(1)

(a) DDL

```
Connected.
SQL> create table murli(roll_no number(5),division char(5),address char(15),clas
 char(5));

Table created.

SQL> alter table murli add surname char(10);

Table altered.

SQL> desc murli;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------

 ROLL_NO                                            NUMBER(5)
 DIVISION                                           CHAR(5)
 ADDRESS                                            CHAR(15)
 CLAS                                               CHAR(5)
 SURNAME                                            CHAR(10)

SQL> drop table murli;

Table dropped.

SQL>
```

(b) DML

```
SQL> insert into murli values(5,'a','kandivali','sybsc');

1 row created.

SQL> select * from murli;

   ROLL_NO DIVIS ADDRESS         CLASS
---------- ----- --------------- -----
         5 a     kandivali       sybsc

SQL> update murli set mob_no=95654 where roll_no=5;
update murli set mob_no=95654 where roll_no=5
                *
ERROR at line 1:
ORA-00904: "MOB_NO": invalid identifier


SQL> update murli set roll_no=10 where roll_no=5;

1 row updated.

SQL> delete from murli;

1 row deleted.
```

(c) Constraints.

```
SQL*Plus: Release 11.2.0.2.0 Production on Tue Jun 27 16:14:01 2017

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

SQL> connect system
Enter password:
Connected.
SQL> create table product(pid number(3),pname char(20),address char(10));

Table created.

SQL> insert into product values(101,'jeans','kandivli');

1 row created.

SQL> insert into product values(102,'phone','anderi');

1 row created.

SQL> create table order1(oid number(3),oname char(20),pid number(3));

Table created.

SQL> insert into product values(201,'order1',101);

1 row created.

SQL> insert into product values(202,'order2',102);

1 row created.

SQL> delete from product ;

4 rows deleted.

SQL> insert into product values(101,'jeans','kandivli');

1 row created.

SQL> insert into product values(102,'phone','anderi');
```

```
1 row created.

SQL> insert into order1 values(201,'order1',101);

1 row created.

SQL> insert into order1 values(202,'order2',102);

1 row created.

SQL> select * from product;

       PID PNAME                ADDRESS
---------- -------------------- ----------
       101 jeans                kandivli
       102 phone                anderi

SQL> select * from order1;

       OID ONAME                       PID
---------- -------------------- ----------
       201 order1                      101
       202 order2                      102

SQL> ALTER table product add PRIMARY KEY (pid);

Table altered.

SQL> ALTER table Order1 add FOREIGN KEY (pid) REFERENCES product(pid);

Table altered.

SQL>
```

(2)

(a) Select order by

```
Table created.

SQL> insert into employee values(201,'suraj','5000');

1 row created.

SQL> insert into employee values(202,'dhiraj','6000');

1 row created.

SQL> insert into employee values(203,'sachin','7000');

1 row created.

SQL> select * from employee order by salary desc;

      E_ID E_NAME              SALARY
---------- --------------- ----------
       203 sachin                7000
       202 dhiraj                6000
       201 suraj                 5000

SQL> select * from employee order by salary asc;

      E_ID E_NAME              SALARY
---------- --------------- ----------
       201 suraj                 5000
       202 dhiraj                6000
       203 sachin                7000

SQL> select * from employee where e_id=201;

      E_ID E_NAME              SALARY
---------- --------------- ----------
       201 suraj                 5000
```

(b) Select group by

```
Table altered.

SQL> insert into employee2('s01','manish',10000,'it');
insert into employee2('s01','manish',10000,'it')
                      *
ERROR at line 1:
ORA-00928: missing SELECT keyword


SQL> insert into employee2 values('s01','manish',10000,'it');

1 row created.

SQL> insert into employee2 values('s02','rakesh',15000,'commercial');

1 row created.

SQL> insert into employee2 values('s02','mishra',5000,'farmer');

1 row created.

SQL> insert into employee2 values('s03','ranjan',25000,'teacher');

1 row created.

SQL> select dept,sum(salary) from employee2 group by dept;

DEPT       SUM(SALARY)
---------- -----------
farmer            5000
it               10000
commercial       15000
teacher          25000
```

