3.4.1.

CREATE TABLE product\_groups(

group\_id INT PRIMARY KEY,

group\_name VARCHAR(255) NOT NULL

);

CREATE TABLE products(

product\_id INT PRIMARY KEY,

product\_name VARCHAR(255) NOT NULL,

price DECIMAL(11, 2),

group\_id INT NOT NULL,

FOREIGN KEY (group\_id) REFERENCES product\_groups (group\_id)

);

INSERT INTO product\_groups (group\_id, group\_name)

VALUES (1, 'Smartphone'), (2, 'Laptop'), (3, 'Tablet');

INSERT INTO products (product\_id, product\_name, price, group\_id)

VALUES (1, 'Microsoft Lumia', 200, 1), (2, 'HTC One', 400, 1), (3, 'Nexus', 500, 1), (4, 'iPhone', 900, 1), (5, 'HP Lite', 1200, 2), (6, 'Lenovo Thinkpad', 700, 2), (7, 'Sony VAIO', 700, 2), (8, 'Dell Vostro', 800, 2), (9, 'iPad', 700, 3), (10, 'Kindle Fire', 150, 3), (11, 'Samsung Galaxy Tab', 200, 3);

3.4.2.

select product\_name, group\_name, price, first\_value (price) over (

partition by group\_name

order by price

) as lower\_price\_per\_group

from products

inner join product\_groups using (group\_id);

3.4.3.

select product\_name, group\_name, price, lag (price, 1) over (

partition by group\_name

order by price

) as prev\_price, price - lag (price, 1) over (

partition by group\_name

order by price

) as cur\_prev\_diff

from products

inner join product\_groups using (group\_id);

3.4.4.

select product\_id, product\_name, price, group\_id, nth\_value (product\_name, 2) over (

partition by group\_id

order by price desc

range between unbounded preceding and unbounded FOLLOWING

)

from products;

3.4.5.

select film\_id, title, (case

when length < 30 then 'short'

when length < 90 then 'Medium'

else 'Long'

end

) length from film

)

select film\_id, title, length from cte\_film

where length = 'Medium'

order by title;

3.4.6.

with cte\_film as (

select film\_id, title, rating, length,

RANK() over (

PARTITION by rating

order by length DESC)

length\_rank from film

)

select \*

from cte\_film

where length\_rank = 2;

3.4.7.

SELECT CONCAT(c.first\_name, ' ', c.last\_name) AS name, a.address, a.postal\_code as zip\_code, a.phone, a.district as country, (case when c.active = 0 then ' ' when c.active = 1 then 'active' end) as notes, c.store\_id as sid

FROM customer c

left join address a

on a.address\_id = c.address\_id

left join city

on city.city\_id = a.city\_id;

3.4.8.

create view customer\_master AS

SELECT CONCAT(c.first\_name, ' ', c.last\_name) AS name, a.address, a.postal\_code as zip\_code, a.phone, a.district as country, (case when c.active = 0 then ' ' when c.active = 1 then 'active' end) as notes, c.store\_id as sid

FROM customer c

left join address a

on a.address\_id = c.address\_id

left join city

on city.city\_id = a.city\_id;

select \* from customer\_master;

3.4.9.

CREATE INDEX idx\_adress\_phone

ON address(phone);

explain select (select 223664661973 from address) from address;

Seq Scan on address (cost=14.03..28.06 rows=603 width=8)

3.4.10.

SELECT \* FROM pg\_indexes WHERE tablename = 'customer';