DATABASE LAB

LAB CYCLE II

Q.SET 1

Create the following tables and execute the queries given below

SAILORS

sid	sname	rating	age
22	Dustin	7	45
29	Brutas	1	33
31	Lubber	8	55
32	Andy	8	25
58	Rusty	10	35
64	Horatio	7	35
71	Zorba	10	16
74	Horatio	9	35
85	Art	3	26
95	Bob	3	64

BOATS

Bid	bname	color
101	Interlake	Blue
102	Interlake	Red
103	Clipper	Green
104	Marine	Red

RESERVES

sid	bid	day
22	101	10/10/98
22	102	10/10/98
22	103	10/8/98
22	104	10/7/98
31	102	11/10/98

31	103	11/6/98
31	104	11/12/98
64	101	9/5/98
64	102	9/8/98
74	103	9/8/98

Table creation query

```
/*-------SAILORS----------------*/
SQL> CREATE TABLE SAILORS
  2    (
  3      sid    NUMBER(2),
  4      sname  VARCHAR(10),
  5      rating NUMBER(2),
  6      age    NUMBER(2)
  7    );

Table created.

SQL> INSERT INTO SAILORS VALUES
  2    (22, 'Dustin', 7, 45
  3    );

1 row created.

SQL> INSERT INTO SAILORS VALUES
  2    (29, 'Brutas', 1, 33
  3    );

1 row created.

SQL> INSERT INTO SAILORS VALUES
  2    (31, 'Lubber', 8, 55
  3    );

1 row created.

SQL> INSERT INTO SAILORS VALUES
  2    (32, 'Andy', 8, 25
  3    );

1 row created.

SQL> INSERT INTO SAILORS VALUES
  2    (58, 'Rusty', 10, 35
  3    );

1 row created.

SQL> INSERT INTO SAILORS VALUES
  2    (64, 'Horatio', 7, 35
  3    );

1 row created.

SQL> INSERT INTO SAILORS VALUES
  2    (71, 'Zorba', 10, 16
  3    );

1 row created.

SQL> INSERT INTO SAILORS VALUES
  2    (74, 'Horatio', 9, 35
  3    );

1 row created.

SQL>   INSERT INTO SAILORS VALUES
  2    (85, 'Art', 3, 26
  3    );

1 row created.

SQL> INSERT INTO SAILORS VALUES
  2    (95, 'Bob', 3, 64
  3    );

1 row created.
```

```
SQL> SELECT * FROM SAILORS;

       SID SNAME          RATING        AGE
---------- ---------- ---------- ----------
        22 Dustin              7         45
        29 Brutas              1         33
        31 Lubber              8         55
        32 Andy                8         25
        58 Rusty              10         35
        64 Horatio             7         35
        71 Zorba              10         16
        74 Horatio             9         35
        85 Art                 3         26
        95 Bob                 3         64

10 rows selected.


/*--------------------BOATS------------------------*/
SQL> CREATE TABLE BOATS
  2    ( Bid NUMBER(3), bname VARCHAR(10), color
VARCHAR(7)
  3    );

Table created.

SQL> INSERT INTO BOATS VALUES
  2    (101, 'Interlake', 'Blue'
  3    );

1 row created.

SQL> INSERT INTO BOATS VALUES
  2    (102, 'Interlake', 'Red'
  3    );

1 row created.

SQL> INSERT INTO BOATS VALUES
  2    (103, 'Clipper', 'Green'
  3    );

1 row created.

SQL> INSERT INTO BOATS VALUES
  2    (104, 'Marine', 'Red'
  3    );

1 row created.

SQL> SELECT * FROM BOATS;

       BID BNAME      COLOR
---------- ---------- -------
       101 Interlake  Blue
       102 Interlake  Red
       103 Clipper    Green
       104 Marine     Red
```

```
/*------------------reserves--------------------*/

SQL> CREATE TABLE RESERVES
  2    (sid NUMBER(2), bid NUMBER(3), DAY DATE
  3    );

Table created.

SQL> INSERT INTO RESERVES VALUES
  2    (22, 101, to_date('101098', 'ddmmyy')
  3    );

1 row created.

SQL> INSERT INTO RESERVES VALUES
  2    (22, 102, to_date('101098', 'ddmmyy')
  3    );

1 row created.
```

```
SQL> INSERT INTO RESERVES VALUES
  2    (22, 103, to_date('100898', 'ddmmyy')
  3    );

1 row created.

SQL> INSERT INTO RESERVES VALUES
  2    (22, 104, to_date('100798', 'ddmmyy')
  3    );

1 row created.

SQL> INSERT INTO RESERVES VALUES
  2    (31, 102, to_date('111098', 'ddmmyy')
  3    );

1 row created.

SQL> INSERT INTO RESERVES VALUES
  2    (31, 103, to_date('110698', 'ddmmyy')
  3    );

1 row created.

SQL> INSERT INTO RESERVES VALUES
  2    (31, 104, to_date('111298', 'ddmmyy')
  3    );

1 row created.

SQL> INSERT INTO RESERVES VALUES
  2    (64, 101, to_date('090598', 'ddmmyy')
  3    );

1 row created.

SQL> INSERT INTO RESERVES VALUES
  2    (64, 102, to_date('090898', 'ddmmyy')
  3    );
```

```
1 row created.

SQL> INSERT INTO RESERVES VALUES
  2    (74, 103, to_date('090898', 'ddmmyy')
  3    );

1 row created.

SQL> SELECT * FROM RESERVES;

       SID        BID   DAY
---------- ----------   ---------
        22        101   10-OCT-98
        22        102   10-OCT-98
        22        103   10-AUG-98
        22        104   10-JUL-98
        31        102   11-OCT-98
        31        103   11-JUN-98
        31        104   11-DEC-98
        64        101   09-MAY-98
        64        102   09-AUG-98
        74        103   09-AUG-98

10 rows selected.
```

