```
SQL> set echo on:
SP2-0158: unknown SET option ":"
SQL>
SQL> DROP TABLE classes;

Table dropped.

SQL>
SQL> REM:**********************Part - I : DML Update operations & TCL statements**********************
SQL>
SQL> REM:Consider the Classes relation that can be described as below:
SQL> REM:The relation Classes records the name of the class - ship class, the type of ships (bb for battleship
SQL> REM:or bc for battle cruiser), the country that built the ship, the number of main guns, the bore
SQL> REM:(diameter of the gun barrel, in inches) of the main guns, and the displacement (weight, in tons).
SQL> REM:Note: Define the relation Classes appropriately to accommodate the following tuples:
SQL>
SQL> CREATE TABLE classes(
  2  class VARCHAR(20) PRIMARY KEY,
  3  type VARCHAR(4) CHECK(type IN('bb', 'bc')),
  4  country VARCHAR(20),
  5  numguns NUMBER(3),
  6  bore NUMBER(3),
  7  displacement NUMBER(10));

Table created.

SQL>
SQL> DESC classes;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------------------------------
 CLASS                                     NOT NULL VARCHAR2(20)
 TYPE                                               VARCHAR2(4)
 COUNTRY                                            VARCHAR2(20)
 NUMGUNS                                            NUMBER(3)
 BORE                                               NUMBER(3)
 DISPLACEMENT                                       NUMBER(10)

SQL>
SQL> REM: 1. Add first two tuples from the above sample data. List the columns explicitly in the INSERT clause.(No ordering
of columns)
SQL>
SQL> INSERT INTO classes(class, type, country, numguns, bore, displacement) VALUES('Bismark', 'bb', 'Germany', 8, 14, 32000);

1 row created.

SQL> INSERT INTO classes(type, class, country, numguns, bore, displacement) VALUES( 'bb', 'Iowa', 'USA', 9, 16, 46000);

1 row created.

SQL>
SQL> REM: 2. Populate the relation with the remaining set of tuples. This time, do not list the columns in the INSERT clause.
```

```
SQL>
SQL> INSERT INTO classes VALUES('Kongo', 'bc', 'Japan', 8, 15, 42000);

1 row created.

SQL> INSERT INTO classes VALUES('North Carolina', 'bb', 'USA', 9, 16, 37000);

1 row created.

SQL> INSERT INTO classes VALUES('Revenge', 'bb', 'Gt. Britain', 8, 15, 29000);

1 row created.

SQL> INSERT INTO classes VALUES('Renown', 'bc', 'Gt. Britain', 6, 15, 32000);

1 row created.

SQL>
SQL> REM: 3. Display the populated relation.
SQL>
SQL> SELECT * FROM classes;

CLASS              TYPE COUNTRY              NUMGUNS      BORE DISPLACEMENT
------------------ ---- -------------------- ---------- ---------- ------------
Bismark            bb   Germany                    8         14        32000
Iowa               bb   USA                        9         16        46000
Kongo              bc   Japan                      8         15        42000
North Carolina     bb   USA                        9         16        37000
Revenge            bb   Gt. Britain                8         15        29000
Renown             bc   Gt. Britain                6         15        32000

6 rows selected.

SQL>
SQL> REM: 4. Mark an intermediate point here in this transaction.
SQL>
SQL> SAVEPOINT table_created_display;

Savepoint created.

SQL>
SQL> REM: 5. Change the displacement of Bismark to 34000.
SQL>
SQL> SELECT * FROM classes;

CLASS              TYPE COUNTRY              NUMGUNS      BORE DISPLACEMENT
------------------ ---- -------------------- ---------- ---------- ------------
Bismark            bb   Germany                    8         14        32000
Iowa               bb   USA                        9         16        46000
Kongo              bc   Japan                      8         15        42000
North Carolina     bb   USA                        9         16        37000
```

```
Revenge              bb   Gt. Britain                 8         15        29000
Renown               bc   Gt. Britain                 6         15        32000

6 rows selected.

SQL> UPDATE classes SET displacement=34000 WHERE class='Bismark';

1 row updated.

SQL> SELECT * FROM classes;

CLASS                TYPE COUNTRY                NUMGUNS      BORE DISPLACEMENT
-------------------- ---- -------------------- ---------- ---------- ------------
Bismark              bb   Germany                     8         14        34000
Iowa                 bb   USA                         9         16        46000
Kongo                bc   Japan                       8         15        42000
North Carolina       bb   USA                         9         16        37000
Revenge              bb   Gt. Britain                 8         15        29000
Renown               bc   Gt. Britain                 6         15        32000

6 rows selected.

