**ОТЧЕТ**

**По лабораторной работе №4  
«Язык SQL. Операторы манипулирования данными»**

**Вариант 15**

Студент группы 13  
Доскоч Р.Д.

**1.Поднимите нижнюю границу минимальной заработной платы в таблице JOB до**

**1000$.**

UPDATE JOB

SET MINSALARY = 1000

WHERE MINSALARY < 1000;

**2. Поднимите минимальную зарплату в таблице JOB на 10% для всех специальностей,**

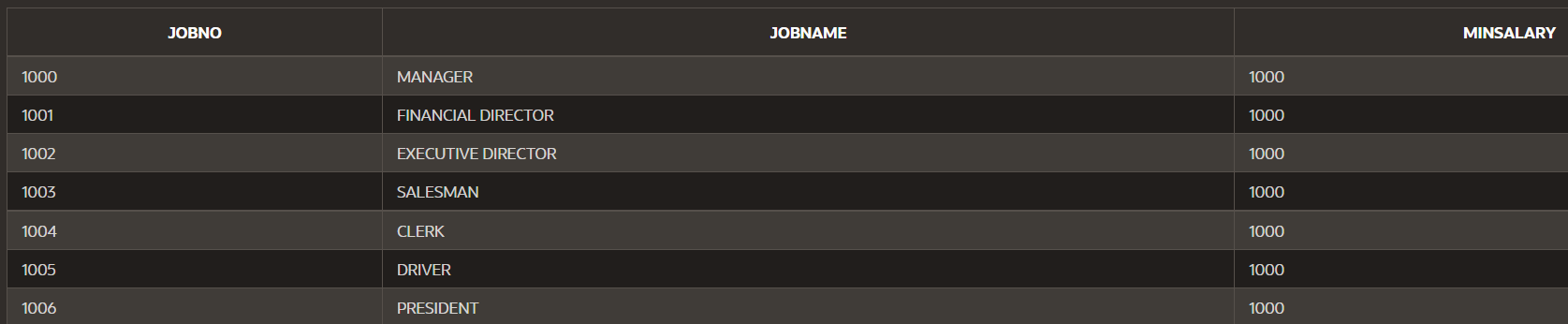
**кроме финансового директора.**

UPDATE JOB

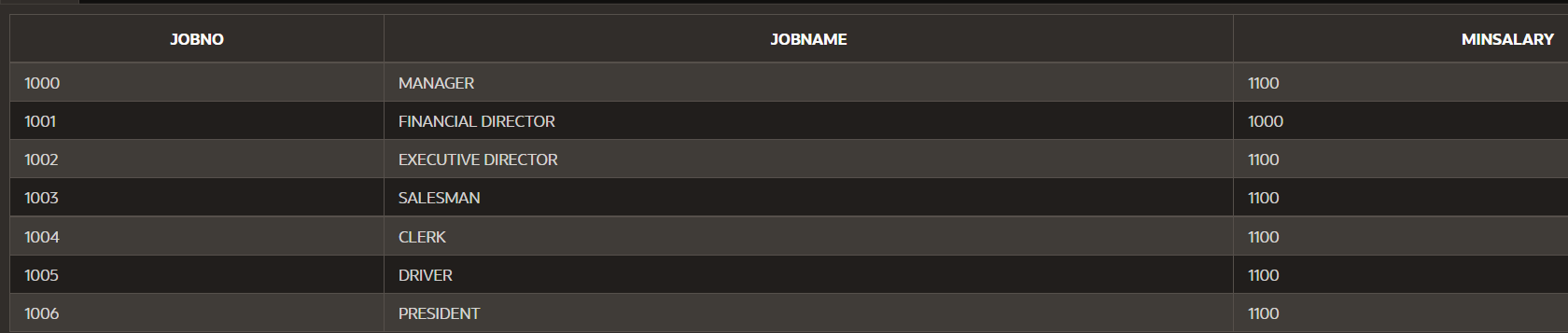
SET MINSALARY = MINSALARY \* 1.1

WHERE JOBNAME !='FINANCIAL DIRECTOR';

