

**Week 11**

17.

```
SELECT
    I.CUS_CODE,
    L.INV_NUMBER,
    ROUND( SUM( L.LINE_UNITS * L.LINE_PRICE ), 2 ) AS "Invoice Total"
FROM
    INVOICE I
    JOIN LINE L ON I.INV_NUMBER = L.INV_NUMBER
GROUP BY
    I.CUS_CODE,
    L.INV_NUMBER
ORDER BY
    I.CUS_CODE;
```

18.

```
SELECT
    I.CUS_CODE,
    COUNT( L.INV_NUMBER ) AS "Number of Invoices",
    ROUND( SUM( L.LINE_UNITS * L.LINE_PRICE ), 2 ) AS "Total Customer
Purchases"
FROM
    INVOICE I
    JOIN LINE L ON I.INV_NUMBER = L.INV_NUMBER
GROUP BY
    I.CUS_CODE;
```

19.

```
SELECT SUM("Number of Invoices") AS "Total Invoices",  
  
SUM("Total Customer Purchases") AS "Total Sales",  
  
MIN("Total Customer Purchases") AS "Minimum Customer Purchases",  
  
MAX("Total Customer Purchases") AS "Largest Customer Purchases",  
  
Round (AVG("Total Customer Purchases"), 2) AS "Average Customer Purchases"  
  
FROM (SELECT DISTINCT INVOICE.CUS_CODE,  
  
COUNT(DISTINCT INVOICE.INV_NUMBER) AS "Number of Invoices",  
  
Round (SUM(LINE_PRICE*LINE_UNITS), 2) AS "Total Customer Purchases"  
  
FROM CUSTOMER, INVOICE, LINE  
  
WHERE CUSTOMER.CUS_CODE=INVOICE.CUS_CODE  
  
AND INVOICE.INV_NUMBER=LINE.INV_NUMBER  
  
GROUP BY INVOICE.CUS_CODE) Customer_Purchases;
```

20.

```
SELECT  
    I.CUS_CODE,  
    C.CUS_BALANCE  
FROM  
    CUSTOMER C  
    INNER JOIN INVOICE I ON C.CUS_CODE = I.CUS_CODE  
GROUP BY  
    I.CUS_CODE,  
    C.CUS_BALANCE;
```

21.

```
SELECT  
    MIN( C.CUS_BALANCE ) AS "Minimum Balance",  
    MAX( C.CUS_BALANCE ) AS "Maximum Balance",  
    ROUND( AVG( C.CUS_BALANCE ), 2 ) AS "Average Balance"  
FROM  
    CUSTOMER C  
WHERE  
    CUS_CODE IN ( SELECT CUS_CODE FROM INVOICE );
```

22.

```
SELECT
    Sum( CUSTOMER.CUS_BALANCE ) AS "Total Balances",
    Min( CUSTOMER.CUS_BALANCE ) AS "Minimum Balance",
    Max( CUSTOMER.CUS_BALANCE ) AS "Maximum Balance",
    ROUND( Avg( CUSTOMER.CUS_BALANCE ), 2 ) AS "Average Balance"
FROM
    CUSTOMER;
```

23.

```
SELECT
    CUS_CODE,
    CUS_BALANCE
FROM
    CUSTOMER
WHERE
    CUSTOMER.CUS_CODE NOT IN ( SELECT DISTINCT CUS_CODE FROM INVOICE );
```

24.

```
SELECT
    SUM( cb.CUS_BALANCE ) AS 'Total Balance',
    MIN( cb.CUS_BALANCE ) AS 'Minimum Balance',
    MAX( cb.CUS_BALANCE ) AS 'Maximum Balance',
    AVG( cb.CUS_BALANCE ) AS 'Average Balance'
FROM
    ( SELECT CUS_CODE, CUS_BALANCE FROM CUSTOMER WHERE CUSTOMER.CUS_CODE
NOT IN ( SELECT DISTINCT CUS_CODE FROM INVOICE ) ) AS cb;
```

25.

```
SELECT
    P.P_DESCRIPT,
    P.P_QOH,
    P.P_PRICE,
    P.P_QOH * P.P_PRICE AS Expr1
FROM
    PRODUCT P
ORDER BY
    P.P_QOH * P.P_PRICE DESC;
```

26.

```
SELECT
    Sum( P.P_QOH * P.P_PRICE ) AS 'Total Value of Inventory'
FROM
    PRODUCT P;
```

27.

```
SELECT
    *
FROM
    LGDEPARTMENT
ORDER BY
    DEPT_NAME;
```

28.

```
SELECT
    PROD_SKU,
    PROD_DESCRIPT,
    PROD_TYPE,
    PROD_BASE,
    PROD_CATEGORY,
    PROD_PRICE
FROM
    LGPRODUCT
WHERE
    PROD_BASE = 'Water'
    AND PROD_CATEGORY = 'Sealer';
```

29.

```
SELECT
    E.EMP_FNAME,
    E.EMP_LNAME,
    E.EMP_EMAIL
FROM
    LGEMPLOYEE E
WHERE
    E.EMP_HIREDATE BETWEEN '2005-01-05'
    AND '2014-12-31'
ORDER BY
    E.EMP_LNAME,
    E.EMP_FNAME;
```

30.

```
SELECT
    EMP_FNAME,
    EMP_LNAME,
    EMP_PHONE,
    EMP_TITLE,
    DEPT_NUM
FROM
    LGEMPLOYEE
WHERE
    DEPT_NUM = 300
    OR EMP_TITLE = 'CLERK I'
ORDER BY
    EMP_LNAME,
    EMP_FNAME;
```