```
 --JOURNAL COMPILATION
 /*
     AIM:To execute different aggregate functions and demonstrate “group by” clause in a
         multi-relation environment.

     PROBLEM STATEMENT:Use the SalesCo database established in Experiment-02 with the
                       below mentioned schemata to execute the listed queries using
                       aggregate functions and group by clause.
                CUSTOMER (C_CODE, LNAME, FNAME, C_AREA, C_PHONE, BALANCE)
                INVOICE (INV_NUM, C_CODE, INV_DATE)
                LINE (INV_NUM, L_NUM, P_CODE, L_UNITS, L_PRICE)
                PRODUCT (P_CODE, DESCRIPT, P_DATE, QTY, P_MIN, P_PRICE, P_DISC, V_CODE)
                VENDOR (V_CODE, V_NAME, V_CONTACT, V_AREA, V_PHONE, V_STATE, V_ORDER)
*/


 ---------------------------------- QUERY-01 ------------------------------
 /*
 Write a SQL code to create a table PART without any tuple from PRODUCT such that it
includes product
 code—PT_CODE, product description—PT_DESC, the unit price—PT_PRICE and the supplier
code. Now populate
 PART with the tuples fetching the contents from PRODUCT.For the PART table created,
compare its schema
 with PRODUCT for the common attributes. Observe all the constraints on PART table (use
USER_CONSTRAINTS)
 and state your inferences.
 */
 ------------------------------------------------------------------------------
SQL> CREATE TABLE PART AS SELECT P_CODE AS PT_CODE,DESCRIPT AS PT_DESC,P_PRICE AS
PT_PRICE,V_CODE FROM PRODUCT WHERE 1=2;

Table created.

SQL> INSERT INTO PART SELECT P_CODE,DESCRIPT,P_PRICE,V_CODE FROM PRODUCT;

16 rows created.

SQL> SELECT CONSTRAINT_NAME,TABLE_NAME FROM USER_CONSTRAINTS WHERE TABLE_NAME LIKE
'PART';

CONSTRAINT_NAME      TABLE_NAME
---------------      ------------


SYS_C0012329         PART



SYS_C0012328         PART



SYS_C0012327         PART



3 rows selected.

 ---------------------------------- QUERY-02 ------------------------------
```

```
 /*
 Write a SQL code that will list all vendors who have supplied a part
 (You must ensure that only unique V_CODE values are displayed). Also retrieve the
number
 of vendors referenced in PRODUCT who have supplied products with prices less than or
equal to 10.
 */
 --------------------------------------------------------------------------
 SELECT DISTINCT V_CODE FROM PART WHERE V_CODE IS NOT NULL;

    V_CODE

----------

     25595

     23119

     21231

     21225

     24288

     21344


6 rows selected.

SQL> SELECT DISTINCT V_CODE FROM PART WHERE V_CODE IS NOT NULL AND PT_PRICE<=10;

    V_CODE

----------

     21231

     21225

     21344


3 rows selected.

 ------------------------------------ QUERY-03 -------------------------------
 /*
 Write a SQL code that will retrieve the product particulars for the parts with the
highest and
 the lowest price. Use this query to retrieve the product particulars for the parts with
the
 highest and the lowest inventory value (In both outputs the highest price products
should be listed first).
 */
 --------------------------------------------------------------------------
SQL> SELECT * FROM PRODUCT WHERE
  2  P_PRICE=(SELECT MAX(P_PRICE) FROM PRODUCT)
  3  OR
  4  P_PRICE=(SELECT MIN(P_PRICE) FROM PRODUCT);

P_COD DESCRIPT          P_DATE        QTY      P_MIN    P_PRICE     P_DISC     V_CODE
```

```
----- ---------------- --------- ------- ---------- ---------- ---------- ----------

RF100 Rat Tail File    15-DEC-11      43         20       4.99          0      21344

HC100 Hicut Chain Saw  07-FEB-12      11          5     256.99        .05      24288


2 rows selected.

