# Examples of tasks on SQL:

**1 Show all info about the employee with ID 8.**

SELECT \* FROM person

WHERE id=208

**2 Show the list of first and last names of the employees from London.**

SELECT  person.first\_name, person.last\_name, city.name As cities

From person

Join city ON person.id\_city = city.id

Where role = 'employee'

Group by city.name, person.first\_name, person.last\_name

**3 Show the list of first and last names of the employees whose first name begins with letter A.**

SELECT first\_name, last\_name FROM person WHERE role='employee' AND first\_name LIKE 'A%';

**4 Show the list of first, last names and ages of the employees whose age is greater than 55. The result should be sorted by last name.**

SELECT first\_name, last\_name,EXTRACT(year from age(current\_date, birthday\_date)) as eldery

From person

Where role = 'employee' and EXTRACT(year from age(current\_date, birthday\_date)) >55

order by last\_name;

**5 Calculate the count of employees from London.**

Select Count (person.id), city.name

From person

Join city On city.id = person.id\_city

Where city.name like '%London%' and person.role='employee'

Group by city.name

**6 Calculate the greatest, the smallest and the average age among the employees from London.**

SELECT Max (extract (year from age(current\_date, birthday\_date))) As Max\_Age, Min (extract (year from age(current\_date, birthday\_date))) As Min\_Age, AVG (extract (year from age(current\_date, birthday\_date))) AS Average, city.name AS cities

FROM person

JOIN city ON person.id\_city = city.id

WHERE role = 'employee' AND city.name = 'London'

GROUP BY city.name

**7 Calculate the greatest, the smallest and the average age of the employees for each city.**

SELECT Max (extract (year from age(current\_date, birthday\_date))) As Max\_Age, Min (extract (year from age(current\_date, birthday\_date))) As Min\_Age, AVG (extract (year from age(current\_date, birthday\_date))) As Average,  city.name As cities

From person

Join city ON person.id\_city = city.id

Where role = 'employee'

Group by city.name

**8 Show the list of cities in which the average age of employees is greater than 60 (the average age is also to be shown)**

SELECT AVG (extract(year from age(current\_date, birthday\_date))) As Age, C.name

FROM person As P

JOIN "city" As C ON P.id\_city=C.id

WHERE P.role='employee'

Group BY C.name

HAVING AVG (extract(year from age(current\_date, birthday\_date))) >60

**9 Show the first and last name(s) of the eldest employee(s). Use a subquery.**

Select \* from person where birthday\_date in (

select birthday\_date

from person

order by birthday\_date

limit 1

)

**10 Show first, last names and ages of 3 eldest employees.**

SELECT id, first\_name, last\_name, EXTRACT(year from age(current\_date, birthday\_date)) as eldery from person ORDER BY eldery DESC, last\_name, first\_name LIMIT 3;

**11 Show the list of all cities where the employees are from.**

SELECT first\_name, last\_name, city.name As cities

FROM person

JOIN city ON person.id\_city = city.id

WHERE role = 'employee'

ORDER by city.name

**12 Show first, last names and dates of birth of the employees who celebrate their birthdays this month.**

SELECT  person.first\_name, person.last\_name, person.birthday\_date

FROM person

WHERE EXTRACT(MONTH FROM birthday\_date) = 3 AND role = 'employee'

GROUP BY  person.first\_name, person.last\_name, person.birthday\_date

**13 Show first and last names of the employees who used to serve orders shipped to Madrid.**

SELECT DISTINCT P.first\_name, P.last\_name

From person AS P

JOIN orders AS O ON P.id=O.id\_employee

WHERE O.id\_customer IN

(SELECT O.id\_customer

FROM orders AS O

JOIN person AS P ON O.id\_customer=P.id

JOIN city AS C ON P.id\_city=C.id

WHERE C.name='Madrid');

**14 Show first and last names of the employees as well as the count of orders each of them have received during the year 1997 (use left join).**

SELECT p.first\_name, p.last\_name, COUNT(o.id) AS Cou

FROM person AS p

JOIN orders AS o ON p.id=o.id\_employee

WHERE p.role='employee'

And EXTRACT(year from o.orders\_date)=1997

GROUP BY P.first\_name, p.last\_name

HAVING COUNT(O.id)>0

**15 Show first and last names of the employees as well as the count of orders each of them have received during the year 1997 (use a subquery).**