**До изменений**

****

**После**

****

**3. Поднимите минимальную зарплату в таблице JOB на 10% для клерков и на 20% для**

**финансового директора (одним оператором).**

UPDATE JOB

SET MINSALARY =

CASE

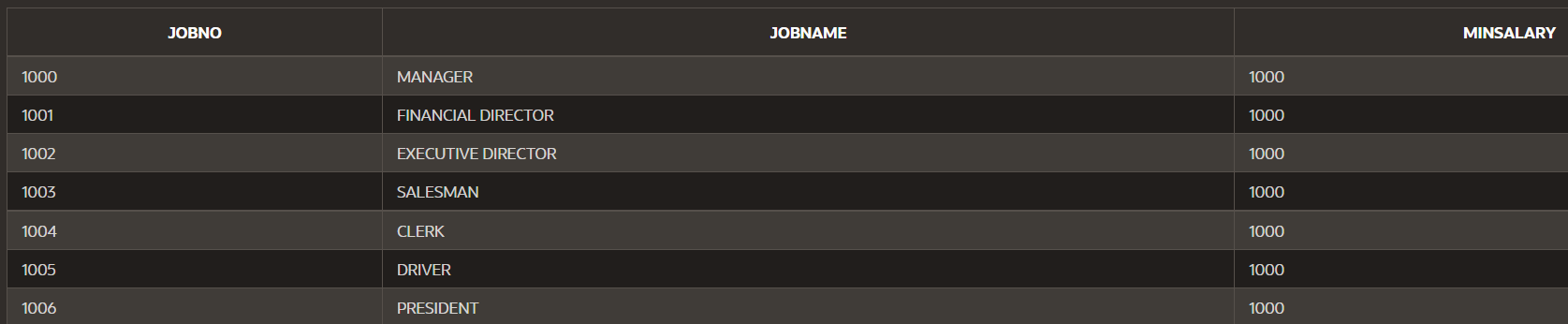
WHEN JOBNAME = 'CLERK' THEN MINSALARY \* 1.1

WHEN JOBNAME = 'FINANCIAL DIRECTOR' MINSALARY \* 1.2

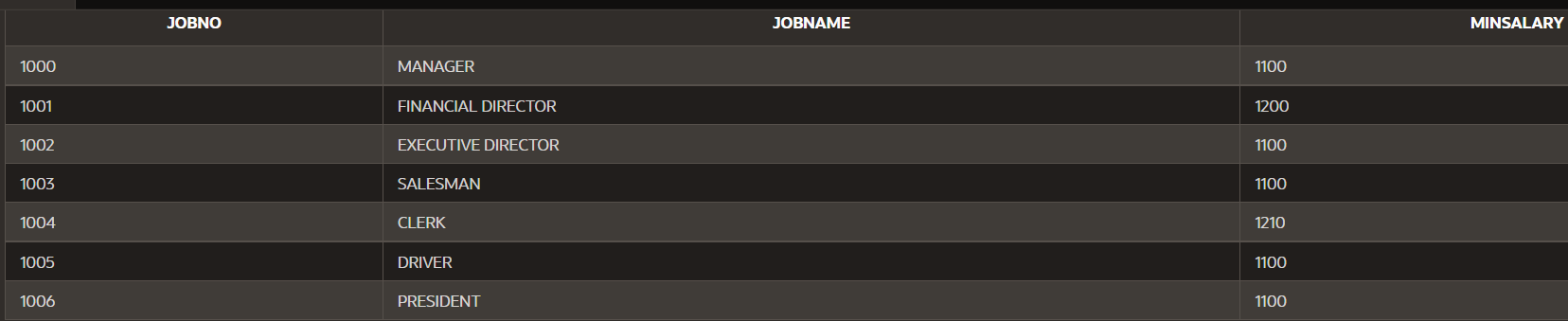
ELSE MINSALARY

END;

