#2602

* SELECT name FROM customers WHERE lower(state) = 'rs'

#2603

* SELECT name, Street FROM customers

WHERE LOWER(city) = 'porto alegre'

#2604:

* SELECT id, name FROM products

WHERE (price < 10) OR (price > 100)

#2605:

* SELECT products.name, providers.name FROM products

INNER JOIN providers ON products.id\_providers = providers.id

INNER JOIN categories ON products.id\_categories = categories.id

WHERE categories.id = 6

#2606:

* SELECT p.id, p.name FROM products AS p

INNER JOIN categories AS c ON p.id\_categories = c.id

WHERE LOWER(c.name) LIKE 'super%'

#2607

* SELECT DISTINCT(city) FROM providers ORDER BY city ASC

#2608:

* SELECT MAX(price), MIN(price) FROM products

#2609:

* SELECT categories.name, SUM(products.amount) FROM products

INNER JOIN categories ON products.id\_categories = categories.id GROUP BY categories.id

ORDER BY categories.name

#2610

* SELECT ROUND(AVG(price), 2) FROM products

#2611:

* SELECT movies.id, movies.name FROM movies

INNER JOIN genres ON movies.id\_genres = genres.id WHERE genres.description = 'Action'

#2613:

* SELECT movies.id, movies.name FROM movies

INNER JOIN prices ON movies.id\_prices = prices.id WHERE prices.value < 2

#2614:

* SELECT customers.name, rentals.rentals\_date FROM rentals

INNER JOIN customers ON rentals.id\_customers = customers.id

WHERE rentals.rentals\_date >= '2016-09-01' AND rentals.rentals\_date < '2016-10-01'

#2615

* SELECT DISTINCT city FROM customers

#2616:

* SELECT customers.id, customers.name FROM customers LEFT JOIN locations

ON customers.id = locations.id\_customers WHERE locations IS NULL

ORDER BY customers.id

#2617:

* SELECT products.name, providers.name FROM products

INNER JOIN providers ON products.id\_providers = providers.id

WHERE providers.name = 'Ajax SA'

#2618:

* SELECT products.name, providers.name, categories.name FROM products

INNER JOIN providers ON products.id\_providers = providers.id

INNER JOIN categories ON products.id\_categories = categories.id

WHERE providers.name = 'Sansul SA' AND categories.name = 'Imported'

#2619:

* SELECT products.name, providers.name, products.price FROM products

INNER JOIN providers ON products.id\_providers = providers.id

INNER JOIN categories ON products.id\_categories = categories.id

WHERE products.price > 1000 AND categories.name = 'Super Luxury'

#2620:

* SELECT customers.name, orders.id FROM orders

INNER JOIN customers ON orders.id\_customers = customers.id

WHERE EXTRACT(MONTH FROM orders.orders\_date) BETWEEN 1 AND 6

AND EXTRACT(YEAR FROM orders.orders\_date) = 2016

#2621:

* SELECT products.name FROM products

INNER JOIN providers ON products.id\_providers = providers.id

WHERE products.amount BETWEEN 10 AND 20 AND providers.name LIKE 'P%'

#2622:

* SELECT customers.name FROM legal\_person INNER JOIN customers

ON legal\_person.id\_customers = customers.id

#2623:

* SELECT products.name, categories.name FROM products

INNER JOIN categories ON products.id\_categories = categories.id

WHERE products.amount > 100 AND categories.id IN (1, 2, 3, 6, 9)

ORDER BY categories.id ASC

#2624:

* SELECT COUNT(DISTINCT city) FROM customers

#2625:

* SELECT SUBSTR(natural\_person.cpf, 1, 3) || '.' ||

SUBSTR(natural\_person.cpf, 4, 3) || '.' ||

SUBSTR(natural\_person.cpf, 7, 3) || '-' ||

SUBSTR(natural\_person.cpf, 10, 2) AS cpf

FROM natural\_person

NNER JOIN customers ON natural\_person.id\_customers = customers.id

#2737:

#2738:

* SELECT candidate.name,

ROUND((((score.math \* 2) + (score.specific \* 3) + (score.project\_plan \* 5)) / 10) , 2) AS avg

FROM score NNER JOIN candidate ON score.candidate\_id = candidate.idORDER BY avg DESC

#2739:

* SELECT name, CAST(EXTRACT(DAY from payday) AS integer) AS day FROM loan

#2740:

#2741:

SELECT 'Approved: ' || name AS name, grade FROM students

WHERE grade >= 7 ORDER BY grade DESC

#2742:

#2743:

* SELECT name, LENGTH(name) as length FROM people ORDER BY length DESC

#2744:

* SELECT id, password, MD5(password) as MD5 FROM account

#2745:

* SELECT name,ROUND((salary \* 0.1), 2) as tax FROM people

WHERE salary > 3000

#2746:

* SELECT REPLACE(name, 'H1', 'X') FROM virus

#2988:

#2989:

#2990:

#2991:

#2992:

#2993:

* SELECT amount as most\_frequent\_value FROM ( SELECT amount, COUNT(amount) AS frequency\_count

FROM value\_table GROUP BY amount ORDER BY frequency\_count DESC

LIMIT 1 ) AS amount\_frequency

#2994:

#2995:

#2996:

* SELECT packages.year, senders.name as sender, receivers.name as receiver

FROM packages INNER JOIN users AS senders ON packages.id\_user\_sender = senders.id

INNER JOIN users AS receivers ON packages.id\_user\_receiver = receivers.id

#2997:

#2998:

#2999:

#3001:

* SELECT name AS type, CASE WHEN type = 'A' THEN 20.0

WHEN type = 'B' THEN 70.0

WHEN type = 'C' THEN 530.5 END AS Price FROM products

ORDER BY products.type, id DESC