

# SQL高级查询技术

单世民



大连理工大学

# 默认日期格式

- 在默认情况下，日期数据的格式是dd-mon-yy。其中，dd表示两位数字的日，例如10日。mon表示3位字符的月份，在英文版的Oracle系统中，月份分别是相应英文月份名称的简写形式，例如Jun；而在中文版中，月份的表现形式是“x月”或“xx月”，例如“8月”。yy表示两位数字年，例如90年和06年。
- 在插入数据时，默认情况下，应该按照dd-mon-yy格式插入数据。系统会自动将两位数字年识别成4位数字年。
- 在中文版的Oracle Database 10g系统中，应该按照“x月”或“xx月”格式输入月份。
- 在插入数据时，如果使用date关键字，那么可以采用yyyy-mm-dd的格式输入日期数据。其中，yyyy表示4位数字年，mm表示两位数字月，即1~12月，dd表示两位数字日。

# 设置默认的时间格式

- 日期数据格式由nls\_date\_format系统参数来设置。系统参数存储在init.ora文件和spfile.ora文件中。可以使用show parameters命令查看这些系统参数的值。如果需要改变这些系统参数的值，可以通过执行alter system命令或alter session命令来实现。

# 设置默认的时间格式

- **ALTER SYSTEM命令**  
修改系统参数的文件，这种修改设置在今后的数据库操作中一直起作用。
- **ALTER SESSION命令**  
设置只在当前的会话中起作用，该会话结束之后，其设置就会失效。

# 设置默认的时间格式

- 使用alter session命令设置时间的语法格式如下：

```
ALTER SESSION SET NLS_DATE_FORMAT='date_format';
```

- 在上面的语法格式中，date\_format参数既可以采用month-dd-yyyy格式，也可以采用yyyy-month-dd格式，还可以采用yyyy-dd-month格式。

```
ALTER SESSION SET NLS_DATE_FORMAT='month-dd-yyyy';
```

# 日期-字符转换

- 在Oracle Database 10g系统中，既可以把字符数据转变成日期数据，也可以把日期数据转变成字符数据。完成日期和字符数据之间的转换操作可以使用：

- ✧ TO\_CHAR()函数

- ✧ TO\_DATE()函数

# 日期-字符转换

- TO\_CHAR()函数可以完成从日期数据到字符串数据的转换过程。
- TO\_CHAR()函数的语法格式如下：

```
TO_CHAR(x[, format])
```

- 在上面的语法格式中，主要参数的意义如下：
  - ✧ *x*—将要转换的源日期表达式类型。
  - ✧ *format*—将要转变的源日期表达式的格式类型。

```
TO_CHAR(sysdate, 'yyyy, month dd, HH24:MI:SS')
```

# 日期-字符转换

- TO\_DATE()函数可以把指定的字符串转换成一个日期和时间。
- TO\_DATE()函数的语法格式如下：

```
TO_DATE(x[, format])
```

- 在上面的语法格式中，该函数将字符串表达式*x*按照由*format*指定的格式转变成日期和时间。如果没有使用*format*日期格式参数，则按照默认的日期格式进行转换。

```
TO_DATE('2008-09-30', 'yyyy-mm-dd')
```