**До**

****

**После**

****

**4. Установите минимальную зарплату финансового директора равной 90% от зарплаты**

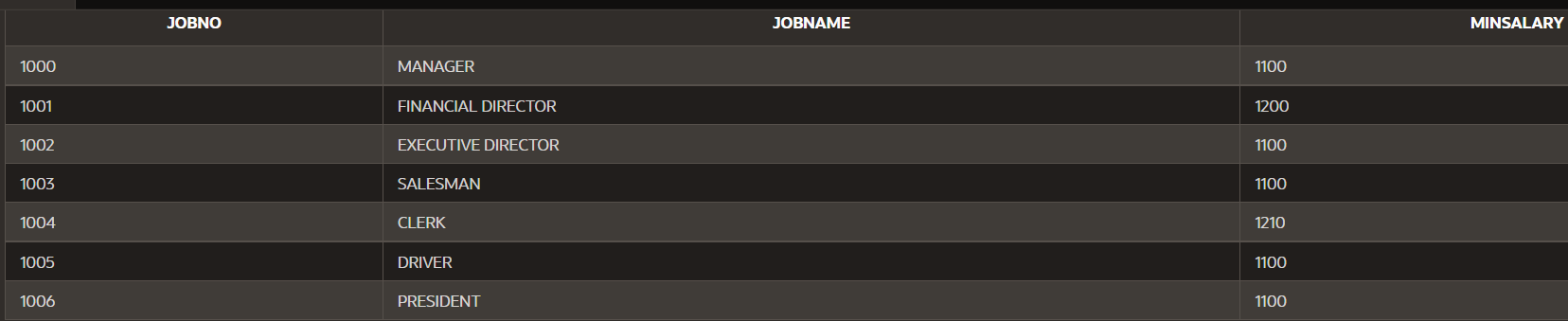
**исполнительного директора.**

UPDATE JOB

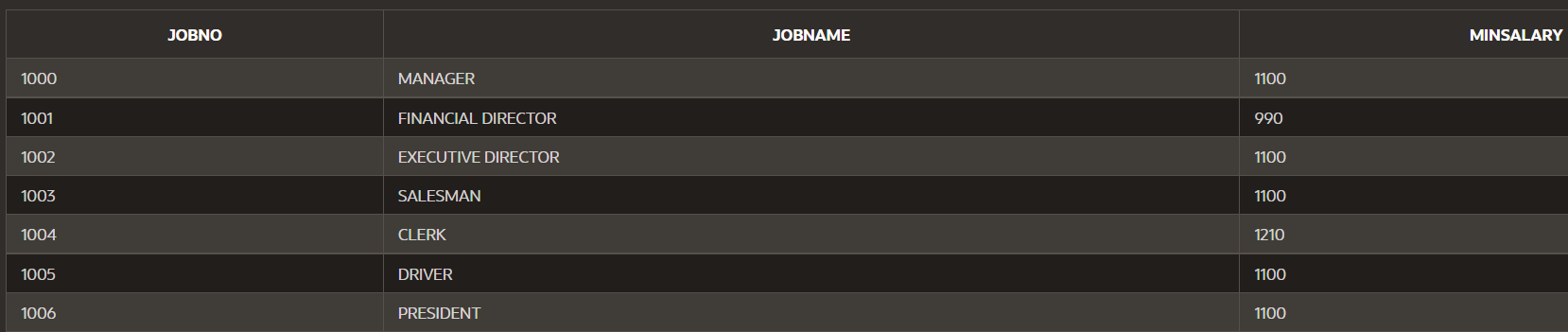
SET MINSALARY = 0.9 \* (SELECT MINSALARY FROM JOB

WHERE JOBNAME = 'EXECUTIVE DIRECTOR')

WHERE JOBNAME = 'FINANCIAL DIRECTOR';

**До**

**После**

****

**5. Приведите в таблице EMP имена служащих, начинающиеся на букву ‘J’, к нижнему**

**регистру.**

UPDATE EMP

SET EMPNAME = LOWER(EMPNAME)

WHERE EMPNAME LIKE 'J%';

**До**

****

**После**

****

**6. Измените в таблице EMP имена служащих, состоящие из двух слов, так, чтобы оба**

**слова в имени начинались с заглавной буквы, а продолжались прописными.**

UPDATE EMP

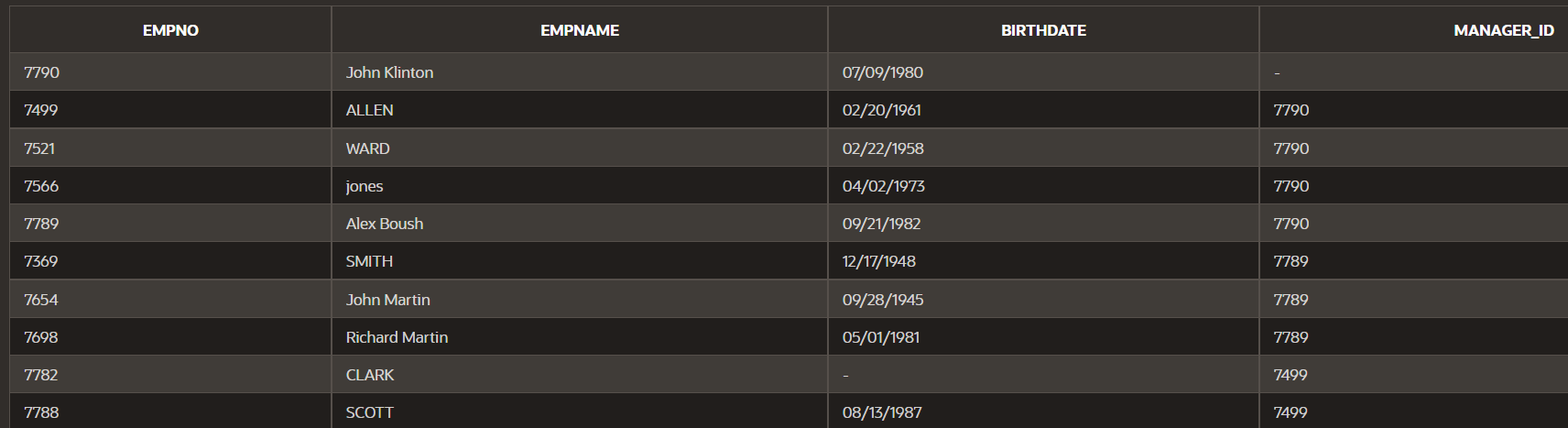
SET EMPNAME = INITCAP(EMPNAME)

