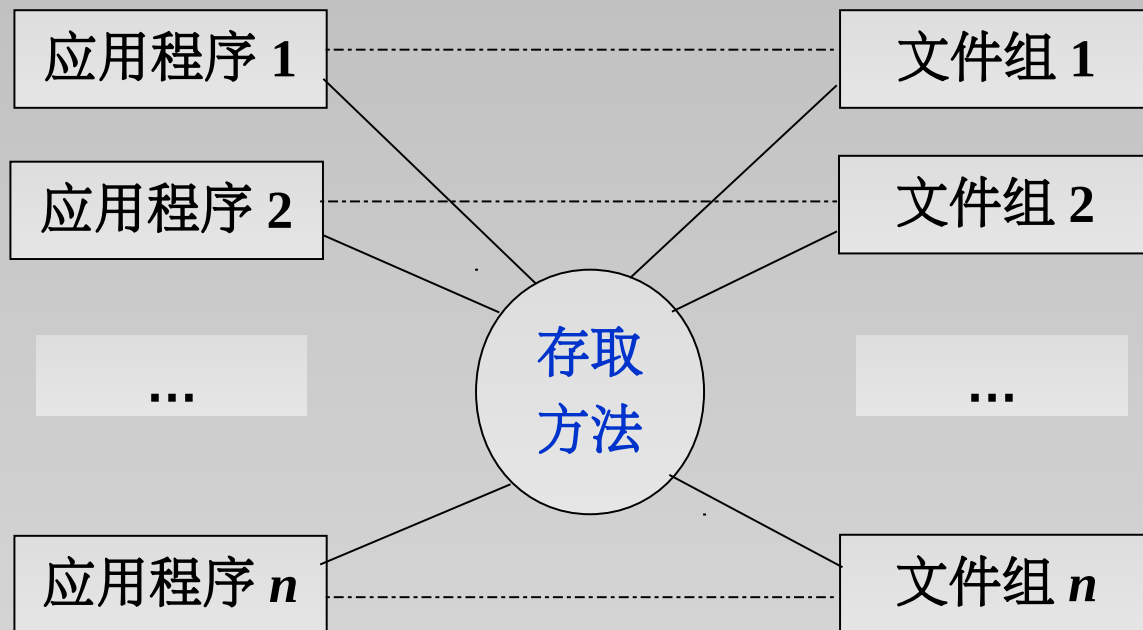


图 1-2 文件系统阶段应用程序与数据之间的对应关系

1.1.2 数据库技术的产生与发展

■ 文件系统的主要特点

- 文件系统实现了文件内的结构性，即一个文件内的数据是按记录进行组织的，这样的数据是有结构的（语义的）。

学生文件 Student 的记录结构

学号	姓名	性别	出生日期	所学专业	家庭住址	联系电话
----	----	----	------	------	------	------

课程文件 Course 的记录结构

课程号	课程名称	学时	学分	教材名称
CS005	数据库原理	64	4	数据库概论
CS012	操作系统	80	5	操作系统原理
.....

1.1.2 数据库技术的产生与发展

■ 文件系统的弊端

● 数据共享性差，数据冗余和不一致

- 数据冗余是指相同的数据在不同的地方 (文件) 重复存储
- 文件系统中的 (或一组) 文件基本上对应于一个应用程序，不同应用程序之间很难共享相同数据
- 如何有效地提高不同应用共享数据的能力成为急需解决的问题

● 数据独立性差

- 文件系统中的文件组是为某一特定应用服务的，其逻辑结构对于该特定应用程序来说是优化的，但系统也不易扩充
- 数据与应用程序之间缺乏逻辑独立性
- 如何有效地提高数据与应用程序之间的独立性成为急需解决的问题

● 数据孤立，数据获取困难

- 对于数据与数据之间的联系，文件系统仍缺乏有效的管理手段
- 如何有效地管理数据与数据之间的联系成为急需解决的问题

1.1.2 数据库技术的产生与发展

■ 文件系统的弊端

● 完整性问题

- 数据的完整性是指数据的正确性、有效性和相容性，也称为一致性约束
- 例如，一个学生需要选修某门课程，该学生必须已经修过了该课程规定的先修课程时才能选修（因为课程之间存在先修后修关系）；必须在该教学班尚未选满时才能选修（因为教室容量有限）；必须在时间上与其它已经选修的课程不冲突时才能选修
- 如何有效地表达和实现一致性约束成为急需解决的问题

● 安全性问题

- 一个系统可能有很多用户，不同用户可能只允许其访问一部分数据，即该用户只有一部分数据的访问权限
- 如何有效地保障数据的安全性就成为急需解决的问题

1.1.2 数据库技术的产生与发展

■ 文件系统的弊端

● 原子性问题

- 计算机系统有时会发生故障，一旦故障发生并被检测到，数据就应该恢复到故障发生前的状态
- 例如，学生选课时，不仅要在选课文件中增加某学生选修某门课的记录，同时也要在该课程教学班记录中将已选课人数加 1，以便学生选课时进行容量控制
- 因此，增加选课记录与选课人数加 1 两个操作要么都发生，要么都不发生，这就是学生选课操作的原子性要求
- 如何有效地保障操作的原子性就成为急需解决的问题

● 并发访问异常

- 系统应该允许多个用户同时访问数据，在这样的环境中由于并发更新操作相互影响，可能会导致数据的不一致
- 如何有效地控制并发操作的正确性就成为急需解决的问题

1.1.2 数据库技术的产生与发展

■ 文件系统的弊端小结

- 数据独立性差
- 数据孤立，数据获取困难
- 数据共享性差，数据冗余和不一致
- 完整性问题
- 安全性问题
- 原子性问题
- 并发访问异常
- ...

1.1.2 数据库技术的产生与发展

■ 数据库管理系统阶段

- 20 世纪 60 年代后期以来
- 数据管理对象的规模越来越大，应用范围越来越广，多种应用共享数据的要求越来越强烈
- 数据库管理系统 (DBMS) 是由一个相互关联的数据的集合和一组用以访问、管理和控制这些数据的程序组成
- 这个数据集合通常称为数据库 (database, DB)，其中包含了关于某个企业信息系统的信息
- DBMS 是位于用户与操作系统之间的一层数据管理软件，它提供一个可以方便且高效地存取、管理和控制数据库信息的环境
- DBMS 和操作系统一样，都是计算机的基础软件（系统软件），也是一个大型复杂的软件系统

1.1.2 数据库技术的产生与发展

- 设计数据库管理系统的目的是为了有效地管理大量的数据，既涉及到数据存储结构的定义，又涉及到数据操作机制的提供
- 解决文件处理系统中存在的问题：
 - 数据共享性差 (数据冗余和不一致)
 - 数据独立性差
 - 数据孤立和数据获取困难
 - 完整性问题
 - 原子性问题
 - 并发访问异常
 - 安全性问题

1.1.2 数据库技术的产生与发展

■ 数据库管理系统的主要特点



1.1.2 数据库技术的产生与发展

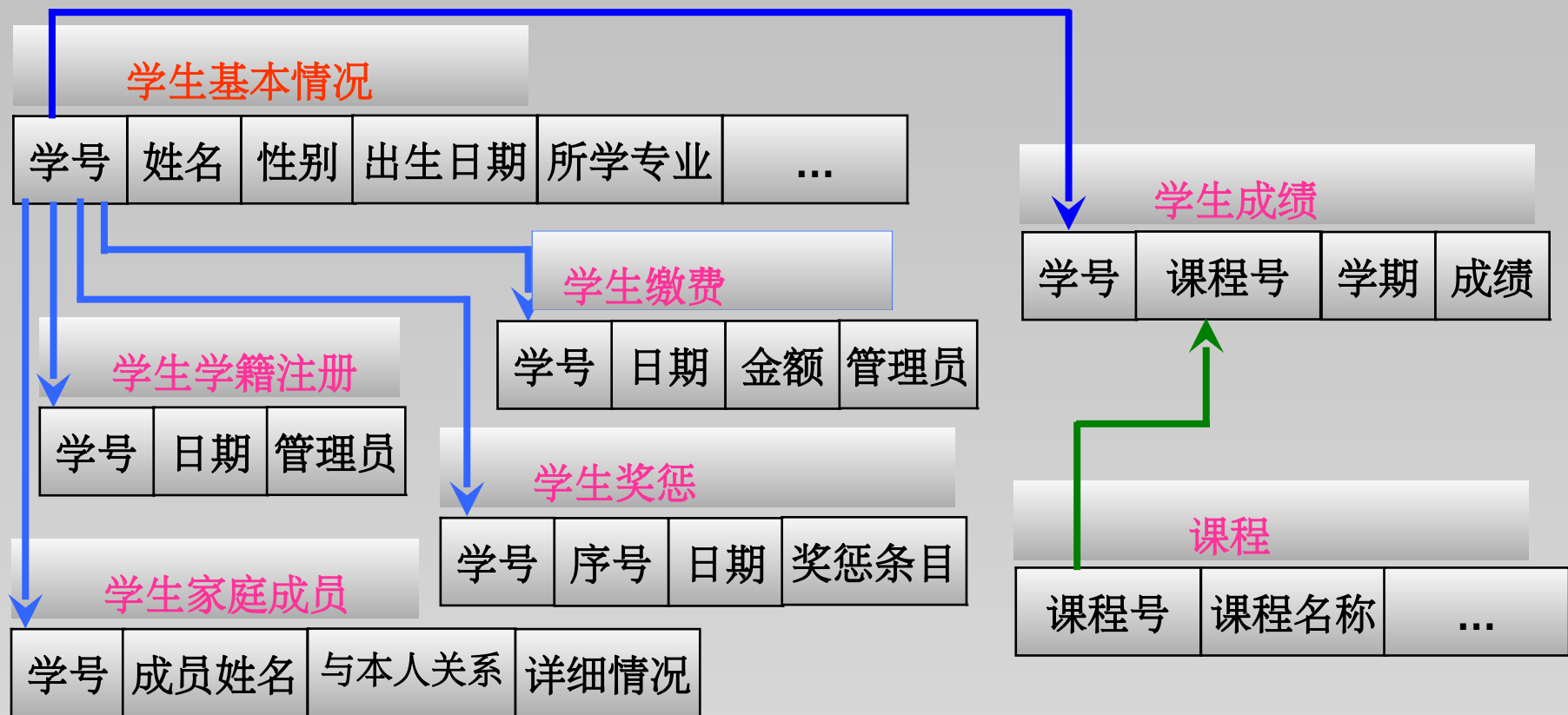


图 1-4 某校信息管理系统中的学生数据

- 二是指在数据库中的数据不是仅仅针对某一个应用，而是面向全组织的所有应用。

1.1.2 数据库技术的产生与发展

■ 数据库管理系统的主要特点

● 数据的共享度高，冗余度低，易扩充

- 数据库管理系统从整体角度描述和组织数据，数据不再是面向某个应用，而是面向整个系统
- 因此，数据可以被多个用户、多个应用共享使用
- 数据共享可以大大减少数据的冗余，避免数据之间的不一致性

