1.

1 point

Query 1: Identify the tables to join in the SELECT statement solution.

- ☑ cust_vendor_dim
- ☐ branch_plant_dim
- ☑ inventory_fact
- ☑ date_dim
- ☐ item_master_dim

2.

1 point

Query 1: Identify the columns with single table conditions in the WHERE clause of the SELECT statement.

- ☐ TransDescription
- ☐ CalMonth
- ☑ CalYear
- ☑ TransTypeKey

3.

1 point

Query 1: Identify the subtotal operator(s) in the GROUP BY clause.

- ○ No subtotal operators
- ○ ROLLUP
- ● CUBE
- ○ UNION

4.

1 point

Query 1: Identify the columns inside the subtotal operator used in the GROUP BY clause.

- ☑ CalMonth
- ☐ No columns used
- ☐ CalQuarter
- ☐ SecondItemId
- ☑ AddrCatCode1

5.

1 point

Query 1: Identify the columns appearing after the SELECT keyword. This question involves columns by themselves, not aggregate function calculations.

- ☑ CalMonth
- ☐ No columns used
- ☑ AddrCatCode1
- ☐ SecondItemId
- ☐ CalQuarter

6.

1 point

Query 1: Identify the aggregate functions appearing after the SELECT keyword. This question involves aggregate calculations, not grouping columns appearing by themselves after the SELECT keyword.

- ☑ SUM(ExtCost)
- ☐ No aggregate function calculations
- ☐ COUNT
- ☐ SUM(UnitCost)
- ☑ SUM(Quantity)

7.

1 point

Query 1: How many rows appear in the result using the original data warehouse tables?

- ○ 124
- ○ 87
- ● 78
- ○ 64

8.

1 point

Query 2: Identify the tables to join in the SELECT statement solution.

- ☑ inventory_fact
- ☑ date_dim
- ☐ item_cat_code1
- ☐ trans_type_dim
- ☑ cust_vendor_dim

9. **Query 2**: Identify the columns with single table conditions in the WHERE clause of the SELECT statement.

1 point

- [ ] CalYear
- [ ] UnitCost
- [ ] TransTypeCodeId
- [x] TransTypeKey

10. **Query 2**: Identify the subtotal operator(s) in the GROUP BY clause.

1 point

- ( ) ROLLUP
- (•) GROUPING SETS
- ( ) No subtotal operators
- ( ) UNION

11. **Query 2**: Identify the columns inside the subtotal operator used in the GROUP BY clause.

1 point

- [ ] CustVendorKey
- [x] Name
- [x] CalQuarter
- [ ] CalYear
- [x] Zip

12. **Query 2**: Identify the columns appearing after the SELECT keyword. This question involves columns by themselves, not aggregate function calculations.

1 point

- [x] CalQuarter
- [x] Zip
- [x] Name
- [ ] CalYear
- [ ] AddrCatCode1

13. **Query 2**: Identify the aggregate functions appearing after the SELECT keyword. This question involves aggregate calculations, not grouping columns appearing by themselves after the SELECT keyword.

1 point

- [ ] No aggregate function calculations
- [ ] SUM(Quantity)
- [x] COUNT(*)
- [ ] MAX(UnitCost)
- [x] SUM(ExtCost)

14. **Query 2**: How many rows appear in the result using the original data warehouse tables?

1 point

- ( ) 90
- (•) 305
- ( ) 510
- ( ) 315

15. **Query 3**: Identify the columns with single table conditions in the WHERE clause of the SELECT statement.

1 point

- ( ) ItemMasterKey
- ( ) CarryingCost
- ( ) CalYear
- (•) TransTypeKey

16. **Query 3**: Identify the subtotal operator(s) in the GROUP BY clause.

1 point

- ( ) Partial CUBE
- (•) ROLLUP
- ( ) CUBE
- ( ) No subtotal operators

17. **Query 3**: Identify the columns inside the subtotal operator used in the GROUP BY clause.

1 point

- [ ] CustVendorKey

- [ ] SecondItemId
- [ ] DateKey
- [x] CompanyName
- [x] BPName

---

18. 1 point

Query 3: Identify the columns appearing after the SELECT keyword. This question involves columns by themselves, not aggregate function calculations.

- [ ] SecondItemI
- [x] BPName
- [ ] DateKey
- [ ] AddrCatCode1
- [x] CompanyName

---

19. 1 point

Query 3: Identify the aggregate functions appearing after the SELECT keyword. This question involves aggregate calculations, not grouping columns appearing by themselves after the SELECT keyword.

- [ ] MAX(UnitCost)
- [ ] COUNT(*)
- [x] SUM(ExtCost)
- [x] SUM(Quantity)

---

20. 1 point

Query 3: How many rows appear in the result using the original data warehouse tables?

- ( ) 215
- ( ) 322
- ( ) 46
- (•) 26

---

21. 1 point

Query 4: Identify the tables to join in the SELECT statement.

