

第1章 數據庫系統概述

中文名稱	英文簡寫
數據	Data
數據庫	DB
數據庫管理系統	DBMS
數據庫系統	DBS

- 1. 數據是描述事物的 符號記錄 ，是指用 物理符號 記錄下來的，可以鑒別的 信息
- 2. 數據庫是指 長期儲存 在計算機中的有組織，可享的 數據集合 ，且數據庫中的數據按一定的 數據模型 組織描述和存儲，具有較小的 冗餘度 ，較高的 獨立性 ，系於易於擴展並可以被多個用戶 共享 。
- 3. 數據庫管理系統是專門用於建立和管理數據庫的一套 軟件 ，介於 應用程序 和 操作系統 之間。
- 4. 數據庫系統是指計算機中引入 數據庫 技術之後的系統。通常，一個完整的數據庫系統包括 數據庫 、數據庫管理系統 及相關實用工具、 應用程序 、 數庫管理員 和 用戶 。
- 5. 數據管理三個階段分別是， 人工管理階段 ， 文件管理階段 ， 數據庫系統階段 。
- 6. 三個階段的特徵

中文名稱	年份	特徵
人工管理階段	20世紀50年代之前	數據不保存
		應用程序管理數據
		數據面向應用
文件管理階段	20世紀50年代到60年代中期	將邏輯結構和物理結構分離
		不能實現數據共享
		只能實現文件共享
		大量的數據冗餘

中文名稱	年份	特徵
數據庫系統階段	20世紀60年代后期	數據集成
		數據共享性高
		數據冗餘小
		數據一致性
		數據獨立性高
		實施統一管理與控制
		減少應用程開發與維護工作量

1. 數據庫外部體系結構通常表現為 集中式結構 、 分布式結構 和 並行結構 。
2. 數據庫三級模式結構是指數據庫系統由 模式 、 外模式 、 內模式 三級構成的。
3. 模式又稱 概念模式 ， 邏輯模式 。
4. 外模式又稱 子模式 ， 用戶模式 。
5. 內模式又稱 存儲模式 。
6. 模式之間映像分是 外模式/模式映像 ， 模式/內模式映像 。
7. C/S全稱為 Client/Server ， 中文是 客戶/服務器 。 應用程序稱為客戶端 、 前台、表示層 。 數據庫管理系統又稱為 服務器 、 數據層 、 后台
8. B/S全稱為 Browser/Server ， 中文是 瀏覽器/服務器 ， 瀏覽器在 表示層 ， 又稱 客戶端 、 前台 ， 服務器在 處理層 ， 又稱 數據層 、 中間層 、 后台 。
9. 模型 是現實世界特徵的模擬和抽象表達 。
10. 數據模型組成要素

數據結構，描述是系統靜態特徵

數據操作，描述是系統動態特徵

數據約束，描述數據結構中數據間語法和語義關聯

1. 數據模型分類

概念層數據模型

邏輯層數據模型

物理層數據模型

1. 數據模型 是模型化數據和信息的工具，也是數據庫 核心和基礎 。
2. 寫出數據模型分層

現實世界->概念層數據模型->邏輯層數據模型->物理層數據模型

1. 數據具有 靜態 和 動態 兩種特徵 。
2. 概念層 是數據抽象級別的 最高層 。 概念層數模型也稱數據的 概念模型 或 信息模型 ， 這類模型主要用於 數據庫設計 階段 。

3. 概念數據模型，信息世界涉及的基本概念

實體

屬性

碼或鍵

域

實體型

實體集

聯系

1. 數據模型中有 型(Type) 和 值(Value) 兩個概念
2. 在設計數據庫時，通常用 概念模型 來抽象、表示現實世界的各種事物及其聯系。
3. 邏輯層 數據抽象級別的 中間層 。邏輯層數據模型，也稱為數據的 邏輯模型 。任何DBMS都是基於某種邏輯 數據模型 。
4. 邏輯數據模型發展階段
 1. 層次模型，最早使用的一種數據模型，有且僅有一個結點沒有 父結點 ，稱為 根結點 。其他結點有且僅有一個 父結點 。
 2. 網狀模型，以 網狀 結構表示實體與實體間的聯系。允許結點有多於一個父點點可以有一個以上的結點沒有父結點。
 3. 關係模型，用 二維表 結構來表示實體間的聯系，優點：建立在嚴格 數學 的基礎上，概念單一，存取路徑對用戶透明，有更高的數據獨立性，更好的安全保密性。
 4. 面向對象模型，既是 概念模型 又是 邏輯模型 。表達能力豐富、對象可復用、維護方便。
1. 物理層數據模型，也稱為數據 物理模型 ，描述數據在存儲介質上的組織結構，是邏輯模型的物理實現。是數據庫 最底層 的抽象，設計目標是提高數據庫性能和有效利用存儲空間。
2. 簡述數據、數據庫、數據庫管理系統、數據庫系統的概念

數據是描述事物的符號記錄，是指用物理符號記錄下來的，可以鑒別的信息

數據庫是指長期儲存在計算機中的有組織，可享的數據集合，且數據庫中的數據按一定的數據模型組織描述和存儲，具有較小的冗餘度，較高的獨立性，系於易於擴展並可以被多個用戶共享。

