```
use role developer;
use database DEBADATTA_MOHANTY_DB;
create schema information_schema_debadatta;
use schema information_schema_debadatta;

create table customer as select * from training_schema_core.customer;
create table lineitem as select * from training_schema_core.lineitem;
create table nation as select * from training_schema_core.nation;
create table orders as select * from training_schema_core.orders;
create table part as select * from training_schema_core.part;
create table partsupp as select * from training_schema_core.partsupp;
create table region as select * from training_schema_core.region;
create table supplier as select * from training_schema_core.supplier;
/*
Simple JAVASCRIPT Function
*/
--

CREATE OR REPLACE FUNCTION UDF_NOWASSTRING()
   RETURNS Date
   LANGUAGE JAVASCRIPT
   AS
   $$
   return Date.now().toString();
   $$;

select get_ddl('function','UDF_NOWASSTRING()');

SELECT UDF_NOWASSTRING();
```

```
CREATE OR REPLACE FUNCTION all_props() returns string
  LANGUAGE JAVASCRIPT
AS
$$
  return Object.getOwnPropertyNames(this);
$$
;

select all_props();



/* simple Query see if its working
*/
select
      l_returnflag,
      l_linestatus,
      sum(l_quantity) as sum_qty,
      sum(l_extendedprice) as sum_base_price,
      sum(l_extendedprice * (1-l_discount)) as sum_disc_price,
      sum(l_extendedprice * (1-l_discount) * (1+l_tax)) as sum_charge,
      avg(l_quantity) as avg_qty,
      avg(l_extendedprice) as avg_price,
      avg(l_discount) as avg_disc,
      count(*) as count_order
 from
      lineitem
 where
      l_shipdate <= dateadd(day, -90, to_date('1998-12-01'))
```

```
 group by
      l_returnflag,
      l_linestatus
 order by
      l_returnflag,
      l_linestatus;



/* we will take in-line calculation to UDF
*/

-- Function with three parameters
CREATE OR REPLACE FUNCTION f_discount_line (extendedprice number,discount number,tax number)
RETURNS NUMBER
LANGUAGE SQL
as
$$
select (extendedprice * (1-discount) * (1+tax))
$$
;

/* -- Function with two parameters
*/

CREATE OR REPLACE FUNCTION f_discount_line (extendedprice number,discount number)
RETURNS NUMBER
AS
$$
select (extendedprice * (1-discount))
```

```
$$
;

show user functions;

/* replace the query calculation with fucnitons
*/
select
    l_returnflag,
    l_linestatus,
    sum(l_quantity) as sum_qty,
    sum(l_extendedprice) as sum_base_price,
    sum(f_discount_line(l_extendedprice,l_discount)) as sum_disc_price,
    sum(f_discount_line(l_extendedprice,l_discount,l_tax)) as sum_charge,
    avg(l_quantity) as avg_qty,
    avg(l_extendedprice) as avg_price,
    avg(l_discount) as avg_disc,
    count(*) as count_order
from
    lineitem
where
    l_shipdate <= dateadd(day, -90, to_date('1998-12-01'))
group by
    l_returnflag,
    l_linestatus
order by
    l_returnflag,
    l_linestatus;
```

```
select get_ddl('function','f_discount_line(number,number,number)');


/* Look at the profiling for any change

*/


/*****************************create secure funcitons
*******************************************/



-- Function with three parameters

CREATE OR REPLACE SECURE FUNCTION f_discount_line (extendedprice number,discount number,tax
number)

RETURNS NUMBER

LANGUAGE SQL

as

$$

select (extendedprice * (1-discount) * (1+tax))

$$

;


/* -- Function with two parameters

*/


CREATE OR REPLACE FUNCTION f_discount_line (extendedprice number,discount number)

RETURNS NUMBER

AS

$$

select (extendedprice * (1-discount))

$$
```

```
;



/* drop requires to provide parameters even when not overloaded

*/

drop function f_discount_line(number ,number ,number );


drop function f_discount_line;-- not allowed


show user functions;

drop function f_discount_line(number ,number);


---*****************************User Define Table Function ***************************

create or replace  function get_return_summary_for_year ( Vyear number )

returns table (return_flag varchar, line_status varchar, sum_qty number, sum_base_price number
,sum_disc_pcount_orderrice number,sum_charge number,avg_qty number,avg_price number, avg_disc
number, count_order number)

as

$$

select

    l_returnflag,

    l_linestatus,

    sum(l_quantity) as sum_qty,

    sum(l_extendedprice) as sum_base_price,

    sum(l_extendedprice * (1-l_discount)) as sum_disc_price,

    sum(l_extendedprice * (1-l_discount) * (1+l_tax)) as sum_charge,

    avg(l_quantity) as avg_qty,

    avg(l_extendedprice) as avg_price,

    avg(l_discount) as avg_disc,
```

```
    count(*) as count_order
from
    lineitem
where
    extract(year from l_shipdate) = Vyear
group by
    l_returnflag,
    l_linestatus
$$
;


create or replace  function get_return_summary_for_year ( Vyear number,vmonth number )
returns table (return_flag varchar, line_status varchar, sum_qty number, sum_base_price number
,sum_disc_pcount_orderrice number,sum_charge number,avg_qty number,avg_price number, avg_disc
number, count_order number)
as
$$
select
    l_returnflag,
    l_linestatus,
    sum(l_quantity) as sum_qty,
    sum(l_extendedprice) as sum_base_price,
    sum(l_extendedprice * (1-l_discount)) as sum_disc_price,
    sum(l_extendedprice * (1-l_discount) * (1+l_tax)) as sum_charge,
    avg(l_quantity) as avg_qty,
    avg(l_extendedprice) as avg_price,
    avg(l_discount) as avg_disc,
    count(*) as count_order
from
```

```
    lineitem
 where
    extract(year from l_shipdate) = Vyear and
    extract(month from l_shipdate) = vmonth
 group by
    l_returnflag,
    l_linestatus
$$
;



select * from table(get_return_summary_for_year(1996));



select * from table(get_return_summary_for_year(1996,11));
```

