



数据库技术与应用

北京邮电大学计算机学院 肖达

xiaoda99@gmail.com

综合举例(续)

■ 计算机产品数据库

- Product(maker, model, type)
- PC(model, speed, ram, hd, price)
- Laptop(model, speed, ram, hd, screen, price)

Product

maker	model	type
A	1001	pc
A	1002	pc
A	2004	laptop
B	1003	pc
B	2005	laptop
C	1004	pc

PC

model	speed	ram	hd	price
1001	2.66	1024	250	2114
1002	2.10	512	250	995
1003	1.42	512	80	478
1004	2.80	1024	250	649

Laptop

model	speed	ram	hd	screen	price
2004	2.00	512	60	13.3	1150
2005	2.16	1024	120	17.0	2500



Product

maker	model	type
A	1001	pc
A	1002	pc
A	2004	laptop
B	1003	pc
B	2005	laptop
C	1004	pc

PC

model	speed	ram	hd	price
1001	2.66	1024	250	2114
1002	2.10	512	250	995
1003	1.42	512	80	478
1004	2.80	1024	250	649

Laptop

model	speed	ram	hd	screen	price
2004	2.00	512	60	13.3	1150
2005	2.16	1024	120	17.0	2500

- 哪些PC型号的速度不小于2.8?

$\pi_{\text{model}} (\sigma_{\text{speed} \geq 2.8} (\text{PC}))$

- 哪些制造商生产的笔记本的硬盘容量不小于100GB ?

$\pi_{\text{maker}} (\text{Product} \bowtie \sigma_{\text{hd} \geq 100} (\text{Laptop}))$

- 查询制造商B生产的所有类型产品的型号和价格。

$\pi_{\text{model, price}} (\sigma_{\text{maker}=\text{B}} (\text{Product}) \bowtie (\pi_{\text{model, price}} (\text{PC}) \cup \pi_{\text{model, price}} (\text{Laptop})))$

■ 象集 (Image Set)

- 关系 $R(X, Z)$, X, Z 是属性组, x 是 X 上的取值, 定义 x 在 R 中的象集为

$$Z_x = \{ t[Z] \mid t \in R \wedge t[X] = x \}$$

从 R 中选出在 X 上取值为 x 的元组, 去掉 X 上的分量, 只留 Z 上的分量

象集的示例

X	Z
姓名	课程
张军	物理
王红	数学
张军	数学

x=张军

Z_x

课程
数学
物理

**张军同学所选修
的全部课程**



除运算

- 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，但必须出自相同的值域。
- 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_x 包含 S 在 Y 上投影的集合。记做：

$$R \div S = \{t_r[X] \mid t_r \in R \wedge \Pi_Y(S) \subseteq Y_x\}$$



除运算示例

R

A	B	C
a1	b1	c2
a2	b3	c7
a3	b4	c6
a1	b2	c3
a4	b6	c6
a2	b2	c3
a1	b2	c1

S

B	C	D
b1	c2	d1
b2	c1	d1
b2	c3	d2

a1在R上的象集

b1	c2
b2	c3
b2	c1

a2在R上的象集

b3	c7
b2	c3

a3在R上的象集

b4	c6
----	----

$R \div S$

A

a1



除运算

如何得到选修了全部课程的学生？

做法：逐个考虑选课关系SC中的元组r，求r在姓名Sname上的分量x，再求x在选课关系中的象集课程C_x，若C_x包含了所有的课程C，则x是满足条件的一个元组

选修全部课程的学生

x同学所选修的全部课程

全部课程

$$\{ x \mid x=r[Sname] \wedge r \in SC \wedge C_x \supseteq C \}$$

除运算

姓名	课程
张军	物理
王红	数学
张军	数学
王红	物理
李铭	数学

÷

课程
数学
物理

=

姓名
张军
王红

[例10] 查询选修了全部课程的学生号码和姓名。

$\pi_{Sno, Cno} (SC) \div \pi_{Cno} (Course) \bowtie \pi_{Sno, Sname} (Student)$

举例：学生-课程数据库



- Student(Sno, Sname, Ssex, Sage, Sdept)
- Course(Cno, Cname, Cpno, Ccredit)
- SC(Sno, Cno, Grade)

[例7] 查询至少选修1号课程和3号课程的学生号码
首先建立一个临时关系K:

Cno
1
3

然后求: $\pi_{Sno, Cno}(SC) \div K$

Sno	Cno
200215121	1
200215121	2
200215121	3
200215122	2
200215122	3

200215121象集 {1, 2, 3}

200215122象集 {2, 3}

K={1, 3}

于是: $\pi_{Sno, Cno}(SC) \div K = \{200215121\}$

Product

maker	model	type
A	1001	pc
A	1002	pc
A	2004	laptop
B	1003	pc
B	2005	laptop
C	1004	pc