SQL>
SQL> SELECT * FROM PRODUCT WHERE
  2  (P_PRICE*QTY)=(SELECT MAX(P_PRICE*QTY) FROM PRODUCT)
  3  OR
  4  (P_PRICE*QTY)=(SELECT MIN(P_PRICE*QTY) FROM PRODUCT);

P_COD DESCRIPT          P_DATE        QTY      P_MIN    P_PRICE     P_DISC     V_CODE

----- ---------------- --------- ------- ---------- ---------- ---------- ----------

SH100 Sledge Hammer    02-JAN-12       8          5       14.4        .05

HC100 Hicut Chain Saw  07-FEB-12      11          5     256.99        .05      24288


2 rows selected.


 ---------------------------------- QUERY-04 ------------------------------
 /*
 Write a SQL code that will retrieve the total amount owed by the customers as
TOT_BALANCE.
 Also compute the total value of all items carried in inventory.
 */
 ------------------------------------------------------------------------------
SQL> SELECT C_CODE,SUM(L_UNITS*L_PRICE) AS TOT_COST FROM LINE,INVOICE WHERE
LINE.INV_NUM=INVOICE.INV_NUM GROUP BY C_CODE;

    C_CODE   TOT_COST

---------- ----------

     10015      34.97

     10014     422.77

     10011     479.57

     10012     153.85

     10018      34.87


5 rows selected.

SQL> SELECT SUM(L_UNITS*L_PRICE)TOTAL_INVENTORY FROM LINE;

TOTAL_INVENTORY

---------------
```

```
        1126.03


1 row selected.


 ---------------------------------- QUERY-05 ------------------------------
 /*
 Write a SQL code that will retrieve the product particulars for all products whose
prices
 (largest first) exceed the average product price of the inventory. Also list the number
of
 products which are supplied by each vendor.
 */
 ------------------------------------------------------------------------------
 SQL>SELECT * FROM PRODUCT WHERE P_PRICE>(SELECT AVG(P_PRICE) FROM PRODUCT) ORDER BY
P_PRICE DESC;

P_COD DESCRIPT            P_DATE        QTY      P_MIN    P_PRICE     P_DISC     V_CODE

----- ------------------ --------- ------- ---------- ---------- ---------- ----------

HC100 Hicut Chain Saw     07-FEB-12      11          5     256.99        .05      24288

SM48X Steel Malting Mesh  17-JAN-12      18          5     119.95         .1      25595

AB112 Power Painter       03-NOV-11       8          5     109.99          0      25595

JB012 Jigsaw 12in Blade   30-DEC-11       8          5     109.92        .05      24288

JB008 Jigsaw 8in Blade    24-DEC-11       6          5      99.87        .05      24288


5 rows selected.

SQL> SELECT V_CODE,COUNT(*) FROM PRODUCT WHERE V_CODE IS NOT NULL GROUP BY V_CODE;

    V_CODE   COUNT(*)

---------- ----------

     25595          3

     23119          2

     21231          1

     21225          2

     24288          3

     21344          3


6 rows selected.

 ---------------------------------- QUERY-06 ------------------------------
 /*
 Write a SQL code to generate a listing of the number of products in the inventory
```

```
supplied
 by each vendor that has prices average below 10. Extend this query to generate a
listing of
 the total cost of products for each vendor – TOT_COST, such that the total cost exceeds
500.00
 and the high value vendor is placed last.
 */
 --------------------------------------------------------------------------
SQL> SELECT V_CODE,COUNT(P_CODE) AS CNT,AVG(P_PRICE) AS AVG_PRICE FROM PRODUCT GROUP BY
V_CODE HAVING AVG(P_PRICE)<10;

    V_CODE        CNT  AVG_PRICE

---------- ---------- ----------

     21231          1       8.45

     21225          2       8.47


2 rows selected.

SQL> SELECT V_CODE,COUNT(P_CODE) AS CNT,AVG(P_PRICE) AS AVG_PRICE,SUM(QTY*P_PRICE) AS
TOT_COST FROM PRODUCT
  2  WHERE V_CODE IS NOT NULL  GROUP BY V_CODE HAVING SUM(QTY*P_PRICE)>500 ORDER BY
TOT_COST DESC;

    V_CODE        CNT  AVG_PRICE   TOT_COST

---------- ---------- ---------- ----------

     24288          3 155.593333    4305.47

     25595          3      89.63    3506.42

     21231          1       8.45    2002.65

     23119          2      41.97    1611.02

     21225          2       8.47    1431.13

     21344          3      12.49    1009.07


6 rows selected.