----------********** Lab **********

-- Create function based on below query **********

/**********Business Query**********

Determine whether selecting less expensive modes of shipping is negatively affecting the critical-priority orders by causing more parts to be received by customers after the committed date.

The Shipping Modes and Order Priority Query counts, by ship mode, for lineitems actually received by customers in a given year, the number of lineitems belonging to orders for which the l_receiptdate exceeds the l_commitdate for two different specified ship modes. Only lineitems that were actually shipped before the l_commitdate are considered. The late lineitems are partitioned into two groups, those with priority URGENT or HIGH, and those with a priority other than URGENT or HIGH.

```
*/

SELECT
   l_shipmode,
   sum(case
      when o_orderpriority = '1-URGENT'
         OR o_orderpriority = '2-HIGH'
         then 1
      else 0
   end) as high_line_count,
   sum(case
      when o_orderpriority <> '1-URGENT'
         AND o_orderpriority <> '2-HIGH'
         then 1
      else 0
   end) AS low_line_count
FROM
   orders,
   lineitem
WHERE
   o_orderkey = l_orderkey
   AND l_shipmode in ('MAIL', 'SHIP')
   AND l_commitdate < l_receiptdate
   AND l_shipdate < l_commitdate
   AND l_receiptdate >= to_date('1994-01-01')
   AND l_receiptdate < dateadd(year, 1, to_date('1994-01-01'))
GROUP BY
   l_shipmode
ORDER BY
```

```
    l_shipmode;



################################## JAVA ##################################


create or replace function echo_varchar(x varchar)
returns varchar
language java
called on null input
handler='TestFunc.echo_varchar'
target_path='@~/testfunc.jar'
as
'class TestFunc {
  public static String echo_varchar(String x) {
    return x;
  }
}';

select echo_varchar('Hello');
```