Questions:

1. Find the names and ages of all sailors

```
SQL> SELECT sname, age FROM sailors;

SNAME             AGE
---------- ----------
Dustin             45
Brutas             33
Lubber             55
Andy               25
Rusty              35
Horatio            35
Zorba              16
Horatio            35
Art                26
Bob                64

10 rows selected.
```

2. Find all information of sailors who have reserved boat number 101.

```
SQL> SELECT *
2  FROM SAILORS,
3    RESERVES
4  WHERE sailors.sid = reserves.sid
5  AND bid           = 101;

SID SNAME          RATING        AGE        SID
  BID DAY
---------- ---------- ---------- ----------
  ---------- ---------- ---------
22 Dustin              7         45         22
  101 10-OCT-98
64 Horatio             7         35         64
  101 09-MAY-98
```

3. Find all sailors with rating above 7

```
SQL> SELECT sname FROM SAILORS WHERE rating >
7;

SNAME
----------
Lubber
Andy
Rusty
Zorba
Horatio
```

4. Find the names of sailors who have reserved boat no 103

```
 SQL> SELECT sname
   2  FROM SAILORS,
   3    RESERVES,
   4    BOATS
   5  WHERE boats.bid = 103
   6  AND boats.bid   = reserves.bid
   7  AND sailors.sid = reserves.sid;

SNAME
----------
Dustin
Lubber
Horatio
```

5. Find the names of sailors who have reserved a red boat, and list in the order of age.

```
SQL> SELECT DISTINCT sname
  2  FROM SAILORS,
  3    RESERVES,
  4    BOATS
  5  WHERE color     = 'Red'
  6  AND boats.bid   = reserves.bid
  7  AND sailors.sid = reserves.sid;
```

```
SNAME
----------
Lubber
Dustin
Horatio
```

6. Find the names of sailors who have reserved either a red or green boat

```
SQL> SELECT DISTINCT sname
  2  FROM SAILORS,
  3    RESERVES,
  4    BOATS
  5  WHERE ( color   = 'Red'
  6  OR color        = 'Green')
  7  AND boats.bid   = reserves.bid
  8  AND sailors.sid = reserves.sid;

SNAME
----------
Lubber
Dustin
Horatio
```

7. Find the colors of boats reserved by “Lubber”.

```
SQL> SELECT DISTINCT color
  2  FROM SAILORS,
  3    RESERVES,
  4    BOATS
  5  WHERE sname     = 'Lubber'
  6  AND boats.bid   = reserves.bid
  7  AND sailors.sid = reserves.sid;

COLOR
-------
Green
Red
```

8. Find the names of sailors who have reserved both red and green boats

```
SQL> SELECT DISTINCT sname
```

```
  2  FROM SAILORS,
  3    RESERVES,
  4    BOATS
  5  WHERE color      = 'Red'
  6  AND boats.bid    = reserves.bid
  7  AND sailors.sid = reserves.sid
  8  INTERSECT
  9  SELECT DISTINCT sname
 10  FROM SAILORS,
 11    RESERVES,
 12    BOATS
 13  WHERE color      = 'Green'
 14  AND boats.bid    = reserves.bid
 15  AND sailors.sid = reserves.sid;

SNAME
----------
Dustin
Horatio
Lubber
```

