数据库系统之二
--数据库语言-SQL

战德臣

哈尔滨工业大学　教授.博士生导师
黑龙江省教学名师
教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

第6讲 概览SQL语言

战德臣

哈尔滨工业大学　教授.博士生导师
黑龙江省教学名师
教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

本讲学习什么?

战德臣 教授

基本内容

1. SQL语言概述?
2. SQL语言之DDL-定义数据库
3. SQL语言之DML-操纵数据库

重点与难点

- SQL-DDL的基本语句：CREATE DATABASE, CREATE TABLE
- SQL-DML的基本语句：INSERT, DELETE，UPDATE, SELECT
- SQL-SELECT语句的训练：正确表达各种查询需求

SQL语言概述

战德臣

哈尔滨工业大学　教授.博士生导师
黑龙江省教学名师
教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

SQL语言概述
(1)SQL语言提出和发展

战德臣 教授

- 1974年，由Boyce和Chamber提出。
- 1975-1979年，由IBM的San Jose研究室在System R上首次实现，称为

Sequel(Structured English QUEry Language) → **SQL**(Structured Query Language)。

- 1986年ANSI/ISO推出SQL标准：**SQL-86**
- 1989年ANSI/ISO推出SQL标准: **SQL-89**
- 1992年进一步推出了SQL标准：**SQL-92**，也称为**SQL2**

  - 是SQL-89的超集
  - 增加了新特性，如新数据类型，更丰富数据操作，更强完整性支持等
  - 原SQL-89被称为entry-SQL, 扩展的被称为Intermediate级和Full级

SQL语言概述
(1)SQL语言提出和发展

战德臣 教授

- 1999年进一步推出了SQL标准： **SQL-99**，也称为**SQL3**
  - 对面向对象的一些特征予以支持，支持抽象数据类型
  - 支持行对象和列对象等
  - 对递归、触发等复杂操作也予以规范化定义
  - 废弃了SQL2的分级，但定义了core-SQL及扩展的SQL
- **SQL 2003; SQL 2006; SQL 2008。**

面向对象数据库

对象关系数据库

数据库应用程序

- SQL还有一个标准是**SQL X/Open**标准，主要强调各厂商产品的可移植性，只包含被各厂商广泛认可的操作

■ **“标准”主要用于衡量一个软件商的产品是否符合共同的约定。**

■ **“标准”使得用户可以学习“标准”规定的语言，而无需关注具体的软件产品。但也应注意不同软件商的数据库产品满足的标准可能是不一样的，具体应用还是略有差异。**

SQL语言概述
(2)SQL语言的功能概述

战德臣 教授

➢SQL语言是集DDL、DML和DCL于一体的数据库语言

➢SQL语言主要由以下9个单词引导的操作语句来构成，但每一种语句都能表达复杂的操作请求

- ❑ DDL语句引导词：**Create**(建立), Alter (修改), **Drop**(撤消)
  - ✓ 模式的定义和删除，包括定义Database, Table, View, Index,完整性约束条件等，也包括定义对象(RowType行对象, Type列对象)
- ❑ DML语句引导词：**Insert** , **Delete**, **Update**, **Select**
  - ✓ 各种方式的更新与检索操作，如直接输入记录，从其他Table(由SubQuery建立)输入
  - ✓ 各种复杂条件的检索，如连接查找，模糊查找，分组查找，嵌套查找等
  - ✓ 各种聚集操作，求平均、求和、...等，分组聚集，分组过滤等
- ❑ DCL语句引导词：**Grant**, **Revoke**
  - ✓ 安全性控制：授权和撤消授权

➢**交互式SQL➔嵌入式SQL➔动态SQL等**

利用SQL语言建立数据库

战德臣

哈尔滨工业大学　教授.博士生导师

黑龙江省教学名师

教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

战德臣 教授

学生选课数据库SCT

- **学生**： 学号S#, 姓名Sname, 性别Ssex, 年龄Sage, 所属系别D#，班级Sclass

  Student (S# char(8), Sname char(10), Ssex char(2),
  Sage integer, D# char(2), Sclass char(6))

- **院系**： 系别D#，系名Dname, 系主任Dean

  Dept (D# char(2), Dname char(10), Dean char(10))

Student

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402

Dept

D#	Dname	Dean
01	机电	李三
02	能源	李四
03	计算机	李五
04	自动控制	李六

利用SQL语言建立数据库
(1)课堂讲义使用的数据库

战德臣 教授

- **课程：** 课号C#, 课名Cname，教师编号T#，学时Chours，学分Credit

  Course (C# char(3), Cname char(12), Chours integer, Credit float(1), T# char(3))

后面要反复使用它们！

- **教师：** 教师编号T#，教师名Tname, 所属院系D#，工资Salary

  Teacher (T# char(3), Tname char(10), D# char(2), Salary float(2))

- **选课：** 学号S#, 课号C#, 成绩Score

  SC (S# char(8), C# char(3), Score float(1))

Course

C#	Cname	Chours	Credit	T#
001	数据库	40	6	001
003	数据结构	40	6	003
004	编译原理	40	6	001
005	C 语言	30	4.5	003
002	高等数学	80	12	004

Teacher

T#	Tname	D#	Salary
001	赵三	01	1200.00
002	赵四	03	1400.00
003	赵五	03	1000.00
004	赵六	04	1100.00

SC

S#	C#	Score
98030101	001	92
98030101	002	85
98030101	003	88
98040202	002	90
98040202	003	80
98040202	001	55
98040203	003	56
98030102	001	54
98030102	002	85
98030102	003	48

战德臣 教授

建立数据库

- **包括两件事：定义数据库和表（使用DDL），向表中追加元组（使用DML）**

- **DDL: Data Definition Language**
  - 创建数据库(DB)—**Create Database**
  - 创建DB中的Table(定义关系模式)---**Create Table**
  - 定义Table及其各个属性的约束条件(定义完整性约束)
  - 定义View (定义外模式及E-C映像)
  - 定义Index、Tablespace… …等(定义物理存储参数)
  - 上述各种定义的撤消与修正
- **DDL通常由DBA来使用，也有经DBA授权后由应用程序员来使用**

先学习简单形式！

战德臣 教授

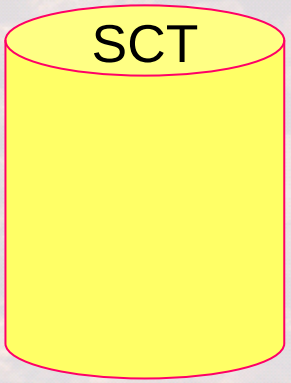
创建Database

- 数据库(Database)是若干具有相互关联关系的Table/Relation的集合
- 数据库可以看作是一个集中存放若干Table的大型文件
- create database的简单语法形式：

create database 数据库名;

示例：创建课程学习数据库SCT

create database SCT;



接着就可创建表了！

创建Table

- **create table**简单语法形式：

  Create table 表名(列名 数据类型 [***Primary key*** |***Unique***] [***Not null***]
  [, 列名 数据类型 [***Not null***] , …]) ;

- “ [ ] ”表示其括起的内容可以省略，“ | ” 表示其隔开的两项可取其一
- **Primary key:** 主键约束。每个表只能创建一个主键约束。
- **Unique:** 唯一性约束(即候选键)。可以有多个唯一性约束。
- **Not null:** 非空约束。是指该列允许不允许有空值出现，如选择了**Not null**表明该列不允许有空值出现。
- 语法中的数据类型在**SQL**标准中有定义

利用SQL语言建立数据库
(3)创建关系/表的语句—Create Table

战德臣 教授

- 在SQL-92标准中定义的数据类型
  - char (n) :固定长度的字符串
  - varchar (n) :可变长字符串
  - int :整数 // 有时不同系统也写作integer
  - numeric (p，q) :固定精度数字，小数点左边p位，右边p-q位
  - real :浮点精度数字 //有时不同系统也写作float(n)，小数点后保留n位
  - date :日期 (如 2003-09-12)
  - time ：时间 (如 23:15:003)
  - …
- 现行商用DBMS的数据类型有时和上面有些差异，请注意;
- 和高级语言的数据类型，总体上是一致的，但也有些差异。

利用SQL语言建立数据库
(3)创建关系/表的语句—Create Table

战德臣 教授

示例：定义学生表 Student

```
Create Table Student ( S# char(8) not null , Sname char(10),
        Ssex char(2), Sage integer, D# char(2), Sclass char(6) );
```

示例：定义课程表Course

```
Create Table Course ( C# char(3) , Cname char(12), Chours integer,
        Credit float(1), T# char(3) );
```

同学可自己定义其他的表！

接着可向表中追加元组了！

Student

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402

Course

C#	Cname	Chours	Credit	T#
001	数据库	40	6	001
003	数据结构	40	6	003
004	编译原理	40	6	001
005	C 语言	30	4.5	003
002	高等数学	80	12	004

战德臣 教授

建立数据库

- 包括两件事：定义数据库和表（使用DDL），向表中追加元组（使用DML）
- DML: Data Manipulation Language
  - 向Table中追加新的元组：**Insert**
  - 修改Table中某些元组中的某些属性的值: Update
  - 删除Table中的某些元组: Delete
  - 对Table中的数据进行各种条件的检索: Select
- DML通常由用户或应用程序员使用，访问经授权的数据库

战德臣 教授

向表中追加元组

- insert into简单语法形式：

insert into 表名[(列名 [, 列名]...]

values (值 [, 值] , ...) ;

- **values**后面值的排列，须与**into**子句后面的列名排列一致
- 若表名后的所有列名省略， 则**values**后的值的排列，须与该表存储中的列名排列一致

利用SQL语言建立数据库
(5)向表中追加元组的值 –INSERT INTO

战德臣 教授

Student

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402

示例：追加学生表中的元组

```
Insert Into  Student
Values ( '98030101' , '张三', '男', 20,  '03',  '980301');

Insert Into  Student ( S#, Sname, Ssex, Sage, D# , Sclass)
Values ( '98030102' , '张四', '女', 20,  '03',  '980301');
```

请练习追加元组吧！

示例：追加课程表中的元组

```
Insert Into  Course                          /*所有列名省略，须与定义或存储的列名顺序一致
Values ( '001' , '数据库', 40, 6,  '001');

                                             /*如列名未省略，须与语句中列名的顺序一致
Insert Into  Course(Cname, C#, Credit, Chours, T#)
Values ('数据库', '001', 6, 40, '001');
```

数据库建立完成后关键是要用呀！

Course

C#	Cname	Chours	Credit	T#
001	数据库	40	6	001
003	数据结构	40	6	003
004	编译原理	40	6	001
005	C 语言	30	4.5	003
002	高等数学	80	12	004

利用SQL语言进行简单查询

战德臣

哈尔滨工业大学　教授.博士生导师

黑龙江省教学名师

教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

战德臣 教授

SQL提供了结构形式一致但功能多样化的检索语句Select

➢Select 的简单语法形式：

Select 列名 [[, 列名] …]
From 表名
[*Where* 检索条件] ;

这是最基本的哟！

➢语义：从表名所给出的表中，查询出满足检索条件的元组，并按给定的列名及顺序进行投影显示。

➢相当于：$\Pi_{列名,\ldots,列名}(\sigma_{检索条件}(表名))$

➢**Select语句中的select … , from… , where…,** 等被称为子句，在以上基本形式基础上会增加许多构成要素，也会增加许多新的子句，满足不同的需求。

利用SQL语言进行简单查询
(1)单表查询-SELECT-FROM-WHERE

战德臣 教授

接下来就要练习了...

