Andrzej Kaczmarek  
  
Bazy i struktury danych

**CW2**

**1.**

DROP TABLE REGIONS CASCADE CONSTRAINTS;

DROP TABLE COUNTRIES CASCADE CONSTRAINTS;

DROP TABLE LOCATIONS CASCADE CONSTRAINTS;

DROP TABLE DEPARTMENTS CASCADE CONSTRAINTS;

DROP TABLE EMPLOYEES CASCADE CONSTRAINTS;

DROP TABLE JOBS CASCADE CONSTRAINTS;

DROP TABLE JOB\_HISTORY CASCADE CONSTRAINTS;

**2.**

CREATE TABLE REGIONS AS SELECT \* FROM hr.REGIONS

CREATE TABLE COUNTRIES AS SELECT \* FROM hr.COUNTRIES

CREATE TABLE LOCATIONS AS SELECT \* FROM hr.LOCATIONS

CREATE TABLE DEPARTMENTS AS SELECT \* FROM hr.DEPARTMENTS

CREATE TABLE EMPLOYEES AS SELECT \* FROM hr.EMPLOYEES

CREATE TABLE JOBS AS SELECT \* FROM hr.JOBS

CREATE TABLE JOB\_HISTORY AS SELECT \* FROM hr.JOB\_HISTORY

ALTER TABLE REGIONS ADD PRIMARY KEY (region\_id);

ALTER TABLE COUNTRIES ADD PRIMARY KEY (country\_id);

ALTER TABLE COUNTRIES ADD FOREIGN KEY (region\_id) REFERENCES REGIONS(region\_id);

ALTER TABLE LOCATIONS ADD PRIMARY KEY (location\_id);

ALTER TABLE LOCATIONS ADD FOREIGN KEY (country\_id) REFERENCES COUNTRIES(country\_id);

ALTER TABLE JOBS ADD PRIMARY KEY (job\_id);

ALTER TABLE DEPARTMENTS ADD PRIMARY KEY (department\_id);

ALTER TABLE EMPLOYEES ADD PRIMARY KEY (employee\_id);

ALTER TABLE JOB\_HISTORY ADD PRIMARY KEY (start\_date);

ALTER TABLE JOB\_HISTORY ADD FOREIGN KEY (job\_id) REFERENCES JOBS(job\_id);

ALTER TABLE JOB\_HISTORY ADD FOREIGN KEY (department\_id) REFERENCES DEPARTMENTS(department\_id);

ALTER TABLE DEPARTMENTS ADD FOREIGN KEY (manager\_id) REFERENCES EMPLOYEES(employee\_id);

ALTER TABLE DEPARTMENTS ADD FOREIGN KEY (location\_id) REFERENCES LOCATIONS(location\_id);

ALTER TABLE EMPLOYEES ADD FOREIGN KEY (job\_id) REFERENCES JOBS(job\_id);

ALTER TABLE EMPLOYEES ADD FOREIGN KEY (manager\_id) REFERENCES EMPLOYEES(employee\_id);

ALTER TABLE EMPLOYEES ADD FOREIGN KEY (department\_id) REFERENCES DEPARTMENTS(department\_id);

**3.**

**1.**

SELECT CONCAT(CONCAT(last\_name, ' '), salary) as wynagrodzenie from EMPLOYEES WHERE department\_id in (20, 50) AND salary BETWEEN 2000 AND 7000;

**2.**

SELECT hire\_date, last\_name, &x as user\_columnFROM EMPLOYEES WHERE manager\_id IS NOT NULL AND EXTRACT(YEAR FROM TO\_DATE(hire\_date, 'DD-MM-RR')) = 2005 ORDER BY user\_column;

**3.**

SELECT CONCAT(CONCAT(first\_name, ' '),last\_name) as Person, salary, phone\_number FROM EMPLOYEES WHERE last\_name LIKE '\_\_e%' AND first\_name LIKE '%' || '&imie' || '%' ORDERBY 1 desc, 2 asc;

**4.**

SELECT first\_name, last\_name, ROUND(MONTHS\_BETWEEN(sysdate,hire\_date),0) summed\_months CASE WHEN summed\_months < 150 THEN salary\* 0.1 WHEN summed\_months BETWEEN 150 AND 200 THEN salary\*0.3 WHEN summed\_months > 200 THEN salary\*0.3 END AS wysokosc\_dodatku FROM EMPLOYEES;

**5.**

SELECT DEPARTMENTS.department\_name, SUM(EMPLOYEES.salary) as Salary\_summary, ROUND(avg(EMPLOYEES.salary),0) as Average\_salary FROM EMPLOYEES, DEPARTMENTS GROUP BY EMPLOYEES.department\_id, DEPARTMENTS.department\_name HAVING MIN(EMPLOYEES.salary) > 5000;

**6.**

SELECT EMPLOYEES.last\_name, EMPLOYEES.department\_id, DEPARTMENTS.department\_name,EMPLOYEES.job\_id FROM EMPLOYEES, DEPARTMENTS, LOCATIONS WHERE EMPLOYEES.department\_id = DEPARTMENTS.department\_id AND DEPARTMENTS.location\_id = LOCATIONS.location\_id AND city='Toronto'

**7.**

SELECT first\_name,last\_name, department\_id FROM EMPLOYEES

WHERE department\_id IN (SELECT DISTINCT department\_id FROM EMPLOYEES WHERE first\_name = 'Jennifer');

**8.**

SELECT department\_name FROM DEPARTMENTS WHERE department\_id NOT IN(SELECT department\_id FROM EMPLOYEES WHERE department\_id IS NOT NULL)

**9.**

CREATE TABLE JOB\_GRADES AS SELECT \* FROM hr.JOB\_GRADES

**10.**

SELECT EMPLOYEES.first\_name, EMPLOYEES.last\_name, EMPLOYEES.job\_id, DEPARTMENTS.department\_name, EMPLOYEES.salary, JOB\_GRADES.grade FROM EMPLOYEES, DEPARTMENTS,JOB\_GRADES WHERE EMPLOYEES.department\_id = DEPARTMENTS.department\_id AND EMPLOYEES.salary BETWEEN JOB\_GRADES.MIN\_SALARY AND JOB\_GRADES.MAX\_SALARY;

**11.**

SELECT first\_name,last\_name, salary FROM EMPLOYEES WHERE salary < (SELECT AVG(salary) FROM EMPLOYEES) ORDER BY 3 DESC;

**12.**

SELECT employee\_id, first\_name, last\_name FROM EMPLOYEES WHERE department\_id IN(SELECT DISTINCT department\_id FROM EMPLOYEES WHERE last\_name LIKE '%u%')