

密训·资料

数据库系统原理（全国）

1904

MI XUN ZI LIAO

编前语

试卷说明：本科目考试题型为单选、填空、简答、设计题和综合题。各题型分值分布情况如下表所示：

题型	分值
单选题	2分×15道=30分
填空题	1分×10道=10分
简答题	4分×6道=24分
设计题	4分×5道=20分
综合题	8分×2道=16分

试卷满分 100 分，考试时间 150 分钟。

资料说明：本资料提供的都是考试重点，但是在重点中，按照知识点的重要程度，对知识点进行了标星。星标数目越多，表示知识点越重要，越可能考到。

目录

第一章 数据库系统概述.....3

第二章 关系数据库.....5

第三章 数据库设计.....10

第四章 SQL 与关系数据库基本操作.....13

第五章 数据库编程.....17

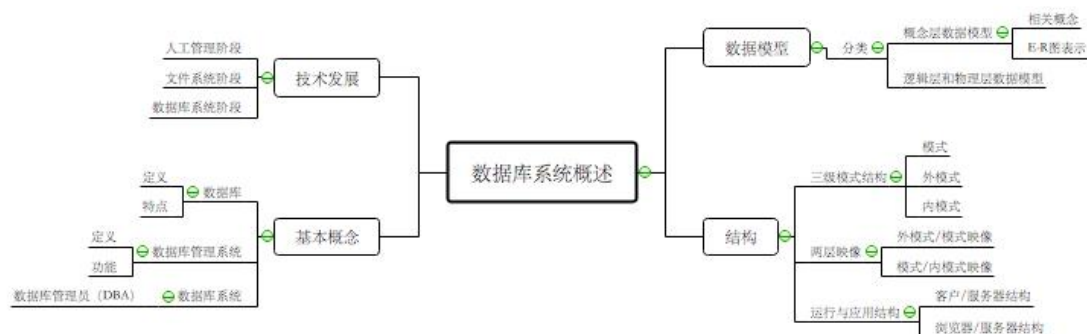
第六章 数据库安全与保护.....18

第七章 数据库应用设计与开发实例.....20

第八章 数据管理技术的发展.....21



第一章 数据库系统概述



1.1.2 数据库 ★★

定义★★	有组织的、可共享的数据集合；较小的冗余度、较高的数据独立性，易于扩展，可多用户共享；
特点★	三个基本特点：永久存储、有组织和可共享。

1.1.3 数据库管理系统 ★

- 数据库管理系统 (DBMS) 是专门用于建立和管理数据库的一套软件，介于应用程序和操作系统之间。

- 数据库管理系统的主要功能。 ★★★

- 1.数据定义；
- 2.数据操纵；
- 3.数据库的运行管理；
- 4.数据库的建立和维护；
- 5.数据组织、存储和管理；
- 6.其他功能：主要包括与其他软件的网络通信功能等。

1.1.4 数据库系统 ★

- 数据库管理员 (DBA)：专门负责对数据库进行维护，并保证数据库正常、高效运行的人员。

1.2.1 人工管理阶段 ★★

- 人工管理阶段的特点：1.数据不保存；2.应用程序管理数据；3.数据面向应用。

1.2.2 文件系统阶段 ★

- 相对于人工管理数据的方法，文件系统提供了物理数据独立性，使应用程序与数据的具体物理存储结构分离。

1.2.3 数据库系统阶段 ★★★

数据库系统具有特点：

- (1) 数据集成；
- (2) 数据共享性高；
- (3) 数据冗余小；
- (4) 数据一致性；引起不一致的根源是数据冗余。★

- (5) 数据独立性高；
- (6) 实施统一管理与控制；主要包括数据安全性、完整性、并发控制与故障恢复等，即数据库保护。★
- (7) 减少应用程序开发与维护的工作量。

1.3.1.0 三级模式结构 ★

- 模式也称为概念模式或逻辑模式。它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。
- 外模式也称为子模式或用户模式。它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，是与某一应用有关的数据的逻辑表示。是用于满足不同数据库用户需求的数据视图，即用户视图。
- 内模式也称为存储模式。DBMS 提供内模式描述语言来严格地定义内模式，即定义所有内部记录类型、索引和文件的组织方式，以及所有数据控制方面的细节。

1.3.1.4 三级模式结构的两层映像与数据独立性 ★★★

- 三级模式之间的两层映像：外模式/模式映像与模式/内模式映像。★
- 外模式/模式映像：定义了各个外模式与概念模式之间的映像关系，这些映像定义通常在各自的外模式中加以描述。数据库管理员（DBA）调整外模式/模式映像，使外模式保持不变，从而应用程序的编程人员就不必去修改那些依据数据的外模式所编写的应用程序，实现了外模式不受概念模式变化的影响，并保证了数据与程序的逻辑独立性。
- 模式/内模式映像：定义了数据库全局逻辑结构与物理存储之间的对应关系，这种映像定义通常是在模式中加以描述的。数据库管理员（DBA）调整模式/内模式映像，使数据库系统的模式保持不变，从而也不必去修改应用程序，实现了概念模式不受内模式变化的影响，并保证了数据与程序的物理独立性。

1.3.2.0 数据库系统的运行与应用结构 ★

- 数据库常见的运行与应用结构有：客户/服务器结构、浏览器/服务器结构。

1.4.1 数据特征与数据模型组成要素 ★

- 数据具有静态与动态两种特征。
- 数据模型通常由数据结构、数据操作和数据约束三个要素组成。

1.4.2 数据模型的分类 ★

- 第一类是概念层数据模型，第二类是逻辑层数据模型和物理层数据模型。

1.4.2.1 概念层数据模型 ★★★★★

- 概念层数据模型相关基本概念：
 - 实体：客观存在并可相互区别的事物称为实体。
 - 属性：实体所具有的某种特性称为实体的属性。
 - 码或键：可唯一标识实体的属性集称为码或键。
 - 实体型：具有相同属性的实体必然具有共同的特征和性质。
 - 实体集：同型实体的集合称为实体集。
 - 联系：实体内部的联系通常是指实体各属性之间的联系，实体之间的联系是指不同实体

