


[\(/\)](#)[fork \(19\) \(/fork/Fm37x8\)](#) [download \(/plain/Fm37x8\)](#)[copy](#)<https://ideone.com/Fm37x8>language: **Text (text 6.10)**

created: 1 year ago

visibility:  public (</faq#visibility-of-a-code>)[Share or Embed source code](#)

```
<script src="https://ideone.com/e.js/Fm37x8" type="text/javascript" ></script>
```

 **sphere online judge****Learn How to Code**[http://spoj.com/?](http://spoj.com/?utm_campaign=permanent&utm_medium=view-right&utm_source=ideone)[utm\\_campaign=permanent&utm\\_medium=view-right&utm\\_source=ideone\)](http://spoj.com/?utm_campaign=permanent&utm_medium=view-right&utm_source=ideone)**Discover > Sphere Engine API (/sphere-engine)**

The brand new service which powers Ideone!

**Discover > IDE Widget (/sphere-engine-widget)**

Widget for compiling and running the source code in a web browser!



1. 1.1
- 2.
3. 1. List the Name and Designation code of the staff who have joined before Jan 2003 and whose salary range is between 12000 and 25000. Display the columns with user defined Column headers. Hint: Use As clause along with other operators
- 4.
5. SQL>SELECT STAFF\_NAME,DESIGN\_CODE FROM STAFFMASTER WHERE (HIREDATE <'01-JAN-2003') AND STAFF\_SAL BETWEEN 12000 AND 25000;
- 6.
7. 2. List the staff code, name, and department number of the staff who have experience of 18 or more years and sort them based on their experience
- 8.
9. SQL>SELECT STAFF\_CODE,STAFF\_NAME,DEPT\_CODE FROM STAFFMASTER WHERE (MONTHS\_BETWEEN(SYS DATE,HIREDATE))>=216 ORDER BY HIREDATE DESC;
- 10.
- 11.
12. 3. Display the staff details who do not have manager. Hint: Use is null
- 13.
14. SQL>SELECT \* FROM STAFFMASTER WHERE MGR\_CODE IS NULL;
- 15.
- 16.
17. 4. Display the Book details that were published during the period of 2001 to 2004. Also display book details with Book name having the character '&' anywhere.
- 18.
19. SQL>SELECT \* FROM BOOK\_MASTER WHERE BOOK\_PUB\_YEAR BETWEEN 2001 AND 2004 AND BOOK\_NAME LIKE '%"%"';
- 20.
- 21.
22. 5. List the names of the staff having '\_' character in their name.
- 23.
24. SQL>SELECT STAFF\_NAME FROM STAFFMASTER WHERE STAFF\_NAME LIKE '%\_';
- 25.
- 26.
27. 2.1
- 28.
29. 1. Create a query which will display Staff Name, Salary of each staff. Format the salary to be 15 characters long and left padded with '\$'.
- 30.
31. SQL>SELECT STAFF\_NAME,'\$'||STAFF\_SAL AS STAFF\_SALARY FROM STAFFMASTER;
- 32.
33. 2. Display name and date of birth of students where date of birth must be displayed in the format similar to "January, 12 1981" for those who were born on Saturday or Sunday.
- 34.
35. SQL> SELECT STUDENT\_NAME,TO\_CHAR(STUDENTDOB,'MONTH DD YYYY') AS STUDENT\_DOB FROM STUDENTMASTER WHERE TO\_CHAR(STUDENTDOB,'DAY') LIKE ('%SATURDAY%') OR TO\_CHAR(STUDENTDOB,'DAY') LIKE ('%SUNDAY%');
- 36.
37. 3. Display each Staff name and number of months they worked for the organization. Label the column as 'Months Worked'. Order your result by number of months employed. Also Round the number of months to closest whole number
- 38.
39. SQL> SELECT STAFF\_NAME,ROUND(MONTHS\_BETWEEN(SYSDATE,HIREDATE)) AS MONTHS\_WORKED FROM STAFFMASTER ORDER BY MONTHS\_WORKED DESC;
- 40.
41. 4. List the details of the staff who have joined in first half of December month (irrespective of the year)
- 42.
43. SQL>SELECT \* FROM STAFFMASTER WHERE TO\_CHAR(HIREDATE,'DD') BETWEEN 1 AND 15 AND TO\_CHAR(HIREDATE,'MONTH') LIKE '%DECEMBER%';
- 44.



