

数据库复习笔记

1. 什么是数据，数据库，数据库管理系统？数据库系统包括哪几部分？

数据（DATA）是描述事物的符号记录，是数据库中存储的基本对象。

数据库（DB）是长期储存在计算机内、有结构的、可共享的数据集合。它不仅包括描述事物的数据本身，还包括相关事物之间的联系。数据库内数据按一定的数据模型组织、描述和储存，可为各种用户共享，冗余度较小，独立性较高，并且易扩展

数据库系统（DBS）是指在计算机系统中引入数据库后的系统，通常由软件、数据库和数据管理人员组成。其软件主要包括操作系统、各种宿主语言、实用程序以及数据库管理系统。

2. 数据库技术经过了哪三个阶段？数据库管理阶段有什么优点？

数据库技术经历了人工管理阶段、文件系统阶段和数据库系统阶段三个阶段。

数据库系统有以下优点：

- 1) 数据结构化：以数据为中心组织数据，形成可共享的综合性数据库。数据库中的数据不再针对某一应用，而是面向整个应用系统，使用复杂的数据模型表示结构。

- 2) 较高的数据共享性：允许多个用户同时存取数据而互不影响，是数据库先进性的体现，数据可以被多个用户、多个应用共享使用，数据共享可以大大减少数据冗余，节约系统空间。
- 3) 较高的数据独立性：程序和数据具有较高的独立性，应用程序不随数据存储结构的变化而变化，简化了应用程序的编制和程序员的工作负担。
- 4) 数据由 DBMS 统一管理和控制：DBMS 提供了完整的数据管理与控制功能（并发、完整性、可恢复性、安全性），并为用户提供丰富的方便。

3. 什么是数据库体系结构？

a) 三级模式：

- 1) 外模式（子模式、用户模式）：是对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。外模式通常是模式的子集，一个数据库可以有多个外模式。

外模式优点：为应用程序提供了逻辑独立性； 可以更好地适应不同用户对数据的需求； 为用户划定了访问数据的范围，有利于数据的保密等。

- 2) 模式（概念模式、逻辑模式）：是对数据库中全部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，一个数据库只有一个模式。主要描述数据的概念记录类型和它们之间的关系，还包括一些数据间的语义约束。

- 3) 内模式（存储模式、物理模式）：是对数据物理结构和存储方式的描述，是数据在数据库内部的表示方式，如数据存储的文件结构、索引、集簇等存取方式和存取路径等，一个数据库只有一个内模式。

概念模式给出系统全部的数据描述，外模式则给出每个用户的部分描述，即外模式为用户所见到的概念模式的一个部分，内模式则考虑数据的实际物理存储过程和组织方式。数据库的三级模式结构把数据的具体组织（内模式）留给 DBMS 去做，用户只要抽象地处理数据（模式——DBA、外模式——程序员），减轻了用户使用系统的负担。

b) 二级映像

三级模式结构之间差别往往很大，为了实现这 3 个抽象级别的联系和转换，DBMS 在三级模式结构之间提供了两级映像

- 1) 外模式/模式映像：对于每个外模式，数据库系统都有一个外模式/模式映像，它定义了该外模式与模式之间的对应关系。当模式改变时，由数据库管理员对各个外模式/模式映像作相应改变，可以使外模式保持不变，应用程序也不必修改。
- 2) 模式/内模式映像：数据库中只有一个模式，也只有一个内模式，所以模式/内模式映像是唯一的，它定义了数据库全局逻辑结构与存储结构之间的对应关系。当数据库的存储

结构改变时，由数据库管理员对模式/内模式映像作相应改变，可以保证模式保持不变，因而应用程序也不必改变。

三级模式是对数据的三个抽象级别，二级映像 DBMS 内部实现这三个抽象层次的联系和转换。其中，外模式/模式映像保证了数据与程序的逻辑独立性，简称逻辑数据独立性；模式/内模式映像保证了数据与程序的物理独立性，简称物理数据独立性。