之间的联系。

- 概念模型的表示方法：用 **E-R** 图来描述现实世界的概念模型，实体用矩形表示；属性用椭圆形表示；联系用菱形表示。

1.4.2.2 逻辑层数据模型 ★★★

- 主要的逻辑数据模型有**层次**模型、**网状**模型、**关系**模型、**面向对象**模型。

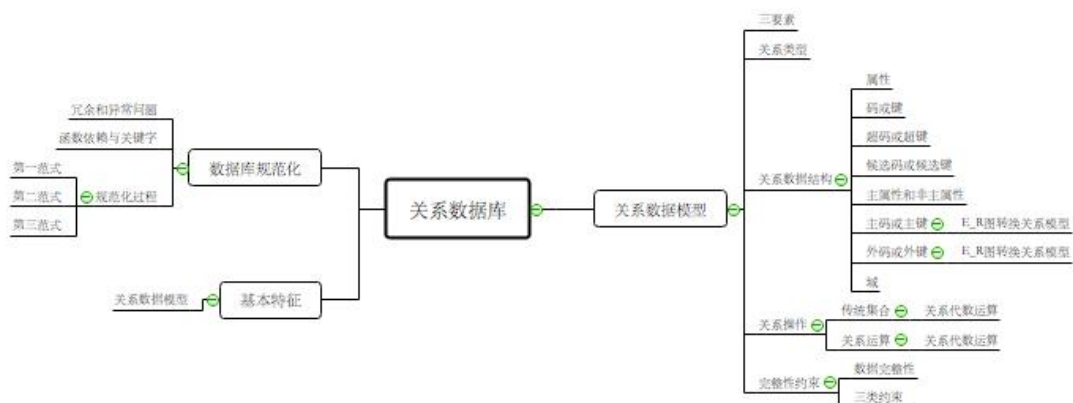
(1) 层次模型：数据库最早使用的数据模型。特点：有且仅有一个结点没有父结点，它称作根结点；其他结点有且仅有一个父结点。

(2) 网状模型：以网状结构表示实体与实体之间的联系。

(3) 关系模型：用二维表结构来表示实体及实体间联系的模型，并以二维表格的形式组织数据库中的数据。优点如下：关系模型是建立在严格的数学概念基础上的；关系模型的概念单一，统一用关系来表示实体以及实体之间的联系，对数据的检索和更新结果同样也是用关系来表示；关系模型的存取路径对用户透明，从而具有更高的数据独立性、更好的安全保密性，也简化了程序员的工作和数据库开发建立的工作。

(4) 面向对象模型：与数据库相结合所构成的数据模型称为面向对象模型

第二章 关系数据库



2.1 关系数据库概述 ★

- 关系数据库的基本特征是使用**关系数据模型**组织数据，这种思想源于数学。

2.2.0 关系数据模型 ★★

- 关系模型包含三个组成要素，分别是**关系数据结构**、**关系操作集合**和**关系完整性约束**。

2.2.1.2 关系 ★★

关系可以有三种类型，即基本关系、**查询表**和**视图表**。

2.2.1.4 属性 ★

- 表中属性的个数称为关系的**元**或度；
- 属性值的取值范围称为**值域**。

2.2.1.8 码或键 ★

- 属性的值都能用来**唯一标识**该关系的元组，则称这些属性为该关系的码或键。

2.2.1.9 超码或超键 ★

- 如果在关系的一个码中移去某个属性，它仍然是这个关系的码，则称这样的码或键为该

关系的超码或超键。

- 默认的超码或超键，即该关系的所有属性的集合，也是这个关系的最大超码或超键。

2.2.1.10 候选码或候选键 ★

- 如果在关系的一个码或键中，不能从中移去任何一个属性，否则它就不是这个关系的码或键。则称这样的码或键为该关系的候选码或候选键。
- 一个关系的候选码或候选键是这个关系的最小超码或超键。

2.2.1.12 全码或全键 ★

- 一个关系模式的所有属性集合是这个关系的主码或主键，则称这样的主码或主键为全码或全键。

2.2.1.13 主属性和非主属性 ★

- 关系中包含在任何一个候选码中的属性称为主属性或码属性，不包含在任何一个候选码中的属性称为非主属性或非码属性。

2.2.1.11 主码或主键 ★★★★★

- 在一个关系的若干个候选码或候选键中指定一个用来唯一标识关系的元组，则称这个被指定的候选码或候选键为该关系的主码或主键。

2.2.1.14 外码或外键 ★★★★★

- 当关系中的某个属性不是这个关系的主码或候选码，而是另一关系的主码时，称该属性为这个关系的外码或外键。

2.2.1.16 域 ★

域表示属性的取值范围。

2.2.2.1 基本的关系操作 ★★

- 关系模型常用关系操作包括查询操作和插入、删除、修改操作两大部分。
- 五种基本查询操作：选择、投影、并、差、笛卡尔积。

2.2.2.2 关系数据语言的分类 ★

- 关系数据语言可以分为三类：关系代数语言、关系演算语言以及兼具两者双重特点的语言（SQL）。
- 关系演算可按谓词变元的基本对象是元组变量还是域变量，分为元组关系演算和域关系演算。

2.2.2.3 关系代数 ★★★★★

A.传统的集合运算

- 4种运算：并、差、交、笛卡尔积。

1.并： R_3 是由属于关系 R_1 或 R_2 的所有不同元组所组成，记为 $R_3 = R_1 \cup R_2$ 。

2.差： R_3 是由属于关系 R_1 ，但不属于 R_2 的元组组成，记为 $R_3 = R_1 - R_2$ 。

考题看一看

关系R和S如下表

R				S			
	A	B	C		A	B	C
	1	2	3		2	4	6
	4	5	6		4	5	6
	7	8	9				

R - S的结果是 ()

A:

A	B	C
2	4	6
4	5	6

B:

A	B	C
1	2	3
7	8	9

C:

A	B	C
1	2	3
4	5	6
7	8	9
2	4	6

D:

A	B	C
2	4	6
7	8	9

答案：B

解析：两个关系R和S，R和S的差运算产生一个新关系R-S。R-S是属于关系R，但不属于S

的元组。所以R-S的结果为

A	B	C
1	2	3
7	8	9

3.交： R3 是由既属于关系 R1，同时又属于 R2 的元组组成，记为 $R3=R1 \cap R2$ 。

考题看一看

已知关系S1和S2如下表所示，则S1与S2进行并运算，其结果的元组数为 ()