數據庫管理系統是專門用於建立和管理數據庫的一套軟件，介於應用程序和操作系統之間。

數據庫系統是指計算機中引入數據庫技術之後的系統。通常，一個完整的數據庫系統包括數據庫、數據庫管理系統及相關實用工具、應用程序、數庫管理員和用戶。

1. 簡述在數據管理技術中、與人工管理、文件系統相比，數據庫系統的優點

數據集成

數據共享性高

數據冗餘小

數據一致性

數據獨立性高

實施統一管理與控制

減少應用程序開發與維護工作量

1. 簡述數據庫系統的三級模式和兩層映象的含義

數據庫系統在體系結構都具有相同特徵，都採取三級模式，即模式，內模式和外模式。

模式：它是數據庫中全體數據的邏輯結構和特徵的描述，是所有用戶的公共數據視圖。

外模式：它是數據庫用戶能夠看見和使用的局部數據的邏輯結構和特徵的描述，是與某一應用有關的數據邏輯表示。

內模式：它對數據庫中數據物理結構和存儲方式的描述，是數據庫內部的表示形式。

為了有效支撐數據三級模式之間聯系及轉換，DBMS提供三級模式之間的兩層映像來實現，即外模式/模式映像與模式/內模式映像。

外模式/模式映像：定義了各個外模式與概念模式之間的映像關係，並保證了數據與程序的邏輯獨立性。

模式/內模式映像：定義了數據全局邏輯結構與物理存儲之間的對應關係，並保證了數據與程序的物理獨立性。

1. 簡述外模式/模式映像及其如何保證數據的邏輯獨立性

外模式/模式映像定義了各個外模式與概念模式之間的映像關係，這些映像定通常在各自的外模式中加以描述。數據庫系統的模式如若發生改變，數據庫管理員通常會對各個外模式/模式的映做出相應的改變，以使那些對用戶可見的外模式保持不變，從而應用程序的編程人不必去修改那些依數據的外模式所編寫的應用程序，如此實現了外模式不受概念模式變化的影響，並保證了數據與程序的邏輯獨立性。

1. 簡述模式/內模式映像及其如何保證數據與程序的物理獨立性

模式/內模式映像定義了數據全局邏輯結構與物理存儲之間的對應關係，這種映像定義通常是在模式中加以描述。由於數據庫中只有一個模式，且也只有一個內模式，所以模式/內模式映是唯一的。數據庫系統的物理存儲如若發生改變，數據管理員會對模式/內模式映像做出相應調整，以使數據庫系統的模式保持不變，從而也不必去修改應用程序，如此實現了概念模式不受內模式變化的影響，並保證了數據與程序的物理獨立。

1. 簡述關係模型與網狀模型、層次模型的區別

2. 簡述關係模型的優點

1. 關係模型是建立在嚴格的數學的基礎之上的
2. 關係模型的概念簡單，統一用關係來表示實體以及實體之間的聯系，關係模型的數據結構簡單、清晰，用戶易懂、易用。
3. 關係模型的存取路徑對用戶透明，從而具有更高的數據獨立性，更好的安全保密性，也簡化了程序員的工作和數據庫開發建立的工作。

1. 簡述概念模型、邏輯模型、物理模型之間的關係。

這三個不同的數據模型之間既相互獨立，又存在着關聯。從現實世界到概念模型的轉換是由數據庫設計人員完成的。從概念模型到邏輯模型的轉換可以由數據庫人員完成，也可以用數據庫設計工具協助設計人員完成。

從邏輯模型到物理模型的轉換主要是由數據庫管理系統完成的。

第2章 關係數據庫

1. 關係數據庫發展歷史

年份	事件
1962年	提出以" 信息代數 "，就是最早數學方法用於數據處理
1970年	提出關係模型
20世紀70年代末	關係方法理論和軟件系統的研制才取得重大突破，其中代表性是 IBM 370 的 System R
1981年	IBM370 system r全部特徵的數據庫管理系統 SQL/DS 問世
商用數據庫管理系統主流變為關係數據庫	
20世紀80年代	例如：Oracle, Sybase, Informix, IBMDB2, MS SQL Server
	個人電腦的資料庫FoxPro, Access, MySQL

1. 關係數據模型的組成要素有 關係數據結構 、 關係操作集合 、 關係完整性約束 。

學號	姓名	性別	出生日期	籍貫	民族	班號	身份證證
123	陳小明	男	2000/01/01	廣東	漢族	A456	1234567

表，也稱 關係 。

關係，在關係邏輯上對應一張 二維表 。有三種類型， 基本關係 、 查詢表 、 視圖表 。

列(Column)，也稱 字段(Field) 或 屬性(Attribute) 。

屬性(Attribute)，表中的一 列 即為一個屬性，給每一個屬性起一個名稱即 屬性名 。列的值為 屬性值 。

行(Row)，也稱為 元組(Tuple) 或 記錄(Record) 。

元組，表的一 行 即為一個元組。

分量，元組中的一個 屬性值 ，稱為分量。

碼&鍵(Key)，在一個關係中，存在這樣的屬性(屬性組)，使得在該關係的任何一個關係狀態中的兩個元組，在該屬性(屬性組)上值的組合都不相同，這些 屬性(屬性組)值 都能用來 唯一標識 該關係的 元組 。(可以理解成unique key)例如，(學號, 班號, 姓名, 性別, 身份證)，(學號)

