**(LAB-TASK DATED: 5-12-2020) Database consistency transactions Dashboard ORACLE LAB (IMCS)**

1. Select whole data from REGION table in order of region\_id
2. (region\_id, region\_name) VALUES (5, 'Africa')
3. Check the added rows
4. Save permanently all previous three transactions
5. As a result of 2, the REGIONS table has a region called 'Middle East and Africa' and a region called 'Africa'. 5. corrects this problem (a very simple transaction) and checks the change, but then rolls back the transaction and checks the rollback.
6. Rolling Back an Entire Transaction
7. Select the roll backed record of region table
8. Transaction (change table):

Update region table change region\_name is ‘middle east ’ which is ‘middle east and afarica’

1. Check the updated record
2. Again rollback the record

Setting Save points in Transactions

1. Check countries in region 4 before transaction (country name,region id, country id which are in region id no 4 and naming order)
2. UPDATE REGIONS

SET REGION\_NAME = 'Middle East'

WHERE REGION\_NAME = 'Middle East and Africa'

1. UPDATE COUNTRIES

SET REGION\_ID = 5

WHERE COUNTRY\_ID = 'ZM';

SAVEPOINT zambia;

1. UPDATE COUNTRIES

SET REGION\_ID = 5

WHERE COUNTRY\_ID = 'NG';

SAVEPOINT nigeria;

1. UPDATE COUNTRIES

SET REGION\_ID = 5

WHERE COUNTRY\_ID = 'ZW';

SAVEPOINT zimbabwe;

1. UPDATE COUNTRIES

SET REGION\_ID = 5

WHERE COUNTRY\_ID = 'EG';

SAVEPOINT egypt;

1. Check REGIONS table after transaction:

SELECT \* FROM REGIONS

ORDER BY REGION\_ID;

1. Check countries in region 4 after transaction same as in 11:
2. Check countries in region 5 after transaction:
3. SELECT COUNTRY\_NAME, COUNTRY\_ID, REGION\_ID

FROM COUNTRIES

WHERE REGION\_ID = 5

ORDER BY COUNTRY\_NAME;

1. Rollback to savepoint Nigeria;
2. Check REGIONS table after rollback (by select)
3. Check countries in region 4 after rollback verify the result

COUNTRY\_NAME CO REGION\_ID

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

Egypt EG 4

Israel IL 4

Kuwait KW 4

Zimbabwe ZW 4

1. Check countries in region 5 after rollback:

COUNTRY\_NAME CO REGION\_ID

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

Nigeria NG 5

Zambia ZM 5

Submit the above tasks on given email [navttcoracleerp@gmail.com](mailto:navttcoracleerp@gmail.com) on due date.

***using the INSERT Statement When All Information Is Available***

***VALUES (***

***10, -- EMPLOYEE\_ID***

***'George', -- FIRST\_NAME***

***'Gordon', -- LAST\_NAME***

***'GGORDON', -- EMAIL***

***'650.506.2222', -- PHONE\_NUMBER***

***'01-JAN-07', -- HIRE\_DATE***

***'SA\_REP', -- JOB\_ID***

***9000, -- SALARY***

***.1, -- COMMISSION\_PCT***

***148, -- MANAGER\_ID***

***80 -- DEPARTMENT\_ID***

***);***

***Result:***

***1 row created.***

***You need not know all column values to insert a row into a table, but you must know the values of all NOT NULL columns. If you do not know the value of a column that can be NULL, you can omit that column from list\_of\_columns. Its value defaults to NULL.***

***The INSERT statement in***[***Example 3-2***](https://docs.oracle.com/database/121/TDDDG/tdddg_dml.htm#CHDJCDJC)***inserts a row into the EMPLOYEES table for an employee for which all column values are known except SALARY. For now, SALARY can have the value NULL. When you know the salary, you can change it with the UPDATE statement (see***[***Example 3-4***](https://docs.oracle.com/database/121/TDDDG/tdddg_dml.htm#CHDCDCCB)***).***

***Example 3-2 Using the INSERT Statement When Not All Information Is Available***

***JOB\_ID, -- Omit SALARY; its value defaults to NULL.***

***VALUES (***

***20, -- EMPLOYEE\_ID***

***'John', -- FIRST\_NAME***

***'Keats', -- LAST\_NAME***

***'JKEATS', -- EMAIL***

***'650.506.3333', -- PHONE\_NUMBER***

***'01-JAN-07', -- HIRE\_DATE***

***'SA\_REP', -- JOB\_ID***

***.1, -- COMMISSION\_PCT***

***148, -- MANAGER\_ID***

***80 -- DEPARTMENT\_ID***

***);***

***Result:***

***1 row created.***

***The INSERT statement in***[***Example 3-3***](https://docs.oracle.com/database/121/TDDDG/tdddg_dml.htm#CHDCJECE)***tries to insert a row into the EMPLOYEES table for an employee for which LAST\_NAME is not known.***

***Example 3-3 Using the INSERT Statement Incorrectly***

***FIRST\_NAME, -- Omit LAST\_NAME (error)***

***VALUES (***

***20, -- EMPLOYEE\_ID***

***'John', -- FIRST\_NAME***

***'JOHN', -- EMAIL***

***'650.506.3333', -- PHONE\_NUMBER***

***'01-JAN-07', -- HIRE\_DATE***

***'SA\_REP', -- JOB\_ID***

***.1, -- COMMISSION\_PCT***

***148, -- MANAGER\_ID***

***80 -- DEPARTMENT\_ID***

***);***

***Result: ORA-01400: cannot insert NULL into ("HR"."EMPLOYEES"."LAST\_NAME")***