S1	<table><tr><th>A</th><th>B</th><th>C</th></tr><tr><td>1</td><td>2</td><td>3</td></tr><tr><td>4</td><td>2</td><td>5</td></tr><tr><td>6</td><td>7</td><td>3</td></tr></table>	A	B	C	1	2	3	4	2	5	6	7	3	S2	<table><tr><th>A</th><th>B</th><th>C</th></tr><tr><td>4</td><td>8</td><td>9</td></tr><tr><td>4</td><td>2</td><td>5</td></tr><tr><td>6</td><td>8</td><td>9</td></tr></table>	A	B	C	4	8	9	4	2	5	6	8	9
A	B	C																									
1	2	3																									
4	2	5																									
6	7	3																									
A	B	C																									
4	8	9																									
4	2	5																									
6	8	9																									

A:0 B:4 C:5 D:6

答案：C

解析：已知两个关系s1和s2，即s1和s2并运算产生一个新关系 $s1 \cup s2$ ， $s1 \cup s2$ 的结果是由属于s1和s2去掉重复元组后的所有元组组成。所以 $s1 \cup s2$ 的元组数为5。

4.笛卡尔积： R1 为 m 元关系，R2 为 n 元关系，新关系 R3，记作 $R3=R1 \times R2$ 。R3 有(m+n) 个分量，($m \times n$) 个元组组成。

B.专门的关系运算

● **投影运算、选择运算、连接运算和除运算；**

(1) 选择 (SELECT)

选择运算表示为： $\sigma_F(R)$ ，其中，F 为条件表达式，R 为指定的被运算关系名。

(2) 投影 (PROJECTION)

投影运算表示为： $\pi_A(R)$ ，其中，R 为被运算关系名，A 为属性序列。形式如下：

(3) 连接 (JOIN)

连接运算表示为： $R \bowtie S$

连接运算是选取两个指定关系中的属性满足给定条件的元组连接在一起来组成一个新关系。

连接操作中有两种最为常用的连接，即等值连接和自然连接。

(4)除 (DIVISION)

除运算表示为： $R \div S$ ，在除运算中，若被除关系为 m 元关系，除关系为 n 元关系，则运算结果为一个 m-n 元关系。

考题看一看

某学生社团管理系统的数据库包含如下关系表：

学生(学号，姓名，年龄，性别，所在系)

协会(协会编号，协会名，办公地点，负责人)

入会(学号，协会编号，入会日期)

实现下列操作：

使用关系代数语言查询加入“科技协会”的学生姓名和所在系。

答案：

$\pi_{\text{姓名, 所在系}} (\sigma_{\text{协会名} = \text{'科技协会'}} (\text{学生} \bowtie \text{协会} \bowtie \text{入会}))$

解析：

投影运算表示为： $\pi_A(R)$ ，其中，R 为被运算关系名，A 为属性序列。

选择运算表示为： $\sigma_F(R)$ ，其中，F 为条件表达式，R 为指定的被运算关系名。

连接运算表示为： $R \bowtie S$ ，其中，R 和 S 代表两个不同的关系。

因为是要查询加入“科技协会”的学生姓名和所在系，所以要对三个关系进行查询，即投

影运算中的属性序列为学生姓名和所在系，选择运算中的条件表达式为“协会名=‘科技协会’”。故最后为：

$\pi_{\text{姓名, 所在系}} (\sigma_{\text{协会名} = \text{'科技协会'}} (\text{学生} \bowtie \text{协会} \bowtie \text{入会}))$

2.2.3.0 关系的完整性约束 ★★

- 数据库的数据完整性是指数据库中数据的正确性、相容性和一致性。
- 关系模型中有三类完整性约束，分别是实体完整性约束、参照完整性约束和用户定义完整性约束。

2.3.1 关系模式中可能存在的冗余和异常问题 ★★

- 关系模式可能会存在的如下问题：

1. 数据冗余
2. 更新异常
3. 插入异常
4. 删除异常

2.3.2 函数依赖与关键字 ★★★★★

- 完全函数依赖

设 R 为任一给定关系，X、Y 为其属性集，若 $X \rightarrow Y$ ，且对 X 中的任何真子集 X' 都有 $X' \not\rightarrow Y$ ，则称 Y 完全函数依赖于 X。

- 部分函数依赖

设 R 为任一给定关系，X、Y 为其属性集，若 $X \rightarrow Y$ ，且 X 中存在一个真子集 X' 满足 $X' \rightarrow Y$ ，则称 Y 部分函数依赖于 X。

- 传递函数依赖

设 R 为任一给定关系, X 、 Y 、 Z 为其不同属性子集, 若 $X \rightarrow Y$, $Y \rightarrow X$, $Y \rightarrow Z$, 则有 $X \rightarrow Z$, 称为 Z 传递函数依赖于 X 。

■ 关键字的定义

设 R 为任一给定关系, U 为其所含的全部属性集合, X 为 U 的子集, 若有完全函数依赖 $X \rightarrow U$, 则 X 为 R 的一个候选关键字。

2.3.3 范式与关系规范化过程 ★★★★★

● 第一范式

设 R 为任一给定关系 若果 R 中每个列与行的交点处的取值都是不可再分的基本元素, 则 R 为第一范式。

● 第二范式

设 R 为任一给定关系, 若 R 为 1NF, 且其所有非主属性都完全函数依赖于候选关键字, 则 R 为第二范式。

● 第三范式

设 R 为任一给定关系, 若 R 为 2NF, 且其每一个非主属性都不传递函数依赖于候选关键字, 则 R 为第三范式。

考题看一看

假设某商业集团数据库中关系模式 R 如下:

R (商店编号, 商品编号, 库存数量, 部门编号, 负责人)

如果规定:

- (1) 每个商店的每种商品只在一个部门销售;
- (2) 每个商店的每个部门只有一个负责人;
- (3) 每个商店的每种商品只有一个库存数量。

试回答下列问题:

1. 根据上述规定, 写出模式 R 的关键码。

答案: 商店编号, 商品编号

解析: 定义 2.5 设 R 为任一给定关系, U 为其所含的全部属性集合, X 为 U 的子集, 若有完全函数依赖 $X \rightarrow U$, 则 X 为 R 的一个候选关键字。

作为候选关键字的属性集 X 唯一标识 R 中的元组, 但该属性集的任何真子集不能唯一标识 R 中的元组。显然, 一个关系 R 中可能存在多个候选关键字, 通常选择其中之一作为主关键字, 即主键。