超碼&超鍵，如果在關係的一個碼中移去某個屬性，它仍然是這個關係的 碼 。例如，(學號, 班號, 姓名, 性別, 身份證)，移去身份證也是碼

候選碼或候選鍵(Candidate Key)，如果在一個關係的一個 碼或鍵 ，不能從中移去任何一個 屬性 ，否則它就不是這個關係的碼或鍵，則稱這樣的碼或鍵為關係的候選碼或候選鍵。也是最小的 超碼和超鍵 。例如，(學

號)，(身份證)

主碼或主鍵(Primary Key)，在一個關係的若干個 候選碼或候選鍵 中指定一個用來唯一標識的元組，則稱這個被指定的 候選碼或候選鍵 為該關係的主碼或主鍵。例如，學號是主碼

全碼或全鍵(All Key)，一個關係模式所有 屬性 集合是這個關係的 主碼或主鍵 。

主屬性和非主屬性(Primary Attribute, Non-primary Attribute)，關係中包含在任何一個 候選碼 中的 屬性 為主屬性或碼屬性，不包含在任何一個候選碼的屬性稱為非主屬性或非屬性。例如，(學號)，(身份證)，學號和身份證是主屬性，其它是非主屬性。

碼=超碼+候選碼

主碼 in 候選碼

外碼與外鍵(Foreign Key)，當關係中的某個屬性不是這個關係的 主碼 或 候選碼 ，而是另一個關係的 主碼 ，稱為該屬性是這個關係的外碼或外鍵。例如，班號

參照關係(Referencing Relation)和被參照關係(Referenced Relation)，參照關係也稱為 從關係 ，被參照關係稱為 主關係 。被參照關係與參照關係是通過 外碼 相關系的。(參照關係指當下的表，被參照關係是另外一個表)

域(Domain)，域表示屬性的 取值範圍 。例如(性別)，只有(男, 女)可選。

數據類型，表中每個列都有相應的 數據類型 ，如姓名的數據類型是字符。

關係模式(Relation Schema)，關係模式是 靜態 、穩定，關係是 動態 。由數據庫的型和值之份，關係模式是 型 ，關係是 值 。

關係數據庫，使用 關係模型 作為數據的 邏輯模型 的數據庫。以關係模型作為數據的邏輯模型，並採用關係作為數據組織方式的一類數據庫，其數據庫操作建立在關係代數的基礎上。

1. 關係操作的特點是 集合操作方式 ，即操作的 對象 和 結果 都是 集合 。這種操作方式也稱 一次一集合 (set-at-a-time)的方式
2. 關係模型中常用的關係操作包括 查詢操作 和 插入 、 刪除 、 修改 。
3. 關係數據語言可以分為三類， 關係代數語言 、 關係演算法 、 兼具兩者雙重特點的語言(結構化查詢語言SQL)。關係演算法又可分為 元組關係演算 和 域關係演算 。它們的共同特點：語言具有完備的表達能力，是非過程化的集合操作語言，功能強，能夠獨立使用也可以嵌入高級語言中使用。
4. 任何一操作都包括 操作對象 、 操作符 、 操作結果 。
5. 關係的完成整約束，有 實體完整性約束 ， 參照完整性約束 ， 用戶定義完整性約束 、 關係模型完整性約束的檢驗 ，當中關係模型完整性約束的檢驗包括 執行插入操作 、 執行刪除操作 、 執行更新操作 。
6. 數據庫的數據完整是數據庫中的數據的 正確性 、 相容性 、 一致性 。
7. 不好的關係模式有 數據冗餘 、 更新異常 、 插入異常 、 刪除異常 。
8. 一個低一級范式的關係模式通過 模式分解 可以轉換為若干個高一級范式的關係模式的集合，這種過程就叫 规范化 。
9. 關係规范化理論 主要應用於數據庫設計中的 概念設計 階段。
10. 一個關係R，當中屬性X的每一個值，與R中屬性Y都有唯一值對應，稱X函數決定Y或Y函數依賴於X，X 稱為 決定因素 ，寫 $X \rightarrow Y$ 。
11. 函數依賴分別有 完全函數依賴 ， 部分函數依賴 ， 傳遞函數依賴 。
12. 一個關係R，當中屬性集有X,Y，其中 $X \rightarrow Y$ ，X的真子集X'，都滿足 $X' \not\rightarrow Y$ ，則稱Y 完全函數 依賴於X。(X為主碼)

13. 一個關係R，當中屬性集有X,Y，其中 $X \rightarrow Y$ ，其中X的真子集X'，都 $X' \rightarrow Y$ ，則稱Y 部分函數 依賴於 X。(X為超碼)
14. 一個關係R，當中屬性集有X,Y,Z，其中 $X \rightarrow Y$ ， $Y \twoheadrightarrow X$ ， $Y \rightarrow Z$ ，則 $X \rightarrow Z$ ，則稱Z 傳遞函數 依賴於X。
15. 簡述完全函數依賴，部分函數依賴，傳遞函數依賴

一個關係R，當中屬性集有X,Y，其中 $X \rightarrow Y$ ，X的所有的真子集X'，都滿足 $X' \twoheadrightarrow Y$ ，則稱Y 完全函數 依賴於 X。(X為主碼)

一個關係R，當中屬性集有X,Y，其中 $X \rightarrow Y$ ，其中X的真子集X'，都 $X' \rightarrow Y$ ，則稱Y 部分函數 依賴於X。(X為超碼)