WHERE length(EMPNAME) - length(replace(EMPNAME, ' ', '')) = 1;

**До**

****

**После**

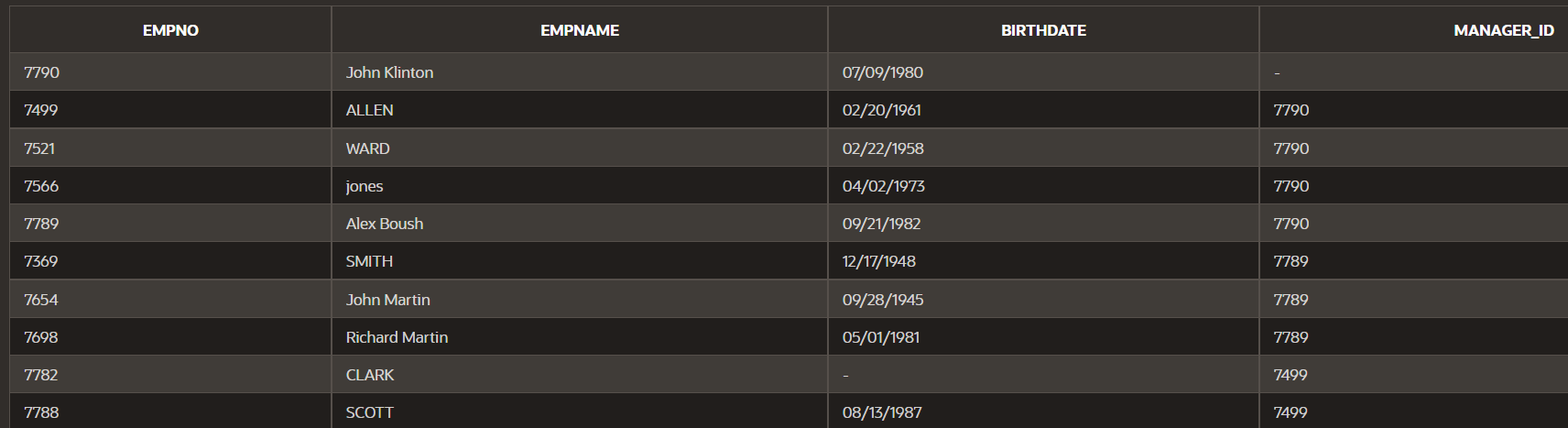
****

**7. Приведите в таблице EMP имена служащих к верхнему регистру.**

UPDATE EMP

SET EMPNAME = UPPER(EMPNAME);

