

chapter1 Introduction

1. 什么是数据、数据库、数据库管理系统、数据库系统 (2018)

数据: 描述事物的符号记录, 可以是数字、文字、图形、图像、音频、视频等;

数据库: 长期储存在计算机内、有组织的、可共享的大量数据的集合。数据库中的数据按一定的数据模型组织、描述和储存, 具有较小的冗余度、较高的数据独立性和易扩展性, 并可

为各种用户共享。

数据库管理系统: 位于用户和操作系统之间的一层数据库管理软件, 和操作系统一样是计算机的基础软件, 也是一个大型复杂的软件系统。

数据库系统: 由数据库、数据库管理系统 (及其应用开发工具)、应用程序和数据库管理员组成的存储、管理、处理和维护数据的系统。

DB 是存储数据的核心, DBMS 在 DB 上加了若干工具 (存储事物的管理、查询和事物处理), DBS 在 DBMS 上开发了应用程序。

*数据管理技术的三个阶段优缺点

1、人工管理阶段:

- (1) 数据不保存。因为当时计算机主要用于科学计算, 对于数据保存的需求尚不迫切。
- (2) 应用程序管理数据。数据由应用程序自己设计, 说明 (定义) 和管理, 没有相应的软件系统负责数据的管理工作, 应用程序中不仅要规定数据的逻辑结构, 而且要设计物理结构, 包括存储结构、存取方法和输入方法等。因此程序员负担很重。
- (3) 数据不共享。数据是面向程序的, 一组数据只能对应一个程序。
- (4) 数据不具有独立性。程序完全依赖于数据, 如果数据的逻辑结构或物理结构发生变化, 则必须对应用程序做出相应的修改。

2、文件系统阶段:

- (1) 数据可以长期保存在计算机外存上, 可以对数据进行反复处理, 并支持文件的查询、修改、插入和删除等操作。
- (2) **由专门的软件**即文件系统进行数据管理, 文件系统把数据组织成相互独立的数据文件, 同时也实现了记录内的结构性。
- (3) 数据共享性差, 冗余度大, 文件系统中文件依然是面向应用的, 不同的应用程序具有部分相同的数据时, 也必须要建立各自的文件。

(4) 数据独立性差，各文件之间是孤立的，不能反映现实世界之间的内在联系。

3、数据库系统阶段：

(1) 数据结构化。在描述数据时不仅要描述数据本身，还要描述数据之间的联系。数据库系统实现整体数据的结构化，是数据库的主要特征之一，也是数据库系统与文件系统的本质区别。

(2) 数据共享性高、冗余少且易扩充。数据不再针对某一个应用，而是面向整个系统，数据可被多个用户和多个应用共享使用，而且容易增加新的应用，所以数据的共享性高且易扩充。数据共享可大大减少数据冗余，节约存储空间。数据共享还能够避免数据之间的不相容性和不一致性。

(3) 数据独立性高。主要包括物理独立性以及逻辑独立性。物理独立性：用户的应用程序与数据库中数据的物理存储是相互独立的；逻辑独立性：用户的应用程序与数据库的逻辑结构是相互独立的

(4) 数据由 DBMS 统一管理和控制。

数据库为多个用户和应用程序所共享，对数据的存取往往是并发的，即多个用户可以同时存取数据库中的数据，甚至可以同时存放数据库中的同一个数据，为确保数据库数据的正确有效和数据库系统的有效运行，数据库管理系统提供以下 4 方面的数据控制功能

①数据的安全性保护：防止因不合法使用数据而造成数据的泄露和破坏，保证数据的安全和机密。

②数据的完整性检查：系统通过设置一些完整性规则，以确保数据的正确性、有效性和相容性。

③并发控制：多用户同时存取或修改数据库时，防止相互干扰而给用户提不正确的数据，并使数据库受到破坏。

④数据恢复：当数据库被破坏或数据不可靠时，系统有能力将数据库从错误状态恢复到最近某一时刻的正确状态。

2. 数据库相比于文件系统，有何优点（2019）

①整体数据结构化。数据不仅针对某一个应用，而是面向整个组织或企业；不仅数据内部是结构化的，而且整体是结构化的，数据之间是具有联系的。

②数据的共享性高。数据面向整个系统，可以被多个用户、多个应用程序共享使用。数据共享减少了由于数据冗余造成的不一致现象，弹性大易扩充。

- ③数据独立性高。物理独立性：用户的应用程序与数据库中数据的物理存储是相互独立的；
逻辑独立性：用户的应用程序与数据库的逻辑结构是相互独立的
- ④数据由数据库管理系统统一管理和控制。提供安全性保护、完整性检查、并发控制和数据库恢复等数据控制功能。
- ⑤数据灵活性高。
- ⑥利用元数据，描述数据变得容易。

