L10-DBS301-savepoint rollback etc

Name: Abdul Rehman, Seneca ID : 149019176

You will **create 2 tables** first, then **remove / restore** these tables and also to **add / modify /remove** certain database objects like **views** in this lab.

1. Create table CITIES **from table LOCATIONS,** but only for location numbers less than 2000 (do NOT create this table from scratch).

🡪 You will have exactly 10 rows here.

When you describe CITIES, the output is shown below:

SQL> DESC cities

Name Null? Type

----------------------------------------- -------- -----------------

LOCATION\_ID NUMBER(4)

STREET\_ADDRESS VARCHAR2(40)

POSTAL\_CODE VARCHAR2(12)

CITY NOT NULL VARCHAR2(30)

STATE\_PROVINCE VARCHAR2(25)

COUNTRY\_ID CHAR(2)

Statements:

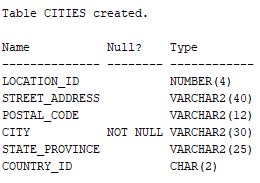
create table cities as

select \*

from locations

where location\_id <2000;

Output:



2. Create table TOWNS **from table LOCATIONS,** but only for location numbers less than 1500 (do NOT create this table from scratch). This table will have same structure as table CITIES.

🡪 You will have exactly 5 rows here.

Statement:

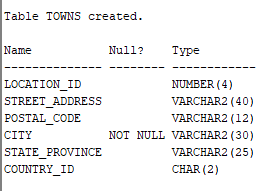
create table towns as

select \*

from cities

where location\_id < 1500;

Output:



3. Now you will insert a row in TOWNS table, then delete the rows with Country\_id CA. Check the new values with SELECT. Later run the command rollback, and then check the values in the table.

4. INSERT INTO cities VALUES (2000, ‘70 Pond Rd’, ‘M1L’, ‘Toronto’, ‘ON’, ‘CA’);

SAVEPOINT Insert\_Done;

DELETE FROM cities WHERE country\_id = ‘IT’;

SAVEPOINT Delete\_Done;

UPDATE city SET city = 'York';

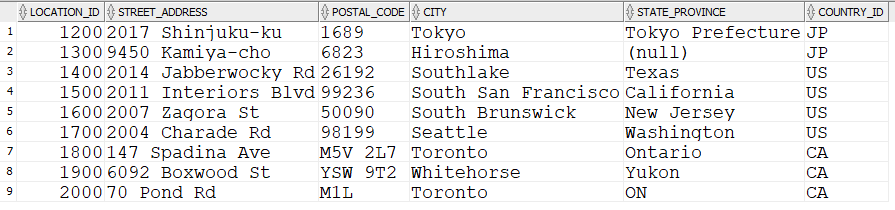
**Statements and Output:**



INSERT INTO cities VALUES (2000, '70 Pond Rd', 'M1L', 'Toronto', 'ON', 'CA');



SAVEPOINT Insert\_Done;



Rows: 9, country\_id =’IT’ deleted.



SAVEPOINT Delete\_Done;

What will be the result if you execute the following command:

* 1. ROLLBACK to SAVEPOINT Delete\_Done;

Cities tables “city” column restored to original values.

* 1. COMMIT Delete\_Done;

Says unknown command: commit deleted\_done;

* 1. Rollback;

Says Rollback Complete but still shows the inserted cities row and the deleted row where country\_id = ‘IT’

5. Add a check constraint in Cities table that will have country\_id as IT and CA only.

alter table cities

add constraint counry\_id\_ck

check (country\_id IN ('CA',’IT’));



6. Create simple view called CA\_CITY\_VU, based on table CITIES so that will contain only columns Street\_Address, City and State\_Province for locations only in CANADA. Then display all data from this view.

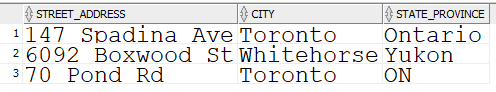
create view CA\_CITY\_VU as

select street\_address, city, state\_province

from cities

where country\_id = 'CA';

select \* from ca\_city\_vu;



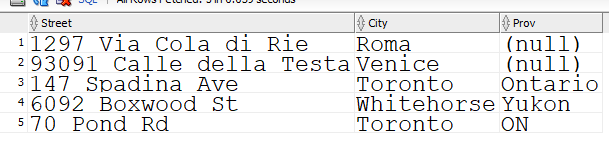
7. Modify your simple view so that will have following aliases instead of original column names: Street, City and Prov and also will include cities from ITALY as well. Then display all data from this view.

create or replace view CA\_CITY\_VU as

select street\_address "Street", city "City", state\_province "Prov"

from cities

where country\_id IN('CA','IT');



8. Create complex view called CITY\_DEPNAME\_VU, based on tables EMPLOYEES, LOCATIONS and DEPARTMENTS, so that will contain only columns Last\_name, Department\_Name, City and State\_Province for locations in ITALY or CANADA. Then display all data from this view.

create view city\_depname\_vu as

select e.last\_name, d.department\_name, l.city, l.state\_province

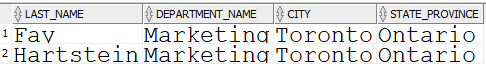
from employees e join departments d

ON e.department\_id = d.department\_id

join locations l

on d.location\_id = l.location\_id

where l.country\_id IN('CA','IT');



9. Modify your complex view so that will have following aliases instead of original column names: Lname, DName, City and Prov and also will include all cities outside United States.

Then display all data from this view.

create or replace view city\_depname\_vu as

select e.last\_name "Lname", d.department\_name "DName", l.city "City", l.state\_province "Prov"

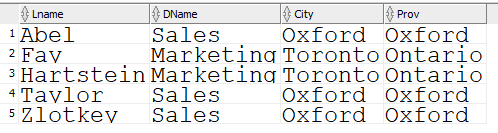
from employees e join departments d

ON e.department\_id = d.department\_id

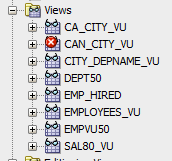
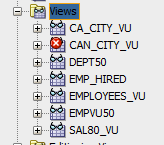
join locations l

on d.location\_id = l.location\_id

where l.country\_id != 'US';



10. Check in the Data Dictionary what Views (their names and definitions) are created so far in your account. Then drop your CITY\_DEPNAME\_VU and check Data Dictionary again. What is different?

Before: After:

City\_depname\_vu is no longer available.