一個關係R，當中屬性集有X,Y,Z，其中 $X \rightarrow Y$ ， $Y \twoheadrightarrow X$ ， $Y \rightarrow Z$ ，則 $X \rightarrow Z$ ，則稱Z 傳遞函數 依賴於X。

1. 第一范式容易有 冗餘高、插入異常 和 刪除異常。
2. 第二范式有 插入異常 和 刪除異常。
3. 第三范式一般可消除大部分有 插入異常 和 刪除異常。數據冗餘 也得到控制。
4. 簡述關係數據庫的基本特徵。

關係數據庫的基本特徵是使用關係數據模型組織中數據，這種思想源於數學。

1. 關係數據庫對關係的限制/要求
 1. 每一個屬性都是不可分解的
 2. 每一個關係僅僅有一種關係模式
 3. 每一個關係模式中的屬性必須命名，在同一個關係模式中，屬性名必須是不同的
 4. 同一個關係中不允許出現候選碼或候選鍵值完全相同的元組
 5. 在關係中元組的順序是無關緊要的，可以任意交換。
 6. 在關係中屬性的順序是無關緊要的，可以任意交換的。

1. 簡述甚麼是實體完整性約束

實體完整性約束是關係的主屬性，即主碼的組成不能為空，也就是關係的主屬性不能是值NULL。關係對應於現實世界中的實體集，而現實世界中的實體是區分的，即說明每個實體具有唯一性標識。

在關係模型中，是使用主碼作為唯一標識的，若假設主碼取空值，則說這個實體不可標識，即不可區分。

1. 簡述甚麼是參照完整性約束。

現實世界中的實體之間往往存在着某種聯系，在關係模型中實體及實體間的聯系都是用關係來描述，這樣就自然存在着關係與關係間的引用。

參照完整性約束是定義外碼與主碼之間的引用規則，它是對關係間引用數據的一種限制。若屬性F是關係R的外碼，它與基本關係S的主碼K對應，則對R中每個元組在F上的值只允許兩種可能，即要麼取空值(NULL)，要麼取等於S中某個元組的主碼值，其中，關係R與S可以是不同的關係，也可以同一個關係。

1. 簡述甚麼是用戶定義完整性約束。

用戶定義完整性約束是針對某一應用環境的完整性約束條件，它反映了某一具體應用所涉及的數據應滿足的要求，關係模型提供定義和檢驗這類完整性規則的機制，其目的是用統一的方式由系統來處理它們，不再由應用程序來完成這項工作。在實際系統中，這類完整性規則一般在建立數據庫表時同時進行定義，也可以由應用編程人員在各塊的具體編程中通過程序進行檢查和控制。

1. 簡述關係规范化過程

第一范式(1NF)，一個不含重複組的關係

第二范式(2NF)，在第一范式基礎上，所有非主屬性對候關係字完全函數依賴。

第三范式(3NF)，在第二范式基礎上，每一個一非主屬性都不傳遞函數依賴於候選關鍵字

在第三范式再改為第三范式的改進形式(BCNF)，R為關係，X,Y為其屬性集，F為其函數依賴集，F所有函數依賴 $X \rightarrow Y$ (Y不於X)中的X必包含候選關鍵字，R則為BCNF。

第3章 數據庫設計

1. 數據庫演變過程的角度來看，數據庫的生周期可分為兩階段，包括 數據庫分析和設計階段，和 數據庫實現與操作階段。
2. 數據庫分析和設計階段包括了 需求分析、概念設計、邏輯設計、物理設計。
3. 數據庫實現與操作階段包括了 數據庫的實現、操作與監督、修改與調整。
4. 數據庫設計具有兩個十分重要的目標，即滿足應用 功能需求 和良好的 數據庫性能。
5. 數據庫設計內容包括 數據庫結構設計 和 數據庫行為設計。數據庫結構設計包括 數據庫的概念結構設計、邏輯結構設計 和 物理結構設計。
6. 數據庫設計方法包括 直觀設計法、規範設計法、計算機輔助設計法。
7. 規範設計法包括 新奧爾良設計方法，基於 E-R模型的數據庫設計方法、基於 第三范式的設計方法。
8. 計算機輔助軟件工程工具稱 CASE工具。
9. 數據庫設計的過程分 需求分析階段，結構設計階段，行為設計階段，數據庫實施階段，數據庫運行和維護階段。當中結構設計階段包括 概念結構設計、邏輯結構設計 和 物理結構設計。行為設計包括 功能設計、事務設計 和 程序設計。數據庫實施階段包括 加載數據庫數據 和 調試運行應用程序。

-----以下有時間就看，不求高分可放棄-----

數據庫設計的基本步驟

1. 需求分析是數據庫設計的起點。目標是了解與分析用戶的信息及應用處理的要求，並將結果按一定格式整理而形成需求分析報告。為後續 概念設計、邏輯設計、物理設計、數據庫建立與維護 的依據。
2. 需求分析分四個步驟，數據庫範圍、分析數據應用過程、收集與分析數據 和 編寫需求分析報告。
3. 收集與分析數據可以從 靜態結構、動態結構 及 數據約束 三個方面展開。
4. 靜態結構可以通過 數據分類表 和 數據元素表 進行說明。
5. 動態結構可以通過 任務分類表 和 數據操作特徵表 進行說明。
6. 收集與分析數據中的 數據約束 概念是 數據的安全性保密性，數據的完整性、響應時間 和 數據恢復。
7. 概念結構設計常用方法有 實體分析法(自頂向下法) 和 屬性綜合法(自底向上法) 兩種。
8. 物理設計具體任務主要是確定數據庫在存儲設備上的 存儲結構 及 存取方法，因DBMS的不同還可能包括 建立索引 和 聚集，以及 物理塊大小、緩沖區個數和大小，數據壓的選擇 等。
9. 數據庫實施包括 加載數據、應用程序設計、數據庫試運行。
10. 邏輯結構設計方法