R 的函数依赖集 = {(商店编号, 商品编号) \rightarrow 部门编号, (商店编号, 部门编号) \rightarrow 负责人, (商店编号, 商品编号) \rightarrow 库存数量}, 故关键码为 (商店编号, 商品编号)。

2. R 最高属于第几范式? 为什么?

答案: 第 2 范式。非主属性为库存数量, 部门编号, 负责人, 它们对候选键都不是部分函数依赖; 但是, 负责人对 (商店编号, 商品编号) 是传递函数依赖, 所以不满足 3NF。

解析: 定义 2.7 设 R 为任一给定关系, 若 R 为 1NF, 且其所有非主属性都完全函数依赖于候选关键字, 则 R 为第二范式 (2NF)。

定义 2.8 设 R 为任一给定关系，若 R 为 2NF，且其每一个非主属性都不传递函数依赖于候选关键字，则 R 为第三范式（3NF）。

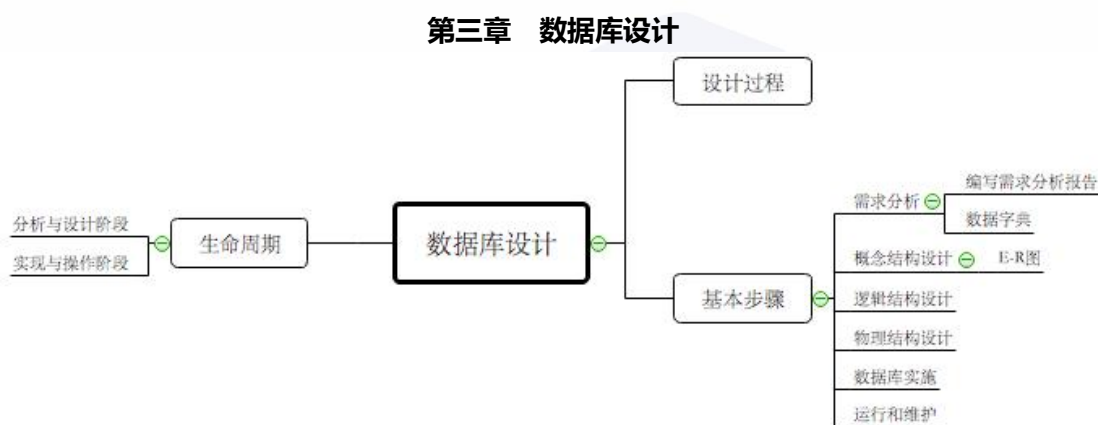
因为非主属性为库存数量，部门编号，负责人，它们对候选键都不是部分函数依赖，满足 2NF；但是，负责人对（商店编号，商品编号）是传递函数依赖，所以不满足 3NF。

3.将 R 规范到 3NF。

答案： R_1 （商店编号，商品编号，部门编号，库存数量）， R_2 （商店编号，部门编号，负责人）

解析：定义 2.8 设 R 为任一给定关系，若 R 为 2NF，且其每一个非主属性都不传递函数依赖于候选关键字，则 R 为第三范式（3NF）。

R 中存在非主属性“负责人”对主属性（商店编号，商品编号）的传递函数依赖。即（商店编号，商品编号） \rightarrow 部门编号，（商店编号，部门编号） \rightarrow 负责人，但不存在部门编号 \rightarrow （商店编号，商品编号）。故将 2NF 关系 R 进一步规范为 3NF： R_1 （商店编号，商品编号，部门编号，库存数量）， R_2 （商店编号，部门编号，负责人），消除非主属性对候选关键字的传递函数依赖。



3.1.1 数据库的生命周期（34-50）★★★

- 数据库的生命周期可分为两个阶段：数据库分析与设计阶段、数据库实现与操作阶段。
- 数据库分析与设计阶段包括：需求分析、概念设计、逻辑设计和物理设计。
- 数据库实现与操作阶段包括：数据库的实现、操作与监督、修改与调整三个子阶段。

3.1.5 数据库设计的过程★★★

- 数据库设计的阶段包括：

- 1.需求分析阶段；
- 2.结构设计阶段:包括概念结构设计、逻辑结构设计和物理结构设计；
- 3.行为设计阶段：包括功能设计、事务设计和程序设计；
- 4.数据库实施阶段：包括加载数据库数据和调试运行应用程序；
- 5.数据库运行和维护阶段。

3.2.0 数据库设计的基本步骤★★★

- 数据库设计的基本步骤：需求分析、概念结构设计、逻辑结构设计、物理结构设计、数据库实施、数据库的运行和维护。

3.2.1.0 需求分析 ★★

- 需求分析是数据库设计的起点；

3.2.1.4 编写需求分析报告 ★

- 作为需求分析的一个总结，数据库设计人员需要编写需求分析报告。
- 数据字典是数据库系统中存储三级结构定义的数据库，通常指的是数据库系统中各类数据详细描述的组合。

3.2.2 概念结构设计 ★★

- 关系数据库的设计-自顶向下法，采用 E-R 图作为概念模型描述的工具。

3.2.5 数据库实施 ★★

- 数据库实施阶段需要完成的工作包括：加载数据、应用程序设计和数据库试运行。

3.3.2.1 E-R 图的表示方法 ★★★★★

- 概念结构设计使用 E-R 图来作为描述现实世界的建模工具。

1. 实体型，用矩形表示，写明实体的名称；
2. 属性，用椭圆形表示，并用无向边将其与其相应的实体连接起来；
3. 联系，用菱形表示，写明联系名称，无向边连接。两个实体型之间存在如下关系：
一对一联系（1:1）
一对多联系（1:N）
多对多联系（M:N）

考题看一看

为体育部门建立数据库，其中包含如下信息：

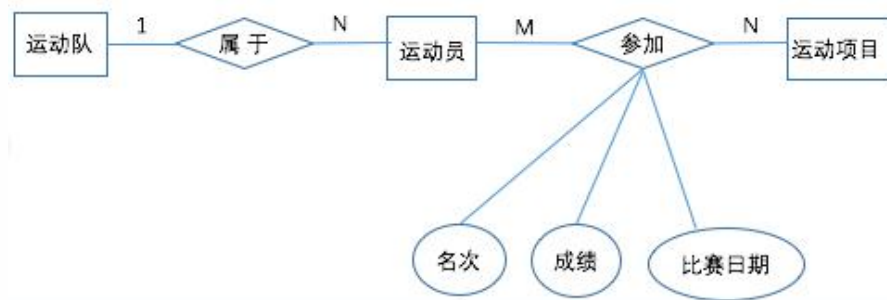
- (1) 运动队：队名、主教练，其中队名惟一标识运动队。
- (2) 运动员：运动员编号、姓名、性别、年龄。
- (3) 运动项目：项目编号、项目名、所属类别。