9. Find the names of sailors who have reserved at least one boat

```
SQL> SELECT DISTINCT sname
  2  FROM SAILORS,
  3    RESERVES
  4  WHERE reserves.sid = sailors.sid;

SNAME
----------
Lubber
Dustin
Horatio
```

10. Find the ids and names of sailors who have reserved two different boats on the same day.

```
SQL> SELECT sname,
  2    reserves.sid
  3  FROM RESERVES,
  4    SAILORS
  5  WHERE sailors.sid = reserves.sid
  6  GROUP BY DAY,
  7    reserves.sid,
  8    sname
  9  HAVING COUNT(DAY)>1 ;

SNAME             SID
---------- ----------
Dustin             22
```

11. Find the name and the age of the youngest sailor.

```
SQL> SELECT sname,
  2    age
  3  FROM SAILORS
  4  WHERE age =
  5    (SELECT MIN(age) FROM sailors
  6    ) ;

SNAME             AGE
---------- ----------
Zorba              16
```

12. Find the names and ratings of sailor whose rating is better than some sailor called Horatio.

```
SQL> SELECT sname
  2  FROM SAILORS
  3  WHERE rating >
  4    (SELECT MAX(rating) FROM SAILORS WHERE
sname =
  5    'Horatio');

SNAME
----------
Rusty
Zorba
```

13. Find the names of sailors who have reserved all boats.

```
SQL> SELECT sname,
  2    sid
  3  FROM
  4    (SELECT sname,
  5      reserves.sid,
  6      COUNT(bid) AS id
  7    FROM RESERVES,
  8      SAILORS
  9    WHERE reserves.sid = sailors.sid
 10    GROUP BY reserves.SID,
 11      sname
 12    ) a
 13  WHERE id =
 14    ( SELECT COUNT(bid) FROM BOATS
 15    );

SNAME             SID
---------- ----------
Dustin             22
```

14.Count the number of different sailor names.

```
SQL> SELECT COUNT(c.sname)
  2  FROM
  3    (SELECT DISTINCT sname FROM Sailors
  4    ) c;

COUNT(C.SNAME)
--------------
             9
```

15.Calculate the average age of all sailors.

```
SQL> SELECT AVG(age) FROM SAILORS;

  AVG(AGE)
----------
      36.9
```

16.Find the average age of sailors for each rating level.

```
SQL> SELECT rating,AVG(age) FROM SAILORS GROUP
BY
  2  rating;

    RATING   AVG(AGE)
---------- ----------
         1         33
         8         40
         7         40
         3         45
        10       25.5
         9         35

6 rows selected.
```

17. Find the average age of sailors for each rating level that has at least two sailors.

```
SQL> SELECT a.rating,
  2    b.mean
  3  FROM
  4    (SELECT COUNT(sname) AS num,
  5      rating
  6    FROM SAILORS
  7    GROUP BY rating
  8    HAVING COUNT(sname)>1
  9    ) a,
 10   (SELECT rating,AVG(age) AS mean FROM
   SAILORS
 11      GROUP BY rating ) b
 12  WHERE a.rating = b.rating;

    RATING       MEAN
---------- ----------
         8         40
         7         40
         3         45
        10       25.5
```

Q.SET 2

1. Create the table STUDENT_INFO with Columns: Sid, Stud_name & stude_score.

```
SQL>    CREATE TABLE STUDENT_INFO
  2    (
  3      Sid         INT UNIQUE ,
  4      Stud_name   VARCHAR(20) NOT NULL,
  5      stude_score NUMBER(5,2) DEFAULT 20
  6    );

Table created.

SQL> ALTER TABLE STUDENT_INFO
  2  DROP UNIQUE(sid);

Table altered.

SQL> ALTER TABLE STUDENT_INFO MODIFY sid
  2  PRIMARY KEY;

Table altered.

SQL> UPDATE STUDENT_INFO SET stude_score =
  2  stude_score + 5 WHERE stude_score > 150;

0 rows updated.
```

2. Create the tables **worker** and **bonus** with the following fields. The primary key of Worker table is Worker_ID. Set Worker_id as foreign key of bonus on update and delete cascade constraints. Each constraint should be given a name

WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
1	Monika	Arora	100000	2014-02-20	HR

