1.

SOURCE:

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **C** |
| 1 | 0 | NULL |
| 0 | NULL | NULL |
| NULL | NULL | NULL |

TARGET:

|  |  |  |
| --- | --- | --- |
| A | B | C |
| 1 | 2 | 3 |

Ans:

select

sum( case when A IS NULL THEN 1 else 0 end) AS A,

sum( case when B is NULL THEN 1 ELSE 0 END) AS B,

SUM( CASE WHEN C IS NULL THEN 1 ELSE 0 END) As C

from TAB\_A

2. What is output after joining TAB\_B and TAB\_C.

TAB\_B

|  |
| --- |
| A |
| 1 |
| 0 |
| NULL |

TAB\_C

|  |
| --- |
| B |
| 0 |
| 0 |
| NULL |

OUTPUT:

select a.\*,b.\* from TAB\_B a join TAB\_C b ON a.A=b.B

|  |  |
| --- | --- |
| **A** | **B** |
| 0 | 0 |
| 0 | 0 |

3. Write a query to find the count of specific character in a word.

SELECT

'banana' AS word,

LENGTH('banana') - LENGTH(REPLACE('banana', 'a', '')) AS count\_of\_a;

4.

|  |  |
| --- | --- |
| **TAB\_D** |  |
| **C1** | **C2** |
| 1 | 2 |
| 2 | 3 |
| 3 | 4 |
|  |  |
| **TAB\_E** |  |
| **C1** | **C2** |
| 1 | 2 |
| 2 | 3 |
| 0 | 1 |

**Select TAB\_D.\*,TAB\_E.\* from TAB\_D join TAB\_E**

**on TAB\_D.C1 < TAB\_E.C1**

|  |  |  |  |
| --- | --- | --- | --- |
| **C1** | **C2** | **C1** | **C2** |
| **1** | **2** | **2** | **3** |

**Scenario-2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TAB\_D** | |  | | |
| **C1** | | **C2** | | |
| 1 | | 2 | | |
| 2 | | 3 | | |
| 3 | | 4 | | |
|  | |  | | |
| **TAB\_E** | |  | | |
| **C1** | | **C2** | | |
| 1 | | 2 | | |
| 2 | | 3 | | |
| 0 | | 1 | | |
|  | |  | | |
| **C1** | **C2** | **C1** | **C2** |
| **1** | **2** | **0** | **1** |
| **2** | **3** | **1** | **2** |
| **2** | **3** | **0** | **1** |
| **3** | **4** | **1** | **2** |
| **3** | **4** | **2** | **3** |
| **3** | **4** | **0** | **1** |

**Select TAB\_D.\*,TAB\_E.\* from TAB\_D join TAB\_E**

**on TAB\_D.C1 > TAB\_E.C1**

**scenario-3**

|  |  |
| --- | --- |
| **TAB\_D** |  |
| **C1** | **C2** |
| 1 | 2 |
| 2 | 3 |
| 3 | 4 |
|  |  |
| **TAB\_E** |  |
| **C1** | **C2** |
| 1 | 2 |
| 2 | 3 |
| 0 | 1 |

**Select TAB\_D.\*,TAB\_E.\* from TAB\_D join TAB\_E**

**on TAB\_D.C1 != TAB\_E.C1**

|  |  |  |  |
| --- | --- | --- | --- |
| **C1** | **C2** | **C1** | **C2** |
| **1** | **2** | **2** | **3** |
| **1** | **2** | **0** | **1** |
| **2** | **3** | **1** | **2** |
| **2** | **3** | **0** | **1** |
| **3** | **4** | **1** | **2** |
| **3** | **4** | **2** | **3** |
| **3** | **4** | **0** | **1** |

**4.**

**for each emp Show the % of total revenue before discount generated**

**by orders shipped to the USA and UK with respect to the total revenue generated by that employee**

**orders with: order\_id, employee\_id, ship\_country**

**order\_details with: order\_id, unit\_price, quantity, discount**

**SELECT**

**o.employee\_id,**

**ROUND(**

**SUM(CASE**

**WHEN o.ship\_country IN ('USA', 'UK')**

**THEN od.unit\_price \* od.quantity**

**ELSE 0**

**END**

**) \* 100.0 /**

**SUM(od.unit\_price \* od.quantity),**

**2) AS percent\_revenue**

**FROM**

**orders o**

**JOIN**

**order\_details od ON o.order\_id = od.order\_id**

|  |
| --- |
| source Table |
| Name City |
| Shyam Mumbai |
| Shyam Bangalore |
| Ram Delhi |
| Ram Pune |
| Ram Chennai |
|  |
| Target Table |
| Name City |
| Shyam Mumbai,Bangalore |
| Ram Delhi,Pune,Chennai |