45. 5. Write a query that displays Staff Name, Salary, and Grade of all staff. Grade depends on the following table.

46. Salary Grade

47. Salary >=50000 A

48. Salary >= 25000 < 50000 B

49. Salary >=10000 < 25000 C

50. OTHERS D

51.

52. SQL>SELECT STAFF\_NAME,STAFF\_SAL,

53. CASE

54. WHEN STAFF\_SAL >=50000 THEN 'A'

55. WHEN STAFF\_SAL >25000 AND STAFF\_SAL<50000 THEN 'B'

56. WHEN STAFF\_SAL >10000 AND STAFF\_SAL<25000 THEN 'C'

57. ELSE 'D'

58. END CASE

59. FROM STAFFMASTER;

60.

61.

62.

63. 6. Display the Staff Name, Hire date and day of the week on which staff was hired. Label the column as DAY. Order the result by the day of the week starting with Monday. Hint :Use to\_char with hiredate and formats 'DY' and 'D'

64.

65. SQL>SELECT STAFF\_NAME,TO\_CHAR(HIREDATE,'DD MONTH YYYY') AS HIRE\_DATE,TO\_CHAR(HIREDATE,'DAY')AS DAY FROM STAFFMASTER ORDER BY TO\_CHAR(HIREDATE,'DAY') DESC;

66.

67. 7. Write a query to find the position of third occurrence of 'i' in the given word 'Mississippi'.

68.

69. SQL> SELECT INSTR('Mississippi','i',2,3) FROM DUAL;

70.

71. 8. Write a query to find the pay date for the month. Pay date is the last Friday of the month. Display the date in the format "Twenty Eighth of January, 2002". Label the heading as PAY DATE. Hint: use to\_char, next\_day and last\_day functions

72.

73. SQL>SELECT TO\_CHAR(NEXT\_DAY(SYSDATE,'TUESDAY'),'DD MONTH ,YYYY') AS DAY FROM DUAL WHERE NEXT\_DAY(SYSDATE,'TUESDAY')<LAST\_DAY(SYSDATE) ;

74.

75.

76. 9. Display Student code, Name and Dept Name. Display "Electricals" if dept code = 20, "Electronics" if Dept code =30 and "Others" for all other Dept codes in the Dept Name column. Hint : Use Decode

77.

78. SQL> SELECT STUDENT\_CODE,STUDENT\_NAME,DECODE(DEPT\_CODE,20,'ELECTRICALS',30,'ELECTRONICS','OTHERS') DEPARTMENT\_NAME FROM STUDENTMASTER;

79.

80.

81. 3.1

82.

83.

84. 3.1: Joins and Subqueries

85. 1. Write a query which displays Staff Name, Department Code, Department Name, and Salary for all staff who earns more than 20000.

86.