4. 什么是数据库管理系统，有哪些功能？

数据库管理系统（DBMS）是数据库系统的核心，是为数据库的建立、使用和维护而配置的软件。它建立在操作系统的基础上，位于用户与操作系统之间的一层数据管理软件，它为用户或应用程序提供访问数据库的方法。

主要有四方面功能：

- 1) 数据定义和操纵功能：提供数据定义语言，对数据库中的对象进行定义和数据操纵语言，以检索、插入、修改、删除数据库中的数据。
- 2) 数据库运行控制功能：对数据库进行并发控制、安全性检查、完整性约束条件的检查和执行、数据库的内部维护等。
- 3) 数据库的组织、存储和管理：确定以何种文件结构和存取方式物理地组织数据，如何实现数据之间的联系，以便提高存储空间利用率和各种操作的时间效率。

- 4) 建立和维护数据库：初始数据的输入与数据转换，数据库的转储与恢复、数据库的重组与重构造、性能的监视与分析等。

5. 什么是概念模型，属性，实体，码

- 1) 概念模型：是现实世界的抽象反应，表示实体类型及实体间的联系，是独立于计算机系统的模型，是现实世界到机器世界的一个中间层。
- 2) 实体：客观存在并可以相互区分的事物。可以是具体的人、事、物或抽象的概念或联系。
- 3) 属性：实体所具有的某些特性。实体是由属性组成的，通过属性对实体进行描述。
- 4) 码：一个实体往往有多个属性，它们构成该实体的属性集合。如果其中有一个属性或属性集能够唯一标识整个属性集合，则称该属性或属性集为该实体的码。

6. ER 模型图是什么，如何绘制？

- 1) 用矩形表示实体，在框内写上实体名。
- 2) 用椭圆形表示实体的属性，并用无向边把实体和属性连接起来。
- 3) 用菱形表示联系本身，菱形框内写明联系名，并用无向边分别与有关实体连接起来，同时无向边旁标上联系的类型。

7. 什么是数据模型？

机器世界又叫数据世界，是信息世界中的信息数据化后对应的产物。现实世界中的客观事物及其联系，在机器世界中用数据模型来描述。数据模型是对客观事物及联系的数据描述，是概念模型的数据化，即数据模型提供表示和组织数据的方法。它是按计算机系统的观点对数据建模，用于 DBMS 实现。

8. 数据模型三个基本要素分别指什么？

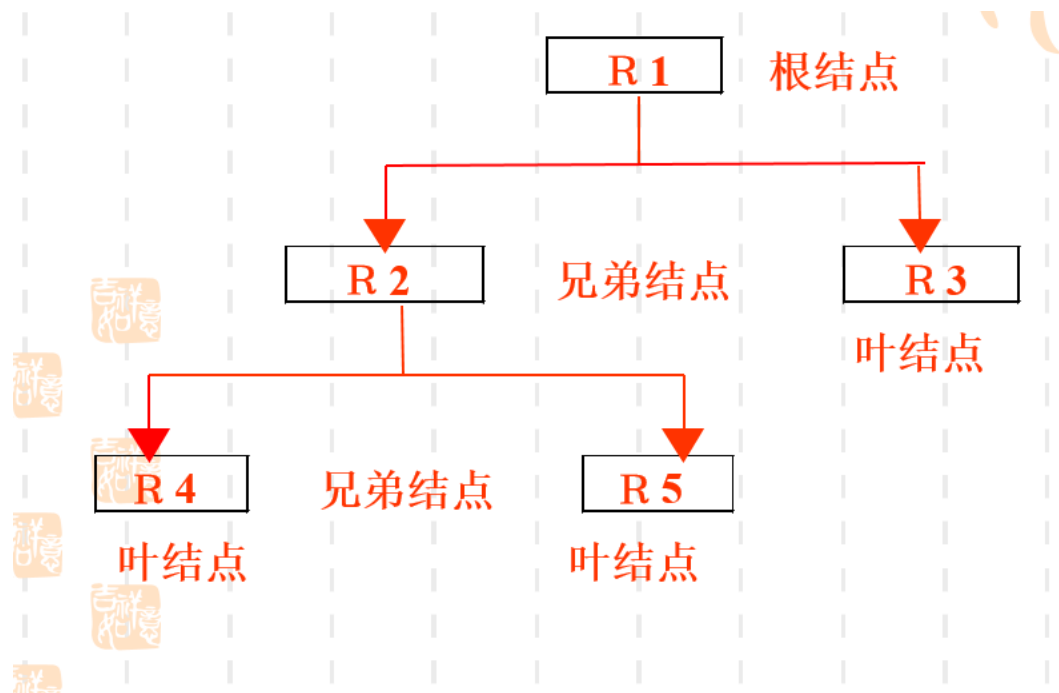
- 1) 数据结构：对计算机的数据组织方式和数据之间联系进行框架性描述的集合，是对数据库静态特征的描述。
- 2) 数据操作：指数据库中各记录允许执行的操作的集合，例如插入、删除、修改、检索、更新等操作，是对数据库动态特征的描述。
- 3) 数据的完整性约束：是关于数据状态和状态变化的一组完整性约束规则的集合，以保证数据的正确性、有效性和一致性。例如，数据库的主码不能允许取空值，性别的取值范围为“男或女”等。

