**1数据正确性问题**

数据正确性的威胁来自两个方面：

1.正常运行时可能遇到的数据丢失、数据冗余、数据不正确、数据不一致等异常情况；

2.故障（包括软件和硬件），或者发生停电之类的外部异常事件时，导致的数据丢失、数据不一致；

**如何来解决？**

1.正常运行时的数据正确性问题要通过数据库设计来解决。数据库设计理论、方法是本课程的核心内容。

2.企业中，数据丢失是不允许的，同时系统故障又是难以避免的。事务处理（Transaction processing)和故障恢复。

**2数据操作简单性问题**

1.数据操作的简单性可通过视图（View）和存储过程来提升。

2.视图概念将数据操作的复杂性从普通用户转移动至数据库专业人员，为普通用户屏蔽了表与表的联接运算以及复杂查询条件的表达；

3.而存储过程则将数据操作语句转化为函数调用形式，进一步为数据库普通用户屏蔽了数据操作语言的语法规则

**3数据处理性能问题**

1.企业的数据都集中存储在数据库中，海量的数据。从海量的数据中查找和定位数据是非常耗时。

2.数据集中存储后，所有用户都要来访问数据库，因此数据库服务器的负载非常重。而且是很多用户同时访问数据库。

3.目前很多企业的数据库系统都因处理性能低下，超出用户容忍范围而受到用户的非议与责难，影响企业的正常运作和发展。

4.数据分类、数据排序、数据索引，都是提升数据处理性能的有效方法。

5.提升数据处理性能可以从挖掘和利用硬件特性，数据特性，以及访问特性入手寻找办法。

**3.1度量指标:事务吞吐量：**单位时间中能够处理的交易(事务)数量

**响应时间：**完成单个交易所用的时间**.**

**3.2提高数据库性能的策略：**

1挖掘和利用：2数据特性，3硬件特性，4数据访问特性；

**3.3提高数据库性能的方法：**

1排序;2索引 （哈希索引）；3: 连续的磁盘存储;4: 分类、聚簇, 分区;5: 减少联接;6: 内存缓冲;7: 并发执行;8: 查询优化;9: 日志和数据分盘存储；

**4.1分布式数据库的定义 ：**物理上分散而逻辑上集中的系统，它使用计算机网络将地理位置分散而管理和控制又需要不同程度集中的多个逻辑单位（通常是集中式数据库系统）连接起来，共同组成一个统一的数据库系统。分布式数据库系统可以看成是计算机网络和数据库系统的有机结合。

简述分布式数据库系统的透明性：1分片透明性（完全分布透明性）：2位置透明性（中级分布透明性）：3局部数据模型透明性（低级分布透明性）：4无分布透明性（异构数据）：

**4.2分布式事务的特性**

(1)原子性：事务的操作要么全部执行, 要么全部不执行 ,保证数据库一致性状态

(2)一致性 ：事务的正确性，串行性，并发执行的多个事务,其操作的结果应与以某种顺序串行执行这几个事务所得的结果相同.

(3)持久性：当事务提交后, 其操作的结果将永久化, 而与提交后发生的故障无关

(4)隔离性 ：虽然可以有多个事务同时执行，但是单个事务的执行不应该感知其他事务的存在，因此事务执行的中间结果应该对其他并发事务隐藏

此外，分布式数据库系统中还要考虑数据传送、通信原语和控制报文等。

**5.1数据库的安全主要受到以下因素的威胁：**

1．数据错误2．软硬件故障与灾害破坏3．缺乏保护机制4．管理漏洞5．黑客攻击6．不掌握数据库核心技术7．敏感数据的泄漏问题

**5.2数据库主要的安全需求**是数据库的完整性、可靠性、保密性、可用性，其中完整性包括物理完整性、逻辑完整性和元素完整性；保密性要求包括访问控制、用户认证、审计跟踪、数据加密等内容。

**5.2.1数据库的完整性：**1.在物理完整性方面，要求从硬件或环境方面保护数据库的安全，防止数据被破坏或不可读2.在逻辑完整性方面，要求保持数据库逻辑结构的完整性3.在元素完整性方面，元素完整性主要是指保持数据字段内容的正确性与准确性

5.2.2可审计性：为了能够跟踪对数据库的访问，及时发现对数据库的非法访问和修改，需要对访问数据库的一些重要事件进行记录，利用这些记录可以协助维护数据库的完整性，还可以帮助事后发现是哪一个用户在什么时间影响过哪些值。5.2.3访问控制与用户认证。1.和操作系统相比，数据库的访问控制的难度要大得多。2.操作系统中控制的粒度是文件，数据库中则需要控制到记录和字段一级。

5.2.4保密性与可用性：DBMS除了通过访问控制机制对数据库中的敏感数据加强防护外，还可以通过加密技术对库中的敏感数据加密。但加密虽然可以防止对数据的恶意访问，也显著地降低数据库访问效率。

