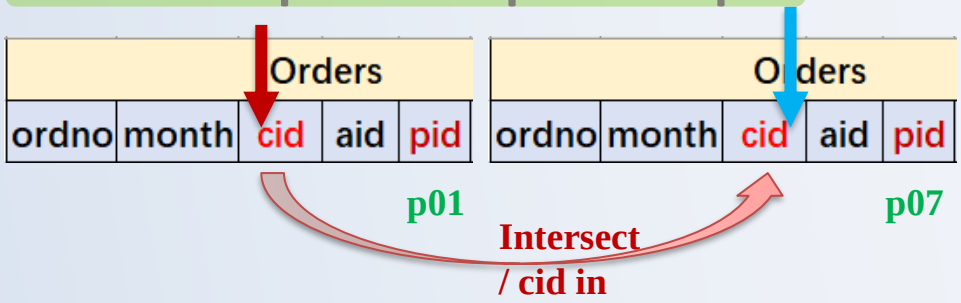
3.6 Some advanced SQL syntax

- **Another SQL Query?**

1. UNION, INTERSECT and EXCEPT Operators

[1] form subquery {UNION [ALL] | INTERSECT [ALL]
| EXCEPT [ALL] subquery}

[2] Example 3.6.1 Find cids who order both products p01 and p07.

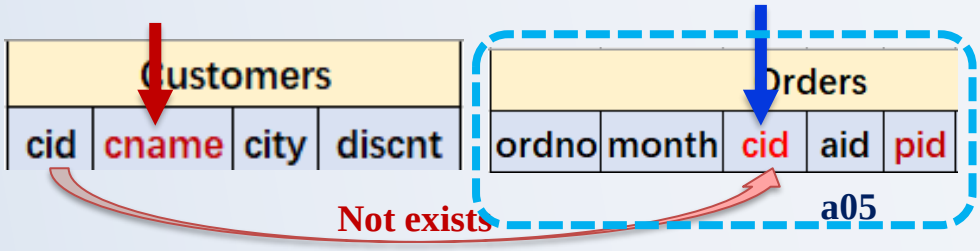


New) (select distinct cid from orders where pid = 'p01')
intersect (select cid from orders where pid = 'p07');

Old) select distinct cid from orders where pid = 'p01' and
cid in (select cid from orders where pid = 'p07');

3.6 Some advanced SQL syntax

[3] Example 3.6.2 (p116) Retrieve all cnames where the customer don't place any order through agent a05.



Old) select c.cname from customers c where
not exists (select * from orders x where x.aid='a05' and x.cid=c.cid);

New) select c.cname from customers c
except (select c.cname from customers c, orders x
where x.aid = 'a05' and x.cid = c.cid);

[4] Operators in DBMS product

- ORACLE 8 : {UNION, UNION ALL, INTERSECT, EXCEPT}
- DB2 Version 5 : {UNION, INTERSECT, EXCEPT} [ALL]
- INFORMIX : UNION [ALL]

3.6 Some advanced SQL syntax

2. Join in advanced SQL

a) Join Predicate

Tableref := tablename [[AS] corr_name[(colname {, colname ...})]]

| (subquery) [AS] corr_name [(colname {, colname ...})]

| tableref1 [INNER | { LEFT | RIGHT | FULL}

[OUTER] JOIN tableref2 ON search_condition |

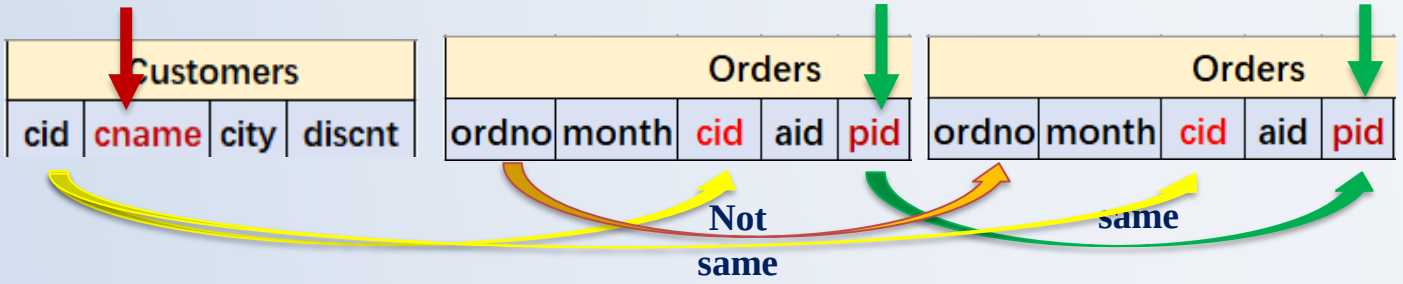
USING (colname {, colname . . .})}

b) FROM 子句中含Subquery

clause := FROM tableref {, tableref . . .}

3.6 Some advanced SQL syntax

[2]Example 3.6.3(p118) Retrieve all cnames where the customer places at least two orders for the same product.



S1) select cname from (select o.cid as spcid from orders o, orders x

where o.cid=x.cid and o.pid=x.pid and o.ordno<>x.ordno) as y,

customers c where y.spcid = c.cid;

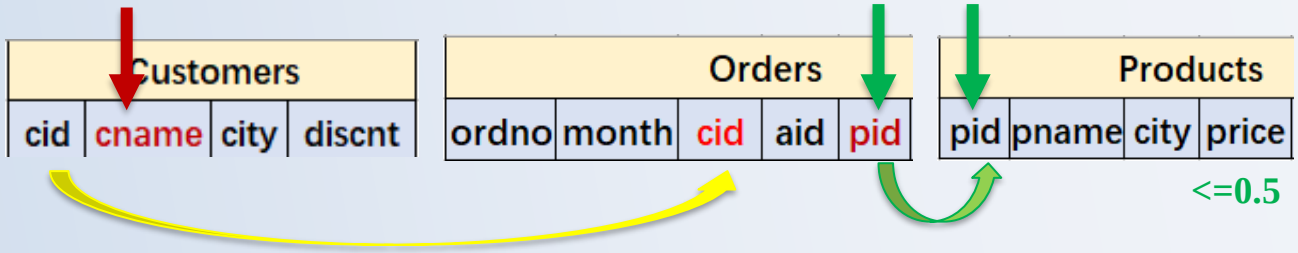
（from子句中出现Subquery：先求View y，再作其它处理）

S2) select cname from customers c, orders o, orders x

where c.cid=o.cid and o.pid=x.pid and o.ordno<>x.ordno;

3.6 Some advanced SQL syntax

[3] Example 3.6.4 (p119) **Retrieve all customers who has buy at least one product costing less than $0.50.**



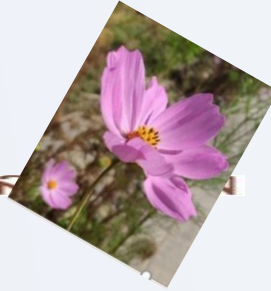
a)Alg. (((P where price <0.50)[pid]) ∞ O ∞ C) [cname]

b)New1
```
select distinct cname from (O join P on O.pid = P.pid )
                join C on O.cid =  C.cid where P.price <0.50;
```