9. 层次模型，网状模型分别指什么，特点是什么？

1) 层次模型：

满足下面两个条件的基本层次联系的集合为层次模型。

- a) 有且只有一个结点没有双亲结点，这个结点称为根结点；
- b) 根以外的其它结点有且只有一个双亲结点



特点如下：

- a) 结点的双亲是唯一的
- b) 只能直接处理一对多的实体联系（多对多关系分解）
- c) 每个记录类型定义一个排序字段，也称为码字段
- d) 任何记录值只有按其路径查看时，才能显出它的全部意义
- e) 没有一个子女记录值能够脱离双亲记录值而独立存在

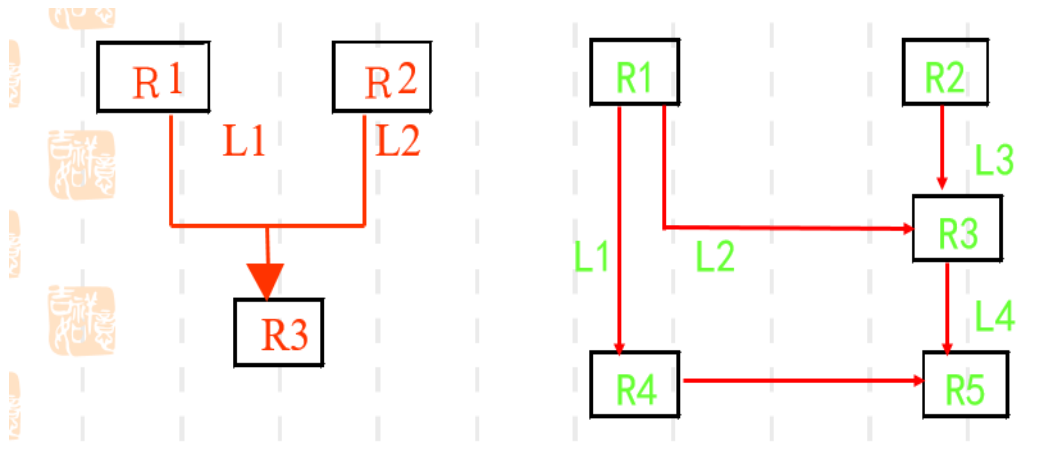
进行插入、删除、更新操作时要满足层次模型的完整性约束条件：

- a) 进行插入操作时，如果没有相应的双亲结点值就不能插入子女结点值。
- b) 进行删除操作时，如果删除双亲结点值，则相应的子女结点值也被同时删除。
- c) 进行更新操作时，应更新所有相应记录，以保证数据的一致性。

2) 网状模型:

满足下面两个条件的基本层次联系的集合为网状模型。

- a) 允许一个以上的结点无双亲;
- b) 一个结点可以有多于一个的双亲。



特点如下:

- a) 对网状模型，具体在计算机上实现时， $m:n$ 的联系仍需分解成若干个 $1:n$ 的联系。
- b) 插入操作允许插入尚未确定双亲结点值的子女结点值。
- c) 删除操作允许只删除双亲结点值。
- d) 更新操作只需要更新指定记录即可。
- e) 查询操作可以有多种方法，可根据具体情况选用。