確定各屬性間的函數依賴關係

對於各個關係模式之間的數據依賴進行極小化處理，消除冗餘的聯系

判斷每個關係模式的范式，根據實際需要確定最合適的范式。

安照需求分析階段得到的處理要求，分析這些模式對這樣的應用環境是否合適，確定是否要對些模式進行合并或分解

對關係模式進行必要的分解，提高數據操作的效率和存儲空間的利用率。

-----以上有時間就看，不求高分可放棄-----

- 1. E-R圖一般適用於建立數據庫的 概念模型 。
- 2. 物理設計可以用 建立索引 和 聚集

第4章 SQL與關係數據庫基本操作

- 1. 結構化查詢語言(SQL) 是專門用來數據庫通信的語言，它可以幫助用戶操作關係數據庫。
- 2. SQL歷史

時間	SQL的名稱
1989年4月	SQL-89
1992年	SQL-92
1999年	SQL-99 (SQL3)

- 1. SQL特點(選擇/填空，理解)

SQL不是某個特定數據庫供應商專有語言

SQL簡單易學 它的語句全都具有很強描述性的英語單詞所組成，而且這些單詞的數目不多

SQL盡管看上去很簡單，但它實際是一種強有力的語言，靈活使用其語言元素，石以進行非常復雜和高級的數據庫操作。

- 1. SQL語句不區分 大小寫 。
- 2. 數據定義語言(DDL)，主要於對數據庫及數據庫中的各種 對象 進行 創建 、 刪除 ， 修改 等操作。

CREATE

ALTER

DROP

- 1. 數據操縱語言(DML)，主要用於 操從 數據庫各種 對象 。特別是 檢索 和 修改數據 。

SELECT

INSERT

DELETE

UPDATE

1. 數據控制語言(DCL)，主要要用於 安全管理 。

GRANT

REVOKE

1. SQL調用和會話規則，以提高SQL的 靈活性 ， 有效性 、 共享性 以及使SQL具有更多的高級語言 特徵。
2. MySQL是一個 關係數據管理系統 (RDBMS)，其優點是 體積小 、 速度快 、 開放源代碼 、 遵循 GPL 。
3. MySQL數據庫管理系統構建信息管理系統或互聯網網站的應用環境，主要有 LAMP 和 WAMP 。
4. LAMP指的是 Linux + Apache + MySQL + PHP/Perl/Python
5. WAMP指的是 Windows + Apache + MySQL + PHP/Perl/Python
6. 變量的念意及其屬性

變量用於臨時存儲數據，變量中的數可以隨着程序的運行而變化，變量有名字和數據類型兩個屬性。其中，變量的名字用於標識變量，變量的數據類型用於確定變量中數值的格式和可執行的運算。

1. 在MySQL，變量分為 用戶變量 和 系統變量 ，前者在變量前添加一個 @ 符號，後者在名稱前加兩個 @ 符號。
2. 寫出SQL

操作	SQL
Create DB	CREATE DATABASE db_name;
Use DB	USE db_name;
Create Table	CREATE TABLE table_name (col_name_1 int, col_name_2 varchar(100) col_name_3 char(10));
Insert row	Insert into table_name (col_name_1, col_name_2, col_name_3) values (1,'TEXT','TEXT2');
Inert column	ALTER TABLE table_name ADD COLUMN col_name_4 varchar(2);
Modify column name	ALTER TABLE table_name CHANGE COLUMN col_name_4 col_name_5 varchar(2);
Delete row	DELETE table_name WHERE col_name_1=1;

1. 視圖與表的區別

1. 視圖不是數據庫中真實的表，是一張虛擬的表，其結構和數據是建立在對數據庫中真實表的查詢基礎上。

2. 視圖的內容是由存儲在數據庫中進行查詢操作的SQL語句來定義的，它的列數據與行數據均來自於定義視圖的查詢所引用的真實表，並且這些數據是在引用圖動態生成。
3. 視圖不是以數據集的形式存儲在數據庫中，它所對應的數據實際上是存儲在視圖所引用的真實表中。
4. 視圖是用來查看存儲在別處的數據的一種虛擬表，而其自身並不存儲數據。

1. 使用視圖的優點

1. 集中分散數據
2. 簡化查詢語句
3. 重用SQL語句
4. 保護數據安全性
5. 共享所需數據
6. 更改數據格式

1. 子查詢

指在SELECT語句中創建子查詢，即可嵌套在其他SELECT查詢中的SELECT查詢。MySQL中，有表子查詢、行子查詢、列子查詢、標量子查詢。

第5章 數據庫編程

1. 存儲過程

存儲過程是一組為了完成某項特定功能的SQL語句集，某實質上就是一段存儲在數據庫中的代碼。

1. 簡述存儲過程的好處

1. 可增強SQL語言的功能和靈活性
2. 良好的封裝性
3. 高性能
4. 可減少網絡流量
5. 存儲過程為一種安全機制來確保數據庫的安全性和數據的完整性。

Delimiter \$\$

Create Procedure p_name(p_parmetere)