**До**

****

**После**

****

**8. Перенесите отдел исследований (RESEARCH) в тот же город, в котором расположен**

**отдел продаж (SALES).**

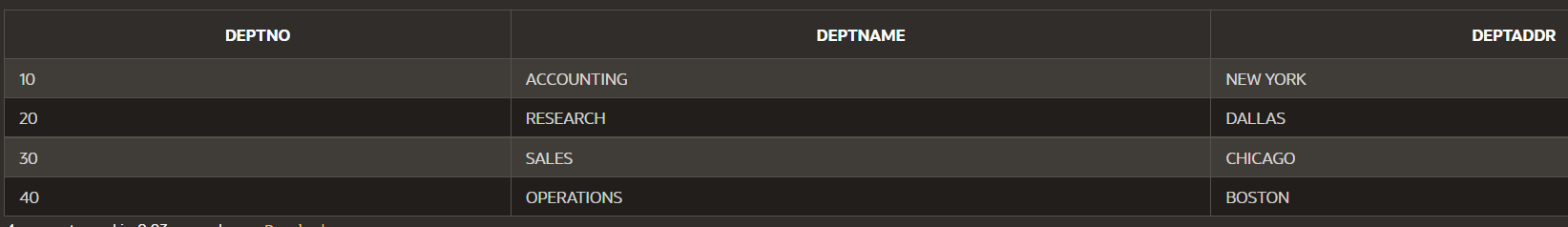
UPDATE DEPT

SET DEPTADDR = (SELECT DEPTADDR FROM DEPT

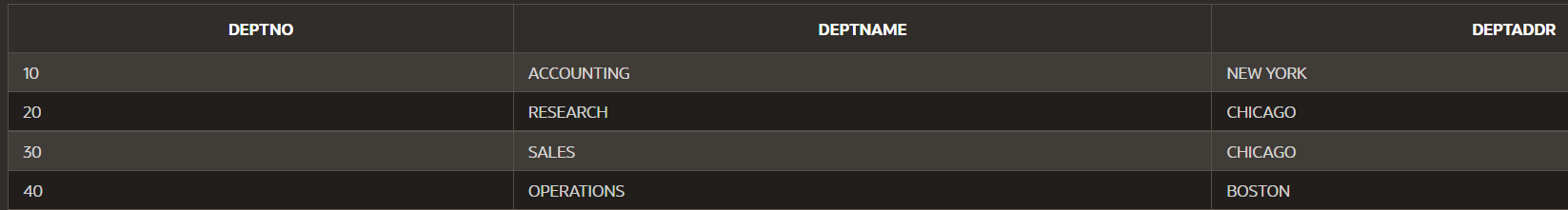
WHERE DEPTNAME = 'SALES')

WHERE DEPTNAME = 'RESEARCH';

**До**

****

**После**

****

**9. Добавьте нового сотрудника в таблицу EMP. Его имя и фамилия должны совпадать с**

**Вашими, записанными латинскими буквами согласно паспорту, дата рождения также**

**совпадает с Вашей.**

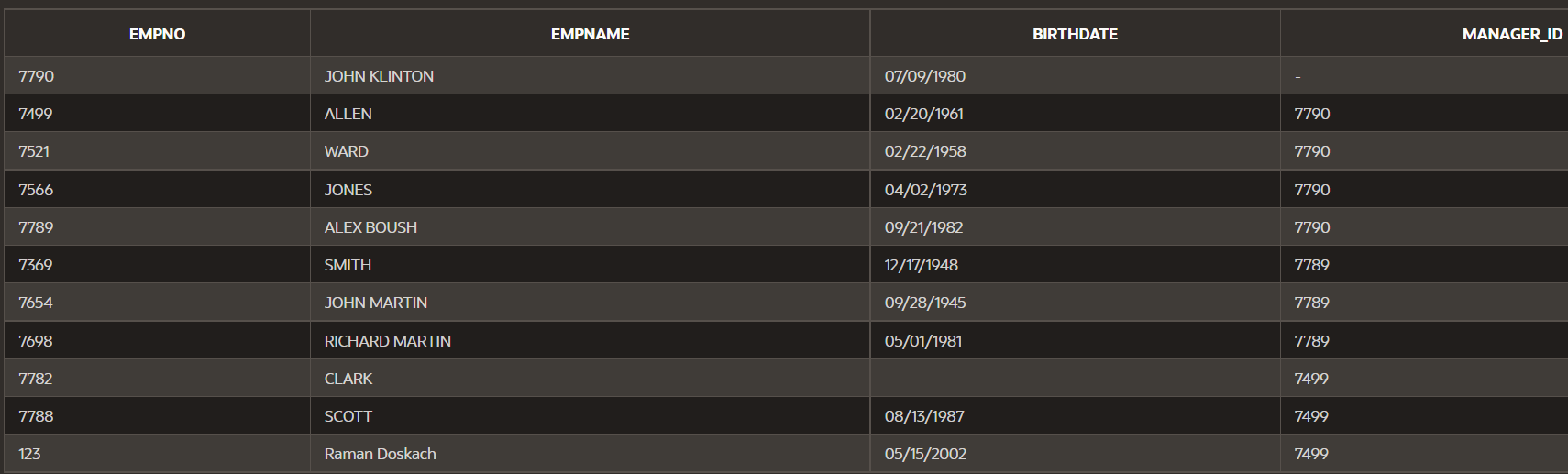
INSERT INTO EMP

VALUES (123, 'Raman Doskach', TO\_DATE('15-05-2002','dd-mm-yyyy'), 7499);