其中：每个运动队有多名运动员，每名运动员只属于一个运动队；每名运动员可以参加多个项目，每个项目可以有多个运动员参加。系统记录每名运动员参加每个项目所得名次和成绩以及比赛日期。

根据以上叙述，建立 ER 模型，要求标注联系类型。（实体的属性可以省略）

答案：

根据题意绘图如下：



解析：

概念结构设计就是将需求分析得到的用户需求抽象为信息结构的过程，通常使用 E-R 图来

作为描述现实世界的建模工具。E-R 图提供了表示信息世界中实体、属性和联系的方法。

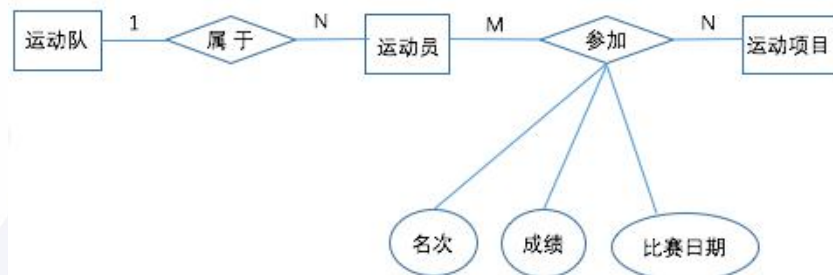
1. 实体型，用矩形表示，写明实体的名称；
2. 属性，用椭圆形表示，并用无向边将其与其相应的实体连接起来；
3. 联系，用菱形表示，写明联系的名称，用无向边分别与有关实体连接起来，同时，在无向边旁标注联系类型（1:1、1:N 或 M:N），如果一个联系具有属性，则这些属性也要用无向边与该联系连接起来。

3.3.3.1 E-R 图向关系模型的转换 ★★★★★

● 遵循原则如下：

1. 一个实体型转换为一个关系模式。实体的属性作为关系的属性，实体的码作为关系的码；
2. 一个一对一联系可以转换为一个独立的关系模式，也可以与任意一端对应的关系模式合并；
3. 一个一对多联系可以转换为一个独立的关系模式，也可以与 N 端对应的关系模式合并；
4. 一个多对多联系转换为一个关系模式。与该联系相连的各实体的码以及联系本身的属性均转换为关系的属性，而关系的码为各实体码的组合；
5. 三个或三个以上实体间的一个多元联系可以转换为一个关系模式；
6. 具有相同码的关系模式可合并。

考题看一看



根据转换规则，将上面 ER 模型转换成关系模型，要求标明每个关系模式的主键和外键（如果存在）。

答案：

运动队（队名，主教练）

运动员（运动员编号，姓名，性别，年龄，队名）

运动项目（项目编号，项目名，所属类别）

参加（运动员编号，项目编号，名次，成绩，比赛日期）

标下划线为主键，加粗斜体为外键（注意区分）

解析：

在一个关系的若干个候选码或候选键中指定一个用来唯一标识关系的元组，则称这个被指定的候选码或候选键为该关系的主码或主键。

当关系中的某个属性（或属性组）不是这个关系的主码或候选码，而是另一个关系的主码时，称该属性（或属性组）为这个关系的外码或外键。

3.3.4 物理设计方法

物理设计的任务主要是通过对关系**建立索引**和**聚集**来实现与应用相关数据的逻辑连接和物理聚集，以改善对数据库的存取效率。

(1) 建立索引的方式通常有静态和动态两种。

(2) 聚集是将相关数据集中存放的物理存储技术，借以提高 I/O 的数据命中率而改善存取速度。数据聚集结构的一种有效方式是块结构方式

第四章 SQL 与关系数据库基本操作



4.1.3.0 SQL 的组成 ★★★

- SQL 四大功能：**数据查询**、**数据定义**、**数据操纵**和**数据控制**。
- 两种构架方式：★

1.**LAMP** ; 2.**WAMP** ;

4.1.3.3 数据控制语言★

数据控制语言主要用于安全管理。主要 SQL 语句如下：

- 1.GRANT：用于授权，把语句许可或对象许可的权限授予其他用户和角色。
- 2.REVOKE：用于收回权限。

4.3.0 数据定义★★

操作对象	操作方式		
	创建	删除	修改
模式	CREATE SCHEMA 语句	DROP SCHEMA 语句	
表	CREATE TABLE 语句	DROP TABLE 语句	ALTER TABLE 语句
视图	CREATE VIEW 语句	DROP VIEW 语句	

4.3.1 数据库模式定义 ★★★★★

- **选择数据库**：USE db_name;
- **修改数据库**：ALTER DATABASE
- **删除数据库**：DROP DATABASE
- **查看数据库**：SHOW DATABASES;

考题看一看

在 MySQL 中创建一个名为 mysql_test 的数据库，请写出相应的 SQL 语句。

答案：

mysql>CREATE DATABASE mysql_test;

解析：

```
mysql>CREATE DATABASE mysql_test;(牢记创建数据库的语法格式)
```

4.3.2.0 表定义★★★

MySQL 中和表定义相关的 SQL 语句：

- 1.创建表：在 MySQL 中，可以使用 CREATE TABLE 语句创建表。
- 2.更新表：在 MySQL 中，可以使用 ALTER TABLE 语句来更改原有表的结构。
- 3.重命名表：除了 ALTER TABLE 语句，还可以直接用语句 RENAME TABLE 来更改表名，并可同时命名多个表。
- 4.删除表：如若需要删除数据库中已存在的表，可以通过使用 DROP TABLE 语句来实现。
- 5.查看表：（1）显示表的名称：在 MySQL 中，可以使用 SHOW TABIES 语句来显示指定数据库中存放的所有表名；（2）显示表的结构：在 MySQL 中，可以使用 SHOW COLUMNS 语句来显示指定数据表的结构