(c) Aggregate function.

```
SQL> create table employee(e_id char(5),e_name char(10),dept char(10),salary num
ber(15));

Table created.

SQL> insert into employee values('s01','manish','it',15000);

1 row created.

SQL> insert into employee values('s02','ajay','commercial',10000);

1 row created.

SQL> insert into employee values('s03','satish','farmer',2000);

1 row created.

SQL> insert into employee values('s04','risabh','business',20000);

1 row created.

SQL> select sum(salry) from employee;
select sum(salry) from employee
           *
ERROR at line 1:
ORA-00904: "SALRY": invalid identifier


SQL> select sum(salary) from employee;

SUM(SALARY)
-----------
      47000
SQL> select avg(salary) from employee;

AVG(SALARY)
-----------
      11750

SQL> select min(salary) from employee;

MIN(SALARY)
-----------
       2000

SQL> select max(salary) from employee;

MAX(SALARY)
-----------
      20000

SQL> _
```

(3)

(a) Sql join.

```
SELECT * FROM student  OUTER JOIN fees on (student.sid=fees.fid)
                                          *
ERROR at line 1:
ORA-00904: "STUDENT"."SID": invalid identifier


SQL> SELECT * FROM student  left JOIN fees on (student.sid=fees.fid);

       SID SNAME                    ROLL_NO ADDRESS            FID   TOTALFEE
---------- -------------------- ---------- ---------- ---------- ----------
       103 sachin                       17 kandivli
       102 murli                        16 kandivli
       101 Dhiraj kumar sinha           25 Anderi
       104 ajay                         18 kandivli

SQL> SELECT * FROM student  inner JOIN fees on (student.sid=fees.fid);

no rows selected

SQL> SELECT * FROM student  inner JOIN fees where (student.sid=fees.fid);
SELECT * FROM student  inner JOIN fees where (student.sid=fees.fid)
                                  *
ERROR at line 1:
ORA-00905: missing keyword


SQL> SELECT * FROM student FULL OUTER JOIN fees on (student.sid=fees.fid);

       SID SNAME                    ROLL_NO ADDRESS            FID   TOTALFEE
---------- -------------------- ---------- ---------- ---------- ----------
                                                       201      10000
                                                       202       8000
                                                       203      12000
                                                       204      15000
       103 sachin                       17 kandivli
       102 murli                        16 kandivli
       101 Dhiraj kumar sinha           25 Anderi
       104 ajay                         18 kandivli

8 rows selected.
```

(b) Sub-query.

```
SQL> select * from product1;

P_ID  P_NAME     COMPANY    UNIT_PRICE
----- ---------- ---------- ----------
p1    shampoo    lux                50
p2    shoes      nike             5000
p3    book       classmate         100
p3    shirt      addidas          1000

SQL> create table order2(o_id char(10),p_id char(3),c_name char(20),total_unit n
umber(5));
create table order2(o_id char(10),p_id char(3),c_name char(20),total_unit number
(5))
            *
ERROR at line 1:
ORA-00955: name is already used by an existing object


SQL> create table order3(o_id char(10),p_id char(3),c_name char(20),total_unit n
umber(5));

Table created.

SQL> insert into order3 values('o1','p1','dhiraj',500);

1 row created.

SQL> insert into order3 values('o2','p2','murli',550);

1 row created.

SQL> insert into order3 values('o3','p3','richa',555);

1 row created.
```

```
SQL> insert into order3 values('o4','p3','rahul',556);

1 row created.

SQL> select c_name from order3
  2  where p_id in (select p_id from product1 where unit_price>100);

C_NAME
--------------------
murli
rahul
richa

SQL> select * from product1 where unit_price >(select avg(unit_price) from produ
ct1);

P_ID  P_NAME     COMPANY    UNIT_PRICE
----- ---------- ---------- ----------
p2    shoes      nike             5000

SQL>
```

(4)

(a) Creating view

```
SQL> select * from student;

       SID DOB       SNAME          SALARY        AGE
---------- --------- ---------- ---------- ----------
         1 10-SEP-98 sinha           10000         12
         2 01-OCT-99 dhiraj           8000         15
         3 01-OCT-99 kumar            8000         18
         4 10-SEP-79 murli           12000         20

SQL> create view raj as select sid,sname,salary from student where sid=1;

View created.

SQL> select * from raj;

       SID SNAME          SALARY
---------- ---------- ----------
         1 sinha           10000

SQL>
```