示例：检索学生表中所有学生的信息

```
Select   S#, Sname, Ssex, Sage, Sclass, D#
  From  Student ;

Select   *   From  Student ;          //如投影所有列，则可以用*来简写
```

示例：检索学生表中所有学生的姓名及年龄

```
Select   Sname, Sage                  //投影出某些列
  From  Student ;
```

示例：检索学生表中所有年龄小于等于19岁的学生的年龄及姓名

```
Select   Sage, Sname                  //投影的列可以重新排定顺序
  From  Student
Where   Sage <= 19;
```

Student

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402

战德臣 教授

检索条件的书写

➢与选择运算$\sigma_{con}(R)$的条件con书写一样，只是其逻辑运算符用 and , or, not 来表示, 同时也要注意运算符的优先次序及括弧的使用。书写要点是注意对自然语言检索条件的正确理解。

示例：检索教师表中所有工资少于1500元或者工资大于2000元 并且是03系的教师姓名？

```
Select   Tname
  From  Teacher
Where   Salary < 1500  or  Salary  > 2000  and  D# = '03';
```

是这样的吗？

```
Select   Tname
  From  Teacher
Where   (Salary < 1500  or  Salary > 2000)  and  D# = '03';
```

Teacher

T#	Tname	D#	Salary
001	赵三	01	1200.00
002	赵四	03	1400.00
003	赵五	03	1000.00
004	赵六	04	1100.00

利用SQL语言进行简单查询
(2)检索条件-SELECT-FROM-WHERE

战德臣 教授

示例：求或者学过001号课程, 或者学过002号课程的学生的学号

```
Select   S#   From  SC
Where   C# = '001'   OR   C#='002';
```

示例：求既学过001号课程, 又学过002号课程的学生的学号?

```
Select   S#   From  SC
Where   C# = '001'   AND   C#='002';
```

//正确的SQL语句在讲义后面的示例中讲解

是这样的吗？

其结果是什么？

SC

S#	C#	Score
98030101	001	92
98030101	002	85
98030101	003	88
98040202	002	90
98040202	003	80
98040202	001	55
98040203	003	56
98030102	001	54
98030102	002	85
98030102	003	48

ICES 战德臣 教授

结果唯一性问题

关系模型不允许出现重复元组。但现实DBMS，却允许出现重复元组，但也允许无重复元组。

➢在Table中要求无重复元组是通过定义Primary key或Unique来保证的;而在检索结果中要求无重复元组, 是通过DISTINCT保留字的使用来实现的。

示例：在选课表中，检索成绩大于80分的所有学号

```
Select   S#
  From  SC
Where   Score > 80 ;
```

//有重复元组出现，比如一个同学两门以上课程大于80

注意有扩展了...

```
Select   DISTINCT  S#
  From  SC
Where   Score > 80;
```

//重复元组被DISTINCT过滤掉，只保留一份

其结果差异清楚了吗？

SC

S#	C#	Score
98030101	001	92
98030101	002	85
98030101	003	88
98040202	002	90
98040202	003	80
98040202	001	55
98040203	003	56
98030102	001	54
98030102	002	85
98030102	003	48

战德臣 教授

结果排序问题

DBMS可以对检索结果进行排序，可以升序排列，也可以降序排列。

- Select语句中结果排序是通过增加order by子句实现的

注意有扩展了…

order by 列名 [*asc* | *desc*]

- 意义为检索结果按指定列名进行排序，若后跟asc或省略，则为升序；若后跟desc, 则为降序。
- **示例：按学号由小到大的顺序显示出所有学生的学号及姓名**

```
Select  S#, Sname  From  Student
Order  By  S#  ASC ;
```

- **示例：检索002号课大于80分的所有同学学号并按成绩由高到低顺序显示**

```
Select  S#  From  SC  Where  C# = '002'  and  Score > 80
Order  By  Score  DESC ;
```

战德臣 教授

模糊查询问题

比如检索姓张的学生，检索张某某；这类查询问题，**Select**语句是通过在检索条件中引入运算符**like**来表示的

- 含有**like**运算符的表达式

列名 [*not*] *like* “字符串”

注意有扩展了…

- 找出匹配给定字符串的字符串。其中给定字符串中可以出现%, _等匹配符.
- 匹配规则：
  - “%” 匹配零个或多个字符
  - “_” 匹配任意单个字符
  - “ \ ” 转义字符，用于去掉一些特殊字符的特定含义，使其被作为普通字符看待, 如用 “\%”去匹配字符%，用_ 去匹配字符_

利用SQL语言进行简单查询
(5)模糊查询-SELECT-FROM-WHERE * LIKE *

战德臣 教授

示例：检索所有姓张的学生学号及姓名

```
Select   S#, Sname   From  Student
Where   Sname  Like   '张%' ;
```

示例：检索名字为张某某的所有同学姓名

```
Select   Sname   From  Student
Where   Sname Like '张_ _';
```

示例：检索名字不姓张的所有同学姓名

```
Select   Sname   From  Student
Where   Sname Not  Like '张%';
```

利用SQL语言进行多表联合查询

战德臣

哈尔滨工业大学　教授.博士生导师

黑龙江省教学名师

教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

战德臣 教授

多表联合查询

多表联合检索可以通过连接运算来完成，而连接运算又可以通过广义笛卡尔积后再进行选择运算来实现。

- Select 的多表联合检索语句

Select 列名 [[, 列名] ...]

From 表名1, 表名2, ...

Where 检索条件；

- 相当于$\Pi_{列名,\ldots,列名}(\sigma_{检索条件}(表名1 \times 表名2 \times \ldots))$
- 检索条件中要包含连接条件，通过不同的连接条件可以实现等值连接、不等值连接及各种θ-连接

战德臣 教授

θ-连接之等值连接

示例：按“001”号课成绩由高到低顺序显示所有学生的姓名(二表连接)

```
Select   Sname  From  Student, SC
Where   Student.S# = SC.S#  and  SC.C# = '001'
Order By  Score  DESC;
```

注意连接条件...

➢多表连接时，如两个表的属性名相同，则需采用表名. 属性名方式来限定该属性是属于哪一个表

示例：按‘数据库’课成绩由高到低顺序显示所有同学姓名(三表连接)

```
Select   Sname  From  Student, SC, Course
Where   Student.S# = SC.S#  and  SC.C# = Course.C#
          and  Cname = '数据库'
Order By  Score  DESC;
```

战德臣 教授

重名之处理

- 连接运算涉及到重名的问题，如两个表中的属性重名，连接的两个表重名(同一表的连接)等，因此需要使用**别名**以便区分
- **select**中采用别名的方式

Select 列名 **as** 列别名 **[[,** 列名 **as** 列别名**] ...]**

From 表名**1 as** 表别名**1,** 表名**2 as** 表别名**2, ...**

Where 检索条件；

- 上述定义中的**as** 可以省略
- 当定义了别名后，在检索条件中可以使用别名来限定属性

θ-连接之不等值连接

- **示例：求有薪水差额的任意两位教师**

```
Select   T1.Tname as Teacher1, T2.Tname  as Teacher2
From  Teacher  T1,  Teacher  T2
Where   T1.Salary > T2.Salary ;
```

- **示例：求年龄有差异的任意两位同学的姓名**

```
Select   S1.Sname as Stud1, S2.Sname  as Stud2
From  Student  S1,  Student  S2
Where   S1.Sage > S2.Sage ;
```

- 请同学书写一下：**求‘001’号课程有成绩差的任意两位同学**
- 有时表名很长时，为书写条件简便，也定义表别名，以简化书写

利用SQL语言进行多表联合查询
(5)多表联合查询训练

战德臣 教授

示例：求既学过“001”号课又学过 “002”号课的所有学生的学号

```
Select   S1.S#   From  SC   S1, SC   S2
Where   S1.S# = S2.S#  and  S1.C#='001'
          and  S2.C#='002 ;
```

示例：求“001”号课成绩比“002”号课成绩高的所有学生的学号

```
Select   S1.S#   From  SC   S1, SC   S2
Where   S1.S# = S2.S#  and  S1.C#='001'
          and  S2.C#='002'  and  S1.Score > S2.Score;
```

特别注意同表的连接条件…

SC **S1**

S#	C#	Score
98030101	001	92.0
98030101	002	85.0
98030101	003	88.0
98040202	002	90.5
98040202	003	80.0
98040202	001	55.0
98050104	003	56.0
98030102	001	54.0
98030102	002	85.0
98030102	003	48.0

SC **S2**

S#	C#	Score
98030101	001	92.0
98030101	002	85.0
98030101	003	88.0
98040202	002	90.5
98040202	003	80.0
98040202	001	55.0
98050104	003	56.0
98030102	001	54.0
98030102	002	85.0
98030102	003	48.0