87. SQL> SELECT S.STAFF\_NAME,

88. D.DEPT\_CODE,

89. D.DEPT\_NAME,

90. S.STAFF\_SAL

91. FROM STAFFMASTER S,

92. DEPARTMENT\_MASTER D

93. WHERE S.DEPT\_CODE=D.DEPT\_CODE

94. AND STAFF\_SAL >20000;



```

95.
96. 2. Display Staff Code, Staff Name, Department Name, and his manager's number and nam
    e. Label the columns Staff#, Staff, Mgr#, Manager.
97.
98. SQL> SELECT S.STAFF_CODE AS STAFF# ,
99.     S.STAFF_NAME      AS STAFF,
100.     D.DEPT_NAME,
101.     S.MGR_CODE AS MGR#
102. FROM STAFFMASTER S,
103.     DEPARTMENT_MASTER D
104. WHERE S.DEPT_CODE=D.DEPT_CODE;
105.
106. 3. Create a query that will display Student Code, Student Name, Book Code, and Book N
    ame for all students whose expected book return date is today.
107.
108. SQL> SELECT S.STUDENT_CODE,S.STUDENT_NAME,B.BOOK_CODE,BB.BOOK_NAME FROM STUDENTMASTER
    S,BOOK_TRANSACTIONS B, BOOK_MASTER BB WHERE S.STUDENT_CODE=B.STUDENT_CODE AND TO_CHAR
    (B.BOOK_EXPECTED_RETURN_DATE,'DD MM YYYY') LIKE TO_CHAR(SYSDATE,'DD MM YYYY');
109.
110. SQL>SELECT S.STUDENT_CODE,
111.     S.STUDENT_NAME,
112.     B.BOOK_CODE,
113.     BB.BOOK_NAME
114. FROM STUDENTMASTER S,
115.     BOOK_TRANSACTIONS B,
116.     BOOK_MASTER BB
117. WHERE S.STUDENT_CODE=B.STUDENT_CODE
118. AND TO_CHAR(B.BOOK_EXPECTED_RETURN_DATE,'DD MM YYYY') LIKE TO_CHAR(SYSDATE,'DD MM YYYY
    Y');
119.
120. 4. Create a query that will display Staff Code, Staff Name, Department Name, Designat
    ion name, Book Code, Book Name, and Issue Date for only those staff who have taken an
    y book in last 30 days. . If required, make changes to the table to create such a sce
    nario. HH
121.
122. SQL>SELECT S.STAFF_CODE,S.STAFF_NAME,D.DEPT_NAME,F.DESIGN_NAME,G.BOOK_NAME,H.BOOK_ISS
    UE_DATE FROM STAFFMASTER S,DEPARTMENT_MASTER D,DESIGNATION-MASTER F,BOOK_MASTER F,BOO
    K_ISSUE_DATE H WHERE MONTHS_BETWEEN(TO_CHAR(H.BOOK_ISSUE_DATE,'MM'),TO_CHAR(SYSDAT
    E,'MM'))<1;
123.
124.
125. 5. Generate a report which contains the following information.
126. Staff Code, Staff Name, Designation Name, Department, Book Code, Book Name,
127. Author, Fine For the staff who has not returned the book. Fine will be calculated as
    Rs. 5 per day.
128. Fine = 5 * (No. of days = Current Date - Expected return date). Include records in th
    e table to suit this problem statement
129.
130.
131.
132. 6. List Staff Code, Staff Name, and Salary for those who are getting less than the av
    erage salary of organization.
133.
134. SQL>SELECT Staff_Code, Staff_Name,STAFF_SAL FROM STAFFMASTER WHERE STAFF_SAL<(SELECT
    AVG(STAFF_SAL) FROM STAFFMASTER);
135.
136. 7. Display Author Name, Book Name for those authors who wrote more than one book.
137.
138. SQL>SELECT AUTHOR,BOOK_NAME FROM BOOK_MASTER GROUP BY AUTHORNAME HAVING COUNT(AUTHOR)
    >1;
139.
140.

```



141. 8. Display Staff Code, Staff Name, and Department Name for those who have taken more than one book.

142.

143. SQL>SELECT S.Staff\_Code,D.Staff\_Name,D.DEPT\_NAME FROM STAFFMASTER S,BOOK\_TRANSACTIONS D GROUP BY S.STAFF\_NAME HAVING COUNT(D.STAFF\_NAME)>1;

144.

145. 9. Display the Student Code, Student Name, and Department Name for that department in which there are maximum number of student studying.

146.

147. SQL> SELECT S.STUDENT\_CODE,S.STUDENT\_NAME,D.DEPT\_NAME FROM STAFFMASTER S,DEPARTMENT\_MASTER D GROUP BY S.DEPT\_CODE HAVING MAX(S.DEPT\_CODE);

148.

149. 10. Display Staff Code, Staff Name, Department Name, and Designation name for those who have joined in last 3 months.

150.

151. SQL>SELECT S.Staff\_Code,S.Staff\_Name,D.DEPT\_NAME,F.DESIGN\_NAME FROM STAFFMASTER S, DEPARTMENT\_MASTER D,DESIGNATION\_MASTER F WHERE MONTHS\_BETWEEN(TO\_CHAR(HIREDATE,'MM'),TO\_CHAR(SYSDATE,'MM'))<3;

152.

153. 11. Display the Manager Name and the total strength of his/her team.

154.

155.

156.

157. 12. Display the details of books that have not been returned and expected return date was last Monday. Book name should be displayed in proper case..

158. Hint: You can change /add records so that the expected return date suits this problem statement

159.

160.

161.

162. 13. Write a query to display number of people in each Department. Output should display Department Code, Department Name and Number of People.

163.