PC

model	speed	ram	hd	price
1001	2.66	1024	250	2114
1002	2.10	512	250	995
1003	1.42	512	80	478
1004	2.80	1024	250	649



Laptop

model	speed	ram	hd	screen	price
2004	2.00	512	60	13.3	1150
2005	2.16	1024	120	17.0	2500

查询卖所有型号的PC的制造商。

$\pi_{\text{maker}}(\text{Product} \div (\pi_{\text{model}}(\text{PC})))$

用基本运算表达除运算

完成			
Student	Task	DB项目	完成 ÷ DB项目
Fred	Database1	Task	Student
Fred	Database2	Database1	Fred
Fred	Compiler1	Database2	Sara
Eugene	Database1		
Eugene	Compiler1		
Sara	Database1		
Sara	Database2		

- “DB项目” 包含数据库项目的任务，除法的结果包含已经完成了数据库项目的学生



完成

Student	Task
Fred	Database1
Fred	Database2
Fred	Compiler1
Eugene	Database1
Eugene	Compiler1
Sara	Database1
Sara	Database2

DB项目

Task
Database1
Database2

完成 ÷
DB项目

Student
Fred
Sara

- 假定 a_1, \dots, a_n 是唯一于 R 的属性名字而 b_1, \dots, b_m 是 S 的属性名字
- 投影 R 于它的唯一属性上，并构造它们与 S 的元组的所有组合

$$T := \pi_{a_1, \dots, a_n}(R) \times S$$

T 表示所有学生与所有给定任务的组合的表

- 从这个关系中减去 R ，并做到唯一于 R 的属性名字的投影

$$U := T - R \quad V := \pi_{a_1, \dots, a_n}(U)$$

U 中都是 R 中没有出现的可能的组合。投影得到的学生与 DB 项目的某些组合未出现在 R 中（即为完成某些 DB 项目）

- 投影 R 到唯一于它的属性名字并减去 V :

$$W := \pi_{a_1, \dots, a_n}(R) - V$$

思考



- 怎样用集合并和差运算表达交运算？



补充：重命名运算

- ✓ 关系代数运算的结果没有可供引用的名字，使得复杂的查询显得非常冗长；
- ✓ 用小写希腊字母 ρ 表示的命名运算可对关系更名，或赋予关系代数运算结果一个名字：
 - ⑩ 对给定的关系代数表达式 E ，表达式 $\rho_x(E)$ 返回表达式 E 的结果，并把名字 x 赋给了它。
- ✓ 假设关系代数表达式 E 是 n 元的，则表达式：

$$\rho_x(A_1, A_2, \dots, A_n)(E)$$

返回表达式 E 的结果，并赋给它名字 x ，同时将 E 的各属性更名为 A_1, A_2, \dots, A_n 。

Product

maker	model	type
A	1001	pc
A	1002	pc
A	2004	laptop
B	1003	pc
B	2005	laptop
C	1004	pc

PC

model	speed	ram	hd	price
1001	2.66	1024	250	2114
1002	2.10	512	250	995
1003	1.42	512	80	478
1004	2.80	1024	250	649



Laptop

model	speed	ram	hd	screen	price
2004	2.00	512	60	13.3	1150
2005	2.16	1024	120	17.0	2500

查询至少出现在两款PC中的硬盘容量。

$$\pi_{PC1.hd}(\rho_{PC1}(PC) \bowtie_{PC1.hd = PC2.hd \wedge PC1.model \neq PC2.model} \rho_{PC2}(PC))$$

查询至少生产两款不同速度PC的制造商。

$$\pi_{R1.maker}(\rho_{R1}(Product \bowtie PC) \bowtie_{R1.speed \neq R2.speed \wedge R1.maker = R2.maker} \rho_{R2}(Product \bowtie PC))$$

Product

maker	model	type
A	1001	pc
A	1002	pc
A	2004	laptop
B	1003	pc
B	2005	laptop
C	1004	pc

PC

model	speed	ram	hd	price
1001	2.66	1024	250	2114
1002	2.10	512	250	995
1003	1.42	512	80	478
1004	2.80	1024	250	649



Laptop

model	speed	ram	hd	screen	price
2004	2.00	512	60	13.3	1150
2005	2.16	1024	120	17.0	2500

查询至少生产了两款速度不小于2.80的计算机（PC或笔记本）的制造商。

$$\rho_R(\pi_{\text{maker,model}}(\sigma_{\text{speed} \geq 2.8}(\text{Product} \triangleright \triangleleft (\pi_{\text{model,speed}}(\text{PC}) \cup \pi_{\text{model,speed}}(\text{Laptop}))))))$$

$$\pi_{\text{maker}}(\rho_{R1}(R) \triangleright \triangleleft R1.\text{maker} = R2.\text{maker} \text{ AND } R1.\text{model} \neq R2.\text{model} \rho_{R2}(R))$$