5.3**数据库安全的重要性：**1．保护敏感信息和数据资产2．计算机网络系统安全的关键环节

6.1**数据库的安全目标：**1完整性：2可信性：3系统灵活性：4用户方便性：5篡改检测：6降低管理成本：

6.2**数据库的安全策略：**是指导信息安全的高级准则，即组织、管理、保护和处理敏感信息的法律、规章及方法的集合。它包括安全管理策略、访问控制策略和信息控制策略。

**6.3实现数据库安全性控制的常用方法和技术有：**

**安全性**问题主要是指保护数据库以防止不合法的使用造成数据泄露、更改或破坏。

数据库安全可分为二类 ：（1）系统安全性：指在系统级控制数据库的存取和使用的机制。（2）数据安全性：指在对象级控制数据库的存取和使用的机制。

1用户标识和鉴定：通过核对用户的名字或身份(ID)，决定该用户对系统的使用权。数据库系统不允许一个未经授权的用户对数据库进行操作。

2存取控制模型（1）访问控制矩阵模型（2）扩展的访问控制矩阵模型（3）多级安全模型（4）信息流安全模型

3视图 ：有了视图机制，就可以在设计数据库应用系统时，对不同的用户定义不同的视图，使机密数据不出现在不应看到这些数据的用户视图上。即通过定义不同的视图及有选择地授予视图上的权限，可以将用户、组或角色限制在不同的数据子集内。

4加密存储：（1）数据库密码系统的选择:可采用对称加密体制。 （2）数据库加密的范围，数据库中不能加密的部分包括：索引字段、关系运算的比较字段 、表间的连接码字段 。 （3）数据库加密对数据库管理系统原有功能的影响

**6.4数据库的完整性**

数据库的完整性是指数据的正确性和相容性。

数据的完整性和安全性是两个不同的概念。前者是为了防止数据库中存在不符合语义的数据，防止错误信息的输入和输出，而后者是保护数据库防止恶意的破坏和非法的存取。

1．设置触发器 。触发器可以完成以下功能： “条件限制”

（1）检查取值类型与范围：检查每个字段输入数据的类型与该字段的类型是否一致 ；（2）依据状态限制：指为保证整个数据库的完整性而设置的一些限制，数据库的值在任何时候都不应该违反这些限制；（3）依据业务限制：指为了使数据库的修改满足数据库存储内容的业务要求，而作出相应的限制。

2．两阶段提交 ：为了保证数据更新结果的正确性，必须防止在数据更新过程中发生处理程序中断或出现错误。解决这个问题的办法是在DBMS中采用两阶段提交(更新)技术。第一阶段称为准备阶段。第二阶段的工作是对需要更新的字段进行真正地修改，这种修改是永久性的。

3．纠错与恢复：许多DBMS提供数据库数据的纠错功能，主要方法是采用冗余的办法，下面介绍几种冗余纠错的技术 ：（1）附加校验纠错码（2）使用镜像技术（3）恢复

**6.5数据库的并发控制**

数据库系统通常支持多用户同时访问数据库，为了有效地利用数据库资源，可能多个程序或一个程序的多个进程并行地运行，这就是数据库的并发操作。当多个用户同时读写同一个字段的时候，会存取不正确的数据，或破坏数据库数据的一致性。据不一致总是由两个因素造成：一是对数据的修改，二是并发操作的发生。并发操作带来的数据不一致性包括三类： 1．丢失修改2．不可重复读3．读“脏”数据

**6.6数据库的备份与恢复**

数据库管理系统必须具有把数据库从错误状态恢复到某一已知的正确状态(亦称为一致状态或完整状态)的功能，这就是数据库的恢复。数据库系统所采用的恢复技术是否行之有效，不仅对系统的可靠程度起着决定性作用，而且对系统的运行效率也有很大影响，是衡量系统性能优劣的重要指标。