数据库是长期存储在计算机内有组织、大量、共享的数据集合。它可以供各种用户分享，具有最小的冗余度和较高的数据独立性。数据库管理系统在数据库建立、运用和维护时对数据库进行统一控制，以保证数据的完整性和安全性，并在多用户同时使用数据库时进行并发控制，在发生故障后对数据库加以恢复。数据库系统的出现使信息系统以加工数据的程序为中心转向围绕共享的数据库为中心的新阶段。

3. 什么是数据模型，有哪些常用的数据模型

数据模型用于描述和操纵数据、数据之间的联系以及组织中数据的约束的概念的集合，是**数据库系统的核心和基础**。数据模型通常由数据结构（对象及对象间联系）、数据操作（增删改查）和数据的完整性约束条件（实体、参照、用户自定义）三部分组成。常用的数据模型有层次模型、网状模型和关系模型。

两类数据模型：概念模型、逻辑模型和物理模型（p28）

现实世界中客观对象的抽象过程：概念模型——逻辑模型——物理模型；

4. 什么是概念模型，如何表达概念模型（以 E-R 图为例），E-R 图如何转换为关系模式？

将现实世界通过概念模型抽象为信息世界，再将信息世界通过数据模型转换为机器世界。概念模型按照用户视角进行数据库设计，数据模型按照计算机视角进行数据库实现。

概念模型用于信息世界的建模，是现实世界到信息世界的第一层抽象，是数据库设计人员进行数据库设计的有力工具，也是数据库设计人员和用户之间进行交流的语言。概念模型具有较强的语义表达能力，能够方便、直接地表达应用中的各种语义知识；简单、清晰、易于用户理解。E-R 图是最常用的表达概念模型的工具，此外还有 ODL、IDEF1x 等。

概念模型的一种表示方法为：实体——联系方法，该方法用 E-R 图来描述现实世界的概念模型，E-R 方法也叫 E-R 模型

E-R 图向关系联系的转换，对于一对一把其中一方的主键纳入另一方做外键；对于一对多把单方的主键纳入多方做外键；对于多对多新成立一个联系集，把双方的主键纳入联系集做外键，如果联系本身有属性也一并纳入联系集。

*数据模型的组成要素

数据模型通常由数据结构、数据操作和完整性约束条件三个部分组成。

①数据结构：描述数据库的组成对象以及对象之间的联系，通常按照其数据结构的类型来命名数据模型，数据结构是所描述的对象类型的集合，是对系统静态特性的描述。

②数据操作：是指对数据库中的各种对象（型）和实例（值）允许执行的操作的集合，包括操作及有关的操作规则。数据库主要有查询和更新（包括插入、删除、修改）两大类操作，数据操作是对系统动态特性的描述。数据操作都是关系，**操作的对象和结果都是关系。**

③数据的完整性约束条件：是一组完整性规则，是给定的数据模型中数据及其联系所具有的制约和依存规则，用以限定符合数据模型的数据库状态以及状态的变化，以保证数据的正确、有效和相容。

主要数据模型：层次模型、网状模型（两者称为格式化模型）、关系模型、面向对象数据模型、对象关系数据模型、半结构化数据模型。

*层次模型、网状模型、关系模型优缺点比较

层次模型优缺点（p35）

网状模型优缺点（p37）

关系模型优缺点（p40）

5. 数据库的体系结构怎么划分的（2018、2019）

三级两映射，目的是达到数据独立性。（作用：P43）

三级模式：外模式是数据库用户能够看见和使用的局部数据的逻辑结构和特征的描述，是数据库用户的数据视图，是与某一应用有关的数据的逻辑表示；模式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；内模式是数据物理结构和存储方式的描述，是数据在数据库内部的组织方式。

