

```

1  SQL> set echo on:
2  SP2-0158: unknown SET option ":"
3  SQL>
4  SQL> DROP TABLE classes;
5
6  Table dropped.
7
8  SQL>
9  SQL> REM:*****Part - I : DML Update operations & TCL statements*****
10 SQL>
11 SQL> REM:Consider the Classes relation that can be described as below:
12 SQL> REM:The relation Classes records the name of the class - ship class, the type of ships (bb for battleship
13 SQL> REM:or bc for battle cruiser), the country that built the ship, the number of main guns, the bore
14 SQL> REM:(diameter of the gun barrel, in inches) of the main guns, and the displacement (weight, in tons).
15 SQL> REM>Note: Define the relation Classes appropriately to accommodate the following tuples:
16 SQL>
17 SQL> CREATE TABLE classes(
18   2 class VARCHAR(20) PRIMARY KEY,
19   3 type VARCHAR(4) CHECK(type IN('bb', 'bc')),
20   4 country VARCHAR(20),
21   5 numguns NUMBER(3),
22   6 bore NUMBER(3),
23   7 displacement NUMBER(10));
24
25 Table created.
26
27 SQL>
28 SQL> DESC classes;
29 Name Null? Type
30 -----
31 CLASS NOT NULL VARCHAR2(20)
32 TYPE VARCHAR2(4)
33 COUNTRY VARCHAR2(20)
34 NUMGUNS NUMBER(3)
35 BORE NUMBER(3)
36 DISPLACEMENT NUMBER(10)
37
38 SQL>
39 SQL> REM: 1. Add first two tuples from the above sample data. List the columns explicitly in the INSERT clause. (No ordering
of columns)
40 SQL>
41 SQL> INSERT INTO classes(class, type, country, numguns, bore, displacement) VALUES('Bismark', 'bb', 'Germany', 8, 14, 32000);
42
43 1 row created.
44
45 SQL> INSERT INTO classes(type, class, country, numguns, bore, displacement) VALUES('bb', 'Iowa', 'USA', 9, 16, 46000);
46
47 1 row created.
48
49 SQL>
50 SQL> REM: 2. Populate the relation with the remaining set of tuples. This time, do not list the columns in the INSERT clause.

```

```

51 SQL>
52 SQL> INSERT INTO classes VALUES('Kongo', 'bc', 'Japan', 8, 15, 42000);
53
54 1 row created.
55
56 SQL> INSERT INTO classes VALUES('North Carolina', 'bb', 'USA', 9, 16, 37000);
57
58 1 row created.
59
60 SQL> INSERT INTO classes VALUES('Revenge', 'bb', 'Gt. Britain', 8, 15, 29000);
61
62 1 row created.
63
64 SQL> INSERT INTO classes VALUES('Renown', 'bc', 'Gt. Britain', 6, 15, 32000);
65
66 1 row created.
67
68 SQL>
69 SQL> REM: 3. Display the populated relation.
70 SQL>
71 SQL> SELECT * FROM classes;

```

```

72
73 CLASS ..... TYPE COUNTRY ..... NUMGUNS ..... BORE DISPLACEMENT .....
74 -----
75 Bismark ..... bb ..... Germany ..... 8 ..... 14 ..... 32000 .....
76 Iowa ..... bb ..... USA ..... 9 ..... 16 ..... 46000 .....
77 Kongo ..... bc ..... Japan ..... 8 ..... 15 ..... 42000 .....
78 North Carolina ..... bb ..... USA ..... 9 ..... 16 ..... 37000 .....
79 Revenge ..... bb ..... Gt. Britain ..... 8 ..... 15 ..... 29000 .....
80 Renown ..... bc ..... Gt. Britain ..... 6 ..... 15 ..... 32000 .....

```

```

81
82 6 rows selected.
83
84 SQL>
85 SQL> REM: 4. Mark an intermediate point here in this transaction.
86 SQL>
87 SQL> SAVEPOINT table_created_display;
88
89 Savepoint created.
90
91 SQL>
92 SQL> REM: 5. Change the displacement of Bismark to 34000.
93 SQL>
94 SQL> SELECT * FROM classes;

```

```

95
96 CLASS ..... TYPE COUNTRY ..... NUMGUNS ..... BORE DISPLACEMENT .....
97 -----
98 Bismark ..... bb ..... Germany ..... 8 ..... 14 ..... 32000 .....
99 Iowa ..... bb ..... USA ..... 9 ..... 16 ..... 46000 .....
100 Kongo ..... bc ..... Japan ..... 8 ..... 15 ..... 42000 .....
101 North Carolina ..... bb ..... USA ..... 9 ..... 16 ..... 37000 .....

