

第一章 数据库原理

第一节 数据库简介（理解）

1. 数据库的重要性：数据库无处不在：它们是电子商务和其他 WEB 应用程序的主要组成部分，是企业操作和决策支持应用程序的核心部分。

2. 数据库的目的：数据库的主要目的是帮助用户更好地管理数据。

3. 关于列表的问题：过于简单的列表可能会导致数据的不一致性和其他问题，如删除行会丢失过多的数据，改变行会导致不一致的数据，插入行会导致数据遗漏。通常情况下，如果一个列表含有的数据指示两个或两个以上的不同事情时，修改数据就会出现这个问题。

4. 可以用关系模型的方法解决列表所产生的问题。一个关系数据库含有一个独立表的集合。在很多情况下，每个表中的数据有且仅有一个主题。如果一个表含有两个或多个主题，我们就需要将其分割为两个或多个表。

第二节 数据库系统的概念

1. 数据库系统的 4 个基本元素是：用户、数据库应用程序、数据库管理系统（DBMS）和数据库。

2. 数据库：在绝大多数情况下，数据库就是关联记录项的自描述集合。对于所有的关系数据库，该定义可以修改为：数据库就是关联表的自描述集合。自描述意味着数据库本身含有对数据库结构的描述。

关于数据库结构的数据称为元数据。元数据的示例是表名、列名和列所属的表、表和列的属性等。应用元数据用于描述应用程序元素，如表单和报表。

3. DBMS 的目标是创建、处理和管理数据库。

DBMS 的功能包括：（1）创建数据库（2）创建表（3）创建支持结构（如索引等）

（4）读取数据库数据（5）修改（插入、更新或删除）数据库数据（6）维护数据库结构

（7）执行规则（8）并发控制（9）提供安全性（10）执行备份和恢复

4. 应用程序的功能包括：（1）创建并处理表单（2）处理用户调查（3）创建并处理报表

（4）执行应用逻辑（5）控制应用

第二章 关系模型

一、关系

- 关系是具有如下特征的二维表：
- 表的每一行存储了某个实体某个属性的数据。
- 表的每类包含了用于表示实体某个属性的数据。
- 关系中的每个单元格的值必须单值。
- 任意一列中所有条目的类型必须为单值。
- 行与行之间顺序任意。
- 表中任意两行不能具有完全相同的数据值。

EmployeeNumber	FristName	LastName	Phone	Email
100	Jerry	Johnson	236-9987	JJ@somewhere.com
200	Mary	Abernathy	444-8898	MA@somewhere.com
300	Liz	Smathers	777-5643	LS@somewhere.com

(每个单元格都只有唯一的值，每列中所有实体的类型都是一样的。列名都是唯一的，即使我们交换任意两行或两列的顺序，也不会丢失任何信息。表中，任意两行都不重复。所以，为关系)

二、键的类型

- 1.键 (Key):是关系中用来标识行的一列或多列。可以是唯一的，也可以是不唯一的。
- 2.复合键: 包含两个或更多属性的键。
- 3.候选键: 是唯一标识关系中每一行的键。可以是一列，也可以是复合键。从候选键里选主键。
- 4.代理键: 是具有 DBMS 分配的唯一标识符的列，该标识符已经作为主键添加到表中。是短暂的数字，并且永远不变，他们是理想的主键。
- 5.外键与参照完整性:

Foreign key: 将关系 CUSTOMER 中的主键 CustomerNumber 放入关系 ENROLLMENT 中。在这种情况下，关系 ENROLLMENT 中的属性 CustomerNumber 就被称为外键。

Reference integrity: 关系 ENROLLMENT 的 CustomerNumber 的值必须在 CUSTOMER 的 CustomerNumber 中存在对应项。这样的规则被称为参照完整性。

三、理解函数依赖

如: $A \rightarrow B$ 可以读作 A 决定 B, B 依赖于 A, A 为决定因子。

$(A, B) \rightarrow C$ 说明复合的 (A, B) 是 C 的决定因子。不可说 $A \rightarrow C$ 或 $B \rightarrow C$ 。

$A \rightarrow (B, C)$ 也可以说 $A \rightarrow B$ 和 $A \rightarrow C$ 。

四、规范化 (考试重点: 要把示例看懂, 考试的要求和书本上是一致的。)