2	Niharika	Verma	80000	2014-06-11	Admin
3	Vishal	Singhal	300000	2014-02-20	HR
4	Amitabh	Singh	500000	2014-02-20	Admin
5	Vivek	Bhati	500000	2014-06-11	Admin
6	Vipul	Diwan	200000	2014-06-11	Account
7	Satish	Kumar	75000	2014-01-20	Account
8	Geetika	Chauhan	90000	2014-04-11	Admin

```
SQL> CREATE TABLE worker
  2    (
  3      Worker_ID    INT,
  4      first_name   VARCHAR(15),
  5      last_name    VARCHAR(15),
  6      salary       NUMBER(8),
  7      joining_date DATE,
  8      department   VARCHAR (15)
  9    );

Table created.

SQL> CREATE TABLE bonus
  2    ( Worker_ID INT, bonus_date DATE,
  3     bonus_amount NUMBER(6) );

Table created.

SQL> ALTER TABLE worker ADD CONSTRAINTS pk_c
  2    PRIMARY KEY(Worker_ID);

Table altered.

SQL> ALTER TABLE bonus ADD CONSTRAINT fk_cod_csd
  2    FOREIGN KEY(Worker_ID) REFERENCES
  3    worker(Worker_ID) ON DELETE CASCADE ;
```

```
Table altered.

SQL> INSERT
  2  INTO worker VALUES
  3    (
  4      1,
  5      '  Monika  ',
  6      ' Arora ',
  7      100000,
  8      to_date( '20140220', 'yyyymmdd' ),
  9      '   HR    '
 10    );

1 row created.

SQL> INSERT
  2  INTO worker VALUES
  3    (
  4      2,
  5      ' Niharika ',
  6      ' Verma ',
  7      80000,
  8      to_date( '20140611', 'yyyymmdd' ),
  9      '  Admin  '
 10    );

1 row created.

SQL> INSERT
  2  INTO worker VALUES
  3    (
  4      3,
  5      '  Vishal  ',
  6      'Singhal',
  7      300000,
  8      to_date( '20140220', 'yyyymmdd' ),
  9      '   HR    '
 10    );
```

```
1 row created.

SQL> INSERT
  2  INTO worker VALUES
  3    (
  4      4,
  5      ' Amitabh  ',
  6      ' Singh ',
  7      500000,
  8      to_date( '20140220', 'yyyymmdd' ),
  9      '  Admin  '
 10    );

1 row created.

SQL> INSERT
  2  INTO worker VALUES
  3    (
  4      5,
  5      '  Vivek   ',
  6      ' Bhati ',
  7      500000,
  8      to_date( '20140611', 'yyyymmdd' ),
  9      '  Admin  '
 10    );

1 row created.

SQL> INSERT
  2  INTO worker VALUES
  3    (
  4      6,
  5      '  Vipul   ',
  6      ' Diwan ',
  7      200000,
  8      to_date( '20140611', 'yyyymmdd' ),
  9      ' Account '
 10    );

1 row created.
```

```
SQL> INSERT
  2  INTO worker VALUES
  3    (
  4      7,
  5      '  Satish  ',
  6      ' Kumar ',
  7      75000,
  8      to_date( '20140120', 'yyyymmdd' ),
  9      ' Account '
 10    );

1 row created.

SQL> INSERT
  2  INTO worker VALUES
  3    (
  4      8,
  5      ' Geetika  ',
  6      'Chauhan',
  7      90000,
  8      to_date( '20140411', 'yyyymmdd' ),
  9      '  Admin  '
 10    );

1 row created.

SQL> SELECT * FROM worker;


 WORKER_ID FIRST_NAME      LAST_NAME
SALARY JOINING_D DEPARTMENT
---------- --------------- ---------------
---------- --------- ---------------
         1   Monika         Arora
100000 20-FEB-14    HR
         2  Niharika        Verma
80000 11-JUN-14   Admin
         3   Vishal        Singhal
300000 20-FEB-14    HR
```

```
         4  Amitabh          Singh
500000 20-FEB-14   Admin
         5   Vivek           Bhati
500000 11-JUN-14   Admin
         6   Vipul           Diwan
200000 11-JUN-14  Account
         7   Satish          Kumar
75000 20-JAN-14  Account
         8  Geetika         Chauhan
90000 11-APR-14   Admin

8 rows selected.
```

