

Ćwiczenie 2

- I. **Usuń wszystkie tabele ze swojej bazy**
- II. **Przekopiuj wszystkie tabele wraz z danymi od użytkownika HR. Poustawiaj klucze główne i obce**

```
SET FOREIGN_KEY_CHECKS = 0;
```

```
SET FOREIGN_KEY_CHECKS = 1;
```

```
DROP TABLE COUNTRIES, JOB_HISTORY, DEPARTMENTS, LOCATIONS, REGIONS,  
EMPLOYEES, JOBS;
```

```
CREATE TABLE employees AS SELECT * FROM HR.employees;
```

```
CREATE TABLE departments AS SELECT * FROM HR.departments;
```

Dodanie kluczy:

```
ALTER TABLE EMPLOYEES ADD PRIMARY KEY (EMPLOYEE_ID);
```

```
ALTER TABLE DEPARTMENTS ADD PRIMARY KEY (DEPARTMENT_ID);
```

```
ALTER TABLE EMPLOYEES ADD FOREIGN KEY (DEPARTMENT_ID) REFERENCES  
DEPARTMENTS(DEPARTMENT_ID);
```

- III. **Stwórz następujące perspektywy lub zapytania, dodaj wszystko do swojego repozytorium:**

1. Z tabeli employees wypisz w jednej kolumnie nazwisko i zarobki – nazwij kolumnę wynagrodzenie, dla osób z departamentów 20 i 50 z zarobkami pomiędzy 2000 a 7000, uporządkuj kolumny według nazwiska

```
SELECT CONCAT(LAST_NAME, ", SALARY) AS wynagrodzenie FROM employees  
WHERE DEPARTMENT_ID IN (20, 50) AND SALARY BETWEEN 2000 AND 7000  
ORDER BY LAST_NAME;
```

2. Z tabeli employees wyciągnąć informację data zatrudnienia, nazwisko oraz kolumnę podaną przez użytkownika dla osób mających menadżera zatrudnionych w roku 2005. Uporządkować według kolumny podanej przez użytkownika

```
SELECT HIRE_DATE, LAST_NAME, &COLUMN_NAME AS additional_column FROM  
employees
```

```
WHERE MANAGER_ID IS NOT NULL AND EXTRACT(YEAR FROM HIRE_DATE) =  
2005
```

ORDER BY &COLUMN_NAME;

3. Wypisać imiona i nazwiska razem, zarobki oraz numer telefonu porządkując dane według pierwszej kolumny malejąco a następnie drugiej rosnąco (użyć numerów do porządkowania) dla osób z trzecią literą nazwiska 'e' oraz częścią imienia podaną przez użytkownika

```
SELECT FIRST_NAME || ' ' || LAST_NAME AS full_name,
```

```
    SALARY,
```

```
    PHONE_NUMBER
```

```
FROM EMPLOYEES
```

```
WHERE SUBSTR(LAST_NAME, 3, 1) = 'e' AND LOWER(FIRST_NAME) LIKE '%' ||  
LOWER('&PART_NAME') || '%'
```

```
ORDER BY 1 DESC, 2 ASC;
```

4. Wypisać imię i nazwisko, liczbę miesięcy przepracowanych – funkcje months_between oraz round oraz kolumnę wysokość dodatku jako (użyć CASE lub DECODE):

10% wynagrodzenia dla liczby miesięcy do 150

20% wynagrodzenia dla liczby miesięcy od 150 do 200

30% wynagrodzenia dla liczby miesięcy od 200

uporządkować według liczby miesięcy

```
SELECT FIRST_NAME || ' ' || LAST_NAME AS full_name,
```

```
    ROUND(MONTHS_BETWEEN(SYSDATE, HIRE_DATE)) AS months_worked,
```

```
    CASE
```

```
        WHEN ROUND(MONTHS_BETWEEN(SYSDATE, HIRE_DATE)) < 150 THEN  
SALARY * 0.1
```

```
        WHEN ROUND(MONTHS_BETWEEN(SYSDATE, HIRE_DATE)) BETWEEN 150  
AND 200 THEN SALARY * 0.2
```

```
        ELSE SALARY * 0.3
```

```
    END AS bonus
```

```
FROM EMPLOYEES
```