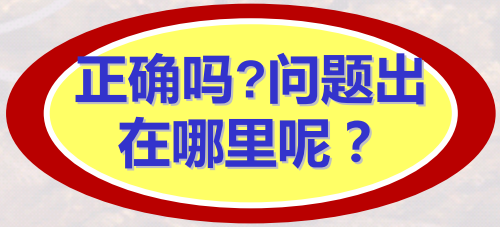
利用SQL语言进行多表联合查询
(5)多表联合查询训练

战德臣 教授

- 注意正确理解用自然语言表达的查询语义，并用SQL正确表达
- **示例：列出没学过李明老师讲授课程的所有同学的姓名?**

```
Select   Sname   From  Student  S, SC, Course  C, Teacher  T
Where  T.Tname <> '李明'  and  C.C# = SC.C#
       and  SC.S# = S.S#  and  T.T# = C.T#;
```

//正确的SQL语句在讲义后面的示例中讲解



利用SQL语言进行增-删-改

战德臣

哈尔滨工业大学　教授.博士生导师
黑龙江省教学名师
教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

利用SQL语言进行增-删-改
(1) SQL-之更新操作

战德臣 教授

- 元组新增Insert：新增一个或一些元组到数据库的Table中
- 元组更新Update: 对某些元组中的某些属性值进行重新设定
- 元组删除Delete：删除某些元组

- SQL-DML既能单一记录操作，也能对记录集合进行批更新操作
- SQL-DML之更新操作需要利用前面介绍的子查询(Subquery)的概念，以便处理“一些”、“某些”等。

利用SQL语言进行增-删-改
(2) SQL-之INSERT

战德臣 教授

- 元组新增Insert命令有两种形式
- 单一元组新增命令形式：插入一条指定元组值的元组

```
insert into 表名 [(列名[，列名]...)]
values (值 [，值]...);
```

- 批数据新增命令形式：插入子查询结果中的若干条元组。待插入的元组由子查询给出。

```
insert into 表名 [(列名[，列名]...)]
子查询;
```

要先学好 Select...From ...Where

战德臣 教授

示例：单一元组新增

```
Insert  Into  Teacher  (T#, Tname, D#, Salary)
 Values  (“005”, “阮小七”, “03”, “1250”);

Insert  Into  Teacher
 Values  (“006”, “李小虎”, “03”, “950”);
```

战德臣 教授

➢示例：批元组新增

新建立Table: St(S#, Sname), 将检索到的满足条件的同学新增到该表中

```
Insert  Into  St  (S#, Sname)
        Select  S#, Sname  From  Student
        Where  Sname like '%伟' ;

Insert Into St (S#, Sname)
        Select S#, Sname From Student Order By Sname;
```

➢注意：当新增元组时，**DBMS**会检查用户定义的完整性约束条件等，如不符合完整性约束条件，则将不会执行新增动作（将在后面介绍）。

利用SQL语言进行增-删-改
(2) SQL-之INSERT

战德臣 教授

示例：新建Table: SCt(S#, C#, Score), 将检索到的成绩及格同学的记录新增到该表中

```
Insert  Into  SCt  (S#, C#, Score)
      Select  S#, C#, Score  From SC
      Where  Score>=60 ;
```

示例：追加成绩优秀同学的记录

```
Insert  Into  SCt  (S#, C#, Score)
      Select  S#, C#, Score  From SC
      Where  Score>=90 ;
```

利用SQL语言进行增-删-改
(2) SQL-之INSERT

战德臣 教授

➢还可以有更复杂的“查询后插入到新表中”的语句，例如可以将中间结果存储成表---这很有用！

示例：新建Table: St(S#, Sname, avgScore), 将检索到的同学的平均成绩新增到该表中

```
Insert  Into  St  (S#, Sname, avgScore)
        Select  S#, Sname, Avg(Score)  From  Student, SC
        Where  Student.S# = SC.S#
        Group by   Student.S# ;
```

➢此SELECT语句的书写语法后面再解释。

要先学好 Select...From ...Where

利用SQL语言进行增-删-改
(3) SQL-之DELETE

战德臣 教授

- 元组删除Delete命令：删除满足指定条件的元组

  Delete From 表名 [Where 条件表达式] ;

- 如果Where条件省略，则删除所有的元组。

示例：删除SC表中所有元组

```
Delete From SC ;
```

示例：删除98030101号同学所选的所有课程

```
Delete From SC Where S# = '98030101' ;
```

示例：删除自动控制系的所有同学

```
Delete From Student Where D# in
        ( Select D# From Dept Where Dname = '自动控制');
```

---此是一简单的嵌套子查询，后面会有更详细解释。

利用SQL语言进行增-删-改
(3) SQL-之DELETE

战德臣 教授

➢还可以有更复杂的“条件控制的删除” 语句，---这很有用！

示例：删除有四门不及格课程的所有同学

```
Delete From Student Where S# in
    ( Select S# From SC Where Score < 60
      Group by S# Having Count(*)>= 4 );
```

➢此SELECT语句的书写语法后面再解释

➢当删除元组时，DBMS会检查用户定义的完整性约束条件等，如不符合完整性约束条件，则将不会执行删除动作

利用SQL语言进行增-删-改
(4) SQL-之UPDATE

➢元组更新**Update**命令：用指定要求的值更新指定表中满足指定条件的元组的指定列的值

```
Update   表名
Set   列名 = 表达式 | (子查询)
      [ [ , 列名 = 表达式 | (子查询) ] ... ]
[ Where   条件表达式] ;
```

➢ 如果**Where**条件省略，则更新所有的元组。

示例：将所有教师工资上调5%

```
Update  Teacher
Set   Salary  = Salary * 1.05  ;
```

示例：将所有计算机系的教师工资上调10%

```
Update  Teacher
Set   Salary  = Salary * 1.1
Where  D#  in
        ( Select   D#  From  Dept   Where  Dname = '计算机');
```

利用SQL语言进行增-删-改
(4) SQL-之UPDATE

战德臣 教授

➢ 还可以有更复杂的“条件控制的更新” 语句，---这很有用！

示例：当某同学001号课的成绩低于该课程平均成绩时，将该同学该门课成绩提高5%

```
Update  SC
Set   Score  = Score * 1.05
Where   C# = '001'  and   Score < some
                ( Select   AVG(Score)  From SC
                  Where  C# = '001' ) ;
```

➢此SELECT语句的书写语法后面再解释。

要先学好 Select...From ...Where

战德臣 教授

示例：将张三同学001号课的成绩置为其班级该门课的平均成绩

```
Update  SC
Set   Score  = ( Select   AVG(SC2.Score)
                  From  SC  SC1, Student S1, SC SC2, Student  S2
                  Where  S1.Sclass = S2.Sclass  and  SC1.S# = S1.S#
                         and SC2.S# = S2.S#  and  S1.Sname='张三'
                         and SC1.C# = '001'  and  SC1.C# = SC2.C#  )
Where   C# = '001'  and   S#  in ( Select   S#  From  Student
                  Where  Sname = '张三' ) ;
```

- 此SELECT语句的书写语法后面再解释。

利用SQL语言修正与撤销数据库

战德臣

哈尔滨工业大学　教授.博士生导师

黑龙江省教学名师

教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

战德臣 教授

修正数据库：修正数据库的定义，主要是修正表的定义

修正基本表的定义

```
alter  table  tablename
[add  {colname  datatype, …}]
[drop {完整性约束名}]
[modify {colname  datatype, …}]
```

增加新列

删除完整性约束

修改列定义

示例：在学生表Student(S#,Sname,Ssex,Sage,D#,Sclass)基础上增加二列Saddr, PID

```
Alter  Table  Student Add  Saddr  char[40],  PID  char[18] ;
```

示例：将上例表中Sname列的数据类型增加两个字符

```
Alter  Table  Student  Modify  Sname char(10) ;
```

示例：删除学生姓名必须取唯一值的约束

```
Alter  Table  Student  Drop  Unique( Sname );
```

战德臣 教授

撤消基本表

drop table 表名

示例：撤消学生表Student

```
Drop Table Student;
```

示例：撤消教师表Teacher

```
Drop Table Teacher;
```

➢注意，SQL-delete语句只是删除表中的元组, 而撤消基本表drop table的操作是撤消包含表格式、表中所有元组、由该表导出的视图等相关的所有内容，所以使用要特别注意。

Delete...From与Drop table的区别清楚了吗？

战德臣 教授

撤消数据库

drop database 数据库名;

示例：撤消SCT数据库

Drop database SCT;

利用SQL语言修正与撤销数据库
(2) SQL-DDL之数据库指定与关闭命令

战德臣 教授

- 有些DBMS提供了操作多个数据库的能力，此时在进行数据库操作时需要指定待操作数据库与关闭数据库的功能。

指定当前数据库

use 数据库名;

关闭当前数据库

close 数据库名;

回顾本讲学了什么?

战德臣

哈尔滨工业大学　教授.博士生导师

黑龙江省教学名师

教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

回顾本讲学习了什么?

ICES 战德臣 教授

SQL：结构化查询语言

DDL

- Create Database
- Create Table

DML

- Insert… Into…
- Update..Set..Where..
- Delete..From..Where..
- Select..From..Where..