● 数据独立性高

- 数据独立是指数据的使用（即应用程序）与数据的说明（即数据的组织结构与存储方式）分离
 - ✓ 这样，应用程序只需要考虑如何使用数据，而无须关心数据库中的数据是如何构造和存储的
 - ✓ 因而，各方（在一定范围内）的变更互不影响

1.1.2 数据库技术的产生与发展

■ 数据库管理系统的主要特点

抽象	例子
应用程序	淘宝的登录、浏览界面等
逻辑结构	数据库表
物理存储结构	如何存储这些数据的组织方式, 如索引

● 数据独立性高

- **数据独立**是指**数据的使用**与**数据的说明**分离
 - ✓ **应用程序**只需考虑**如何使用数据**, 不必关心**数据是如何构造和存储的**
 - ✓ 因而, 各方 (在一定范围内) 的**变更互不影响**
- **数据独立性**用来描述**应用程序**与**数据结构**之间的**依赖程度**, 包括数据的**物理独立性**和数据的**逻辑独立性**, 依赖程度越低则独立性越高
- **物理独立性**是指用户的**应用程序**与数据库中数据的**物理结构**是相互独立的。数据在磁盘上如何组织和存储由 **DBMS** 负责, **应用程序**只关心数据的**逻辑结构**; 当数据的**物理存储结构**改变时, **应用程序**不用修改
- **逻辑独立性**是指用户的**应用程序**与数据库中数据的**逻辑结构**是相互独立的。数据的 (全局) **逻辑结构**由 **DBMS** 负责, **应用程序**只关心数据的**局部逻辑结构 (即应用视图)**, 数据的 (全局) **逻辑结构**改变了, **应用程序**也可以不用修改

1.1.2 数据库技术的产生与发展

■ 数据库管理系统的主要特点

- 数据由数据库管理系统（DBMS）统一管理和控制

➤ 数
的
➤ 数
据
➤ 并
作
整
➤ 数
数

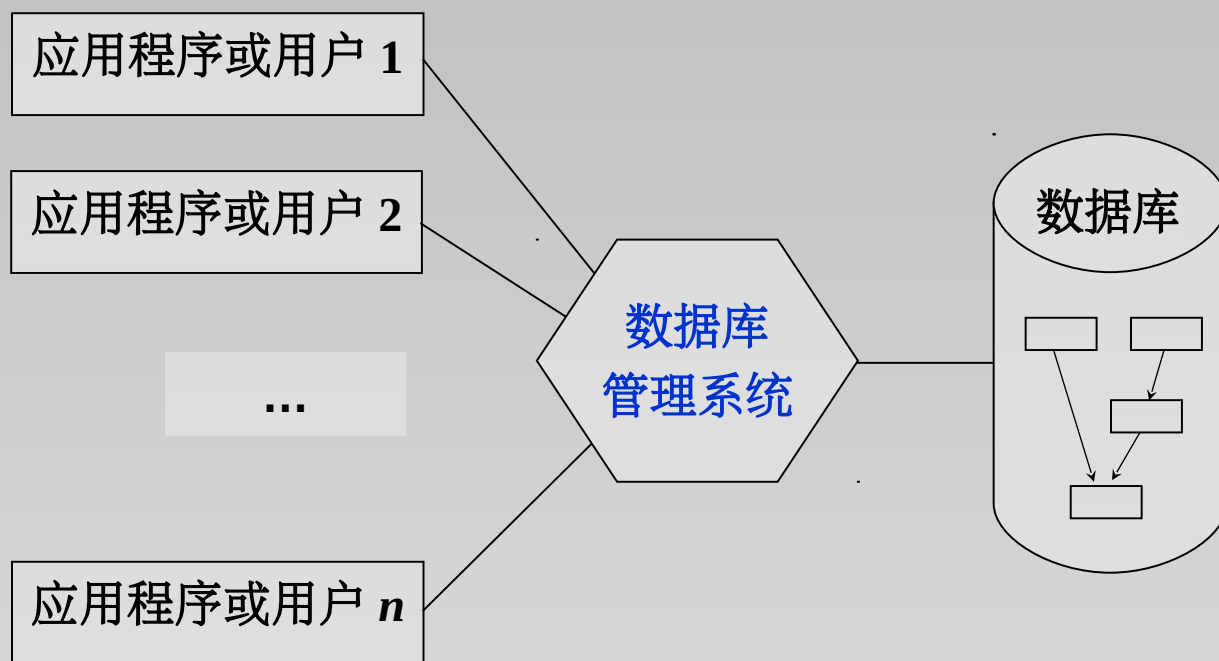


图 1-5 数据库管理系统阶段应用程序与数据之间的对应关系

1.1.2 数据库技术的产生与发展

■ 数据库管理系统

PostgreSQL 数据库管理系统

SQLSever 数据库管理系统

DB2 数据库管理系统

Oracle 数据库管理系统

GaussDB 100/200

openGauss



openGauss 是一款开源关系型数据库管理系统，采用木兰宽松许可证 v2 发行。

- **openGauss** 内核源自 PostgreSQL，深度融合华为在数据库领域多年的经验，结合企业级场景需求，持续构建竞争力特性。同时 **openGauss** 也是一个开源、免费的数据库平台，鼓励社区贡献、合作。

其中，**openGauss** 做为该课程其中一个重要实验平台进行实验。



HUAWEI

openGauss&GaussDB数据库理论与 x 弹性云服务器 - 控制台 +

console.huaweicloud.com/ecm/?agencyId=0ce39e5f68800fc21fd8c0019308c7e8&locale=zh-cn®ion=cn-north-4&charging=1&flavor=c6s.large.2&sysdisk=GPSSD:40&pe...



华为云 控制台

搜索

更多

中文(简体)

hwid_w0asl2ttkro0naf

English

放心购

拖拽上传



弹性云服务器

自定义购买

快速购买

① 基础配置 ② 网络配置 ③ 高级配置 ④ 确认配置

配置

基础配置

计费模式

按需计费

区域

北京四

可用区

可用区2

规格

鲲鹏通用计算增强型 | kc1.large.2 ...

镜像

openEuler 20.03 64bit with ARM

主机安全

基础版

系统盘

通用型SSD, 40 GiB

网络配置

虚拟私有云

vpc-default(192.168.0.0/16)

安全组

default

主网卡

subnet-default(192.168.0.0/24)

弹性公网IP

全动态BGP | 计费方式: 按流量计...

高级配置

云服务器名称

ecs-wzq

登录凭证

密码

云服务器组

--

配置费用 ¥0.3388/小时 + 弹性公网IP流量费用 ¥0.80/Gb

参考价格, 具体扣费请以账单为准。 [了解计费详情](#)

上一步

立即购买

```
121.36.7.169 - FaITF
[omm@ecs-wzq ~]$ gs install -X /opt/software/openGauss/clusterconfig.xml --gsini
t-parameter="--encoding=UTF8" --dn-guc="max_process_memory=4GB" --dn-guc="shar
ed_buffers=256MB" --dn-guc="bulk_write_ring_size=256MB" --dn-guc="cstore_buffers
=16MB"
Parsing the configuration file.
Check preinstall on every node.
Successfully checked preinstall on every node.
Creating the backup directory.
Successfully created the backup directory.
begin deploy..
Installing the cluster.
begin prepare Install Cluster..
Checking the installation environment on all nodes.
begin install Cluster...
Installing applications on all nodes.
Successfully installed APP.
begin init Instance..
encrypt cipher and rand files for database.
Please enter password for database:
Please repeat for database:
begin to create CA cert files
The sslcert will be generated in /opt/ Gaussdb/app/share/sslcert/om
Cluster installation is completed.
Configuring.
Deleting instances from all nodes.
Successfully deleted instances from all nodes.
Checking node configuration on all nodes.
Initializing instances on all nodes.
Updating instance configuration on all nodes.
Check consistence of memCheck and coresCheck on database nodes.
Configuring pg_hba on all nodes.
Configuration is completed.
Successfully started cluster.
Successfully installed application.
end deploy..
[omm@ecs-wzq ~]$
```

```
redistributing : No
-----
[omm@ecs-wzq ~]$ gs sql -d postgres -p 26000 -r
gs sql ((openGauss 1.1.0 build 392c0438) compiled at 2020-12-31 20:08:06 commit 0
last mr )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

postgres=# alter role omm identified by 'Gauss@123' replace 'GaussDB@123';
ERROR: The old password is invalid.
postgres=# alter role omm identified by 'Bigdata@123' replace 'Gauss@123';
ERROR: The old password is invalid.
postgres=# alter role omm identified by 'Bigdata@123' replace 'Admin@123';
ALTER ROLE
postgres=# CREATE USER joe WITH PASSWORD "Bigdata@123";
CREATE ROLE
postgres=# create database db_tpcc OWNER joe;
CREATE DATABASE
postgres=# \q
[omm@ecs-wzq ~]$ gs sql -d db_tpcc -p 26000 -U joe -W Bigdata@123 -r
gs sql: FATAL: database "db_tpcc" does not exist
[omm@ecs-wzq ~]$ gs sql -d postgres -p 26000 -r
gs sql ((openGauss 1.1.0 build 392c0438) compiled at 2020-12-31 20:08:06 commit 0
last mr )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

postgres=# CREATE DATABASE db_tpcc OWNER joe;
CREATE DATABASE
postgres=# \q
[omm@ecs-wzq ~]$ gs sql -d db_tpcc -p 26000 -U joe -W Bigdata@123 -r
gs sql ((openGauss 1.1.0 build 392c0438) compiled at 2020-12-31 20:08:06 commit 0
last mr )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

db_tpcc=>
```

Data Studio : joe.nytable-db_tpccnydhecon

文件(F) 编辑(E) 运行(R) 设置(O) 帮助(H)

db_tpccnydhecon

对象浏览器

输入过滤条件，按回车键开始

nydhecon (121.36.7.169:26000)