164. SQL> SELECT DEPT\_CODE,DEPT\_NAME,COUNT(S.STAFF\_NAME) AS NUMBEROFPEOPLE FROM STAFFMASTER S,DEPARTMENT\_MASTER D GROUP BY DEPT\_CODE;

165.

166.

167.

168. 4.1

169.

170. 4.1: Database Objects

171.

172.

173. 1. Create the Customer table with the following columns.

174. CustomerId Number(5)

175. Cust\_Name varchar2(20)

176. Address1 Varchar2(30)

177. Address2 Varchar2(30)

178. ==>create table customer

179. (

180. customerid number(5),

181. cust\_name varchar2(20),

182. Address1 varchar2(30),

183. Address2 varchar2(30)

184. );

185.

186. 2. Modify the Customer table Cust\_Name column of datatype with Varchar2(30), rename the column to CustomerName and it should not accept Nulls.

187. ==>Alter table customer rename column cust\_name to customername;

188. ==>Alter table customer modify customername varchar2(30) Not Null;

189.

190. 3. a) Add the following Columns to the Customer table.



```

191.      Gender Varchar2(1)
192.      Age Number(3)
193.      PhoneNo Number(10)
194.      ==>Alter table customer add Gender varchar2(1);
195.      ==>Alter table customer add Age Number(3);
196.      ==>Alter table customer add phoneNo(10);
197.      b) Rename the Customer table to Cust_Table
198.      ==>Rename customer to cust_table;
199.
200. 4. Insert rows with the following data in to the Customer table.
201.      ==>insert into cust_table(&CustomerId,'&cust_Name','&Address1','&Address2','&G
ender','&Age,&phoneNo);
202.      ==>1000, 'Allen', '#115 Chicago', '#115 Chicago', 'M', '25, 7878776'
203.      ==>1001, George, #116 France, #116 France, M, 25, 434524
204.      ==>1002, Becker, #114 New York, #114 New York, M, 45, 431525
205.
206. 5. Add the Primary key constraint for CustomerId with the name CustId_Prim.
207.
208.      ==>Alter table cust_table add constraints Custid_prim PRIMARY KEY (customerid);
209.
210. 6. Insert the row given below in the Customer table and see the message generated by
the Oracle server.
211.      1002, John, #114 Chicago, #114 Chicago, M, 45, 439525
212.
213.      ==>
214.
215. 7. Disable the constraint on CustomerId, and insert the following data:
216.      1002, Becker, #114 New York, #114 New york , M, 45, 431525
217.      1003, Nanapatekar, #115 India, #115 India , M, 45, 431525
218.
219.      ==>Alter table cust_table drop PRIMARY KEY custid_prim;
220.
221. 8. Enable the constraint on CustomerId of the Customer table, and see the message ge
nerated by the Oracle server.
222.
223.      ==>Alter table cust_table add constraints Custid_prim PRIMARY KEY (customerid);
224.      ==>
225.
226. 9. Drop the constraint CustId_Prim on CustomerId and insert the following Data. Alte
r Customer table, drop constraint Custid_Prim.
227.      1002, Becker, #114 New York, #114 New york , M, 45, 431525, 15000.50
228.      1003, Nanapatekar, #115 India, #115 India , M, 45, 431525, 20000.50
229.
230.      ==>Alter table cust_table drop PRIMARY KEY custid_prim;
231.      ==>Insert into cust_table(1002, Becker, #114 New York, #114 New york , M, 45,4
31525, 15000.50);
232.      ==>Insert into cust_table(1003, Nanapatekar, #115 India, #115 India , M, 45, 4
31525,20000.50);
233.
234. 10. Delete all the existing rows from Customer table, and let the structure remain it
self using TRUNCATE statement.
235.      ==>TRUNCATE table cust_table;
236.
237.
238. 11. In the Customer table, add a column E_mail.
239.      ==>Alter table add e_mail varchar2(30);
240.
241.
242. 12. Drop the E_mail column from Customer table.
243.      ==>Alter table cust_table DROP e_mail;