Select..From..Where.. 包含：

单表查询
多表查询
模糊查询
结果排序
结果去重复
条件书写

典型DBMS交互环境简介-SQL Server

战德臣

哈尔滨工业大学　教授.博士生导师

黑龙江省教学名师

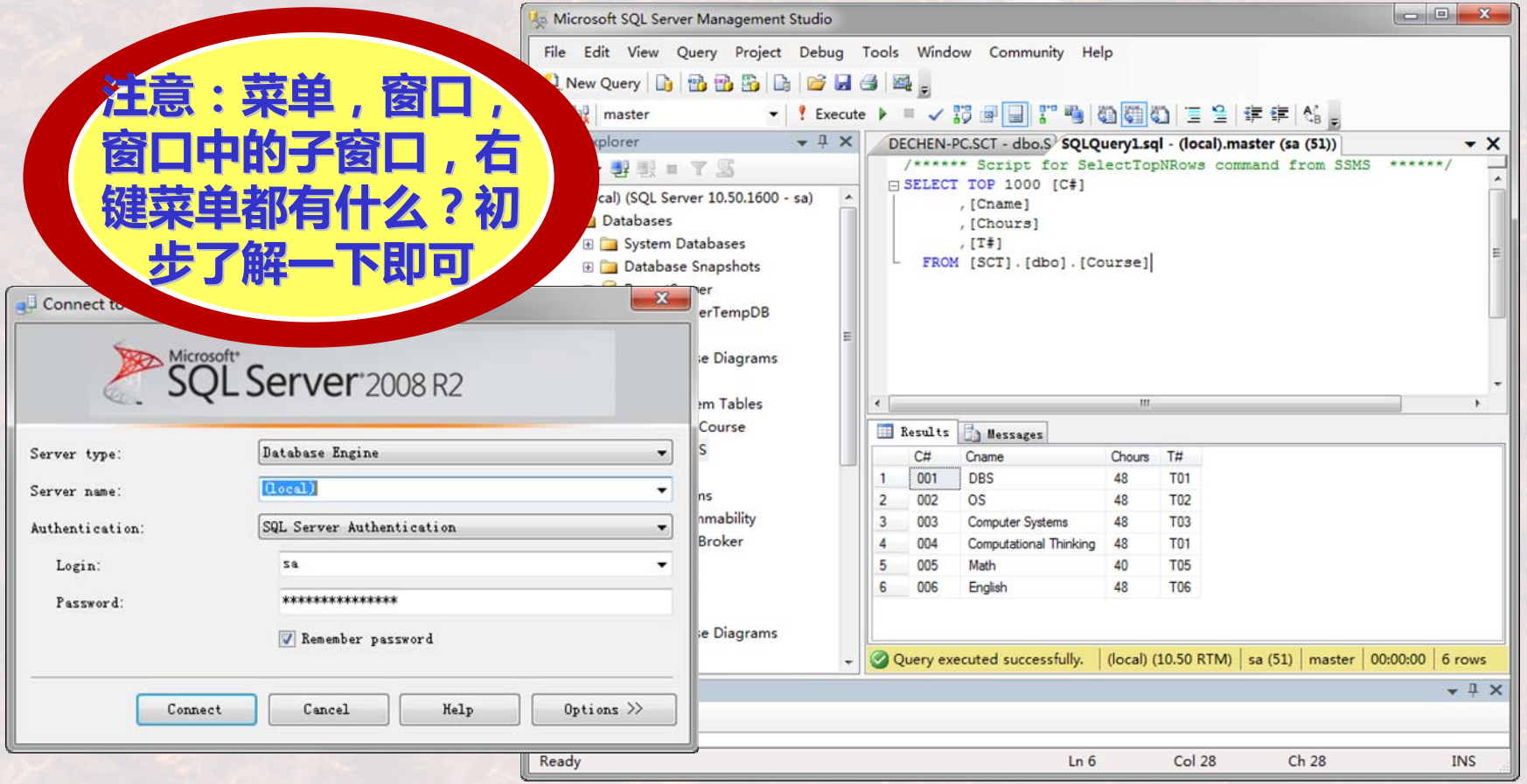
教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

战德臣 教授

SQL Server

- Microsoft提供的一款关系数据库管理系统
- 安装SQL Server并熟悉其操作界面

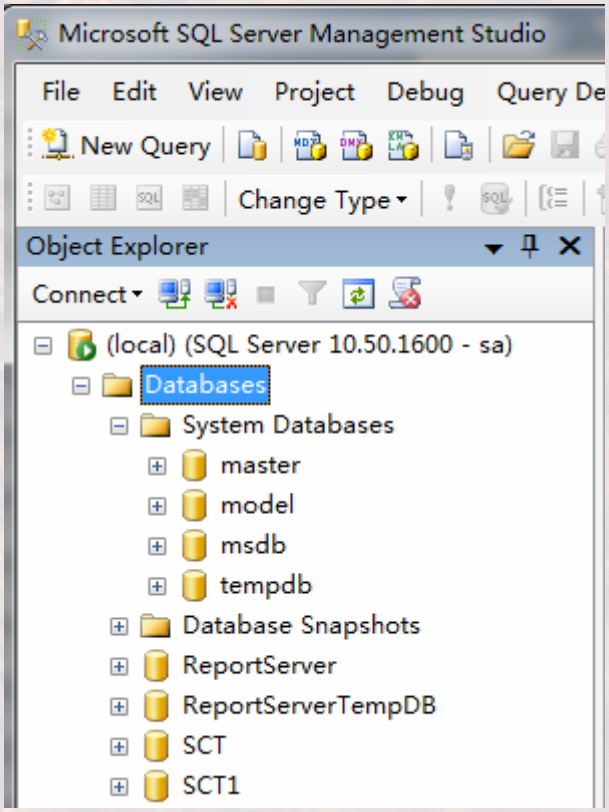


战德臣 教授

SQL Server 的系统数据库

- **Master**：是SQL Server中最重要的系统数据库，存储SQL Server中的元数据。
- **Model**：模板数据库，在创建新的数据库时，SQL Server将会复制此数据库作为新数据库的基础。
- **Msdb**：代理服务数据库，提供一个存储空间。
- **Tempdb**：临时数据库，为所有的临时表、临时存储过程及其他临时操作提供存储空间，断开连接时，临时表与存储过程自动被删除。

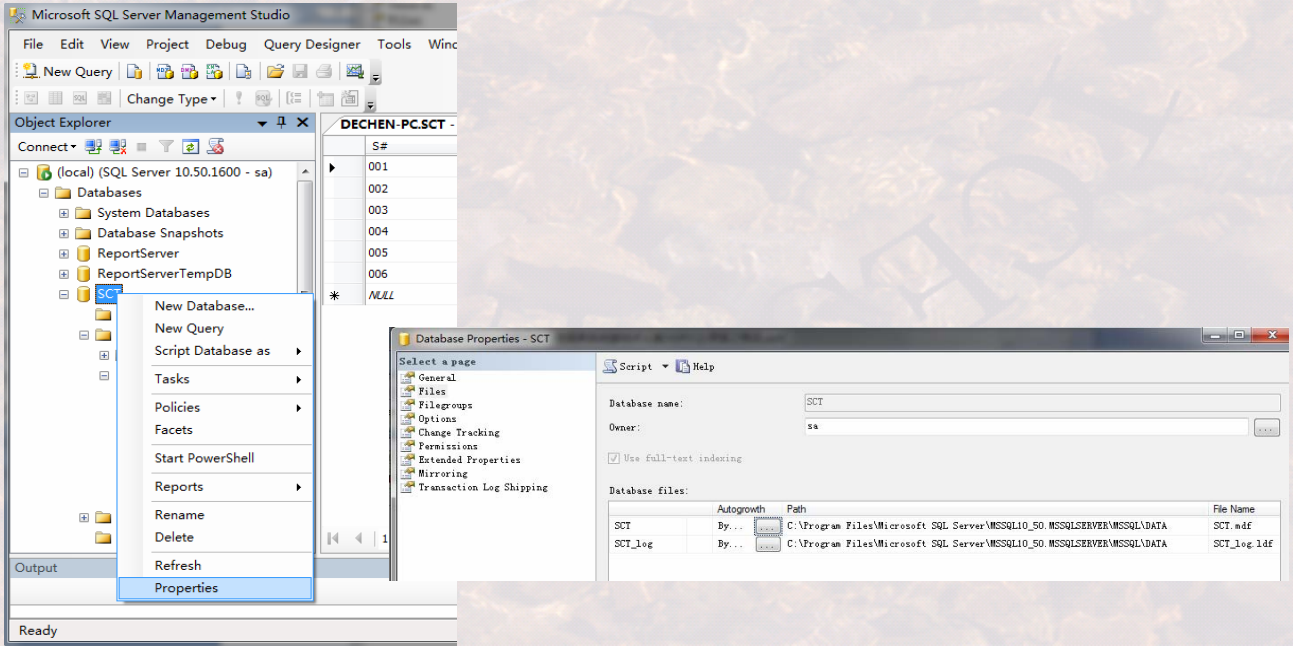
SQL Server安装完成，系统会自动安装！



战德臣 教授

SQL Server的数据库

- 文件：有三种文件扩展名：.mdf、.ndf、.ldf
  - 主数据库文件：扩展名为. mdf，是存储数据库的启动信息和部分或全部数据。一个数据库可以有多个数据库文件，但主数据库文件只有一个。
  - 辅助数据文件：扩展名为. ndf，用于放置主数据库文件中所定义数据库的其它数据，可有多个。在数据庞大时，可以帮助存储数据。
  - 日志文件：扩展名.ldf。每个数据库至少有一个事务日志文件。
- 页面：是SQL Server存储的最小单位。一页为8K或8192字节。
- 空间(extent)：是8个连续的页面，即64K数据，是分配数据表存储空间的一种单位

