SQL | USING Clause

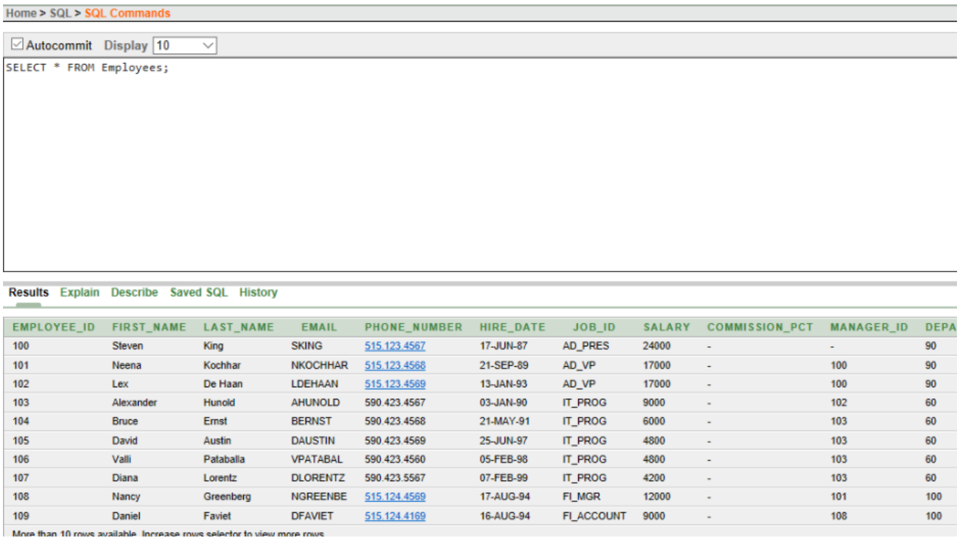
akanshgupta

If several columns have the same names but the datatypes do not match, the NATURAL JOIN clause can be modified with the **USING** clause to specify the columns that should be used for an EQUIJOIN.

- USING Clause is used to match only one column when more than one column matches.
- NATURAL JOIN and USING Clause are mutually exclusive.
- It should not have a qualifier(table name or Alias) in the referenced columns.
- NATURAL JOIN uses all the columns with matching names and datatypes to join the tables. The USING Clause can be used to specify only those columns that should be used for an EQUIJOIN.

EXAMPLES:

We will apply the below mentioned commands on the following base tables:

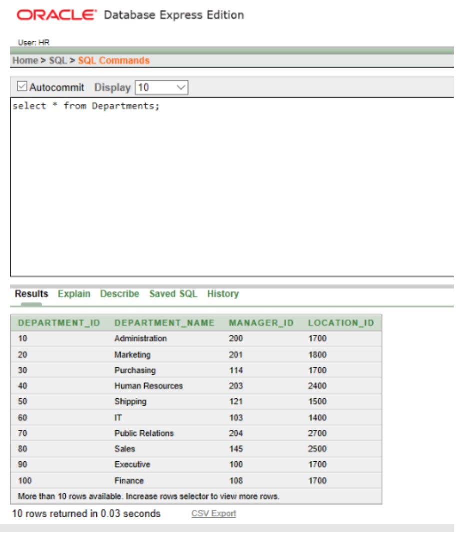


```
SELECT * FROM Employees;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPA
100	Steven	King	SKING	515.123.4567	17-JUN-87	AD_PRES	24000	-	-	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	17000	-	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	17000	-	100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	9000	-	102	60
104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-91	IT_PROG	6000	-	103	60
105	David	Austin	DAUSTIN	590.423.4569	25-JUN-97	IT_PROG	4800	-	103	60
106	Valli	Pataballa	VPATABAL	590.423.4560	05-FEB-98	IT_PROG	4800	-	103	60
107	Diana	Lorentz	DLORENTZ	590.423.5567	07-FEB-99	IT_PROG	4200	-	103	60
108	Nancy	Greenberg	NGREENBE	515.124.4569	17-AUG-94	FI_MGR	12000	-	101	100
109	Daniel	Faviet	DFAVIET	515.124.4169	16-AUG-94	FI_ACCOUNT	9000	-	108	100

More than 10 rows available. Increase rows selector to view more rows.

Employee Table


ORACLE Database Express Edition
User: HR
Home > SQL > SQL Commands
Results Explain Describe Saved SQL History
CSV Export

☑ Autocommit Display 10

```
select * from Departments;
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700
40	Human Resources	203	2400
50	Shipping	121	1500
60	IT	103	1400
70	Public Relations	204	2700
80	Sales	145	2500
90	Executive	100	1700
100	Finance	108	1700

