WEEK 5 ASSESSMENT: Questions, Answers, and Feedback Responses

Question 1: How many distinct skus have the brand "Polo fas", and are either size "XXL" or "black" in color?

A. 84
B. 5,224
C. 13,623
D. 27,271

Correct Answers: C

Correct Feedback: The query you used to arrive at your answer might have looked like this:

```
SELECT COUNT(DISTINCT sku)
FROM skuinfo
WHERE brand = 'polo fas' AND (color = 'black' OR size = 'XXL');
```

Incorrect Feedback: Pay careful attention to the order of operations in your query.

Question 2: There was one store in the database, which had only 11 days in one of its months (in other words, that store/month/year combination only contained 11 days of transaction data). In what city and state was this store located?

A. Tulsa, OK
B. Richmond, VA
C. Little Rock, AR
D. Atlanta, GA

Correct Answers: D

Correct Feedback: The query you used to arrive at your answer might have looked like this:

```
SELECT DISTINCT t.store, s.city, s.state
FROM trnsact t JOIN strinfo s
ON t.store=s.store
WHERE t.store IN (SELECT days_in_month.store
FROM(SELECT EXTRACT(YEAR from saledate) AS sales_year,
EXTRACT(MONTH from saledate) AS sales_month, store, COUNT
(DISTINCT  saledate) as numdays
FROM trnsact
GROUP BY sales_year, sales_month, store
HAVING numdays=11) as days_in_month)
```

Incorrect Feedback: Try using subqueries to arrive at your answer.

Question 3: Which sku number had the greatest increase in total sales revenue from November to December?

A. 2637537
B. 3949538
C. 4737469

D. 6966816

Correct Answer: B

Correct Feedback: The query you used to arrive at your answer might have looked like this:

```
SELECT sku,
sum(case when extract(month from saledate)=11 then amt end) as November,
sum(case when extract(month from saledate)=12 then amt end) as December,
December-November AS sales_bump
FROM trnsact
WHERE stype='P'
GROUP BY sku
ORDER BY sales_bump DESC;
```

Incorrect Feedback: Consider using a CASE statement within an aggregate function to arrive at your answer.

Question 4. What vendor has the greatest number of distinct skus in the transaction table that do not exist in the skstinfo table? (Remember that vendors are listed as distinct numbers in our data set).

A. 3313116
B. 5511283
C. 5715232
D. 9514659

Correct Answer: C

Correct Feedback: The query you used to arrive at your answer might have looked like this:

```
SELECT count(DISTINCT t.sku) as num_skus, si.vendor
FROM trnsact t
LEFT JOIN skstinfo s
ON t.sku=s.sku AND t.store=s.store
JOIN skuinfo si  ON t.sku=si.sku
WHERE s.sku IS NULL
GROUP BY si.vendor
ORDER BY num_skus DESC;
```

Incorrect Feedback: You may want to review the MySQL Exercise 8, "Joining Tables with Outer Joins."

Question 5: What is the brand of the sku with the greatest standard deviation in sprice? Only examine skus which have been part of over 100 transactions.

A. Hart Sch
B. Polo Fas
C. Vanity F
D. Clinique

Correct Answer: A

Correct Feedback: You could have arrived at this answer with a subquery:

```
SELECT DISTINCT top10skus.sku, top10skus.sprice_stdev, top10skus.num_transactions, si.style, si.color,
si size,si.packsize, si.vendor, si.brand
FROM (SELECT TOP 1 sku, STDDEV_POP(sprice) AS sprice_stdev, count(sprice) AS num_transactions
FROM trnsact
WHERE stype='P'
GROUP BY sku
HAVING num_transactions > 100
ORDER BY sprice_stdev DESC) AS top10skus JOIN skuinfo si
ON top10skus.sku = si.sku
ORDER BY top10skus.sprice_stdev DESC;
```