# 日期时间函数

- Oracle Database 10g系统提供了许多用来处理日期和时间的函数

表 7-2 日期和时间函数

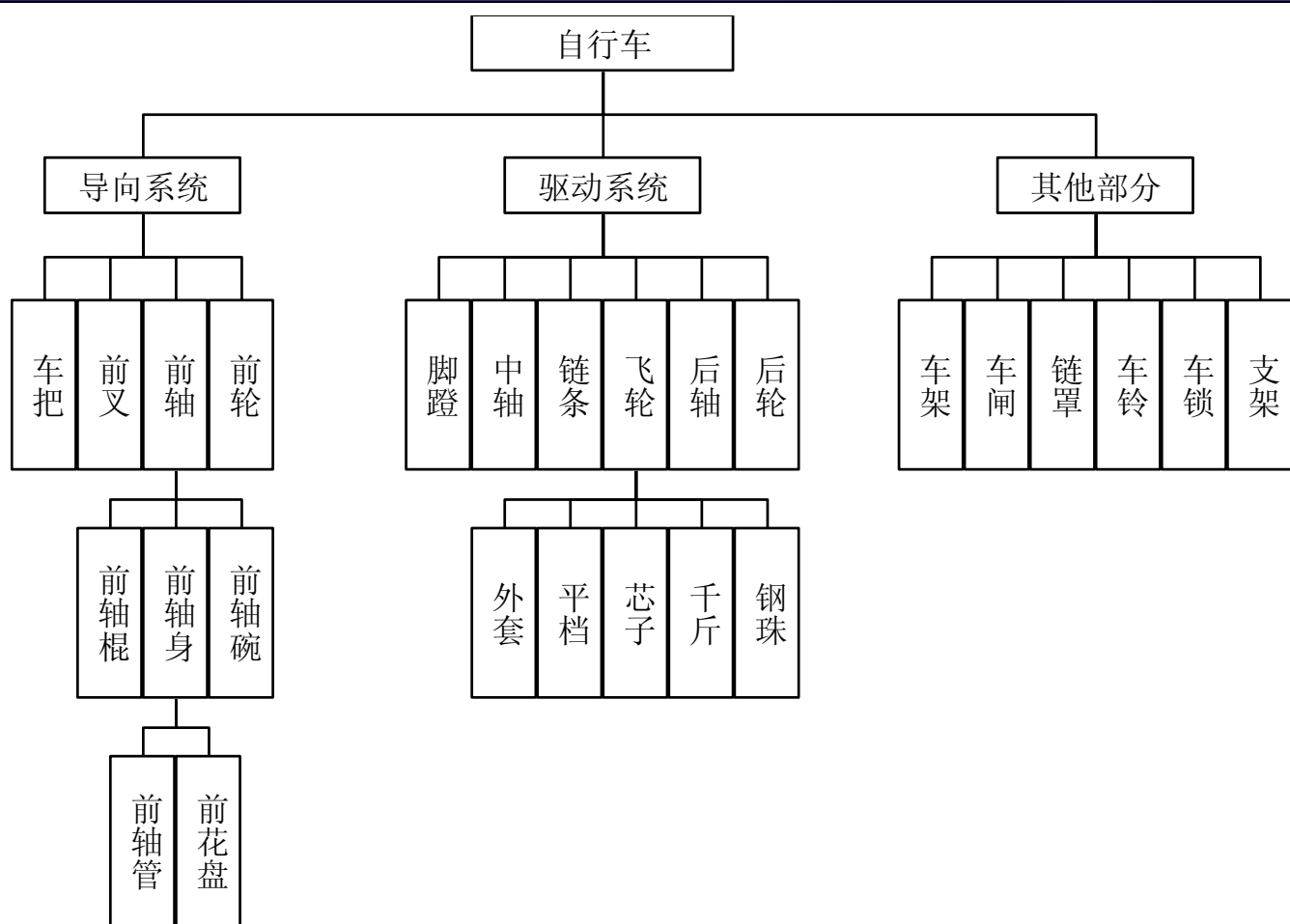
函 数	描 述
<code>add_months(x, y)</code>	在 x 上增加 y 个月。如果 y 是负数，表示从 x 中减去 y 个月
<code>last_day(x)</code>	返回包含在 x 中的月份的最后一天
<code>months_between(x, y)</code>	返回 x 和 y 之间的月数。从日历上来看，如果 x 在 y 之前，返回负数；否则返回正数
<code>next_day(x, day)</code>	返回紧接着 x 的下一天，day 是一个字符串
<code>round(x[, unit])</code>	圆整 x。在默认情况下，x 被圆整到最接近的一天。可以使用 unit 参数提供圆整的单位，例如 mm 表示圆整到最接近月的第一天
<code>sysdate()</code>	返回当前系统的日期
<code>trunc(x[, unit])</code>	截断 x。在默认情况下，x 被截断为当天的开始。可以使用 unit 参数提供截断的单位，例如 mm 表示截断到该月的第一天

# 层次查询

- 在许多实际的应用中，数据之间的关系是层次关系。例如，图书编码、由零部件组成的汽车结构、组织结构和家族谱系等都是典型的层次结构关系。在各种的ERP应用中，产品结构都是由一种树状的层次结构。
- Oracle Database 10g系统提供了层次查询的功能。用户利用这些功能可以很方便地查询各种层次结构的数据。

# 层次查询

- 自行车层次示例



# 层次查询

- bicycle表的结构

```
CREATE TABLE bicycle(  
    part_id number(5)  
        constraint pk_bicycle_part_id primary key,  
    parent_id number(5)  
        constraint fk_bicycle_pid  
            references bicycle(part_id),  
    part_name varchar2(30) not null,  
    mp_cost number(9, 2),  
    describe varchar2(30)  
);
```

# 层次查询

- 为了执行层次查询，需要在select语句中添加 **start with** 子句和 **connect by** 子句。添加这些子句后的select语句的语法格式如下：

```
SELECT level, column_name, expression, ...  
FROM table_name  
[WHERE where_condition]  
START WITH start_condition  
CONNECT BY PRIOR prior_condition;
```