- [x] trans_type_dim
- [x] company_dim
- [x] inventory_fact
- [ ] item_master_dim
- [x] branch_plant_dim

---

22. 1 point

Query 4: Identify the subtotal operator(s) in the GROUP BY clause.

- ( ) CUBE
- ( ) No subtotal operators
- (•) GROUPING SETS
- ( ) Nested ROLLUP

---

23. 1 point

Query 4: Identify the columns inside the subtotal operator used in the GROUP BY clause.

- [x] BPName
- [ ] Quantity
- [ ] CompanyKey
- [x] CompanyName
- [x] TransDescription

---

24. 1 point

Query 4: Identify the columns appearing after the SELECT keyword. This question involves columns by themselves, not aggregate function calculations.

- [ ] ExtCost
- [ ] CompanyId
- [x] CompanyName
- [x] BPName
- [x] TransDescription

---

25. 1 point

Query 4: Identify the aggregate function(s) appearing after the SELECT keyword. This question involves aggregate calculations, not grouping columns appearing by themselves after the SELECT keyword.

- [x] SUM(ExtCost)
- [x] COUNT(*)

- [ ] No aggregate function calculations
- [ ] SUM(UnitCost)

---

26. **Query 4**: How many rows appear in the result using the original data warehouse tables?

1 point

- ( ) 105
- (•) 163
- ( ) 296
- ( ) 16

---

27. **Query 5**: Identify the tables to join in the SELECT statement.

1 point

- [ ] item_master_dim
- [x] inventory_fact
- [x] date_dim
- [ ] branch_plant_dim
- [x] cust_vendor_dim

---

28. **Query 5**: Identify the columns with single table conditions in the WHERE clause of the SELECT statement.

1 point

- [ ] TransDescription
- [ ] CalMonth
- [x] CalYear
- [x] TransTypeKey

---

29. **Query 5**: Identify the subtotal operator(s) in the GROUP BY clause.

1 point

- (•) ROLLUP on all grouping columns
- ( ) CUBE on all grouping columns
- ( ) Partial ROLLUP on some grouping columns
- ( ) Partial CUBE

---

30. **Query 5**: Identify the columns inside the subtotal operator used in the GROUP BY clause.

1 point

- [ ] AddrCatCode1
- [x] CalQuarter
- [x] CalYear
- [ ] SecondItemId
- [ ] No columns used

---

31. **Query 5**: Identify the aggregate functions appearing after the SELECT keyword. This question involves aggregate calculations, not grouping columns appearing by themselves after the SELECT keyword.

1 point

- [ ] SUM(Quantity)
- [ ] SUM(UnitCost)
- [x] COUNT(*)
- [x] SUM(ExtCost)
- [ ] No aggregate function calculations

---

32. **Query 5**: How many rows appear in the result using the original data warehouse tables?

1 point

- ( ) 211
- ( ) 21
- ( ) 10
- (•) 220

---

33. **Query 6**: Identify the aggregate functions appearing after the SELECT keyword for all blocks. This question involves aggregate calculations, not grouping columns appearing by themselves after the SELECT keyword.

1 point

- [ ] SUM(UnitCost)
- [x] SUM(ExtCost)
- [ ] COUNT(*)
- [x] SUM(Quantity)

34. **Query 6**: How many SELECT blocks appear in the statement?

1 point

○ 4

○ 5

○ 2

● 3

35. **Query 6**: In the query block to produce the grand totals, what expressions for the grouping columns are used in the SELECT clause?

1 point

○ NULL,NULL

○ CalMonth, NULL

● CalMonth, AddrCode1

○ NULL. AddrCode1

36. **Query 6**: In the query block to produce the grand totals, what column(s) appear in the GROUP BY clause?

1 point

○ AddrCode1

○ CalMonth, AddrCode1

● CalMonth

○ No GROUP BY clause is used.

37. **Query 6**: In the query block to produce subtotals for AddrCatCode1, what expressions for the grouping columns are used in the SELECT clause?

1 point

○ NULL, AddrCode1

○ NULL, NULL

○ CalMonth, AddrCode1

● CalMonth, NULL

38. **Query 6**: In the query block to produce the subtotals for CalMonth, what column(s) appear in the GROUP BY clause?

1 point

● No GROUP BY clause is used.

○ CalMonth

○ AddrCode1

○ CalMonth, AddrCode1

39. **Query 7**: Identify the aggregate functions appearing after the SELECT keyword for all blocks. This question involves aggregate calculations, not grouping columns appearing by themselves after the SELECT keyword.

1 point

☑ SUM(ExtCost)

☑ SUM(Quantity)

☐ COUNT(*)

☐ SUM(UnitCost)

40. **Query 7**: How many SELECT blocks appear in the statement?

1 point

○ 4

● 3

○ 5

○ 2

41. **Query 7**: In the query block to produce the grand total, what expressions for the grouping columns are used in the SELECT clause?

1 point

● CompanyName, BPName

○ CompanyName, NULL

○ NULL, BPName

○ NULL, NULL

42. **Query 7**: In the query block to produce the grand total, what column(s) appear in the GROUP BY clause?