Or without a subquery:

```
SELECT TOP 1 t.sku, STDDEV_POP(t.sprice) AS sprice_stdev, count(t.sprice) AS num_transactions, si.style, si.color, si.size,
si.packsize, si.vendor, si.brand
FROM trnsact t JOIN skuinfo si
ON t.sku = si.sku
WHERE stype='P'
GROUP BY t.sku, si.style, si.color, si.size, si.packsize, si.vendor, si.brand
HAVING num_transactions > 100
ORDER BY sprice_stdev DESC;
```

Incorrect Feedback: Make sure your ORDER BY statements are sorting the results in the correct direction, and remember that SELECT requests that specify a TOP operator cannot also specify a DISTINCT condition in Teradata.

Question 6. What is the city and state of the store that had the greatest increase in average daily revenue (as defined in Teradata Week 5 Exercise Guide) from November to December?

A. Little Rock, AK
B. McAllen, TX
C. Tucson, AZ
D. Metairie, LA

Correct Answer: D

Correct Feedback: The query you used to arrive at your answer might have looked like this:

```
SELECT s.city, s.state, t.store,
SUM(case WHEN EXTRACT(MONTH from saledate) =11 then amt END) as November,
SUM(case WHEN EXTRACT(MONTH from saledate) =12 then amt END) as December,
COUNT(DISTINCT (case WHEN EXTRACT(MONTH from saledate) =11 then saledate END)) as Nov_numdays,
COUNT(DISTINCT (case WHEN EXTRACT(MONTH from saledate) =12 then saledate END)) as Dec_numdays,
(December/Dec_numdays)-(November/Nov_numdays) AS dip
FROM trnsact t JOIN strinfo s
ON t.store=s.store
```

```
WHERE t.stype='P' AND t.store||EXTRACT(YEAR from t.saledate)||EXTRACT(MONTH from t.saledate) IN
 (SELECT store||EXTRACT(YEAR from saledate)||EXTRACT(MONTH from saledate)
FROM trnsact
GROUP BY store, EXTRACT(YEAR from saledate), EXTRACT(MONTH from saledate)
HAVING COUNT(DISTINCT saledate)>= 20)
GROUP BY s.city, s.state, t.store
ORDER BY dip DESC;
```

Incorrect Feedback: Make sure you are sorting your output in the correct direction, and computing average daily revenue by dividing by the correct number of days.

Question 7: Compare the average daily revenue (as defined in Teradata Week 5 Exercise Guide) of the store with the highest msa_income and the store with the lowest median msa_income (according to the msa_income field). In what city and state were these two stores, and which store had a higher average daily revenue?

A. The store with the highest median msa_income was in McAllen, TX. It had a higher average daily revenue than the store with the lowest median msa_income, which was in Spanish Fort, AL.
B. The store with the highest median msa_income was in Spanish Fort, AL. It had a lower average daily revenue than the store with the lowest median msa_income, which was in McAllen, TX.
C. The store with the highest median msa_income was in Littleton, CO. It had a higher average daily revenue than the store with the lowest median msa_income, which was in Cincinnati, OH.
D. The store with the highest median msa_income was in Cincinnati, OH. It had a lower average daily revenue than the store with the lowest median msa_income, which was in Littleton, CO.

Correct Answers: B

Correct Feedback: The query you used to arrive at your answer might have looked like this:

```
SELECT SUM(store_rev. tot_sales)/SUM(store_rev.numdays) AS daily_average, store_rev.msa_income as med_income,
store_rev.city,  store_rev.state
FROM (SELECT COUNT (DISTINCT t.saledate) as numdays, EXTRACT(YEAR from t.saledate) as s_year, EXTRACT(MONTH
from t.saledate) as s_month, t.store, sum(t.amt) as tot_sales, CASE when extract(year from t.saledate) = 2005 AND
extract(month from t.saledate) = 8 then 'exclude'
END as exclude_flag, m.msa_income, s.city, s.state
FROM trnsact t JOIN store_msa m
ON m.store=t.store JOIN strinfo s
ON t.store=s.store
WHERE t.stype = 'P' AND exclude_flag IS NULL
GROUP BY s_year, s_month, t.store, m.msa_income, s.city, s.state
HAVING numdays >= 20) as store_rev
WHERE store_rev.msa_income IN ((SELECT MAX(msa_income) FROM store_msa),(SELECT MIN(msa_income) FROM
store_msa))
GROUP BY med_income, store_rev.city, store_rev.state;
```