ORDER BY months_worked;

5. Dla każdego działów w których minimalna płaca jest wyższa niż 5000 wypisz sumę oraz średnią zarobków zaokrągloną do całości nazwij odpowiednio kolumny

```
SELECT DEPARTMENT_ID,  
       ROUND(SUM(SALARY)) AS total_salary,  
       ROUND(AVG(SALARY)) AS average_salary  
FROM EMPLOYEES  
GROUP BY DEPARTMENT_ID  
HAVING MIN(SALARY) > 5000;
```

6. Wypisać nazwisko, numer departamentu, nazwę departamentu, id pracy, dla osób z pracujących Toronto
7. Dla pracowników o imieniu „Jennifer” wypisz imię i nazwisko tego pracownika oraz osoby które z nim współpracują

```
SELECT employees.FIRST_NAME || ' ' || employees.LAST_NAME AS jennifer_name,  
       colleagues.FIRST_NAME || ' ' || colleagues.LAST_NAME AS colleague_name  
FROM employees  
JOIN employees colleagues ON employees.DEPARTMENT_ID =  
colleagues.DEPARTMENT_ID  
WHERE employees.FIRST_NAME = 'Jennifer' AND employees.EMPLOYEE_ID !=  
colleagues.EMPLOYEE_ID;
```

8. Wypisać wszystkie departamenty w których nie ma pracowników

```
SELECT departments.DEPARTMENT_ID, departments.DEPARTMENT_NAME  
FROM departments  
LEFT JOIN employees ON departments.DEPARTMENT_ID =  
employees.DEPARTMENT_ID  
WHERE employees.EMPLOYEE_ID IS NULL;
```

9. Skopiuj tabelę Job_grades od użytkownika HR

```
CREATE TABLE job_grades_copy AS SELECT * FROM HR.JOB_GRADES;
```

10. Wypisz imię i nazwisko, id pracy, nazwę departamentu, zarobki, oraz odpowiedni grade dla każdego pracownika

```
SELECT EMPLOYEES.FIRST_NAME || ' ' || EMPLOYEES.LAST_NAME AS  
employee_name,  
  
EMPLOYEES.JOB_ID,  
  
departments.DEPARTMENT_NAME,  
  
EMPLOYEES.SALARY,  
  
JOB_GRADES_COPY.GRADE  
  
FROM EMPLOYEES  
  
JOIN departments ON EMPLOYEES.DEPARTMENT_ID =  
departments.DEPARTMENT_ID  
  
JOIN JOB_GRADES_COPY ON EMPLOYEES.SALARY BETWEEN  
JOB_GRADES_COPY.MIN_SALARY AND JOB_GRADES_COPY.MAX_SALARY;
```

11. Wypisz imię nazwisko oraz zarobki dla osób które zarabiają więcej niż średnia wszystkich, uporządkuj malejąco według zarobków

```
SELECT employees.FIRST_NAME || ' ' || employees.LAST_NAME AS full_name,  
  
employees.SALARY  
  
FROM employees  
  
WHERE employees.SALARY > (SELECT AVG(employees.SALARY) FROM employees)  
  
ORDER BY employees.SALARY DESC;
```

12. Wypisz id imię i nazwisko osób, które pracują w departamencie z osobami mającymi w nazwisku „u”

```
SELECT employees.EMPLOYEE_ID, employees.FIRST_NAME, employees.LAST_NAME  
  
FROM employees  
  
WHERE employees.DEPARTMENT_ID IN (  
  
SELECT DISTINCT employees.DEPARTMENT_ID  
  
FROM employees  
  
WHERE LOWER(employees.LAST_NAME) LIKE '%u%'  
  
);
```