 ----------------------------------- QUERY-07 -------------------------------
 /*
 Write a SQL code to create a view – PRODUCT_STATS from PRODUCT that generate a report
that shows
 a summary of total product cost – TOT_COST, and statistics on the quantity on hand
 [maximum – MX_QTY, minimum – MN_QTY, average – AV_QTY] for each vendor.
 */
 --------------------------------------------------------------------------
SQL> CREATE OR REPLACE VIEW PRODUCT_STATS AS
  2  SELECT V_CODE,SUM(QTY*P_PRICE) TOT_COST,MIN(QTY) MINQTY,MAX(QTY) MAXQTY,AVG(QTY) AS
AVGQTY FROM PRODUCT WHERE V_CODE IS NOT NULL GROUP BY V_CODE;

View created.
```

```
SQL> SELECT * FROM PRODUCT_STATS;

    V_CODE   TOT_COST     MINQTY     MAXQTY     AVGQTY

---------- ---------- ---------- ---------- ----------

     25595    3506.42          8         18 12.6666667

     23119    1611.02         15         23         19

     21231    2002.65        237        237        237

     21225    1431.13         23        172       97.5

     24288    4305.47          6         11 8.33333333

     21344    1009.07         18         43         31


6 rows selected.

 ----------------------------------- QUERY-08 -----------------------------
 /*
 Write a SQL code to count the number of invoices. Also count the number of customers
with a customer balance over 500.
 */
 ------------------------------------------------------------------------
SQL> SELECT COUNT(*)FROM INVOICE;

  COUNT(*)

----------

         8


1 row selected.

SQL> SELECT COUNT(CUSTOMER.C_CODE)FROM CUSTOMER WHERE BALANCE>500;

COUNT(CUSTOMER.C_CODE)

----------------------

                     2


1 row selected.

 ----------------------------------- QUERY-09 -----------------------------
 /*
 Write a SQL query that will list for each customer who has made purchases, the customer
number,
 the customer balance and the aggregate purchase amount.
 */
 ------------------------------------------------------------------------
SQL> SELECT C.C_CODE,C.BALANCE, AVG(L.L_UNITS*L.L_PRICE) AGGREGATE_PURCHASE
  2  FROM LINE L JOIN INVOICE I ON I.INV_NUM=L.INV_NUM
  3  JOIN CUSTOMER C
  4  ON C.C_CODE=I.C_CODE
```

```
  5  GROUP BY C.C_CODE,C.BALANCE;

    C_CODE    BALANCE AGGREGATE_PURCHASE

---------- ---------- ------------------

     10011          0             95.914

     10014          0         70.4616667

     10012     345.86         51.2833333

     10018     216.55             17.435

     10015          0             17.485


5 rows selected.
```

```
 ------------------------------------ QUERY-10 -------------------------------
 /*
 Modify Query-09 to include the number of individual product purchases made by each
customer.
 */
 ------------------------------------------------------------------------------
SQL> SELECT C.C_CODE,(FNAME||' '||LNAME) AS
NAME,L.P_CODE,BALANCE,(L_UNITS*L_PRICE)PRODUCT_PURCHASE FROM CUSTOMER C,INVOICE I,LINE L
WHERE
  2  L.INV_NUM=I.INV_NUM AND I.C_CODE=C.C_CODE ORDER BY C.C_CODE;

    C_CODE NAME                 P_COD    BALANCE PRODUCT_PURCHASE

---------- -------------------- ----- ---------- ----------------

     10011 Elena Kurtis         PP101          0             70.44

     10011 Elena Kurtis         SM48X          0            359.85

     10011 Elena Kurtis         PP101          0             29.35

     10011 Elena Kurtis         CH10X          0              9.95

     10011 Elena Kurtis         RF100          0              9.98

     10012 Kathy Smith          SB725     345.86             74.95

     10012 Kathy Smith          CD00X     345.86             38.95

     10012 Kathy Smith          CL025     345.86             39.95

     10014 Bill Johnson         CH10X          0              9.95

     10014 Bill Johnson         CH10X          0              9.95

     10014 Bill Johnson         HC100          0            256.99


    C_CODE NAME                 P_COD    BALANCE PRODUCT_PURCHASE

---------- -------------------- ----- ---------- ----------------
```

```
     10014 Bill Johnson          SB725          0            14.99

     10014 Bill Johnson          MC001          0            20.97

     10014 Bill Johnson          JB012          0           109.92

     10015 Julia Samuels         SB725          0            29.98

     10015 Julia Samuels         RF100          0             4.99

     10018 Ming Lee              RF100     216.55            14.97

     10018 Ming Lee              CH10X     216.55             19.9


18 rows selected.

