创建表：

create table stu(

id number(8),

name varchar2(200),

age number(5)

);

表stu2 表stu3 和 表stu4 数据结构和stu 都一样。

1. 传统插入方式，逐条在表中插入数据，浪费时间。 values 必须有

insert into stu(id, name, age) values(1, 'test', 20);

1. 利用select方式，切记不可加values 关键字,select语句括号可有可无，为了清晰，建议

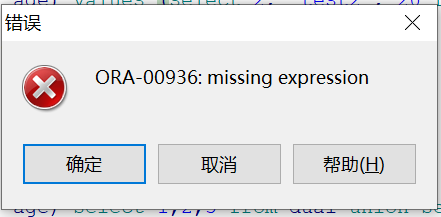
加上括号

正确写法：

insert into stu(id, name, age) (select 2, 'test2', 20 from dual);

错误写法：

insert into stu(id, name, age) values(select 2, 'test2', 20 from dual);



1. 以并集方式批量插入数据

insert into stu(id, name, age) (select 1, 'test1', 10 from dual union all select 2, 'test2', 20 from dual);

由上面第2中方式插入表中数据，我们可以推出下面第3中情况插入数据

1. 将其他的数据表插入到同一张表中

4.1 一般将一张表数据字段处理复制到另一张表中

insert into stu(id, name, age) (select t.field1, t.field2, t.field3|| t.field4 from 表B t);

4.2 将多张表数据复制到另一张表中

insert into stu(id, name, age) (select b.field1 id, b.field2 name, b.field3 age as id from 表B b union all select c.field1 id, c.field2 name, c.field3 age from 表C c union all select 3 id, ‘test3’ name, 30 age from dual);

这里要注意：1、数据表并集数据类型必须一致。表B的字段field1 field2 field3 和 表C的字段field1 field2 field3 以及 3, ‘test3’, 30 的数据类型必须一致，目前数据类型依次为整型，字符串，整型，否则并集失败。2、字段处理。通常需要将select表的字段需要简单处理进行插入。3、 字段顺序。select 查询语句的字段顺序 和 insert 字段顺序一致。 无论select的语句是否有字段别名于insert字段对应，都是以顺序为主，与名称无关。

1. 给多张表插入数据

insert all

into stu2(id, name, age) values(1, ‘test1’, 11)

into stu3(id, name, age) values(2, ‘test2’, 22)

select 1 from dual;

注意：这里select 1 from dual 语句不能缺少。

特殊场景1：当stu2 和 stu3 为同一张表是，也就是对一张表批量插入

insert all

into stu(id, name, age) values(2, 'test2', 22)

into stu(id, name, age) values(3, 'test3', 33)

select 1 from dual;

特殊场景2：将stu表中数据复制到stu2 和 stu3 stu4 中：

insert all

into stu2(id, name, age)

into stu3(id, name, age)

into stu4(id1, name1, age1)

select t.id, t.name, t.age from stu t;

实际应用： 注意sf\_ins1.i\_open\_provision$seq.nextval不能放在最里层的select语句中

insert into sf\_ins1.i\_open\_provision\_852

(ps\_id,

busi\_code,

done\_code,

ps\_type,

prio\_level,

dead\_line,

sort\_id,

ps\_service\_type,

bill\_id,

sub\_bill\_id,

plan\_id,

sub\_valid\_date,

upp\_create\_date,

create\_date,

start\_date,

end\_date,

ret\_date,

status\_upd\_date,

mon\_flag,

action\_id,

old\_ps\_param,

ps\_param,

target\_param,

ps\_status,

fail\_num,

fail\_reason,

fail\_code,

hand\_id,

hand\_op\_id,

hand\_notes,

ret\_op\_id,

ret\_notes,

op\_id,

sp\_id,

region\_code,

org\_id,

stop\_type,

old\_ps\_id,

rollback\_flag,

async\_flag,

act\_flag,

ps\_net\_code,

ps\_device\_code,

acc\_id,

sub\_id,

sub\_passwd,

hand\_date,

service\_id,

sub\_plan\_no,

notes,

retry\_times,

fail\_log,

org\_ps\_id,

order\_date,

suspend\_ps\_id,

ps\_service\_code,

region\_id)

(select

sf\_ins1.i\_open\_provision$seq.nextval ps\_id,

'500161121057',

'000',

'1',

'1',

'',

'0',

'HLR',

t.access\_num,

t.imsi,

'' plan\_id,

sysdate,

sysdate,

sysdate,

sysdate,

sysdate,

sysdate,

sysdate,

'0',

'6',

'PASSWORD='||t.oldpassword||';',

'PASSWORD='||t.password||';',

'',

'0',

'',

'' fail\_reason,

'' fail\_code,

'' hand\_id,

'' hand\_op\_id,

'' hand\_notes,

'' ret\_op\_id,

'000',

'91110001' op\_id,

'',

'852',

'21000100' org\_id,

'' stop\_type,

'' old\_ps\_id,

'' rollback\_flag,

'' async\_flag,

'' act\_flag,

'' ps\_net\_code,

'' ps\_device\_code,

'' acc\_id,

'0' sub\_id,

'' sub\_passwd,

sysdate,

'0' service\_id,

'0' sub\_plan\_no,

''notes,

'0' retry\_times,

'' fail\_log,

'' org\_ps\_id,

'' order\_date,

'' suspend\_ps\_id,

'' ps\_service\_code,

'852'

from (select b.\*, rownum as num from CVT.LK\_CHANGE\_PASSW\_TMP\_20191015@DBLK\_TO\_YY\_BTS1 b) t where t.num>=100 and t.num<2000 );