New2
```
select distinct cname from  (O join  P using(pid) )
                join C using(cid)  where price<0.50;
```

Old3
```
select cname from Customers C, Orders O, Products P
       where C.cid=O.cid and O.pid=P.pid and P.price<0.50;
```

3.6 Some advanced SQL syntax



3. Outer Join

[1] Form: [{LEFT | RIGHT | FULL} [OUTER]] JOIN

[2] Example: (p120) We have two tables S and T

(a) select * from S full outer join T using(A) ;

(b) select * from S left outer join T using(A)

UNION select * from S right outer join T using(A);

(a)

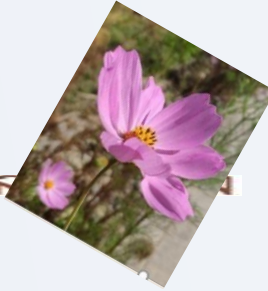
S.C	S.A	T.A	T.B
c1	a1	a1	b1
c3	a3	a3	b3
c4	a4	a4	b4
		a2	b2

?

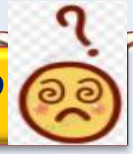
(b)

S.C	S.A	T.B
c1	a1	b1
c3	a3	b3
c4	a4	b4
null	a2	b2

3.7 Set function in SQL



- *Is there some Functions in the SQL?*



1.Set functions in SQL

[1] Set Functions

- **Five types : count, max, min, sum, avg**
- **Operate on sets of values, return a single value**

[2] Different terms in Products

- **X/OPEN and INGRES refer to** — ***set function***
- **ANSI says** — ***aggregate functions***
- **ORACLE has** — ***group functions***
- **DB2 says** — ***column functions***

3.7 Set function in SQL

[3] Set function syntax of SQL

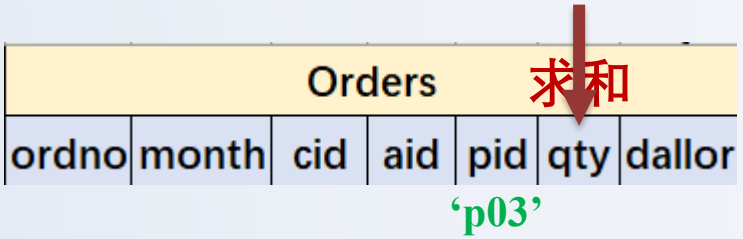
Select SET_functionName ([ALL | DISTINCT] colname) from…..

Select COUNT(*) from……

Name	Argument type	Result type	Description
count	any (can be *)	numeric	count of occurrences
sum	numeric	numeric	sum of arguments
max	char or numeric	same as arg.	maximum value
min	char or numeric	same as arg.	minimum value
avg	numeric	numeric	average of arguments

3.7 Set function in SQL

2. Example 3.7.2 Please to determine the total quantity of product p03 that has been ordered.




```
select sum(qty) as TOTAL from orders where pid = 'p03';
```

3. Example 3.7.4 Get number of cities where customers are based.

```
select  count (distinct city)  from customers;
```

Comparing:

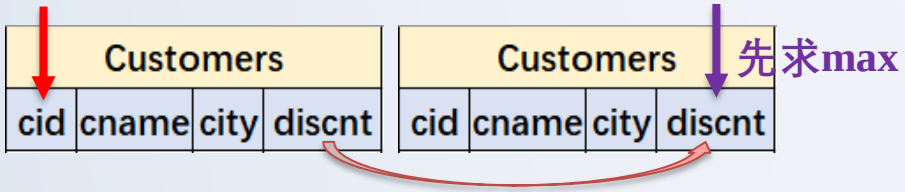
```
select count(*)     from customers;     /* all the numbers of rows */
select count(cid)   from customers;     /* null values not counted, cid is PK of customers*/
select count(city)  from customers;     /* only if no null city values*/
```

3.7 Set function in SQL

4. Restriction !!! set functions are not allowed to appear in comparison of a WHERE clause

5. Example 3.7.5 List cids who have a discount less than maximum discount.

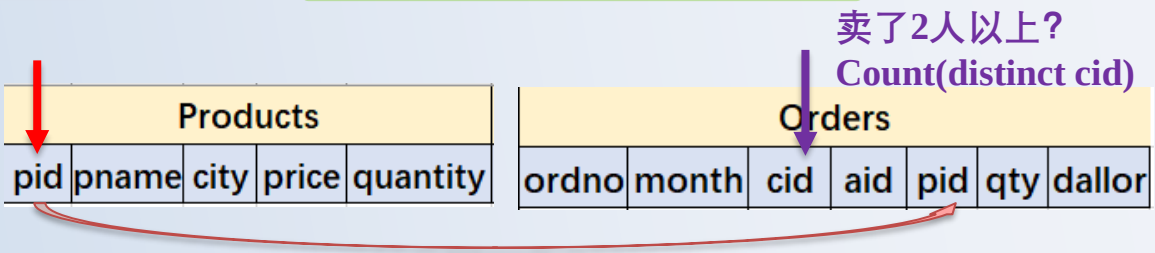
a)Invalid: select cid from customers where discnt < max(discnt);



b)Effective: select cid from customers

where discnt < (select max(discnt) from customers);

6. Example 3.7.6 Find products ordered by at least two customers.



select pid from products

where 2 <= (select count(distinct cid) from order o

where o.pid=p.pid)

3.7 Set function in SQL

7.Handing NULL values

- *null* is a special constant(常数)
- meaningful a value that is *UNKNOWN*.

8.Example 3.7.8 After adding a row (c007, Windix, Dallas, null) to the customers table, Pose the query to retrieve this row.

a) SQL select * from customers where discnt is null;

b) wrong select * from customers where discnt<=10 and discnt>10;

3.7 Set function in SQL

9.Example 3.7.9 After adding the row (c007,Windix,Dallas,null) to customers table. We wish to find the average discount of all customers.

a) the null is discarded（抛弃，排除）

```
select avg(discnt) from customers;
```

b) A set function acting on a empty set

- **count()** — returns zero for a empty set.
- **sum(), avg(), max() and min()** — return the null value.

3.8 Group of rows in SQL

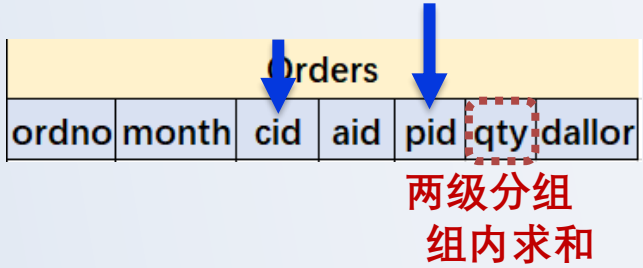
- How to classify the Data in different groups?