**1．故障的种类：**1事务内部的故障 2系统故障 3介质故障 4人为破坏

**2．恢复的实现技术 1**数据转储 2登记日志 3数据库的镜像

**3 SQL Server的安全管理**主要包括以下4个方面：1数据库登录管理2数据库用户管理;3数据库角色管理;4数据库权限管理

**4 SQL Server的安全策略 ：**主要包括安全密码和安全账号、数据库安全配置、操作系统安全设置、日志记录检测、扩展存储过程管理、网络及协议安全管理等方面的策略。下面介绍常用的4种主要的安全策略：

1．安全密码和安全账号策略2．日志审计策略3．扩展存储过程管理策略4．网络及协议安全管理策略

**5 数据库设计要回答的三个问题：**

1) 覆盖性问题：确定一个单位有哪些数据项（字段）？

2） 划分问题： 确定哪些字段构成一个表？

3） 关系问题： 确定表之间有什么关系？

**特征：**业务表单和数据库中表的不一致；

业务表单中通常包含的是综合信息；

而数据库中的一个表只能存储单一类的信息；

如果直接把业务表单中的数据项组成一个表，会带来一系列的数据正确性问题；

正确性

处理性

操使用简单性

分布式数据库

事务处理和故障恢复

数据库性能提升技术

数据正确性问题

数据正确性的威胁来自两个方面：

正常运行时可能遇到的数据丢失、数据冗余、数据不正确数据不一致等异常情况；

故障（包括软件和硬件），或者发生停电之类的外部异常事件时，导致的数据丢失、数据不一致；

正常运行时的数据正确性问题要通过数据库设计来解决。完整性, 一致性;

数据丢失是不允许的，同时系统故障又是难以避免的。事务处理和故障恢复。

数据完整性约束对保证数据的有用性必不可少。数据完整性约束的表达方法。

数据库设计要回答的三个问题：

1) 覆盖性问题：确定一个单位有哪些数据项（字段）？

2） 划分问题： 确定哪些字段构成一个表？3

） 关系问题： 确定表之间有什么关系？特征：业务表单和数据库中表的不一致；业务表单中通常包含的是综合信息；而数据库中的一个表只能存储单一类的信息；如果直接把业务表单中的数据项组成一个表，会带来一系列的数据正确性问题；

数据库中表的定义和现实中使用的业务表单不能直接一一对应；根源在于数据正确性的取得和数据操作简单性的取得存在矛盾。数据正确性的取得要求数据严格按照类和关系的原则来设计表；业务表单通常都是由多个类的数据综合而成的；类之间存在关系，也就是表之间存在联系，因此可以沿着表之间的联系，把多个表中的数据综合起来，形成业务表单；把多个表中的数据综合起来，要执行表与表的联接运算。

处理性能问题

企业的数据都集中存储在数据库中，海量的数据。从海量的数据中查找和定位数据是非常耗时。数据集中存储后，所有用户都要来访问数据库，因此数据库服务器的负载非常重。而且是很多用户同时访问数据库。目前很多企业的数据库系统都因处理性能低下，超出用户容忍范围而受到用户的非议与责难，影响企业的正常运作和发展。

数据分类、数据排序、数据索引，都是提升数据处理性能的有效方法。

提升数据处理性能可以从挖掘和利用硬件特性，数据特性，以及访问特性入手寻找办法。

存储器特性：速度、存储容量、容错性；数据库中的数据量；

数据库性能度量指标:

事务吞吐量(Transaction throughput): 单位时间中能够处理的交易(事务)数量.

响应时间(Response time): 完成单个交易所用的时间.

提高数据库性能的策略:挖掘和利用：数据特性，硬件特性，以及数据访问特性；

提高数据库性能的方法:方法1: 排序;方法2: 索引 （哈希索引）；方法3: 连续的磁盘存储;方法4: 分类、聚簇, 分区;方法5: 减少联接;方法6: 内存缓冲;方法7: 并发执行;方法8: 查询优化;方法9: 日志和数据分盘存储；

其中与数据库设计有关的方法:方法1: 排序;方法2: 索引 ；方法3: 连续的磁盘存储;方法4: 分类、聚簇，分区;方法5: 减少联接;方法6: 内存缓冲;方法9: 日志和数据分盘存储；

简单性问题

操作简单性的实现 - 分层策略

4 layers (层):data presentation(Formatted visualization form),database system(table),File system(Byte stream),Disk(不连续的bit数据块)

数据操作的简单性可通过视图（View）和存储过程来提升。

视图概念将数据操作的复杂性从普通用户转移动至数据库专业人员，为普通用户屏蔽了表与表的联接运算以及复杂查询条件的表达；而存储过程则将数据操作语句转化为函数调用形式，进一步为数据库普通用户屏蔽了数据操作语言的语法规则。

三级模式结构及二级映像实现了数据库系统的数据独立性

外模式的好处:数据独立 – 表结构改变不影响用户视图.安全 – 对不同的用户，见到不同的数据视图；简化数据访问 – 避免写复杂的SQL语句.自定义－给用户的数据不多不少；

什么是分布式数据库

同构/异构系统。自治系统，联邦系统;分布式处理，合作;

动机：资源共享;性能（整个throughout，响应时间）;容错：通过冗余少的可用组件

分布式计算：

形成：组合；拓展。协作：沟通语言: XML；沟通方式：HTTP;

数据安全问题

数据库存储了企业的所有数据。数据库中的数据并不是完全开放的，而是有安全性要求。在企业中，有些数据仅只允许部分人看到，例如，企业中的职工奖金数据，普通职工只允许看到自己的，而门部经理则允许看到本部门所有职工的。职工的工资数据只允许财务部门的专责人员进行修改。数据库的安全控制机制以及安全控制方法。