二级映射：外模式-模式映射体现数据的逻辑独立性，实现模式改变而外模式不变；模式-内模式映射体现数据的物理独立性，实现内模式改变而模式不变。

chapter2 Relational database

1. 关系数据模型的三要素是什么

关系数据结构、关系操作、关系模型的完整性约束。

2. 关系数据模型的数据结构

关系（关系模型中数据的逻辑结构是一张扁平的二维表）——关系模型实例化

关系的描述称为关系模式，是型（关系名（属性 1，属性 2...）），关系是值；

关系模式是静态的、而关系是动态的。

关系数据库的型称为关系数据库模式，而值是这些关系模式在某一时刻对应的关系的集合。

关系是数据库中存放数据的基本单位。

*关系的三种类型

①基本关系，通常又称为基本表或基表，是实际存在的表，它是实际存储数据的逻辑表示。

②查询表：是查询结果对应的表。

③视图表：由基本表或其他视图导出的表，是虚表，不对应实际存储的数据。

但这三种类型都是二维表。

*关系操作

关系操作主要包括查询和更新（插入、删除、修改）两大操作。

查询操作包括选择、投影、连接、除、并、差、交、笛卡尔积等。

选择、投影、并、差、笛卡尔积为 5 种基本操作。

特点：集合操作方式，操作对象和结果都是集合，一次一集合。

非关系操作侧为一次一记录。

3. 关系数据语言的分类

①关系代数语言

②关系演算语言（包括元组关系演算语言、域关系演算语言）

③具有关系代数和关系演算的双重特点的语言（SQL）

4. 关系数据模型的数据约束

- ①实体完整性约束：主键不能为空。
- ②参照完整性约束：通过外键表示两个关系之间的引用，外键为空或与主键相同。
- ③用户自定义完整性约束：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。基于现实世界中的真实需求，如性别只能输入男或女。关系模型提供定义和检验这类完整性的机制，以使用统一系统的方法处理它们。

5. 关系代数

运算对象是关系，运算结果也是关系（集合运算符、专门的关系运算符）

- ① 传统的集合运算：并 \cup 、差 $-$ 、交 \cap 、笛卡儿积 \times ——涉及行
两个关系要具有相同的目、且相应的属性要取自同一个域
- ② 专门的关系运算：选择 σ 、投影 Π 、连接 \bowtie 、除 \div ——涉及行和列
- ③五种基本的运算：并、差、笛卡儿积、选择、投影，其它三种运算均可以用这五种基本运算来表达。

chapter3 SQL

1. 什么是 SQL??

任何一种数据库都支持的一种结构化查询语言，分为4种。数据定义语言 DDL（搭框架），提供定义、删除和修改关系模式和约束的命令，关键词 `create`；数据查询语言 DQL，是整个 SQL 语言的核心，关键词 `select`；数据操纵语言 DML（数据增减），包含基于关系代数和元组关系演算的查询语言；数据控制语言 DCL（安全与权限），包含用于指定关系和视图的访问权限的命令，权限分配如 `grant`、`revoke` 和 `deny`。

2. 数据定义语言（Create）

3. 数据操纵语言（查询-query, 更新-insert, update, delete）

从单表或多表查询 `SELECT`

对多表查询的连接有自然连接 `Inner join`，左外连接 `Left outer join`，右外连接 `Right outer join`，全外连接 `Full outer join`，`cross join`

嵌套查询分相关和非相关，关键词有 `IN`，`Comparison`，`ANY/ALL`，`EXISTS`

对于集合的运算有并 `UNION`，交 `INTERSECT`，差 `EXCEPT`

把查询结果物理地存储为一个表 `INTO`

对列或者表重命名，对子查询建立临时表 AS

4. 什么是视图（Views），为什么要用视图

视图是从一个或几个基本表或其他视图导出的表定义。它本身不独立存储在数据库中，即数据库中只存放视图的定义而不存放视图对应的数据，这些数据仍存放在导出视图的基本表中。因此视图是一个虚表。视图在概念上与基本表等同，用户可以在视图上再定义视图。（视图可以更新，可以嵌套，但不可以在视图上建立基本表）

