SELECT \* FROM hr.employees;

SELECT CONCAT(first\_name, ' ', last\_name) AS 'employee\_name', department\_name FROM departments

JOIN employees USING (department\_id);

SELECT CONCAT(first\_name, ' ', last\_name) AS 'employee\_name', salary, salary/20 AS 'monthly\_salary', salary/20/8 AS 'salary\_by\_hour' FROM employees;

SELECT CONCAT(first\_name, ' ', last\_name) AS 'employee\_name', CONCAT(email,'@mail.somecompany.com') AS 'Full Email Address' FROM employees;

SELECT DISTINCT salary, CONCAT(first\_name, ' ', last\_name) AS 'employee\_name' FROM hr.employees ;

SELECT department\_name AS 'everything' FROM departments UNION

SELECT region\_name AS 'everything' FROM regions UNION

SELECT country\_name AS 'everything' FROM countries UNION

SELECT city AS 'everything' FROM locations;

SELECT \* FROM employees WHERE job\_id = 'AC\_MGR';

SELECT \* FROM employees WHERE first\_name LIKE 'Sa%';

SELECT \* FROM employees WHERE last\_name LIKE '%ei%';

SELECT \* FROM employees WHERE salary BETWEEN 3000 AND 5000;

SELECT

CONCAT(first\_name, ' ', last\_name) AS employees\_salary\_filtered

FROM

hr.employees

WHERE

salary BETWEEN 2000 AND 15000

AND salary NOT BETWEEN 5000 AND 10000;

SELECT \* FROM employees WHERE salary IN(2500,4000,5000);

SELECT \* FROM locations WHERE state\_province IS NULL AND postal\_code IS NULL;

SELECT \* FROM employees WHERE salary > 10000 ORDER BY salary;

SELECT \* FROM employees ORDER BY hire\_date DESC LIMIT 10;

SELECT \* FROM departments NATURAL JOIN locations;

SELECT \* FROM departments JOIN locations USING (location\_id);

SELECT \* FROM departments JOIN locations ON departments.location\_id = locations.location\_id;

SELECT department\_name, city, CONCAT(first\_name,' ', last\_name) AS 'Manager name' FROM departments

JOIN locations ON departments.location\_id = locations.location\_id

JOIN employees ON departments.manager\_id = employees.manager\_id;

SELECT city, department\_name FROM departments RIGHT OUTER JOIN locations ON departments.location\_id = locations.location\_id;

SELECT city, department\_name FROM departments LEFT OUTER JOIN locations ON departments.location\_id = locations.location\_id;

SELECT city, department\_name FROM departments , locations WHERE departments.location\_id = locations.location\_id;

SELECT CONCAT(employees.first\_name,' ', employees.last\_name) AS 'Manager name', departments.department\_name, locations.city FROM employees, departments, locations;

SELECT CONCAT(employees.first\_name,' ', employees.last\_name) AS 'Employee name', departments.department\_name, hire\_date FROM employees, departments

WHERE hire\_date > '1995' AND hire\_date < '2000' ORDER BY departments.department\_name;

SELECT DISTINCT CONCAT(employees.first\_name,' ', employees.last\_name) AS 'Employee name that worked in Sales', job\_history.department\_id

FROM departments, employees

JOIN job\_history ON employees.employee\_id = job\_history.employee\_id

WHERE departments.department\_id = job\_history.department\_id AND departments.department\_name = 'Sales';

SELECT CONCAT(e.first\_name,' ', e.last\_name) AS 'employee name', CONCAT(m.first\_name,' ', m.last\_name) AS 'manager name' FROM employees e

JOIN employees m ON e.MANAGER\_ID = m.EMPLOYEE\_ID;

SELECT first\_name , department\_name FROM employees

CROSS JOIN departments;

SELECT CONCAT(employees.first\_name,' ', employees.last\_name) AS 'employee name', job\_title, department\_name, city, country\_name, region\_name FROM employees

JOIN jobs USING (job\_id)

JOIN departments USING (department\_id)

JOIN locations USING (location\_id)

JOIN countries USING (country\_id)

JOIN regions USING (region\_id);

SELECT CONCAT(e.first\_name,' ', e.last\_name) AS 'employee name', COALESCE (CONCAT(m.first\_name,' ', m.last\_name), 'No Manager') AS 'Manager name',job\_title, department\_name, city, country\_name, region\_name FROM employees e

JOIN jobs USING (job\_id)

JOIN departments USING (department\_id)

JOIN locations USING (location\_id)

JOIN countries USING (country\_id)

JOIN regions USING (region\_id)

LEFT JOIN employees m ON m.employee\_id = e.manager\_id;

SELECT \* from job\_history;

SELECT concat(e.first\_name,' ',e.last\_name) AS name,jh.job\_id AS position\_in\_past1

,jh2.job\_id AS position\_in\_past2,j.job\_id AS position\_now,j.job\_title AS job\_title\_now

FROM employees AS e

JOIN job\_history AS jh ON(e.employee\_id = jh.employee\_id AND jh.job\_id = 'AC\_ACCOUNT')

JOIN job\_history AS jh2 ON (e.employee\_id = jh2.employee\_id AND jh2.job\_id = 'AC\_MGR')

JOIN jobs AS j ON (e.job\_id = j.job\_id);

/\*

SQL Basics

Задача

1.

Напишете и тествайте следните заявки :

1.

Write a SQL query to display all information about all departments.

2.

Write a SQL query to find all department names.

3.

Write a SQL query to find the salary of each employee by month, by day

and

hour. Consider that one month has 20 workdays and each workday

has 8 work hours.

4.

Write a SQL query to find the email addresses of each employee.

Consider that the mail domain is mail.somecompany.com. Emails should

look like "bernst@mail.somecompany.com".

The produced column

should be named "Full Email Address".

5.

Write a SQL query to find all different salaries that are paid to

the employees. Use DISTINCT.

6.

Write a SQL query to find all departments and all region

names, country names and city names as a singl

e list. Use UNION.

7.

Write a SQL query to find all information about the

employees whose position is "AC\_MGR" (Accounting Manager).

8.

Write a SQL query to find the names of all employees whose

first name starts with "Sa". Use LIKE.

9.

Write a SQL query to find th

e names of all employees whose last name

contains the character sequence "ei". Use LIKE.

10.

Write a SQL query to find the names of all employees whose

salary is in the range [3000...5000]. Use BETWEEN.

11.

Write a SQL query to find the names of all employees whos

e

salary is in the range [2000...15000] but is not in range [5000 ... 10000].

Use MINUS.

12.

Write a SQL query to find the names of all employees whose

salary is 2500, 4000 or 5000. Use IN.

13.

Write a SQL query to find all locations that have no state or

post code

defined. Use IS NULL.

14.

Write a SQL query to find all employees that are paid more

than 10 000. Order them in decreasing order by salary. Use ORDER BY.

15.

Write a SQL query to find the first 10 employees joined the

company (most senior people).

16.

Write a SQL

query to find all departments and the town of

their location. Use NATURAL JOIN.

17.

Write a SQL query to find all departments and the town of

their location. Use join with USING clause.