1. GROUP and HAVING clause

SELECT [ALL|DISTINCT] {*|expr [[AS] c_alias]{,expr [[AS] c_alias]...}}
FROM tableref {, tabletrf...}
[WHERE search_condition]
[GROUP BY column {, column...}] [HAVING search_condition]
| subquery UNION [ALL] | INTERSECT [ALL] | EXCEPT [ALL]
[CORRESPONDING [BY] (colname {, colname . . .})] subquery

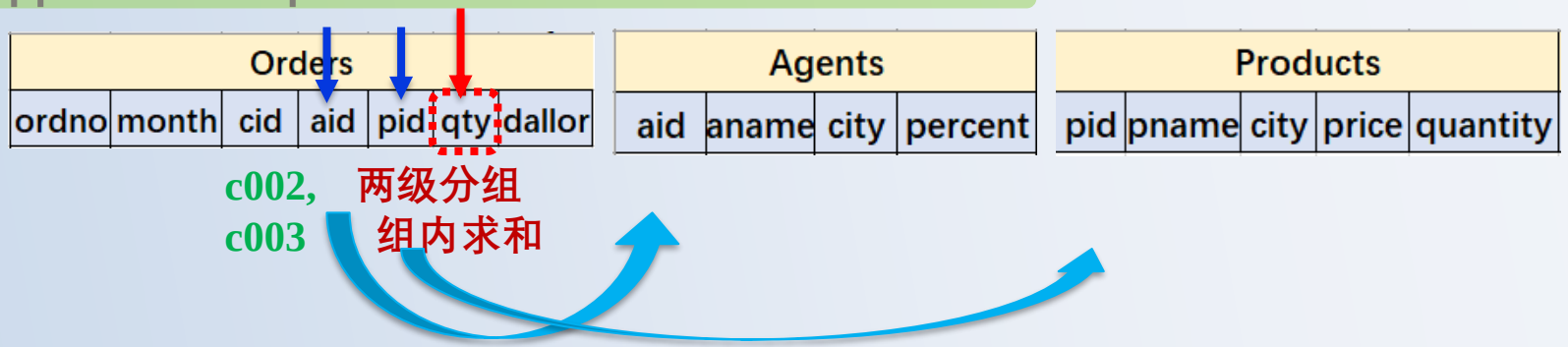
2. Example 3.8.1 Let's create a query to calculate the total product quantity ordered of each individual product by each individual customer. (查每个(pid, cid)对之间的销货数).

select pid, cid, sum(qty) as TOTAL from orders group by pid, cid;



3.8 Group of rows in SQL

3.Example 3.8.2 Print out the aname, aid, pname, pid together with the total quantity each agent supplies of that product to customers c002 and c003.

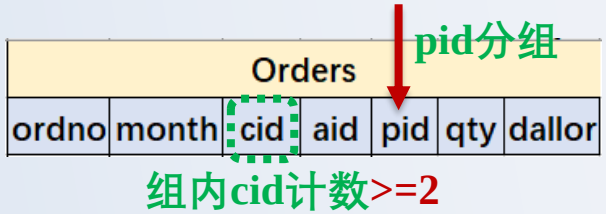



```
select  x.aid, x.pid, sum(qty)  from orders x
        where x.cid in ('c002','c003')   group by aid, pid;

select aname, x.aid, pname, x.pid, sum(qty) as total_qty
        from agent a, products p, orders x
        where x.cid in ('c002','c003') and x.aid=a.aid and x.pid=p.pid
        group by aid, pid;
```

3.8 Group of rows in SQL

4. Example 3.8.4 Provided pids purchased by (sell to) at least two customers.




```
select pid from orders
        group by pid having count(distinct cid) >= 2;
```

5. Example 3.8.5 Pose a query to find the average, over all agents, of the maximum dollar sales made by each agent.


```
select aid, max(dollars) as x
        from orders  group by aid;

select avg(t.x) from  (select aid, max(dollars) as x
        from orders  group by aid) t;
```

3.9 A complete description of Select

1. Experiments of making a Select statement

a) The **Cartesian Product** of all tables in ***FROM*** clause is formed.

b) **Rows not satisfying** in ***WHERE*** condition are eliminated (消去/限制).

c) **Rearranging rows** are grouped in accordance with the ***GROUP BY***.

d) **Groups not satisfying** the ***HAVING*** clause are then eliminated.

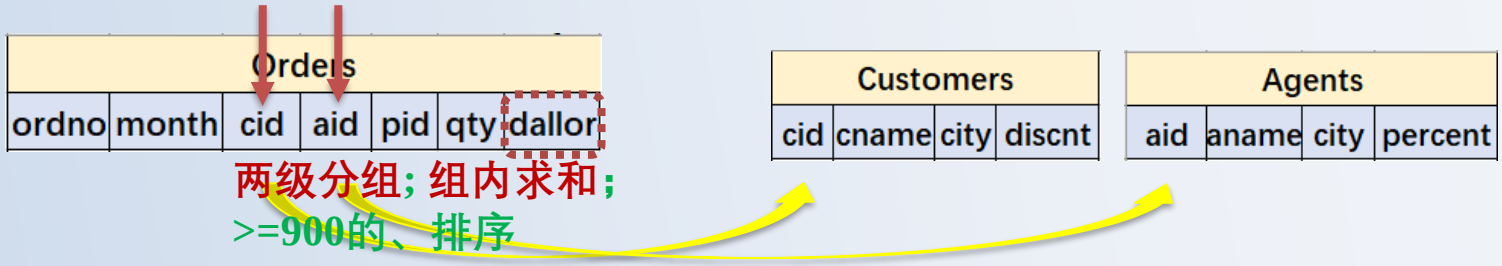
e) **The expressions of the Select Clause** ***target list*** are calculated.

f) If the keyword **DISTINCT** is present, ***duplicate rows are eliminated***. The **UNION**, **INTERSECT**, **EXCEPT** is taken after each subquery.

g) If an **ORDER BY** is present that set of ***all selected rows is sorted***.

3.9 A complete description of Select

2.Example 3.9.1 List all customers, agents and the dollar sales for pairs of customers and agents, and order the result from largest to smallest sales totals. Retain only those pairs for which the dollar amount is at least 900.




```
select c.cname, o.cid, a.aname, o.aid, sum(o.dollars) as casales
        from customers c, orders o, agents a
        where c.cid = o.cid and o.aid = a.aid
        group by  (c.cname,) o.cid, (a.aname,) o.aid
                                having sum(o.dollars) >= 900.00
        order by  5  desc;
```

Note: 5 is means to the fifth column "casales" is on descending.

3.9 A complete description of Select

3. Expressions, Predicates, Search_condition

[1] Expressions Form

expr = num-expr | strv-expr | date-xpr | intv-expr | con-dexpr

[2] Data Types

- ***Numeric*** **value expressions**
- ***String*** **value expression**
- ***Datetime*** **expression**
- ***Interval value*** **expression**
- ***Conditional*** **expression**

Note:

- **Difference *DBMS product* provides difference function;**
- **Data types for columns defined in Appendix A.3**

3.9 A complete description of Select

[3] Mathematical functions

abs(n), mod(n,b), sqrt(n). NOT standardized in SQL-99.