SQL>
SQL> REM: 6. For the battleships having at least 9 number of guns or the ships with at least 15 inch bore, increase the
displacement by 10%.
SQL> REM:    Verify your changes to the table.
SQL>
SQL> UPDATE classes SET displacement=displacement+0.1*displacement WHERE numguns>=9 OR bore>=15;

5 rows updated.

SQL> SELECT * FROM classes;

CLASS                TYPE COUNTRY                NUMGUNS      BORE DISPLACEMENT
-------------------- ---- -------------------- ---------- ---------- ------------
Bismark              bb   Germany                     8         14        34000
Iowa                 bb   USA                         9         16        50600
Kongo                bc   Japan                       8         15        46200
North Carolina       bb   USA                         9         16        40700
Revenge              bb   Gt. Britain                 8         15        31900
Renown               bc   Gt. Britain                 6         15        35200

6 rows selected.

SQL>
SQL> REM: 7. Delete Kongo class of ship from Classes table.
SQL>
SQL> DELETE FROM classes WHERE class='Kongo';

1 row deleted.

```

```
SQL>
SQL> REM: 8. Display your changes to the table.
SQL> SELECT * FROM classes;

CLASS                TYPE COUNTRY                NUMGUNS      BORE DISPLACEMENT
-------------------- ---- -------------------- ---------- ---------- ------------
Bismark              bb   Germany                      8         14        34000
Iowa                 bb   USA                          9         16        50600
North Carolina       bb   USA                          9         16        40700
Revenge              bb   Gt. Britain                  8         15        31900
Renown               bc   Gt. Britain                  6         15        35200

SQL>
SQL> REM: 9. Discard the recent updates to the relation without discarding the earlier INSERT operation(s).
SQL>
SQL> ROLLBACK TO table_created_display;

Rollback complete.

SQL> SELECT * FROM classes;

CLASS                TYPE COUNTRY                NUMGUNS      BORE DISPLACEMENT
-------------------- ---- -------------------- ---------- ---------- ------------
Bismark              bb   Germany                      8         14        32000
Iowa                 bb   USA                          9         16        46000
Kongo                bc   Japan                        8         15        42000
North Carolina       bb   USA                          9         16        37000
Revenge              bb   Gt. Britain                  8         15        29000
Renown               bc   Gt. Britain                  6         15        32000

6 rows selected.

SQL>
SQL> REM: 10. Commit the changes.
SQL>
SQL> COMMIT;

Commit complete.

SQL> SELECT * FROM classes;

CLASS                TYPE COUNTRY                NUMGUNS      BORE DISPLACEMENT
-------------------- ---- -------------------- ---------- ---------- ------------
Bismark              bb   Germany                      8         14        32000
Iowa                 bb   USA                          9         16        46000
Kongo                bc   Japan                        8         15        42000
North Carolina       bb   USA                          9         16        37000
Revenge              bb   Gt. Britain                  8         15        29000
Renown               bc   Gt. Britain                  6         15        32000

6 rows selected.
```

```
SQL>
SQL> REM:**********************Part - II : DML Retrieval operations**********************
SQL> REM:Use the employees.sql to create the database and write the SQL statements for the following:
SQL>
SQL> @z:/employees.sql;
SP2-0310: unable to open file "z:/employees.sql"
SQL>
SQL> REM: 11. Display first name, job id and salary of all the employees.
SQL>
SQL> SELECT first_name, job_id, salary FROM employees;

FIRST_NAME           JOB_ID         SALARY
-------------------- ---------- ----------
Steven               AD_PRES         24000
Neena                AD_VP           17000
Lex                  AD_VP           17000
Alexander            IT_PROG          9000
Bruce                IT_PROG          6000
David                IT_PROG          4800
Valli                IT_PROG          4800
Diana                IT_PROG          4200
Kevin                ST_MAN           5800
Trenna               ST_CLERK         3500
Curtis               ST_CLERK         3100

FIRST_NAME           JOB_ID         SALARY
-------------------- ---------- ----------
Randall              ST_CLERK         2600
Peter                ST_CLERK         2500
Eleni                SA_MAN          10500
Ellen                SA_REP          11000
Jonathon             SA_REP           8600
Kimberely            SA_REP           7000
Jennifer             AD_ASST          4400
Michael              MK_MAN          13000
Pat                  MK_REP           6000
Shelley              AC_MGR          12000
William              AC_ACCOUNT       8300

22 rows selected.