4.3.2.1 创建表 ★★★★★

- 创建表：CREATE TABLE 语句
- 主要的数据类型包括：数值类型、日期和时间类型、字符串类型、空间数据类型等。
- NULL 值是指没有值或缺值。不能将 NULL 值与空串相混淆，NULL 值是没有值，它不是空串。
- 主键是通过 PRIMARY KEY 关键字来指定的。
- 主键值必须唯一，而且主键一定要为 NOT NULL。

4.3.2.2 更新表 ★★★★★

- ALTER TABLE 语句；

考题看一看

在一个已有数据库 mysql_test 中新建一个包含客户姓名、性别、地址、联系方式等内容的客户基本信息表，要求将客户的 id 号指定为该表的主键，写出相应 SQL 语句。

答案：

SQL 语句如下：

```
mysql>USE mysql_test;
Database changed
mysql>CREATE TABLE customers
->(
-> cust_id INT NOT NULL AUTO_INCREMENT,
-> cust_name CHAR(1)NOT NULL,
-> cust_sex CHAR(1) NOT NULL DEFAULT 0,
-> cust_address CHAR(50) NULL,
-> cust_contact CHAR(50) NULL,
-> PRIMARY KEY(cust_id)
-> );
```


4.3.3.0 索引定义 ★★

- 索引在逻辑上通常包含以下几类：

- 1.普通索引 (INDEX)：最基本的索引类型，没有任何限制。
- 2.唯一性索引 (UNIQUE)：索引列中的所有值都只能出现一次，必须是唯一的。
- 3.主键 (PRIMARY KEY)：是一种唯一性索引。

4.4.1 插入数据 ★★

- INSERT 语句有三种语法形式，分别对应的是 INSERT...VALUES 语句、INSERT...SET 语句和 INSERT...SELECT 语句。

4.4.1.1 使用 INSERT...VALUES 语句插入单行或多行元组数据 ★★

考题看一看

设某数据库中一个表 S 的结构为 S(SNAME, CNAME, GRADE)，其中 SNAME 为学生名，CNAME 为课程名，二者均为字符型，GRADE 为成绩，数值型，取值范围是 0 — 100，若要把“张一的化学成绩 80 分”插入 S 中，则下列语句正确的是 ()

- A:ADD INTO S VALUES('张一', '化学', '80')
- B:INSERT INTO S VALUES('张一', '化学', '80')
- C:ADD INTO S VALUES('张一', '化学', 80)
- D:INSERT INTO S VALUES('张一', '化学', 80)

答案：D

解析：

使用 INSERT...VALUES 语句插入单行或多行元组数据的语法格式如下：

INSERT [INTO] tbl_name [(col_name,...)] {VALUES|VALUE} ({expr|DEFAULT},...),(...),...

其中“expr”，表示一个常量、变量或一个表达式，也可以是空值 NULL，其值的数据类型要与列的数据类型一致，当列值为字符型时，需要用单引号扩起，成绩为数值格式，不用加引号，所以选 D。

4.4.3 修改数据 ★★★★★

考题看一看

设有学生关系 S(学号，姓名，性别，奖学金)，选课关系 SC(学号，课号，成绩)，用 SQL 语句完成如下操作：对成绩得过满分(100)的学生，如果没有得过奖学金(NULL 值)，将其奖学金设为 1000 元。

答案：

```
UPDATE S
SET 奖学金=1000
WHERE 奖学金 IS NULL AND 学号 IN
    (SELECT 学号
     FROM SC
     WHERE 成绩=100);
```

解析：

在 SC 表查询出成绩为 100 的学号作为更新 S 表的其中一个条件，需注意 NULL 表示“没有值”或“无数据”，对 NULL 进行条件判断需要用 IS 和 NOT IS。

4.5.1 SELECT 语句 ★★★★★

常用语法格式如下：

```
SELECT
  [ALL | DISTINCT | DISTINCTROW ]
  select_expr [, select_expr ...]
  FROM table_references
  [WHERE where_condition]
  [GROUP BY {col_name | expr | position}
   [ASC | DESC], ... [WITH ROLLUP]]
  [HAVING where_condition]
  [ORDER BY {col_name | expr | position}
   [ASC | DESC], ...]
  [LIMIT {[offset,] row_count | row_count OFFSET offset}]
```

4.5.3.0 FROM 子句与多表连接查询 ★★★★★

- 若一个查询同时涉及两个或两个以上的表，则称之为多表连接查询，也称多表查询或连接查询。
- 其连接方式主要包括交叉连接、内连接和外连接。

4.5.3.1 交叉连接 ★★★★★

又称笛卡尔积，在 FROM 子句中，也可以省略关键字 CROSS JOIN，而使用逗号分隔交叉连接的两张表。

4.5.4.0 WHERE 子句与条件查询 ★★★★★

- 使用 WHERE 子句指定过滤条件。

4.5.4.2 判定范围 ★★★★★

- 关键字是“BETWEEN”和“IN”两个。

1.BETWEEN...AND

2.IN

4.5.5 GROUP BY 子句与分组数据 ★★★★★

- 允许使用 GROUP BY 子句，将结果集中的数据行根据选择列的值进行逻辑分组，以便能汇总表内容的子集，即实现对每个组的聚集计算。

4.5.7 ORDER BY 子句 ★★★★★

- 在 SELECT 语句中，可以使用 ORDER BY 子句将结果集中的数据行按一定的顺序进行排列。

4.5.8 LIMIT 子句★

在 MySQL 中，可以使用 LIMIT 来限制 SELECT 语句所返回的行数。

4.6.0 视图

使用视图的优点：1) 集中分散数据；2) 简化查询语句；3) 重用 SQL 语句；4) 保护数据安全；5) 共享所需数据；6) 更改数据格式

4.6.1 创建视图 ★★★★★

- 用 CREATE VIEW 语句来创建视图

考题看一看

设一个图书借阅管理数据库中包括三个关系模式：

图书(图书编号，书名，作者，出版社，单价)

读者(借书证号，姓名，性别，单位，地址)

借阅(借书证号，图书编号，借阅日期，归还日期，备注)

用 SQL 语句完成下面 1-4 题。

1.查询价格在 50 到 60 元之间的图书，结果按出版社及单价升序排列。

答案：

```
SELECT *
```

```
FROM 图书
```

```
WHERE 单价 BETWEEN 50 AND 60
```

```
ORDER BY 出版社，单价；
```