1. 简化用户的操作
2. 让用户以不同的视角看同样的数据
3. 数据逻辑独立性
4. 数据的安全性

5. 数据控制语言（Authorization，权限分配与收回）

对象：大到数据库，小到属性

对存储过程来说是执行权限，对数据库来说是建库和删库的权限，都可以作为分配的单元分配给用户。

Object	Object Types	Privilege
Attributes	Table	SELECT,INSERT,UPDATE,DELETE,ALL PRIVILEGES
Views	Table	SELECT,INSERT,UPDATE,DELETE,ALL PRIVILEGES
Table	Table	SELECT,INSERT,UPDATE,DELETE,ALL PRIVILEGES
Database	Database	CREATE/DROP
Procedure	Procedure	Execute

6. SQL 延伸知识

1. 存储过程
2. 触发器

Chapter 4 Data Integrity

1. 什么是数据完整性

是数据模型组成部分中施加在数据库数据之上的语义约束条件，是完整性控制的核心，

DBMS 应提供定义数据库完整性约束条件，并把它们作为模式的一部分存入数据库中。

Different from Security

Integrity: Protecting against accidental introduction of inconsistency.

完整性: 防止意外的错误数据进入数据

Security: Protecting database from unauthorized access and malicious destruction or alteration.

安全性: 防止非授权用户的恶意破坏

2. 有哪些数据完整性约束 Six Types of Data Integrity Constraints

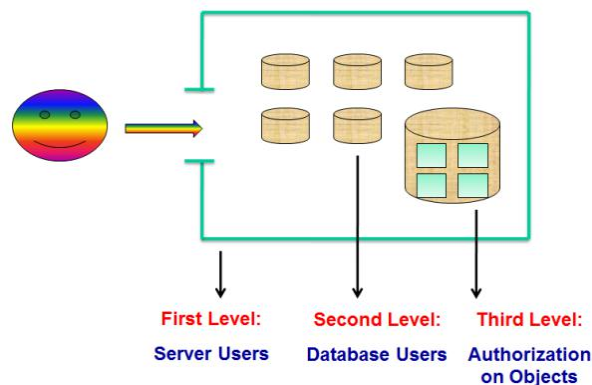
	Column	Tuple	Relation
Static	列类型	元组值应满足的条件	实体完整性约束
	列格式		参照完整性约束
	列值域		函数依赖约束
	列空值		统计约束
Dynamic	改变列定义或列值	元组新旧值之间应满足的约束条件	关系新旧状态间应满足的约束条件

3. SQL SERVER 中的数据完整性 Data Integrity in Sql Server

1. 用 DDL 定义约束
2. 触发器

Chapter 5 Database Security

1. 数据库的安全机制 (2019)



院子（服务器）中有若干房子（数据库），每个房子中又可以放若干个柜子（表），很多东西是放在柜子里面的（数据），拿到柜子里的东西要过三关：先进院子（成为服务器的登陆用户），再进房子（成为数据库的用户），具有柜子的钥匙（数据的权限）

1. Windows 集成登陆

2. 基于 SQL Server 的混合登陆

2. 数据库的用户管理机制 User Management

3. 数据库的角色及管理 Role Management

角色：权限的合集，方便对权限的管理

4. 权限分配 Authorization

Chapter 6 Relational Database Theory

1. 关系数据库设计流程

消除不合适的数据依赖，使各关系模式达到某种程度的“分离”

采用“一事一地”的模式设计原则（一个关系只描述一个概念或一个联系）

让一个关系描述一个概念、一个实体或者实体间的一种联系。若多于一个概念就把它“分离”出去

所谓规范化实质上是概念单一化的过程

不能说规范化程度越高的关系模式就越好（规范化程度高减少了数据冗余，但查询和使用时效率低）

在设计数据库模式结构时，必须对现实世界的实际情况和用户应用需求作进一步分析，确定一个合适的、能够反映现实世界的模式

上面的规范化步骤可以在其中任何一步终止

2. 函数依赖

Armstrong 公理：找属性集闭包的工具（无法一眼看出哪个属性做候选键）

3. 第一范式 The First Normal Form(1NF)

1NF：一个表格被认为是 1NF，当表中每个单元恰好包含一个值时。

1NF 解决了关系的不可再分性（原子性）。

存在问题：数据冗余、更新异常、插入异常、删除异常

4. 第二范式 The Second Normal Form(2NF)

2NF：一个表格被认为是 2NF，当它是 1NF 且该行中的每个属性是函数依赖于整个键，而不是部分的键。2NF 消除了关系的部分依赖。

5. 第三范式 The Third Normal Form(3NF)