3. Sample Table – Bonus

WORKER_ID	BONUS_DATE	BONUS_AMOUNT
1	2016-02-20	5000
2	2016-06-11	3000
3	2016-02-20	4000
1	2016-02-20	4500
2	2016-06-11	3500

```
SQL> INSERT
  2  INTO bonus VALUES
  3    (
  4      1,
  5      to_date( '20160220', 'yyyymmdd' ),
  6      5000
  7    );

1 row created.


SQL> INSERT INTO bonus VALUES
```

```
      2    ( 2, to_date( '20160611', 'yyyymmdd' ),
3000
      3    );

    1 row created.

    SQL> INSERT INTO bonus VALUES
      2    ( 3, to_date( '20160220', 'yyyymmdd' ),
4000
      3    );

    1 row created.

    SQL> INSERT INTO bonus VALUES
      2    ( 1, to_date( '20160220', 'yyyymmdd' ),
4500
      3    );

    1 row created.

    SQL> INSERT INTO bonus VALUES
      2    ( 2, to_date( '20160611', 'yyyymmdd' ),
3500
      3    );

    1 row created.

    SQL> SELECT * FROM bonus;

     WORKER_ID BONUS_DAT BONUS_AMOUNT
    ---------- --------- ------------
             1 20-FEB-16         5000
             2 11-JUN-16         3000
             3 20-FEB-16         4000
             1 20-FEB-16         4500
             2 11-JUN-16         3500
```

4. Write An SQL Query To Fetch “FIRST_NAME” From Worker Table Using The Alias Name As <WORKER_NAME>.

```
SQL> SELECT first_name AS worker_name FROM
worker;

WORKER_NAME
---------------
  Monika
 Niharika
  Vishal
 Amitabh
  Vivek
  Vipul
  Satish
 Geetika

8 rows selected.
```

5. Write An SQL Query To Print All Worker Details From The Worker Table Order By FIRST_NAME Ascending

```
SQL> SELECT * FROM worker ORDER BY
trim(first_name) ASC ;

 WORKER_ID FIRST_NAME      LAST_NAME
SALARY JOINING_D DEPARTMENT
---------- --------------- ---------------
---------- --------- ---------------
         4  Amitabh         Singh
500000 20-FEB-14   Admin
         8  Geetika        Chauhan
90000 11-APR-14   Admin
         1   Monika         Arora
100000 20-FEB-14    HR
         2  Niharika        Verma
80000 11-JUN-14   Admin
         7   Satish         Kumar
75000 20-JAN-14  Account
         6   Vipul          Diwan
200000 11-JUN-14  Account
         3   Vishal        Singhal
300000 20-FEB-14    HR
```

```
         5   Vivek          Bhati
500000 11-JUN-14   Admin

8 rows selected.
```

6. Write An SQL Query To Print Details Of Workers Excluding First Names, “Vipul” And “Satish” From Worker Table.

```
SQL> SELECT *
  2  FROM worker
  3  WHERE trim(first_name) != 'Vipul'
  4  AND trim(first_name)   != 'Satish';

 WORKER_ID FIRST_NAME      LAST_NAME
SALARY JOINING_D DEPARTMENT
---------- --------------- ---------------
---------- --------- ---------------
         1   Monika         Arora
100000 20-FEB-14    HR
         2  Niharika        Verma
80000 11-JUN-14   Admin
         3   Vishal        Singhal
300000 20-FEB-14    HR
         4  Amitabh         Singh
500000 20-FEB-14   Admin
         5   Vivek          Bhati
500000 11-JUN-14   Admin
         8  Geetika        Chauhan
90000 11-APR-14   Admin

6 rows selected.
```

7. Write An SQL Query To Print Details Of Workers With DEPARTMENT Name As “Admin”.

```
WORKER_ID FIRST_NAME      LAST_NAME
SALARY JOINING_D DEPARTMENT
---------- --------------- ---------------
---------- --------- ---------------
```

```
         2  Niharika        Verma
80000 11-JUN-14   Admin
         4  Amitabh         Singh
500000 20-FEB-14   Admin
         5   Vivek          Bhati
500000 11-JUN-14   Admin
         8  Geetika        Chauhan
90000 11-APR-14   Admin
```

8. Write An SQL Query To Print Details Of The Workers Whose SALARY Lies Between 100000 And 500000

```
SQL> SELECT * FROM worker WHERE salary BETWEEN
100000 AND 500000;

 WORKER_ID FIRST_NAME      LAST_NAME
SALARY JOINING_D DEPARTMENT
---------- --------------- ---------------
---------- --------- ---------------
         1   Monika         Arora
100000 20-FEB-14    HR
         3   Vishal        Singhal
300000 20-FEB-14    HR
         4  Amitabh         Singh
500000 20-FEB-14   Admin
         5   Vivek          Bhati
500000 11-JUN-14   Admin
         6   Vipul          Diwan
200000 11-JUN-14  Account
```

9. Write An SQL Query To Fetch “FIRST_NAME” From Worker Table In Upper Case. (upper())

```
SQL> SELECT upper(first_name) FROM worker;

UPPER(FIRST_NAM
---------------
  MONIKA
 NIHARIKA
  VISHAL
 AMITABH
```

```
  VIVEK
  VIPUL
  SATISH
 GEETIKA

8 rows selected.
```

