```
Case :-

An Expression is used to evaluate a clause to return values. There are different SQL expressions
that can be used in HANA –
Case Expressions
Function Expressions
Aggregate Expressions
Subqueries in Expressions

CASE :- This is used to pass multiple conditions in a SQL expression. It allows the use of IF-
ELSE-THEN logic without using procedures in SQL statements.

SELECT COUNT( CASE WHEN sal < 2000 THEN 1 ELSE NULL END ) count1,
COUNT( CASE WHEN sal BETWEEN 2001 AND 4000 THEN 1 ELSE NULL END ) count2,
COUNT( CASE WHEN sal > 4000 THEN 1 ELSE NULL END ) count3 FROM emp;
-----------------------------------------------------------------------------------------------
Function Expressions
Function expressions involve SQL inbuilt functions to be used in Expressions.
------------------------------------------------------------------------------------------------
Aggregate Expressions
Aggregate functions are used to perform complex calculations like Sum, Percentage, Min, Max,
Count, Mode, Median, etc. Aggregate Expression uses Aggregate functions to calculate single value
from multiple values.
Aggregate Functions – Sum, Count, Minimum, Maximum. These are applied on measure values (facts)
and It is always associated with a dimension. An aggregate expression uses an aggregate function
to calculate a single value from the values of multiple rows in one or more columns
Common aggregate functions include –
Average ()
Count ()
Maximum ()
Median ()
Minimum ()
Mode ()
Sum ()

SubQuery Expression
A subquery as an expression is a Select statement. When it is used in an expression, it returns a
zero or a single value.
A subquery is used to return data that will be used in the main query as a condition to further
restrict the data to be retrieved.
Subqueries can be used with the SELECT, INSERT, UPDATE, and DELETE statements along with the
operators like =, <, >, >=, <=, IN, BETWEEN etc.
There are a few rules that subqueries must follow –
Subqueries must be enclosed within parentheses.
A subquery can have only one column in the SELECT clause, unless multiple columns are in the main
query for the subquery to compare its selected columns.
An ORDER BY cannot be used in a subquery, although the main query can use an ORDER BY. The GROUP
BY can be used to perform the same function as the ORDER BY in a subquery.
Subqueries that return more than one row can only be used with multiple value operators, such as
the IN operator.

Example :-
SELECT * FROM CUSTOMERS
WHERE ID IN (SELECT ID
FROM CUSTOMERS


Window Function
<window_function> ::=
 <window_function_name_or_type>
 OVER ( [ <window_partition_by_clause> ] [ <window_order_by_clause> ] [ <window_frame_clause> ] )

he window functions allow you to divide the result sets of a query, or a logical partition of a
query, into groups of rows called window partitions.


TRIGGERS :-
Triggers in SAP HANA
This statement is used to triggers which is a set of statements that are executed when a given
```

operation (INSERT/UPDATE/DELETE) takes place on a given subject table or subject view.
A trigger is also a stored procedure that automatically executes when an event happens on a given table or view.
The database users only having the TRIGGER privilege for the given &lt;subject_table_name>; are allowed to create a trigger for that table or view.

```
Create trigger “KABIL_PRACTICE”.”SALES_TRIGGER”
after insert on “KABIL_PRACTICE”.”SALES” REFERENCING NEW ROW AS newrow for each row
begin
update “KABIL_PRACTICE”.”Inventory” set “Inventory” = “Inventory” – :newrow.QTY
where “P_ID” = :newrow.P_ID ;
end;
```

DEFINER • DEFINER - Specifies that the execution of the procedure is performed with the privileges of the definer of the procedure.
INVOKER - Specifies that the execution of the procedure is performed with the privileges of the invoker of the procedure.

```
CREATE PROCEDURE <proc_name> [(<parameter_clause>)]
[LANGUAGE <lang>] [SQL SECURITY <mode>] [DEFAULT SCHEMA <default_schema_name>]
[READS SQL DATA [WITH RESULT VIEW <view_name>]] AS { BEGIN [SEQUENTIAL EXECUTION] <procedure_body>
END | HEADER ONLY }
```