

Database Management Systems

Lab 7

Oracle Indexing
and Oracle PL/SQL Exceptions

Lab Topics

- ❑ How to manage an index for a table in **Oracle Database**.
- ❑ **PL/SQL** exceptions and exception handling.

在一些**ppt**页面上方，我们给出了相关内容详细介绍的网址。

Managing Indexes

■ Create index

create **[unique]** **index** *index_name* **on** *table_name* (*attribute_name1*, *attribute_name2*, ...);

Example: Create an index on **admissionYear** of the **Student** table.

create index AdmissionYearIndex **on** Student (admissionYear);

Example: Create an index on **courseName** of the **Course** table.

create unique index CourseNameIndex **on** Course (courseName);

The **unique** keyword specifies that the attribute **courseName** must have unique values.

■ Drop index

drop index *index_name*;

■ Display information about an index

select *index_name* **from** user_indexes;

RECALL: Oracle PL/SQL Basic Structure

create or replace procedure *procedure_name* [**as** | **is**]

Declarative section: declaration of variables, types, and local subprograms go here.

begin **Executable section:** procedural and **SQL** statements go here. This is the only required section of the block.

exception **Exception handling section:** error handling statements go here.

end;

Oracle PL/SQL Exceptions

http://docs.oracle.com/cd/B10501_01/appdev.920/a96624/07_errs.htm

- ❑ Predefined exceptions are raised *implicitly* by **Oracle PL/SQL** if the exception occurs. 系统定义的异常类型，用户不需要指定异常触发条件，会自动报错。
- ❑ User-defined exceptions are raised *explicitly* by the command. 用户自定义异常类型，需要用户自己指定异常出发调教，并且手动触发异常如下：

raise *exception_name*;

Predefined Exceptions

ACCESS_INTO_NULL	ORA-06530
CASE_NOT_FOUND	ORA-06592
COLLECTION_IS_NULL	ORA-06531
CURSOR_ALREADY_OPEN	ORA-06511
DUP_VAL_ON_INDEX	ORA-00001
INVALID_CURSOR	ORA-01001
INVALID_NUMBER	ORA-01722
LOGIN_DENIED	ORA-01017
NO_DATA_FOUND	ORA-01403
NOT_LOGGED_ON	ORA-01012
PROGRAM_ERROR	ORA-06501
ROWTYPE_MISMATCH	ORA-06504
SELF_IS_NULL	ORA-30625
STORAGE_ERROR	ORA-06500
SUBSCRIPT_BEYOND_COUNT	ORA-06533
SUBSCRIPT_OUTSIDE_LIMIT	ORA-06532
SYS_INVALID_ROWID	ORA-01410
TIMEOUT_ON_RESOURCE	ORA-00051
TOO_MANY_ROWS	ORA-01422
VALUE_ERROR	ORA-06502
ZERO_DIVIDE	ORA-01476

User-defined Exceptions

- ❑ Declare the exception in the **declaration** section.

myException **exception**;

- ❑ Raise it within a **begin...end** block.

if *condition* **then**

raise *myException*;
end if;

- ❑ Define the exception-handling code in the **exception** section within the **begin...end** block.

exception

when *myException* **then**
 :
 :

```
/* DECLARATION SECTION */  
myException exception;  
:  
begin  
:  
    begin  
        :  
        if condition then  
            raise myException;  
        endif;  
        :  
        exception  
            when myException then  
                /* Code to handle exception */  
                :  
            end;  
    end;  
:  
end;
```

Continuing After An Exception (1)

```
/* DECLARATION SECTION */
```

```
⋮
```

```
begin
```

```
⋮
```

```
begin
```

```
⋮
```

```
exception
```

```
when invalid_cursor
```

```
then
```

```
/* Code to handle  
exception */
```

```
⋮
```

```
when value_error then
```

```
/* Code to handle  
exception */
```

```
⋮
```

```
end;
```

```
⋮
```

```
end;
```

□ 如果程序发生异常，则程序运行跳转到异常处理语句。

□ 异常处理完毕后，跳转到exception block后面的语句。

Continuing After An Exception (2)

declare

peRatio **number**(3,1);

nvl(表达式1, 表达式2):如果表达式1为空值, nvl返回值为表达式2的值, 否则返回表达式1的值。

begin

delete from stats **where** symbol = 'xyz';

begin -- sub-block begins

-- The select statement will throw an exception if nvl(earnings, 0) is zero

select price / nvl(earnings, 0) **into** peRatio **from** stocks **where** symbol = 'xyz';

exception

when zero_divide **then**

peRatio := 0;

After the `zero_divide` exception is handled, execution continues with the `insert` statement, which is outside the inner `begin...end` block.

end; -- sub-block ends

insert into stats (symbol, ratio) **values** ('xyz', peRatio);

exception

when others then

⋮

end;

Note: `others`指代其他异常类型

Exception Handling Examples

```
OVERALLAVERAGEGRADEREPORT
Code Grants Dependencies Errors References Details Profiles

1 create or replace procedure OverallAverageGradeReport authid current_user as
2   currentStudentId      Student.studentId%type;
3   sumGrades              number(5,1);
4   avgGrade               EnrollsIn.grade%type;
5   numCourses             int;
6   failingStudent         exception;
7   -- Cursor to fetch Student records
8   cursor studentCursor is select studentId, firstName, lastName from Student order by lastName;
9   -- Cursor to fetch EnrollsIn records for a specified student
10  cursor enrollsInCursor is select grade from EnrollsIn where studentId=currentStudentId;
11  begin
12    for studentRecord in studentCursor loop
13      numCourses := 0;
14      sumGrades := 0;
15      currentStudentId := studentRecord.studentId;
16      begin -- sub-block begins
17        -- Get the number of courses and sum of the grades for a student
18        for enrollsInRecord in enrollsInCursor loop
19          numCourses := numCourses + 1;
20          sumGrades := sumGrades + enrollsInRecord.grade;
21        end loop;
22        -- Calculate the average grade
23        avgGrade := sumGrades/numCourses;
24        if avgGrade < 50 then raise failingStudent; end if;
25        dbms_output.put_line(chr(9) || studentRecord.firstName || ' ' || studentRecord.lastName
26          || 's overall average grade is ' || avgGrade || '.');
27      exception
28        -- Handle divide by zero error
29        when zero_divide then
30          dbms_output.put_line('No courses' || chr(9) || studentRecord.firstName || ' '
31            || studentRecord.lastName || ' is not enrolled in any course. ');
32        -- Handle failing student
33        when failingStudent then
34          dbms_output.put_line('** Failing **' || chr(9) || studentRecord.firstName || ' '
35            || studentRecord.lastName || 's overall average grade is ' || avgGrade || '.');
36      end; -- sub-block ends
37    end loop;
38  end OverallAverageGradeReport;
```

Declaring a user-defined exception

Checking for the user-defined exception

Handling an implicit exception

Handling the user-defined exception