- 数据库 (1)
 - db_tpcc
 - 系统模式 (510)
 - 用户模式 (0)
 - db_tpcc
 - 系统模式 (510)
 - 用户模式 (1)
 - joe (1)
 - 函数/过程 (0)
 - 普通表 (1)
 - nytable
 - firstcol
 - 约束
 - 视图 (0)
 - 外表 (0)
 - 序列 (0)
 - 同义词 (0)
- postgres
 - 系统模式 (510)
 - 用户模式 (1)
- 表空间 (2)
 - pg_default
 - pg_global
- 用户/角色 (2)
 - dbuser
 - joe

包含搜索内容

输入搜索内容

firstcol
1 100
2 200

SQL助手

语法 操作列表

所有语法: SELECT

SELECT

功能描述

SELECT用于从表或视图中取出数据。
SELECT语句就像叠加在数据库表上的过滤器，利用SQL关键字从数据表中过滤出用户需要的数据。

注意事项

- 必须对每个在SELECT命令中使用的字段有SELECT权限。

示例

```
--先通过子查询得到一张临时表temp_t，然后查询表temp_t中的所有数据。
postgres=# WITH temp_t ( name, idba )
AS ( SELECT username,userid FROM pg_us
er ) SELECT * FROM temp_t;
--查询tpods.reason表的所有r_reason_sk记录，且去除重复。
postgres=# SELECT DISTINCT ( r_reason_
sk ) FROM tpods.reason;
--LIMIT子句示例：获取表中一条记录。
postgres=# SELECT * FROM tpods.reason
LIMIT 1;
--查询所有记录，且按字母升序排列。
postgres=# SELECT r_reason_desc FROM t
pods.reason ORDER BY r_reason_desc;
--通过表别名，从pg_user和pg_user_status这两张表中获取数据。
postgres=# SELECT a.username,b.locktime
FROM pg_user a,pg_user_status b WHERE
a.userid=b.rolid;
--FULL JOIN子句示例：将pg_user和pg_user_
status这两张表的数据库进行全连接显示，即数据库的
```

Data Studio - finance@mydbcon (1)

文件(F) 编辑(E) 运行(R) 设置(S) 帮助(H)

finance@mydbcon

对象浏览器

请输入过滤条件: 按回车键开始

mydbcon (121.36.7.169:26000)

- 数据库 (4)
 - db_tpc
 - db_tpc
 - finance
 - 系统模式 (510)
 - 用户模式 (6)
 - postgres
- 表空间 (2)
- 用户/角色 (2)

finance@mydbcon (1)

```
1 select count(*) from finance.bank_card;
```

运行时间: 211 ms

消息 结果

包含搜索内容 请输入搜索内容

	count
1	20

查询提交时间: 2021-07-09 17:28:55.976 获取1行, 共157 ms 无法编辑查询结果。

查询结果

自动保存完成: Fri Jul 09 17:26:21 CST 2021

SQL助手

语法 操作列表

所有语法: SELECT

SELECT

功能描述

SELECT用于从表或视图中取出数据。
SELECT语句就像叠加在数据库表上的过滤器, 利用SQL关键字从数据表中过滤出用户需要的数据。

注意事项

- 必须对每个在SELECT命令中使用的字段有SELECT权限。

示例

```
--先通过子查询得到一张临时表temp_t, 然后查询temp_t中的所有数据。
postgres=# WITH temp_t (
name, isdba ) AS ( SELECT
username, usesuper FROM pg
```

Data Studio - finance@mydbcon (1)

文件(F) 编辑(E) 运行(R) 设置(S) 帮助(H)

finance@mydbcon

对象浏览器

请输入过滤条件: 按回车键开始

mydbcon (121.36.7.169:26000)

- 数据库 (4)
 - db_tpc
 - db_tpc
 - finance
 - 系统模式 (510)
 - 用户模式 (6)
 - postgres
- 表空间 (2)
- 用户/角色 (2)

finance@mydbcon (1)

```
1 DROP INDEX finance.idx_property;
2 CREATE INDEX finance.idx_property ON finance.property(pro_c_id DESC, pro_income, pro_purchase_time);
3
```

运行时间: 286 ms

消息

包含搜索内容 请输入搜索内容

查询提交时间: 2021-07-09 17:30:33.189 CST [INFO] 操作影响的记录行数: 1
[INFO] 执行时间: 50 ms
2021-07-09 17:30:33.189 CST [INFO] 操作影响的记录行数: 1
[INFO] 执行时间: 150 ms
2021-07-09 17:30:33.189 CST [INFO] 操作影响的记录行数: 0
[INFO] 执行时间: 50 ms
2021-07-09 17:30:33.189 CST [INFO] 执行成功...

查询结果

Data Studio - finance@mydbcon (1)

文件(F) 编辑(E) 运行(R) 设置(S) 帮助(H)

finance@mydbcon

对象浏览器

请输入过滤条件: 按回车键开始

mydbcon (121.36.7.169:26000)

- 数据库 (4)
 - db_tpc
 - db_tpc
 - finance
 - 系统模式 (510)
 - 用户模式 (6)
 - postgres
- 表空间 (2)
- 用户/角色 (2)

finance@mydbcon (1)

```
1 DROP INDEX finance.idx_property;
2 CREATE INDEX finance.idx_property ON finance.property(pro_c_id DESC, pro_income, pro_purchase_time);
3 ALTER INDEX finance.idx_property RENAME TO idx_property_temp;
```

运行时间: 286 ms

消息

包含搜索内容 请输入搜索内容

查询提交时间: 2021-07-09 17:30:33.189 CST [INFO] 操作影响的记录行数: 1
[INFO] 执行时间: 50 ms
2021-07-09 17:30:33.189 CST [INFO] 操作影响的记录行数: 1
[INFO] 执行时间: 150 ms
2021-07-09 17:30:33.189 CST [INFO] 操作影响的记录行数: 0
[INFO] 执行时间: 50 ms
2021-07-09 17:30:33.189 CST [INFO] 执行成功...

查询结果

1.1.3 数据库应用

■ 图书馆管理

- 用于存储图书馆的馆藏资料 (图书、期刊等)、读者 (教师、学生等) 信息, 以及图书和期刊的借阅、归还记录等, 方便读者查找资料, 方便管理人员办理图书和期刊的借阅、归还和催还等手续, 提高图书馆管理水平

■ 书店管理

- 用于存储员工、客户信息以及图书采购、库存、销售记录等, 提高图书的采购、库存和销售管理水平, 方便书店的账务处理

■ 教学管理

- 用于存储各专业教学计划、教师和学生信息、教室信息、教材信息、教师开课以及学生选课记录等, 提高排课、选课、成绩管理、毕业管理效率

■ 科研管理

- 用于存储教师信息、科研成果记录等, 方便科研成果的考核、检索和统计工作

1.1.3 数据库应用

■ 银行管理

- 用于存储客户信息、存款账户和贷款账户记录以及银行之间的转账交易记录等，提高存款、贷款管理水平，加速资金流转和银行结算

■ 售票管理

- 用于存储客户信息和客运飞机、火车、汽车班次等信息，以及订票、改签和退票记录等，提高交通客运管理水平，方便客户订票

■ 电信管理

- 用于存储客户信息、通话记录等，自动结算话费，维护预付电话卡的余额，产生每月账单，提高电信管理水平

■ 证券管理

- 用于存储客户信息以及股票、债券等金融票据的持有、出售和买入信息，也可以存储实时的市场交易数据，以便客户能够进行联机交易，公司能够进行自动交易和结算

1.1.3 数据库应用

■ 销售管理

- 用于存储客户、商品信息以及销售记录，以便能够实时的订单跟踪、销售结算、库存管理和商品推荐

■ 制造业管理

- 用于存储客户信息、生产工艺信息，以及采购、生产、入库、出库记录等，实现供应链管理，跟踪工厂的产品生产情况，实现零部件、半成品、产成品的库存管理等

■ 固定资产管理

- 用于存储客户信息、部门信息和员工信息，固定资产的采购记录、领用记录和报废记录等，自动计提固定资产折旧，提供各种固定资产报表

■ 人力资源管理

- 用于存储部门信息、员工信息，以及出勤记录、计件记录等，自动计算员工的工资、所得税和津贴，产生工资单

数据库系统作用小结

- 1: 什么是数据？其分类？
- 2: 数据管理技术发展的阶段和各阶段特点？
- 3: 一些应用，为以后大作业准备？

目 录



数据库系统的作用



数据模型



数据抽象与数据库三级模式



数据库系统

1.2.1 数据

■ 数据库结构的基础是**数据模型**

■ **数据模型**是一个描述**数据结构**的**数学形式体系**（即**概念及其**

- **数据结构**用于刻画数据、数据之间的关系
- **数据约束**是对**数据结构**和**数据操作**为数据完整性约束

■ 通过**数据模型**可以对现实世界的**数据特征**进行**抽象**

■ 根据**数据抽象**的不同级别，将**数据模型**划分为 3 类：

- **概念模型**：概念层次的数据模型，也称为**信息模型**
- **逻辑模型**：用于描述数据库数据的**整体逻辑结构**
- **物理模型**：用来描述数据的**物理存储结构和存取方法**

数据（Data）是描述事物的符号记录，模型（Model）是现实世界的抽象。数据模型从抽象层次上描述了系统的静态特征、动态行为和约束条件，为数据库系统的信息表示与操作提供了一个抽象的框架。数据模型所描述的内容有三部分：数据结构、数据操作和数据约束。

1.2.1 数据模型的分类

■ 概念模型

- 按用户的观点或认识对现实世界的数据和信息进行建模
- 主要用于数据库设计——强调语义表达功能
- 常用的概念模型有实体 - 联系模型 (E-R 模型) 和面向对象模型 (OO 模型)
- E-R 模型基于对现实世界的如下认识：现实世界是由一组称作实体的基本对象以及这些对象间的联系构成
 - 实体是现实世界中可区别于其他对象的一件“事情”或一个“物体”
 - 如，一个学生、一个部门、一个教室、一种商品、一本书、一门课程，以及一次选课、采购、销售、存款业务 (记录) 等都是实体
- OO 模型是用面向对象观点来描述现实世界实体 (对象) 的逻辑组织、对象间限制和联系等的模型
 - 对象是由一组数据结构和在这组数据结构上操作的程序代码封装起来的基本单位

1.2.1 数据模型的分类

■ 逻辑模型

- 是用户通过数据库管理系统看到的现实世界，是按计算机系统的观点对数据建模，即数据的计算机实现形式
- 主要用于 **DBMS 的实现**。它既要考虑用户容易理解，又要考虑便于 **DBMS 实现**
- 不同的 **DBMS** 提供不同的**逻辑数据模型**
 - 层次模型 (hierarchical model)
 - 网状模型 (network model)
 - 关系模型 (relational model)
 - 面向对象模型 (即 OO 模型)
 - XML 模型
 - 对象关系模型 (object relational model)