10. 什么是关系，关系模型，关系模式，元组，属性，域，码，外码？

1) 关系：一个关系就是一张二维表，每个关系都有一个关系名，在计算机里，一个关系可以存储为一个文件

2) 关系模型：用二维表格结构表示实体以及实体之间的联系的数据模型称为关系模型。关系模型是数据模型中最重要的模型。它以关系数学为理论基础，在关系模型中，操作的对象和操作结果都是二维表。

3) 关系模式：关系数据库中，关系是值，关系模式是型，关系模式是对关系的描述。一个关系模式是一个五元组，表示为

$R(U, D, dom, F)$ 其中 R 是关系名， U 是属性名集合， D 是 U 中属性所来自的域， dom 是属性向域的映像集合， F 是属性间的数据依赖关系集合。

4) 元组：二维表中的行称为元组，每一行是一个元组。元组对应存储文件中的一个记录。

5) 属性：二维表的列称为属性，每一列有一个属性名，属性值是属性的具体值。属性对应存储文件中的一个字段，属性的具体取值就形成表中的一个元组。

6) 域：域是属性的取值范围。

7) 码：一个实体往往有多个属性，它们构成该实体的属性集合。

如果其中有一个属性或属性集能够唯一标识整个属性集合，则称该属性或属性集为该实体的码。

8) 外码：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 相对应，则称 F 是基本关系 R 的外码 (Foreign key)，并称基本关系 R 为参照关系，基本关系 S 为被参照关系或目标关系。

11. 关系有哪些性质？

1) 列是同质的：每一列的分量是同一类型的数据，来自于同一个域。

2) 在同一个关系中，不同的列的数据可以是同一种数据类型，但个属性的名称必须是互不相同的。

3) 同一关系中，任意两个元组不能完全相同。

4) 在同一关系中，列的次序无关紧要。

5) 同一关系中，元组的顺序无关紧要，可以任意交换两行位置。

6) 关系中每个属性必须是单值，即关系的结构不能嵌套，这是关系应满足的最基本条件。

12. 实体完整性约束和参照完整性约束分别指什么？

- 1) 实体完整性约束：实体完整性规则要求关系的主码中属性不能取空值。
- 2) 参照完整性约束：若 F 是关系 R 的外键，它与基本关系 S 的主码 K 相对应，则对于 R 中的每个元组在 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 中某个某个关键字值。

13. 关系运算

运算类别	运算符	含 义	运算类别	运算符	含 义
集 合 运算符	\cup	并	逻 辑 运算符	\neg	非
	$-$	差		\wedge	与
	\cap	交		\vee	或
	\times	广义笛卡尔积	比 较 运算符	$>$	大于
关 系 运算符	σ	选择		\geq	大于等于
	Π	投影		$<$	小于
	\bowtie	连接		\leq	小于等于
	\div	除		$=$	等于
				\neq	不等于

关系运算的应用题见 PPT。

14. 关系模式规范化

- 1) 关系模式规范化目的：

关系模式规范化的目的是解决关系模式中存在的冗余、插入和删除异常以及更新异常等问题

- 2) 关系模式规范化：一个低一级范式的关系模式，通过模式分解可以转换为若干个高一级范式的关系模式的集合，这个过程称为规范化。通常转化到 3NF。
- 3) 第一范式 (1NF)：如果关系模式 R 的每一个属性都是不可分解的，则 $R \in 1NF$ 。
- 4) 第二范式 (2NF)：如果关系模式 R 是第一范式，且每个非主属性都完全函数依赖于码，则称 R 为满足第二范式的模式，记为： $R \in 2NF$ 。
- 5) 第三范式 (3NF)：如果关系模式 R 是第二范式，且没有一个非主属性是传递函数依赖于码，则称 R 为满足第三范式的模式，记为： $R \in 3NF$ 。

15. 数据库设计六大步骤

1) 需求分析：

需求分析就是数据库设计人员，通过仔细地调查和向用户详细地咨询，掌握用户的需求，理解用户的需求。

需求分析的任务：重点是“任务”和“处理”