```



```

244.
245. 13. Create the Suppliers table based on the structure of the Customer table. Include
    only the CustomerId, CustomerName, Address1, Address2, and phoneno columns.
246.     Name the columns in the new table as SuppID, SName, Addr1, Addr2, and Contactno
    respectively.
247.     ==>create table Suppliers as select(customerid as suppid,customername as snam
e,address1 as addr1,address2 as addr2,phoneno as contactno) from cust_table;
248.
249. 14. Drop the above table and recreate the following table with the name CustomerMaste
r.
250.     Customerid Number(5) Primary key(Name of constraint is CustId_PK)
251.     CustomerName Varchar2(30) Not Null
252.     Address1 Varchar2(30) Not Null
253.     Address2 Varchar2(30)
254.     Gender Varchar2(1)
255.     Age Number(3)
256.     PhoneNo Number(10)
257.     ==>Drop table Suppliers;
258.     ==>create table customermaster(customerid(10) primary key(custid_pk),customern
ame varchar2(30),Address1 varchar2(30),Address2 varchar2(30),Gender varchar2(1),Age n
umber(3),phoneno number(10));
259.
260. 15. Create the AccountsMaster table with the following Columns. Use sequence to gener
ate Account number
261.     Customerid Number(5)
262.     AccountNumber Number(10,2) Primary key(Name of constraint is Acc_PK)
263.     AccountType Char(3)
264.     LedgerBalance Number(10,2) Not Null
265.     ==>Create table Accountmaster(customerid number(5),Accountnumber number(10) pr
imary key(acno),accounttype char(3),ledgerbalance number(10) Not Null);
266.     ==>Create sequence seq_ano
267.     MINVALUE 101
268.     MAXVALUE 10000
269.     START WITH 101
270.     INCREMENT BY 1
271.     CACHE 101;
272.
273. 16. Relate AccountsMaster table and CustomerMaster table through CustomerId column wi
th the constraint name Cust_acc.
274.     ==>Alter table Accountmaster ADD constraint ass_fk FOREIGN KEY(customerid) REF
ERENCES customermaster(customerid);
275.
276. 17. Insert the following rows to the CustomerMaster table:
277.     1000, Allen, #115 Chicago, #115 Chicago, M, 25, 7878776
278.     1001, George, #116 France, #116 France, M, 25, 434524
279.     1002, Becker, #114 New York, #114 New York, M, 45, 431525
280.     ==>Insert into customermaster values(1000, Allen, #115 Chicago, #115 Chicago,
M, 25, 7878776);
281.     ==>Insert into customermaster values(1001, George, #116 France, #116 France,
M, 25, 4345240;
282.     ==>Insert into customermaster values(1002, Becker, #114 New York, #114 New Yor
k, M, 45, 4315250;
283.
284. 18. Modify the AccountMaster table with the Check constraint to ensure AccountType sh
ould be either NRI or IND.
285.     ==>alter table Accountmaster add constraint ck_ac check(accounttype='NRI' or ac
counttype='IND');
286.
287. 19. Modify the AccountsMaster table keeping a Check constraint with the name Balance_
Check for the Minimum Balance which should be greater than 5000.
288.     ==>alter table Accountmaster add constraint Balance_check(ledger balance > 50
00);