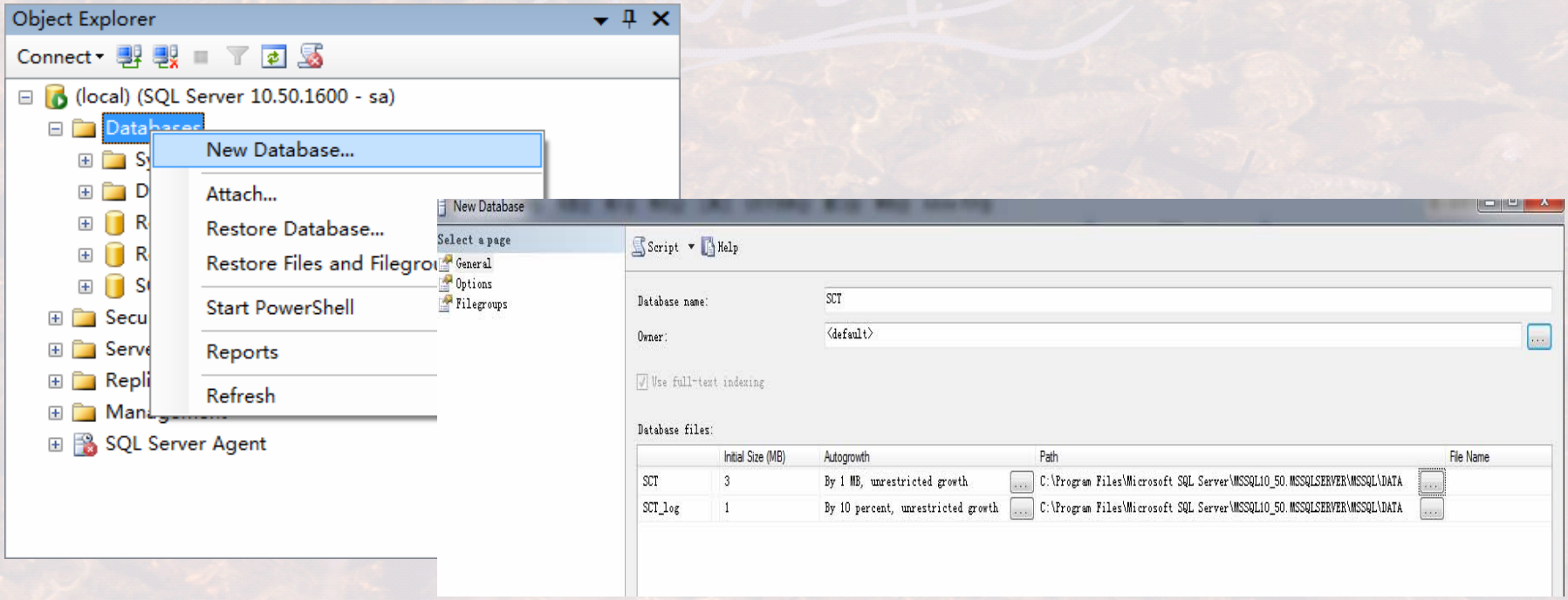


战德臣 教授

创建数据库

➢创建数据库的过程就是为数据库设计名称、设计所占用存储空间和存 放文件位置的过程。特别是在网络数据库中，对数据库的设计显得尤为重要。如估计数据可能占用的磁盘空间有多大，日志文件及其他要占用多大空间。