**До**

****

**После**

****

**10. Определите нового сотрудника (см. предыдущее задание) на работу в бухгалтерию**

**(отдел ACCOUNTING) начиная с текущей даты.**

INSERT INTO CAREER

VALUES (

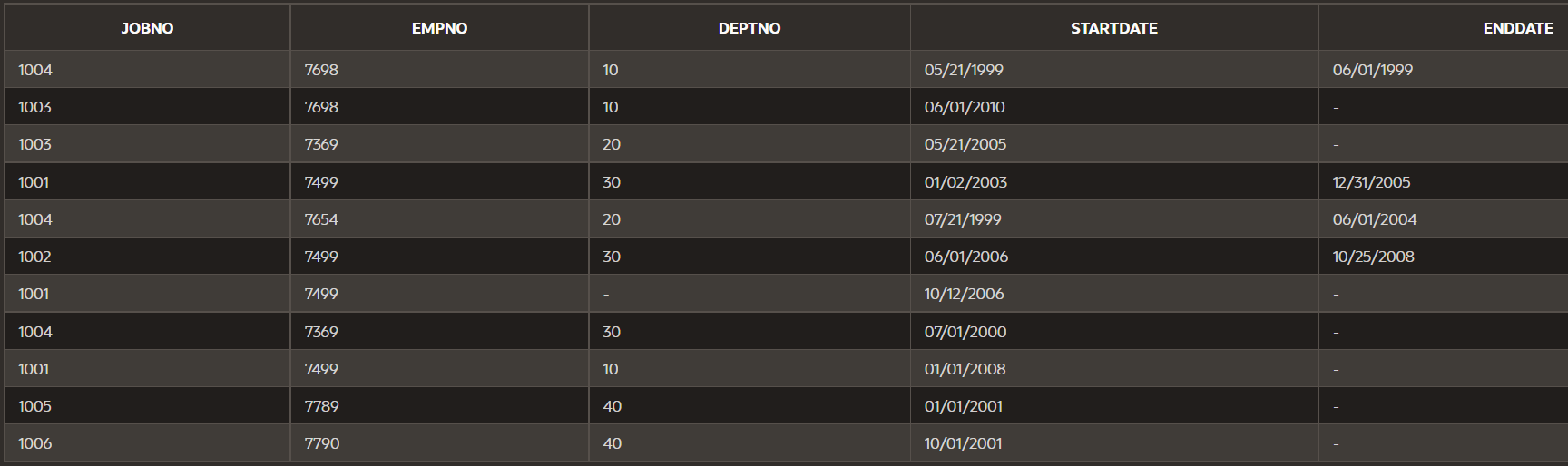
(SELECT JOBNO FROM JOB WHERE JOBNAME = 'CLERK'),

(SELECT EMPNO FROM EMP WHERE EMPNAME = 'Raman Doskach'),

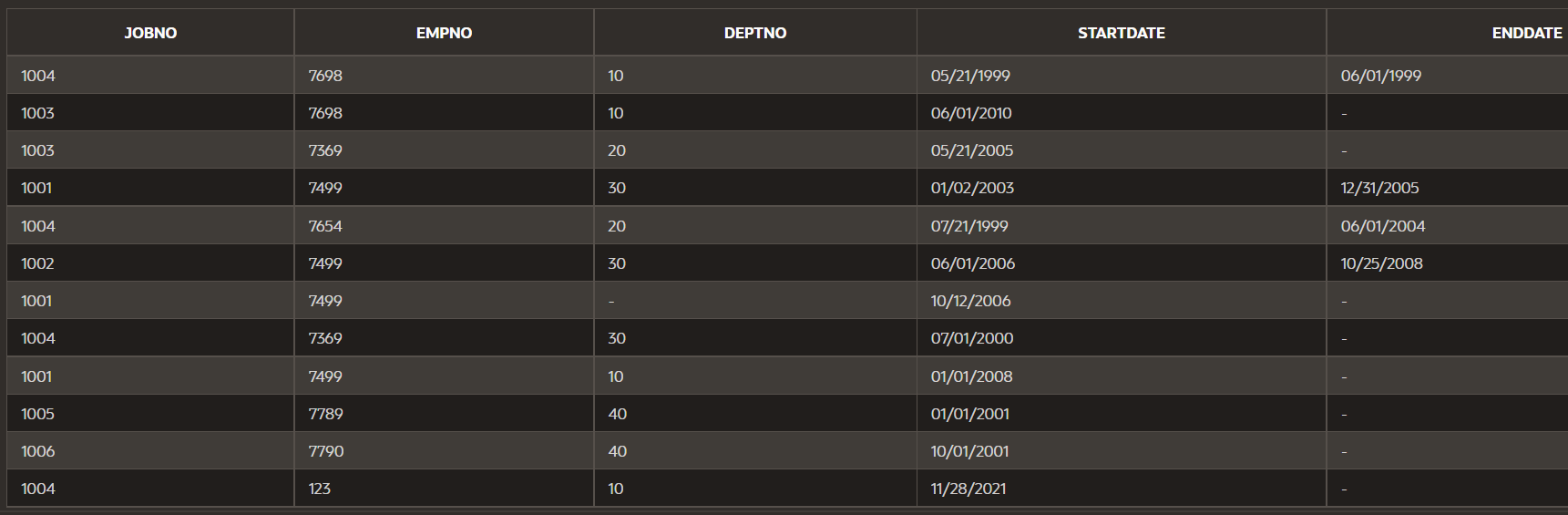
(SELECT DEPTNO FROM DEPT WHERE DEPTNAME = 'ACCOUNTING'),