```



```

289.
290. 20. Modify the AccountsMaster table such that if Customer is deleted from Customer ta
ble then all his details should be deleted from AccountsMaster table.
291.     ==>Delete from Accountmaster,customer table where customerid = 1001
292.
293.
294. 21. Create Backup copy for the AccountsMaster table with the name 'AccountDetails'.
295.
296.     ==>Create table accountdetails as select * from Accountmaster;
297.
298. 22. Create a view 'Acc_view' with columns CustomerId, CustomerName, AccountNumber, Ac
countType, and LedgerBalance from AccountsMaster. In the view Acc_view, the colum
n names should be CustomerCode, AccountHolderName, AccountNumber, Type, and B
alance for the respective columns from AccountsMaster table.
299.     ==>CREATE VIEW Acc_view AS SELECT(Customerid,Customername,Accountnumber,Accoun
tType,ledgerBalance)
300.         from AccountMaster;
301.
302. 23. Create a view on AccountsMaster table with name vAccs_Dtls. This view should list
all customers whose AccountType is 'IND' and their balance amount should not be less
than 10000. Using this view any DML operation should not violate the view condition
s.
303.     ==>CREATE VIEW vAccs_Dtls AS SELECT Accounttype,ledgerbalance from Accountmas
ter where accounttype = 'IND' and ledgerbalance < 10000;
304.
305. 24. Create a view accsvw10 which will not allow DML statement against it.
306.     ==>
307.
308. 25. Create a Sequence with the name Seq_Dept on Deptno column of Department_Masters t
able. It should start from 40 and stop at 200. Increment parameter for the sequence S
eq_Dept should be in step of 10.
309.     ==>CREATE sequence SEQ_DEPT minvalue 40 start with 40
310.         increment by 10 MAX VALUE 200 cache 40;
311.
312. 26. Insert three sample rows by using the above sequence in Department_Masters table.
313.     ==>create table departmentmaster(deptno number(50),Dname varchar2(25),location
varchar2(25));
314.     ==>insert into departmentmaster values(seq_dept.NEXTVAL,'MARKETING','NEW DELH
I');
315.     ==>insert into departmentmaster values(seq_dept.NEXTVAL,'SALES','chennai');
316.     ==>insert into departmentmaster values(seq_dept.NEXTVAL,'RESEARCH','BOSTON');
317.
318. 27. Drop the Seq_Dept sequence.
319.
320.     ==>DROP sequence seq_dept;
321.
322. 28. Get information on the index No_Name from the Data Dictionary.
323.     ==>CREATE INDEX no_name on emp(empno);
324.     ==>select * from emp;
325. 29. Create synonym synEmp for the EMP table.
326.     ==>create SYNONYM synemp for emp;
327.
328. 30. Get Information on synonym synEmp from the Data Dictionary.
329.     ==>select * from synemp;
330.
331. 31. Note: Perform this after creating the Employee Table mentioned in the next Lab as
signment. Create Index on HireDate column and give the name as idx_emp_hiredate for t
his object.
332.     ==>CREATE INDEX IDX_EMP_HIREDATE on emp(HIREDATE);
333.
334.

```





```

335. 32. Create a Sequence with the name Seq_Emp on Empno column of Employee table. It should start from 1001. Try to set Minimum value for this sequence which is less than / greater than 1001, use the sequence to generate Empno while inserting records in Employee table and check the values generated.
336.
337.      ==> ==>CREATE sequence SEQ_EMP minvalue 1001 start with 1001
338.      increment by 1 cache 1001;
339.
340. 5.1
341.
342. 5.1: Data Manipulation Language
343. 1.Create Employee table with same structure as EMP table.
344. SQL>Create table employee as select * from emp where 1=3;
345. SQL>desc employee;
346.
347. Name      Null?     Type
348. EMPNO      NOT NULL   NUMBER(4)
349. ENAME              VARCHAR2(10)
350. JOB              VARCHAR2(50)
351. MGR              NUMBER(4)
352. HIREDATE          DATE
353. SAL              NUMBER(7,2)
354. COMM              NUMBER(7,2)
355. DEPTNO           NUMBER(2)
356.
357. SQL>select * from employee
358.
359. 2. Write a query to populate Employee table using EMP table's empno, ename, sal, deptno columns.
360.
361. SQL>select * from employee;
362. EMPNO  ENAME  JOB    MGR  HIREDATE  SAL    COMM  DEPTNO
363. 7369   SMITH                800      20
364. 7499   ALLEN                1600     30
365. 7521   WARD                1250     30
366. 7566   JONES                2975     20
367. 7654   MARTIN              1250     30
368. 7698   BLAKE                2850     30
369. 7782   CLARK                2450     10
370. 7788   SCOTT                3000     20
371. 7839   KING                5000     10
372. 7844   TURNER              1500     30
373. 7876   ADAMS                1100     20
374. 7900   JAMES                950      30
375. 7902   FORD                3000     20
376. 7934   MILLER              1300     10
377. 14 rows selected.
378.
379. 3. Write a query to change the job and deptno of employee whose empno is 7698 to the job and deptno of employee having empno 7788.
380.
381.      SQL> update table employee set job=(select job from employee where empno=7788),deptno=(select deptno from employee where empno=7788) where empno=7698;
382.
383. 4. Delete the details of department whose department name is 'SALES'.
384.
385.      SQL> delete from employee where departmentname like '%sales%';
386.
387. 5. Write a query to change the deptno of employee with empno 7788 to that of employee having empno 7698.
388.