BEGIN

DECLARE x1 int(10);

DECLARE y1 int(10);

SET x1=3

SELECT X1 FROM table_name WHERE col_name='x' INTO y1,

UPDATE table_name SET COL_NAME=x1 WHERE col_name='Y';

END \$\$

當中\$\$指的是用戶定義的結束符。DECLARE是聲明局部變量。SET是賦值給DECLARE聲明的。

1. 局部變量的注意事項

1. 只能在存儲過程體的BEGIN...END語句塊中聲明
2. 必須在存儲過程的開頭處聲明
3. 局部變量的作用範圍僅限於聲明它的BEGIN...END語句塊，其他語句塊中的語句不可以使用它。
4. 局部變量不同於用戶變量

1. 局部變量與用戶變量區別是

1. 局部變量聲明時，在其前面沒有使用@符號，並它只能被聲明它的BEGIN...END語句塊中的語句所使用。
2. 用戶變量在聲明時，會在其名稱前面使用@符號，已聲明的用戶變量存在於整個會話之中。
1. 游標是一被SELECT語句檢索出來的 結果集 。在存儲了游標後，應用程序或用戶就可以根據需要滾動或瀏覽其中的數據。

DELIMITER \$\$

CREATE PROCEDURE p_name(parameter CHAR(10))

BEGIN

DECLARE var_name varchar;

DECLARE cursor_name CURSOR FOR SELECT x1 FROM TABLE_NAME;

OPEN cursor_name;

FETCH cursor_name INTO var_name;

...

CLOSE cursor_name;

END \$\$

當中FETCH是用作 取出數據 ，與 SELECT...INTO 用法一樣。OPEN與CLOSE用作 打開 與 關閉 游標。調用存儲過程用 CALL(p_name);

1. 使用游標注意事項

1. 游標只能用於存儲過程或存儲函數中，不能單獨在單詢操作中使用。
2. 在存儲過程或存儲函數中可以定義多個游標，但是在一個BEGIN...END語句塊中每個游標的名字必須是唯一的。
3. 游標不是一條SELECT語句，是被SELECT語句檢索出來結果集。

1. 添加 IF EXISTS 可以判斷有沒有存在，有就做操作，沒有就不做操作。

2. 存儲過程與存儲函數差別

存儲函數不能擁有輸出參數，這是因為存儲函數自身就是輸出參數，而存儲過程可以擁有輸出參數。

可以直接對存儲函數進行調用，且不需要使用CALL語句，而對存儲過程的調用，需要使用CALL語句。

存儲函數中必須包含一條RETURN語句，而這條特殊的SQL語句不允包含於存儲過程中。

```
DELIMITER $$

CREATE FUNCTION fun_name(parameter varchar)

RETRUNS CHAR(2)

NAME

BEGIN

DECLARE x1 INT;

...

END $$
```

1. 存儲函數不能與 存儲過程 同一樣的名字。

第6章 數據庫安全與保護

1. 數據庫完整性是指數據庫中數據的 正確性 和 相容性 。
2. 完整性約束的條件的作用對象可以是 列 、 元組 和 表 。
3. 關係模型中可以有 三類完整性約束，分別是 實體完整性 、 參照完整性 和 用戶定義的完整性 。
4. 實體完整性約束分別有 主鍵約束 和 候選鍵約束 。分別用關鍵字 PRIMARY KEY和 UNIQUE 實現。參照完整性用關鍵字 FOREIGN KEY(col_name,...) reference_definition 實現。
5. 用戶定義的完整性包括 非空約束 、 CHECK約束 和 觸發器 。
6. 非空約束的關鍵字是 NOT NULL ，CHECK約束的關鍵字為 CHECK(expr) 。
7. 命名完整性約束用 CONSTRAINT ，用 ALTER TABLE 更改完整性約束，是先刪除該約束，再增加一個新的約束。
8. 寫出創建table時添加Primary key, unique key, foreign key兩種寫法

CREATE TABLE tb_name (CREATE TABLE tb_name (
id int(2) PRIMARY KEY,	id int(2),
name char (20)	name char (20),
);	PRIMARY KEY(id)
);

CREATE TABLE tb_name (CREATE TABLE tb_name (
id int(2) PRIMARY KEY,	id int(2),
name char (20)	name char (20),
);	PRIMARY KEY(id)
);
<hr/>	
CREATE TABLE tb_name (CREATE TABLE tb_name (
id int(2) UNIQUE,	id int(2),
name char (20)	name char (20) ,
);	UNIQUE(id)
);
<hr/>	
CREATE TABLE tb_name (CREATE TABLE tb_name (
id int(2) REFERENCES customer(cus_id),	id int(2),
name char (20)	name char (20) ,
);	FOREIGN KEY (id) REFERENCES customer(cus_id)
);