关系代数小结

■ 关系代数运算

➤ 关系代数运算

并、差、交、笛卡尔积、投影、选择、连接、除

➤ 基本运算

并、差、笛卡尔积、投影、选择

➤ 交、连接、除

可以用**5**种基本运算来表达

引进它们并不增加语言的能力，但可以简化表达

■ 关系代数表达式

➤ 关系代数运算经有限次复合后形成的式子



作业

- 计算机产品数据库，写出关系代数表达式：
 - 查询所有速度在**2.0**以上的产品（包括**PC**和笔记本）的型号和价格
 - 查询所有只卖笔记本而不卖**PC**的制造商。
 - 查询所有具有相同**CPU**速度和内存容量的**PC**型号对，每对只列出一次，如：列出了(i, j)就不要列出(j, i)。（提示：型号可以比较大小）
 - 查询最快速度的**PC** (提示：最快速度意味着不小于任何其它电脑的速度，可以从找出所有小于某一**PC**速度的**PC**的角度解决此问题)



第二章 关系数据库

2.1 关系数据结构及形式化定义

2.2 关系操作

2.3 关系代数

2.4 关系的完整性



2.4.1 关系的三类完整性约束

- 实体完整性和参照完整性：

关系模型必须满足的完整性约束条件

称为关系的两个不变性，应该由关系系统自动支持

- 用户定义的完整性：

应用领域需要遵循的约束条件，体现了具体领域中的语义约束



2.4 关系的完整性

2.4.1 关系的三类完整性约束

2.4.2 实体完整性

2.4.3 参照完整性

2.4.4 用户定义的完整性



2.4.2 实体完整性

规则2.1 实体完整性规则 (**Entity Integrity**)

若属性 A 是基本关系 R 的主属性, 则属性 A 不能取空值

例:

SAP(SUPERVISOR, SPECIALITY, POSTGRADUATE)

POSTGRADUATE:

主码 (假设研究生不会重名)

不能取空值



实体完整性(续)

实体完整性规则的说明

- (1) 实体完整性规则是针对基本关系而言的。一个基本表通常对应现实世界的一个实体集。
- (2) 现实世界中的实体是可区分的，即它们具有某种唯一性标识。
- (3) 关系模型中以主码作为唯一性标识。
- (4) 主码中的属性即主属性不能取空值。

主属性取空值，就说明存在某个不可标识的实体，即存在不可区分的实体，这与第（2）点相矛盾，因此这个规则称为实体完整性



2.4 关系的完整性

2.4.1 关系的三类完整性约束

2.4.2 实体完整性

2.4.3 参照完整性

2.4.4 用户定义的完整性

1. 关系间的引用

- 在关系模型中实体及实体间的联系都是用关系来描述的，因此可能存在着关系与关系间的引用。

例1 学生实体、专业实体

学生（学号，姓名，性别，**专业号**，年龄）

专业（**专业号**，专业名）



- ❖ 学生关系引用了专业关系的主码“专业号”。
- ❖ 学生关系中的“专业号”值必须是确实存在的专业的专业号，即专业关系中有该专业的记录。



关系间的引用(续)

例2 学生、课程、学生与课程之间的多对多联系

学生（学号，姓名，性别，专业号，年龄）

课程（课程号，课程名，学分）

选修（学号，课程号，成绩）

例3 学生实体及其内部的一对多联系

学生（学号，姓名，性别，专业号，年龄，班长）

学号	姓名	性别	专业号	年龄	班长
801	张三	女	01	19	802
802	李四	男	01	20	
803	王五	男	01	20	802
804	赵六	女	02	20	805
805	钱七	男	02	19	

❖ “学号”是主码，“班长”是外码，它引用了本关系的“学号”

❖ “班长”必须是确实存在的学生的学号



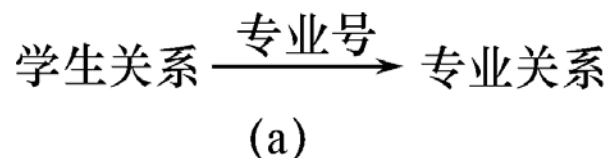
2. 外码 (Foreign Key)

- 设 F 是基本关系 R 的一个或一组属性，但不是关系 R 的码。如果 F 与基本关系 S 的主码 K_s 相对应，则称 F 是基本关系 R 的**外码**
- 基本关系 R 称为**参照关系** (Referencing Relation)
- 基本关系 S 称为**被参照关系** (Referenced Relation) 或**目标关系** (Target Relation)

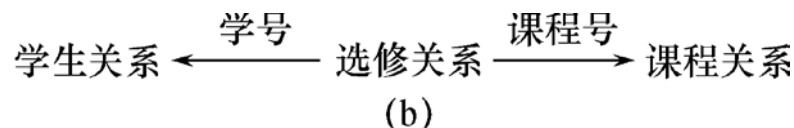


外码(续)