- a) 信息(数据)要求：调查用户的信息要求，确定在数据库中需要存储哪些数据。
- b) 处理要求：指用户要完成什么处理功能，处理时间要求及处理方式。

- c) 安全性要求：安全性要求是指对数据库的用户、角色、权限、加密方法等安全保密措施的要求。
- d) 完整性要求：完整性要求是指对数据取值范围、数据之间各种联系的要求等等。

2) 概念结构设计

概念结构设计阶段不是直接将需求分析得到的数据转换为 DBMS 能处理的数据库模式，而是将需求分析得到的用户需求抽象为反映用户观点的概念模型（E-R 模型）。

概念结构的模型的特点：

- a) 有丰富的语义表达能力：表达用户的各种需求，反映数据及其联系。
- b) 易于交流和理解：系统分析师、数据库设计人员和用户交流、理解。
- c) 易于修改：灵活性、反映用户需求和环境的变化。
- d) 易于向各种数据模型转换。

常用方法：

- a) 自顶向下法：建立初步框架，即全局 E-R 模型，然后再逐步细化
- b) 自底向上法。先得到局部 E-R 模型，再进一步综合成全局 E-R 模型

c) 逐步扩张法。先定义最重要的核心概念 E-R 模型，然后向外扩充

d) 混合策略：a、b 混合使用

注：其中最常用的策略是自底向上法。

3) 逻辑结构设计

把数据库概念设计阶段得到的数据模式转换成某个具体的 DBMS 所支持的数据模式，并建立相应的外模式，这是数据库逻辑设计的任务，是数据库结构设计的重要阶段。

逻辑结构设计一般要分为三步：

a) 将 E-R 图转化为关系数据模型：实际上是要将实体、属性和实体之间的联系转化为关系模式。

b) 关系模式的优化：规范化（一般到第三范式。），调整结构（提高使用效率的优化）：合并与分解

c) 设计用户外模式(子模式)

4) 数据库物理设计

将逻辑设计中产生的数据库逻辑模型结合指定的 DBMS，设计出最适合应用环境的物理结构的过程，称为数据库的物理结构设计。

物理结构设计分为两个步骤：

- a) 确定数据库的物理结构。
- b) 对所设计的物理结构进行评价

5) 数据库实施

根据逻辑和物理设计的结果，在计算机上建立起实际的数据库结构，并装入数据，进行试运行和评价的过程，叫做数据库的实施（或实现）。

6) 数据库的运行和维护

数据库试运行结果符合设计目标后，数据库就可以真正投入运行了。数据库投入运行标志着开发任务的基本完成和维护工作的开始。

维护工作包括：

- a) 数据库的转储和恢复。
- b) 数据库的安全性和完整性控制。
- c) 数据库性能的监督、分析和改造 。
- d) 数据库的重组和重构造 。

16. ER 图转化为关系模型的规则

- 1) 对于 E-R 图中每个实体，都应转换为一个关系，该关系应包括对应实体的全部属性，并应根据关系所表达的语义确定哪个属

性(或哪几个属性组合)作为“主键”。键在关系模型中是实现联系的主要手段。

- 2) 对于 E-R 图中的联系,情况比较复杂,要根据实体联系方式的不同,采取不同的手段加以实现。

17. 物理数据库包括哪几种文件?

- 1) 主数据文件:用来存储数据库的启动信息以及部分或者全部数据,是所有数据库文件的起点,包含指向其它数据库文件的指针。一个数据库只能有一个主数据库文件。扩展名推荐为 .mdf。
- 2) 辅助数据文件:用于存储主数据库文件中未存储的剩余数据和数据库对象,一个数据库可以没有辅助数据库文件,但也可以同时拥有多个辅助数据库文件。扩展名为 .ndf。
- 3) 事务日志文件:存储数据库的更新情况等事务日志信息,当数据库损坏时,管理员使用事务日志恢复数据库。每一个数据库至少必须拥有一个事务日志文件,而且允许拥有多个日志文件。事务日志文件的扩展名为 ldf,日志文件的大小至少是 512KB。

注:一般地,将主数据文件和辅助数据文件统称为数据文件。

18. 逻辑数据库包含哪几种?