```

```

102 Revenge ..... bb ..... Gt. Britain ..... 8 ..... 15 ..... 29000 .....
103 Renown ..... bc ..... Gt. Britain ..... 6 ..... 15 ..... 32000 .....
104
105 6 rows selected.
106
107 SQL> UPDATE classes SET displacement=34000 WHERE class='Bismark';
108
109 1 row updated.
110
111 SQL> SELECT * FROM classes;
112
113 CLASS ..... TYPE ..... COUNTRY ..... NUMGUNS ..... BORE ..... DISPLACEMENT .....
114 -----
115 Bismark ..... bb ..... Germany ..... 8 ..... 14 ..... 34000 .....
116 Iowa ..... bb ..... USA ..... 9 ..... 16 ..... 46000 .....
117 Kongo ..... bc ..... Japan ..... 8 ..... 15 ..... 42000 .....
118 North Carolina ..... bb ..... USA ..... 9 ..... 16 ..... 37000 .....
119 Revenge ..... bb ..... Gt. Britain ..... 8 ..... 15 ..... 29000 .....
120 Renown ..... bc ..... Gt. Britain ..... 6 ..... 15 ..... 32000 .....
121
122 6 rows selected.
123
124 SQL>
125 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%.
126 SQL> REM: ..... Verify your changes to the table.
127 SQL>
128 SQL> UPDATE classes SET displacement=displacement+0.1*displacement WHERE numguns>=9 OR bore>=15;
129
130 5 rows updated.
131
132 SQL> SELECT * FROM classes;
133
134 CLASS ..... TYPE ..... COUNTRY ..... NUMGUNS ..... BORE ..... DISPLACEMENT .....
135 -----
136 Bismark ..... bb ..... Germany ..... 8 ..... 14 ..... 34000 .....
137 Iowa ..... bb ..... USA ..... 9 ..... 16 ..... 50600 .....
138 Kongo ..... bc ..... Japan ..... 8 ..... 15 ..... 46200 .....
139 North Carolina ..... bb ..... USA ..... 9 ..... 16 ..... 40700 .....
140 Revenge ..... bb ..... Gt. Britain ..... 8 ..... 15 ..... 31900 .....
141 Renown ..... bc ..... Gt. Britain ..... 6 ..... 15 ..... 35200 .....
142
143 6 rows selected.
144
145 SQL>
146 SQL> REM: 7. Delete Kongo class of ship from Classes table.
147 SQL>
148 SQL> DELETE FROM classes WHERE class='Kongo';
149
150 1 row deleted.
151

```

```

152 SQL>
153 SQL> REM: 8. Display your changes to the table.
154 SQL> SELECT * FROM classes;
155
156 CLASS ..... TYPE COUNTRY ..... NUMGUNS ..... BORE DISPLACEMENT .....
157 -----
158 Bismark ..... bb Germany ..... 8 ..... 14 ..... 34000 .....
159 Iowa ..... bb USA ..... 9 ..... 16 ..... 50600 .....
160 North Carolina ..... bb USA ..... 9 ..... 16 ..... 40700 .....
161 Revenge ..... bb Gt. Britain ..... 8 ..... 15 ..... 31900 .....
162 Renown ..... bc Gt. Britain ..... 6 ..... 15 ..... 35200 .....
163
164 SQL>
165 SQL> REM: 9. Discard the recent updates to the relation without discarding the earlier INSERT operation(s).
166 SQL>
167 SQL> ROLLBACK TO table_created_display;
168
169 Rollback complete.
170
171 SQL> SELECT * FROM classes;
172
173 CLASS ..... TYPE COUNTRY ..... NUMGUNS ..... BORE DISPLACEMENT .....
174 -----
175 Bismark ..... bb Germany ..... 8 ..... 14 ..... 32000 .....
176 Iowa ..... bb USA ..... 9 ..... 16 ..... 46000 .....
177 Kongo ..... bc Japan ..... 8 ..... 15 ..... 42000 .....
178 North Carolina ..... bb USA ..... 9 ..... 16 ..... 37000 .....
179 Revenge ..... bb Gt. Britain ..... 8 ..... 15 ..... 29000 .....
180 Renown ..... bc Gt. Britain ..... 6 ..... 15 ..... 32000 .....
181
182 6 rows selected.
183
184 SQL>
185 SQL> REM: 10. Commit the changes.
186 SQL>
187 SQL> COMMIT;
188
189 Commit complete.
190
191 SQL> SELECT * FROM classes;
192
193 CLASS ..... TYPE COUNTRY ..... NUMGUNS ..... BORE DISPLACEMENT .....
194 -----
195 Bismark ..... bb Germany ..... 8 ..... 14 ..... 32000 .....
196 Iowa ..... bb USA ..... 9 ..... 16 ..... 46000 .....
197 Kongo ..... bc Japan ..... 8 ..... 15 ..... 42000 .....
198 North Carolina ..... bb USA ..... 9 ..... 16 ..... 37000 .....
199 Revenge ..... bb Gt. Britain ..... 8 ..... 15 ..... 29000 .....
200 Renown ..... bc Gt. Britain ..... 6 ..... 15 ..... 32000 .....
201
202 6 rows selected.