- 例1：学生关系的“专业号”与专业关系的主码“专业号”相对应
 - “专业号”属性是学生关系的外码
 - 专业关系是被参照关系，学生关系为参照关系

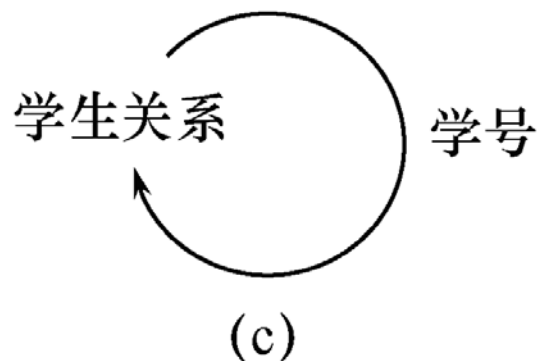


- 例2：选修关系的“学号”与学生关系的主码“学号”相对应
- 选修关系的“课程号”与课程关系的主码“课程号”相对应
 - “学号”和“课程号”是选修关系的外码
 - 学生关系和课程关系均为被参照关系
 - 选修关系为参照关系



外码(续)

- 例3: “班长”与本身的主码“学号”相对应
 - “班长”是外码
 - 学生关系既是参照关系也是被参照关系





外码(续)

- 关系 R 和 S 不一定是不同的关系
- 目标关系 S 的主码 K_s 和参照关系的外码 F 必须定义在同一个（或一组）域上
- 外码并不一定要与相应的主码同名

当外码与相应的主码属于不同关系时，往往取相同的名字，以便于识别

思考题

■ 计算机产品数据库有哪些参照？

Product

maker	model	type
A	1001	pc
A	1002	pc
A	2004	laptop
B	1003	pc
B	2005	laptop
C	1004	pc

PC

model	speed	ram	hd	price
1001	2.66	1024	250	2114
1002	2.10	512	250	995
1003	1.42	512	80	478
1004	2.80	1024	250	649

Laptop

model	speed	ram	hd	screen	price
2004	2.00	512	60	13.3	1150
2005	2.16	1024	120	17.0	2500



3. 参照完整性规则

规则2.2 参照完整性规则

若属性（或属性组） F 是基本关系 R 的外码，它与基本关系 S 的主码 K_s 相对应（基本关系 R 和 S 不一定是不同的关系），则对于 R 中每个元组在 F 上的值必须为：

- 或者取空值（ F 的每个属性值均为空值）
- 或者等于 S 中某个元组的主码值



参照完整性规则(续)

- 例1：学生关系中每个元组的“专业号”属性只取两类值：
 - 空值，表示尚未给该学生分配专业
 - 非空值，这时该值必须是专业关系中某个元组的“专业号”值，表示该学生不可能分配一个不存在的专业
- 例2：选修（学号，课程号，成绩）

“学号”和“课程号”可能的取值：

 - 选修关系中的主属性，不能取空值
 - 只能取相应被参照关系中已经存在的主码值



参照完整性规则(续)

例3：学生（学号，姓名，性别，专业号，年龄，班长）

“班长”属性值可以取两类值：

- （1）空值，表示该学生所在班级尚未选出班长
- （2）非空值，该值必须是本关系中某个元组的学号值

关系的完整性(续)



2.4.1 关系的三类完整性约束

2.4.2 实体完整性

2.4.3 参照完整性

2.4.4 用户定义的完整性



2.4.4 用户定义的完整性

- 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
- 关系模型应提供定义和检验这类完整性的机制，以便使用统一的系统的方法处理它们，而不要由应用程序承担这一功能



用户定义的完整性(续)

例： 课程(课程号， 课程名， 学分)

- “课程号” 属性必须取唯一值
- 非主属性 “课程名” 也不能取空值
- “学分” 属性只能取值{1, 2, 3, 4, 5}



SQL Server介绍



SQL Server发展历程

- 1988年 SQL Server
 - 与Sybase共同开发的、运行于OS/2上
- 1993年 SQL Server 4.2
 - 一种功能较少的桌面数据库，能够满足小部门数据存储和处理的需求。这是支持Windows NT的第一个服务器版本
- 1994年
 - 微软与Sybase终止合作关系
- 1995年 SQL Server 6.05
 - 对核心数据库引擎做了重大的改写。
- 1996年 SQL Server 6.5
 - SQL Server逐渐突显实力，以至于Oracle推出了运行于NT平台上的7.1版本作为直接的竞争
- 1998年 SQL Server 7.0
 - 再一次对核心数据库引擎进行了重大改写。
- 2005年 SQL Server 2005
 - 对SQL Server的许多地方进行了改写，例如通过名为集成服务（Integration Service）的工具来加载数据，引入了.NET Framework。
- 2008年 SQLServer 2008