SELECT count (orders.id), person.first\_name, person.last\_name, orders.orders\_date

FROM orders

LEFT JOIN person ON person.id = orders.id\_employee

WHERE role = 'employee' AND orders\_date >= '1997-01-01' AND orders\_date <= '1997-12-31'

GROUP BY person.first\_name, person.last\_name, orders.orders\_date

**16 Show first and last names of the employees as well as the count of their orders shipped after required date during the year 1997 (use left join).**

SELECT count (orders.id), person.first\_name, person.last\_name, orders.shipping\_date

From orders

LEFT JOIN person On person.id = orders.id\_employee

Where role = 'employee' AND shipping\_date >= '1997-01-01' AND shipping\_date <= '1997-12-31'

group by person.first\_name, person.last\_name, orders.shipping\_date

**17 Show the count of orders made by each customer from France.**

SELECT COUNT (orders.id), person.first\_name, person.last\_name, city.country

FROM orders

JOIN person ON person.id = orders.id\_customer

JOIN city ON city.id = person.id\_city

WHERE city.country = 'France'

GROUP BY person.first\_name, person.last\_name, city.country

**18 Show the list of french customers’ names who have made more than one order (use grouping).**

SELECT P.first\_name, COUNT(O.id) AS Cou

FROM person AS P

JOIN city AS C ON P.id\_city=C.id

JOIN orders AS O ON P.id=O.id\_customer

WHERE C.country='France'

AND P.role='customer'

GROUP BY P.first\_name

HAVING COUNT(O.id)>1

**19 Show the list of french customers’ names who have made more than one order (use a subquery).**

Select p.last\_name,p.first\_name,COUNT(o.id) AS Cou

from person as p

join orders as o on o.id\_customer=p.id

where o.id\_customer in

(SELECT p.id

FROM person as p INNER JOIN city as c ON p.id\_city = c.id

where c.country='France' and p.role='customer')

GROUP BY p.last\_name,p.first\_name

having count(o.id)>1;

**20 Show the list of customers’ names who used to order the ‘Tofu’ product (use a subquery).**

SELECT id,first\_name, last\_name from person WHERE id IN (SELECT id\_customer from "order" WHERE id IN (SELECT id\_order FROM "order/product" WHERE id\_product IN (SELECT id FROM product WHERE name='Tofu')))

**21 \*Show the list of customers’ names who used to order the ‘Tofu’ product, along with the total amount of the product they have ordered and with the total sum for ordered product calculated.**

SELECT product.name, person.first\_name, person.last\_name, SUM (orders\_products.price \* count) AS orders\_sum, count (orders\_products.count)

FROM orders\_products

JOIN product ON orders\_products.id\_product = product.id

JOIN orders ON orders.id = orders\_products.id\_order

JOIN person ON person.id = orders.id\_customer

WHERE name like '%Samsung%'

GROUP BY product.name, person.first\_name, person.last\_name, orders\_products.count

ORDER BY orders\_sum

**22 \*Show the list of french customers’ names who used to order non-french products (use left join).**

SELECT category.name, orders\_date, SUM (orders\_products.price \* count)

FROM orders\_products

JOIN product ON orders\_products.id\_product = product.id

JOIN orders ON orders.id = orders\_products.id\_order

JOIN category ON category.id = product.id\_category

Where orders\_date >= '1997-01-01' and orders\_date <= '1997-12-31'

GROUP BY  category.name, orders\_date

Order by category.name

**23 \*Show the list of french customers’ names who used to order non-french products (use a subquery).**

SELECT P.first\_name

FROM person AS P

JOIN city AS C ON P.id\_city=C.id

JOIN orders AS O ON P.id=O.id\_customer

WHERE C.country='France' AND

P.role='customer' AND

O.id IN

(SELECT OP.id\_order

FROM orders\_products AS OP

WHERE OP.id\_product IN

(SELECT Pr.id

FROM product AS Pr

JOIN city AS C ON Pr.id\_city=C.id

WHERE C.country<>'France')

GROUP BY OP.id\_order)