规范化定义: 将一个具有多个主题的表或关系分割为一组表, 使得每一个表只有一个主题。规范化过程:

- (1) 确认关系的所有候选键。
- (2) 确认关系中所有函数依赖关系。
- (3) 检查函数依赖关系的决定因子。
- (4) 根据需要, 多次重复步骤。
- (5) 直至每个关系的决定因子都是候选键。

例: PRESCRIPTION 关系的示例数据

PrescriptionNumber	Date	Drug	Dosage	CustomerName	CustomerPhone	CustomerEmail
P10001	10/17/2006	DrugA	10mg	XY	575-323-2233	XY@somewhere.com
P10003	10/17/2006	DrugB	35mg	WY	575-334-7786	WY@somewhere.com
P10004	10/17/2006	DrugA	20mg	SG	575-323-2233	SG@somewhere.com
P10007	10/18/2006	DrugC	20 mg	JH	575-334-7786	JH@somewhere.com

步骤 1: 根据规范化的过程, 首先确定所有的候选键。考虑一切可能性, 从左至右, 一个一个分析。先分析单独一列, 再分析组合键。经过分析, 得出 PRESCRIPTION 的唯一候选键为 PrescriptionNumber.

步骤 2: 根据规范化的过程, 可以确定所有的函数依赖关系。

Drug→Dosage

Customer Email→(Customer Name, Customer Phone)

步骤 3: 决定存在的决定因子是否为候选键。

Customer Email 是一个决定因子, 不是候选键。

CUSTOMER (CustomerEmail, CustomerPhone, CustomerName)

PRESCRIPTION (PrescriptionNumber, Date, Drug, Dosage, CustomerName)

PRESCRIPTION 中的 CustomerEmail 必须在 CUSTOMER 的值中找到对应项。

补充:

1. 第一范式 (1NF) 无重复的列

在任何一个关系数据库中, 第一范式 (1NF) 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。

2. 第二范式 (2NF) 属性完全依赖于主键, 要求数据库表中的每个实例或行必须可以被唯一地区分。

3. 第三范式 (3NF) 属性不依赖于其它非主属性。简而言之, 第三范式 (3NF) 要求一个数据库表中不包含已在其它表中已包含的非主关键字信息。

所有的高级范式要满足低级范式。

五、怎么创建报表, 要了解过程。(看书, 这个不好整理出来) P91——2.5.5

第三章 结构化查询语言

1. SQL 并不是完整的编程语言, 而更像是一种数据子语言。SQL 是面向文本的。

2. SQL 命令可以划分为两个主要的类别: 用于定义数据库结构的数据定义语言 (DDL) 以及用于查询和修改数据库的数据操作语言 (DML)。

3. 外键不需要与它所引用的主键同名。(p106)

4. 业务规则 (p109)

(1) 如果要删除表中的某一行, 而该行又连接到另外一个表, 则禁止对该表这一行进行删除操作。

(1) 如果删除了表中的某一行, 则另一个表中连接到这个已经删除的表的行的所有行也将随之被删除。

5. SQL DDL 用于创建和改变数据库结构, 并且用于插入、修改和删除表中的数据。

SQL CREAT TABLE 语句用来创建表的结构, 这个语句的基本结构参照 (p110)

列定义的 3 个部分为列名、列数据类型和对列值的约束。

6. 本书中讨论的列约束有 PRIMARY KEY、NOT NULL、NULL、UNIQUE。除了这些列约束之外, 还有 CHECK 列约束。而 DEFAULT (不是列约束) 可用于设置初始值。(p111)

7. 最基本的数据类型（具体参照 p114）

整数（Int）

字符（Char）

时间（Date Time）

数字（Numeric）

注意：Phone（Char）因为电话号码有分隔符

其他的数据类型参见书上

8. 使用表约束定义主键(p115)

利用表约束创建主键有 3 个优点：

（1）这是定义复合键的要求，因为 PRIMARY KEY（主键）列的约束不可以用于多个列。

（2）其次开发人员可以使用表约束来命名定义主键的约束。

（3）可以很方便的在一些 DBMS 产品中定义代理键。（p116）

9. 使用表约束定义外键(p117)