[4] Char & String functions

CHAR_LENGTH (str)

SUBSTRING (str FROM start [FOR length])

TRIM ([[LEADING|TRAILING|BOTH][set] FROM] str)

POSITION (str1 IN str2)

UPPER(strval), LOWER(strval)

[5] Logical Values

- TRUE/ FALSE/ UNKNOW ('U' is not equivalent to 'F')

AND	T	F	U	OR	T	F	U		NOT
T	T	F	U	T	T	T	T	T	F
F	F	F	F	F	T	F	U	F	T
U	U	F	U	U	T	U	U	U	U

3.9 A complete description of Select

[6] Predicates

a) Comparison predicate: expr1 { expr2 | expr1 (Subquery)}

b) Between predicate: c.discnt between 10 and 12

c) In predicate: expr [not] in (subquery)

d) Quantified predicate: expr [all |any |some] (subquery)

e) Exists predicate: [not] exists (subquery)

f) Is null predicate: columnname is [not] null

g) Like predicate: columnname [not] like 'pattern'

3.9 A complete description of Select

4. Summary of Predicates

[1]Comparison predicate expr1 | expr1 (Subquery){=, <>, >, <,>=, <=}

➢Example 3.9.2 Listed cids with a discount less than the maximum discount.

```
select max(discnt) from customers
```

S1
```
select cid from customers
        where discnt < (select max(discnt) from customers);
```

S2
```
select cid from customers
        where discnt < any(select discnt from customers);
```

➢Example: Find orders made by customers whose city come after 'M' in the alphabetic. （选 'M' 以后字母开头的city）

```
select * from orders o where
        'M'<(select city from customers c where c.cid=o.cid);
```

3.9 A complete description of Select

[2] Between predicate: expr [NOT] BETWEEN expr2 and expr3

select * from products where price between 1 and 10

[3] Quantified predicate: expr [ALL | ANY | SOME] (subquery)

- Example 3.9.3 Get maximum discount of all customer.

S1

```
select max(discnt) as maxdis from customers;
```

S2

```
select distinct discnt from customers c where discnt
    >=all (select discnt from customers d where d.cid<>c.cid);
```

[4] Exists predicate [NOT] EXIST (subquery)

[5] In predicate expr [NOT] IN (subquery)

[6] Is null is [NOT] NULL

3.9 A complete description of Select

[7] Like predicate: columnname [NOT] LIKE 'pattern'

- Wildcards (通配符. Like windows file wildcards)

Underscore (_)	Any single character
Percent (%)	Zero or more characters of any form
Escape character (\)	Precedes quoted literal character
All other characters	Represent themselves

➢ Example 3.9.4 Retrieve all data of customers whose name begins with the letter 'A'.

```
select * from customers where cname like 'A%' ;
```

3.9 A complete description of Select

- **Example 3.9.6** Retrieve cids of customers whose cname begins 'Tip_' and arbitary(any) number of character .

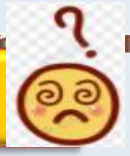
```
select cid from customers
        where cname like 'Tip\_%' escape '\';
```

- **Example 3.9.7** Retrieve cids of customers whose cname starts with the sequence 'ab\' .

```
select cid from customers
        where cname like 'ab\\%' escape '\';
```

3.10 Insert, Update and Delete statements

How to Updates the Data in DB?



- Update statements : Insert , Update , Delete
- Need update *privilege* on a table

1. Insert Statement

[1] Form

➢ 标准式：**INSERT INTO** tablename [(column {, column…})]

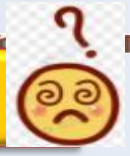
{**VALUES** (expr | null { , expr | null…}) | **subquery**}

[2] Example 3.10.1 (p148) Add a row specified values to the orders table. (no qty or dollars, so on this new row they are null).

```
insert into orders (ordno, month, cid, aid, pid)
        values('1107', 'aug', 'c006', 'a04', 'p01');
select * from order where dollars is null;
```

3.10 Insert, Update and Delete statements

How to Updates the Data in DB?



- Update statements : Insert , Update , Delete
- Need update *privilege* on a table

1. Insert Statement

[1] Form

- 标准式：**INSERT INTO** tablename [(column {, column…})]
  {VALUES (expr | null { , expr | null…}) | **subquery}**
- 插入子查询结果集

  INSERT INTO tablename [colname[, colname…]）
  Subquery;

[2] Example 3.10.2 Create a new table Swcusts of customers, and insert into all customers from Dallas and Austin.

```
select * from customers where city in ('Dallas', 'Austin')
insert into swcusts (select * from customers where city in ('Dallas', 'Austin') );
```

3.10 Insert, Update and Delete statements

2. Update Statement

[1] Form: **UPDATE** tablename **SET** colname = {expr | null | (subquery)}
{, {column = expr |null | (subquery)…}}
[**WHERE** search_condition];

[2] Example 3.10.3 (p150) Give All agents in New York a 10% raise in the percent commission they earn on an order.

```
UPDATE  agents  SET percent = 1.1*percent
WHERE city = 'New York';
```

3.10 Insert, Update and Delete statements

[3] Example 3.10.5 Update the *discnt* values in rows of the *Swcusts* table created in Example 3.10.2 with more up-to-date *discnt* values from the customers table.

[2] Example 3.10.2 Create a new table Swcusts of customers, and insert into all customers from Dallas and Austin.

```
insert into swcusts (select * from customers
                     where city in ('Dallas', 'Austin') );
```

```
update swcusts s set discnt=(select discnt from customers c
                             where c.cid = s.cid );
```

3.10 Insert, Update and Delete statements

3. Delete Statement

[1] Form: DELETE FROM tablename [WHERE search_condition];

[2] Example 3.10.6 (p151) Delete all agents in New York.

```
delete from agents where city = 'New York';
```

[3] Example 3.10.7 Delete agents who have total orders of less than $600.


```
select aid from orders
        group by aid having sum(dollars) < 600
```

```
delete from agents where aid in
        ( select aid from orders
                group by aid having sum(dollars) < 600 );
```

3.10 Insert, Update and Delete statements

[4] Notice: Table Drop

drop table agents; ~~(表结构+数据)~~

Suggestions for doing exercises:

(1) reading through Appendix B, and Appendix C

C.2, C.3, C.4, C.14, C.17, C.25, C.28, and C.31

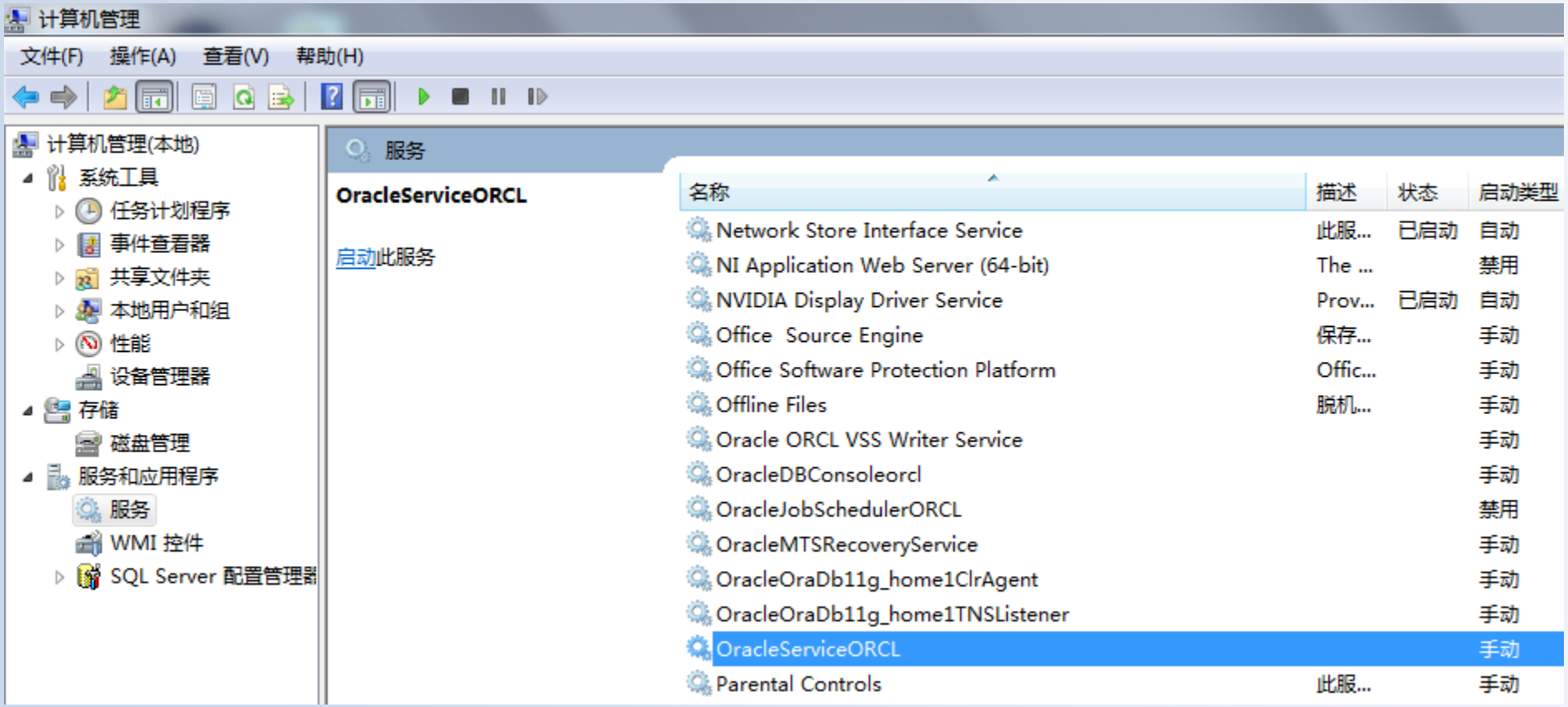
(2) Exercises: all exercises of charter 3 .

specialized for [3.1], [3.2], [3.5], [3.7], [3.8], [3.10], [3.11], [3.12], [3.14], [3.15], [3.16], [3.18], [3.19], [3.20]

Additional： DBMS Example

1.DBMS Example ORACLE

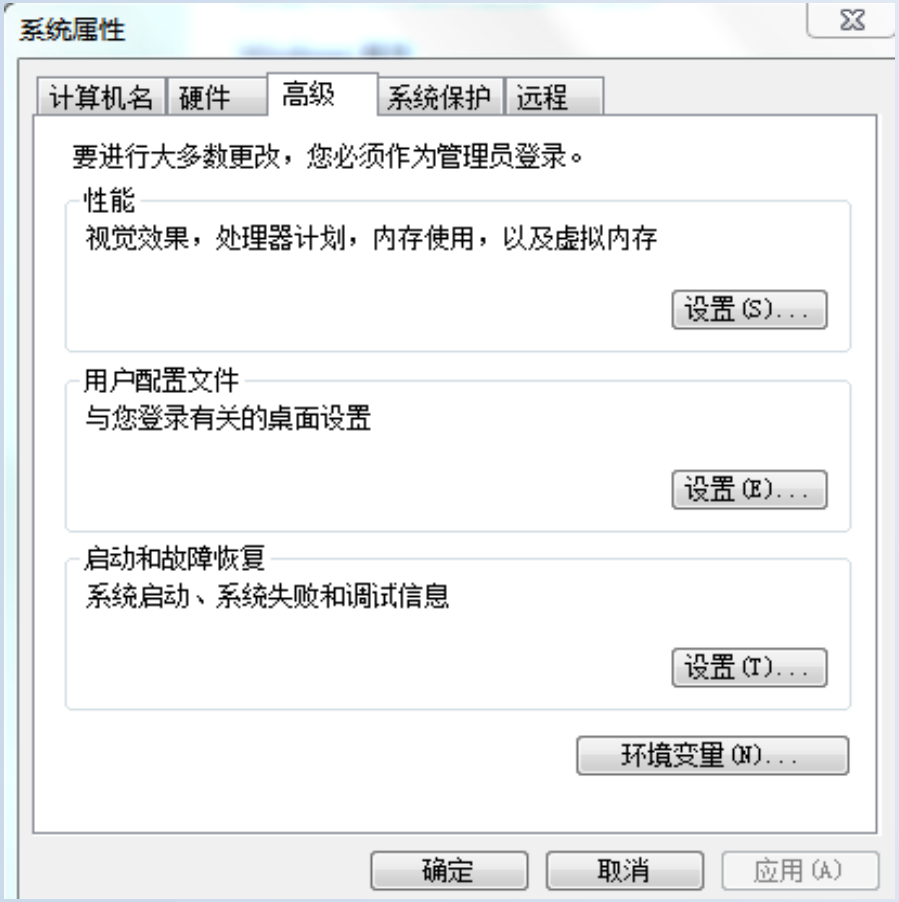
- ✓ Setup ORACLE 11g (for Windows 7 64bit)
- ✓ There are 7 Serves in Windows:

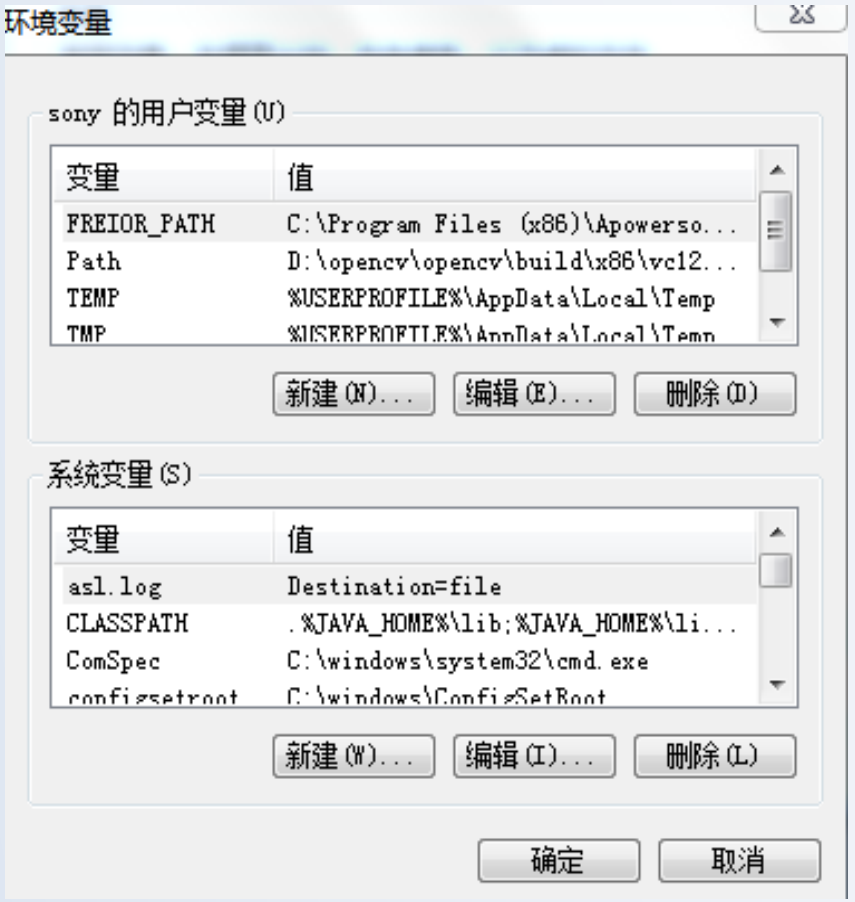


Additional： DBMS Example

1.DBMS Example ORACLE

- **Setup the Environment variables**

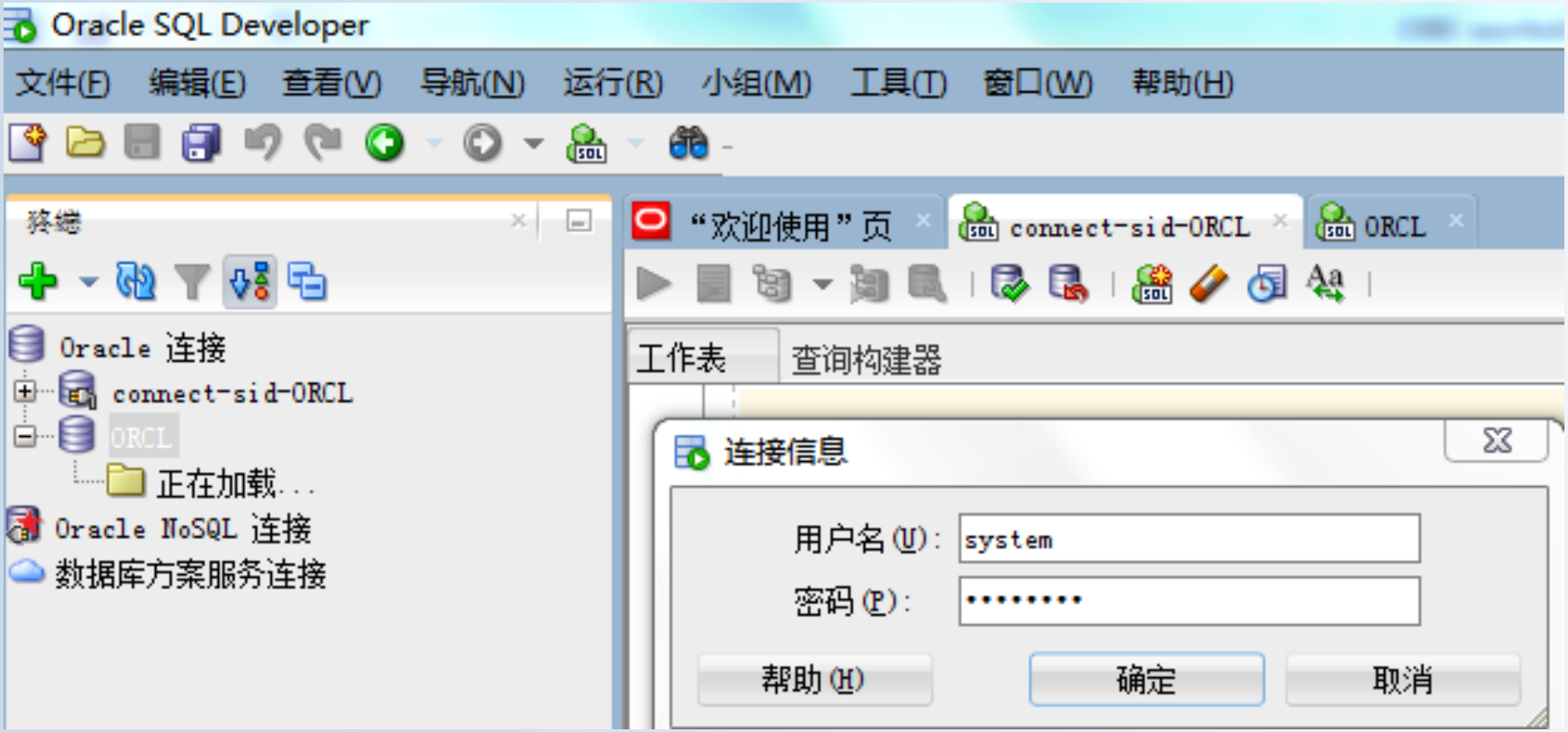




Additional： DBMS Example

1.DBMS Example ORACLE

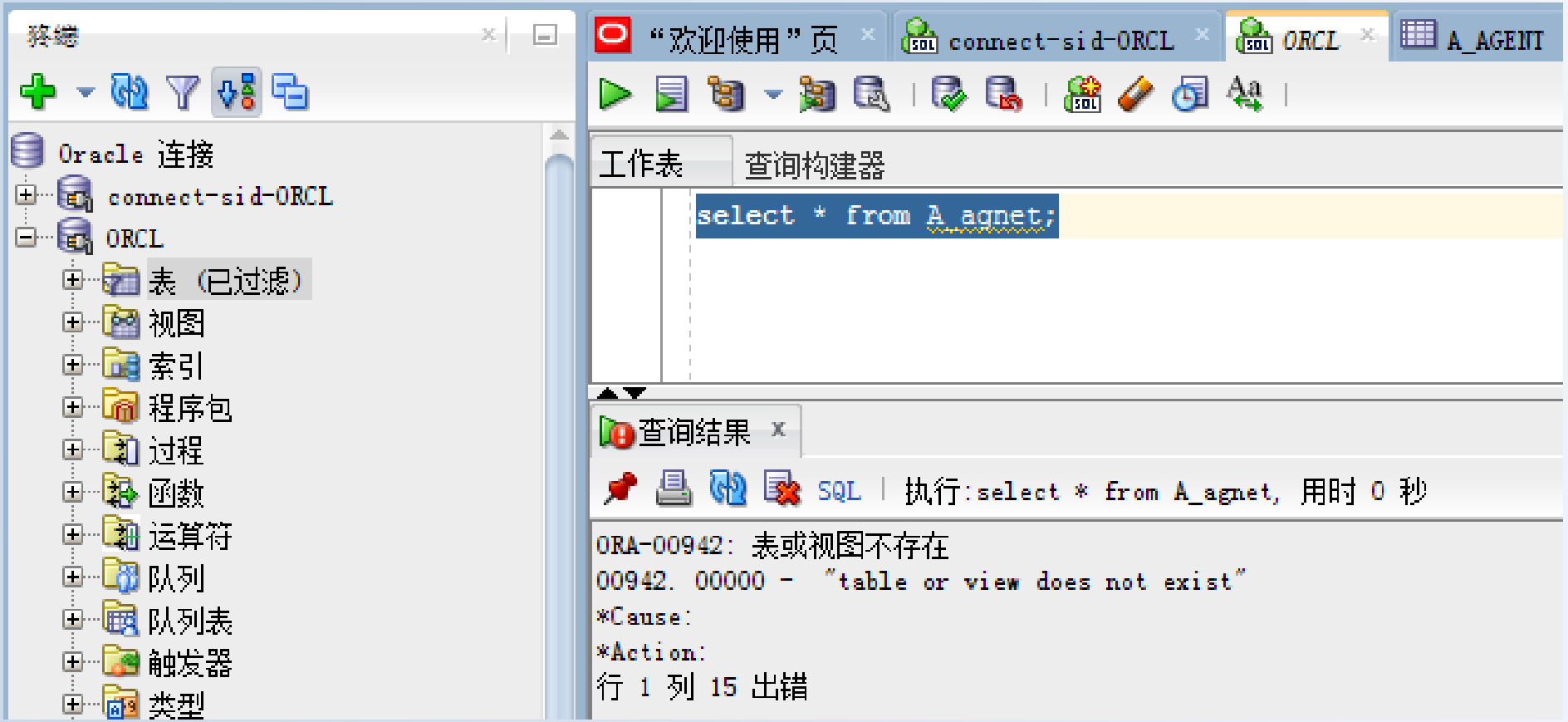
- Start up ORACLE servies
- Start up SQL Plus or SQLdeveloper
- Login/Connect the ORACLE with Account & Password



