

# 第一章

## 第1章 SQL 核心

### 1.2 数据库的接口

eg: C/C++程序块中嵌入SQL语句

```
{  
    int a;  
    /* ... */  
    exec sql select salary into :a from hr.employees where employee_id = 108;  
    /* ... */  
    printf("The salary is %d\n", a);  
    /* ... */  
}
```

### 1.3 SQL\*Plus 回顾

#### 1.3.2 配置SQL\*Plus环境

eg:

help index : 显示 SQL\*Plus命令列表

help set : 显示set 命令的用法

eg:

set lines 3000 : set width of display line

set pages 1000 : set number of lines per page

set timing on : sets display of elapsed time

set null <null> : sets display of nulls to show <null>

set SQLPROMPT '& user@@\_connect\_identifier>' : sets the prompt to show connected user and instance.

以下预定义变量可设置到SQL\*Plus命令行提示符:

\_connect\_identifier

\_date

\_editor

\_o\_version

\_o\_release

\_privilege

\_sqlplus\_release

\_user

eg: \$ORACLE\_HOME/sqlplus/admin/glogin.sql文件保存SQL\*Plus的设置。

eg:

define\_editor='</full path>/myeditor.exe' : 设置SQL\*Plus默认的文本编辑器，这样就可以在SQL\*Plus中以默认的编辑器编辑sql文件

### 1.5

eg: 查询下订单超过4此的女顾客

```
select c.customer_id, count(o.order_id) as orders_ct from oe.customers c join oe.orders o on c.customer_id = o.customer_id where  
c.gender = 'F' group by c.customer_id having count(o.order_id) > 4 order by orders_ct, c.customer_id;
```

#### 1.5.1

连接语句的顺序:

交叉连接 (笛卡尔积)

内连接

外连接

#### 1.5.3 Group By 子句

Group by 子句的 Rollup运算: 用来产生部分求和值

Group by 子句的 Cube运算: 用来求得交互分类值

### 1.6 INSERT 语句

#### 1.6.2 多表插入

eg:

```

insert all
when sum_orders < 10000 then
into small_customers
when sum_orders >= 10000 and sum_orders < 100000 then
into medium_customers
else
into large_customers
select customer_id, sum(order_total) sum_orders
from oe.orders
group by customer_id;

```

Note: all 子句执行无条件的多标插入。first子句实现每一个when子句按照其出现的顺序评估。

### 1.7 UPDATE语句

eg: update a table using a select statement to define the table and column values

```

update (select e1.salary, e2.salary new_sal from employees e1, employees e2 where e1.employee_id = e2.employee_id and
e1.department_id = 90) set salary = new_sal;

```

eg: update multiple columns using a subquery

```

update employees set (salary, commission_pct) = (select employees2.salary, .10 comm_pct from employees2 where
employees2.employee_id = employees.employee_id and employees.salary != employees2.salary) where department_id = 90;

```

### 1.8 delete 语句

eg: delete rows using a subquery in the from clause

```

delete from (select * from employees2 where department_id = 90);

```

### 1.9 merge 语句

Syntax:

```

merge <hint>
into <table_name>
using <table_view_or_query>
on (<condition>)
when matched then <update_clause>
delete <where_clause>
when not matched then <insert_clause>
[log errors <log_errors_clause> <reject limit <integer | unlimited>];

```