Incorrect Feedback: You might want to use a subquery to examine the details of the maximum and minimum msa_income values at the same time.

Question 8: Divide the msa_income groups up so that msa_incomes between 1 and 20,000 are labeled 'low', msa_incomes between 20,001 and 30,000 are labeled 'med-low', msa_incomes between 30,001 and 40,000 are labeled 'med-high', and msa_incomes between 40,001 and 60,000 are labeled 'high'. Which of these groups has the highest average daily revenue (as defined in Teradata Week 5 Exercise Guide) per store?

- **A. low**
- B. med-low
- C. med-high
- D. high

Correct Answers: A

Correct Feedback: The query you used to arrive at your answer might have looked like this:

```
SELECT SUM(revenue_per_store.revenue)/SUM(numdays) AS avg_group_revenue,
CASE WHEN revenue_per_store.msa_income BETWEEN 1 AND 20000 THEN 'low'
WHEN revenue_per_store.msa_income BETWEEN 20001 AND 30000 THEN 'med-low'
WHEN revenue_per_store.msa_income BETWEEN 30001 AND 40000 THEN 'med-high'
WHEN revenue_per_store.msa_income BETWEEN 40001 AND 60000 THEN 'high'
END as income_group
FROM (SELECT m.msa_income, t.store,
CASE when extract(year from t.saledate) = 2005 AND extract(month from t.saledate) = 8 then 'exclude'
END as exclude_flag, SUM(t.amt) AS revenue, COUNT(DISTINCT t.saledate) as numdays, EXTRACT(MONTH from
t.saledate) as monthID
FROM store_msa m JOIN trnsact t
ON m.store=t.store
WHERE t.stype='P' AND exclude_flag IS NULL AND t.store||EXTRACT(YEAR from t.saledate)||EXTRACT(MONTH from
t.saledate) IN(SELECT store||EXTRACT(YEAR from saledate)||EXTRACT(MONTH from saledate)
FROM trnsact
GROUP BY store, EXTRACT(YEAR from saledate), EXTRACT(MONTH from saledate)
HAVING COUNT(DISTINCT saledate)>= 20)
GROUP BY t.store, m.msa_income, monthID, exclude_flag) AS revenue_per_store
GROUP BY income_group
ORDER BY avg_group_revenue;
```

Incorrect Feedback: Make sure you are excluding the appropriate data, and calculating average daily revenue by summing together all the revenue associated with an income group, and then dividing that summed total by the total number of sales days that contributed to that particular summed revenue total. Do not compute averages of averages.

Question 9: Divide stores up so that stores with msa populations between 1 and 100,000 are labeled 'very small', stores with msa populations between 100,001 and 200,000 are labeled 'small', stores with msa populations between 200,001 and 500,000 are labeled 'med_small', stores with msa populations between 500,001 and 1,000,000 are labeled 'med_large', stores with msa populations between 1,000,001 and 5,000,000 are labeled "large", and stores with msa_incomes greater than 5,000,000 are labeled "very large". What is the average daily revenue (as defined in Teradata Week 5 Exercise Guide) for a store in a "very large" population msa?

