**Work Task-(5-12-2020) :**

1. Select whole data from REGION table in order of region\_id

Query Solution: (select \*

from regions

order by region\_id)

statement : select all available data from the regions table.

1. region\_id, region\_name) VALUES (5, 'Africa')

Query Solution: { insert into regions(region\_id, region\_name)

VALUES (5, 'Africa') }

statement : Add new row in regions table where region\_id = 5, region\_name = Africa;

1. Check the added rows

statement : check that row is added or not in regions table.

1. Save permanently all previous three transactions

Query Solution: (commit;)

statement : to save changes permanently.

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

Query Solution: (savepoint regions;)

statement : to savepoint the transactions.

1. Rolling Back an Entire Transaction

Query Solution: (rollback to regions;)

statement : to rollback the transactions.

1. Select the roll backed record of region table

Query Solution: (select \*

from regions)

statement : to that region able is rollbacked or not.

1. Transaction (change table):

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

Query Solution: (update regions

set region\_name = 'Africa'

where region\_name = 'Middle East and Africa')

statement : to update region\_name to Africa which was Middle East and Africa.

1. Check the updated record

Query Solution: (select \*

from regions)

statement : to check the updated table.

1. Again rollback the record

Query Solution-1: (rollback to regions)

Query Solution-2

(select \*

from regions)

statement : do again rollback and then check the table.

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

Query Solution: (select \*

from countries

where region\_id = 4

order by country\_name;)

statement : to check countries in region\_id having 4.

1. UPDATE REGIONS

SET REGION\_NAME = 'Middle East'

WHERE REGION\_NAME = 'Middle East and Africa'

Query Solution: (update regions

Set region\_name = ‘Middle East’

Where region\_name =’Middle East and Africa’)

statement : : to update region\_name to Middle East which was Middle East and Africa.

1. UPDATE COUNTRIES

SET REGION\_ID = 5

WHERE COUNTRY\_ID = 'ZM';

Query Solution: (update contries

Set region\_id = 5

Where country\_id = ‘ZM’)

statement : to update region\_id to 5 to the country country\_id = ZM.

1. UPDATE COUNTRIES

SET REGION\_ID = 5

WHERE COUNTRY\_ID = 'NG';

Query Solution: (update contries

Set region\_id = 5

Where country\_id= ‘NG’)

statement : to update region\_id to 5 to the country country\_id = NG.

1. UPDATE COUNTRIES

SET REGION\_ID = 5

WHERE COUNTRY\_ID = 'ZW';

Query Solution: (update contries

Set region\_id = 5

Where country\_id= ‘ZW’)

statement : to update region\_id to 5 to the country country\_id = ZW.

1. UPDATE COUNTRIES

SET REGION\_ID = 5

WHERE COUNTRY\_ID = 'EG';

SAVEPOINT egypt;

Query Solution: (update contries

Set region\_id = 5

Where country\_id= ‘EG’

Savepoint egypt)

statement : to update region\_id to 5 to the country country\_id = ZW and savepoint Egypt.

1. Check REGIONS table after transaction:

SELECT \* FROM REGIONS

ORDER BY REGION\_ID;

Query Solution: (select \*

From regions

Order by region\_by)

statement : to view the table that rows are updated.

1. Check countries in region 4 after transaction same as in 11:

Query Solution: (select \*

from countries

where region\_id = 4

order by country\_name;)

statement : to check countries in region\_id having 4.

1. Check countries in region 5 after transaction:

Query Solution: (select \*

from countries

where region\_id = 5

order by country\_name;)

statement : to check countries in region\_id having 5.

1. SELECT COUNTRY\_NAME, COUNTRY\_ID, REGION\_ID

FROM COUNTRIES

WHERE REGION\_ID = 5

ORDER BY COUNTRY\_NAME;

Query Solution:( select country\_name, country\_id, region\_id

from countries

where region\_id = 5

order by country\_name;)

statement : to view country\_name, country\_id, region\_id in countries table having region\_id = 5.

1. Rollback to savepoint Nigeria;

Query Solution:( Rollback to savepoint Nigeria;)

statement : To Rollback to the last savepoint of Nigeria

1. Check REGIONS table after rollback (by select)

Query Solution:( select \*

from regions)

statement : To check results after rollbacked.

1. 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

Query Solution:( select \*

from countries

where region\_id = 4

order by country\_name;)

statement : to check countries in region\_id having 4 and order by country\_name.

1. Check countries in region 5 after rollback:

COUNTRY\_NAME CO REGION\_ID

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

Nigeria NG 5

Zambia ZM 5

Query Solution:( select \*

from countries

where region\_id = 5

order by country\_name;)

Statement: to check countries in region\_id having 5 and order by country\_name.

25.Insert row in a Table:

Query Solution:( insert into employees;)

statement : To insert the row in the employees table.

***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***

***);***

Query Solution:( insert into employees (employee\_id, first\_name, email, phone\_number, hire\_date, job\_id, commission\_pct, manager\_id, department\_id)

values (20, 'John', 'JOHN', '650.506.3333', '01-JAN-07', 'SA\_REP', .1, 148, 80);

)

statement : To insert the row in the employees table.