SQL>
SQL> REM: 12. Display the id, name(first and last), salary and annual salary of all the employees.
SQL> REM: Sort the employees by first name.
SQL> REM: Label the columns as shown below:(EMPLOYEE_ID, FULL NAME, MONTHLY SAL, ANNUAL SALARY)
SQL>
SQL> SELECT employee_id, first_name || ' ' || last_name AS full_name, salary AS monthly_sal, salary*12 AS annual_sal
  2  FROM employees
  3  ORDER BY first_name;
```

```
EMPLOYEE_ID FULL_NAME                                          MONTHLY_SAL ANNUAL_SAL
----------- ---------------------------------------------- ----------- ----------
        103 Alexander Hunold                                        9000     108000
        104 Bruce Ernst                                             6000      72000
        142 Curtis Davies                                           3100      37200
        105 David Austin                                            4800      57600
        107 Diana Lorentz                                           4200      50400
        149 Eleni Zlotkey                                          10500     126000
        174 Ellen Abel                                             11000     132000
        200 Jennifer Whalen                                         4400      52800
        176 Jonathon Taylor                                         8600     103200
        124 Kevin Mourgos                                           5800      69600
        178 Kimberely Grant                                         7000      84000

EMPLOYEE_ID FULL_NAME                                          MONTHLY_SAL ANNUAL_SAL
----------- ---------------------------------------------- ----------- ----------
        102 Lex De Haan                                            17000     204000
        201 Michael Hartstein                                      13000     156000
        101 Neena Kochhar                                          17000     204000
        202 Pat Fay                                                 6000      72000
        144 Peter Vargas                                            2500      30000
        143 Randall Matos                                           2600      31200
        205 Shelley Higgins                                        12000     144000
        100 Steven King                                            24000     288000
        141 Trenna Rajs                                             3500      42000
        106 Valli Pataballa                                         4800      57600
        206 William Gietz                                           8300      99600

22 rows selected.

SQL>
SQL> REM: 13. List the different jobs in which the employees are working for.
SQL>
SQL> SELECT DISTINCT(job_id) FROM employees;

JOB_ID
----------
IT_PROG
AC_MGR
AC_ACCOUNT
ST_MAN
AD_ASST
AD_VP
SA_MAN
MK_MAN
AD_PRES
SA_REP
MK_REP

JOB_ID
----------
```

```
ST_CLERK

12 rows selected.

SQL>
SQL> REM: 14. Display the id, first name, job id, salary and commission of employees who are earning commissions.
SQL>
SQL> SELECT employee_id, first_name, job_id, salary, commission_pct
  2  FROM employees
  3  WHERE commission_pct IS NOT NULL;

EMPLOYEE_ID FIRST_NAME           JOB_ID         SALARY COMMISSION_PCT
----------- -------------------- ---------- ---------- --------------
        149 Eleni                SA_MAN          10500             .2
        174 Ellen                SA_REP          11000             .3
        176 Jonathon             SA_REP           8600             .2
        178 Kimberely            SA_REP           7000            .15

SQL>
SQL> REM: 15. Display the details (id, first name, job id, salary and dept id) of employees who are MANAGERS.
SQL>
SQL> SELECT DISTINCT(e2.employee_id), e2.first_name, e2.job_id, e2.salary, e2.department_id
  2  FROM employees e1, employees e2
  3  WHERE e1.manager_id=e2.employee_id;

EMPLOYEE_ID FIRST_NAME           JOB_ID         SALARY DEPARTMENT_ID
----------- -------------------- ---------- ---------- -------------
        102 Lex                  AD_VP           17000            90
        103 Alexander            IT_PROG          9000            60
        100 Steven               AD_PRES         24000            90
        124 Kevin                ST_MAN           5800            50
        149 Eleni                SA_MAN          10500            80
        201 Michael              MK_MAN          13000            20
        101 Neena                AD_VP           17000            90
        205 Shelley              AC_MGR          12000           110

8 rows selected.

SQL>
SQL> REM: 16. Display the details of employees other than sales representatives (id, first name, hire date, job id, salary
and dept id)
SQL> REM:     who are hired after `01May1999' or whose salary is at least 10000.