A. $6,298
B. $16,355
C. $24,341
D. $25,452

Correct Answer: D

Correct Feedback: The query you used to arrive at your answer might have looked like this:

```
SELECT SUM(store_rev. tot_sales)/SUM(store_rev.numdays) AS daily_avg,
CASE WHEN store_rev.msa_pop BETWEEN 1 AND 100000 THEN 'very small'
WHEN store_rev.msa_pop BETWEEN 100001 AND 200000 THEN 'small'
WHEN store_rev.msa_pop BETWEEN 200001 AND 500000 THEN 'med_small'
WHEN store_rev.msa_pop BETWEEN 500001 AND 1000000 THEN 'med_large'
WHEN store_rev.msa_pop BETWEEN 1000001 AND 5000000 THEN 'large'
WHEN store_rev.msa_pop > 5000000 then 'very large'
END as pop_group
FROM(SELECT COUNT (DISTINCT t.saledate) as numdays, EXTRACT(YEAR from t.saledate) as s_year, EXTRACT(MONTH
from t.saledate) as s_month, t.store, sum(t.amt) AS tot_sales,
CASE when extract(year from t.saledate) = 2005 AND extract(month from t.saledate) = 8 then 'exclude'
END as exclude_flag, m.msa_pop
FROM trnsact t JOIN store_msa m
ON m.store=t.store
WHERE t.stype = 'P' AND exclude_flag IS NULL
GROUP BY s_year, s_month, t.store, m.msa_pop
HAVING numdays >= 20) as store_rev
GROUP BY pop_group
ORDER BY daily_avg;
```

Incorrect Feedback: Make sure you are excluding the appropriate data, and calculating average daily revenue by summing together all the revenue associated with a population group, and then dividing that summed total by the total number of sales days that contributed to that particular summed revenue total. Do not compute averages of averages.

Question 10: Which department in which store had the greatest percent increase in average daily sales revenue from November to December, and what city and state was that store located in? Only examine departments whose total sales were at least $1,000 in both November and December.

A. Louisvl department, Salina, KS
B. Clinique department, Odessa, TX
C. Jacques department, Jackson, MS
D. Gottex department, Pine Bluff, AR

Correct Answer: A

Correct Feedback: The query you used to arrive at your answer might have looked like this:

```
SELECT s.store, s.city, s.state, d.deptdesc, sum(case when extract(month from saledate)=11 then amt
end) as November,
COUNT(DISTINCT (case WHEN EXTRACT(MONTH from saledate) ='11' then saledate END)) as Nov_numdays,
sum(case when extract(month from saledate)=12 then amt end) as December,
COUNT(DISTINCT (case WHEN EXTRACT(MONTH from saledate) ='12' then saledate END)) as Dec_numdays,
 ((December/Dec_numdays)-(November/Nov_numdays))/(November/Nov_numdays)*100 AS bump
FROM trnsact t JOIN strinfo s
ON t.store=s.store JOIN skuinfo si
ON t.sku=si.sku JOIN deptinfo d
ON si.dept=d.dept
WHERE t.stype='P' and t.store||EXTRACT(YEAR from t.saledate)||EXTRACT(MONTH from t.saledate) IN
(SELECT store||EXTRACT(YEAR from saledate)||EXTRACT(MONTH from saledate)
FROM trnsact
GROUP BY store, EXTRACT(YEAR from saledate), EXTRACT(MONTH from saledate)
HAVING COUNT(DISTINCT saledate)>= 20)
GROUP BY s.store, s.city, s.state, d.deptdesc
HAVING November > 1000 AND December > 1000
ORDER BY bump DESC;
```

Incorrect Feedback: Consider using a CASE statement within an aggregate function to arrive at your answer.

Question 11. Which department within what store had the greatest decrease in average daily sales revenue from August to September, and what city and state was that store located in?