```

```

203
204 SQL>
205 SQL> REM:*****Part - II : DML Retrieval operations*****
206 SQL> REM:Use the employees.sql to create the database and write the SQL statements for the following:
207 SQL>
208 SQL> @z:/employees.sql;
209 SP2-0310: unable to open file "z:/employees.sql"
210 SQL>
211 SQL> REM: 11. Display first name, job id and salary of all the employees.
212 SQL>
213 SQL> SELECT first_name, job_id, salary FROM employees;
214
215 FIRST_NAME      JOB_ID      SALARY
216 -----
217 Steven          AD_PRES     24000
218 Neena           AD_VP       17000
219 Lex             AD_VP       17000
220 Alexander       IT_PROG     9000
221 Bruce           IT_PROG     6000
222 David           IT_PROG     4800
223 Valli           IT_PROG     4800
224 Diana           IT_PROG     4200
225 Kevin           ST_MAN      5800
226 Trena           ST_CLERK    3500
227 Curtis          ST_CLERK    3100
228
229 FIRST_NAME      JOB_ID      SALARY
230 -----
231 Randall         ST_CLERK    2600
232 Peter           ST_CLERK    2500
233 Eleni           SA_MAN      10500
234 Ellen           SA_REP      11000
235 Jonathon        SA_REP      8600
236 Kimberely       SA_REP      7000
237 Jennifer        AD_ASST     4400
238 Michael         MK_MAN      13000
239 Pat             MK_REP      6000
240 Shelley         AC_MGR      12000
241 William         AC_ACCOUNT  8300
242
243 22 rows selected.
244
245 SQL>
246 SQL> REM: 12. Display the id, name(first and last), salary and annual salary of all the employees.
247 SQL> REM: Sort the employees by first name.
248 SQL> REM: Label the columns as shown below: (EMPLOYEE_ID, FULL NAME, MONTHLY SAL, ANNUAL SALARY)
249 SQL>
250 SQL> SELECT employee_id, first_name || ' ' || last_name AS full_name, salary AS monthly_sal, salary*12 AS annual_sal
251        2 FROM employees
252        3 ORDER BY first_name;
253

```

```

254 EMPLOYEE_ID FULL_NAME MONTHLY_SAL ANNUAL_SAL
255 -----
256 ..... 103 Alexander Hunold ..... 9000 ..... 108000 .....
257 ..... 104 Bruce Ernst ..... 6000 ..... 72000 .....
258 ..... 142 Curtis Davies ..... 3100 ..... 37200 .....
259 ..... 105 David Austin ..... 4800 ..... 57600 .....
260 ..... 107 Diana Lorentz ..... 4200 ..... 50400 .....
261 ..... 149 Eleni Zlotkey ..... 10500 ..... 126000 .....
262 ..... 174 Ellen Abel ..... 11000 ..... 132000 .....
263 ..... 200 Jennifer Whalen ..... 4400 ..... 52800 .....
264 ..... 176 Jonathon Taylor ..... 8600 ..... 103200 .....
265 ..... 124 Kevin Mourgos ..... 5800 ..... 69600 .....
266 ..... 178 Kimberely Grant ..... 7000 ..... 84000 .....

```

```

267
268 EMPLOYEE_ID FULL_NAME MONTHLY_SAL ANNUAL_SAL
269 -----
270 ..... 102 Lex De Haan ..... 17000 ..... 204000 .....
271 ..... 201 Michael Hartstein ..... 13000 ..... 156000 .....
272 ..... 101 Neena Kochhar ..... 17000 ..... 204000 .....
273 ..... 202 Pat Fay ..... 6000 ..... 72000 .....
274 ..... 144 Peter Vargas ..... 2500 ..... 30000 .....
275 ..... 143 Randall Matos ..... 2600 ..... 31200 .....
276 ..... 205 Shelley Higgins ..... 12000 ..... 144000 .....
277 ..... 100 Steven King ..... 24000 ..... 288000 .....
278 ..... 141 Tenna Rajs ..... 3500 ..... 42000 .....
279 ..... 106 Valli Pataballa ..... 4800 ..... 57600 .....
280 ..... 206 William Gietz ..... 8300 ..... 99600 .....