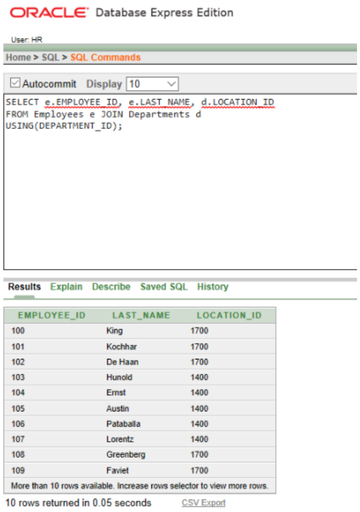
More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.03 seconds

Department Table

QUERY 1: Write SQL query to find the working location of the employees. Also give their respective employee_id and last_name?

```
Input : SELECT e.EMPLOYEE_ID, e.LAST_NAME, d.LOCATION_ID
FROM Employees e JOIN Departments d
USING(DEPARTMENT_ID);
Output :
```


ORACLE Database Express Edition
User: HR
Home > SQL > SQL Commands
Results Explain Describe Saved SQL History
CSV Export

☑ Autocommit Display 10

```
SELECT e.EMPLOYEE_ID, e.LAST_NAME, d.LOCATION_ID
FROM Employees e JOIN Departments d
USING(DEPARTMENT_ID);
```

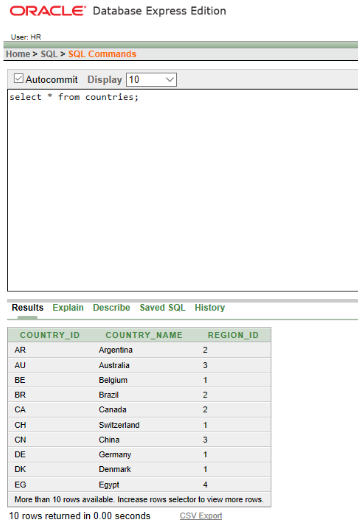
EMPLOYEE_ID	LAST_NAME	LOCATION_ID
100	King	1700
101	Kochhar	1700
102	De Haan	1700
103	Hunold	1400
104	Ernst	1400
105	Austin	1400
106	Pataballa	1400
107	Lorentz	1400
108	Greenberg	1700
109	Faviet	1700

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.05 seconds

Explanation: The example shown joins the DEPARTMENT_ID column in the EMPLOYEES and DEPARTMENTS
tables, and thus shows the location where an employee works.

We will apply the below mentioned commands on the following base tables:


ORACLE Database Express Edition
User: HR
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

☑ Autocommit Display 10

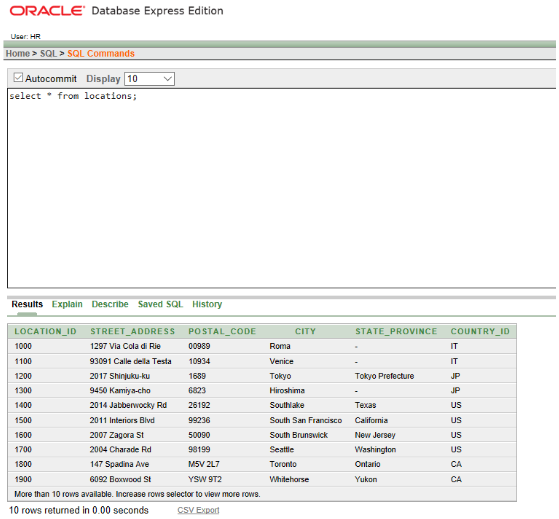
```
select * from countries;
```

COUNTRY_ID	COUNTRY_NAME	REGION_ID
AR	Argentina	2
AU	Australia	3
BE	Belgium	1
BR	Brazil	2
CA	Canada	2
CH	Switzerland	1
CN	China	3
DE	Germany	1
DK	Denmark	1
EG	Egypt	4

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.00 seconds

Country Table


ORACLE Database Express Edition
User: HR
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

☑ Autocommit Display 10

```
select * from locations;
```

LOCATION_ID	STREET_ADDRESS	POSTAL_CODE	CITY	STATE_PROVINCE	COUNTRY_ID
1000	1297 Via Cola di Rie	00989	Roma	-	IT
1100	93091 Calle della Testa	10934	Venice	-	IT
1200	2017 Shinjuku-ku	1689	Tokyo	Tokyo Prefecture	JP
1300	9450 Kamiya-cho	6823	Hiroshima	-	JP
1400	2014 Jabberwocky Rd	26192	Southlake	Texas	US
1500	2011 Interiors Blvd	99236	South San Francisco	California	US
1600	2007 Zagora St	50090	South Brunswick	New Jersey	US
1700	2004 Charade Rd	98199	Seattle	Washington	US
1800	147 Spadina Ave	M5V 2L7	Toronto	Ontario	CA
1900	6092 Boxwood St	YSW 9T2	Whitehorse	Yukon	CA

