

1. Таблиця Employees.

Напишіть запит для вибору імен, прізвищ, окладів усіх працівників та їх id департаменту, оклад яких не перевищує 10000.

```
Запит:  SELECT FIRST_NAME, LAST_NAME, SALARY,  
DEPARTMENT_ID  
FROM HR.EMPLOYEES  
WHERE SALARY <=10000;
```

2. Таблиця Employees.

Напишіть запит для відображення імен, прізвищ, ідентифікаторів посад та комісійних винагород всіх працівників, які не мають права на отримання комісійної винагороди.

```
Запит:  SELECT FIRST_NAME, LAST_NAME, JOB_ID,  
COMMISSION_PCT  
FROM HR.EMPLOYEES  
WHERE COMMISSION_PCT IS NULL;
```

3. Таблиця Employees.

Напишіть 2 варіанти запиту для відображення списку всіх імен, прізвищ та їх id департаменту, де ім'я співробітника John.

```
Запит 1.: SELECT FIRST_NAME, LAST_NAME, DEPARTMENT_ID  
FROM HR.EMPLOYEES  
WHERE FIRST_NAME LIKE 'John';
```

```
Запит 2.: SELECT FIRST_NAME, LAST_NAME, DEPARTMENT_ID  
FROM HR.EMPLOYEES  
WHERE FIRST_NAME = 'John';
```

4. Таблиця Employees.

Напишіть запит для відображення імен, прізвищ співробітників із ЗП із 80-го департаменту і ЗП менше 10000.

```
Запит:  SELECT FIRST_NAME, LAST_NAME, SALARY,  
DEPARTMENT_ID  
FROM HR.EMPLOYEES  
WHERE SALARY <10000 AND DEPARTMENT_ID=80;
```

5. Таблиця Employees.

Напишіть запит для відображення імен, прізвищ співробітників із ЗП із 80-го і із 110 департаменту, ЗП яких знаходиться в діапазоні від 10000 до 17000.

Запит: SELECT FIRST_NAME, LAST_NAME, SALARY,
DEPARTMENT_ID
FROM HR.EMPLOYEES
WHERE SALARY BETWEEN 10000 AND 17000 AND DEPARTMENT_ID
IN (80,110);

6. Таблиця Employees.

Напишіть запит для відображення імен, прізвищ співробітників із ЗП із 80-го і із 110 департаменту, ЗП яких знаходиться в діапазоні від 10000 до 17000 і які отримують комісію.

Запит: SELECT FIRST_NAME, LAST_NAME, SALARY,
DEPARTMENT_ID, COMMISSION_PCT
FROM HR.EMPLOYEES
WHERE SALARY BETWEEN 10000 AND 17000 AND DEPARTMENT_ID
IN (80,110) AND COMMISSION_PCT IS NOT NULL;

7. Таблиця Employees.

Напишіть запит для відображення списку всіх імен, прізвищ співробітників, у кого ім'я починається на D та підпорядковується менеджеру 108.

Запит: SELECT FIRST_NAME, LAST_NAME, MANAGER_ID FROM
HR.EMPLOYEES
WHERE FIRST_NAME LIKE 'D%' AND MANAGER_ID =108;

8. Таблиця Employees.

Напишіть запит для відображення всіх даних співробітників, хто підпорядковується менеджерам 100, 108, 124.

Запит 1: SELECT * FROM HR.EMPLOYEES WHERE MANAGER_ID
IN (100,108,124);

Запит 2: SELECT * FROM HR.EMPLOYEES WHERE
MANAGER_ID=100 OR MANAGER_ID=108 OR MANAGER_ID=124;

9. Таблиця Employees.

Напишіть запит для відображення всіх даних співробітників, хто не отримує комісію та має ЗП менше 10000 та відсортуйте вибірку за зменшенням ЗП

Запит: SELECT * FROM HR.EMPLOYEES
WHERE COMMISSION_PCT IS NULL AND SALARY <10000 ORDER
BY SALARY DESC;

10. Таблиця Employees.

Напишіть запит для відображення прізвища і ім'я та ЗП, хто не отримує комісію та має ЗП менше 10000 та відсортуйте вибірку за зменшенням ЗП. Прізвище і ім'я відобразить в одному полі, яке назвіть "List of employees for review".