```



```

389.      SQL>update table employee set deptno=(select deptno from employee where deptno=77
      88) where deptno=7698;
390.
391. 6. Insert the following rows to the Employee table through parameter substitution.
392.
393. • SQL> insert into emp (empno,'ename','job',mgr,'hiredate',sal,comm,deptno) values
      (1000,Allen, Clerk,1001,12-jan-01, 3000, 2,10);
394. • SQL> insert into emp (empno,'ename','job',mgr,'hiredate',sal,comm,deptno) values
      (1001,George, analyst, null, 08 Sep 92, 5000,0, 10);
395. • SQL> insert into emp (empno,'ename','job',mgr,'hiredate',sal,comm,deptno) values
      (1002, Becker, Manager, 1000, 4 Nov 92, 2800,4, 20);
396. • SQL> insert into emp (empno,'ename','job',mgr,'hiredate',sal,comm,deptno) values
      (1003, 'Bill', Clerk, 1002, 4 Nov 92,3000, 0, 20);
397.
398.
399. 6.1
400.
401. 6.1: Transaction Control Language Statements
402. 1. Insert rows with the following data into the Customer table.
403.
404.      SQL>insert into customermaster (customerid,'customername','address1','address
      2','gender',age,'phoneno) values ( 6000, John, #115 Chicago, #115 Chicago, M, 25, 787
      8776, 10000 );
405.
406.      SQL>insert into customermaster (customerid,'customername','address1','address
      2','gender',age,'phoneno) values ( 6001, Jack, #116 France, #116 France, M, 25, 4345
      24, 20000 );
407.
408.      SQL>insert into customermaster (customerid,'customername','address1','address
      2','gender',age,'phoneno) values ( 6002, James, #114 New York, #114 New York, M, 45,
      431525, 15000.50);
409.
410.
411. 2. Create a Savepoint named 'SP1' after third record in the Customer table .
412.      SQL>insert into customermaster (customerid,'customername','address1','address
      2','gender',age,'phoneno) values ( 6000, John, #115 Chicago, #115 Chicago, M, 25, 787
      8776, 10000 );
413.
414.      SQL>insert into customermaster (customerid,'customername','address1','address
      2','gender',age,'phoneno) values ( 6001, Jack, #116 France, #116 France, M, 25, 4345
      24, 20000 );
415.
416.      SQL>insert into customermaster (customerid,'customername','address1','address
      2','gender',age,'phoneno) values ( 6002, James, #114 New York, #114 New York, M, 45,
      431525, 15000.50);
417.
418.      SQL> savepoint p1;
419.
420.
421. 3. Insert the below row in the Customer table.
422.      SQL>insert into customermaster (customerid,'customername','address1','address
      2','gender',age,'phoneno) values ( 6003, John, #114 Chicago, #114 Chicago, M, 45, 43
      9525, 19000.60);
423.
424.
425. 4. Execute rollback statement in such a way that whatever manipulations done before
      Savepoint sp1 are permanently implemented, and the ones after Savepoint SP1 are not
      stored as a part of the Customer table.
426.
427.      SQL>rollback p1;
428.
429.

```



430.  
431.  
432.  
433.  
434.  
435.  
436.  
437.


Not running #stdin #stdout 0s 0KB

 comments (?)

 stdin

 copy

Standard input is empty

 stdout

 copy

Standard output is empty

Sphere Research Labs ([http://sphere-research.com?utm\\_campaign=permanent&utm\\_medium=footer&utm\\_source=ideone](http://sphere-research.com?utm_campaign=permanent&utm_medium=footer&utm_source=ideone)). Ideone is powered by  
Sphere Engine™ ([http://sphere-engine.com/?utm\\_campaign=permanent&utm\\_medium=footer&utm\\_source=ideone](http://sphere-engine.com/?utm_campaign=permanent&utm_medium=footer&utm_source=ideone))  
[home \(/\)](#) [api \(/sphere-engine\)](#) [widget \(/sphere-engine-widget\)](#) [language](#) [faq \(/faq\)](#) [credits \(/credits\)](#) [desktop](#) [mobile](#)  
[\(/switch/mobile/l0ztmzd4oa==\)](#)  
[terms of service \(/legal-tos\)](#) [privacy policy \(/legal-pp\)](#) [gdpr info \(/legal-gdpr\)](#)  
[Feedback & Bugs \(/ideone/Tools/bug/form/1/link/Fm37x8/compiler/62\)](#)