 ------------------------------------ QUERY-11 -------------------------------
 /*
 Write a SQL query to compute the average purchase amount per product made by each
customer.
*/
 ---------------------------------------------------------------------------
SQL> SELECT C_CODE,P_CODE,AVG(L_UNITS*L_PRICE) AS AVERAGE FROM INVOICE I,LINE L WHERE
I.INV_NUM=L.INV_NUM GROUP BY (P_CODE,C_CODE);

    C_CODE P_COD    AVERAGE

---------- ----- ----------

     10011 RF100       9.98

     10018 RF100      14.97

     10014 CH10X       9.95

     10018 CH10X       19.9

     10012 CL025      39.95

     10015 SB725      29.98

     10014 SB725      14.99

     10012 CD00X      38.95

     10012 SB725      74.95

     10011 PP101     49.895

     10014 JB012     109.92


    C_CODE P_COD    AVERAGE

---------- ----- ----------

     10014 HC100     256.99
```

```
     10015 RF100       4.99

     10014 MC001      20.97

     10011 SM48X     359.85

     10011 CH10X       9.95


16 rows selected.

 ------------------------------------ QUERY-12 ------------------------------
 /*
 Write a SQL query to produce the total purchase per invoice (The invoice total is the
sum
 of the product purchases in the LINE that corresponds to the INVOICE). Further, produce
a
 listing showing invoice numbers with corresponding invoice total identified to a
customer
 (Use GROUP BY on C_CODE). Also generate a listing showing the number of invoices and
the
 total purchase amounts by customer.
 */
 ------------------------------------------------------------------------------
SQL> SELECT INV_NUM,SUM(L_UNITS * L_PRICE)TOTAL_PURCHASE FROM LINE GROUP BY INV_NUM
ORDER BY INV_NUM;

   INV_NUM TOTAL_PURCHASE

---------- --------------

      1001          24.94

      1002           9.98

      1003         153.85

      1004          34.87

      1005          70.44

      1006         397.83

      1007          34.97

      1008         399.15


8 rows selected.

SQL>
SQL> SELECT I.C_CODE,L.INV_NUM,SUM(L.L_UNITS * L.L_PRICE)TOTAL_PURCHASE FROM LINE
L,INVOICE I WHERE L.INV_NUM = I.INV_NUM GROUP BY L.INV_NUM,I.C_CODE ORDER BY I.C_CODE;

    C_CODE    INV_NUM TOTAL_PURCHASE

---------- ---------- --------------

     10011       1002           9.98
```

```
     10011       1005          70.44

     10011       1008         399.15

     10012       1003         153.85

     10014       1001          24.94

     10014       1006         397.83

     10015       1007          34.97

     10018       1004          34.87


8 rows selected.

SQL>
SQL> SELECT  DISTINCT I.C_CODE,COUNT(L.INV_NUM),SUM(L.L_UNITS*L.L_PRICE)TOT_PURCHASE
FROM INVOICE I,LINE L  WHERE L.INV_NUM = I.INV_NUM GROUP BY I.C_CODE ORDER BY I.C_CODE;

    C_CODE COUNT(L.INV_NUM) TOT_PURCHASE

---------- ---------------- ------------

     10011                5       479.57

     10012                3       153.85

     10014                6       422.77

     10015                2        34.97

     10018                2        34.87


5 rows selected.