Напишіть 2 варіанти запиту використовуючи різні оператори об'єднання полів.

```
Запит 1.: SELECT CONCAT (FIRST_NAME, CONCAT (' ',  
LAST_NAME)) AS List_of_employees_for_review , SALARY,  
COMMISSION_PCT  
FROM HR.EMPLOYEES  
WHERE COMMISSION_PCT IS NULL AND SALARY <10000 ORDER  
BY SALARY DESC;
```

```
Запит 2.: SELECT FIRST_NAME || ' ' || LAST_NAME AS  
List_of_employees_for_review , SALARY, COMMISSION_PCT  
FROM HR.EMPLOYEES  
WHERE COMMISSION_PCT IS NULL AND SALARY <10000 ORDER  
BY SALARY DESC;
```

11. Таблиця DEPARTMENTS.

Застосовуючи оператор CASE напишіть скрипт, який буде виводити список назв департаментів та нове поле "State", яке буде заповнюватися в залежності від того, яке значення має поле LOCATION_ID.

Для LOCATION_ID 1700 нове поле "State" має містити запис "Washington",

Для 1400 - "Texas"

Для 1500 і 2500 - "California"

Для 1800 - "Ontario"

2400 - "Other"

2700 - "Bavaria"

Запит:

```
SELECT department_name, location_id,  
CASE  
    WHEN location_id = 1700 THEN 'Washington'  
    WHEN location_id = 1400 THEN 'Texas'  
    WHEN location_id in (1500,2500) THEN 'California'  
    WHEN location_id = 1800 THEN 'Ontario'  
    WHEN location_id = 2400 THEN 'Other'  
    WHEN location_id = 2700 THEN 'Bavaria'  
    ELSE null  
END AS State  
FROM hr.departments;
```

12. Таблиця LOCATIONS.

Напишіть запит для відображення інформації про кількість локацій компанії у кожному COUNTRY_ID та відсортуйте список від країни із найбільшою кількістю локацій компанії.

Запит:

```
SELECT COUNT (location_id), country_id
FROM hr.locations
GROUP BY country_id
ORDER BY COUNT (location_id) DESC;
```

13. Таблиця LOCATIONS.

Напишіть запит для відображення інформації про кількість локацій компанії у кожному COUNTRY_ID, вивівши інформацію тільки по країнам, де кількість локацій більше двох

Запит:

```
SELECT COUNT (location_id), country_id
FROM hr.locations
GROUP BY country_id
HAVING COUNT (location_id) > 2;
```

14. Таблиці LOCATIONS та COUNTRIES.

Напишіть запит для відображення всіх значень полів STREET_ADDRESS, STATE_PROVINCE та COUNTRY_NAME та відсортуйте за алфавітом (від А до Z) назву країн.

Запит:

```
SELECT hr.locations.street_address,
       hr.locations.state_province,
       hr.countries.country_name
FROM hr.locations
      FULL JOIN hr.countries ON hr.locations.country_id =
       hr.countries.country_id
Order by country_name ASC;
```

15. Таблиці LOCATIONS та COUNTRIES.

Напишіть запит для розрахунку кількості STREET_ADDRESS за кожною із COUNTRY_NAME, які мають відповідні значення в обох таблицях. Виведіть усі показники.

Запит:

```
SELECT COUNT (hr.locations.street_address),
       hr.countries.country_name
FROM hr.locations
```

```
JOIN hr.countries ON hr.locations.country_id =  
hr.countries.country_id  
GROUP BY country_name;
```

16. Таблиці LOCATIONS, COUNTRIES, REGIONS.

Напишіть запит для відображення всіх значень полів, STATE_PROVINCE та COUNTRY_NAME, REGION_NAME.

Запит:

```
SELECT hr.locations.state_province,  
       hr.countries.country_name, hr.regions.region_name  
FROM hr.locations  
FULL JOIN hr.countries ON hr.locations.country_id =  
       hr.countries.country_id  
FULL JOIN hr.regions ON hr.countries.region_id =  
       hr.regions.region_id;
```

17. Таблиці LOCATIONS, COUNTRIES, REGIONS.

Напишіть запит для розрахунку кількості STREET_ADDRESS кількості COUNTRY_NAME для кожного REGION_NAME, які мають відповідні значення в усіх трьох таблицях. Виведіть усі показники.