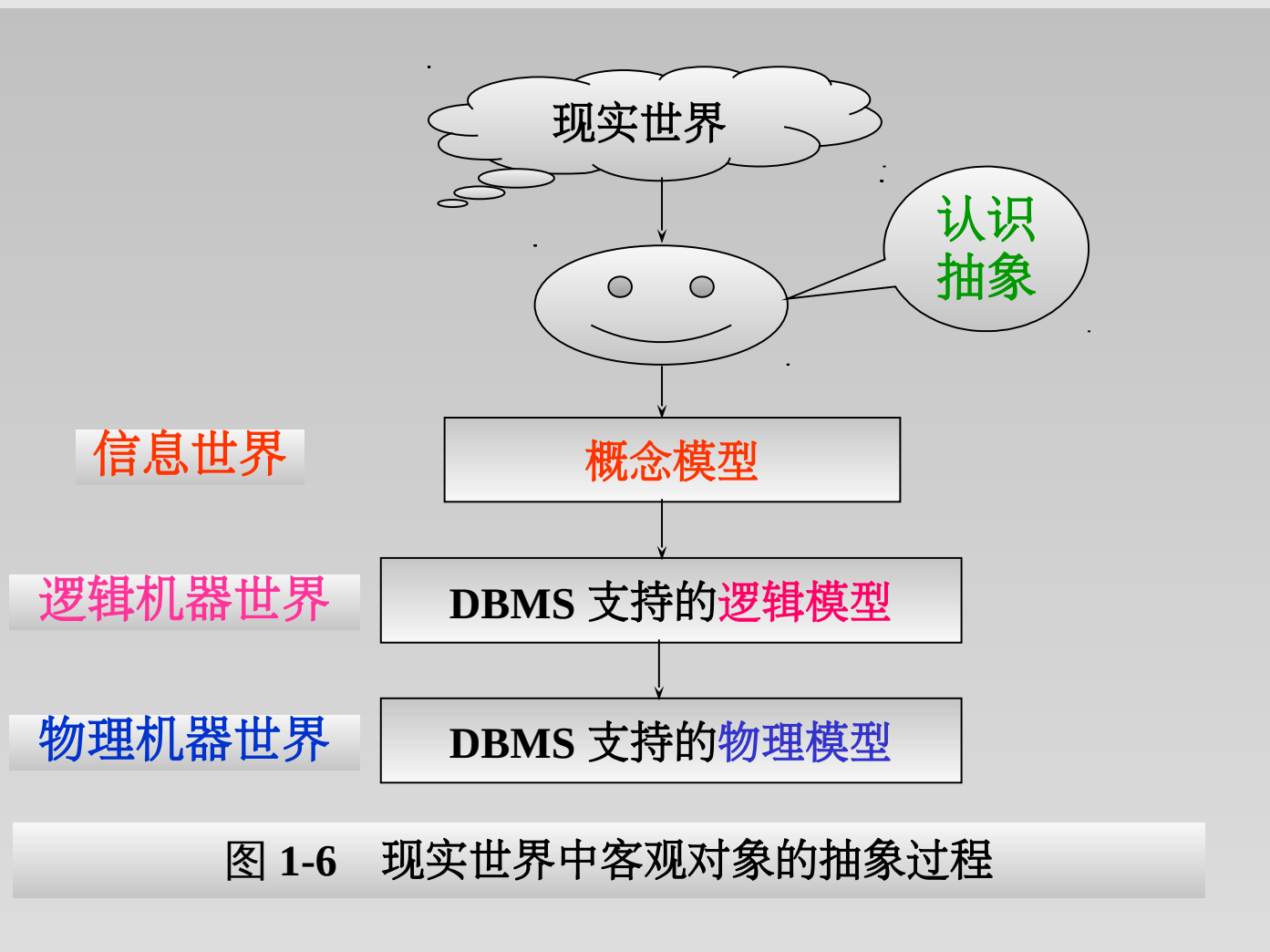
1.2.1 数据模型的分类

■ 物理模型

- 物理层
- 例如，
还是相
的；数
- 物理模
解和选

■ 模型间转

- 从现实
- 从概念
也可以
- 从逻辑



1.2.2 数据模型的组成要素

- 数据模型是一个描述数据结构、数据操作以及数据完整性约束的数学形式体系 (即概念及其符号表示系统)
- 这些概念及其符号表示系统精确地描述了系统的静态特性、动态特性和完整性规则
- 数据模型的组成要素有:
 - 数据结构: 描述数据库的组成对象以及对象之间的联系
 - 数据操作: 指对数据库中各种对象 (型) 的实例 (值) 允许执行的操作集合, 包括操作及有关的操作规则
 - 数据完整性约束: 一组数据完整性规则, 是数据、数据语义和数据联系所具有的制约和依存规则, 包括数据结构完整性规则和数据操作完整性规则, 用以限定符合数据模型的数据库状态以及状态的变化, 以保证数据库中数据的正确、有效和相容

■ 典型代表是 1968 年
管理系统——IMS(

■ 层次模型用树形结构
体用记录来表示，实

■ 满足如下两个条件的

- 有且只有一个结点没有双亲结点，这个结点称为根结点
- 根以外的其他结点有且只有一个双亲结点

■ 在层次模型中，每个结点表示一个记录型，记录(型)之间的联系用结点之间的连线(有向边)表示，这种联系是父子之间的一对多的联系

■ 每个记录型由若干个字段组成，记录型描述的是实体，字段描述的是实体的属性。每个记录型可以定义一个排序字段，也称为码字段，如果所定义的排序字段的值唯一，则它也可以用来唯一标识一个记录值

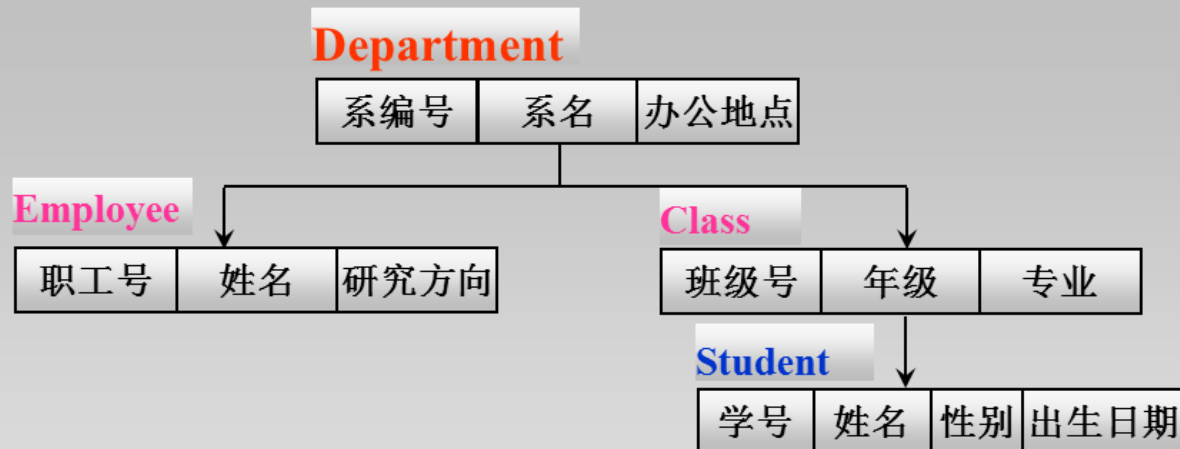


图1-7 一个简单的教学管理系统的层次数据模型

1.2.3 层次模

Department

D05	计算机系	信息大楼
-----	------	------

D08	信息系	信息大楼
-----	-----	------

Employee

E0501	万家乐	数据库
E0803	王国方	数据结构
E0805	张长工	编译原理
E0809	李小红	运筹学

Class

C01	2008	计算机
C03	2008	电子商务
C04	2009	信息系统

Student

S080125	李小勇	男	11/26/90
S080325	万元户	男	11/26/90
S080348	李 兴	女	05/21/91
S090204	王红敏	女	06/12/92
S090404	刘月高	女	06/12/92
S090412	王永民	男	09/25/91
S090428	英达成	女	02/18/92

Department

系编号	系名	办公地点
-----	----	------

Employee

职工号	姓名	研究方向
-----	----	------

Class

班级号	年级	专业
-----	----	----

Student

学号	姓名	性别	出生日期
----	----	----	------

图1-7 一个简单的教学管理系统的层次数据模型

1.2.3 层次模型

■ 层次模型的主要优点：

- 数据结构比较简单清晰
- 查询效率高
- 提供了良好的数据完整性支持

■ 层次模型的主要缺点：

- 现实世界中很多联系是非层次的（如多对多联系），层次模型在表示这类联系时，解决的办法：
- 一是通过引入冗余数据（易产生不一致性），二是创建非自然的数据结构（引入虚拟结点）。对插入和删除操作的限制比较多，因此应用程序的编写比较复杂
- 查询孩子结点必须通过双亲结点
- 由于结构严密，层次命令趋于程序化

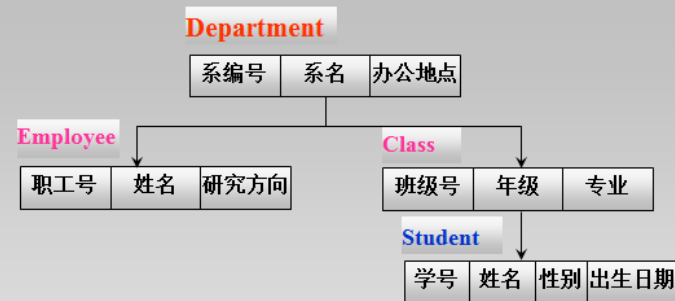


图1-7 一个简单的教学管理系统的层次数据模型

1.2.4 网状模型

- 典型代表是 DBTG 系统，亦称为 CODASYL 系统，它是 20 世纪 70 年代由数据系统语言研究会 (conference on data system language, CODASYL) 下属的数据库任务组 (data base task group, DBTG) 提出的一个系统方案
- 满足如下两个条件的基本层次联系的集合称为网状模型
 - 允许一个以上的结点无双亲
 - 一个结点可以有多个双亲
- 网状模型是一种比层次模型更具普遍性的结构，它去掉了层次模型的两个限制，还允许两个结点之间有多种联系 (称为复合联系)。因此，网状模型可以更直接地去描述现实世界

1.2.4 网状模型

■ 网状模型的主要优点:

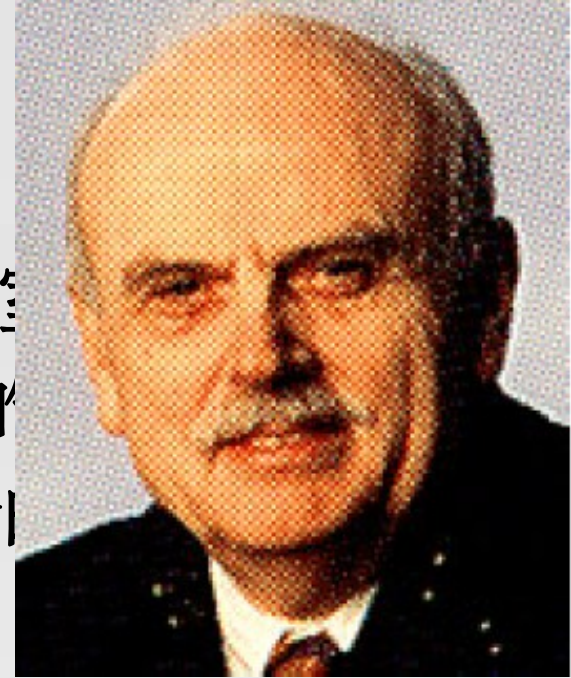
- 能够更为直接地描述现实世界
- 具有良好的性能，存取效率较高

■ 网状模型的主要缺点:

- 结构比较复杂，而且随着应用规模的扩大，数据库的结构会变得越来越复杂，不利于最终用户掌握
- 操作语言比较复杂

1.2.5 关系模型

- 1970 年美国 IBM 公司 San Jose 研究员 E. F. Codd 首次提出了数据库管理系统的概念，开创了数据库关系方法和关系数据理论，为数据库技术奠定了理论基础。
- 由于 E. F. Codd 的杰出工作，他于 1981 年获得 ACM 图灵奖
- 20 世纪 80 年代以来，计算机厂商新推出的数据库管理系统几乎都支持关系模型，数据库领域当前的研究工作也都是以关系方法为基础
- 本书的重点也主要是讲授关系数据库



1.2.5 关系数据库

■ 关系数据模型的数据结构

● 关系模型中的常用术语

➤ **关系 (relation)**：一个关系有一个名称，即关系名；

➤ **元组 (tuple)**：表中的一行称为一个元组；

➤ **属性 (attribute)**：表中的一列称为一个属性，每一个属性有一个名称，即属性名；

➤ **码 (key)**：也称为**码键**。表中的某个属性或属性组，它可以唯一地确定关系中的一个元组，如**关系 Student**中的**学号**，它可以唯一地标识一个学生；

➤ **域 (domain)**：属性的取值范围；

➤ **分量 (component)**：元组中的一个属性值；

Student关系

学号	姓名	性别	出生日期	所学专业
0701001	李小勇	男	1990-12-21	计算机
0701008	王红	男	1992-04-26	计算机
0802002	刘方晨	女	1990-11-11	信息系统
0802005	王红敏	女	1990-10-01	信息系统
0703045	王红	男	1992-04-26	会计学
0703010	李宏冰	女	1992-03-09	会计学
...

Course关系

课程号	课程名称	学时	学分
CS005	数据库系统概论	64	4
CS012	操作系统	80	5
CP001	基础会计	48	3
...

Score关系

学号	课程号	学期	成绩
0701001	CS005	072	92
0701001	CS012	081	88
0701008	CS005	072	86
0701008	CS012	081	93
0701008	CP001	081	78
0802002	CS005	082	85
0802002	CP001	091	95
0802005	CS005	082	72
0802005	CP001	091	88
0703045	CP001	072	84
0703010	CP001	072	92
...

图1-9 关系模型的数据结构

1.2.5 关系数据库的数据结构

关系数据模型的数据结构

关系模型中的常用术语：

➤ **外码 (foreign key)**：表中的元组 (实体) 与另一表中的元组 (实体) 与另一

✓ **外码**的取值范围对应于另一表中的元组 (实体) 与另一

✓ 如**关系 Score** 中的**学号**，它描述了**关系 Score** 与**关系 Student** 的**联系** (即哪个学生选修了课程)，因此**学号**是**关系 Score** 的**外码**

✓ 同理，**课程号**也是**关系 Score** 的**外码**，它描述了**关系 Score** 与**关系 Course** 的**联系** (即哪门课程被学生选修了)

➤ **关系模式 (relational schema)**：通过关系名和属性名列表对关系进行描述，即二维表的**表头部分** (表格的描述部分)

关系模式的一般形式：

➤ **关系名 (属性名 1 ， 属性名 2 ， … ， 属性名 n)**

Student关系

学号	姓名	性别	出生日期	所学专业
0701001	李小勇	男	1990-12-21	计算机
0701008	王 红	男	1992-04-26	计算机
0802002	刘方晨	女	1990-11-11	信息系统
0802005	王红敏	女	1990-10-01	信息系统
0703045	王 红	男	1992-04-26	会计学
0703010	李宏冰	女	1992-03-09	会计学
...

Course关系

课程号	课程名称	学时	学分
CS005	数据库系统概论	64	4
CS012	操作系统	80	5
CP001	基础会计	48	3
...

Score关系

学号	课程号	学期	成绩
0701001	CS005	072	92
0701001	CS012	081	88
0701008	CS005	072	86
0701008	CS012	081	93
0701008	CP001	081	78
0802002	CS005	082	85
0802002	CP001	091	95
0802005	CS005	082	72
0802005	CP001	091	88
0703045	CP001	072	84
0703010	CP001	072	92
...

图1-9 关系模型的数据结构

Student 关系

学号	姓名	性别	出生日期	所学专业
0701001	李小勇	男	1990-12-21	计算机
0701008	王 红	男	1992-04-26	计算机
0802002	刘方晨	女	1990-11-11	信息系统
0802005	王红敏	女	1990-10-01	信息系统
0703045	王 红	男	1992-04-26	会计学
0703010	李宏冰	女	1992-03-09	会计学
...

Course 关系

课程号	课程名称	学时	学分
CS005	数据库系统概论	64	4
CS012	操作系统	80	5
CP001	基础会计	48	3
...

Score 关系

学号	课程号	学期	成绩
0701001	CS005	072	92
0701001	CS012	081	88
0701008	CS005	072	86
0701008	CS012	081	93
0701008	CP001	081	78
0802002	CS005	082	85
0802002	CP001	091	95
0802005	CS005	082	72
0802005	CP001	091	88
0703045	CP001	072	84
0703010	CP001	072	92
...

图 1-9 关系模型的数据结构

- 关系的每一个元组必须是可分区的，即属性的属性
- 关系的每一个属性 (即元组的分量) 必须是一个不可分的数据项，即不允许表中有表 (stuid,sex,birthday(y,m,d)..)

1.2.5 关系数据库

Student关系

学号	姓名	性别	出生日期	所学专业
0701001	李小勇	男	1990-12-21	计算机
0701008	王红	男	1992-04-26	计算机
0802002	刘方晨	女	1990-11-11	信息系统
0802005	王红敏	女	1990-10-01	信息系统
0703045	王红	男	1992-04-26	会计学
0703010	李宏冰	女	1992-03-09	会计学
...

Course关系

课程号	课程名称	学时	学分
CS005	数据库系统概论	64	4
CS012	操作系统	80	5
CP001	基础会计	48	3
...

Score关系

学号	课程号	学期	成绩
0701001	CS005	072	92
0701001	CS012	081	88
0701008	CS005	072	86
0701008	CS012	081	93
0701008	CP001	081	78
0802002	CS005	082	85
0802002	CP001	091	95
0802005	CS005	082	72
0802005	CP001	091	88
0703045	CP001	072	84
0703010	CP001	072	92
...

图1-9 关系模型的数据结构

■ 关系数据模型的操作

- 关系数据模型的操作主要(增、删、改)

- 关系：元组的集合，关系模型

对象和操作结果都是关系(元组的集合)——不同于传统的非关系模型的数据操作：单记录操作

- 关系模型：存取路径是透明的，用户只要指出“干什么”或“找什么”，不必说明“怎么干”或“怎么找”，从而大大地提高了数据的独立性，提高了软件的开发和维护效率

■ 关系数据模型的完整性约束

- 实体完整性、参照完整性和用户自定义完整性

1.2.5 关系模型 层次模型关系模型查询举例

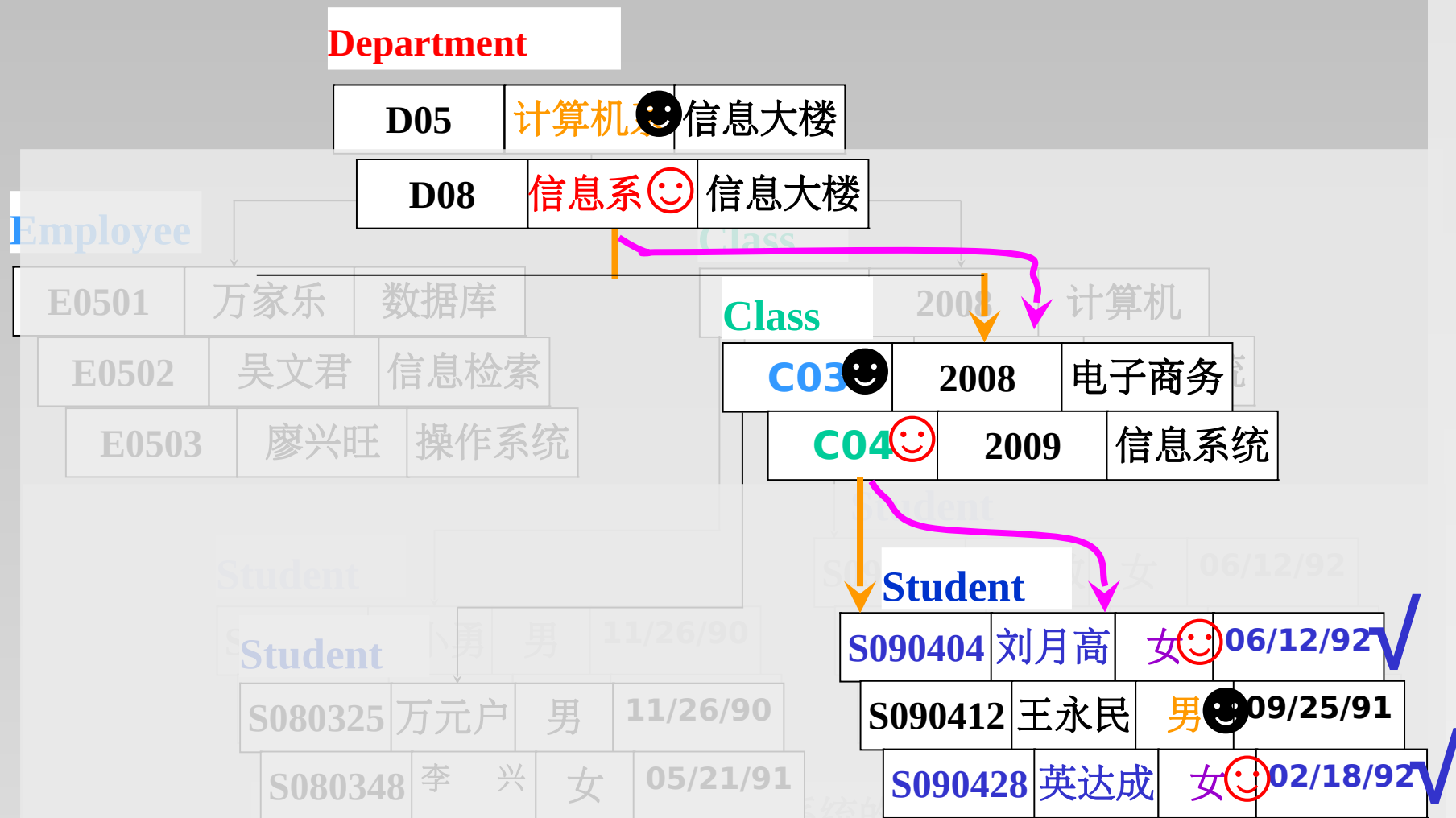
■ 关系数据模型的优点:

- 严格的数学基础: 有关系代数作为语言模型, 有关系数据理论作为理论基础
- 概念单一: 无论实体还是实体之间的联系都是用关系来表示, 对数据 (关系) 的操作 (查询和更新) 结果还是关系。所以其数据结构简单、清晰, 用户易懂易用
- 存取路径透明: 具有更高的数据独立性、更好的安全保密性, 简化了程序员的工作, 提高了软件的开发和维护效率

■ 关系数据模型的缺点:

- 由于存取路径对用户透明, 查询效率往往不如非关系数据模型
- 为了提高性能, DBMS 必须对用户的查询请求进行查询优化, 这样就增加了 DBMS 的开发难度

层次模型：查询信息系 C04 班的女同学



关系模型：查询信息系 C04 班的女同学

学生关系 Student

学号	姓名	性别	出生日期	所在系	班级
S080125	李小勇	男☹️	11/26/90	D05☹️	C01☹️
S080148	黄小红	女😊	05/21/91	D05☹️	C01☹️
S090204	王红敏	女😊	06/12/92	D05☹️	C02☹️
S090212	刘宏昊	男☹️	09/25/91	D05☹️	C02☹️
S090228	李 立	女😊	02/18/92	D05☹️	C02☹️
S080325	万元户	男☹️	11/26/90	D08😊	C03☹️
S080348	李 兴	女😊	05/21/91	D08😊	C03☹️
S090404	刘月高	女😊	06/12/92	D08😊	C04😊
S090412	王永民	男☹️	09/25/91	D08😊	C04😊
S090428	英达成	女😊	02/18/92	D08😊	C04😊

D05——计算机系

D08——信息系



1.2.6 面向对象模型

■ 以面向对象数据模型为核心的面向对象数据库的主要特征：

- **对象**。面向对象数据模型将客观世界模拟成由各个相互作用的称为对象的单元组成的复杂系统。对象的定义包括**状态**和**行为**两方面，**状态**由一组属性值组成，**行为**由一组方法组成
- **对象类**。具有相同属性和方法的对象组成对象类，对象只属于某一个类并作为该类的实例
- **继承**。定义子类时可以直接继承超类的属性和方法，在此基础上定义不同于超类的属性和方法，这称为子类对超类的继承 (inheritance)
- **持久性和对象标识**。在大多数程序设计语言中，对象是临时的；但在面向对象数据库中，对象在被删除前是永久存在的。一个对象的对象标识在系统中是唯一的，在整个生存期内是不变的
- **阻抗失配**。数据库查询语言是由系统自选查询路径的**非过程化语言**。非过程化语言**面向集合的操作**方式与高级程序设计语言**面向记录的操作**方式之间会产生不协调现象，称为**阻抗失配**。阻抗失配的根本原因在于数据库的数据模型与程序设计语言的不一致，因而对所有嵌入式数据库查询语言来说，阻抗失配是不可避免的

1.2.7 XML 模型

- XML（可扩展标记语言）是 SGML（标准通用标记语言）的子集，目标是允许普通的 SGML 在 Web 上以目前 HTML（超文本标记语言）的方式被服务、接收和处理。XML 被设计成易于实现，且可在 SGML 和 HTML 之间互操作

数据模型小结

数据模型定义？ 分层？

逻辑模型中多个模型？ 外加两个模型？

数据模型的三要素？ 特别知道关系数据模型情况！

目 录



数据库系统的作用



数据模型



数据抽象与数据库三级模式



数据库系统

1.3.1 数据抽象

- **DBMS**：隐藏关于数据存储和维护的某些细节，为用户提供数据在不同层次上的视图，即数据抽象，方便不同的使用者可以从不同的角度去观察和利用数据库中的数据
- **物理层抽象**
 - 最低层次的抽象，描述数据实际上是怎样存储的
- **逻辑层抽象**
 - 描述数据库中存储什么数据以及这些数据之间存在什么关系
 - 提供给数据库管理员和数据库应用开发人员使用的，他们必须明确知道数据库中应该保存哪些信息
- **视图层抽象**
 - 最高层次的抽象，只描述整个数据库的某个部分
 - 系统可以为同一数据库提供多个视图，每一个视图对应一个具体的应用，亦称为应用视图

1.3.2 数据库

- 根据数据抽象的 3 个不同观察数据库的 3 个不同角用数据库的需要。这就是

Student关系				
学号	姓名	性别	出生日期	所学专业
0701001	李小勇	男	1990-12-21	计算机
0701008	王 红	男	1992-04-26	计算机
0802002	刘方晨	女	1990-11-11	信息系统
0802005	王红敏	女	1990-10-01	信息系统
0703045	王 红	男	1992-04-26	会计学
0703010	李宏冰	女	1992-03-09	会计学
...

Course关系			
课程号	课程名称	学时	学分
CS005	数据库系统概论	64	4
CS012	操作系统	80	5
CP001	基础会计	48	3
...

Score关系			
学号	课程号	学期	成绩
0701001	CS005	072	92
0701001	CS012	081	88
0701008	CS005	072	86
0701008	CS012	081	93
0701008	CP001	081	78
0802002	CS005	082	85
0802002	CP001	091	95
0802005	CS005	082	72
0802005	CP001	091	88
0703045	CP001	072	84
0703010	CP001	072	92
...

图1-9 关系模型的数据结构

■ 模式

- 也称为逻辑模式，对应于逻辑层数据抽象，是数据库中全体数据的逻辑结构和特征的描述，是所有用户的公共数据视图。
- 模式的一个具体值称为模式的一个实例 (instance)
- 它是 DBMS 模式结构的中间层，既不涉及数据的物理存储细节和硬件环境，也与具体的应用程序、所使用的应用开发工具及高级程序设计语言无关