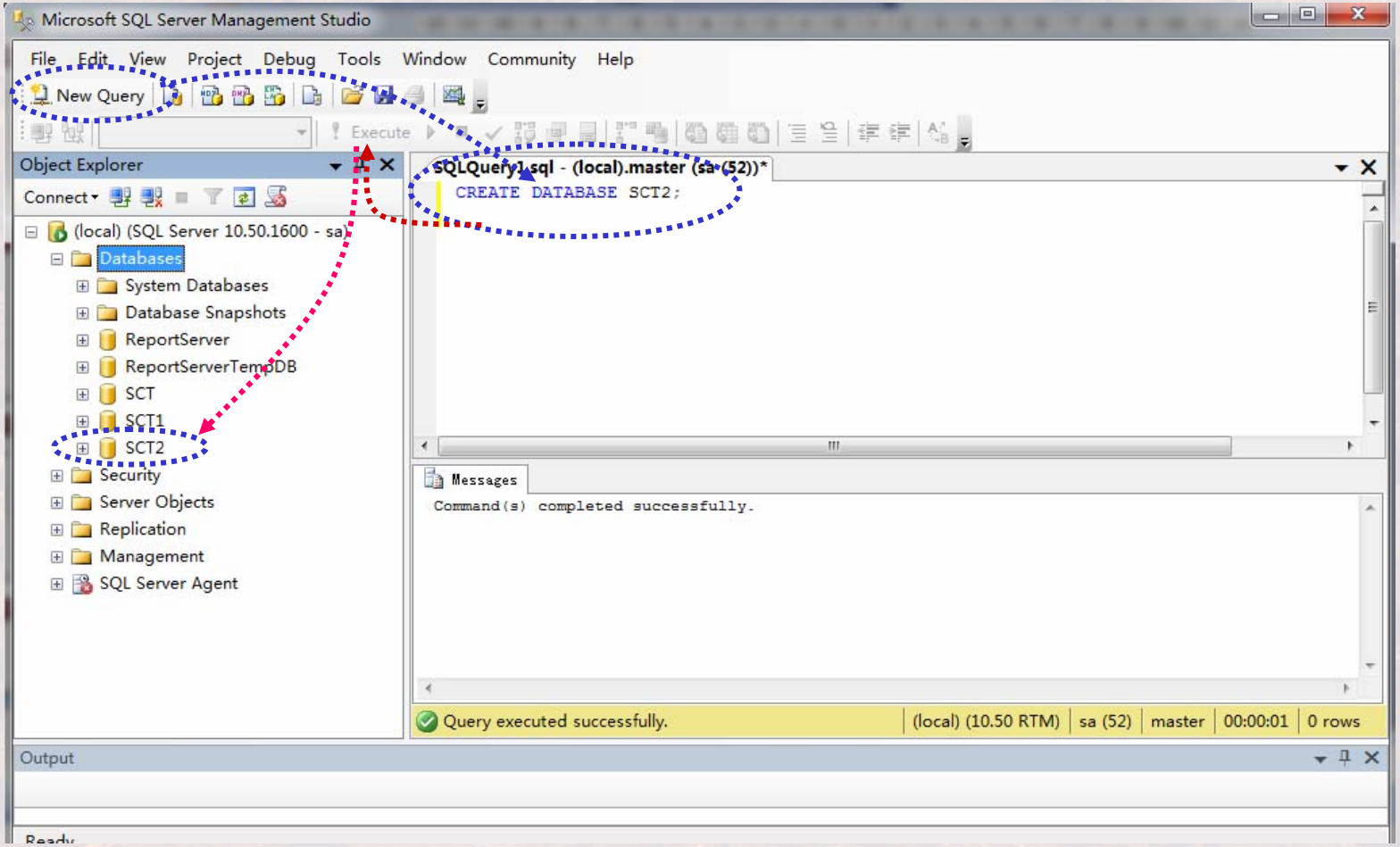
➢创建数据库的用户自动成为数据库的拥有者。



战德臣 教授

创建数据库

➢通过查询分析器或者交互式数据库查询引擎



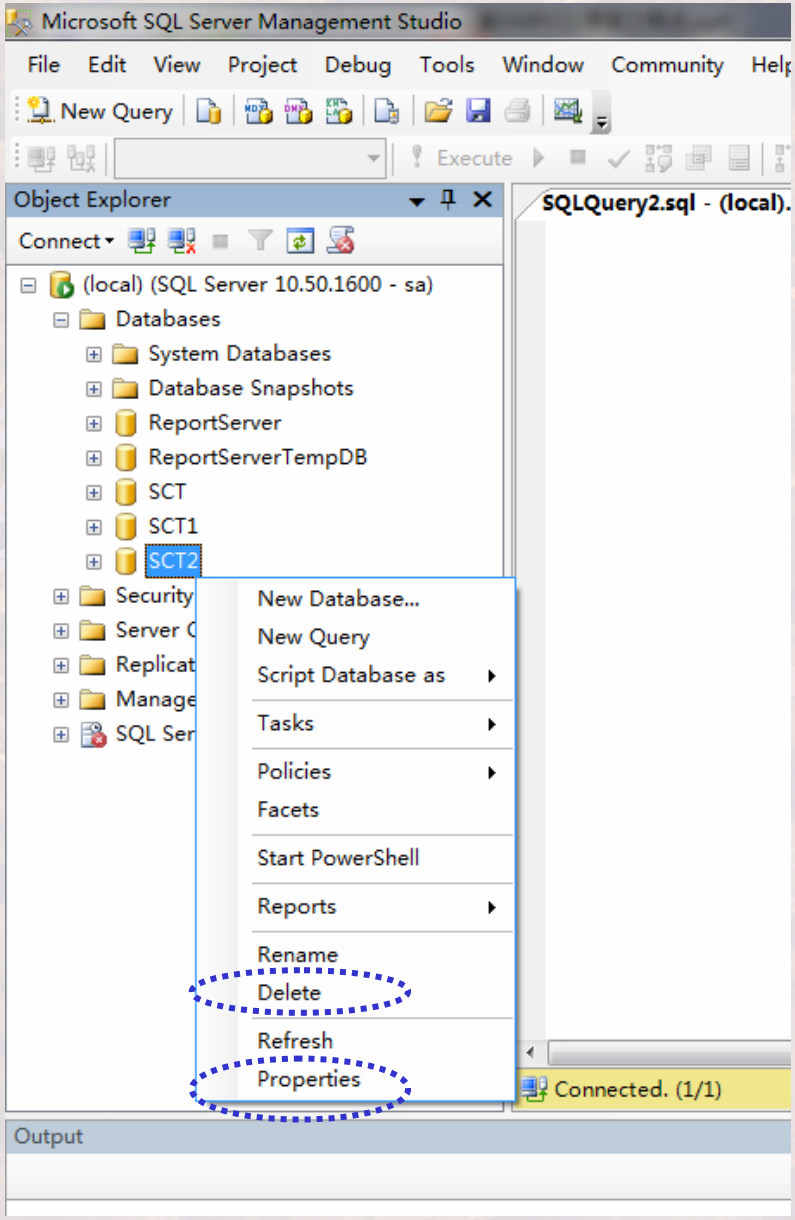
战德臣 教授

删除数据库

➢对不再需要的数据库，应删除以释放空间。删除的结果将是所有数据库文件都一并被删除。

➢当数据库处于正在使用或正在恢复状态时，不能删除。

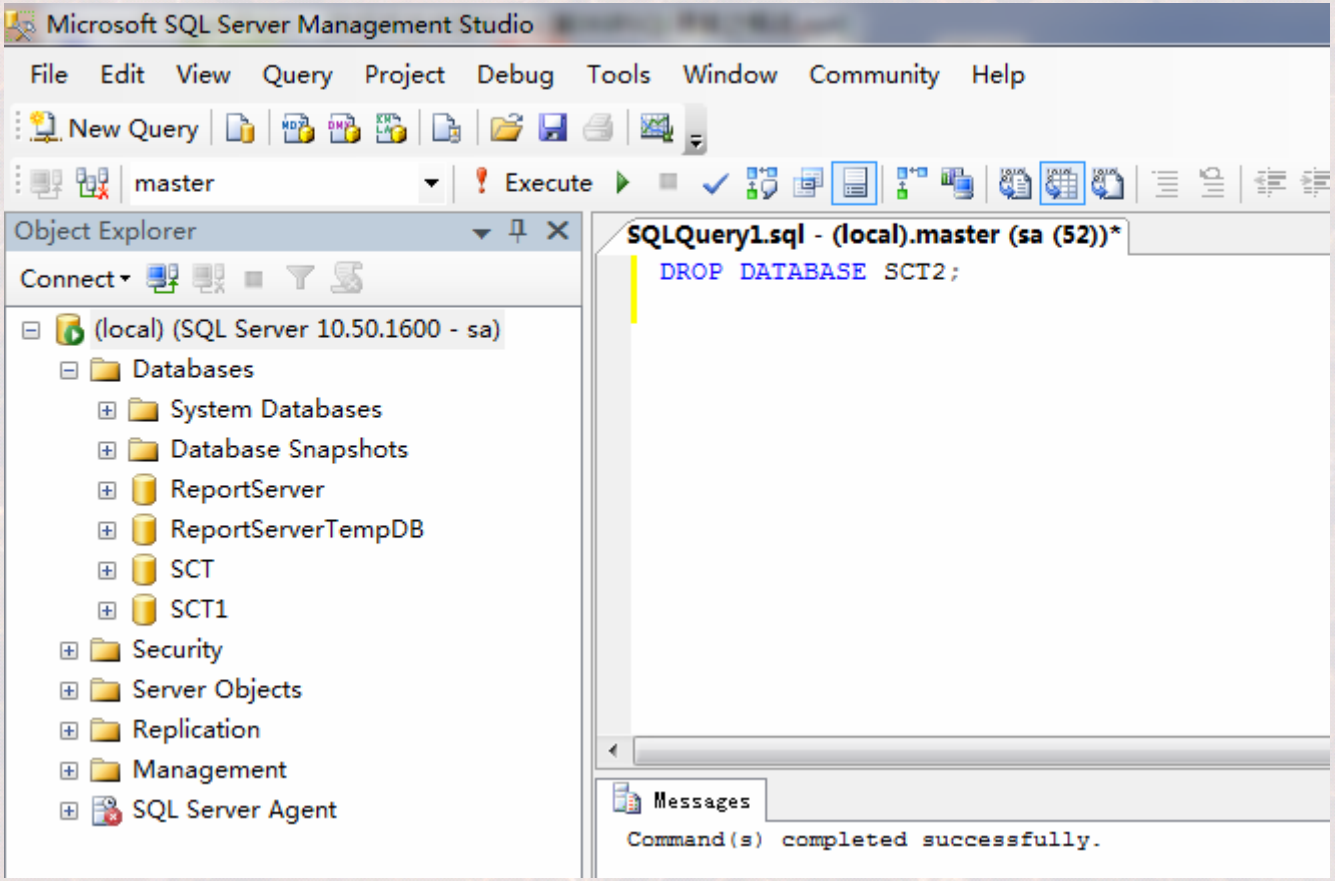
➢查看数据库的属性，如文件存储位置等



战德臣 教授

删除数据库

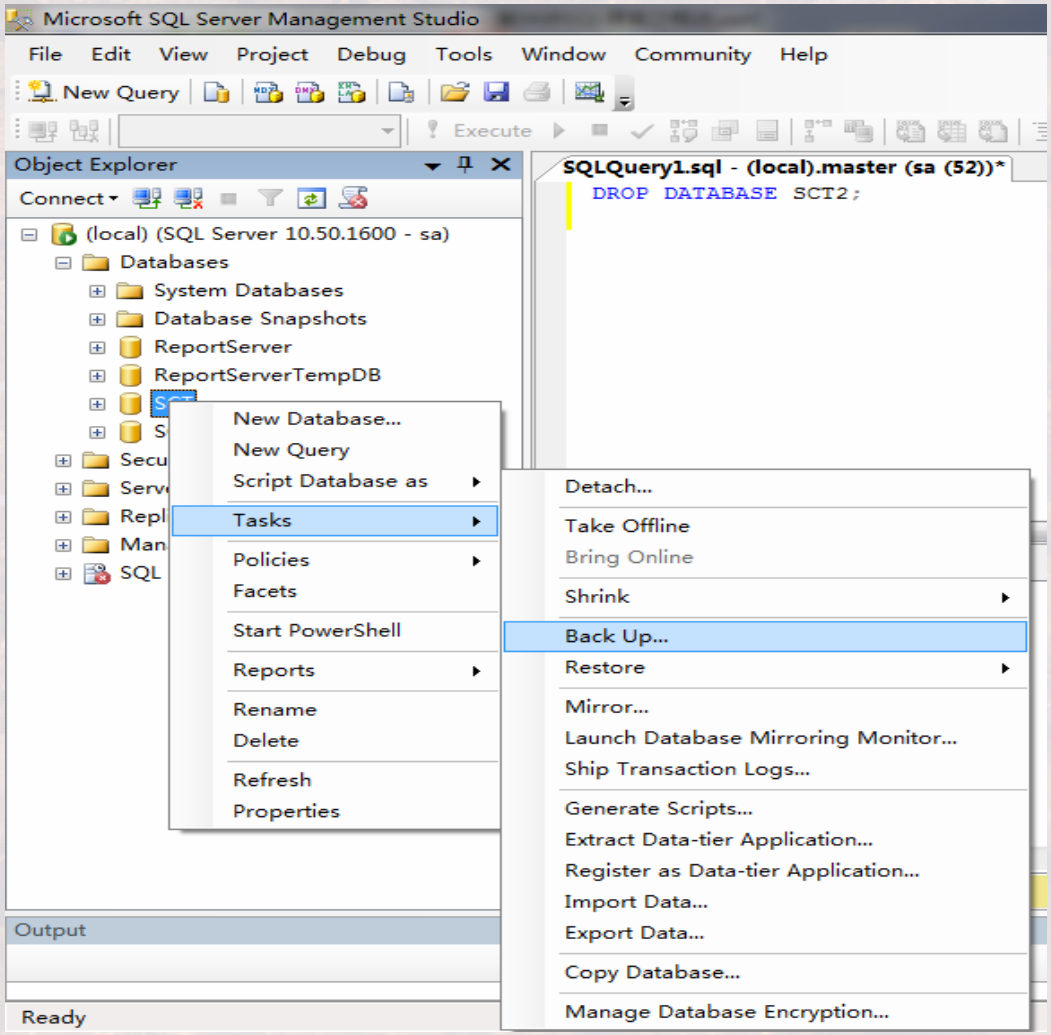
- 利用查询分析器或交互式数据库引擎删除数据库



战德臣 教授

备份数据库

➢备份就是对数据库或事务日志进行备份。SQL的备份是动态的，备份的过程还可以让用户继续改写。只有系统管理员、数据库的拥有者及数据库的备份者才有权限进行数据备份。可以通过企业管理器进行数据库备份。



战德臣 教授

备份数据库

◆ 完全数据库备份：完全备份数据文件和日志文件。

◆ 差异备份（增量备份）：对最近一次数据库备份以来发生的数据变化进行备份。这要在完全备份的基础上进行。特点是速度快。

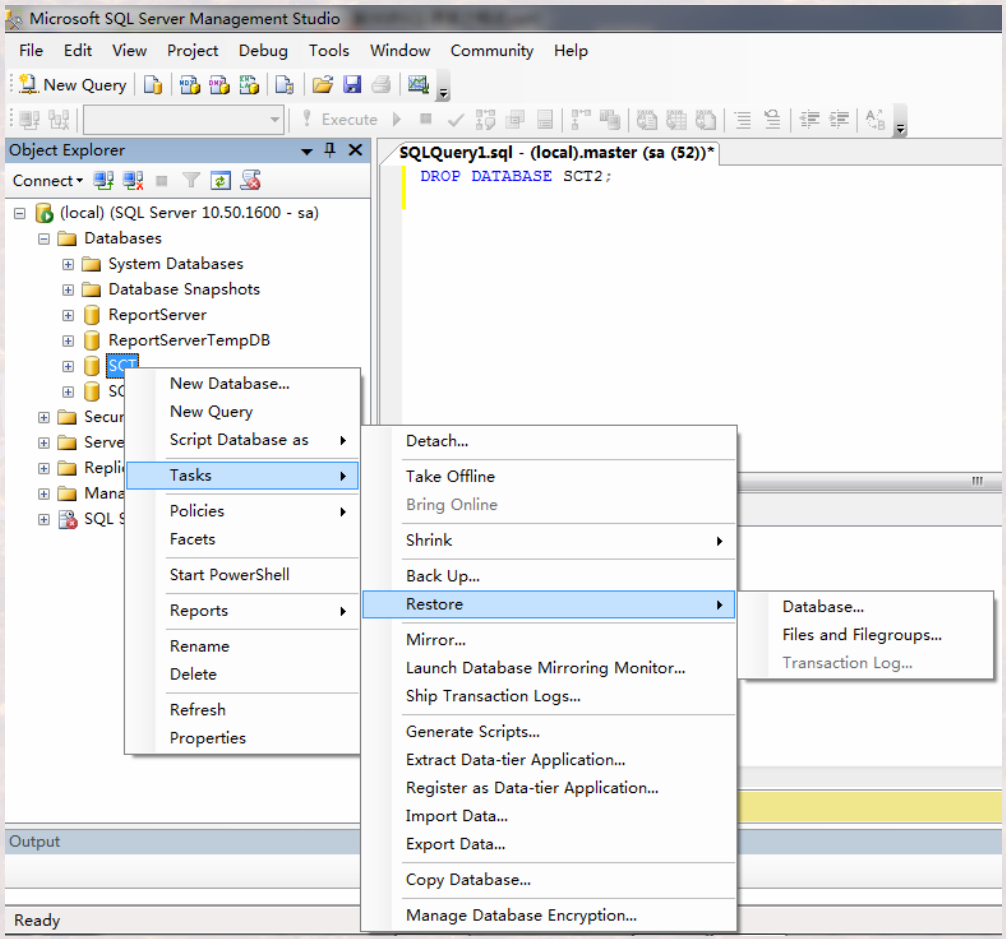
◆ 事务日志备份：对数据库发生的事务进行备份。包括从上次进行事务日志备份、差异备份和数据库完全备份之后，所有已经完成的事务。能尽可能的恢复最新的数据库记录。特点是所需磁盘空间小，时间少。