```

281
282 22 rows selected.

```

283
284 SQL>
285 SQL> REM: 13. List the different jobs in which the employees are working for.
286 SQL>
287 SQL> SELECT DISTINCT(job_id) FROM employees;

```

```

288
289 JOB_ID .....
290 -----
291 IT_PROG .....
292 AC_MGR .....
293 AC_ACCOUNT .....
294 ST_MAN .....
295 AD_ASST .....
296 AD_VP .....
297 SA_MAN .....
298 MK_MAN .....
299 AD_PRES .....
300 SA_REP .....
301 MK_REP .....

```

```

302
303 JOB_ID .....
304 -----

```

```

305 ST_CLERK .....
306
307 12 rows selected.
308
309 SQL>
310 SQL> REM: 14. Display the id, first name, job id, salary and commission of employees who are earning commissions.
311 SQL>
312 SQL> SELECT employee_id, first_name, job_id, salary, commission_pct
313 ..2 FROM employees
314 ..3 WHERE commission_pct IS NOT NULL;
315
316 EMPLOYEE_ID FIRST_NAME ..... JOB_ID ..... SALARY COMMISSION_PCT .....
317 -----
318 ..... 149 Eleni ..... SA_MAN ..... 10500 ..... .2 .....
319 ..... 174 Ellen ..... SA_REP ..... 11000 ..... .3 .....
320 ..... 176 Jonathon ..... SA_REP ..... 8600 ..... .2 .....
321 ..... 178 Kimberly ..... SA_REP ..... 7000 ..... .15 .....
322
323 SQL>
324 SQL> REM: 15. Display the details (id, first name, job id, salary and dept id) of employees who are MANAGERS.
325 SQL>
326 SQL> SELECT DISTINCT(e2.employee_id), e2.first_name, e2.job_id, e2.salary, e2.department_id
327 ..2 FROM employees e1, employees e2
328 ..3 WHERE e1.manager_id=e2.employee_id;
329
330 EMPLOYEE_ID FIRST_NAME ..... JOB_ID ..... SALARY DEPARTMENT_ID .....
331 -----
332 ..... 102 Lex ..... AD_VP ..... 17000 ..... 90 .....
333 ..... 103 Alexander ..... IT_PROG ..... 9000 ..... 60 .....
334 ..... 100 Steven ..... AD_PRES ..... 24000 ..... 90 .....
335 ..... 124 Kevin ..... ST_MAN ..... 5800 ..... 50 .....
336 ..... 149 Eleni ..... SA_MAN ..... 10500 ..... 80 .....
337 ..... 201 Michael ..... MK_MAN ..... 13000 ..... 20 .....
338 ..... 101 Neena ..... AD_VP ..... 17000 ..... 90 .....
339 ..... 205 Shelley ..... AC_MGR ..... 12000 ..... 110 .....
340
341 8 rows selected.
342
343 SQL>
344 SQL> REM: 16. Display the details of employees other than sales representatives (id, first name, hire date, job id, salary
and dept id)
345 SQL> REM: ..... who are hired after `01May1999` or whose salary is at least 10000.
346 SQL>
347 SQL> SELECT employee_id, first_name, hire_date, job_id, salary, department_id
348 ..2 FROM employees
349 ..3 WHERE (hire_date > TO_DATE('1999-05-01','YYYY-MM-DD') OR salary>=10000) AND job_id<>'SA_REP';
350
351 EMPLOYEE_ID FIRST_NAME ..... HIRE_DATE JOB_ID ..... SALARY DEPARTMENT_ID .....
352 -----
353 ..... 100 Steven ..... 17-JUN-87 AD_PRES ..... 24000 ..... 90 .....
354 ..... 101 Neena ..... 21-SEP-89 AD_VP ..... 17000 ..... 90 .....