1. 甚麼時觸發器

觸發器是用戶定義在關係表上的一類由事件驅動的數據庫對象，也是一種保證數據完整性的方法。觸發器一旦被定義，任何對表的修改操作均由數據庫服務器自動激活相應的觸發器。

1. MySQL支持三種觸發器，分別是 INSERT觸發器、DELETE觸發器、UPDATE觸發器。連同AFTER BEFORE共 6 種觸發器。

CREATE TRIGGER trigger_name AFTER/BEFORE INSERT

On table_name FOR EACH ROW trigger_body;

1. 數據庫安全性是指保護數據庫以防止 不合法的使用 造成 數據泄露、更改 或 破壞。
2. MySQL，數據庫系統對數據的安全管理是使用 身份驗證、數據庫用戶權限確認。
3. MySQL訪問控制兩種方法：用戶賬戶 與 權限管理。
4. 寫下下列SQL

事情	SQL
Create user	CREATE USER user_name IDENTIFIED BY password;

事情	SQL
Delete user	DROP USER user_name; DROP USER user_name@localhost;
Delete 多位user	DROP USER user_name_1, user_name_2;
Rename user	RENAME USER user_name TO user_1 ;
Change password	SET PASSWORD FOR user_1 PASSWORD('123456');
Grant user	GRANT privilege ON object TO user; GRANT select ON table_name TO user_1 ;
Revoke user	REVOKE privilege on object FROM user; REVOKE select ON table_name FROM user_1;

1. 授權用戶權限帶 WITH GRANT OPTION 指給被授權用戶具有將該權限授予給其它用戶的權利。
2. 使用REVOKE語句，必須擁有MySQL數據庫的全局 CREATE USER 權限或 UPDATE 權限。
3. 為了保證數據的一致性和正確性，數據庫系統必須保證事務具有四個特徵，即 原子性 、 一致性 、 隔離性 、 和 持續性 ，英文簡稱為 ACID 。
4. 甚麼是事務和事務特徵。

事務是用戶定義的一個數據操作序列，些操作可作為一個完整的工作單元，要麼 全部執行，要麼全部不執行，是一個不可分割的工作單位。

數據庫系統必須保證事務具有四個特徵，即 原子性 、 一致性 、 隔離性 、 和 持續性 ，英文簡稱為 ACID 。

1. 簡述數據庫系統必須保證事務具有四個特徵

1. 原子性，保證事務包含的一組更新操作是原子不可分的，即事務是不可分割的最小工作單位，所包含的這些操作是一個整體。
2. 一致性，要求事務必須滿足數據庫的完整性約束，毋事務執行完畢後將數據由一個一致性狀態轉變到另一個一致性狀態。
3. 隔離性，要求事務是彼此獨立、隔離的，即一個事務的執行不能被其他事務所干擾 一個事務對數據庫變更的結果必須在它COMMIT後，另一個事務才能存取。
4. 持續性，也稱為永久性，提指一個事務一旦提交，它對數據庫中數據的改變就應該是永久性的，且接下來的其它操作或故障不應對其執行結果有任何影響。

1. MSQL中，只有使用了 InnoDB引擎 的數據庫或表才支持事務。
2. 并發操作會有 丟失更新 、 不可重復讀 、 讀脏數據 。
3. 封鎖是最常用的 並發控制 技術。一個鎖實質上就是允許或阻止一個事務對一個數據對象的 存取特權 。
4. 需要時，事務通過向系統請求對它所希望的數據對象加鎖，以確保它不被非預期改變。
5. 排他鎖，簡寫 X鎖 ，寫操作要求加。共享鎖，簡寫 S鎖 ，讀操作要求加。其邏輯為

X鎖，直到結束，其它事務的鎖才可請求

S鎖，其它事務可以加S鎖，但X鎖要等待S鎖釋放才可以請求成功。

數據庫會自動請對應的鎖

鎖是直到COMMIT或ROLLBACK才釋放。

1. 鎖的級別又稱為 一致性級別 和 隔離度 。
2. 0級鎖，封鎖的事務不重寫其他非0級封鎖事務的未提交數據
3. 1級鎖，被封鎖的事務不允許重寫未提交的更新數據。這防止 丟失更新 的發生。
4. 2級鎖，被封鎖的事務既不重寫也不讀未提交的更新數據。這具有 1級鎖 的效果外還防止了 讀脏數據 。
5. 3級鎖，被封鎖的事務不讀未提交的更新數據，不寫任何未提交數據，這除了包括 2級鎖 外，還不寫未提交的讀收據，防止了 不可重讀 的問題。
6. 封鎖可能引起 活鎖 與 死鎖 。死鎖是指 兩個以上事務循環等待被同一組中另一事務鎖住的數據單元的 情況 。
7. 預防死鎖辦法有

1. 一次性鎖請求
2. 鎖請求排序
3. 序列化處理
4. 資源剝奪

1. 可串行性指 一組事務的一個調度就是它們的基本操作的一種排序。若在一個調度中，對任意兩個事務 T1和T2，要麼T1的所有操作都在T2所有操作之前，要麼反之，則該調度是串行的 。
2. 兩段封鎖法(2PL)是一種最簡單而有效保障封鎖其調度是 串行性 的方法。兩段封鎖法是事務遵循兩段鎖協議的調度方法。分別有兩個段 發展或加鎖階段 和 收縮或釋放鎖階段 。
3. S(x)：加共享鎖， X(y)：加排他鎖， UL(x)：解鎖， R(x)：讀數據， W(y,a)：寫a到y中， C：commit