(b) Create index, sequence and synonym.

```
SQL> select * from product;

       PID PNAME      PRICE CNAME
---------- ----- ---------- ----------
       101 oil           50 mukesh
       102 sampo        110 suresh
       103 soap         120 mahesh
       104 deo          120 ramu

SQL> create index ind_first on product(cname);

Index created.

SQL> create sequence sinha
  2  start with 105
  3  increment by 1
  4  maxvalue 1000
  5  cycle;

Sequence created.

SQL> insert into product values(sinha.nextval,'oil',220,'rakesh');

1 row created.

SQL> select * from product;


       PID PNAME      PRICE CNAME
---------- ----- ---------- ----------
       101 oil           50 mukesh
       102 sampo        110 suresh
       103 soap         120 mahesh
       104 deo          120 ramu
       105 oil          220 rakesh

SQL> create synonym syn for product;

Synonym created.

SQL> select * from syn product;

       PID PNAME      PRICE CNAME
---------- ----- ---------- ----------
       101 oil           50 mukesh
       102 sampo        110 suresh
       103 soap         120 mahesh
       104 deo          120 ramu
       105 oil          220 rakesh
```

(5).

(a) Grant revoke privileges.

```
SQL> create user sinha identified by sinha;

User created.

SQL> grant create session to sinha;

Grant succeeded.

SQL> connect sinha
Enter password:
Connected.
SQL> connect system
Enter password:
Connected.
SQL> grant create table to sinha;

Grant succeeded.

SQL> alter user sinha default tablespace users temporary tablespace temp quota
  2  unlimited on users;

User altered.

SQL> connect sinha
Enter password:
Connected.
SQL> create table stud(sid number(3),sname char(10));

Table created.
```



```
SQL> connect system
Enter password:
Connected.
SQL> revoke create session from sinha;

Revoke succeeded.

SQL> drop user sinha cascade;

User dropped.

SQL> connect sinha
Enter password:
ERROR:
ORA-01017: invalid username/password; logon denied


Warning: You are no longer connected to ORACLE.
SQL>
```

(b) Role by

```
SQL> select * from product;

       PID PNAME      PRICE CNAME
---------- ----- ---------- ----------
       101 oil           50 mukesh
       102 sampo        110 suresh
       103 soap         120 mahesh
       104 deo          120 ramu
       105 oil          220 rakesh

SQL> create role sy;

Role created.

SQL> grant create session to sy;

Grant succeeded.

SQL> grant create table to sy;

Grant succeeded.

SQL> create user sinha identified by sinha;

User created.
```

```
SQL> create user kumar identified by kumar;

User created.

SQL> connect sinha
Enter password:
ERROR:
ORA-01045: user SINHA lacks CREATE SESSION privilege; logon denied


Warning: You are no longer connected to ORACLE.
SQL> connect system
Enter password:
Connected.
SQL> grant sy to sinha,kumar;

Grant succeeded.

SQL> connect sinha
Enter password:
Connected.
SQL> create table stud(sid number(3),sname char(10));
create table stud(sid number(3),sname char(10))
*
ERROR at line 1:
ORA-01950: no privileges on tablespace 'SYSTEM'
```


(6)

(a) Set operation.

```
SQL> select * from journey;
      J_ID J_DATE      TRAIN_NO
---------- --------- ----------
      1001 10-SEP-17       1244
      1002 14-SEP-17       1247
      1003 10-SEP-17       1249
      1004 14-SEP-17       1259

SQL> select * from train;

  TRAIN_NO TRAIN_NAME SRC        DES                                  ---------- ---------- ---------- ----------
      1244 rajdhani   mumbai     delhi
      1247 godan      mumbai     pnbe
      1249 superfast  mumbai     cpr
      1259 superfast  delhi      mumbai

SQL> select * from passenger;

      P_ID P_NAME            AGE
---------- ---------- ----------
       101 rahul              18
       102 raj                20
       103 raju               30
       104 dhiraj             19
```

```
SQL> select * from ticket;

T_I       J_ID CLA COA   BIRTH_NO       P_ID       FARE
--- ---------- --- --- ---------- ---------- ----------
t1        1001 sl  s1          20        101        750
t2        1002 ac  s2          22        102       1000
t3        1003 ac  a2          24        103       2000
t4        1004 sl  s8          28        104       1200

SQL> select * from passenger where age<20
  2  union
  3  select * from passenger where p_id in
  4  (select p_id from ticket where j_id in
  5  (select j_id from journey where train_no in
  6  (select train_no from train where src='mumbai')));
      P_ID P_NAME            AGE
---------- ---------- ----------
       101 rahul              18
       102 raj                20
       103 raju               30
       104 dhiraj             19
```

