

## Practice for DML (INSERT, UPDATE, & DELETE)

### First...Understanding ROLLBACK vs COMMIT

1. Create a test table called **vendor\_only\_ca** that is based on the following SELECT statement:

```
select vendor_name, vendor_address1, vendor_address2, vendor_city, vendor_phone
from vendors
where vendor_state = 'CA'
order by vendor_city;
```

2. Once the table is successfully created, run the following SELECT statement to see the data in the test table:

```
select * from vendor_only_ca;
```

Since **vendor\_only\_ca** is a “test table” (i.e. a partial copy **vendors**), we can play around and practice DML using this table without concern of messing up data in the other main tables. So let’s get crazy!

3. Write and run an UPDATE statement change the vendor\_name for all vendors to “Who cares!?!”. Once this runs it should update close to 75 rows which is all the vendors on the table. If it runs correct, go rerun the SELECT from #2 above and see if the statement updated all the records.
4. Since this change wasn’t committed we can undo it. Write and run a ROLLBACK statement to undo the change. After you run the ROLLBACK, rerun the SELECT statement again (i.e. from #2) to see if the change was removed.
5. Next, rerun the update of vendor\_name field (from #3) again but this time run a COMMIT statement to save the change. After you run the commit, rerun your select to see if the change is there. Try rolling back the change now that you’ve run the commit and then SELECT \* to see if the change was reverted or not.
6. Lastly, write a statement to drop the **vendor\_only\_ca** table

---

### Practice basic INSERT, UPDATE, and DELETE – REMINDER: Be sure to commit changes after you run DML

1. Insert a new member in the **ubc\_members** table but don’t specify their years\_oc
2. Add this member to the SP20 VIP Committee. *Hint: you’ll need to look up the committee\_id on **ubc\_committee***
3. Update the first name of the member whose uteid is **ieo328**. Update first from Igor to Iggy.
4. Update members phone column to be NULL if their years on council is > 1
5. Update all members’ years\_oc to increase by 1.
6. Delete the new member you added and only that member. *Hint: if you get an error deleting member, figure it out*
7. **ADVANCED EXAMPLE:** Delete all members that have 4 years on council from their assigned committees in **ubc\_member\_committee**. To avoid having to look up members’ year\_oc to lookup uteid, try a subquery. To do this you will first write a query that pulls all uteids from ubc\_members for members that have 4 years\_oc. Then make that be the subquery with a query that deletes from members where utied in (<place your select statement here>)

```
--Use this to check if your inserts, updates, and deletes worked for ubc_members
select * from ubc_members;
```

```
--Use this to check if your inserts, updates, and deletes worked for ubc_committees
select * from ubc_committees;
```

```
--Use this to check if your inserts, updates, and deletes worked for ubc_members_committees
select * from ubc_member_committees;
```

```
--Use this to join all tables and confirm inserts work across all tables
Select m.first_name, c.committee_name, c.semester_year
from ubc_members m inner join ubc_member_committees mc on m.uteid = mc.uteid
inner join ubc_committees c on mc.committee_id = c.committee_id;
```

### **BONUS PRACTICE SUBQUERY:**

1. First, write a select statement that pulls a list of vendor\_ids from a certain location like vendor\_state of CA
2. Update invoice\_date for just those vendors to be 20-FEB-2020
3. Delete invoices for just those vendors.