- 1) 系统数据库:
 - a) master 数据库:是 SQL Server 2008 中的主数据库,它是最重要的系统数据库,记录系统中所有系统级的信息。

- b) model 数据库: 为用户新创建的数据库提供模板, 当用户创建数据库时, 系统会自动地把 model 数据库中的内容复制到新建的用户数据库中。
- c) 主存 (msdb) 数据库: 数据库记录备份及还原的历史信息、维护计划信息、作业信息、异常信息以及操作者信息等。
- d) tempdb 数据库: 是系统提供的一个空间用来存储临时对象。比如, 用户修改表的某一行数据时, 往往修改后的数据构成临时表, 而这个临时表就被临时存储在该数据库中。

2) 用户自定义数据库

19. Create database 语句

见 PPT

20. 表的操作, create table 语句, 插入修改删除数据

见 PPT

21. 列的几种约束, 建立约束 SQL 语句

- 1) 主键约束: 用于在表中定义主键, 它是惟一确定表中每一条记录的字段集。每个表中只能有一列或一组列被指定为主键, 主键列的值不能为 Null 值, 并且主键所在列的组合值必须惟一。
- 2) UNIQUE 约束: 指定表中某一个列或多个列不能有相同的两行或两行以上的数据存在。当表中已经有了一个主键约束时, 如

果需要在其他列上实现实体完整性，就只能通过创建 UNIQUE 约束来实现。

- 3) CHECK 约束:通过指定的逻辑表达式来限制某列的取值范围。
- 4) DEFAULT 约束: 当使用 INSERT 语句插入数据时, 如果没有为某一个列指定数据, 那么 DEFAULT 约束就在该列中输入一个默认值。
- 5) 外键约束:定义一个或多个列, 这些列可以引用同一个表或另外一个表中的主键约束列或 UNIQUE 约束列。通过创建外键约束可以实现表和表之间的关联关系

22. SQL 查询语句

见 PPT

23. 视图的概念, 使用视图有什么优点?

视图(View,外模式) 是从一个或几个基本表(模式)导出来的表, 是一个虚表, 并不表示任何物理数据。数据库中只存储视图的定义, 而不存储视图对应的数据, 这些数据仍存储在导出视图的基本表中。

使用视图有以下优点:

- 1) 虚表, 是从一个或几个基本表(或视图)导出的表:

视图机制使用户可以将注意力集中在所关心的数据上。当视图中数据不是直接来自基本表时, 定义视图能够简化用户的操作

2) 视图对重构数据库提供了一定程度的逻辑独立性:

只存放视图的定义,不会出现数据冗余,基表中的数据发生变化,从视图中查询出的数据也随之改变

3) 视图使用户能以多种角度看待同一数据:

视图机制能使不同用户以不同方式看待同一数据,当许多不同种类的用户共享同一个数据库时,这种灵活性是非常重要的。适应数据库共享的需要。

4) 视图能够对机密数据提供安全保护:对不同用户定义不同视图,使每个用户只能看到他有权看到的数据。通过 WITH CHECK OPTION 对关键数据定义操作时间限制

24. 如何创建, 修改, 删除视图?

见 PPT

25. 什么是索引? 如何分类? 使用索引有什么优点?

索引通过记录表中的关键值指向表中的记录,不用扫描整个表而定位到相关的记录。

1) 索引可以分为:

a) 单列索引: 根据单列创建的索引。

b) 复合索引: 根据多列组合创建的索引。

c) 唯一索引: 索引列不会有两行记录相同。

d) 不唯一索引: 索引列上可以有多行记录相同。

- e) 聚集索引：聚集索引会对表中数据进行物理排序，所以这种索引对查询非常有效，在表和视图中只能有一个聚集索引。当建立主键约束时，如果表中没有聚集索引，SQL Server 会用主键列作为聚集索引键。
- f) 非聚集索引：非聚集索引不会对表中数据进行物理排序。如果表中不存在聚集索引，则表是未排序的。
- g) 注：SQL Server 中不管聚集还是非聚集索引都采用 B+树的存储结构。