(b) Date time function.

```
SQL> select * from journey;

      J_ID J_DATE      TRAIN_NO
---------- --------- ----------
      1001 10-SEP-17       1244
      1002 14-SEP-17       1247
      1003 10-SEP-17       1249
      1004 14-SEP-17       1259

SQL> select sysdate from journey;

SYSDATE
---------
26-SEP-17
26-SEP-17
26-SEP-17
26-SEP-17

SQL> select last_day(sysdate)from journey;

LAST_DAY(
---------
30-SEP-17
30-SEP-17
30-SEP-17
30-SEP-17

SQL> select round(sysdate,'year')from journey;
```

```
ROUND(SYS
---------
01-JAN-18
01-JAN-18
01-JAN-18
01-JAN-18

SQL> select localtimestamp from journey;

LOCALTIMESTAMP
---------------------------------------------------------------------------
26-SEP-17 10.43.09.353000 PM
26-SEP-17 10.43.09.353000 PM
26-SEP-17 10.43.09.353000 PM
26-SEP-17 10.43.09.353000 PM

SQL> alter session set time_zone='+5:30';

Session altered.

SQL> select j_id,j_date+2 "newdate" from journey;

      J_ID newdate
---------- ---------
      1001 12-SEP-17
      1002 16-SEP-17
      1003 12-SEP-17
      1004 16-SEP-17

SQL>
```


(7) PL/SQL Basics

(a) declare variable.

add two number in pl/sql.

```
SQL> set serveroutput on;
SQL> declare
  2  n1 number(5);
  3  n2 number(5);
  4  nsum number(5);
  5  begin
  6  n1:=10;
  7  n2:=10;
  8  nsum:=n1+n2;
  9  dbms_output.put_line('sum of'||n1||'and'||n2||'='||nsum);
 10  end;
 11  /
sum of10and10=20

PL/SQL procedure successfully completed.
```

(b) writing excutable statement.

Add two number in pl/sql taken by user.

```
SQL> set serveroutput on;
SQL> declare
  2  n1 number(5);
  3  n2 number(5);
  4  nsum number(5);
  5  begin
  6  n1:=&number1;
  7  n2:=&number2;
  8  nsum:=n1+n2;
  9  dbms_output.put_line('sum of' ||n1||'and'||n2||'='||nsum);
 10  end;
 11  /
Enter value for number1: 12
old   6: n1:=&number1;
new   6: n1:=12;
Enter value for number2: 23
old   7: n2:=&number2;
new   7: n2:=23;
sum of12and23=35

PL/SQL procedure successfully completed.
```

(c) Interacting with the Oracle Server.

```
SQL> select * from mobile;

MID   MODEL COMPANY    DESCRIPTIO      PRICE
----- ----- ---------- ---------- ----------
m1    6+    apple      4gbRAM64GB      70000
m2    s4    samsung    3gbRAM32GB      12000
m3    3s    redmi      3gbRAM32GB      10000
m3    z10   blackberry 3gbRAM64GB      10000

SQL> declare
  2  m mobile.model%type;
  3  c mobile.company%type;
  4  p mobile.price%type;
  5  begin
  6  select model,company,price into m,c,p
  7  from mobile where price=(select max(price)from mobile);
  8  dbms_output.put_line('model no ='||m);
  9  dbms_output.put_line('company name ='||c);
 10  dbms_output.put_line('price ='||p);
 11  end;
 12  /
model no =6+
company name =apple
price =70000

PL/SQL procedure successfully completed.
```