10. Write An SQL Query To Fetch Unique Values Of DEPARTMENT From Worker Table.

```
SQL> SELECT UNIQUE department FROM worker;

DEPARTMENT
---------------
   HR
  Admin
 Account
```

11. Write An SQL Query To Print First Three Characters Of FIRST_NAME From Worker Table.(substring())

```
SQL> SELECT SUBSTR(first_name, 1, 3) AS small
FROM worker;

SMALL
------------
  M
 Ni
  V
 Am
  V
  V
  S
 Ge

8 rows selected.
```

12. Write An SQL Query To Print The FIRST_NAME From Worker Table After Removing White Spaces From The Right Side(RTRIM ( ))

```
SQL> SELECT rtrim(first_name) FROM worker;

RTRIM(FIRST_NAM
---------------
  Monika
 Niharika
  Vishal
 Amitabh
  Vivek
  Vipul
  Satish
 Geetika

8 rows selected.
```

13. Write An SQL Query To Print The DEPARTMENT From Worker Table After Removing White Spaces From The Left Side. (LTRIM ( ))

```
SQL> SELECT ltrim(department) FROM worker;

LTRIM(DEPARTMEN
---------------
HR
Admin
HR
Admin
Admin
Account
Account
Admin

8 rows selected.
```

14. Write An SQL Query That Fetches The Unique Values Of DEPARTMENT From Worker Table And Prints Its Length.(length())

```
SQL> SELECT UNIQUE department,
LENGTH(department) FROM worker ;

DEPARTMENT      LENGTH(DEPARTMENT)
```

```
--------------- ------------------
 Account                         9
   HR                            9
  Admin                          9
```

15. Write An SQL Query To Print The FIRST_NAME From Worker Table After Replacing ‘a’ With ‘A’.(REPLACE( ))

```
SQL> SELECT REPLACE(first_name, 'a', 'A') AS
fname FROM worker;

FNAME
---------------
  MonikA
 NihArikA
  VishAl
 AmitAbh
  Vivek
  Vipul
  SAtish
 GeetikA

8 rows selected.
```

16. Find the First name , last name ,Department, Salary and Bonus of employees whose bonus amount is greater than 4000

```
SQL> SELECT first_name,
  2    last_name,
  3    department,
  4    salary,
  5    bonus_amount
  6  FROM worker,
  7    bonus
  8  WHERE worker.worker_id = bonus.worker_id
  9  AND bonus_amount       > 4000 ;

FIRST_NAME      LAST_NAME       DEPARTMENT
SALARY BONUS_AMOUNT
```

```
--------------- --------------- ---------------
---------- ------------
  Monika          Arora             HR
100000        5000
  Monika          Arora             HR
100000        4500
```

17. Delete the employee with worker_id=7 from worker and display the details of both tables. ~~(SN: actually useless question bcz 7 is not in BONUS table)~~ .

```
SQL> DELETE worker WHERE worker_id=7;

1 row deleted.

SQL> SELECT * FROM worker;

 WORKER_ID FIRST_NAME      LAST_NAME
SALARY JOINING_D DEPARTMENT
---------- --------------- ---------------
---------- --------- ---------------
         1   Monika          Arora
100000 20-FEB-14    HR
         2  Niharika         Verma
80000 11-JUN-14   Admin
         3   Vishal         Singhal
300000 20-FEB-14    HR
         4  Amitabh          Singh
500000 20-FEB-14   Admin
         5   Vivek           Bhati
500000 11-JUN-14   Admin
         6   Vipul           Diwan
200000 11-JUN-14  Account
         8  Geetika         Chauhan
90000 11-APR-14   Admin

7 rows selected.

SQL> SELECT * FROM bonus;
```

```
 WORKER_ID BONUS_DAT BONUS_AMOUNT
---------- --------- ------------
         1 20-FEB-16         5000
         2 11-JUN-16         3000
         3 20-FEB-16         4000
         1 20-FEB-16         4500
         2 11-JUN-16         3500
```

18. Drop the foreign key constraint and add a new referential integrity constraint with ‘on update or delete with no action’ ~~(SN: no action, set default not working)~~

```
SQL> ALTER TABLE bonus
  2  DROP CONSTRAINT fk_cod_csd;

Table altered.

SQL> ALTER TABLE bonus
  2   ADD CONSTRAINT fk_cod_na
  3   FOREIGN KEY(Worker_ID) REFERENCES
  4   worker(Worker_ID) ON DELETE no action ;

DELETE no action
       *
ERROR at line 2:
ORA-00905: missing keyword
```