2) 使用索引有如下优点：

- a) 大大加快数据的检索速度，这是创建索引的最主要的原因。
- b) 创建唯一性索引，保证表中每一行数据的唯一性。
- c) 加速表和表之间的连接。
- d) 在使用分组和排序子句进行数据检索时，同样可以显著减少查询中分组和排序的时间。
- e) 查询优化器可以提高系统的性能，但它是依靠索引起作用的。

26. 索引的建立原则，如何创建，修改删除索引？

索引建立原则如下：

- 1) 对一个表中建大量的索引，应进行权衡：对于 SELECT 查询，大量索引可以提高性能，可以选择最快的查询方法；但是，

会影响 INSERT、UPDATE 和 DELETE 语句的性能，因为对表中的数据进行修改时，索引也要动态的维护。

- 2) 避免对经常更新的表建立过多的索引。
- 3) 对更新少而且数据量大的表创建多个索引，可以大大提高查询性能。
- 4) 对于小型表（行数较少）进行索引可能不会产生优化效果。
- 5) 对于主键和外键列应考虑建索引，因为经常通过主键查询数据，而外键用于表间的连接。
- 6) 很少在查询中使用的列以及值很少的列不应考虑建索引。
- 7) 视图中如果包含聚合函数或连接时，创建视图的索引可以显著提升查询性能。

27. 建立简单存储过程的 SQL 语句

见 PPT

28. 什么是事务？事务有什么特点？

事务(Transaction)是用户定义的一个数据库操作序列，这些操作要么全做，要么全不做，是一个不可分割的工作单位。一个应用程序通常包含多个事务，事务是恢复和并发控制的基本单位，和程序是两个概念。在关系数据库中，一个事务可以是一条 SQL 语句，一组 SQL 语句或整个程序。

事物有如下特性：

1) 原子性 (Atomicity)：

事务是数据库的逻辑工作单位，事务中包括的诸操作要么都做，要么都不做。

2) 一致性 (Consistency)

事务执行的结果必须是使数据库从一个一致性状态（数据库中只包含成功事务提交的结果）变到另一个一致性状态。

3) 隔离性 (Isolation)

对并发执行而言，一个事务的执行不能被其他事务干扰；一个事务内部的操作及使用的数据对其他并发事务是隔离的；并发执行的各个事务之间不能互相干扰。

4) 持续性 (Durability)

一个事务一旦提交，它对数据库中数据的改变就应该是永久性的。接下来的其他操作或故障不应该对其执行结果有任何影响。

29. 什么是并发控制？

并发控制指多个用户同时更新行时用于保护数据库完整的各项技术。目的是保证一个用户的工作不会对另一个用户的工作产生不合理的影响。

并发控制方法是确保数据一致性的方法之一。是当一个事务访问某个数据项时，其他任何事务都不能修改该数据项。实现这一目的的通常通过封锁的方法，只允许事务访问当前加锁的数据对象。

30. 什么是锁？

封锁就是事务 T 在对某个数据对象（例如表、记录等）操作之前，先向系统发出请求，对其加锁，加锁后事务 T 就对该数据对象有了一定的控制，在事务 T 释放它的锁之前，其它的事务不能更新此数据对象。

封锁可以分为：

- 1) 共享锁（S 锁）：又称为读锁，若事务 T 对数据对象 A 加上 S 锁，则事务 T 可以读 A 但不能修改 A，则其它事务只能再对 A 加 S 锁，而不能加 X 锁，直到 T 释放 A 上的 S 锁
- 2) 排它锁（X 锁）：又称为写锁，若事务 T 对数据对象 A 加上 X 锁，则只允许 T 读取和修改 A，其它任何事务都不能再对 A 加任何类型的锁，直到 T 释放 A 上的锁。

31. 数据库应用开发的几大步骤是什么？每个步骤有什么任务？

- 1) 系统分析：通过调查研究，了解应用系统使用者的组织机构、业务范围，确定新系统的目标，并且对系统开发能否达到这一目标做出可行性分析。给系统开发人员感性与理性的认识