SQL Server vs Oracle

■ 可扩展性和灵活性

- **Oracle**有许多部件，非常适宜为大型公司提供大型解决方案。**Oracle**还非常灵活，允许使用者按需要添加工具，比**SQL Server**具有更强的适应性。
- 例如，无论用户是否需要使用新的.NET特性，**SQL Server 2005**都要求用户必须在服务器上安装.NET Framework。

■ 使用和开发难度

- 从开发者的角度来看，**Oracle**在许多方面都算不上是用户友好的，例如它专门的**SQL Query**工具、**XML**和**Web**技术工具
- **Oracle**比**SQL Server**安装配置复杂。

■ 成本

- **SQL Server**向来是一次性购买的解决方案，如果购买了正确版本的**SQL Server**，那么用于分析数据或从一个数据源（如**Excel**）向**SQL Server**中复制数据的所有工具都将包含于其中。而对于**Oracle**，必须额外购买所需要的每一个附加特性。



SQL Server vs MySQL

	MySQL	SQL Server
公司	Oracle	Microsoft
操作界面	命令行（可装图形操作软件）	图形化界面
费用	开源、部分免费、收费低（Community版本）	非开源、除express版外价格惊人
性能	速度更快	稳定性更佳
是否跨平台	跨平台，在Linux上运行更好	只能建立在Windows上
适用范围	小型项目	中小型项目
恢复性	不慎损坏，所有数据丢失	时刻监测数据交换点，把损坏的过程保存下来



其他不同

- 支持字段类型不同。如MySQL支持enum和set类型，SQL Server不支持。MySQL不支持nchar，nvarchar，ntext类型，SQL Server支持。
- MySQL默认不支持外键和级联操作，需要修改配置文件。
- 具体语句与方法不同

Microsoft® SQL Server™ 2005



Enterprise Data Management

- 针对企业应用的高可用性
- 关键的安全及性能特性
- 聚焦于自管理与优化

Developer Productivity

- 与 Visual Studio 及 .NET 集成
- 内含的 XML 技术
- 与 Web Services 的互操作

Business Intelligence

- 整合及转换数据
- 分析，存储和挖掘数据
- 报告及与数据的交互

Microsoft® SQL Server™ 2005



Enterprise Data Management



可用性和安全性的提升:

- 数据库镜像, 增强了可用性的在线操作
 - 数据的安全性和私有性: 加密, 增强的审核
- 持续聚焦于可管理性

- 针对关系型和**OLAP**数据库的集成管理工具集
- 扩展的自调整

性能和可扩展性

- 伴随商业发展而扩展: 分区, 快照
- 从移动设备到 **64-bit** 数据中心

Microsoft® SQL Server™ 2005



Developer Productivity

利用现有的开发技能

- 与 Visual Studio 的集成
- 多种语言的支持: T-SQL, VB.NET, C#...