19. Delete the employee with worker_id = 8 from worker.

```
SQL> DELETE worker WHERE worker_id=8;

1 row deleted.

SQL> SELECT * FROM worker;

 WORKER_ID FIRST_NAME      LAST_NAME
SALARY JOINING_D DEPARTMENT
---------- --------------- ---------------
---------- --------- ---------------
```

```
         1   Monika         Arora
100000 20-FEB-14    HR
         2  Niharika        Verma
80000 11-JUN-14   Admin
         3   Vishal        Singhal
300000 20-FEB-14    HR
         4  Amitabh         Singh
500000 20-FEB-14   Admin
         5   Vivek          Bhati
500000 11-JUN-14   Admin
         6   Vipul          Diwan
200000 11-JUN-14  Account

6 rows selected.
```

Q.SET 3

Create the tables given below and execute the queries:

Customer(Cust id : integer, cust_name: string)

Item(item_id: integer, item_name: string, price: integer)

Sale(bill_no: integer, bill_date: date, cust_id: integer, item_id: integer, qty_sold: integer)

For the above schema, perform the following—

a) Create the tables with the appropriate integrity constraints

```
SQL> CREATE TABLE customer
  2    (
  3      cust_id   INT PRIMARY KEY,
  4      cust_name VARCHAR(20) NOT NULL
  5    );

Table created.

SQL> CREATE TABLE item
  2    (item_id INT PRIMARY KEY, item_name
VARCHAR(25), price INT
  3    );

Table created.

SQL> CREATE TABLE sale
  2    (
  3      bill_no   INT PRIMARY KEY,
  4      bill_date DATE NOT NULL,
  5      cust_id   INT NOT NULL,
  6      item_id   INT NOT NULL,
  7      qty_sold  NUMERIC(6,3) NOT NULL,
  8      FOREIGN KEY(item_id)
  9         REFERENCES item(item_id),
 10      FOREIGN KEY (cust_id)
 11         REFERENCES customer(cust_id)
 12    );

Table created.
```

b) Insert details of 5 customers, 5 items and 10 sales details. There should be one customer ‘rekha’ who had purchased 3 different products on the same date. And there should be atleast one customer who had purchased 2 different products on the same date in the year ‘2018’.

```
SQL> INSERT INTO customer VALUES
  2    ( 1, 'Rekha'
  3    );

1 row created.

SQL> INSERT INTO customer VALUES
  2    ( 2, 'Rani'
  3    );

1 row created.

SQL> INSERT INTO customer VALUES
  2    ( 3, 'Rocky'
  3    );

1 row created.

SQL> INSERT INTO customer VALUES
  2    ( 4, 'Xin'
  3    );

1 row created.

SQL> INSERT INTO customer VALUES
  2    ( 5, 'Kim'
  3    );

1 row created.

SQL> INSERT INTO item VALUES
  2    ( 1, 'Rusk', 120
  3    );

1 row created.

SQL> INSERT INTO item VALUES
  2    ( 2, 'Banana', 50
  3    );

1 row created.

SQL> INSERT INTO item VALUES
  2    ( 3, 'Sanitizer', 60
  3    );

1 row created.

SQL> INSERT INTO item VALUES
  2    ( 4, 'Cake', 420
  3    );

1 row created.

SQL> INSERT INTO item VALUES
  2    ( 5, 'Laddu', 25
  3    );

1 row created.

SQL> INSERT INTO sale VALUES
  2    ( 10, to_date('20201001','yyyymmdd'), 1, 1,
3
  3    );

1 row created.
```

```
SQL> INSERT INTO sale VALUES
  2    ( 11, to_date('20201001','yyyymmdd'), 1, 3,
2
  3    );

1 row created.

SQL> INSERT INTO sale VALUES
  2    ( 12, to_date('20201001','yyyymmdd'), 1, 5,
7
  3    );

1 row created.

SQL> INSERT INTO sale VALUES
  2    ( 13, to_date('20181001','yyyymmdd'), 4, 4,
1
  3    );

1 row created.

SQL> INSERT INTO sale VALUES
  2    ( 14, to_date('20181011','yyyymmdd'), 4, 2,
2
  3    );

1 row created.

SQL> INSERT INTO sale VALUES
  2    ( 15, to_date('20180929','yyyymmdd'), 5, 1,
5
  3    );

1 row created.

SQL> INSERT INTO sale VALUES
  2    ( 16, to_date('20191225','yyyymmdd'), 3, 1,
5
  3    );
```

```
1 row created.

SQL> INSERT INTO sale VALUES
  2     ( 17, to_date('19950621','yyyymmdd'), 5, 4,
4
  3    );

1 row created.

SQL> INSERT INTO sale VALUES
  2     ( 18, to_date('20020401','yyyymmdd'), 4, 5,
5
  3    );

1 row created.

SQL> INSERT INTO sale VALUES
  2     ( 19, to_date('20200212','yyyymmdd'), 1, 2,
1
  3    );

1 row created.

SQL> SELECT * FROM sale;

     BILL_NO  BILL_DATE       CUST_ID       ITEM_ID
QTY_SOLD
----------   ---------    ----------    ----------
----------
          10  01-OCT-20             1             1
3
          11  01-OCT-20             1             3
2
          12  01-OCT-20             1             5
7
          13  01-OCT-18             4             4
1
          14  11-OCT-18             4             2
2
```

```
           15  29-SEP-18                     5             1
5
           16  25-DEC-19                     3             1
5
           17  21-JUN-95                     5             4
4
           18  01-APR-02                     4             5
5
           19  12-FEB-20                     1             2
1

10 rows selected.
```

