断言

- 断言（assertion）是表达要求数据库永远满足的条件的谓词（复杂check条件）
- SQL 中的断言形式如下：

  create assertion <assertion-name> check <predicate>
- 创建了某断言之后，系统将检查它的合法性，并对每一个可能破坏该断言的数据库更新进行检测
  - 这种检测会产生大量的开销， 因此断言的使用应非常谨慎
- 由于SQL不提供“for all X, P(X) ”结构，我们可以通过迂回的方式表达：not exists X such that not P(X)

  ∵ $(\forall x)P(x) = \neg (\exists x) \neg P(x)$

断言

- 例1，对于*student*关系中的每个元组，它在属性*tot_cred*上的取值必须等于其所成功修完课程的学分总和

```
create assertion credits_earned_constraint check
        (not exists (select ID
          from student
          where tot_cred <> (select sum(credits)
                             from takes natural join course
                             where student.ID= takes.ID
                             and grade is not null and
                             grade <> ' F' );
```

断言

- 例2，每位教师不能在同一个学期的同一个时间段在两个不同的教室授课

```
create assertion ins_teaches_constraint check not exists
        (select ID, name, section_id, semester, year,time_slot_id,
                count(distinct building, room_number)
         from instructor natural join teaches natural join section
         group by (ID, name, section_id, semester, year, time_slot_id)
         having count(building, room_number) > 1)
```

触发器

- 触发器（trigger）是由数据库更新操作引起的被系统自动执行的语句
- 设计触发器必须:
  - 指明触发器被执行的条件
  - 指明触发器执行时所做的具体操作
- 引入触发器的SQL标准是SQL:1999，但多数数据库产品早已支持非标准语法的触发器

触发器

- 例，使用触发器来确保关系*section*中属性*time_slot_id*的参照完整性

```
create trigger timeslot_check1 after insert on section
        referencing new row as nrow
        for each row
        when (nrow.time_slot_id  not in
                  (select time_slot_id
                   from time_slot)) /* time_slot中不存在该
                                       time_slot_id */
        begin
             rollback
        end;
```

触发器

```
create trigger timeslot_check2 after delete on time_slot
        referencing old row as orow
        for each row
        when (orow.time_slot_id  not in
                    (select time_slot_id
                     from time_slot) /* 在time_slot 中刚刚被删除的time_
                                        slot_id */
        and orow.time_slot_id in
                    (select time_slot_id
                     from section)) /* 在section中仍含有该time_
                                       slot_id 的引用*/
        begin
           rollback
        end;
```

触发器

- 触发事件包括insert, delete和update
- 针对update的触发器可以指定具体修改的属性

  create trigger *takes_trigger* after update of *takes* on *grade*
- 更新前后的属性值可通过下列方法被引用
  - referencing old row as orow：对删除和修改有效
  - referencing new row as nrow：对插入和修改有效

语句级触发器

- 除了可以针对受影响的每一行执行一次单独的操作，也可以针对受到一个事务影响的所有行只执行一次操作
  - for each statement vs. for each row
  - 用referencing old table 或 referencing new table 来引用包含受影响的行的临时表
  - 对更新大量元组的SQL语句更高效

外部动作

- 有时要求数据库更新能触发外部动作
  - 例如当某种物品库存量小到一定程度就发订货单，或者打开报警灯
- 触发器不能直接实现外部动作，但是
  - 触发器可以在某个表中记录将采取的行动，而让另一个外部进程不断扫描该表并执行相应的外部动作
- 例，假设仓库库存有如下关系
  - *inventory*(*item*, *level*)：仓库中每种物品的库存量
  - *minlevel*(*item*, *level*)：每中物品的最小库存量
  - *reorder*(*item*, *amount*)：当物品小于库存量的时候要订购的数量
  - *orders*(*item*, *amount*) ：所下定单(由外部进程读取)

外部动作

```
create trigger reorder_trigger after update of level on inventory
     referencing old row as orow, new row as nrow
     for each row
     when nrow.level <=
            (select level
             from minlevel
             where minlevel.item = nrow.item) and orow.level >
                   (select level
                    from minlevel
                    where minlevel.item = orow.item)
     begin
        insert into orders
                 (select item, amount
                  from reorder
                  where reorder.item = orow.item)
     end
```

SQL Server触发器语法

```
create trigger timeslot_check1 on section
        after insert as
        if
        (inserted.time_slot_id  not in
                  (select time_slot_id
                   from time_slot)) /* time_slot中不存在该
                                       time_slot_id */
        begin
             rollback
        end;
```

inserted、deleted相当于前面的nrow和orow（成为过渡表）

Oracle触发器语法

```
create or replace trigger secure_student before insert or delete or
                                                    update on student
    begin
        if(to_char(sysdate, 'DY') in ('星期六', '星期日'))
        or(to_char(sysdate, 'HH24') not between 8 and 17 )
        then raise_application_error(-20506, '你只能在上班时间修改数据');
        end if;
    end;
```

- 注：运行该程序，实际是对其进行编译，若出错，可查看数据字典中user_errors的出错信息。user_triggers登记已建立了哪些触发器及定义内容
- 删除触发器：drop trigger <触发器名>

何时不用触发器

- 有时要求数据库更新能触发外部动作
  - 例如当某种物品库存量小到一定程度就发订货单，或者打开报警灯
- 早期触发器被用于如下任务
  - 维护综合数据（如，每门课的选课人数）
  - 复制数据库：记录特定关系(称为change或delta关系)的变化并由一单独进程将此变化反映到所有副本
- 上述任务现在有更好的做法：
  - 现在的数据库提供内建的物化视图来维护综合数据
  - 现代的数据库系统提供内置的数据库复制工具

讨论

- 比较一下三者的区别：
  - check
  - assertion
  - trigger