3NF：一个表格被认为是 3NF，当它是 2NF 且每个非键属性只依赖于主键。3NF 解决了关系的传递依赖。（消除非主属性之间的依赖关系）

6. 其他范式 The Other Normal Form(BCNF)

BCNF：一个表格被认为是 BCNF，当且仅当每一个决定的因子都是候选键。

【如果 $X \rightarrow Y$ ，则 X 为决定因子】

（解决一个关系模式中有多个候选键时考虑，当关系模式中只有一个候选键则不需要考虑 BC 范式）

模式分解应满足：无损连接性（分解后的关系通过自然连接可以恢复成原来的关系，既不多出信息、又不丢失信息）、保持函数依赖（模式分解不能破坏原来的语义）。

Chapter 7 Database Design

Phase 1: Requirements collection and analysis	数据字典、全系统中数据项、数据流、数据存储描述
Phase 2: Conceptual database design	概念模型（E-R图），数据字典
Phase 3: Choice of DBMS	
Phase 4: Data model mapping (logical design)	某种数据模型
Phase 5: Physical design	存储安排，方法选择，存储路径建立
Phase 6: System implementation and tuning	编写模式，装入数据，数据试运行

六个步骤：需求分析、概念结构设计、逻辑结构设计、物理结构设计、数据库事实、数据库运行和维护

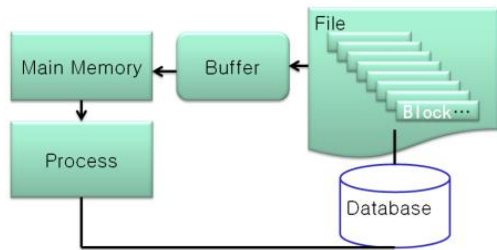
Chapter 8 Database Programming

游标：一种能从包括多条数据记录的结果集中每次提取一条记录的机制，把作为面向集合的数据库管理系统和面向行的程序设计两者联系起来，使两个数据处理方式能够进行沟通。

开放数据库连接（ODBC）：微软公司开发的有关数据库的一组规范，并且提供一组访问数据库的标准 API。（是一种接口，达到编一次、到处使用的功能）

Chapter 9 Data Storage and Querying

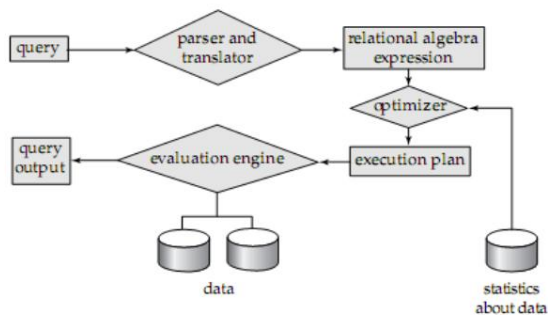
1. 数据库存储与数据结构



2. 数据库索引

1. 通过创建唯一性索引，可以保证数据库表中每一行数据的唯一性。（允许为空）
2. 可以大大加快数据的检索速度，这也是创建索引的主要原因。
3. 可以加速表和表之间的连接，特别是在实现数据的参考完整性方面特别有意义。
4. 在使用分组和排序子句进行数据库检索时，同样可以显著减少查询中分组和排序的时间。
5. 通过使用索引，可以在查询的过程中，使用优化隐藏器，提高系统的性能。

3. 查询过程



4. 查询优化规则与技巧

Rules:

- 1) 选择运算最先做，减少中间结果
- 2) 连接之前做一些预处理，如排序或建索引
- 3) 投影和选择运算尽可能同时完成
- 4) 投影与它之前或之后的操作合并进行
- 5) 一些选择运算和笛卡尔积合并成为连接运算
- 6) 使用公共表达式，如 INTO

Skills:

- 1) 充分使用索引
- 2) 避免或简化排序运算
- 3) 避免对表的顺序访问
- 4) 避免使用相关子查询
- 5) 避免使用 like 关键词

6) 使用临时表加快查询速度

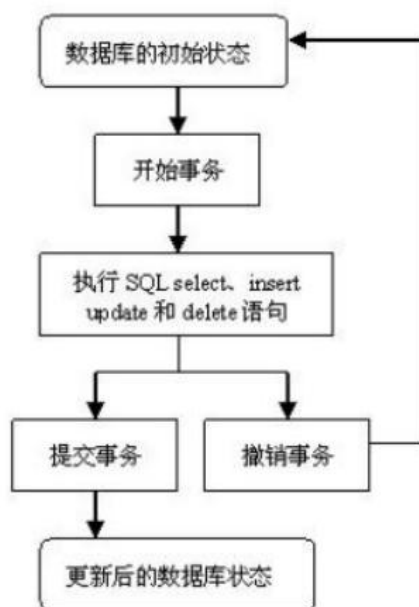
7) 少用如下运算符 'Is Null' or 'Is Not Null', Or, <> (Replaced by '>' and '<'), In, Order by, Like, Distinct

Chapter 10 Concurrency Control

1. 什么是事务，有什么特性 (Transactions)

构成单个逻辑工作单元的操作集合称为事务。事务是恢复和并发控制的基本单位。

ACID: 原子性、一致性、隔离性、持久性



多个事务并发运行时的问题：丢失修改，读脏数据，不可重复读

2. 什么是锁 (Locks)，锁的类型？什么是活锁和死锁 (Livelock / Deadlock)，如何预防？

锁：加在访问对象上的表示其访问限制的标志，如共享锁 S、排它锁 X。

活锁：某个事务永远处于等待封锁的状态。(预防：先来先服务)

死锁：多个事务同时处于等待状态，每个事务都在等待其它事务释放锁使其能够继续执行，从而出现多个事务互相等待的僵局，每个事务永远不能结束。(预防：一次封锁法、顺序封锁法、诊断解除法)

3. 什么是锁协议，哪三级？解决哪些并发异常问题？ (locking protocol)

一级封锁协议：只规定写加 X 锁，对读无要求

二级封锁协议：写加 X 锁，读加 S 锁，读完即可释放

三级封锁协议：写加 X 锁，读加 S 锁，直到事务结束后才释放

	X锁		S锁		一致性保证		
	操作结束 释放	事务结束 释放	操作结束 释放	事务结束 释放	不丢失更新	不读未提交 数据	可重复读
一级封锁协议		✓			✓		
二级封锁协议		✓	✓		✓	✓	
三级封锁协议		✓		✓	✓	✓	✓

Chapter 11 Backup/Restore

1. 数据库的备份恢复系统有什么意义 (Recovery system)

(最后一道安全屏障)

- ①确保事务的原子性和持久性属性得到保留。
 - ②将数据库恢复到故障前的一致状态。
 - ③恢复方案还必须提供高可用性;也就是说,它必须最小化崩溃后数据库不可用的时间。
- ①Ensure that the atomicity and durability properties of transactions are preserved.
 - ②Restore the database to the consistent state that existed before the failure.
 - ③The recovery scheme must also provide high availability; that is, it must minimize the time for which the database is not usable after a crash.

2. 数据库的故障类型 (Failure Classification)

- (1) Transaction failure (事务故障)
- (2) System crash (系统故障)
- (3) Disk failure (介质故障)
- (4) Others. (病毒、自然灾害)

3. 数据库的备份恢复类型 (Backup/Restore)

		转储状态	
		动态转储	静态转储
转储 方式	海量转储	动态海量转储	静态海量转储
	增量转储	动态增量转储	静态增量转储

1. 简单恢复: 允许将数据库恢复到最新的备份, 即恢复到上次备份的即时点, 而无法将数据库恢复到故障点或特定的即时点, 简单恢复只能做数据库备份, 而无法做日志备份;
2. 完全恢复: 允许将数据库恢复到故障点状态, 即完全恢复使用数据库备份和事务日志备份对介质故障的完全防范。

4. SQL SERVER 中的备份与恢复操作 (Restore in SQL Server)

Chapter 13 Database New Technology

数据库系统给我们提供了那些安全保护？

约束+视图+触发器/存储过程+规范化控制+身份鉴别和权限控制+并发控制+备份恢复

如何设计一个数据库？

用户需求调查得到概念模型，利用范式理论规范设计结果（数据模型），选择一种关系数据库产品实现数据库，数据库编程触发器、存储过程、嵌入式，输入模拟数据进行测试，考虑物理上的存放优化（建索引、选择存储设备），保证安全性的情况下向用户提供服务。