- 2) 需求分析：就是要准确、详细地描述出目标系统的功能、性能要求。进行数据需求和处理需求分析, 完成数据库的概念结构设计 (ER 模型)
- 3) 总体设计：
 - ①软件体系结构设计：完成子系统的划分和功能模块的划分。
 - ②逻辑结构设计：数据库的逻辑结构设计的目标就是将概念结构转换成特定的 DBMS 所支持的数据模型，并对其优化的过程。
- 4) 详细设计：
 - ①模块算法设计：总体设计中各模块的实现方法。
 - ②模块算法设计：对于每一个模块，要求首先对模块的接口参数、局部而后全局参数进行描述，列出每个模块存取的关系表，对每个模块的功能进行描述。
 - ③界面设计：所见即所得。
 - ④数据库的物理结构设计：数据库存储结构和物理实现方法；根据要求选用存储设备、存储方法，考虑数据的完整性、安全性和转储策略等；设计索引。
- 5) 系统实施：按照设计的结果进行具体实施，建立数据库并装入原始数据，设置数据库的运行环境，建立存储过程，编写和调试应用程序代码。进行单元测试。
- 6) 系统测试：对各个子系统、各个模块进行联合并试运行。及系统培训。
- 7) 系统运行与维护

32. 客户机服务器结构

1) 文件服务器结构：

数据库存储于网络文件服务器，只提供文件存储功能。客户机的数据库应用程序发出文件请求，网络文件服务器返回数据库文件，数据库操作在客户机发生。

2) 客户机/服务器结构（C/S）：

将 DBAS 的数据管理和数据管理功能与数据库应用相分离，将 DBMS 的数据管理功能在客户端和服务端之间进行合理的分布和配置。其中数据库服务器完成 DBMS 的核心功能。而客户端或应用服务器则负责完成用户交互功能，接收用户数据，根据业务规则处理应用任务，生成并向数据库服务器发出数据操作请求，然后从数据库服务器接收数据查询结果并通过客户端反馈给用户。C/S 结构的数据库应用系统可以拥有多个数据库服务器，数据库服务器、客户端和应用服务器相互协同工作，实现了对数据库服务器上的数据资源共享。

客户端除了完成人机交互外，还需要完成面向应用的数据处理功能，负荷较重，属于典型的“胖客户端”。客户端通过人机交互界面实现人机交互功能；数据库服务器则通过 DBMS，依据事物逻辑完成相应的数据管理功能。

3) 三层客户机/服务器结构 (B/S):

数据处理功能分解并分布在表示层、功能层和数据层三个层次上，分别由 Web 浏览器、Web 应用服务器和数据库服务器来实现。将人机交互、应用业务逻辑处理和数据管理三类功能相互分离，提高了系统的可维护性。并且通过 Web 浏览器可以访问多个异构平台，解决了跨平台数据管理问题。

表示层位于客户端，由 Web 浏览器实现。客户端根据显示逻辑完成具体的数据表示和人机交互功能。客户端功能单一，一般只安装 Web 浏览器，没有其他用户应用程序，属于典型的“瘦客户端”。

功能层位于 Web 应用服务器，实现面向具体应用领域的业务规则。应用服务器接收来自于 Web 浏览器的用户请求，根据应用领域业务规则执行相应的数据库应用程序，通过数据库访问接口向数据库服务器提出数据操作请求；接收来自于数据库服务器的数据访问结果，并通过客户端将结果返回给用户。

数据层位于数据库服务器，通过 DBMS 完成具体的数据存储和数据存取等数据管理功能。数据库服务器接收应用服务器提出的数据操作请求，按照事物逻辑对数据库进行查询和修改，并将数据访问结果返回给应用服务器。

33. ODBC 是什么

ODBC 是由微软定义的一种数据库访问标准。一般的 ODBC 支持的：

- 1) 建立与数据源的连接
- 2) 向数据源提交 SQL 请求
- 3) 处理查询结果
- 4) 将数据源错误转换为标准错误代码，并返回给应用程序
- 5) 提交事务的开始、完成和撤消请求等