在代理键永不变的情况下，ON UPDATE 短语被设置为 NO ACTION（P118）

（1）ON DELETE CASCADE 表示删除表中的某一行时，必须级联删除

（2）ON DELETE NO ACTION 是默认设置，表示 A 表连接到 B 表，则不允许删除 A 表的行。

10. 不同于 SQL Server2005 和 MySQL，MS Access 不支持 SQL 脚本。可以使用 Create Table 命令创建表，并且使用 Insert 命令插入数据，但一次只能执行一条 SQL 命令。

11. SQL 语句

（1）SQL DML 用于查询数据库以及修改表中的数据。

（2）3 种可能的数据修改操作：插入---更新---删除

（3）可以使用 SQL INSERT 命令为关系添加数据。（p121）

（4）SQL 查询语句 Select/from/where（p124）

删除重复行 DISTINCT。

读取指定行用 WHERE 界定条件，指定列则不用 WHERE 界定。（p127）

使用 AND 关键字在 WHERE 字句中放置更多的条件，意思是选择满足所有的条件。OR 则是选择满足任一个条件的行。（p129）

（5）关键字 IN，指定列必须是值集中的某个值。（p130）

查询表中值不是所列值的行，可以使用 NOT IN。

IN 列值等于所列项的某一个值。

NOT IN 列值不等于所列值的任意一个值。

这有点类似于 AND 和 OR。

（6）BETWEEN AND 指定值的范围

LIKE 查询部分匹配的值

SQL 通配字符是下划线_（一个未指定的字符）和%（一些列或者多个未指定的字符）具体例子参照（p132）

（7）IS NULL 查找含有 NULL 值的行（p133）

（8）结果排序 ORDER BY ASC（升序）DESC（降序）

(9) SQL 的包括内置函数: COUNT、SUM、AVG、MAX、MIN。这些事针对 SELECT 语句结果进行操作的。

1) COUNT 用于任何数据类型, 其他四个只能对整数、数值、其他数字类型执行操作。

2) COUNT 函数统计了结果集中行的数目, SUM 一个数值列行的总和 (p136)

注意列的名称是不允许和内置函数一起混合使用的 (p137)

(10) GROUP BY 语句分组

1) HAVING 语句应用某些条件, 进一步限制所得结果 (p138)

2) GROUP BY 和 WHERE 同时使用的时候, 首先运用 WHERE

(11) 用子查询处理多个表, 一级查询内容来自一个表。具体例子 p139

(12) 使用连接查询多个表, 查询内容包括多个表的内容。p141

子查询只有在结果来自一个表的情况下才有用, 要显示两个或者多个表中的数据就不能用只查询, 要用到连接操作。

连接的基本思想是: 在连接原来两个或者多个关系的内容基础上再建立一个新关系。

1) 这里介绍了 JOIN IN 语法, 连接一个表和连接两个表 (p145)

2) 外连接, 这样可以避免数据的丢失 (p149)

(13) 修改删除表的数据 (p150)

用 SQL UPDATE SET 修改表中已有的数据的值

(14) 删除数据

DELETE 语句

DROP TABLE 删除表的结构和所有表的数据 (p153)

ALTER TABLE 用于添加、修改和删除列和约束。

第四章 数据建模与实体—关系模型

数据库系统开发的三个主要步骤: 需求, 设计, 实现。

1. 需求分析阶段: 设计者和系统用户进行交流, 获得抽样表单、报表、查询和有关数据跟新活动的说明。

2. 设计阶段: 数据库模型被转换成特定的数据库设计。包括数据库中各种表的设计、关联和约束的设计。

3. 实现阶段: 构建数据库, 并向其中填充数据; 同时建立各种查询、表单和报表; 此外还将编写程序。

4.1 需求分析阶段

1. 数据库应用程序的需求分析来源: 与用户交流、表单、报表、查询、用例、业务规则。

2. 需求分析阶段的工作:

(1) 与用户交流

(2) 从用户那里得到已有的示例表单、报表和查询等, 询问用户是否需要修改现有表单以及是否需要添加新的表单、报表和查询。

(3) 用例描述用户将如何使用新信息系统的各个特征和功能。

(4) 系统开发人员确定限定数据库操作的业务规则。

4.2 实体-关系数据模型

1. 数据需求转化为数据模型。

2. 创建数据模型的方法：实体-关系模型等。

3. E-R 模型中最重要的元素包括：实体、属性、标识符、关系。

4. 实体：用户希望跟踪的对象。给定类型的实体被分为实体集，实体集是实体的集合，由该集合中实体的结构来表示。实体集中有许多实体的实例。

5. 属性：实体都有属性，用来描述实体的特征。E-R 模型中假定实体集的所有实例都具有相同的属性。

6. 标识符：实体都含有标识符，用于给不同的实体实例进行命名或标识的属性。实体实例的标识符可以由实体的一个或多个属性构成。标识符可以唯一或不唯一。但是关系模型中的主键或候选键必须具有唯一性，而标识符不一定。

7. 关系：实体之间通过关系进行关联。

(1) 关系中实体集的数目称为这个关系的度数。度数为 2 的关系叫做二元关系，度数为 3 的关系称为三元关系。

(2) 二元关系的三种类型：一对一 1: 1； 一对多 1: N； 多对多 N:M

(3) 最大基数：关系菱形内的数字表示关系一侧可以出现的示例的最大数目。

(4) 最小基数，必须参与到关系实例中的实体实例的最小数目。

(5) 在关系直线上加“1”标明该关系中必须存在一个实体，加“椭圆”标明该关系中可能存在一个或不存在任何实体。

4.3 实体-关系图

实体-关系图的表示：实体集使用矩形表示，关系通过菱形表示，关系的最大基数在菱形内显示出来，最小基数则通过实体附近的椭圆或 1 的标记来表示；实体的名称写在矩形内，关系的名称写在菱形旁。

4.3.1 E-R 模型的不同版本：

1. Information Engineering (IE)，由 James Martin 于 1990 年开发的。该模型使用“鸟足 (crow's feet)”来显示关系的多个方面，因此有时候又被称作鸟足模型，且易于理解。

2. 1993 年，National Institute of Standards and Technology 宣布 IDEF1X 成为国家标准。该标准融入了基本的 E-R 模型思想，但使用了另一种图形化的符号，因此使得其更加让人难以理解和使用。

3. UML (Unified Modeling Language)，采用了 E-R 模型，但是在把面向对象的编程方法引入到该模型中的同时，该方法还引入了自己的符号。UML 已经开始被面向对象的编程 (OOP) 的从业人员广泛地使用。

4.3.2 数据建模产品中 E-R 模型的变化 (示例请见书 P206)

鸟足模型使用图中所给出的符号以指明关系的基数。离实体最近的符号显示了最大基数，而其他符号则表示最小基数。1 的标记表示有一个 (因此是强制性的) 实体实例，椭圆符号表示有零个 (因而是可选的)，鸟足标记表示有多个。

所用符号	含义
	一个-强制性的
	多个-强制性的
	一个-可选的
	多个-可选的

4.3.3 弱实体

E-R 模型中还定义了一种特殊类型的实体——弱实体 (Weak Entity)，这种实体只有依赖其他实体才可以存在于数据库中。如果一个实体不是弱实体，就称其为强实体 (Strong Entity)。

4.3.4 ID 依赖实体 (示例请见书 P208)

1. E-R 模型中存在着一种特殊的弱实体，成为标识符依赖实体 (ID-dependent Entity)。该实体的标识符中包含另一个实体的标识符。
2. 标识符依赖实体的标识符总是包含了实体标识的复合标识符，而该实体正是标识符依赖实体存在而依赖的实体。
3. 在 E-R 模型中，使用通过一个方框表示的实体来表示标识符依赖实体，使用实线来表示标识符依赖实体与其父辈之间的关系，这种关系称作标识关系 (identifying relationship)。而以虚线连接的关系用于表示两个强实体之间的关系，因为在该关系中没有任何标识符依赖实体，所以该关系又被称作非标识关系 (nonidentifying relationship)。
4. 标识符依赖实体只有在其父辈 (即标识符依赖实体所依赖的实体) 存在的情况下才会存在，因此，由标识符依赖实体到其父辈的最小基数总是 1。另一方面，父辈是否必须拥有一个标识符依赖实体，这取决于业务需要。
5. 只有在创建了父辈实体的实例之后才能添加标识符依赖实体的实例，同时，在删除父辈实体实例时，必须同时删除所有标识符依赖实体的实例。