TO\_DATE(SYSDATE, 'mm/dd/yyyy'), NULL);

**До**

****

**После**

****

**11. Удалите все записи из таблицы TMP\_EMP. Добавьте в нее информацию о**

**сотрудниках, которые работают клерками в настоящий момент.**

CREATE TABLE TMP\_EMP AS SELECT \* FROM EMP;

DELETE FROM TMP\_EMP;

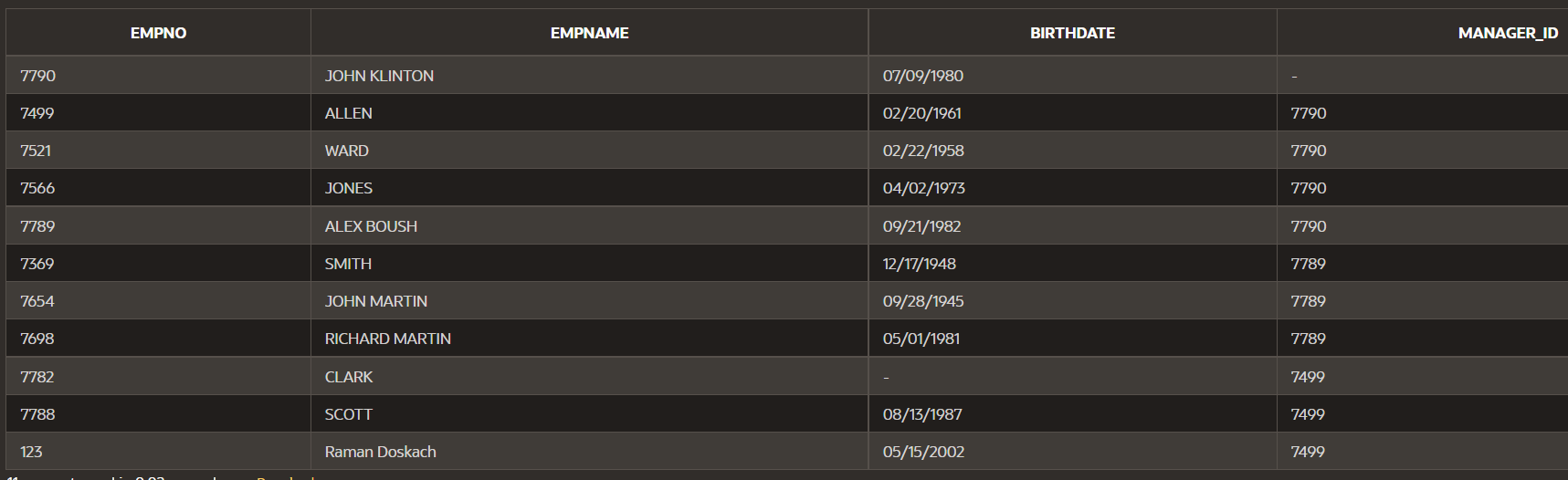
INSERT INTO TMP\_EMP

SELECT \* FROM EMP

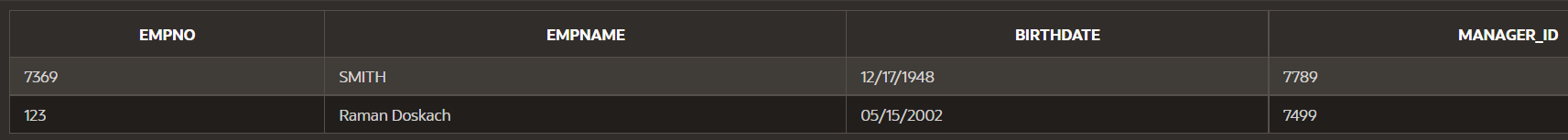
WHERE EMPNO IN (SELECT EMPNO FROM CAREER NATURAL JOIN JOB

WHERE JOBNAME = 'CLERK' AND **ENDDATE IS NULL**);