```

```

355 .....102 Lex .....13-JAN-93 AD_VP .....17000 .....90 .....
356 .....124 Kevin .....16-NOV-99 ST_MAN .....5800 .....50 .....
357 .....149 Eleni .....29-JAN-00 SA_MAN .....10500 .....80 .....
358 .....201 Michael .....17-FEB-96 MK_MAN .....13000 .....20 .....
359 .....205 Shelley .....07-JUN-94 AC_MGR .....12000 .....110 .....
360
361 7 rows selected.
362
363 SQL>
364 SQL> REM: 17. Display the employee details (first name, salary, hire date and dept id)
365 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.
366 SQL>
367 SQL> SELECT first_name, salary, hire_date, department_id
368 .....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%'
369 .....3 ORDER BY first_name;
370
371 FIRST_NAME .....SALARY HIRE_DATE DEPARTMENT_ID .....
372 -----
373 Alexander .....9000 03-JAN-90 .....60 .....
374 Jennifer .....4400 17-SEP-87 .....10 .....
375 Jonathon .....8600 24-MAR-98 .....80 .....
376 Kevin .....5800 16-NOV-99 .....50 .....
377 Kimberly .....7000 24-MAY-99 .....
378 Shelley .....12000 07-JUN-94 .....110 .....
379 Steven .....24000 17-JUN-87 .....90 .....
380
381 7 rows selected.
382
383 SQL>
384 SQL> REM: 18. Display the experience of employees in no. of years and months who were hired after 1998.
385 SQL> REM: ..... Label the columns as: (EMPLOYEE_ID, FIRST_NAME, HIRE_DATE, EXPYRS, EXPMONTHS) .
386 SQL>
387 SQL> SELECT employee_id, first_name, hire_date, EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM hire_date) AS expyrs,
388 .....2 (EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM hire_date)) *12 AS expmonths
389 .....3 FROM employees
390 .....4 WHERE hire_date > TO_DATE('31-12-1998','DD-MM-YYYY');
391
392 EMPLOYEE_ID FIRST_NAME .....HIRE_DATE .....EXPYRS .....EXPMONTHS .....
393 -----
394 .....107 Diana .....07-FEB-99 .....21 .....252 .....
395 .....124 Kevin .....16-NOV-99 .....21 .....252 .....
396 .....149 Eleni .....29-JAN-00 .....20 .....240 .....
397 .....178 Kimberly .....24-MAY-99 .....21 .....252 .....
398
399 SQL>
400 SQL> REM: 19. Display the total number of departments.
401 SQL>
402 SQL> SELECT COUNT(DISTINCT(DEPARTMENT_ID)) FROM employees;
403

```



```

404 COUNT(DISTINCT(DEPARTMENT_ID)) .....
405 ----- .....
406 .....7 .....
407
408 SQL>
409 SQL> REM: 20. Show the number of employees hired by yearwise. Sort the result by yearwise.
410 SQL>
411 SQL> SELECT COUNT(*) AS num_employees, EXTRACT(year from hire_date) AS hire_yr
412 ..2 FROM employees
413 ..3 GROUP BY EXTRACT(year from hire_date)
414 ..4 ORDER BY EXTRACT(year from hire_date);
415
416 NUM_EMPLOYEES ..HIRE_YR .....
417 ----- .....
418 .....2 .....1987 .....
419 .....1 .....1989 .....
420 .....1 .....1990 .....
421 .....1 .....1991 .....
422 .....1 .....1993 .....
423 .....2 .....1994 .....
424 .....1 .....1995 .....
425 .....2 .....1996 .....
426 .....3 .....1997 .....
427 .....4 .....1998 .....
428 .....3 .....1999 .....
429
430 NUM_EMPLOYEES ..HIRE_YR .....
431 ----- .....
432 .....1 .....2000 .....
433
434 12 rows selected.
435
436 SQL>
437 SQL> REM: 21. Display the minimum, maximum and average salary, number of employees for each department.
438 SQL> REM: .... Exclude the employee(s) who are not in any department.
439 SQL> REM: .... Include the department(s) with at least 2 employees and the average salary is more than 10000.
440 SQL> REM: .... Sort the result by minimum salary in descending order.
441 SQL>
442 SQL> SELECT MIN(salary) AS min_sal, MAX(salary) AS max_sal, AVG(salary) AS avg_sal, COUNT(*) AS num_employees, department_id
443 ..2 FROM employees WHERE department_id IS NOT NULL
444 ..3 GROUP BY department_id
445 ..4 HAVING COUNT(*) > 1 AND AVG(salary) > 10000 ORDER BY min_sal DESC;
446
447 ..MIN_SAL ..MAX_SAL ..AVG_SAL NUM_EMPLOYEES DEPARTMENT_ID .....
448 ----- .....
449 .....17000 .....24000 19333.3333 .....3 .....90 .....
450 .....8600 .....11000 10033.3333 .....3 .....80 .....
451 .....8300 .....12000 .....10150 .....2 .....110 .....
452
453 SQL>
454 SQL> spool off

```