# Task C: Functions

## 1. \*\*Scalar Function: `fn\_customer\_lifetime\_value(customer\_id)`\*\*     - Return total \*\*paid\*\* amount for the customer's delivered/shipped/placed (non-cancelled) orders.

**CREATE** **OR** **REPLACE** **FUNCTION** training\_ecom.fn\_customer\_lifetime\_value(p\_customer\_id **INT**)

**RETURNS** **NUMERIC** **AS** **$$**

**BEGIN**

**RETURN** (

**SELECT** **COALESCE**(**SUM**(p.amount), 0)

**FROM** training\_ecom.payments p

**JOIN** training\_ecom.orders o **ON** p.order\_id = o.order\_id

**WHERE** o.customer\_id = p\_customer\_id

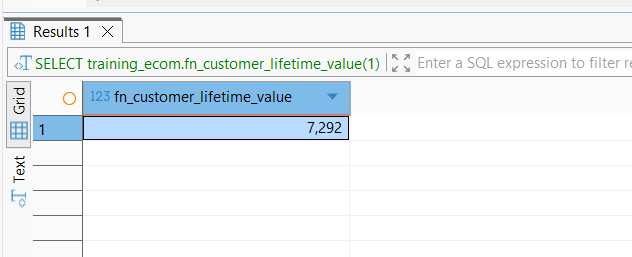
**AND** o.status **IN** ('placed', 'shipped', 'delivered')

);

**END**;

**$$** **LANGUAGE** plpgsql;

**SELECT** training\_ecom.**fn\_customer\_lifetime\_value**(1);



**2. \*\*Table Function: `fn\_recent\_orders(p\_days INT)`\*\*  
    - Return `order\_id, customer\_id, order\_date, status, order\_total` for orders in the last `p\_days` days.**

**CREATE** **OR** **REPLACE** **FUNCTION** training\_ecom.fn\_recent\_orders(p\_days **INT**)

**RETURNS** **TABLE** (

order\_id **INT**,

customer\_id **INT**,

order\_date **TIMESTAMP**,

status **VARCHAR**,

order\_total **NUMERIC**

) **AS** **$$**

**BEGIN**

**RETURN** QUERY

**SELECT**

o.order\_id,

o.customer\_id,

o.order\_date,

o.status,

**SUM**(oi.quantity \* oi.unit\_price) **AS** order\_total

**FROM** training\_ecom.orders o

**JOIN** training\_ecom.order\_items oi **ON** o.order\_id = oi.order\_id

**WHERE** o.order\_date >= CURRENT\_DATE - **INTERVAL** '1 day' \* p\_days

**AND** o.status != 'cancelled'

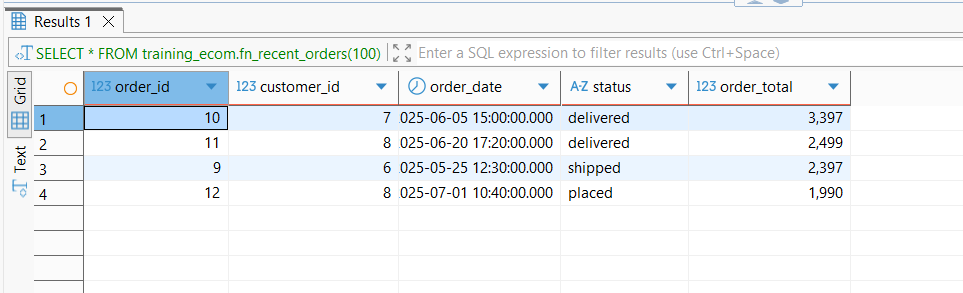
**GROUP** **BY** o.order\_id, o.customer\_id, o.order\_date, o.status;

**END**;

**$$** **LANGUAGE** plpgsql;

**SELECT** \*

**FROM** training\_ecom.**fn\_recent\_orders**(100);



**3. \*\*Utility Function: `fn\_title\_case\_city(text)`\*\*  
    - Return city name with first letter of each word capitalized (hint: split/upper/lower or use `initcap()` in PostgreSQL).**

**CREATE** **OR** **REPLACE** **FUNCTION** training\_ecom.fn\_title\_case\_city(p\_city **TEXT**)

**RETURNS** **TEXT** **AS** **$$**

**BEGIN**

**RETURN** **initcap**(p\_city);

**END**;

**$$** **LANGUAGE** plpgsql;

**SELECT** training\_ecom.**fn\_title\_case\_city**('kathmandu');