2.查询王明所借阅的所有图书的书名及借阅日期。

答案：

```
SELECT 书名,借阅日期
```

```
FROM 图书,借阅,读者
```

```
WHERE 姓名 = '王明' AND 读者.借书证号 = 借阅.借书证号 AND 借阅.图书编号 = 图书.图书编号;
```

3.查询各个出版社图书的最高价格、最低价格和平均价格。

答案： SELECT 出版社,MAX(单价),MIN(单价),AVG(单价)

```
FROM 图书
```

```
GROUP BY 出版社;
```

4.建立“红星汽车厂”读者的视图 RST。

答案： CREATE VIEW RST(借书证号,姓名,性别,单位,地址)

```
AS SELECT *
```

```
FROM 读者
```

```
WHERE 单位 = '红星汽车厂' ;
```

第五章 数据库编程

增强SQL语言的功能和灵活性

良好封装性

高性能

减少网络流量

确保安全性和完整性

使用存储过程的好处

数据库编程

创建存储过程

DELIMITER (定义的结束符)

CREATE PROCEDURE

5.1.1 存储过程的基本概念 ★★★

● 使用存储过程通常具有以下好处：

- (1) 可增强 SQL 语言的功能和灵活性；
- (2) 良好的封装性；
- (3) 高性能；
- (4) 可减少网络流量；

(5) 存储过程可作为一种安全机制来确保数据库的安全性和数据的完整性。

5.1.2 创建存储过程 ★

- DELIMITER 命令的使用语法格式是：**DELIMITER \$\$**
- CREATE **PROCEDURE** 语句来创建存储过程。

5.1.3.5 游标★

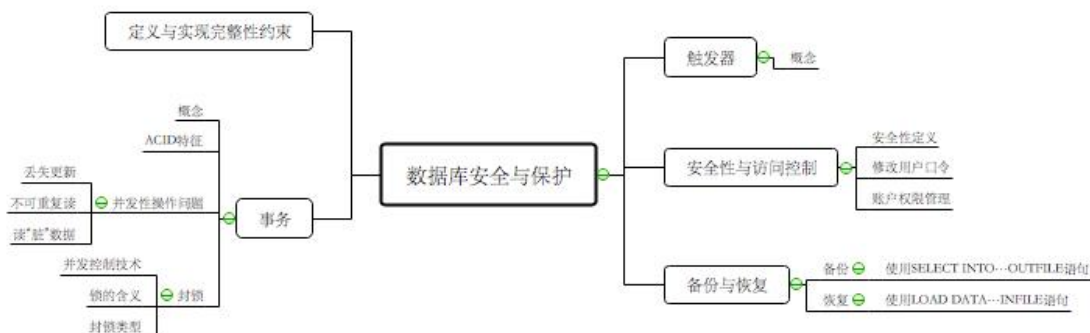
- (1) 可以使用 DECLARE CURSOR 语句创建游标。
- (2) 可以使用 OPEN 语句打开游标。
- (3) 可以使用 FETCH...INTO 语句从中读取数据。
- (4) 可以使用 CLOSE 语句关闭游标

5.2.0 存储函数 ★★

存储函数和存储过程的区别：

- 1、存储函数不能拥有输出参数，这是因为存储函数自身就是输出参数；而存储过程可以拥有输出参数。
- 2、可以直接对存储函数进行调用，且不需要使用 CALL 语句；而对存储过程的调用，需要使用 CALL 语句。
- 3、存储函数中必须包含一条 RETURN 语句，而这条特殊的 SQL 语句不允许包含于存储过程中

第六章 数据库安全与保护



6.1.2.0 定义与实现完整性约束 ★★

- 关系模型中可以有**三类完整性约束**，分别是**实体完整性**、**参照完整性**和**用户定义的完整性**。
- 实体完整性中**主键的值**，也称为**键值**，必须能够唯一标志表中的每一行记录，且**不能为 NULL**。

6.1.2.2 参照完整性★★★

关键字“CASCADE”表示级联策略，即从被参照表中删除或更新记录行时，自动删除或更新参照表中匹配的记录行；关键字“SET NULL”表示置空策略，即当从被参照表中删除或更新记录行时，设置参照表中与之对应的外键列的值为 NULL，这个策略需要被参照表中的外键列没有声明限定词 NOT NULL。

6.2.0 触发器 ★

- **触发器**是用户定义在关系表上的一类由事件驱动的数据库对象,也是一种保证数据完整性的方法。

6.3.0 安全性与访问控制 ★

- 数据库的**安全性**是指保护数据库以防止不合法的使用而造成数据泄露、更改或破坏,所以安全性对于任何一个 DBMS 来说都是至关重要的。

6.3.1.4 修改用户口令 ★

可以使用 **SET PASSWORD** 语句修改一个用户的登录口令。

6.3.2 账户权限管理 ★

- 新建的 MySQL 用户必须被授权,可以使用 **GRANT** 语句来实现。
- 最有效率的权限是**用户**权限。
- 当需要回收某些特定的权限时,可使用 **REVOKE** 语句。

6.3.2.2 权限的转移★★

权限的转移可以通过 GRANT 语句中使用 WITH 子句来实现。如果将 WITH 子句指定为关键字“WITH GRANT OPTION”,则表示 TO 子句中所指定的所有用户都具有把自己所拥有的权限授予给其他用户的权利,而无论那些其他用户是否拥有该权限。

6.4.1 事务的概念 ★★

- 定义事务的语句一般有三条:**BEGIN TRANSACTION**、**COMMIT** 和 **ROLLBACK**,且事务通常是以 BEGIN TRANSACTION 语句开始,以 COMMIT 语句或 ROLLBACK 语句结束。

6.4.2 事务的特征 ★★★

● 事务的 ACID 特征:

- 1、**原子性**:事务是不可分割的最小工作单位。
- 2、**一致性**:满足数据库的完整性约束。
- 3、**隔离性**:一个事务的执行不能被其他事务所干扰,一个事务对数据库变更的结果必须要在它 COMMIT 后,另一个事务才能存取。
- 4、**持续性**:其他操作或故障不应该对其执行结果有任何影响。

6.4.3 并发操作问题 ★★

- 典型的并发操作问题有三种:**丢失更新**、**不可重复读**和**读“脏”数据**。

1、丢失更新

设有两个事务 T1 和 T2,当它们同时读入同一数据并加以修改时,事务 T2 的提交结果会破坏事务 T1 提交的结果,由此导致事务 T1 的修改被丢失。

2、不可重复读

不可重复读包括三种情况:

- (1) 事务 T1 读取某一数据后,事务 T2 对其做了修改,当事务 T1 再次读该数据时,得到与前一次不同的值。
- (2) 事务 T1 按一定条件从数据库中读取了某些数据记录后,事务 T2 删除了其中部分记

录，当事务 T1 再次按相同条件读取数据时，发现某些记录神秘地消失了。

- (3) 事务 T1 按一定条件从数据库中读取某些数据记录后，事务 T2 插入了一些记录，当事务 T1 再次按相同条件读取数据时，发现多了一些记录。

3、读“脏”数据

读“脏”数据是指，事务 T1 修改某一数据，并将其写回磁盘，事务 T2 读取同一数据后，事务 T1 由于某种原因被撤销，这时事务 T1 已修改过的数据恢复原值，事务 T2 读到的数据就与数据库中的数据不一致，则事务 T2 读到的数据就为“脏”数据，即不正确的数据。

- 产生上述三类数据不一致的主要原因是并发操作破坏了事务的隔离性。

6.4.4 封锁 ★★

封锁是最常用的并发控制技术，它的基本思想是：需要时，事务通过向系统请求对它所希望的数据对象加锁，以确保它不被非预期改变。

6.4.4.1 锁 ★

- 一个锁实际上就是允许或组织一个事务对一个数据对象的存取特权。
- 基本的封锁类型有两种：排他锁（X 锁）和共享锁（S 锁）。
- 写操作要求 X 锁，读操作要求 S 锁。

6.4.4.2 用封锁进行并发控制（43-50）★★

- 利用封锁机制可以解决并发操作所带来的不一致问题。

6.4.4.4 封锁的级别★★

3 级封锁：被封锁的事务不读未提交的更新数据，不写任何（包括读操作的）未提交的数据。显然，这除了包含 2 级封锁外，还不写未提交的读数据，因而防止了不可重读的问题。这是严格的封锁，它保证了多个事务并发执行的“可串行化”。

6.4.4.7 两段封锁法★★

定理：遵循两段锁协议的事务的任何并发调度都是可串行化的。

6.5.0 备份与恢复 ★★★

- 数据库备份是指通过导出数据或者复制表文件的方式来制作数据库的副本；
- 数据库恢复则是当数据库出现故障或遭到破坏时，将备份的数据库加载到系统，从而使数据库从错误状态恢复到备份时的正确状态。
- 使用 SELECT INTO...OUTFILE 语句备份数据
- 使用 LOAD DATA...INFILE 语句恢复数据

第七章 数据库应用设计与开发实例

7.2.2.3 全局信息结构★

在构建出局部信息结构的基础上，通过采用逐步合并、进行累加的方式，以及消除可能存在的属性冲突、命名冲突和结构冲突，最终形成一个本系统的全局信息结构。同一联系的问题属于结构冲突。

（备注：本章节属于贯穿整本书的开发实例，没有明确的知识点考点，根据大纲要求，附录部分的 PHP 例题可作为参考押题，挂靠本章节）

考题看一看

- 编写一个检索数据的 PHP 示例程序 select.php，要求在数据库 db_xuanke 的表 teacher 中查询职工号为 10021 的教师的姓名。

答案：

```
<?php
    $con=mysql_connect("localhost","root","123456") or die("数据库服务器连接失败！
<br>");
    mysql_select_db("db_xuanke",$con) or die("数据库选择失败！<br>");
    mysql_query("set names 'gbk'"); //设置中文字符集
    $sql="SELECT TeachName FROM teacher";
    $sql=$sql."WHERE TeachNo=10021";
    $result=mysql_query($sql,$con);
    if($result)
    {
        echo "查询成功！<br>";
        $array=mysql_fetch_array($result,MYSQL_NUM);
        if($array)
        {
            echo "所要查询教师的姓名是：".$array[0];
        }
    }
    else
        echo "查询失败！<br>";
?>
```

解析：

首先，在文本编辑器（例如，记事本）中输入如下 PHP 程序，并命名为 select.php（注意，PHP 程序是被包含在标记符 “<?php” 与 “?>” 之间的代码段，同时 PHP 程序中的变量名是以 “\$” 开头）

然后，将程序 select.php 部署在已开启的 WAMP 平台环境中，并在浏览器地址栏输入 <http://localhost/select.php> 即可查看程序执行结果。

第八章 数据管理技术的发展



8.1.0 数据库技术发展概述 ★★★

- 数据库技术三个发展阶段，即第一代的网状、层次数据库系统，第二代的关系数据库系统，以及新一代的数据库系统。

8.2.1 从数据库到数据仓库 ★★★

- 数据仓库的**特征**。
(1) 面向主题；(2) 集成性；(3) 数据的非易失性；(4) 数据的时变性。

8.2.2 数据挖掘技术 ★★★

- **数据挖掘**是从大量的、不完全的、有噪声的、模糊的、随机的实际应用数据中发现并提取隐藏在其中的、人们事先不知道的、但又是潜在有用的信息和知识的一种技术。
- **数据挖掘**具备的**功能**。
(1) 概念描述；(2) 关联分析；(3) 分类与预测；(4) 聚类；(5) 孤立点检测；(6) 趋势和演变分析。

8.3.1 大数据的定义 ★★★

- 大数据的**特征**：
(1) 数据量巨大，即大量化。(2) 数据种类繁多，即多样化。(3) 处理速度快，即快速化。(4) 价值密度低。

8.3.2 大数据管理技术典型代表 ★★★

典型代表：**分布式文件系统**、**Key/Value 非关系数据**模型和 **MapReduce 并行编程**模型。

8.3.2.2 NoSQL 数据管理系统★★★

NoSQL 系统支持的数据存储模型：

- (1) 键值 (Key-Value) 存储。NoSQL 数据库采用最多的数据存储方式。适合通过主键进行查询或遍历。
- (2) 文档存储。适合存储系统日志等非结构化数据。可以通过复杂的查询条件来获取数据。
- (3) 列存储。比较适合对某一列进行随机查询处理。主要应用于需要处理大量数据的情况。
- (4) 图存储。图存储数据库是基于图理论构建的，使用结点、属性和边的概念