1. Overview of the SQL Query Language  
   包含：  
   DDL   
   DML   
   DDL的功能：  
   Integrity   
   View definition  
   Transaction control  
   Embedded SQL and dynamic SQL  
   Authorization
2. SQL Data Definition
   1. 基本数据类型：char(n)、varchar(n)、int、smallint、numeric(p,d)、real,double precision、float(n)   
      int smallint real float依赖机器精度
      1. char(n)不够的用空格补齐
         1. 比较两个char(n)补齐成一样的长度
         2. 比较char和varchar 有的数据库会补齐，有的不会，所以最好都用varchar
      2. Nvarchar 存储Unicode字符
   2. Basic schema Definition
      1. Create table department (dept\_name varchar(20), budget numeric(12,2), primary key(dept\_name));  
           
         Create table r   
         (A1, D1,  
         … ,  
         <integrity-constraint1>,  
         <integrity-constraint2>,  
         …);  
         常用的一致性约束类型有：主键、外键、非空
3. Basic Structire of SQL Queries
   1. Queries on a Single Relation  
      基本结构由select from和where 组成  
      去重select distinct dept\_name from instructor  
      计算 select salary\*1.1 from instructor
   2. Queries on Multiple Relations  
      where
   3. The Nature Join
   5. Set Operation  
      并集union、交集intersect、差集except  
      查询2009年秋季开课的课程和2010年春季开课课程分别为：  
      select course\_id  
      from section  
      where semester=’Fall’ and year=2009  
      和  
      select course\_id  
      from section  
      where semester=’Spring’ and year=2010  
      要得出两个季度所有的课程可以用Union；使用intersect可以查找到两个季度都开课的课程；而使用except可以得到第一个结果集中存在但第二个结果集不存在的内容，这三种操作如果不需要去重，可以对应使用union all, intersect all, except al。
   6. Null  
      null与其它值类型的算术运算结果都为null；  
      比较运算中，1<null的结果为unknow，这样除了“是”与“否”两种逻辑结果，有多了一个unknow；
      1. AND   
         true,unknown=unknown  
         false,unknown=false  
         unknown, unknown= unknown
      2. OR  
         true, unknown=true  
         false, unknown= unknown  
         unknown, unknown= unknown
      3. NOT  
         NOT unknown= unknown
   7. Aggregate Function聚合函数  
      avg min max sum count
   8. Nested Subqueries 嵌套子查询  
      子查询是嵌套在另一个查询中的select-from-where表达式，用于对集合的成员资格进行检查以及对集合的比较。
      1. Set Membership检查集合成员资格  
         比如前面用交集操作实现的查询也可以写为：  
         select course\_id  
         from section  
         where semester=’Fall’ and year=2009 and  
         course\_id in (select course\_id  
         from section  
         where semester=’Spring’ and year=2010)  
         可见SQL实现同一查询目的的方法可以是多样的。
      2. Set Comparison集合的比较  
         集合比较用到的写法有>some, >=some, =some, >all等，比如要查找比生物系中至少一位教师工资高的人，可以写为：  
         select distinct T.name  
         from instructor as T, instructor as S  
         where T.salary > S.salary and S.dept name = ‘Biology’  
         也可以使用>some的写法：  
         select name   
         from instructor   
         where salary > some (select salary   
          from instructor   
          where dept name = ‘Biology’);
      3. Test for Empty Relations空关系测试  
         可以用exist来测试关系中是否存在元组，对应还有not exist  
         前面要查询的2009秋季和2010春季都开课的课程，也可以写为：  
         select course id   
         from section as S   
         where semester = ‘Fall’ and year= 2009 and   
          exists (select \*   
          from section as T   
          where semester = ‘Spring’ and year= 2010 and S.course id= T.course id);
      4. Test for the Absence of Duplicate Tuples测试重复元组  
         使用unique来检查关系中是否存在重复元组，对应也有not unique。比如要查找2009年秋季至多开课一次的课程：  
         select T.course id   
         from course as T   
         where unique (select R.course id   
          from section as R   
          where T.course id= R.course id and R.year = 2009);  
         对于当时没开课的课程，因为结果为empty，unique对empty的计算结果也是true
      5. Subqueries in the From Clause From子句中的子查询  
         在from子句中也可以使用子查询，因为任何select-from-where返回的结果都是关系，所以可以在其上面继续使用from子句。  
         查询平均薪水超过42000的部门，如果使用having子句可以是：  
         select dept name, avg (salary) as avg\_salary   
         from instructor   
         group by dept name   
         having avg (salary) > 42000;  
         也可以采用From子查询的方式：  
         select dept name, avg\_salary   
         from (select dept name, avg (salary) as avg salary   
          from instructor   
          group by dept name)   
         where avg\_salary > 42000;  
         同时还可以为from子查询的的表和字段重命名：  
         select dept name, avg\_salary   
         from (select dept name, avg (salary)   
          from instructor   
          group by dept name)   
          as dept\_avg (dept name, avg\_salary)   
         where avg salary > 42000;
      6. The with Clause With子句  
         with子句用来定义临时关系，这个定义只对包含with子句的查询有效。比如查询拥有最多预算的部门，可以使用子查询，但子查询往往结构复杂、可读性差，而使用with子句就会好很多：  
         with max budget (value) as   
          (select max(budget)   
          from department)   
         select budget   
         from department, max budget   
         where department.budget = max budget.value;  
         虽然with子句只能在紧接着的查询中使用，但比子查询方便的是，它可以被多次使用。
      7. Scalar Subqueries 标量查询  
         标量查询是指返回结果只是一个值的子查询，比如查询每个部门的员工人数：  
         select dept\_name,   
          (select count(\*)   
          from instructor   
          where department.dept\_name = instructor.dept name)   
          as num instructors   
         from department;  
         由于使用了count，这儿的子查询结果只有一个值，虽然这仍然是一张表，但数据库会自动从表中取出值使用。标量查询可应用于select, where, having等处。而且编译时无法确保子查询结果确实是一个值，如果不是，在运行时会报错。
   9. Modification of the Database
      1. Delete
      2. Insert  
         插入数据时可以直接使用select的结果，但下面的写法会造成死循环，插入无限多条：  
         insert into student   
          select \*   
          from student;  
         而且数据库产品一般会提供批量插入的方式，用于快速地从格式化文本读取并插入大批量的数据。
      3. Update  
         更新数据时可以使用case when来区分不同的情况：  
         update instructor   
         set salary = case   
          when salary <= 100000 then salary \* 1.05   
          else salary \* 1.03   
          end  
         此外，set子句也可以使用子查询