◆ 数据库文件和文件组备份：用在数据库相当大的情况下。

战德臣 教授

恢复数据库

- 数据库的恢复是指将数据库备份加载到系统中的过程。在根据数据库备份文件恢复过程中，系统将自动执行安全性检查、重建数据库结构及完成填写数据库内容。
- 数据库的恢复是静态的。所以在恢复前，应将需要恢复的数据库访问属性设为单用户，不要让其他用户操作。
- 可以通过企业管理器来完成数据库恢复。



战德臣 教授

数据库授权

◆ 语法形式

grant 权限 on 表名 to 用户名

◆ 权限有：select, update, insert , delete, exec, dri。

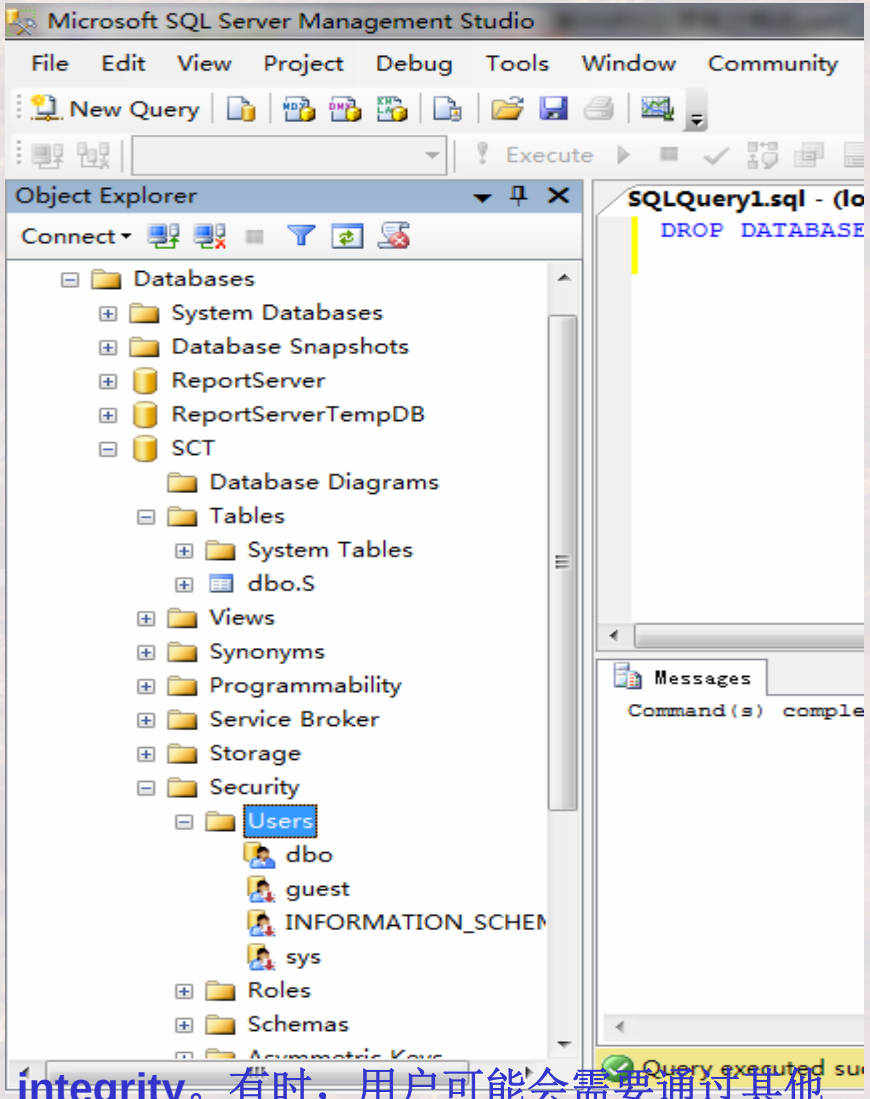
◆ 对被授权的用户，要先成为该数据库的使用者，即要把用户加到数据库里,才能授权.

■ 数据库授权实例

◆ 添加用户

◆ 对用户授权

后面还要讲的哟...



dri, 声明引用完整性declarative referential integrity。有时，用户可能会需要通过其他的表来检验其输入的数据，但又不能直接地读取这个表，在这种情况下，用户就需要DRI权限而不是Select权限。

战德臣 教授

创建表

➢同一用户不能建立同一个表名的表，同一表名的表可有多个拥有者。但在使用时，需要在这些表上加上所有者的表名。

➢用T-SQL语句创建表， 语法形式：

CREATE TABLE [数据库名.所有者名.]表名 ({<列名 数据类型>} [缺省值] [约束][是否为空] …)

➢注意：T-SQL是SQL Server软件的SQL语言，与标准版有些差异。但标准版SQL，一般情况下SQL Server软件也都支持

示例：应用查询分析器创建表实例

✓在查询分析器中输入 **use SCT;**

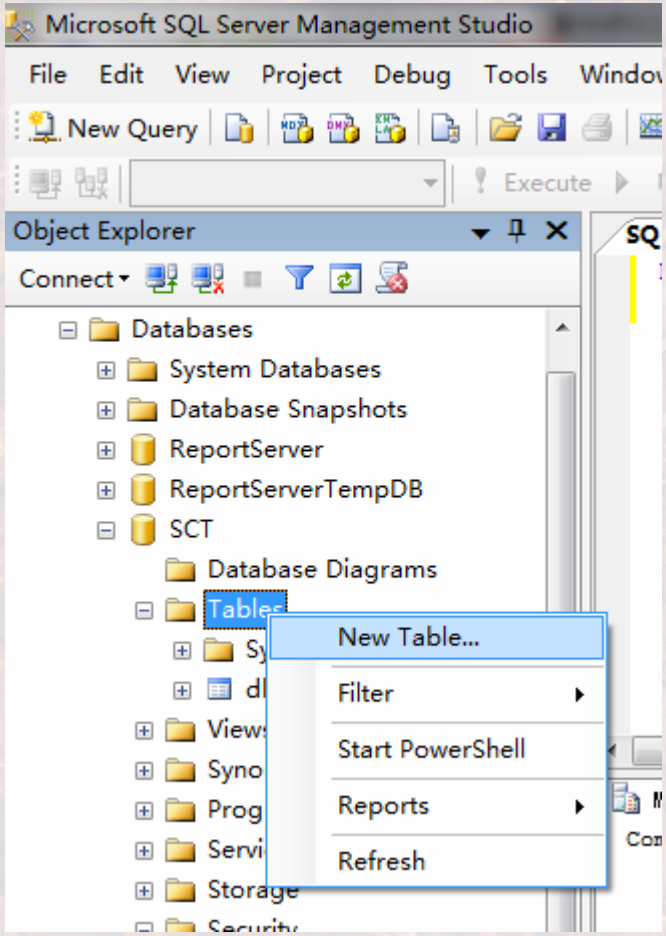
✓在查询分析器中输入

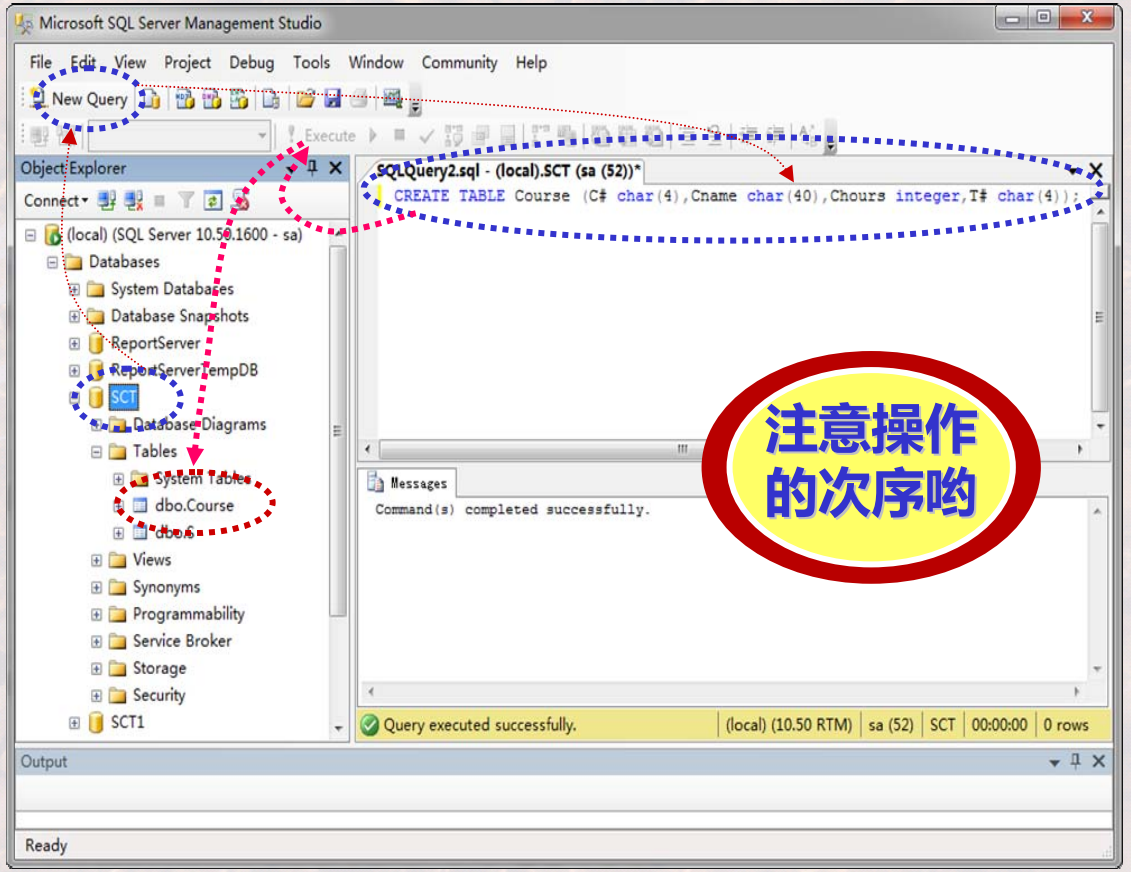
Create Table Student (S# char(8) not null , Sname char(10), Ssex char(2), Sage integer, D# char(2), Sclass char(6));