4.3.5 非标识符依赖实体的弱实体 (示例请见书 P210)

1. 所有的标识符依赖实体都是弱实体，但并非所有的弱实体都是标识符依赖实体。
2. 弱实体必须在逻辑上依赖于其他实体。但是，不是所有最小基数为 1 且与其他实体相关的实体都是弱实体，而仅当它们在逻辑上依赖于其他实体时，才称为弱实体。即弱实体相对其所依赖实体的最小基数都为 1，但是最小基数为 1 的实体并不一定都是弱实体。
3. 在 E-R 模型中，使用圆角来表示非标识符依赖的实体，但是也使用了虚线来表示非标识符依赖实体及其父辈之间的未标识出的关系。

4.3.6 子类实体 (示例请见书 P212)

1. 子型 (subtype) 实体另一种名为父型 (supertype) 实体的特殊样例。父型实体的标识符也是子型实体的标识符。
2. 在 E-R 模型中，使用了一个圆及其下方的一道直线作为子型标志来指明父型和子型实体之间的关系。可以把这个关系看作是一个可选的 (圆) 1:1 (直线) 的关系。同样的，由于每个子型实体的标识符都依赖于其父型实体，所以还使用了一条实线来代表标识符依赖的子型实体。并且，没有使用在鸟足模型中所用的直线末端标志来作为连接直线。

3. 在某些情况下，父型实体的一个属性指明了对于给定的实例哪个子型实体比较合适。确定哪个子型实体合适的那个属性称作辨别器 (discriminator)。在 E-R 图中，辨别器与子型标志紧紧相邻。如果父型实体确实没有辨别器，那么必须通过编写应用程序代码来创建合适的子型实体。

4. 子型实体可以是排他的，也可以是包括在内的。如果子型实体是排他的 (exclusive)，那么父型实体最多只能和一个子型实体相关联，在图中用圆里面的“叉”来表示。如果是内含 (inclusive)，父型实体可以和一个或多个子型实体相关联，在图中，相应的圆圈里面没有“叉”。

5. 子型实体常常用于数据模型中来避免出现不合适的 NULL 数值。

6. 因为子型实体与父型实体是相同的实体，所以连接父型实体和子型实体的关系称作 IS-A 关系。也正因为这样，父型实体与其所有的子型实体的标识符必须相同，它们都表示了同一个实体的不同方面。

4.3.7 递归关系

递归关系 (recursive relationship) : 实体通过某种关系与其自身关联。对于二元关系而言，递归关系可能是 1:1、1:N、N:M。

第五章数据库设计

本章老师没有划重点，但是关键前提是要会设计关系数据模型，进行数据方案设计。以下是设计数据模型的要点：

1. 把数据模型转换为数据库设计方案的步骤：

- (1) 构建一个关系，将实体的所有属性作为列。
- (2) 使实体的标识符成为关系的主键。
- (3) 规范化关系。有时候要考虑对关系的一些部分进行非规范化，这种情况必须建立在充分了解而非一无所知的基础上。

2. 代理键

最理想的主键要求该键简短、数字的并且不会发生变化。代理键做主键时：首先，生成的编号没有内在的意义。其次，尽管代理键的数值在同一个表中可能不会相同，但是在两个数据库中可能并不唯一。

3. 弱实体的通过表的表示

为标识符依赖实体创建表时，必须确保父实体的标识符与标识符依赖弱实体的标识符同时出现在该表中。

为非标识符依赖实体创建表时，只要按照数据可设计方案的步骤就可完成，但必须记录它们存在的依赖性以及业务规则。

4. 关系的表示

一对一 (1:1) 强实体关系的表示，把一个表的建放入另一个表作为外键。

一对多 (1: N) 强实体关系的表示，把父表的建放入子表中作为外键。

多对多 (N: M) 强实体关系的表示，创建一个交集表，其中包含另外两个表的建。

5.子型实体关系的表示

父型实体和子型实体可以分别通过单独的表来表示。每个子型表的主键也是用于父型表的同一主键，每个子型表的主键还作为连接子型表回到父型表的外键。

6.递归关系的表示

递归关系的元素都来自同一实体，递归关系的元素都来自于同一类实体，其表示方式类似于非递归关系。对于一对一和一对多关系，子啊表示实体的关系中添加一个外键。对于多对多递归关系，创建一个表示该关系的交集表。。