T1: B=A+1

T1: S(A), X(B), a=R(A), W(B,a+1),UL(A),UL(B),C

1. 簡述數據庫備份概念(識記，即選擇與填空題多)

數據庫備份是指通過導出數據或者復制表文件的方式來制作數據庫複本，數據庫恢復則是當數據庫出現故障或遭到破壞時，將備份的數據庫加載到系統，從而使數據庫從錯誤狀態恢復到備份時的正確狀態。

SELECT INTO OUTFILE 'file_name' export options

FIELDS

Terminated by 'string'

Enclosed by char

Line terminated by 'string';

1. 簡述數據庫恢復概念(識記，即選擇與填空題多)

數據庫恢復是以備份為基礎的，它是與備份相應的系統維護和管理操作。系統進行恢復操作時，先執行一些系統安全性的檢查，包括檢查所要恢復的數據是否存在、數據是否變化及數據庫文件是否兼容等。然後根據所採用的數據庫備分類型採取相應的恢復措施。

```
LOAD DATA INFILE 'file_name' INTO TABLE table_name
```

```
FIELDS
```

```
Terminated by 'string'
```

```
Enclosed by char
```

```
Line terminated by 'string';
```

第7章 數據庫應用設計與開發實例

跳過

第8章 數據庫管理技術的發展

\1. 第一代數據庫系統以層次模型和網狀模型(識記，即選擇與填空題多)

(1) 支持三級模式，內模式、外模式、模式

(2) 用存取路徑來表示數據之間的聯系

(3) 獨立的數據定義語言

(4) 導航的數據語言

\2. 第二代數據庫系統以關係模型(識記，即選擇與填空題多)

(1) 奠定了關係模型的基礎

(2) 研究了關係數據語言，有關係代數，關係演算法，SQL語言和QBE等

(3) 研制了大量的RDBMS的原型，模型簡單清晰，理論基礎好，獨立性強，數據庫結合非過程化和標準化等特點。

\3. 新一代數據庫最有以下三個基本特徵(識記，即選擇與填空題多)

(1) 第三代數據庫系統應支持數據管理、對象管理和知識管理

(2) 第三代數據庫系統必須保持或繼承第二代數據庫系統技術

(3) 第三代數據庫系統必須對其他系統開發

4. 數據倉庫和數據挖掘，聯機事務處理(OLTP)，主要針對少量記錄進行查詢或修改。聯機分析處理(OLAP)，主要對大量數據查詢與分析。

5. 在20世紀80年代 數據倉庫技術應運而生。

數據倉庫的建立將操作型處理和分析型處理區分開來，傳統的數據庫技術為操作處理服務。數據倉庫為分析型處理服務。

4. 數據倉庫的特徵(識記，即選擇與填空題多)

1. 面向主題
2. 集成性
3. 數據的非易失性
4. 數據時變性

4. 數據倉庫的重要概念(識記，即選擇與填空題多)

1. 粒度，它影響存儲在數據倉庫中的數據量的大小，同時影響數據倉庫所能回答的查詢類型。在數據倉庫中數據量大小與查詢的詳細程度之間要做出權衡。
2. 分割，是將數據分散到各自的物理單元上，以便能分別處理，以提高數據處理的效率。
3. 維，數據倉庫中的數據按照不同的維組織來形成一個多維的立方體，維的概念使用戶能夠從多個角度觀察數據倉庫中的數據，以便深入了解包含在數據中的信息。

4. 數據挖掘技術具有下幾種功能(識記，即選擇與填空題多)

1. 概念描述
2. 關聯分析
3. 分類與預測
4. 聚類
5. 孤立點檢測
6. 趨勢和演變分析

4. 概念描述是歸納總結出數據的某些特徵。
5. 分類是找出一個類別的概念描述。常用的分類模型和算法有 決策樹模型、神網絡模型、綫性回歸模型。
6. 兩個或多個變量的取值之間存在某種規律性，稱為關聯。
7. 聚類是把數據按照相似性歸納成若干類別。
8. 孤立點是指數據中與整體表現行為不一致的數據集合。
9. 新一代數據庫與計算機結合

計算機技術	數據庫技術
分布式處理技術	分布式數據庫
並行處理技術	並行處理數據庫
	演繹數據庫
人工智能技術	智識數據庫
	主動數據庫
多媒體技術	多媒體數據庫
模糊技術	模糊數據庫

計算機技術	數據庫技術
移動通訊技術	移動通訊數據庫
Web技術	Web數據庫

4. 大數據，IBM提出4V分別是 大量化 、 多樣化 、 快速化 、 真實化 。
5. 大數據管理技術典型代表， 大數據存儲 ， NoSQL數據管理系統 ， MapReduce技術 。
6. NoSQL數據管理系統通常有 鍵值模型 、 文檔模型 、 列模型 和 圖模型 等。
7. 鍵值存儲是 NoSQL 數據庫採用最多的數據存儲方式，使用 哈希函數 實現關鍵字到值的快速映射。適合通過主鍵進行 查詢與遍歷 。
8. 文檔存儲的存儲格式可以多樣化，適合存儲系統日誌等 非結構化 數據。
9. 列存儲是以列為單位來存儲數據的，擅長以列為單位讀入數，比較適合對某一列進行隨機 查詢處理 。
10. 圖存儲數據庫是共圖理論構建的，使用 結點 、 屬性 和 邊 的概念。