1.3.2 数据库的三级模式

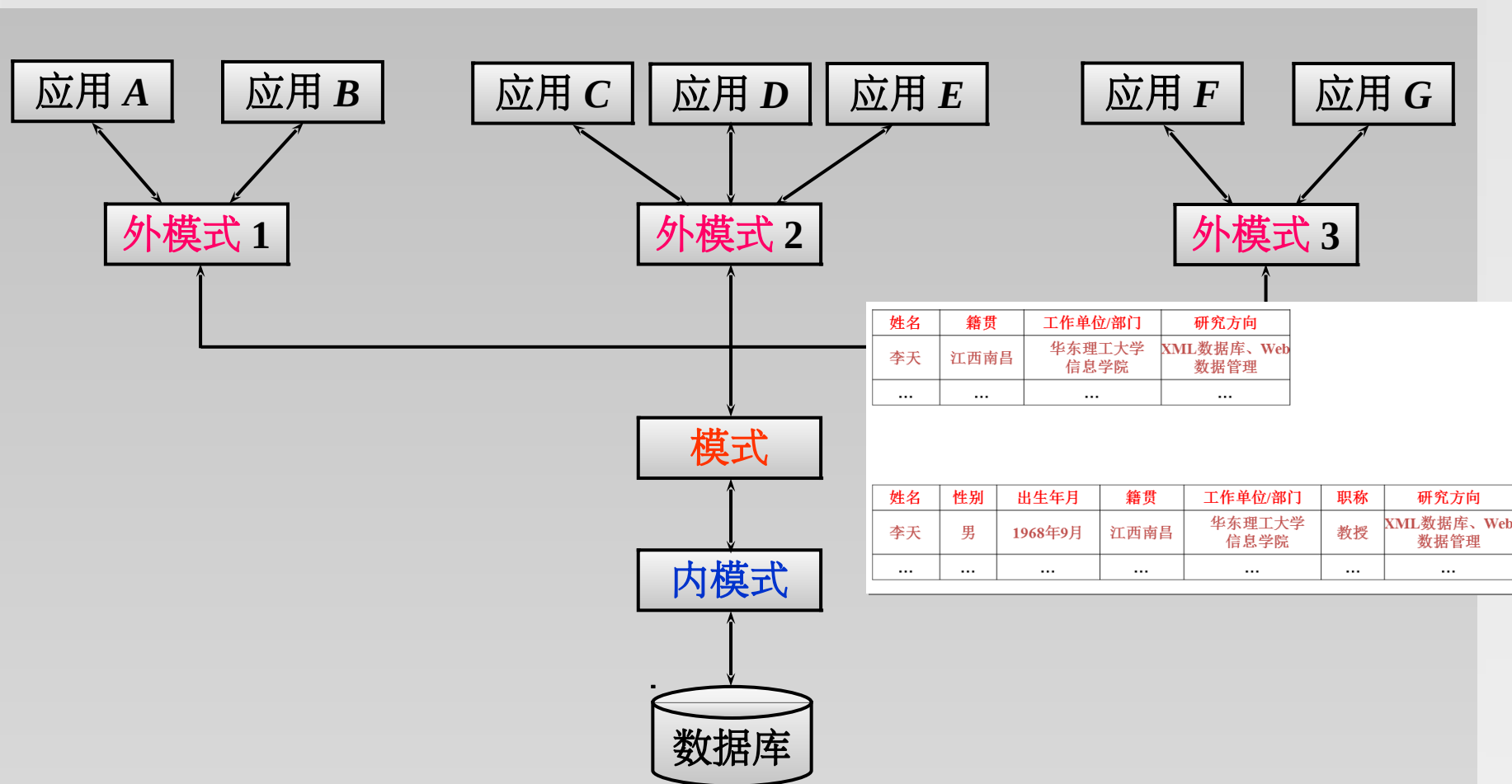


图 1-13 数据库的三级模式结构

1.3.2 数据库的三级模式

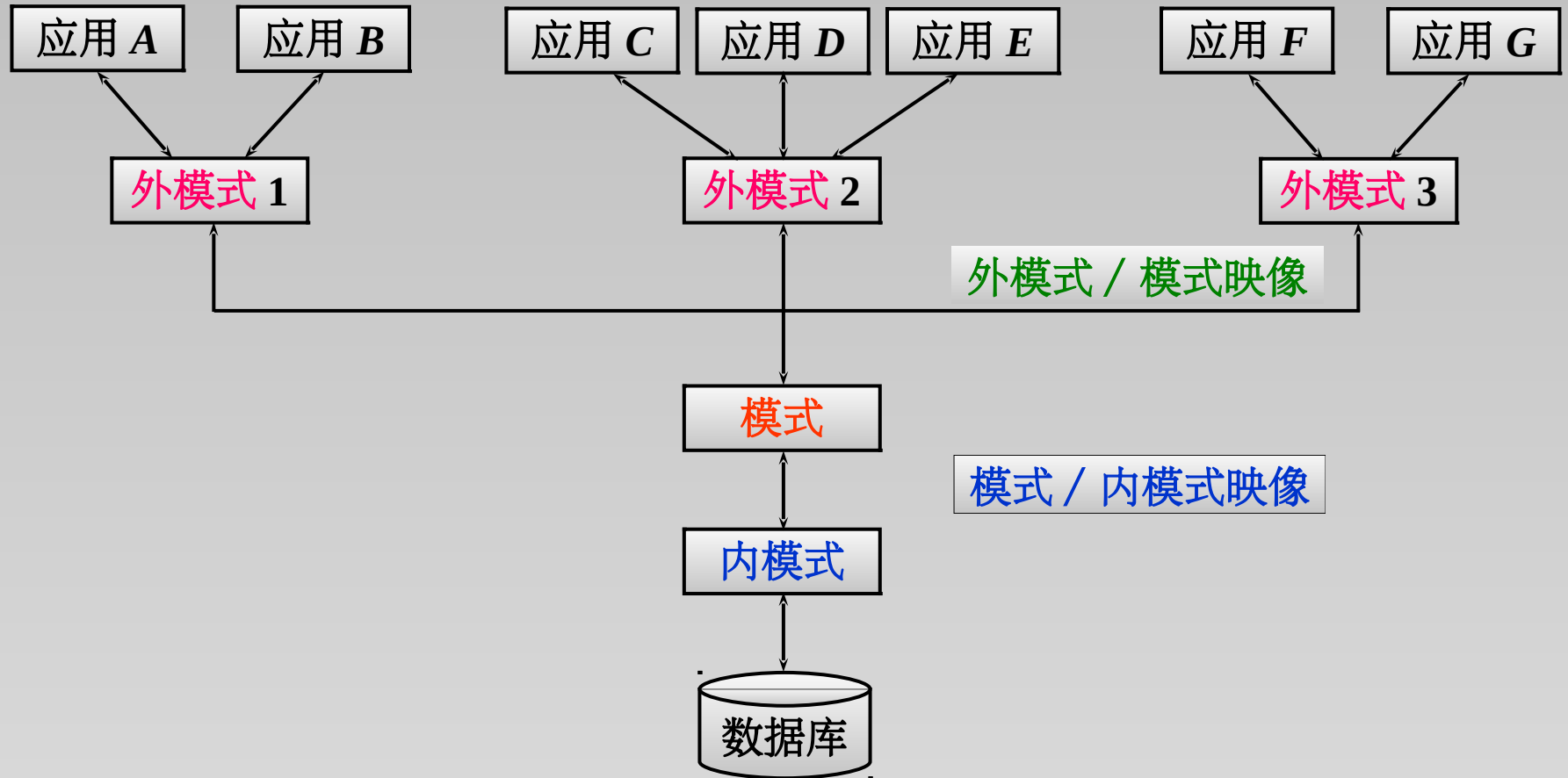


图 1-13 数据库的三级模式结构

1.3.2 数据库的三级模式

- 在数据库的三级模式结构中，**模式即全局逻辑结构是数据库的核心和关键**，它独立于数据库的其他层次。因此，设计数据库模式结构时，应**首先确定数据库的逻辑模式**
- 总结
- 模式，外模式，内模式，外模式 / 模式映像，模式 / 内模式映像是什么？另外
 - 一方面由于**数据与应用程序之间的独立性**，使得**数据的定义和描述可以从应用程序中分离出来**
 - 另一方面由于**数据的存取由 DBMS 管理**，用户**不必考虑存取路径等细节**
 - 从而**大大简化了应用程序的编制**，也大大提高了应用程序的**维护和修改的效率**

目 录



数据库系统的作用



数据模型



数据抽象与数据库三级模式



数据库系统

1.4.1 数据库系统组成

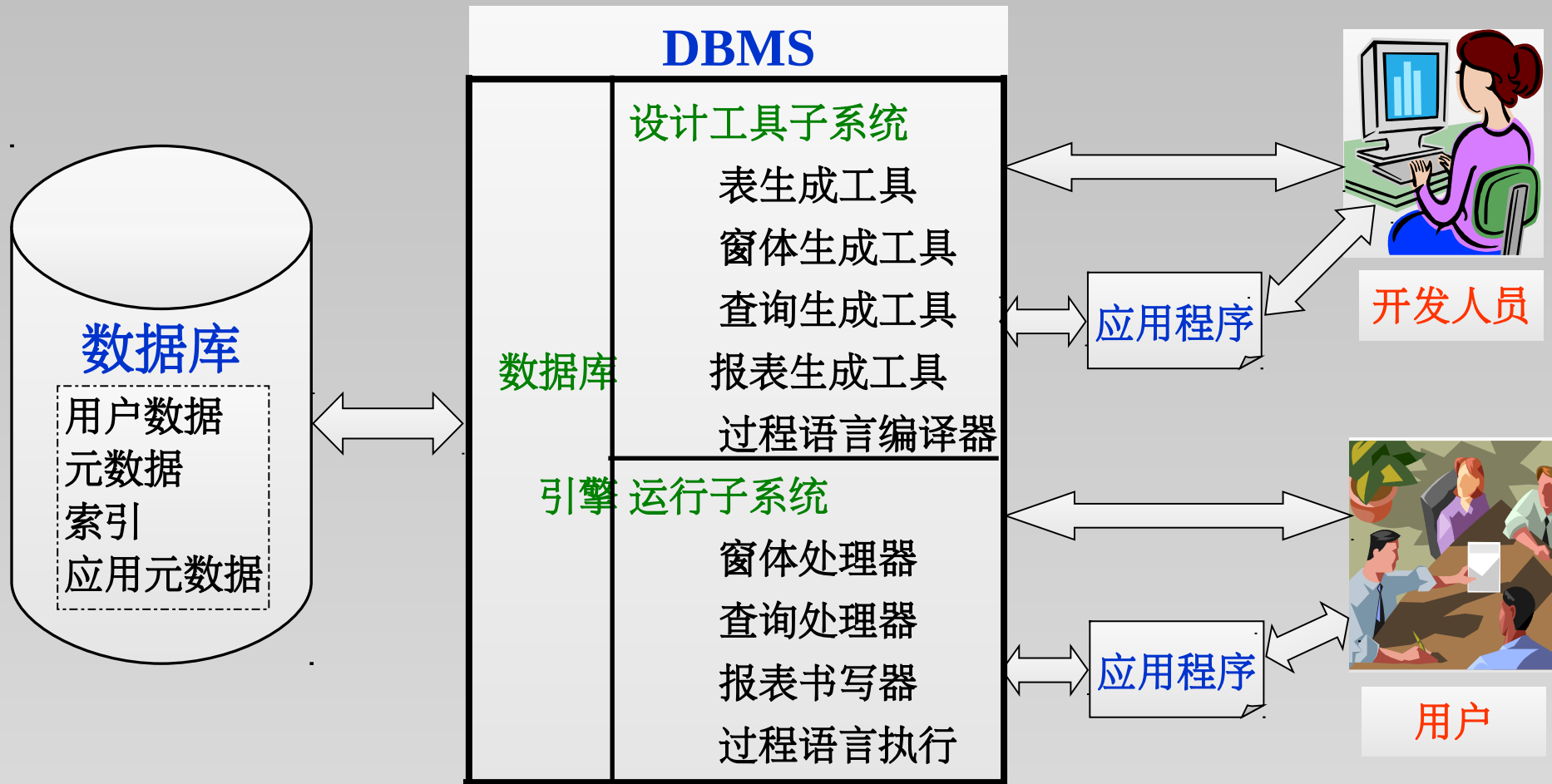


图 1-14 数据库系统的组成

1.4.2 数据库管理系统——DBMS

- 数据库管理系统 (DBMS) 是一组软件，负责数据库的存取、维护和管理
- DBMS 的功能
 - 数据定义：DBMS 提供数据定义语言 (DDL)
 - 数据组织、存储和管理：DBMS 要分类组织、存储和管理各种数据，包括数据字典、用户数据、数据的存取路径等
 - 数据操纵：DBMS 还提供数据操纵语言 (DML)
 - 数据库的事务管理和运行管理：数据库在建立、运行和维护时由 DBMS 统一管理、统一控制，以保证数据的安全性、完整性（一致性），以及多用户对数据并发操作时的数据库正确性（称为并发控制）和系统发生故障后的数据库正确性（称为恢复与备份）
 - 数据库的建立和维护
 - 其他功能