SQL>
SQL> SELECT employee_id, first_name, hire_date, job_id, salary, department_id
  2  FROM employees
  3  WHERE (hire_date > TO_DATE('1999-05-01','YYYY-MM-DD') OR salary>=10000) AND job_id <> 'SA_REP';

EMPLOYEE_ID FIRST_NAME           HIRE_DATE JOB_ID         SALARY DEPARTMENT_ID
----------- -------------------- --------- ---------- ---------- -------------
        100 Steven               17-JUN-87 AD_PRES         24000            90
        101 Neena                21-SEP-89 AD_VP           17000            90
```

```
       102 Lex                  13-JAN-93 AD_VP          17000            90
       124 Kevin                16-NOV-99 ST_MAN          5800            50
       149 Eleni                29-JAN-00 SA_MAN         10500            80
       201 Michael              17-FEB-96 MK_MAN         13000            20
       205 Shelley              07-JUN-94 AC_MGR         12000           110

7 rows selected.

SQL>
SQL> REM: 17. Display the employee details (first name, salary, hire date and dept id)
SQL> REM:     whose salary falls in the range of 5000 to 15000 and his/her name begins with any of characters (A,J,K,S). Sort
the output by first name.
SQL>
SQL> SELECT first_name, salary, hire_date, department_id
  2  FROM employees WHERE salary BETWEEN 5000 AND 15000 AND first_name LIKE 'A%' OR first_name LIKE 'J%' OR first_name LIKE
  'K%' OR first_name LIKE 'S%'
  3  ORDER BY first_name;

FIRST_NAME               SALARY HIRE_DATE DEPARTMENT_ID
-------------------- ---------- --------- -------------
Alexander                  9000 03-JAN-90            60
Jennifer                   4400 17-SEP-87            10
Jonathon                   8600 24-MAR-98            80
Kevin                      5800 16-NOV-99            50
Kimberely                  7000 24-MAY-99
Shelley                   12000 07-JUN-94           110
Steven                    24000 17-JUN-87            90

7 rows selected.

SQL>
SQL> REM: 18. Display the experience of employees in no. of years and months who were hired after 1998.
SQL> REM:     Label the columns as: (EMPLOYEE_ID, FIRST_NAME, HIRE_DATE, EXPYRS, EXPMONTHS).
SQL>
SQL> SELECT employee_id, first_name, hire_date, EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM hire_date) AS expyrs,
  2  (EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM hire_date))*12 AS expmonths
  3  FROM employees
  4  WHERE hire_date > TO_DATE('31-12-1998','DD-MM-YYYY');

EMPLOYEE_ID FIRST_NAME           HIRE_DATE     EXPYRS  EXPMONTHS
----------- -------------------- --------- ---------- ----------
       107 Diana                07-FEB-99         21        252
       124 Kevin                16-NOV-99         21        252
       149 Eleni                29-JAN-00         20        240
       178 Kimberely            24-MAY-99         21        252

SQL>
SQL> REM: 19. Display the total number of departments.
SQL>
SQL> SELECT COUNT(DISTINCT(DEPARTMENT_ID)) FROM employees;
```

```
COUNT(DISTINCT(DEPARTMENT_ID))
------------------------------
                             7

SQL>
SQL> REM: 20. Show the number of employees hired by yearwise. Sort the result by yearwise.
SQL>
SQL> SELECT COUNT(*) AS num_employees, EXTRACT(year from hire_date) AS hire_yr
  2  FROM employees
  3  GROUP BY EXTRACT(year from hire_date)
  4  ORDER BY EXTRACT(year from hire_date);

NUM_EMPLOYEES    HIRE_YR
------------- ----------
            2       1987
            1       1989
            1       1990
            1       1991
            1       1993
            2       1994
            1       1995
            2       1996
            3       1997
            4       1998
            3       1999

NUM_EMPLOYEES    HIRE_YR
------------- ----------
            1       2000

12 rows selected.

SQL>
SQL> REM: 21. Display the minimum, maximum and average salary, number of employees for each department.
SQL> REM:     Exclude the employee(s) who are not in any department.
SQL> REM:     Include the department(s) with at least 2 employees and the average salary is more than 10000.
SQL> REM:     Sort the result by minimum salary in descending order.
SQL>
SQL> SELECT MIN(salary) AS min_sal, MAX(salary) AS max_sal, AVG(salary) AS avg_sal, COUNT(*) AS num_employees, department_id
  2  FROM employees WHERE department_id IS NOT NULL
  3  GROUP BY department_id
  4  HAVING COUNT(*) > 1 AND AVG(salary) > 10000 ORDER BY min_sal DESC;

   MIN_SAL    MAX_SAL    AVG_SAL NUM_EMPLOYEES DEPARTMENT_ID
---------- ---------- ---------- ------------- -------------
     17000      24000 19333.3333             3            90
      8600      11000 10033.3333             3            80
      8300      12000      10150             2           110

SQL>
SQL> spool off
```