A. Clinique department, Cincinnati, OH
B. Clinique department, Louisville, KY
C. Polomen department, Knoxville, TN
D. Polomen department, Greenville, SC

Correct Answer: B

Correct Feedback: The query you used to arrive at your answer might have looked like this:

```
SELECT s.city, s.state, d.deptdesc, t.store,
CASE when extract(year from t.saledate) = 2005 AND extract(month from t.saledate) = 8 then 'exclude'
END as exclude_flag,
SUM(case WHEN EXTRACT(MONTH from saledate) =’8’ THEN amt END) as August,
SUM(case WHEN EXTRACT(MONTH from saledate) =’9’ THEN amt END) as September,
COUNT(DISTINCT (case WHEN EXTRACT(MONTH from saledate) ='8' then saledate END)) as Aug_numdays,
COUNT(DISTINCT (case WHEN EXTRACT(MONTH from saledate) ='9' then saledate END)) as Sept_numdays,
(August/Aug_numdays)-(September/Sept_numdays) AS dip
FROM trnsact t JOIN strinfo s
ON t.store=s.store JOIN skuinfo si
ON t.sku=si.sku JOIN deptinfo d
ON si.dept=d.dept WHERE t.stype='P' AND exclude_flag IS NULL AND t.store||EXTRACT(YEAR from
t.saledate)||EXTRACT(MONTH from t.saledate) IN (SELECT store||EXTRACT(YEAR from saledate)||EXTRACT(MONTH
from saledate)
FROM trnsact
GROUP BY store, EXTRACT(YEAR from saledate), EXTRACT(MONTH from saledate)
HAVING COUNT(DISTINCT saledate)>= 20)
GROUP BY s.city, s.state, d.deptdesc, t.store, exclude_flag
```

ORDER BY dip DESC;

Incorrect Feedback: Make sure you are excluding data from August 2005.

Question 12: Identify which department, in which city and state of what store, had the greatest decrease in number of items sold from August to September. How many fewer items did that department sell in September compared to August?

A. **The Clinique department in Louisville, KY sold 13,491 fewer items**
B. The Clinique department in Greenville, SC sold 18,553 fewer items
C. **The R Lauren department in Toledo, OH sold 12,009 fewer items**
D. The R Lauren department in Charlotte, NC sold 5,856 fewer items

Correct Answer: A

Correct Feedback: There are several possible queries that could have given you the right answer, one of which is:

```
SELECT s.city, s.state, d.deptdesc, t.store,

CASE when extract(year from t.saledate) = 2005 AND extract(month from t.saledate) = 8 then 'exclude'

END as exclude_flag,

SUM(case WHEN EXTRACT(MONTH from saledate) = 8 then t.quantity END) as August,

SUM(case WHEN EXTRACT(MONTH from saledate) = 9 then t.quantity END) as September, August-September AS dip

FROM trnsact t JOIN strinfo s

ON t.store=s.store JOIN skuinfo si

ON t.sku=si.sku JOIN deptinfo d

ON si.dept=d.dept

WHERE t.stype='P' AND exclude_flag IS NULL AND

t.store||EXTRACT(YEAR from t.saledate)||EXTRACT(MONTH from t.saledate) IN

(SELECT store||EXTRACT(YEAR from saledate)||EXTRACT(MONTH from saledate)

FROM trnsact

GROUP BY store, EXTRACT(YEAR from saledate), EXTRACT(MONTH from saledate)

HAVING COUNT(DISTINCT saledate)>= 20)

GROUP BY s.city, s.state, d.deptdesc, t.store, exclude_flag
```

```
ORDER BY dip DESC;
```

Incorrect Feedback: Make sure you are excluding data from August 2005.

Question 13: For each store, determine the month with the minimum average daily revenue (as I define it in Teradata Week 5 Exercise Guide) . For each of the twelve months of the year, count how many stores' minimum average daily revenue was in that month. During which month(s) did over 100 stores have their minimum average daily revenue?