✓在菜单中选择执行查询；

战德臣 教授

示例：应用查询分析器创建表实例

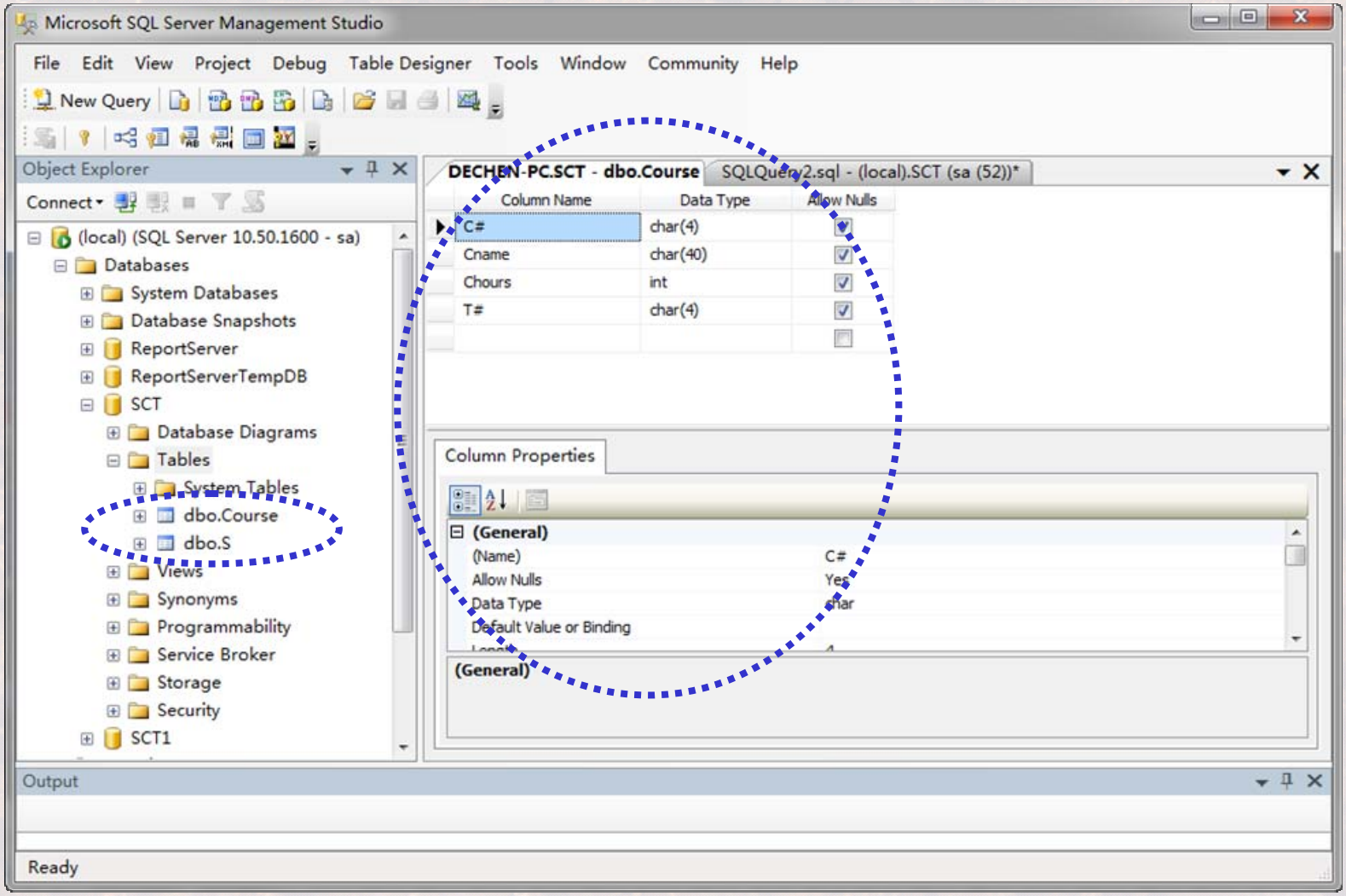




典型DBMS交互环境简介-SQL Server
(3)数据表的创建-与增/删/改/查

战德臣 教授

示例：应用查询分析器创建表实例



战德臣 教授

增加、修改表字段

语法形式：ALTER TABLE ADD | ALTER 字段名 <类型>

示例： alter table student
add Sadress char(40);

alter table student modify Ssex int not null;

战德臣 教授

创建、删除与修改约束

◆约束是SQL提供自动保持数据库完整性的一种方法，共5种。

◆用T-SQL语句建立约束，语法形式：

CONSTRAINT 约束名 约束类型 (列名)

约束名：在库中应该唯一，如不指定，系统会给出

约束类型 (5种)：

- primary key constraint (主键值)
- unique constraint (唯一性)
- check constraint (检查性)
- default constraint (默认)
- foreign key constraint (外部键)

列名：要约束的字段名

后面还会讲授的...

➢ 创建主键约束实例

```
Create Table Course ( C# char(3) , Cname char(12), Chours integer,
        Credit float(1), T# char(3) ) constraint pk primary key(C# ));
```

战德臣 教授

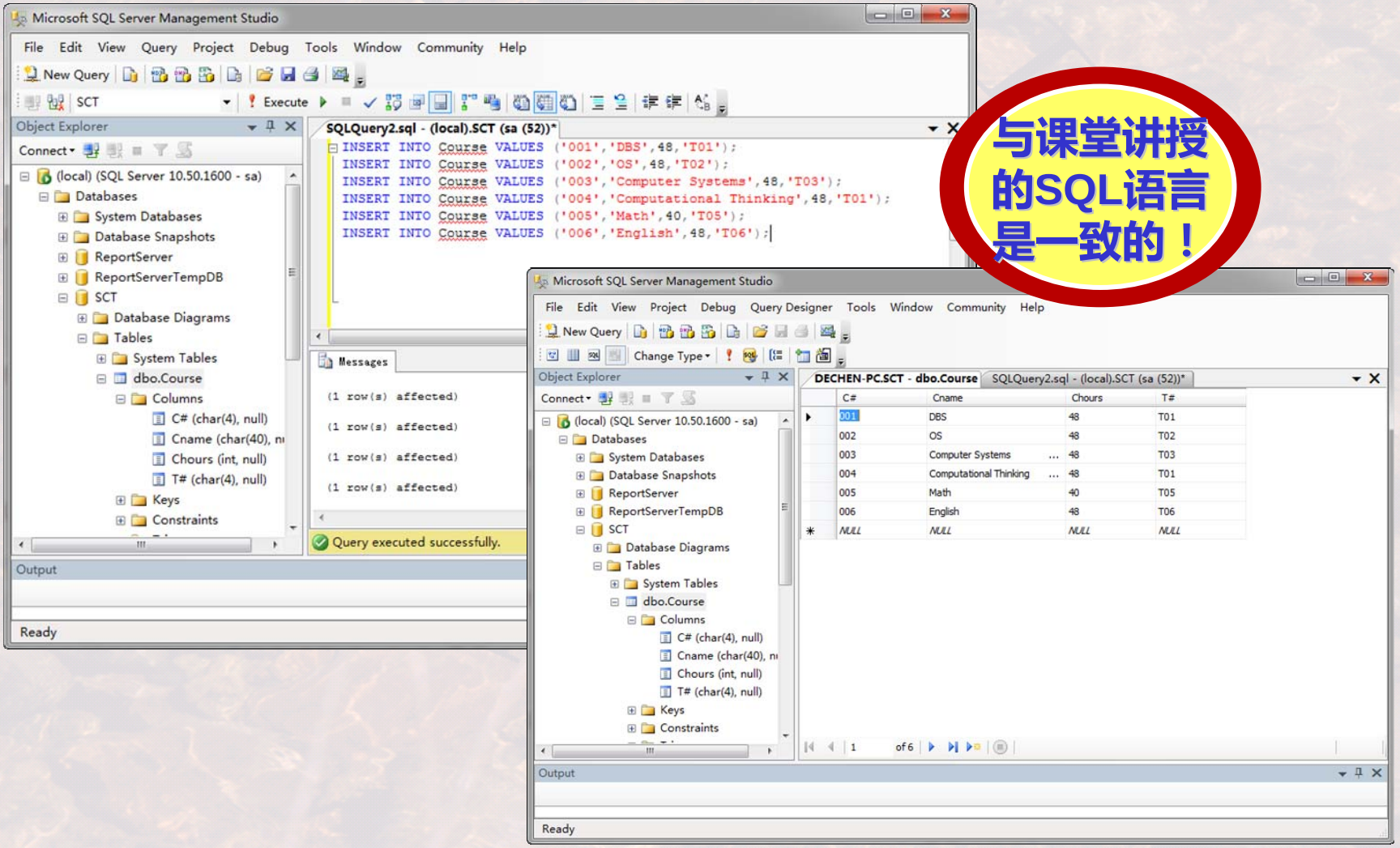
在表中插入数据

示例：在查询分析器中输入以下语句

```
Insert Into  Student
Values ( '98030101' , '张三', '男', 20，  '03'，  '980301');

Insert Into  Student ( S#, Sname, Ssex, Sage, D# , Sclass)
Values ( '98030102' , '张四', '女', 20，  '03'，  '980301');

Insert Into  Course
Values ( '001' , '数据库', 40, 6，  '001');

Insert Into  Course(Cname, C#, Credit, Chours, T#)
Values ('数据库', '002', 7, 30, '002');
```

战德臣 教授

在表中插入数据



典型DBMS交互环境简介-SQL Server
(4)开始SQL语言的书写训练吧

战德臣 教授

在查询分析器中输入SELECT-FROM-WHERE语句并查看查询结果