Additional： DBMS Example

1.DBMS Example ORACLE

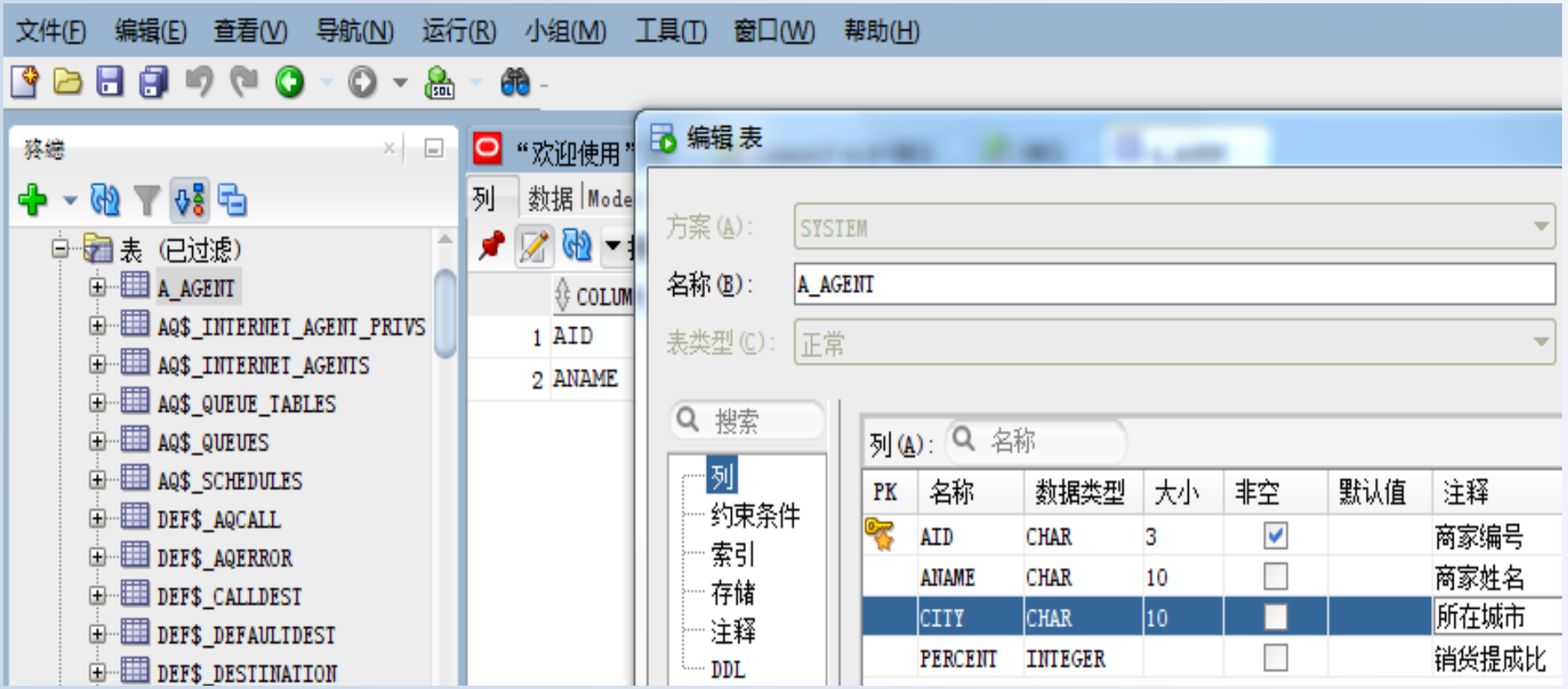
- Input SQL statement



Additional： DBMS Example

1.DBMS Example： ORACLE

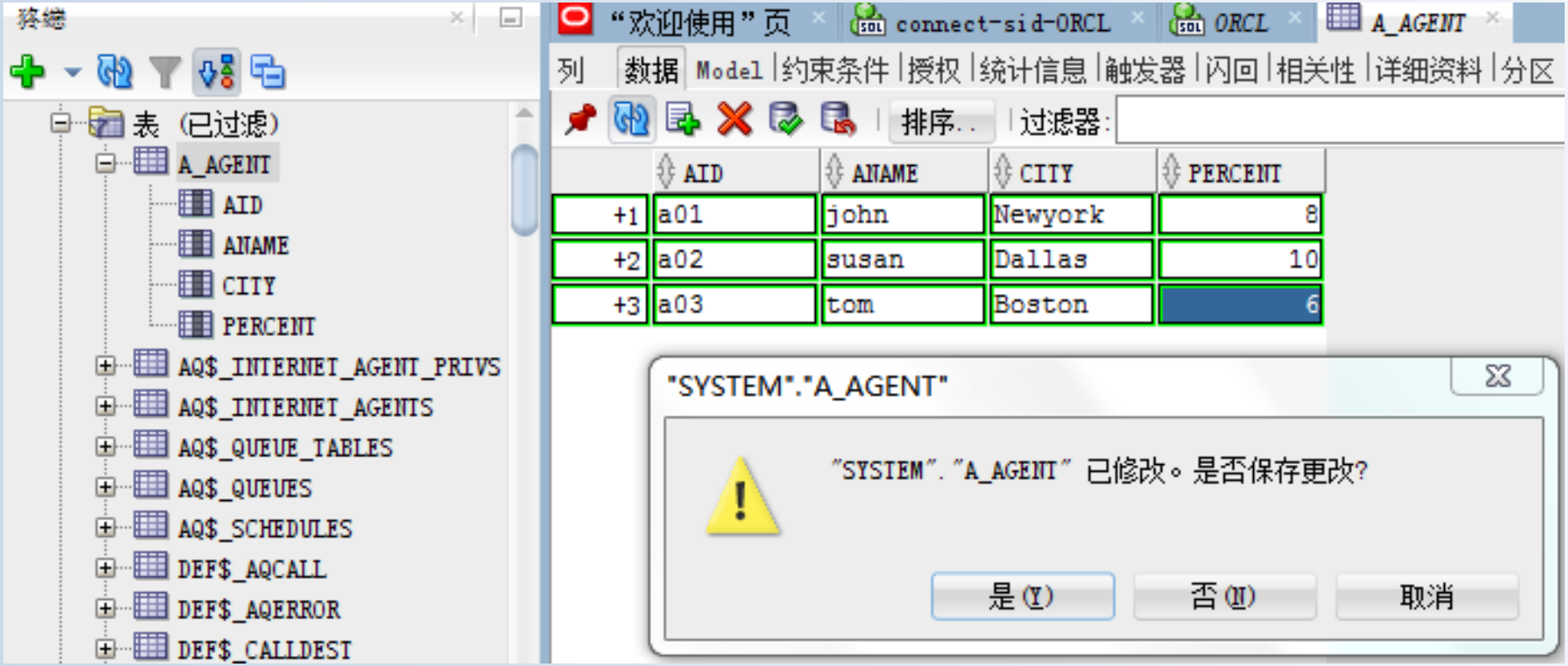
- Create table A_agent



Additional： DBMS Example