第六章 数据库管理

本章简介：数据库管理（DBA）的目的是对数据库进行管理，以便最大程度地发挥数据库对其所属机构的作用。其中，缩略语 DBA 可以指这个部门也可以指其经理。

数据库管理的 3 个重要功能：并发控制、安全性、备份和恢复。

6.2 并发控制

1. 事务：即逻辑工作单元（LUW）

原子事务：以单元的形式执行的事务。

2. 并发事务：一个数据库同时处理两个事务，则称这两个事务为并发事务。

丢失更新问题：当一个用户读取记录时，另一个用户已经有了需要更新的副本，这种情况成为更新丢失问题。

4.并发问题：脏读取、不可重复读取、幻象读取

脏读取：一个事务读取了一个尚未提交到数据库但已经经过修改的记录。

不可重复读取：事务读取以前读取过的数据，并发现另一个事务对其进行了修改和删除。

幻象读取：事务重新读取数据但发现在读取数据后已有另一事务插入了新的数据行。

5. 资源锁定：在将要修改某些数据行或表时禁止多个应用程序同时获取这些行或表的副本。（排它锁定，共享锁定）

6. 串行事务：并发发出了两个或多个事务时，所得数据库的结果在逻辑上应该和事务以任意串行方式处理后所得到的结果保持一致，以这种方式处理并发事务的模式称为串行事务。

7. 死锁：每个用户都在等待另一个用户已经锁定的资源的状态称为死锁。

8. 乐观锁定：没有冲突时采用

读取数据，处理事务，执行更新，检查是否发生冲突，不冲突，事务可以完成，存在冲突重复处理直到不冲突。

悲观锁定：假设存在冲突，先用一个锁定处理事务，然后释放锁定。

事务隔离级别：读取未提交，读取已提交，可重复读取，可串行化

6.3 游标

1.概念：是一种指标针，指向从 SQL SELECT 语句得到的结果行集，它常常通过 SELECT 语句来定义。

2.四种类型：

只向前游标：应用程序只能向前面执行。

可滚动游标：1)静态游标 进行的修改可见，来自其他源的修改不可见。

2)动态游标 所有以行的顺序进行的插入、更新、删除、修改都可见，除非是脏读取，否则只有提交的修改才可见。

3)键集游标 打开该游标将保存每一行的主键的值。

6.4 数据库安全

目的是确保只有授权用户才可以在授权时间内进行授权的操作，通常分为两部分：身份验证和授权。

6.5 数据库备份与恢复

周期性地复制数据库被称为数据库备份

前滚：通过已保存的数据库文件恢复数据库，并且重新应用保存后的所有有效事务。

回滚：通过取消错误地执行或者没有全部完成的事务对数据库更改，来纠正错误的一种方法。

DBA 的其他职责

需要创建并管理一个控制数据库配置的进程；负责正确维护文档，从而记录数据库结构、并发控制、安全、备份、恢复、应用程序的使用，以及大量与数据库库的管理和使用相关的细节。

6.6 分布式数据库的处理

分布式数据库：在多台计算机上进行存储和处理的数据。

分布式数据库的类型：

可以通过分区来分布数据库，也就是将数据库分割为不同的片段并将这些片段存储在多台计算机中；也可以通过复制来分布数据库，也就是将数据库的副本存储在多台计算机中；或者通过分区和复制的联合。

6.7 对象—关系数据库

对象含有方法（即执行任务的计算机程序）和属性（改对象特有的数据项）。存储对象的属性值称为对象持久化。虽然关系数据库可以用于对象持久化，但做到这一点需要程序员进行大量的实际工作。对象—关系数据库可以同时处理关系数据。