跨越任何平台共享数据

- 对 XML, Web services 的内置支持
- 与任何平台, 应用程序的互操作性

可扩展性

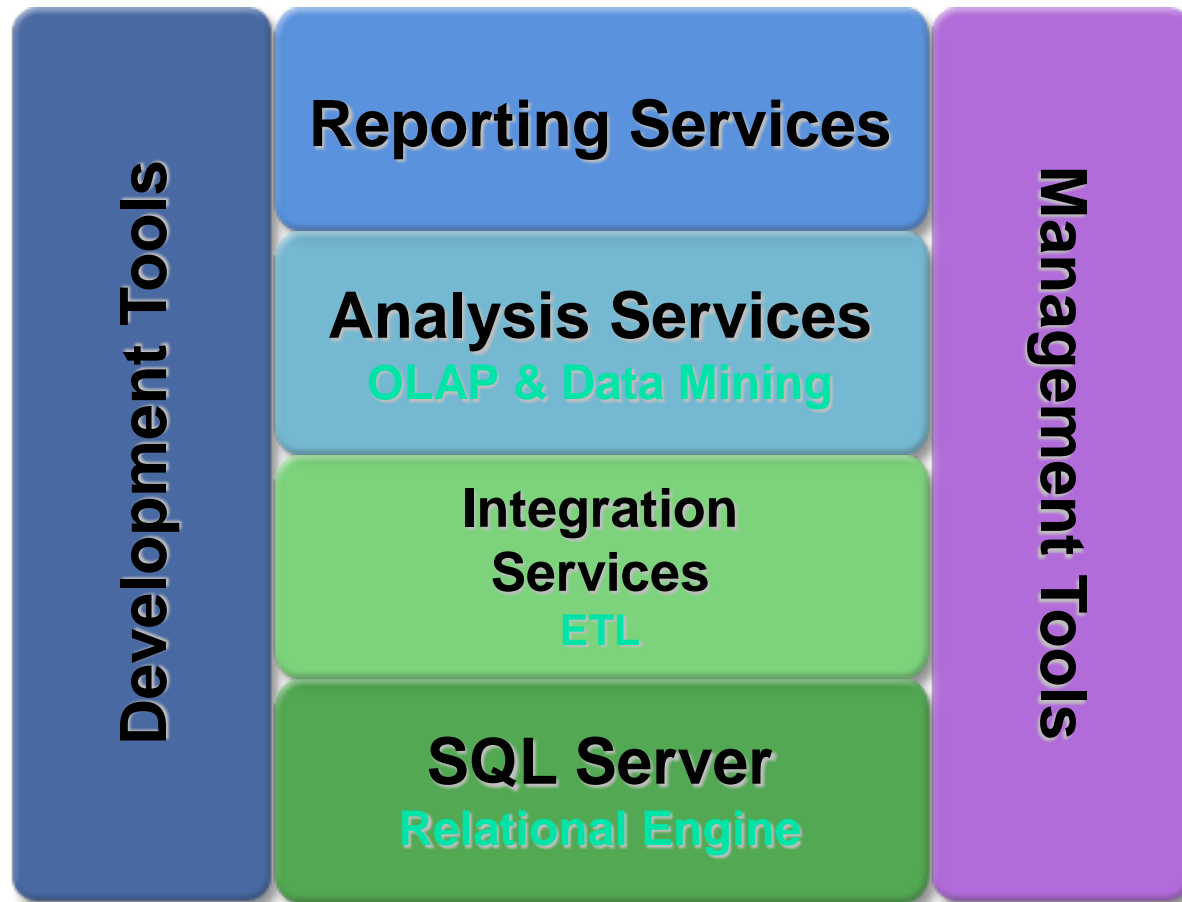
- 内含的 XML 数据类型, 用户定义的数据类型
- 新的分布式应用程序架构: Service Broker



SQL Server 2005



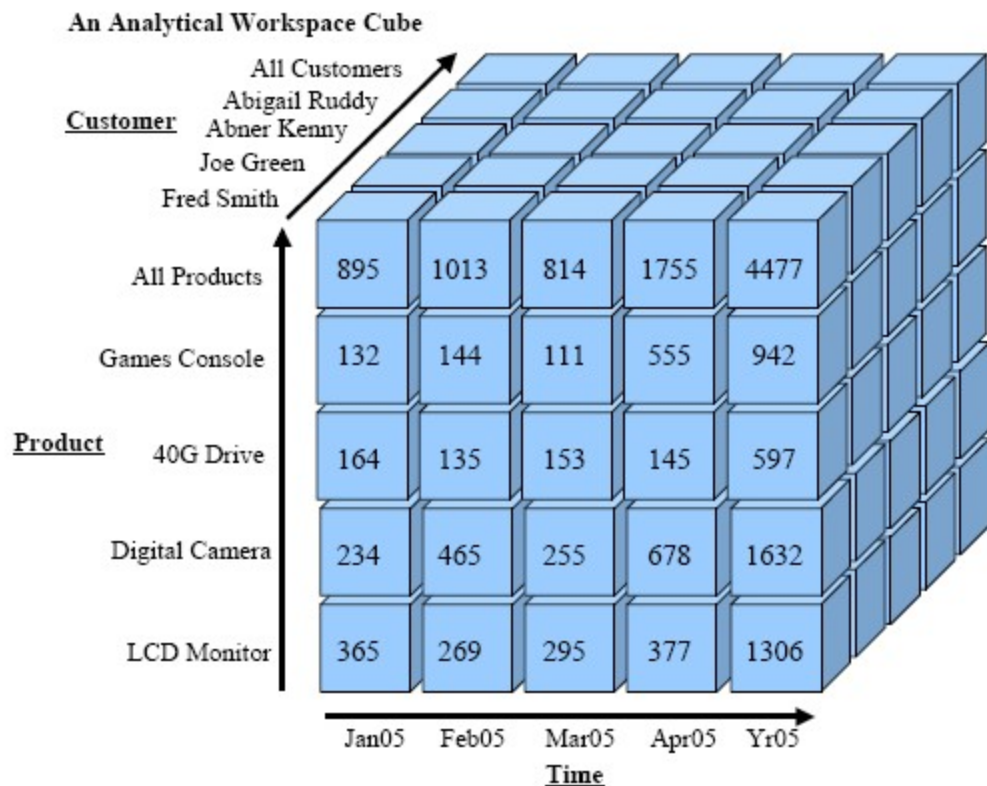
一个完整的企业数据管理和 BI 解决方案





SQL Server 2005平台主要部分

- **数据库引擎**
 - 即传统意义上的数据库，用于存储、处理和保护数据的核心服务。
- **分析服务: Analysis Services**
 - 为商业智能应用程序提供了联机分析处理 (OLAP) 和数据挖掘功能。
- **集成服务: Integration Services**
 - 可从不同的数据源提取、转换和合并数据，并转移到目标。
- **复制**
 - 用于在数据库间复制和分发数据，并同步操作以维持一致性。
- **报表服务: Reporting Services**
 - 用于创建和管理来自关系数据库的报表。
- **通知服务: Notification Services**
 - 用于开发和部署可生成并发送消息的服务。
- **服务代理: Service Broker**
 - 帮助开发人员生成可伸缩的消息队列。
- **全文搜索**
 - 快速、灵活检索文本数据。