一、数据库及数据库表定义

　　1、创建基本表

　　　　create table <表名> (<列名><数据类型>[列级完整性约束条件][,<列名><数据类型>[列级完整性约束条件].........[,<表级完整性约束条件>]);

例：   
create table student  ( Sno **char**(**5**) not null unique,  
                        Sname **char**(**20**) unique,  
                        Ssex **char**(**1**),  
                        Sage **int**,  
                        Sdept **char**(**15**));  
2、修改基本表

alert  table  <表名> [add <新列名> <数据类型> [完整性约束]]

                     [drop <完整性约束名>]

                     [modify <列名><数据类型>];

          add: 用于向表中增加新列及新列相关的完整性约束条件，新的列增加成功后不带任何数据

          drop:用于删除指定的完整性约束条件

          modify:用于修改原来的列定义，包括修改列名和数据类型，列数据类型修改后可能会丢失原有的数据  
alert table student add Scome date;  
alert table student modify Sage **smallint**;  
alert table student drop unique(Sname);

3、SQL中没有提供删除属性列的功能，若需要删除列，必须间接通过select语句来实现，将需要的列的内容复制到另一个表中，再删除原表，然后将新表的名字命名为原表名

4、删除基本表：drop table <表名>

5、建立与删除索引

 create [unique] [cluster] index <索引名> on <表名> (<列名> [<次序][,<列名>[<次序]].....);

 注：unique:表明此索引的每个索引值只对应惟一的一个记录

cluster:表时要建立的索引是聚簇索引，即按某一列建好索引后，硬盘上的数据存储顺序也调整为按这个列的顺序存储，索引的顺序和存储顺序是一致的

             drop index <索引名>;

create unique index studentIndex on student (sno);  
drop index studentIndex;

二、数据查询

select  [distinct | distinctRow | all] 列名[，列名][，....]

  [into {outfile | dumpfile} ''文件名'' export\_options]

         [from 表名]

         [where 查询条件]

         [group by 列名[, ....]]

         [having 条件定义]

         [order by {无符号整数|列名|公式}[asc | desc],...]

         [limit [偏移量，] 行数]

         [procedure 过程名称]

注：select语句中所有使用的关键词必须精确的以上面的顺序给出 如果以降序排列，则在order by中加入关键字desc，升序则加入关键字asc,默认是升序；having子名能引用任何列或在查询中命名的别名，它最后运用，就在项目被送到客户之前是，没有变化。需要注意，不要对应该在where子句中的项目使用having。limit子名被用来限制select返回的行数，如果给定1个参数，它指出返回行的最大数目；如果给定2个参数，第一个指定要返回的第一行的偏移量，第二个指定返回行的最大数目，初始行的偏移量是0  
//返回前5个记录  
select \* from table limit **5**;  
  
//返回第6到第15个记录  
select \* from table limit **5**,**10**;

select .... into outfile ''文件名''　格式的select语句将选择的行写入一个文件，文件在服务器上被创建，并且不能是已经存在的，且在服务器主机上还必须有file权限以使用这种select。distinct,使用了这个关键字后，再执行select语句的结果集中，如果有重复的值，则会在结果集中去掉重复的值而只保留一个值。

//选择学生表中所有学生的姓名记录，并且筛选掉重复的记录，将记录保存在名字为name的结果集中，并按学号由大到小排列  
select distinct sname as name, sno from student order by sno desc;  
  
//选择学生表中所有学生的姓名和年龄，并将每个同学的姓名后加上“同学”二字，将每个人的年龄增加1岁，同是计算所有同学年龄的平均值　  
// sum() ---- 求所有数据的总和  
// max() ----求所有数据中的最大值  
// min() -----求所有数据中的最小值  
// count() ---求所有数据的个数  
// avg() ----　求所有数据的平均值  
select sname, "同学", age+**1**, avg(age) from student;  
  
//列出学号在1和100之间，并且成绩大于85分的所有姓”赵“同学的所胡记录  
// % ---代表任意字符  
//使用通配符时，前面必须用关键字”like”  
select \* from student where sname like ''赵%’ and sno between 1 and 100 and sno in (select sno from student where grade>85);  
  
//使用学生表、课程表以及学生选课表这3个基本表，列出所有学生的姓名、学号、所选课程的课程名称及该课程的学分  
select sname, sno, cno, credit from student, course, sc where student.sno=sc.sno and sc.cno=course.cno;

三、数据更新

1、数据插入

insert into <表名> [<列名>[,<列名>]....) values (value1, value2,...);

             如果表中的每一列均有数据插入，则可不用指定任何表名

2、数据修改

　　　update <表名> set <列名>=新值 where <条件表达式>

3、数据删除

           删除指的是删除数据库中的一个记录，而不是删除某一列

           delete from <表名> where <条件表达式>