(d). Writing Control Structures

```
SQL> select * from mobile;

MID   COMPANY    MODEL DISC            PRICE
----- ---------- ----- ---------- ----------
m101  samsung    s1    3gb32gb         12000
m102  apple      4s    2gb64gb         22000
m103  htc        526   2gb16gb         10000

SQL> set serveroutput on;
SQL> variable modelbind char(5);
SQL> declare
  2  x mobile.model%type;
  3  y mobile.company%type;
  4  z mobile.price%type;
  5  begin
  6  :modelbind:=&modelname;
  7  select model,company,price into x,y,z from mobile where
  8  model=:modelbind;
  9  dbms_output.put_line('model no :'||x);
 10  dbms_output.put_line('company :'||y);
 11  dbms_output.put_line('price :'||z);
 12  end;
 13  /
Enter value for modelname: 's1'
old   6: :modelbind:=&modelname;
new   6: :modelbind:='s1';
model no :s1
company :samsung
price :12000

PL/SQL procedure successfully completed.
```

```
SQL> declare
  2  x mobile.company%type;
  3  begin
  4  select company into x from mobile where model=&modelname;
  5  if sql%found=true then
  6  dbms_output.put_line('record found:');
  7  dbms_output.put_line('company name:'||x);
  8  else
  9  dbms_output.put_line('record not found');
 10  end if;
 11  end;
 12  /
Enter value for modelname: 's1'
old   4: select company into x from mobile where model=&modelname;
new   4: select company into x from mobile where model='s1';
record found:
company name:samsung

PL/SQL procedure successfully completed.
```

(8) .

(a) Writing Explicit Cursors

```
SQL> select * from emp;

       EID ENAME          SALARY DESG
---------- ---------- ---------- -----
       101 dhiraj          12000 manag
       102 raju            22000 it
       103 ramesh          24000 iit
       104 manish          34000 iiit

SQL> set serveroutput on;
```

```
SQL> declare
  2  a emp.eid%type;
  3  b emp.ename%type;
  4  c emp.salary%type;
  5  d emp.desg%type;
  6  cursor c_emp is
  7  select eid ,ename,salary,desg from emp;
  8  begin
  9  open c_emp;
 10  loop
 11  fetch c_emp into a,b,c,d;
 12  exit when c_emp%notfound;
 13  c:=c+5000;
 14  update emp set salary=c where eid=a;
 15  end loop;
 16  close c_emp;
 17  end;
 18  /

PL/SQL procedure successfully completed.

SQL> select * from emp;

       EID ENAME          SALARY DESG
---------- ---------- ---------- -----
       101 dhiraj          17000 manag
       102 raju            27000 it
       103 ramesh          29000 iit
       104 manish          39000 iiit
```

(b) Handling Exceptions.

```
DECLARE
   c_id customers.id%type := 8;
   c_name  customerS.No.ame%type;
   c_addr customers.address%type;
BEGIN
   SELECT  name, address INTO  c_name, c_addr
   FROM customers
   WHERE id = c_id;
   DBMS_OUTPUT.PUT_LINE ('Name: '||  c_name);
   DBMS_OUTPUT.PUT_LINE ('Address: ' || c_addr);

EXCEPTION
   WHEN no_data_found THEN
      dbms_output.put_line('No such customer!');
   WHEN others THEN
      dbms_output.put_line('Error!');
END;
/
```

```
No such customer!


PL/SQL procedure successfully completed.
```

(9).