**6. Snowflake SCD2.**

**create or replace procedure SCD2()**

**returns string**

**language sql**

**AS**

**$$**

**BEGIN**

**MERGE INTO DIM\_PRODUCT AS T USING STG\_PRODUCT AS S**

**ON T.PRODUCT\_ID=S.PRODUCT\_ID**

**WHEN MATCHED AND (S.PRODUCT\_NAME<> T.PRODUCT\_NAME OR S.CATEGORY<>T.CATEGORY) THEN**

**UPDATE SET T.END\_DATE=CURRENT\_DATE,T.IS\_CURRENT='NO'**

**WHEN NOT MATCHED THEN**

**INSERT VALUES(S.product\_id,S.product\_name,S.category,S.price,S.updated\_date,'9999-12-31','YES');**

**INSERT INTO DIM\_PRODUCT(PRODUCT\_ID,PRODUCT\_NAME,CATEGORY,PRICE,START\_DATE,END\_DATE)**

**SELECT A.PRODUCT\_ID,A.PRODUCT\_NAME,A.CATEGORY,A.PRICE,A.UPDATED\_DATE,**

**'9999-12-31' AS END\_DATE FROM STG\_PRODUCT A LEFT JOIN DIM\_PRODUCT B ON A.PRODUCT\_ID=B.PRODUCT\_ID**

**and is\_current='TRUE' WHERE b.PRODUCT\_ID is NULL or (A.PRODUCT\_NAME != B.PRODUCT\_NAME);**

**END;**

**$$**

**7.**

**Table structure- T1(C1 NUMBER) and T2(c2 NUMBER)**

**Table Values - T1(1**

**,2,**

**NULL) ---3 ROWS**

**T2(1**

**,1**

**,NULL,**

**NULL) ---4 ROWS**

**How many rows we get as output if we do Inner-2 rows , Left - 3 rows, Right-2 rows and Full Join-3 rows.**

**ANS:**

**TAB\_F**

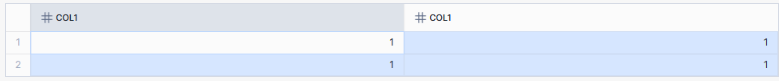
|  |
| --- |
| COL1 |
| 1 |
| 2 |
| NULL |

**TAB\_G**

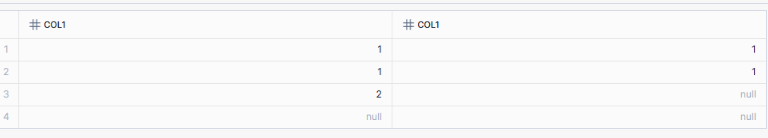
|  |
| --- |
| COL1 |
| 1 |
| 1 |
| NULL |
| NULL |

**INNER JOIN**

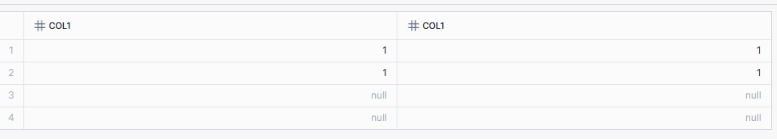
**select a.\*,b.\* from TAB\_F a join TAB\_G b on a.col1=b.col1**

****

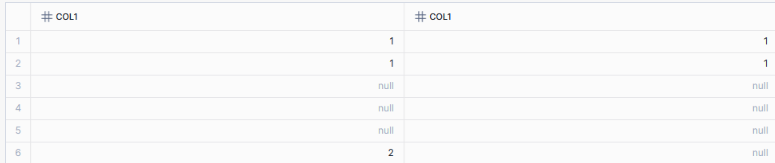
**LEFT JOIN**

****

**RIGHT JOIN**

****

**FULL OUTER JOIN**

****

**8.**

**How to generate the Monthly report for business users using SQL query, based on below table**

**Emp(Empid number, salary number, salaryDate Date)**

**Primary Key is EmpId and SalaryDate**

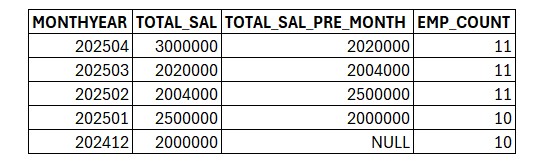
**Output :-**

**MonthYear in YYYYMM format**

**Sum of Total Salary for all employees**

**Sum of Total Salary for Previous Month**

**Count of employees for each month**



**9. Find the duplicate records emp table only based on email\_id.**

**ANS:**

**select \*, count(\*) over(partition by email\_id) as cnt from emp\_1**

**10.**