A. **August only**
B. January and September
C. January and August
D. August and September

Correct Answer: A

Correct Feedback: The query you used to arrive at your answer might have looked like this:

```
SELECT CASE when max_month_table.month_num = 1 then 'January' when
max_month_table.month_num = 2 then 'February' when
max_month_table.month_num = 3 then 'March' when
max_month_table.month_num = 4 then 'April' when
max_month_table.month_num = 5 then 'May' when
max_month_table.month_num = 6 then 'June' when
max_month_table.month_num = 7 then 'July' when
max_month_table.month_num = 8 then 'August' when
max_month_table.month_num = 9 then 'September' when
max_month_table.month_num = 10 then 'October' when
max_month_table.month_num = 11 then 'November' when
max_month_table.month_num = 12 then 'December' END, COUNT(*)
FROM (SELECT DISTINCT extract(year from saledate) as year_num, extract(month from saledate) as month_num,
CASE when extract(year from saledate) = 2005 AND extract(month from saledate) = 8 then 'exclude END as exclude_flag,
store, SUM(amt) AS tot_sales, COUNT (DISTINCT saledate) as numdays,  tot_sales/numdays as dailyrev, ROW_NUMBER
() over (PARTITION BY store ORDER BY dailyrev DESC) AS month_rank
FROM trnsact
WHERE stype='P' AND exclude_flag IS NULL AND store||EXTRACT(YEAR from   saledate)||EXTRACT(MONTH from
saledate) IN (SELECT store||EXTRACT(YEAR from saledate)||EXTRACT(MONTH from saledate)
FROM trnsact
GROUP BY store, EXTRACT(YEAR from saledate), EXTRACT(MONTH from saledate)
HAVING COUNT(DISTINCT saledate)>= 20)
GROUP BY store, month_num, year_num
HAVING numdays>=20 QUALIFY month_rank=12) as max_month_table
GROUP BY max_month_table.month_num
ORDER BY max_month_table.month_num;
```

Incorrect Feedback: Make sure you only examine purchases (not returns), exclude all stores with fewer than 20 days of data, and exclude all data from August 2005 (as instructed in the Teradata Week 5 Exercise Guide).

Question 14: Write a query that determines the month in which each store had its maximum number of sku units returned. During which month did the greatest number of stores have their maximum number of sku units returned?

A. January
B. March
C. September
D. December

Correct Answer: D

Correct Feedback: There are several possible queries that would have arrived at the right answers such as this query:

```
SELECT CASE when max_month_table.month_num = 1 then 'January' when
max_month_table.month_num = 2 then 'February' when
max_month_table.month_num = 3 then 'March' when
max_month_table.month_num = 4 then 'April' when
max_month_table.month_num = 5 then 'May' when
max_month_table.month_num = 6 then 'June' when
max_month_table.month_num = 7 then 'July' when
max_month_table.month_num = 8 then 'August' when
max_month_table.month_num = 9 then 'September' when
max_month_table.month_num = 10 then 'October' when
max_month_table.month_num = 11 then 'November' when
max_month_table.month_num = 12 then 'December' END, COUNT(*)
FROM (SELECT DISTINCT extract(year from saledate) as year_num, extract(month from saledate) as month_num,
CASE when extract(year from saledate) = 2004 AND extract(month from saledate) = 8 then 'exclude' END as
exclude_flag, store, SUM(quantity) AS tot_returns, ROW_NUMBER () over (PARTITION BY store ORDER BY tot_returns
DESC) AS month_rank
FROM trnsact
WHERE stype='R' AND exclude_flag IS NULL AND store||EXTRACT(YEAR from   saledate)||EXTRACT(MONTH from
saledate) IN (SELECT store||EXTRACT(YEAR from saledate)||EXTRACT(MONTH from saledate)
FROM trnsact
GROUP BY store, EXTRACT(YEAR from saledate), EXTRACT(MONTH from saledate)
HAVING COUNT(DISTINCT saledate)>= 20)
GROUP BY store, month_num, year_num
QUALIFY month_rank=1) as max_month_table
GROUP BY max_month_table.month_num
ORDER BY max_month_table.month_num
```

Incorrect Feedback: Make sure you only examine purchases (not returns), exclude all stores with less than 20 days of data, and exclude all data from August 2005.