(a) Creating Procedures and package

```
L> set serveroutput on;
L> declare
2  grade char(1);
3  begin
4  grade:='A';
5  case grade
6  when 'O' then dbms_output.put_line('Outstanding');
7  when 'A' then dbms_output.put_line('Excellent');
8  else dbms_output.put_line('Invalid grade');
9  end case;
0  end;
1  /
cellent

/SQL procedure successfully completed.
```

```
L> create table  Account (acno number(5),name char(10),balance number(5));

ble created.

L> insert into Account values(1001,'Sammer',10000);

row created.

L> commit;

mmit complete.

L> create or replace package pkg_bank as
2  procedure acc_open(a number,b char,c number);
3  procedure deposit(x number,y number);
4  procedure withdraw(x number,y number);
5  procedure dispacc(a number);
6  end pkg_bank;
7  /

ckage created.
```

```
L> create or replace package body pkg_bank as
2  procedure acc_open(a number,b char,c number)
3  as
4  begin
5  insert into Account values(a,b,c);
6  commit;
7  end;
8  procedure deposit(x number,y number)
9  as begin
0  update account set balance=balance+x where acno=y;
1  exception when no_data_found then
2  dbms_output.put_line('Account number does not exist');
3  end;
4  procedure withdraw(x number,y number)
5  as begin
6  update account set balance=balance-x where acno=y;
7  exception when no_data_found then
8  dbms_output.put_line('Account number does not exist');
9  end;
0  procedure dispacc(a number)
1  as
2  acrec account%rowtype;
3  begin
```

```
4  select * into acrec from account where acno=a;
5  dbms_output.put_line('acc no :'||acrec.acno);
6  dbms_output.put_line('name :'||acrec.name);
7  dbms_output.put_line('balance :'||acrec.balance);
8  end;
9  end;
0  /

ckage body created.

L> begin
2  pkg_bank.acc_open(1002,'Niraj',5000);
3  pkg_bank.acc_open(1003,'Suraj',6000);
4  end;
5  /

/SQL procedure successfully completed.

L> begin
2  pkg_bank.dispacc(1002);
3  pkg_bank.dispacc(1003);
4  end;
5  /
c no :1002
me :Niraj
```

```
lance :5000
c no :1003
me :Suraj
lance :6000

/SQL procedure successfully completed.

L> begin
2  pkg_bank.diposit(1000,1002);
3  end;
4  /
g_bank.diposit(1000,1002);
       *
ROR at line 2:
A-06550: line 2, column 10:
S-00302: component 'DIPOSIT' must be declared
A-06550: line 2, column 1:
/SQL: Statement ignored


L> begin
2  pkg_bank.deposit(1000,1002);
3  end;
4  /
```

```
/SQL procedure successfully completed.

L> begin
2  pkg_bank.dispacc(1002);
3  end;
4  /
c no :1002
me :Niraj
lance :6000

/SQL procedure successfully completed.
```

(b) Creating Functions

```
Select * from customers;


+----+----------+-----+-----------+----------+
| ID | NAME     | AGE | ADDRESS   | SALARY   |
+----+----------+-----+-----------+----------+
|  1 | Ramesh   |  32 | Ahmedabad |  2000.00 |
|  2 | Khilan   |  25 | Delhi     |  1500.00 |
|  3 | kaushik  |  23 | Kota      |  2000.00 |
|  4 | Chaitali |  25 | Mumbai    |  6500.00 |
|  5 | Hardik   |  27 | Bhopal    |  8500.00 |
|  6 | Komal    |  22 | MP        |  4500.00 |
+----+----------+-----+-----------+----------+
```

```
CREATE OR REPLACE FUNCTION totalCustomers
RETURN number IS
   total number(2) := 0;
BEGIN
   SELECT count(*) into total
   FROM customers;

   RETURN total;
END;
/
```

```
Function created.
```

(10) Creating Database Triggers

```
SQL> create or replace trigger tri_s
  2  after insert on student
  3  for each row
  4  begin
  5  dbms_output.put_line('trigger is called');
  6  dbms_output.put_line('new row inserted');
  7  dbms_output.put_line('kjdfiu78du');
  8  end;
  9  /

Trigger created.

SQL> insert into student values(101,'dhiraj',5000);
trigger is called
new row inserted
kjdfiu78du

1 row created.
```

```
SQL> create or replace trigger tri_s
  2  before update of salary on student
  3  for each row
  4  begin
  5  if :new.salary<:old.salary then
  6  raise_application_error(-20001,'salary can not be reduced');
  7  end if;
  8  end;
  9  /

Trigger created.

SQL> update student set salary=salary-1000 where sid=101;
update student set salary=salary-1000 where sid=101
       *
ERROR at line 1:
ORA-20001: salary can not be reduced
ORA-06512: at "SYSTEM.TRI_S", line 3
ORA-04088: error during execution of trigger 'SYSTEM.TRI_S'


SQL> update student set salary=salary+1000 where sid=101;

1 row updated.
```

```
SQL> select * from student;

       SID SNAME          SALARY
---------- ---------- ----------
       101 dhiraj           6000
       102 raj              4000

SQL> update student set salary=5000 where sid=101;
update student set salary=5000 where sid=101
       *
ERROR at line 1:
ORA-20001: salary can not be reduced
ORA-06512: at "SYSTEM.TRI_S", line 3
ORA-04088: error during execution of trigger 'SYSTEM.TRI_S'


SQL> update student set salary=5000 where sid=102;

1 row updated.

SQL> select * from student;

       SID SNAME          SALARY
---------- ---------- ----------
       101 dhiraj           6000
       102 raj              5000
```