# 层次查询

```
SELECT part_id, parent_id, part_name, mp_cost  
FROM bicycle  
START WITH part_id = 1  
CONNECT BY PRIOR part_id = parent_id;
```

```
SELECT level, part_id, parent_id, part_name, mp_cost  
FROM bicycle  
START WITH part_id = 1  
CONNECT BY PRIOR part_id = parent_id  
ORDER BY level;
```



*Connect by*子句中可以使用复合条件

# 层次查询

- 层次查询中使用level伪列，该伪列表示树状结构的层次。执行层次查询，并且使用level伪列，其中order by子句用于对level列的数据进行排序。

# 情景查询

- 在执行select操作过程中，经常需要对不同的数据显示不同的结果。也就是说，在不同的情景下，需要显示不同的结果。这种情景查询非常类似于if...then...else程序流程控制语句。
- 在Oracle Database 10g系统中，可以使用decode()函数和case表达式完成情景查询。



# 情景查询

- 在Oracle Database 10g系统中，可以使用decode()函数执行类似于if...then...else程序流程控制语句的操作。
- decode()函数的语法格式如下：

```
DECODE (value_expression, search_value,  
        result_value, default_value)
```

# 情景查询

```
SELECT  ename,sal,  
        decode(deptno, '10', '会计部',  
                  '20', '研发部',  
                  '30', '销售部',  
                  '其他部门')  
FROM    emp;
```

# 情景查询

- 除了使用decode()函数执行情景查询之外，还可以使用case表达式执行情景查询。与decode()函数相比，case表达式的功能更加强大，使用更加灵活。

# 情景查询

- 在Database 10g系统中，有两种类型的case表达式，即简单case表达式和搜索case表达式。

```
CASE search_expression
  WHEN expression_1 THEN display_result_1
  WHEN expression_2 THEN display_result_2
  .....
  WHEN expression_n THEN display_result_n
  ELSE default_display_result
END
```

```
CASE
  WHEN expression_1 THEN display_result_1
  WHEN expression_2 THEN display_result_2
  .....
  WHEN expression_n THEN display_result_n
  ELSE default_display_result
END
```

# 翻译函数查询

- 除了前面介绍的查询之外，Oracle Database 10g系统还提供了一个有意义的翻译函数 `translate()`。顾名思义，可以使用 `translate()` 函数把指定的表达式翻译成另一个表达式。该函数类似于加密函数，可以使用一种密钥加密表达式，并且可以使用同样的密钥解密。因此，可以说 `translate()` 函数是一种对称加密方式。

# 翻译函数查询

- translate()函数的语法格式如下:

```
TRANSLATE (message_expression,  
           source_encode_key, destination_encode_key)
```

```
SELECT part_name,  
       mp_cost as actualCost,  
       TRANSLATE (mp_cost, 1234567890, 5129837046)  
         as translatedCost  
FROM bicycle;
```

# 分析查询

- 在执行分析查询过程中需要借助于分析函数。Oracle Database 10g系统提供了许多可以用于分析查询的分析函数。可以把这些分析函数分成以下几种类型。
  - ❑ **窗口函数**：允许用户在指定的窗口中计算累加值、移动值等运算；
  - ❑ **制表函数**：允许计算诸如市场份额、销售总计和销售比率等数据；
  - ❑ **第一个值和最后一个值函数**：可以在一组数据中得到第一个值和最后一个值；
  - ❑ **落后位置和提前位置函数**：可以使得行中的某一个列的数据位置落后或提前指定的距离；
  - ❑ **等级函数**：可以计算一组数据的等级、百分比等；
  - ❑ **线性回归函数**：可以计算最小方框回归线。

# 分析查询

- 分析查询示例
- 计算每个部门薪水的平均值，并将每个雇员薪水与该部门薪水的平均值比较

```
select deptno,ename,sal,  
       round(average_sal_dept,0) as average_sal_dept  
       round(sal-average_sal_dept,0) as sal_variance  
from (select deptno,ename,sal,  
            avg(sal)over(partition by deptno) as average_sal_dept  
      from emp);
```



# 分析查询

- 分析查询示例
- 查询本单位最先招聘的3个雇员

```
select rownum as rank,ename,hiredate
from (select ename,hiredate from emp order by hiredate
      nulls last)
where rownum<=3;
```

# 分析查询

- 分析查询示例
- 查询本单位最先招聘的3个雇员

```
select hire_rank,ename,hiredate
from (select ename,hiredate,
            rank()over(order by hiredate asc nulls last) hire_rank
      from emp)
where hire_rank<=3;
```

# 小结

- 日期的默认格式
- 日期和时间函数
- 层次查询技术
- 情景查询技术
- 翻译查询技术
- 分析查询技术

*The End*