Запит:

```
SELECT COUNT (hr.locations.street_address), COUNT  
       (hr.countries.country_name), hr.regions.region_name  
FROM hr.locations  
JOIN hr.countries ON hr.locations.country_id =  
       hr.countries.country_id  
JOIN hr.regions ON hr.countries.region_id =  
       hr.regions.region_id  
GROUP BY region_name;
```

18. Таблиці LOCATIONS, COUNTRIES, REGIONS.

Напишіть запит для розрахунку кількості STREET_ADDRESS кількості COUNTRY_NAME для кожного REGION_NAME, які мають відповідні значення в усіх трьох таблицях. Та виведіть ті REGION_NAME, де кількість COUNTRY_NAME буде більше 6.

Запит:

```
SELECT COUNT (hr.locations.street_address), COUNT  
       (hr.countries.country_name), hr.regions.region_name  
FROM hr.locations  
JOIN hr.countries ON hr.locations.country_id =  
       hr.countries.country_id
```

```
JOIN hr.regions ON hr.countries.region_id =  
    hr.regions.region_id  
GROUP BY region_name  
HAVING COUNT (country_name) > 6;
```

19. Таблиці LOCATIONS, COUNTRIES, REGIONS.

Напишіть запит для розрахунку кількості STREET_ADDRESS
кількості COUNTRY_NAME для кожного REGION_NAME, які
мають відповідні значення в усіх трьох таблицях. Та виведіть ті
REGION_NAME, де кількість COUNTRY_NAME буде більше 6.
Виведіть усі показники. Напишіть скрипт використовуючи аліаси
для назв полів.

Запит:

```
SELECT COUNT (hr.locations.street_address) as  
    STREET_ADDRESS, COUNT (hr.countries.country_name)  
    as COUNTRY_NAME, hr.regions.region_name  
    REGION_NAME  
FROM hr.locations  
JOIN hr.countries ON hr.locations.country_id =  
    hr.countries.country_id  
JOIN hr.regions ON hr.countries.region_id =  
    hr.regions.region_id  
GROUP BY region_name  
HAVING COUNT (country_name) > 6;
```

20. Таблиці JOB_HISTORY, EMPLOYEES

Оператор UNION

Напишіть запит в якому будуть відображатися поля ідентифікатор
співробітника, дата найму та дата звільнення. Якщо співробітник
досі працює в компанії дата звільнення має бути null.

Запит:

```
SELECT EMPLOYEE_ID, START_DATE, END_DATE  
FROM hr.JOB_HISTORY  
UNION  
SELECT EMPLOYEE_ID, HIRE_DATE, null as END_DATE  
FROM hr.EMPLOYEES;
```

21. Таблиці JOB_HISTORY, EMPLOYEES

Оператор UNION

Напишіть запит в якому буде відображатися статус співробітника
(status) із варіантами заповнення current_employees - якщо
співробітник досі працює в компанії і fired_employees - якщо
співробітник звільнився та кількість співробітників за кожним із
цих статусів

Запит:

```
SELECT count (EMPLOYEE_ID), 'current_employees' as  
      status  
FROM hr.EMPLOYEES  
UNION  
SELECT count (EMPLOYEE_ID), 'fired_employees' as status  
      FROM hr.JOB_HISTORY;
```

22. Таблиці JOB_HISTORY, EMPLOYEES

Оператор UNION

Напишіть запит в якому буде відображатися статус співробітника (status) із варіантами заповнення current_employees - якщо співробітник досі працює в компанії і fired_employees - якщо співробітник звільнився та кількість співробітників за кожним із цих статусів групуючи ці показники за ідентифікатором посади.

Запит:

```
SELECT count (EMPLOYEE_ID), 'current_employees' as  
      status, job_id  
FROM hr.EMPLOYEES  
Group by job_id  
UNION  
SELECT count (EMPLOYEE_ID), 'fired_employees' as  
      status, job_id  
FROM hr.JOB_HISTORY  
Group by job_id;
```

23. Таблиці JOB_HISTORY, EMPLOYEES.

SUBQUERY

Напишіть запит в якому буде відображатися за зменшенням кількість співробітників за кожним департаментом, лише серед тих департаментів, де було звільнено більше 1 співробітника.