c) List the details of the customer who have bought a product which has a price>200

```
SQL> SELECT *
  2  FROM customer,
  3    item,
  4    sale
  5  WHERE price      > 200
  6  AND sale.item_id = item.item_id
  7  AND sale.cust_id = customer.cust_id;

    CUST_ID  CUST_NAME                      ITEM_ID
ITEM_NAME                     PRICE
----------     --------------------     ----------
------------------------- ----------
     BILL_NO  BILL_DATE       CUST_ID       ITEM_ID
QTY_SOLD
----------   ---------    ----------    ----------
----------
          4 Xin                                4 Cake
420
           13  01-OCT-18                     4             4
1

          5 Kim                                4 Cake
420
```

```
           17  21-JUN-95             5             4
4
```

d) Give a count of how many products have been bought by each customer group by bill date.

```
SQL> SELECT cust_name,
       prod_nums,
       bill_date
     FROM customer,
       (SELECT cust_id,
         COUNT(item_id) AS prod_nums,
         bill_date
       FROM sale
       GROUP BY (bill_date, cust_id )
       ) ci
     WHERE ci.cust_id = customer.cust_id;

CUST_NAME              PROD_NUMS BILL_DATE
-------------------- ---------- ---------
Xin                           1 01-OCT-18
Rocky                         1 25-DEC-19
Rekha                         1 12-FEB-20
Rekha                         3 01-OCT-20
Xin                           1 11-OCT-18
Kim                           1 29-SEP-18
Kim                           1 21-JUN-95
Xin                           1 01-APR-02

8 rows selected.
```

e) Give a count of how many products have been bought by each customer group by bill date only for the year 2018.

```
SQL> SELECT cust_name,
       COUNT(item_id),
       bill_date
     FROM customer c,
       sale s
```

```
  6  WHERE c.cust_id = s.cust_id
  7  GROUP BY cust_name,
  8    bill_date
  9  HAVING extract(YEAR FROM bill_date) = 2018;

CUST_NAME            COUNT(ITEM_ID) BILL_DATE
-------------------- -------------- ---------
Xin                               1 11-OCT-18
Xin                               1 01-OCT-18
Kim                               1 29-SEP-18
```

f) Give a list of products bought by a customer having cust_id as 5

```
SQL> SELECT item_name
  2  FROM item,
  3    sale
  4  WHERE sale.item_id = item.item_id
  5  AND sale.cust_id   = 5;

ITEM_NAME
-------------------------
Rusk
Cake
```

g) List the item details which are sold as of today

```
SQL> SELECT item_name,
  2    price,
  3    qty_sold
  4  FROM item ,
  5    sale,
  6    (SELECT TO_date(sysdate) AS Today FROM dual
  7    ) tod
  8  WHERE sale.item_id = item.item_id
  9  AND bill_date      = today ;

no rows selected
```

h) Print the bill in a neat format with the quantity sold, price of the item and the final amount of customer ‘rekha’

```
SQL> SELECT item_name,
  2    qty_sold,
  3    price,
  4    (qty_sold * price) AS total_amount
  5  FROM customer,
  6    item,
  7    sale
  8  WHERE cust_name  = 'Rekha'
  9  AND sale.item_id = item.item_id
 10  AND sale.cust_id = customer.cust_id;

ITEM_NAME                      QTY_SOLD       PRICE
TOTAL_AMOUNT
-------------------------   ----------   ----------
------------
Rusk                                  3         120
360
Sanitizer                             2          60
120
Laddu                                 7          25
175
Banana                                1          50
50
```