第七章数据库处理应用程序和商业智能

学习目标：

理解并能创建 web 数据库处理

掌握可扩展标记语言的基本概念

掌握商业智能的基本概念

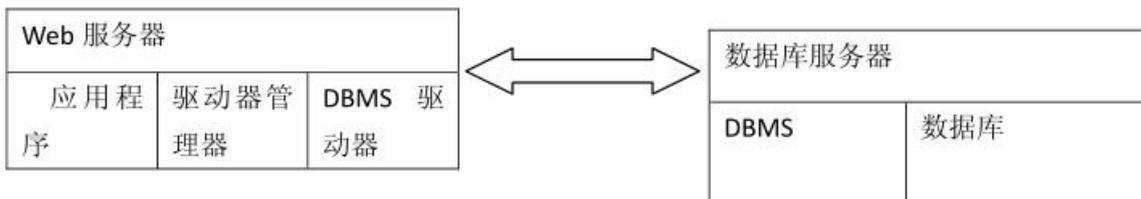
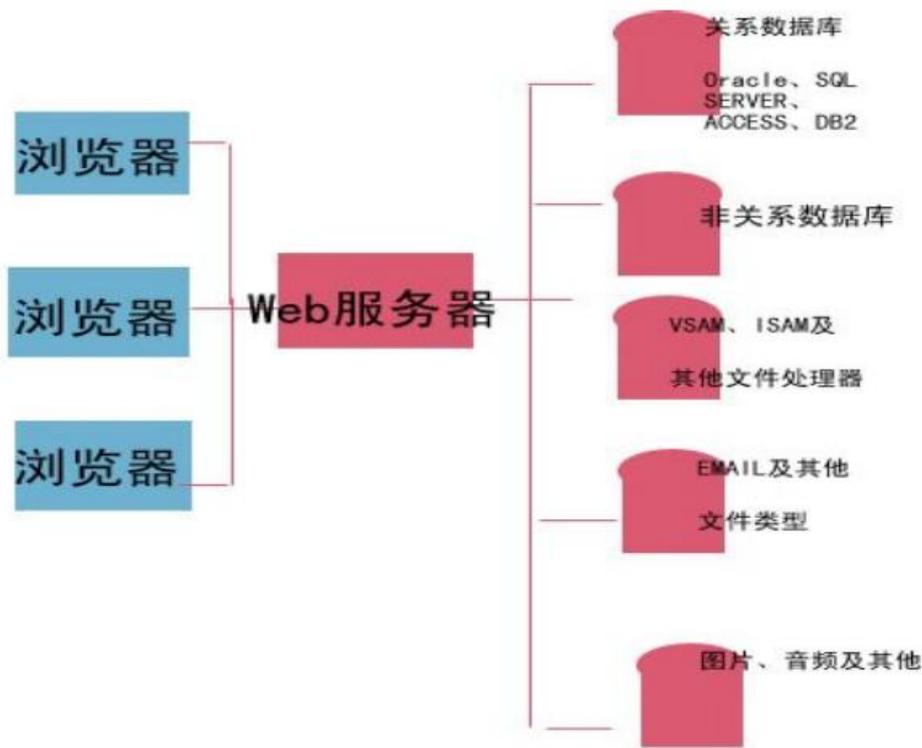
OLAP 和数据库挖掘的基本概念

第一节 数据库处理环境

触发器是一个存储在数据库里的程序并在如使用 INSERT、UPDATE 或 DELETE 语句的命令等特殊事件时由 DBMS 执行。

存储过程类似于计算机程序的子例程或函数，但它存储在执行数据库操作的数据库中。

第二节 Web 数据库处理



第三节 数据库处理和 xml

XML (Extensible Markup Language) 可扩展标记语言。类似于 HTML 功能更为强大

Xml 中可由开发人员自己创建标记集，可以声明标记之间的关系。

Xml 文件可以声明定义所使用的模式文件的名称和位置。

Xml 模式本身就是 xml 文件，这就意味着他们可以通过自己的模式进行验证。自动验证文件的能力意味着大量成本的节省。

Xml 使用数据库应用程序将文件中的数据放置到数据库中。

第四节 商业智能系统

商业智能 (business intelligence, BI) 系统是协助管理员和其他专家分析当前以前操作及预测可能发生事件的信息系统。

BI 分为两大类：报表和数据挖掘。报表系统 (reporting system) 对可操作的数据进行分类、挑选、分组等一些基本处理。数据挖掘 (data-mining application) 对数据进行精密分析。现在比较受欢迎的 BI 程序有 OLAP 和数据挖掘。

数据挖掘有集群分析 (cluster analysis)、回归分析、连续时间分析、因素分析等。