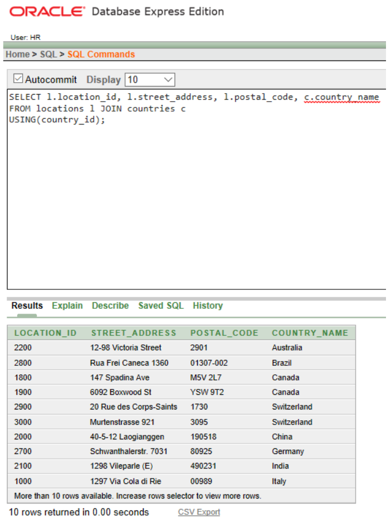
More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.00 seconds

Location Table

QUERY 2: Write SQL query to find the location_id, street_address, postal_code and their respective country name?

```
Input : SELECT l.location_id, l.street_address, l.postal_code, c.country_name
FROM locations l JOIN countries c
USING(country_id);
Output :
```


ORACLE Database Express Edition
User: HR
Home > SQL > SQL Commands
☑ Autocommit Display 10
Results Explain Describe Saved SQL History

```
SELECT l.location_id, l.street_address, l.postal_code, c.country_name
FROM locations l JOIN countries c
USING(country_id);
```

LOCATION_ID	STREET_ADDRESS	POSTAL_CODE	COUNTRY_NAME
2200	12-98 Victoria Street	2901	Australia
2800	Rua Frei Caneca 1360	01307-002	Brazil
1800	147 Spadina Ave	M5V 2L7	Canada
1900	6092 Boxwood St	YSW 9T2	Canada
2900	20 Rue des Corps-Saints	1730	Switzerland
3000	Murtenstrasse 921	3095	Switzerland
2000	40-5-12 Laogianggen	190518	China
2700	Schwanthalerstr. 7031	80925	Germany
2100	1298 Vileparle (E)	490231	India
1000	1297 Via Cola di Rie	00989	Italy

More than 10 rows available. Increase rows selector to view more rows.

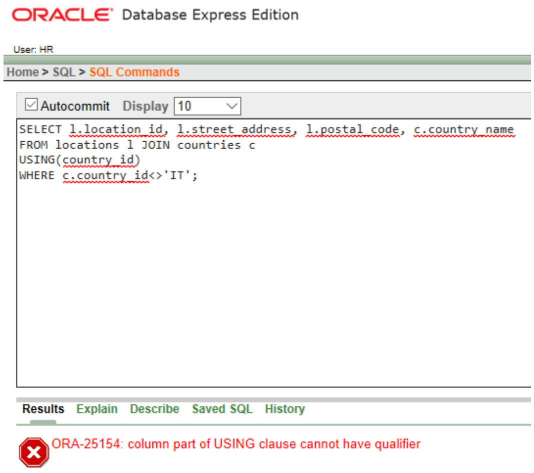
10 rows returned in 0.00 seconds CSV Export

Explanation: The example shown joins the COUNTRY_ID column in the LOCATIONS and COUNTRIES
tables, and thus shows the required details.

NOTE: When we use the USING clause in a join statement, the join column is not qualified with table Alias. Do not Alias it even if the same column is used elsewhere in the SQL statement:

Example:

```
Input: SELECT l.location_id, l.street_address, l.postal_code, c.country_name
FROM locations l JOIN countries c
USING(country_id)
WHERE c.country_id'IT';
Output:
```


ORACLE Database Express Edition
User: HR
Home > SQL > SQL Commands
☑ Autocommit Display 10
Results Explain Describe Saved SQL History

```
SELECT l.location_id, l.street_address, l.postal_code, c.country_name
FROM locations l JOIN countries c
USING(country_id)
WHERE c.country_id<>'IT';
```

ORA-25154: column part of USING clause cannot have qualifier

Explanation: Since the column in USING Clause is used again in WHERE Clause, thus it throws an error to the user.

Last Updated : 21 Mar, 2018

14

Similar Reads

1. Difference between Having clause and Group by clause
2. Combining aggregate and non-aggregate values in SQL using Joins and Over clause
3. SQL Full Outer Join Using Left and Right Outer Join and Union Clause
4. SQL query using COUNT and HAVING clause
5. SQL Full Outer Join Using Union Clause
6. SQL Full Outer Join Using Where Clause
7. Using CASE in ORDER BY Clause to Sort Records By Lowest Value of 2 Columns in SQL
8. Configure SQL Jobs in SQL Server using T-SQL
9. SQL | Distinct Clause
10. SQL | WHERE Clause

Previous Next

Article Contributed By :

akanshgupta
akanshgupta

Vote for difficulty

Current difficulty : Easy

Easy Normal Medium Hard Expert

Article Tags : SQL-Clauses-Operators, SQL

Practice Tags : SQL