1.DBMS Example ORACLE

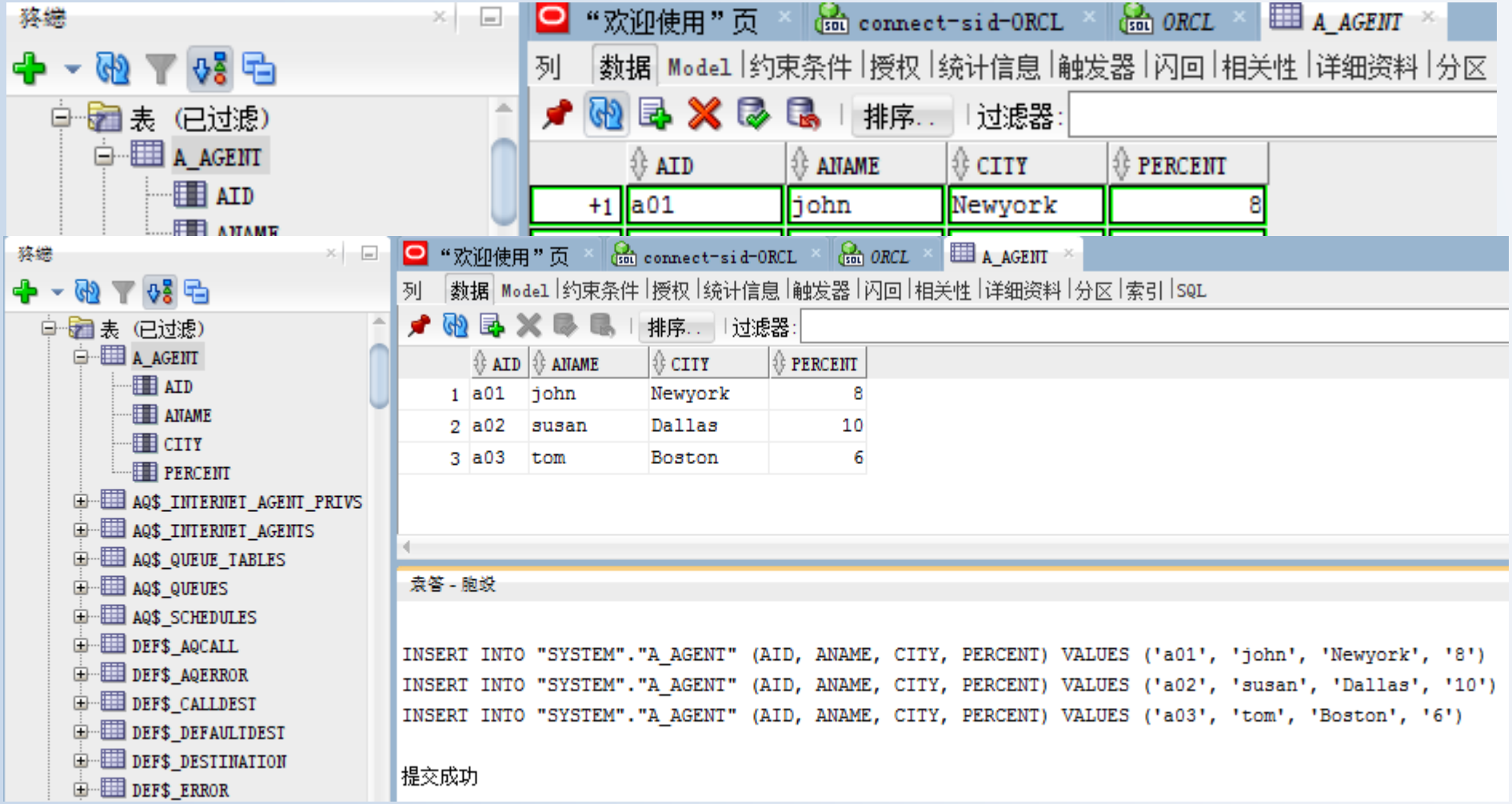
- Insert the Data to the table by Visual box A_agent



Additional： DBMS Example

1.DBMS Example ORACLE

- Insert the Data to the table using SQL A_agent



Additional： DBMS Example

1.DBMS Example ORACLE