1.4.2 数据库管理系统——DBMS

■ DBMS 的组成

- 查询处理器
而找到一个
，使其执行
- 存储管理器
，或更新数
- 事务管理器
的事务不发
不会丢失

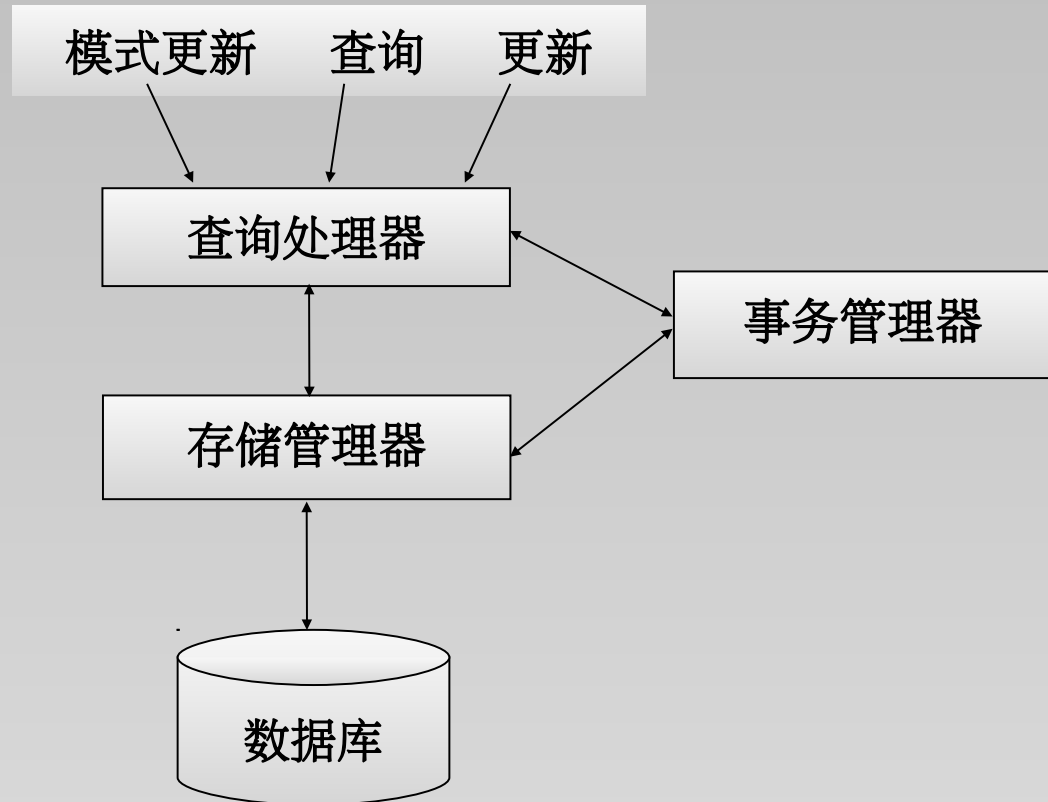


图 1-15 DBMS 的主要组成部分

1.4.2 数据库管理系统——DBMS

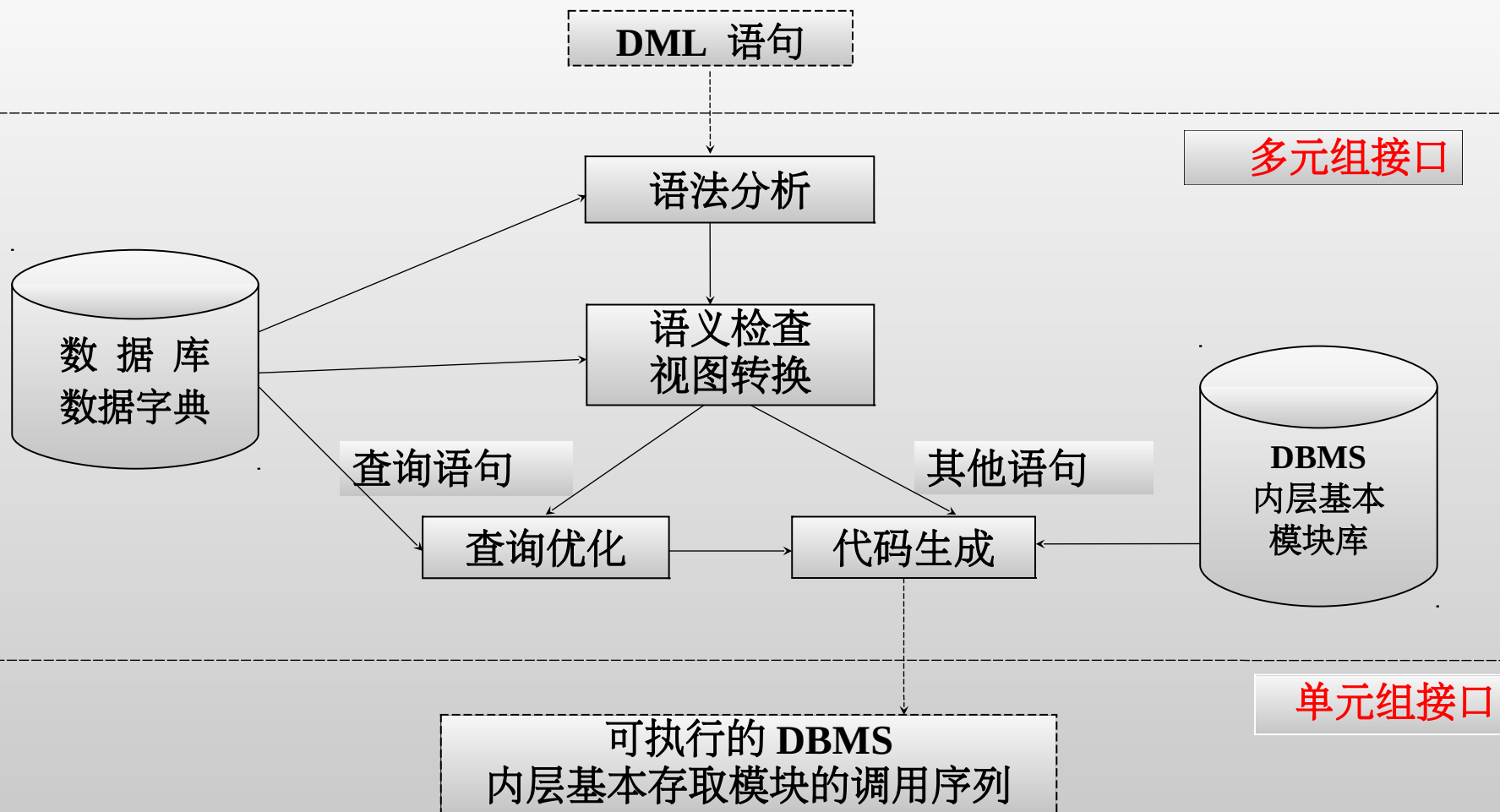


图 1-16 DML 语句处理过程

1.4.3 数据库系统的相关人员

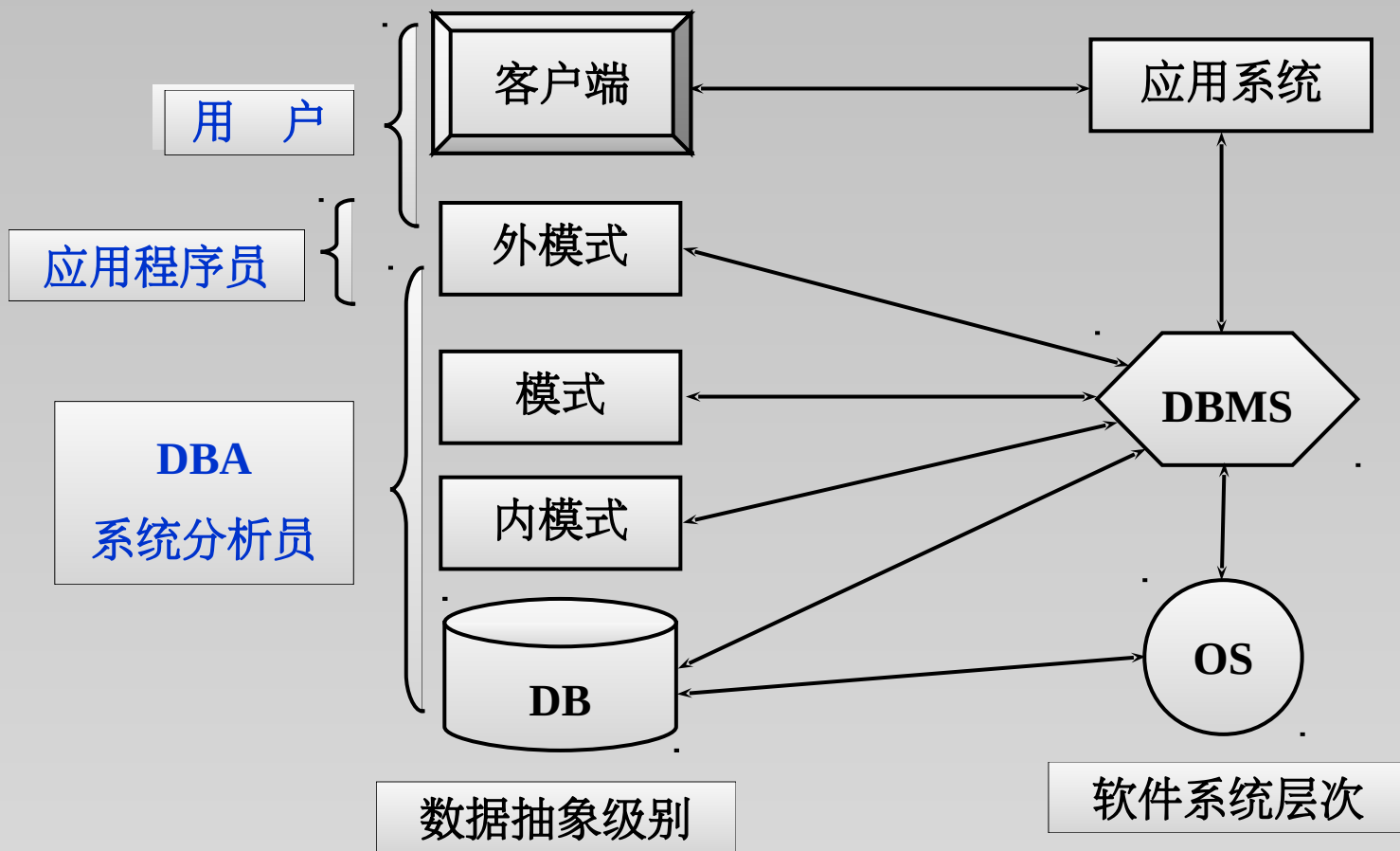


图 1-19 数据库系统中相关人员的数据视图

1.4.3 数据库系统的相关人员

■ 数据库管理员的主要职责

- 决定数据库中的信息内容和结构
- 决定数据库的存储结构和存取策略
- 定义数据的安全性要求和完整性约束条件
- 监控数据库的使用和运行
- 数据库的改进和重组重构