GROUP BY P.first\_name

**24 \*Show the list of french customers’ names who used to order french products.**

SELECT P.first\_name, P.last\_name

FROM person AS P

JOIN city AS C ON P.id\_city=C.id

JOIN orders AS O ON P.id=O.id\_customer

WHERE C.country='France' AND

P.role='customer' AND O.id IN

(SELECT OP.id\_order

FROM orders\_products AS OP

WHERE OP.id\_product IN

(SELECT Pr.id

FROM product AS Pr

JOIN city AS C ON Pr.id\_city=C.id

WHERE C.country='France')

GROUP BY OP.id\_order)

GROUP BY P.first\_name, P.last\_name

**25 \*Show the total ordering sum calculated for each country of customer.**

SELECT C1.country, SUM(OP.price\*OP.count) AS total

FROM orders\_products AS OP

JOIN orders AS Ord ON OP.id\_order=Ord.id

JOIN person AS P ON Ord.id\_customer=P.id

JOIN city AS C1 ON P.id\_city=C1.id

GROUP BY C1.country

**26 \*Show the total ordering sums calculated for each customer’s country for domestic and non-domestic products separately (e.g.: “France – French products ordered – Non-french products ordered” and so on for each country).**

SELECT P.first\_name, P.last\_name, c.country AS domestic

FROM person AS P

JOIN city AS C ON P.id\_city=C.id

JOIN orders AS O ON P.id=O.id\_customer

WHERE C.country='France' AND

P.role='customer' AND O.id IN

(SELECT OP.id\_order

FROM orders\_products AS OP

WHERE OP.id\_product IN

(SELECT Pr.id

FROM product AS Pr

JOIN city AS C ON Pr.id\_city=C.id

WHERE C.country='Great Britain')

GROUP BY OP.id\_order)

GROUP BY P.first\_name, P.last\_name, c.country

**27 \*Show the list of product categories along with total ordering sums calculated for the orders made for the products of each category, during the year 1997.**

SELECT category.name, orders\_date, SUM (orders\_products.price \* count)

FROM orders\_products

JOIN product ON orders\_products.id\_product = product.id

JOIN orders ON orders.id = orders\_products.id\_order

JOIN category ON category.id = product.id\_category

Where orders\_date >= '1997-01-01' and orders\_date <= '1997-12-31'

GROUP BY  category.name, orders\_date

Order by category.name

**28 \*Show the list of product names along with unit prices and the history of unit prices taken from the orders (show ‘Product name – Unit price – Historical price’). The duplicate records should be eliminated. If no orders were made for a certain product, then the result for this product should look like ‘Product name – Unit price – NULL’. Sort the list by the product name.**

SELECT DISTINCT Pr.name,Pr.price AS curent, OP.price AS histor

FROM product AS Pr

LEFT JOIN orders\_products AS OP

ON Pr.id=OP.id\_product

ORDER BY Pr.name

**29 \*Show the list of employees’ names along with names of their chiefs (use left join with the same table).**

SELECT P2.first\_name AS k\_Employee, P.first\_name AS k\_Chief

FROM person P

LEFT JOIN person P2 ON P.id=P2.id\_chief

WHERE P2.first\_name IS NOT NULL

**30 \*Show the list of cities where employees and customers are from and where orders have been made to. Duplicates should be eliminated.**

**31 \*Insert 5 new records into Employees table. Fill in the following fields: LastName, FirstName, BirthDate, HireDate, Address, City, Country, Notes. The Notes field should contain your own name (to distinguish your records from the ones inserted by other trainees).**

ALTER TABLE person

ADD LastName varchar, FirstName varchar, BirthDate date, HireDate date, Address , City varchar, Country varchar, Notes varchar

**32 \*Fetch the records you have inserted by the SELECT statement**

INSERT INTO table2 (column1, column2, column3, ...)

SELECT column1, column2, column3, ...

FROM table1

WHERE condition;

**33 \*Change the City field in one of your records using the UPDATE statement (first run the SELECT statement to check whether you are updating the appropriate records!).**

Update person

SET first\_name='Ira'

Where first\_name=(SELECT first\_name FROM person WHERE first\_name='Iryna')

**34 \*Change the HireDate field in all your records to current date (first run the SELECT statement to check whether you are updating the appropriate records!).**

SET hire\_date=current\_date

WHERE role='employee';

**35 \*Delete one of your records (first run the SELECT statement to check whether you are deleting the appropriate record!).**

DELETE FROM person WHERE id=304;