 ----------------------------------- QUERY-13 ------------------------------
 /*
 Using the results of Query-12, write a SQL code to generate the total number of
invoices, the invoice
 total for all of the invoices, the smallest invoice amount, the largest invoice amount,
and the average
 of all of the invoices.
 */
 ------------------------------------------------------------------------------
SQL> SELECT DISTINCT INV_NUM,COUNT(INV_NUM),SUM(L_UNITS*L_PRICE)
TOT,MIN(L_UNITS*L_PRICE)IMIN,MAX(L_UNITS*L_PRICE)IMAX,AVG(L_UNITS*L_PRICE)IAVG
  2  FROM LINE GROUP BY INV_NUM ORDER BY INV_NUM;

   INV_NUM COUNT(INV_NUM)        TOT       IMIN       IMAX       IAVG

---------- -------------- ---------- ---------- ---------- ----------

      1001              2      24.94       9.95      14.99      12.47

      1002              1       9.98       9.98       9.98       9.98
```

1003	3	153.85	38.95	74.95	51.2833333
1004	2	34.87	14.97	19.9	17.435
1005	1	70.44	70.44	70.44	70.44
1006	4	397.83	9.95	256.99	99.4575
1007	2	34.97	4.99	29.98	17.485
1008	3	399.15	9.95	359.85	133.05

8 rows selected.

```
 ---------------------------------- QUERY-14 ------------------------------
 /*
 Write a SQL code to find the customer balance summary for all customers who have not
made purchases during
 the current invoicing. Use this query to generate a summary of the customer balance
characteristics
 */
 ------------------------------------------------------------------------------
SQL> SELECT C_CODE,BALANCE FROM CUSTOMER WHERE C_CODE NOT IN (SELECT C_CODE FROM INVOICE
WHERE C_CODE IS NOT NULL);
```

C_CODE	BALANCE
10016	221.19
10017	768.93
10013	536.75
10010	0
10019	0

5 rows selected.

```
SQL> SELECT C.C_CODE,SUM(L_UNITS*L_PRICE)
TOT,MIN(L_UNITS*L_PRICE)MINIMUM,MAX(L_UNITS*L_PRICE)MAXIMUM,AVG(L_UNITS*L_PRICE)AVERAGE
  2  FROM INVOICE I,LINE L ,CUSTOMER C WHERE C.C_CODE=I.C_CODE AND I.INV_NUM=L.INV_NUM
GROUP BY C.C_CODE;
```

C_CODE	TOT	MINIMUM	MAXIMUM	AVERAGE
10015	34.97	4.99	29.98	17.485
10014	422.77	9.95	256.99	70.4616667
10011	479.57	9.95	359.85	95.914
10012	153.85	38.95	74.95	51.2833333

```
     10018      34.87      14.97       19.9     17.435


5 rows selected.

 ---------------------------------- QUERY-15 ------------------------------
 /*
 Write a SQL code to find the customer balance summary for all customers who have not
made purchases during
 the current invoicing period (the output should include the total balances, the
minimum, maximum and average
 balances over across all purchases). Also compute the total value of the product
inventory.
 */
 ------------------------------------------------------------------------------
SQL> SELECT SUM(BALANCE) SUM_BAL,MIN(BALANCE)MIN_BAL, MAX(BALANCE)MAX_BAL,
AVG(BALANCE)AVG_BAL
  2      FROM
  3      (SELECT C_CODE ,BALANCE FROM CUSTOMER
  4      MINUS
  5      SELECT C.C_CODE,BALANCE FROM CUSTOMER C,INVOICE I WHERE C.C_CODE=I.C_CODE);

   SUM_BAL    MIN_BAL    MAX_BAL    AVG_BAL

---------- ---------- ---------- ----------

   1526.87          0     768.93    305.374


1 row selected.

SQL> SELECT (SELECT SUM(P_PRICE*QTY)  FROM PRODUCT
  2  MINUS
  3  SELECT SUM(P.P_PRICE*P.QTY)FROM PRODUCT P,LINE L,INVOICE I WHERE P.P_CODE =
L.P_CODE AND I.INV_NUM=L.INV_NUM) INV_COST FROM DUAL;

  INV_COST

----------

  15084.52


1 row selected.

 ---------------------------------- END OF QUERIES----------------------------

SQL> SET FEEDBACK OFF
SQL> SPOOL OFF
```