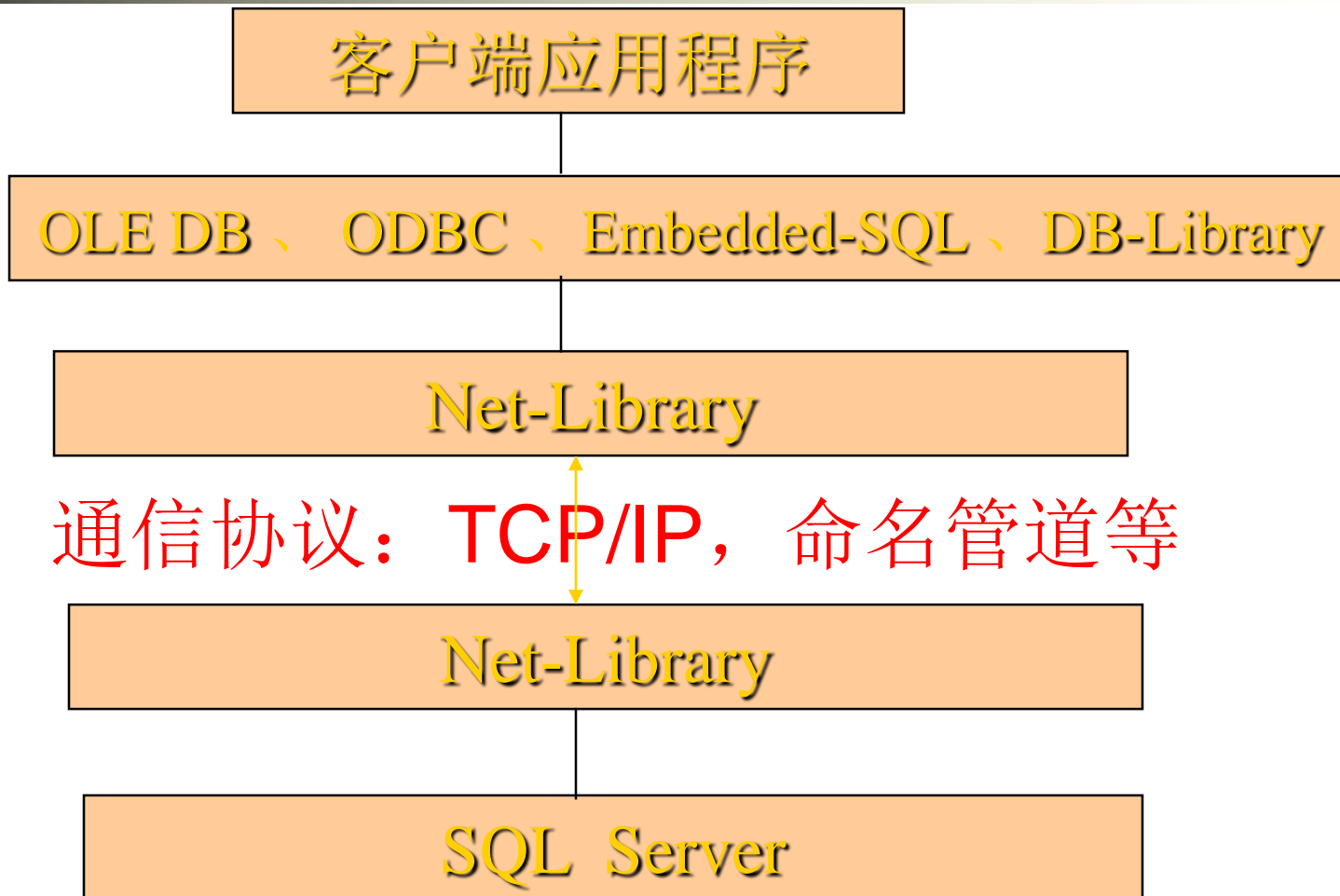




SQL Server架构

- 主从式（Client/Server）结构
- 数据存取接口：OLE DB、ODBC、Embedded-SQL、DB-Library
- 沟通的桥梁：Net-Library
 - 在同一台计算机中，客户端和服务端之间的通信是通过Windows操作系统的程序间通信(IPC)来实现;在不同的计算机之间，则是通过Network IPC来通信
 - 客户端和服务端有了可互通的接口后，建立两边IPC通信管道的工作是由Net-Library（网络函数库）来负责的