Запит:

```
SELECT COUNT (employee_ID), DEPARTMENT_ID  
FROM HR.EMPLOYEES  
WHERE DEPARTMENT_ID IN (SELECT department_id  
FROM HR.JOB_HISTORY group by department_id having Count  
      (employee_id) > 1)  
GROUP BY DEPARTMENT_ID  
ORDER BY COUNT (employee_ID) DESC;
```

24. Таблиці JOB_HISTORY, EMPLOYEES.
SUBQUERY

Напишіть запит в якому буде відображатися ідентифікатори співробітників, імена, прізвища тільки із тих департаментів, де є співробітники, які працювали менше року

Запит:

```
SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME
FROM hr.EMPLOYEES
WHERE DEPARTMENT_ID IN (SELECT DEPARTMENT_ID FROM
    hr.JOB_HISTORY WHERE END_DATE - START_DATE <
    365);
```

25. Таблиці JOB_HISTORY, EMPLOYEES.
SUBQUERY

Напишіть запит в якому буде відображатися ідентифікатори співробітників, імена, прізвища тільки за умови, якщо в цьому департаменті є співробітники, які пропрацювали більше року

Запит:

```
SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME
FROM hr.EMPLOYEES
WHERE DEPARTMENT_ID IN (SELECT DEPARTMENT_ID FROM
    hr.JOB_HISTORY WHERE END_DATE - START_DATE >
    365);
```

26. Таблиці JOB_HISTORY, EMPLOYEES.
SUBQUERY

Напишіть запит в якому буде відображатися кількість співробітників за кожним із департаментів тільки за умови, якщо в цьому департаменті є співробітники, які пропрацювали більше року. Відсортуйте всі записи за зменшенням кількості співробітників.

Запит:

```
SELECT Count (EMPLOYEE_ID), DEPARTMENT_ID
FROM hr.EMPLOYEES
WHERE DEPARTMENT_ID IN (SELECT DEPARTMENT_ID FROM
    hr.JOB_HISTORY WHERE END_DATE - START_DATE > 365)
GROUP BY DEPARTMENT_ID
ORDER BY COUNT (employee_ID) DESC;
```

27. Таблиця EMPLOYEES

Напишіть скрипт, який має додавати запис у таблицю із вашим іменем і прізвищем. Дата найму завжди має бути сьогоднішнім

числом, зп = 25000, департамент = 110, менеджер 205. Інші поля можуть бути заповнені довільно.

Запит:

```
INSERT INTO hr.employees (EMPLOYEE_ID, FIRST_NAME,  
    LAST_NAME, EMAIL, PHONE_NUMBER, HIRE_DATE, JOB_ID,  
    SALARY, COMMISSION_PCT, MANAGER_ID, DEPARTMENT_ID)  
VALUES (205, 'Kateryna', 'Tuchkova', 'kata1990',  
    '1234567', 'SYSDATE', 'AD_VP', 25000, null, null,  
    110);
```

28. Таблиця EMPLOYEES.

Напишіть скрипт, який має оновлювати створений вище запис у таблиці із вашим іменем і прізвищем. Змініть у цьому записі департамент на 90, а ЗП на 30000.

Запит:

```
UPDATE hr.employees  
SET DEPARTMENT_ID = 90, SALARY = 3000  
WHERE EMPLOYEE_ID = 205;
```

29. Таблиця EMPLOYEES

Напишіть скрипт, який має видаляти створений вами запис у таблиці.

Запит:

```
DELETE  
FROM hr.employees  
WHERE EMPLOYEE_ID = 205;
```

30. Напишіть запит для відображення всіх записів полів OrderHeader_Number, OrderHeader_Description та OrderDetail_ArticleNumber, OrderDetail_Quantity, OrderDetail_Price за умови, що OrderDetail_Quantity більше 3 та відсортуйте всі записи за збільшенням вартості.

```
Запит: SELECT OrderHeader.OrderHeader_Number,  
    OrderHeader.OrderHeader_Description,  
    OrderDetail.OrderDetail_ArticleNumber, OrderDetail.OrderDe  
tail_Quantity, OrderDetail.OrderDetail_Price  
FROM OrderHeader  
JOIN OrderDetail  
ON  
OrderHeader.OrderHeader_ID=  
OrderDetail.OrderDetail_HeadID  
WHERE OrderDetail.OrderDetail_Quantity > 3  
ORDER BY OrderDetail_Price ASC;
```