1 point

○ BPName

○ CompanyName, BPName

○ CompanyName

● No GROUP BY clause is used.

43. 1 point **Query 7**: In the query block to produce subtotals for CompanyName, what expressions for the grouping columns are used in the SELECT clause?

- ● NULL, NULL
- ○ NULL, BPName
- ○ CompanyName, NULL
- ○ CompanyName, BPName

44. 1 point **Query 7**: In the query block to produce the subtotals for BPName, what column(s) appear in the GROUP BY clause?

- ○ No subtotals are produced for BPName so there is no query block.
- ● CompanyName, BPName
- ○ BPName
- ○ CompanyName

45. 1 point **Query 8**: Identify the tables to join in the SELECT statement solution.

- ☐ item_master_dim
- ☑ cust_vendor_dim
- ☑ inventory_fact
- ☑ date_dim
- ☐ branch_plant_dim

46. 1 point **Query 8**: Identify the columns with conditions in the WHERE clause of the SELECT statement.

- ☐ TransDescription
- ☑ TransTypeKey
- ☑ CalYear
- ☐ CalMonth

47. 1 point **Query 8**: Identify the subtotal operator(s) in the GROUP BY clause.

- ○ Partial ROLLUP on some grouping columns
- ○ Partial CUBE on some grouping columns
- ● CUBE using a grouping column and a composite column
- ○ ROLLUP on all grouping columns

48. 1 point **Query 8**: Identify the columns inside the subtotal operator used in the GROUP BY clause.

- ☑ Name
- ☐ CalYear
- ☐ CalQuarter
- ☑ (CalYear, CalQuarter)
- ☐ SecondItemId

49. 1 point **Query 8**: Identify the columns appearing after the SELECT keyword. This question involves columns by themselves, not aggregate function calculations.

- ☑ Name
- ☑ CalQuarter
- ☐ AddrCatCode1
- ☐ SecondItemId
- ☑ CalYear

50. 1 point **Query 8**: How many rows appear in the result using the original data warehouse tables?

- ○ 220
- ● 189
- ○ 130
- ○ 211

51. 1 point **Query 9**: Identify the tables to join in the SELECT statement solution.

- ☑ cust_vendor_dim
- ☑ date_dim
- ☐ item_master_dim

□
□ branch_plant_dim

■ inventory_fact

---

52. **1 point**   **Query 9**: Identify the subtotal operator(s) in the GROUP BY clause.

○ CUBE on all grouping columns

○ CUBE using a composite column

● Partial CUBE on some grouping columns

○ ROLLUP on all grouping columns

---

53. **1 point**   **Query 9**: Identify the columns inside the subtotal operator used in the GROUP BY clause.

■ CalMonth

□ AddrCatCode1

□ CalQuarter

□ (CalYear, CalMonth)

■ CalYear

---

54. **1 point**   **Query 9**: How many rows appear in the result using the original data warehouse tables?

○ 189

○ 90

● 78

○ 130

---

55. **1 point**   **Query 9**: What is the GROUPING_ID value of the grand total row?

○ 2

● 0

○ 3

○ 1

---

56. **1 point**   **Query 9**: What is the GROUPING_ID value of normal GROUP BY rows with a value for all grouping columns?

● 1

○ 3

○ 2

○ 0

---

57. **1 point**   **Query 10**: Identify the tables to join in the SELECT statement solution.

■ cust_vendor_dim

□ branch_plant_dim

□ item_master_dim

■ inventory_fact

■ date_dim

---

58. **1 point**   **Query 10**: Identify the subtotal operator(s) in the GROUP BY clause.

■ GROUPING SETS

□ Partial ROLLUP on some grouping columns

□ CUBE nested inside GROUPING SETS

■ ROLLUP nested inside GROUPING SETS

---

59. **1 point**   **Query 10**: Identify the columns inside the subtotal operator used in the GROUP BY clause.

■ Nested ROLLUP using CalYear, CalQuarter

□ CalYear inside GROUPING SETS but not nested in another subtotal operator

□ CalMonth

■ Name inside GROUPING SETS

□ CalQuarter inside GROUPING SETS but not nested in another subtotal operator

---

60. **1 point**   **Query 10**: Identify the columns appearing after the SELECT keyword. This question involves columns by themselves, not aggregate function calculations.

- [■] Name
- [■] CalQuarter
- [ ] CalMonth
- [ ] AddrCatCode1
- [■] CalYear

---

1 point

61. **Query 10**: Identify the aggregate functions appearing after the SELECT keyword. This question involves aggregate calculations, not grouping columns appearing by themselves after the SELECT keyword.

- [ ] No aggregate function calculations
- [ ] SUM(UnitCost)
- [ ] SUM(Quantity)
- [■] SUM(ExtCost)
- [■] COUNT(*)

---

1 point

62. **Query 10**: How many rows appear in the result using the original data warehouse tables?

- ( ) 78
- ( ) 45
- ( ) 130
- (●) 31

Submit Quiz