用户





多执行实例

- 可在同一台计算机上同时执行多个SQL Server 2005的实例，就好象装在不同计算机上的SQL Server上一样独立运作
- 要使用多执行实例的功能，必须重复安装SQL Server 2005. 缺省实例即名称沿用该服务器的计算机名称；而安装后面的执行实例，就需另行指定不同的名称

SQL Server 2005版本



- **SQL Server 2005 Enterprise Edition : 企业版 , 超大型企业商业应用**
- **SQL Server 2005 Standard Edition : 标准版 , 中小型企业商业应用**
- **SQL Server 2005 Workgroup Edition : 工作组版 , 小型企业**
- **SQL Server 2005 Developer Edition : 开发版 , 软件供应商、企业开发人员使用**
- **SQL Server 2005 Express Edition : 简易版 , 非专业人员、客户端程序编程爱好者**

Enterprise Edition、Standard Edition 和 Workgroup Edition 在生产服务器环境中安装和使用。



SQL Server 2005的数据库及数据库对象

数据库的类别

(1) 物理数据库和逻辑数据库

物理数据库由构成数据库的物理文件构成。SQL Server 2005的一个物理数据库中至少有一个数据库信息文件和一个数据库事务日志文件。物理数据库由**DBA**（数据库管理员）负责创建和管理。

逻辑数据库是数据库中用户可见的表或视图，用户利用逻辑数据库的数据库对象，存储或读取数据库中的数据。

(2) 系统数据库和用户数据库

系统数据库是由系统创建和维护的数据库。系统数据库中记录着SQL Server 2005的配置情况、任务情况和用户数据库的情况等系统管理的信息，它实际上就是我们常说的数据字典。

用户数据库是根据管理对象要求创建的数据库，用户数据库中保存着用户直接需要的数据信息。

SQL Server 2005的系统数据库



1. Master数据库

主文件名为Master.mdf，日志文件为Masterlog.ldf。Master中内含许多系统表，用来跟踪和记录SQL Server相关信息。

如：所有的登录名或用户ID所属的角色、所有的系统配置设置、服务器中的数据库的名称及相关信息、数据库的位置。

2. Msdb数据库

主文件名为Msdb.mdf，日志文件名为Msdb.ldf。Msdb由SQL Server代理服务使用。Msdb中记录着任务计划信息、事件处理信息、数据备份及恢复信息和警告及异常信息。

3. Model数据库

主文件是model.mdf，日志文件为model.ldf。Model数据库是SQL Server 2005为用户数据库提供的样板，新的用户数据库都以model数据库为基础。

4. tempdb数据库

主文件名和日志文件名分别为tempdb.dbf和tempdb.ldf。tempdb是一个共享的工作空间，SQL Server 2005中的所有数据库都可以使用它，它为临时表和其他临时工作提供了一个存储区。

SQL Server 2005的数据库对象



- (1) **表和视图**：表是在数据库中存放的实际关系。视图是为了用户查询方便或根据数据安全的需要而建立的虚表。
- (2) **角色**：由一个或多个用户组成的单元，也称职能组。一个用户可以成为多个角色中的成员。
- (3) **索引**：来加速数据访问和保证表的实体完整性的数据库对象。
- (4) **存储过程**：通过Transact-SQL编写的程序。包括系统存储过程和用户存储过程：系统存储过程是由SQL Server 2005提供的，用户过程是由用户编写的，它可以自动执行过程中安排的任务。
- (5) **触发器**：一种特殊类型的存储过程，当表中发生特殊事件时执行。触发器主要用于保证数据的完整性。
- (6) **约束**：约束规则用于加强数据完整性。



实验1 认识DBMS

- 目的：通过安装和使用**SQL Server**（也可安装**MySQL**），初步了解**DBMS**的工作环境和系统架构。
- 安装
 - 安装**SQL Server**数据库及相关组件
- 创建用户、赋予权限
 - 利用**Management Studio**创建一个用户，赋予**DBA**权限（**sysadmin**角色）
- 服务的启动和停止
 - 利用**Configuration Manager**启动和停止数据库服务，并验证效果
- 创建数据库和表
 - 利用**Management Studio**创建选课数据库和**Student**表，定义表的属性并输入数据（可创建自己设计的数据库和表）



实验报告

■ 内容

- 实验环境
- 实验内容与完成情况
- 出现的问题及解决方法

■ 要求

- 实验过程中要进行抓图，把某些关键步骤的屏幕截图贴到实验报告中
- 列出出现的问题以及采取的解决方法
- 独立完成

■ 提交

- 11月3日晚12:00前发到助教邮箱
 - 15651638081@163.com
 - 命名格式：学号+姓名+实验N.doc/docx/rar