**Name: Avinash Singh**

**Lab 10**

**Prof.: Ayesha Manzer**

You will **create 2 tables** firstly, 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:

CREATE TABLE CITIES AS

SELECT LOCATION\_ID,

STREET\_ADDRESS,

POSTAL\_CODE,

CITY,

STATE\_PROVINCE,

COUNTRY\_ID

FROM LOCATIONS

WHERE LOCATION\_ID < 2000;

DESC CITIES;

SELECT \* FROM CITIES;

--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.

CREATE TABLE TOWNS AS

SELECT LOCATION\_ID,

STREET\_ADDRESS,

POSTAL\_CODE,

CITY,

STATE\_PROVINCE,

COUNTRY\_ID

FROM LOCATIONS

WHERE LOCATION\_ID < 1500;

DESC TOWNS;

SELECT \* FROM TOWNS;

--3. NOW you will EMPTY your RECYCLE BIN WITH ONE powerful command. THEN REMOVE your TABLE TOWNS, so that will remain IN the RECYCLE bin.

-- CHECK that it IS really there AND what TIME was removed.

PURGE RECYCLEBIN;

DROP TABLE TOWNS;

SELECT \* FROM RECYCLEBIN;

--4. Restore your table TOWNS from recycle bin and describe it. Check what is in your recycle bin now.

FLASHBACK TABLE TOWNS TO BEFORE DROP;

SELECT \* FROM TOWNS;

--5. Now remove table TOWNS so that does NOT remain in the recycle bin. Check that is really NOT there and then try to restore it. Explain what happened?

DROP TABLE TOWNS;

FLASHBACK TABLE TOWNS TO BEFORE DROP;

--6. CREATE simple VIEW called CAN\_CITY\_VU, based ON TABLE CITIES so that will contain

-- only columns Street\_Address, Postal\_Code, City and State\_Province for locations only in CANADA. Then display all data from this view.

CREATE OR REPLACE VIEW CAN\_CITY\_VU AS

SELECT Street\_Address,

Postal\_Code,

City,

State\_Province

FROM LOCATIONS

WHERE COUNTRY\_ID = 'CA';

SELECT \* FROM CAN\_CITY\_VU;

--7. MODIFY your simple VIEW so that will have FOLLOWING aliases INSTEAD OF original

-- column names: Str\_Adr, P\_Code, City and Prov and also will include cities from ITALY as well. Then display all data from this view.

DROP VIEW CAN\_CITY\_VU;

CREATE VIEW CAN\_CITY\_VU AS

SELECT Street\_Address AS "Str\_Adr",

Postal\_Code AS "P\_Code",

City, State\_Province AS "Prov"

FROM CITIES

WHERE COUNTRY\_ID = ‘CA';

--8. CREATE complex VIEW called CITY\_DNAME\_VU, based ON TABLES LOCATIONS AND DEPARTMENTS,

-- so that will contain ONLY COLUMNS Department\_Name, City AND State\_Province FOR locations IN ITALY OR CANADA.

-- Include situations even when city does NOT have department established yet. Then display all data from this view.

CREATE VIEW CITY\_DNAME\_VU AS

SELECT D.DEPARTMENT\_NAME,

L.City,

L.State\_Province

FROM Locations L

LEFT JOIN Departments D

ON L.Location\_ID = D.Location\_ID

WHERE COUNTRY\_ID IN ('CA','IT');

SELECT \* FROM CITY\_DNAME\_VU;

--9. MODIFY your complex VIEW so that will have FOLLOWING aliases

-- INSTEAD OF original COLUMN names: DName, City AND Prov AND also will INCLUDE ALL cities outside United States

-- Include situations even when city does NOT have department established yet. Then display all data from this view.

CREATE VIEW CITY\_DNAME\_VU AS

SELECT Department\_Name AS "DName",

City,

State\_Province AS "Prov"

FROM LOCATIONS L

FULL OUTER JOIN DEPARTMENTS D

ON (L.LOCATION\_ID = D.DEPARTMENT\_ID)

WHERE COUNTRY\_ID not in ('US');

--10. CHECK IN the DATA DICTIONARY what Views (their names AND definitions) ARE created so FAR IN your ACCOUNT.

-- Then drop your CITY\_DNAME\_VU and check Data Dictionary again. What is different?

SELECT OBJECT\_NAME,

OBJECT\_TYPE

FROM USER\_OBJECTS

ORDER BY OBJECT\_NAME;