# Task B: Views & Subqueries

## 1. \*\*Create View: `vw\_recent\_orders\_30d`\*\*    - View of orders placed in the \*\*last 30 days\*\* from `CURRENT\_DATE`, excluding `cancelled`.    - Columns: order\_id, customer\_id, order\_date, status, order\_total (sum of items).

**CREATE** **OR** **REPLACE** VIEW training\_ecom.vw\_recent\_orders\_30d **AS**

**SELECT**

o.order\_id,

o.customer\_id,

o.order\_date,

o.status,

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

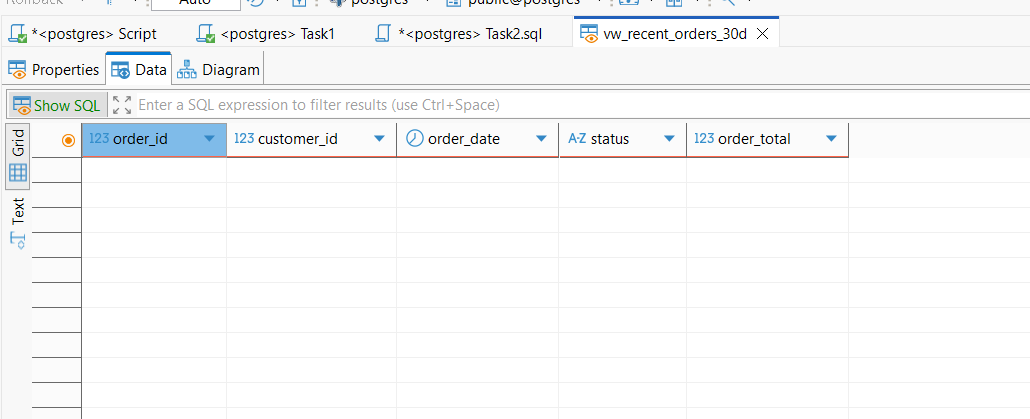
**FROM** training\_ecom.orders o

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

**WHERE** o.order\_date >= CURRENT\_DATE - **INTERVAL** '30 days'

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

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



**2. \*\*Products Never Ordered\*\*  
   - Using a subquery, list products that \*\*never\*\* appear in `order\_items`.**

**SELECT**

*p*.product\_id,

*p*.product\_name,

*p*.category,

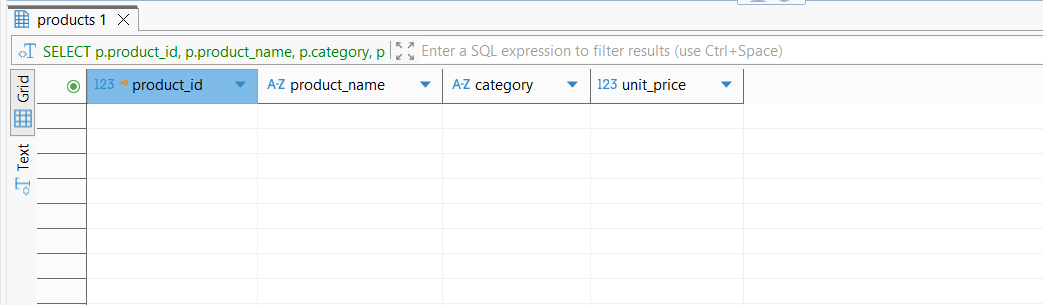
*p*.unit\_price

**FROM** training\_ecom.products *p*

**WHERE** *p*.product\_id **NOT** **IN** (

**SELECT** **DISTINCT** product\_id **FROM** training\_ecom.order\_items

);



## 3. \*\*Top Category by City\*\*    - For each `city`, find the \*\*single category\*\* with the highest total revenue. Use an inner subquery or a view plus a filter on rank.

**SELECT**

city,

category,

*total\_revenue*

**FROM** (

**SELECT**

*c*.city,

*p*.category,

**SUM**(*oi*.quantity \* *oi*.unit\_price) **AS** *total\_revenue*,

**RANK**() **OVER** (

**PARTITION** **BY** c.city

**ORDER** **BY** **SUM**(oi.quantity \* oi.unit\_price) **DESC**

) **AS** *revenue\_rank*

**FROM** training\_ecom.customers *c*

**JOIN** training\_ecom.orders *o* **ON** *c*.customer\_id = *o*.customer\_id

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

**JOIN** training\_ecom.products *p* **ON** *oi*.product\_id = *p*.product\_id

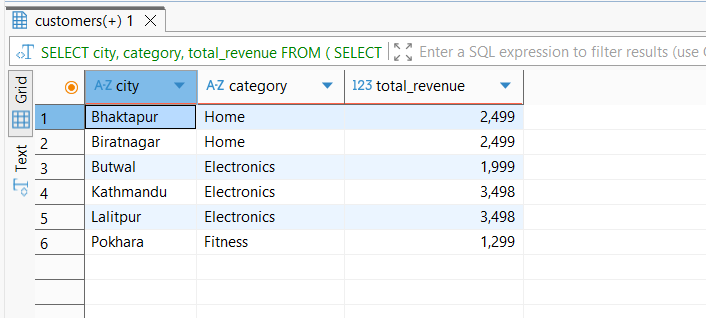
**WHERE** *o*.status != 'cancelled'

**GROUP** **BY** *c*.city, *p*.category

) **AS** *city\_category\_revenue*

**WHERE** *revenue\_rank* = 1

**ORDER** **BY** city;



## 4. \*\*Customers Without Delivered Orders\*\*    - Using `NOT EXISTS`, list customers who have \*\*no orders\*\* with status `delivered`.

**SELECT**

*c*.customer\_id,

*c*.full\_name

**FROM** training\_ecom.customers *c*

**WHERE** **NOT** **EXISTS** (

**SELECT** 1

**FROM** training\_ecom.orders *o*

**WHERE** *o*.customer\_id = *c*.customer\_id

**AND** *o*.status = 'delivered'

)

**ORDER** **BY** *c*.customer\_id;