**До**



**После**

****

**12. Добавьте в таблицу TMP\_EMP информацию о тех сотрудниках, которые уже не**

**работают на предприятии, а в период работы занимали только одну должность.**

INSERT INTO TMP\_EMP

SELECT \* FROM EMP

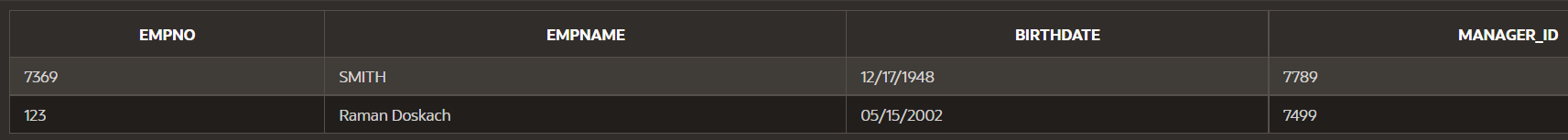
WHERE EMPNO IN (SELECT EMPNO FROM CAREER first\_c

WHERE ENDDATE IS NOT NULL AND

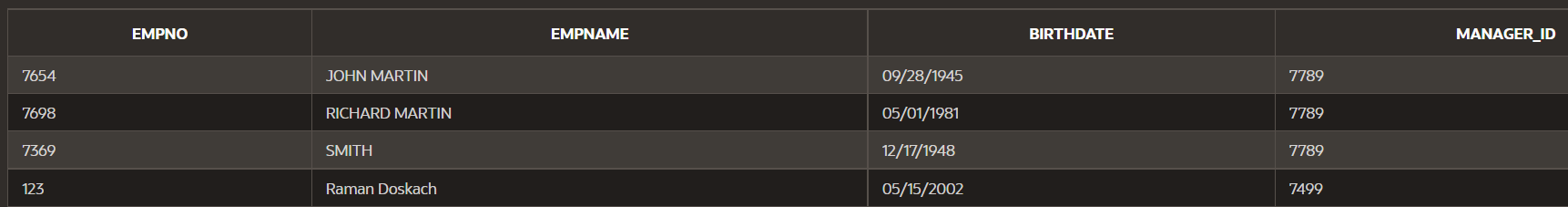
(SELECT COUNT(\*) FROM CAREER second\_c

WHERE ENDDATE IS NOT NULL AND first\_c.EMPNO = second\_c.EMPNO) = 1);

**До**

****

**После**

****

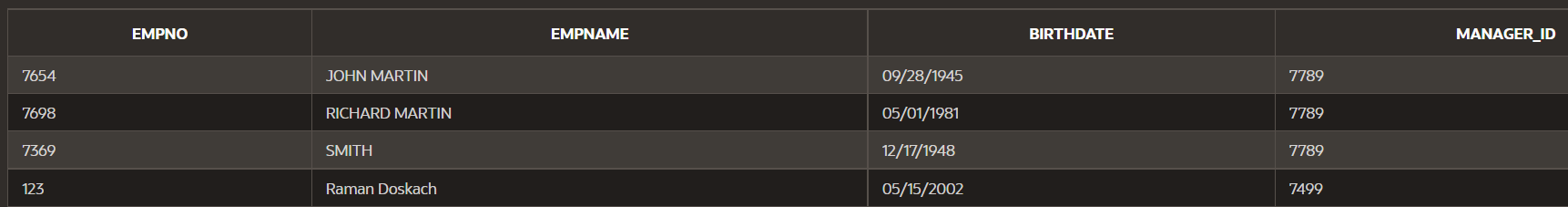
**13. Выполните тот же запрос для тех сотрудников, которые никогда не приступали к**

**работе на предприятии.**

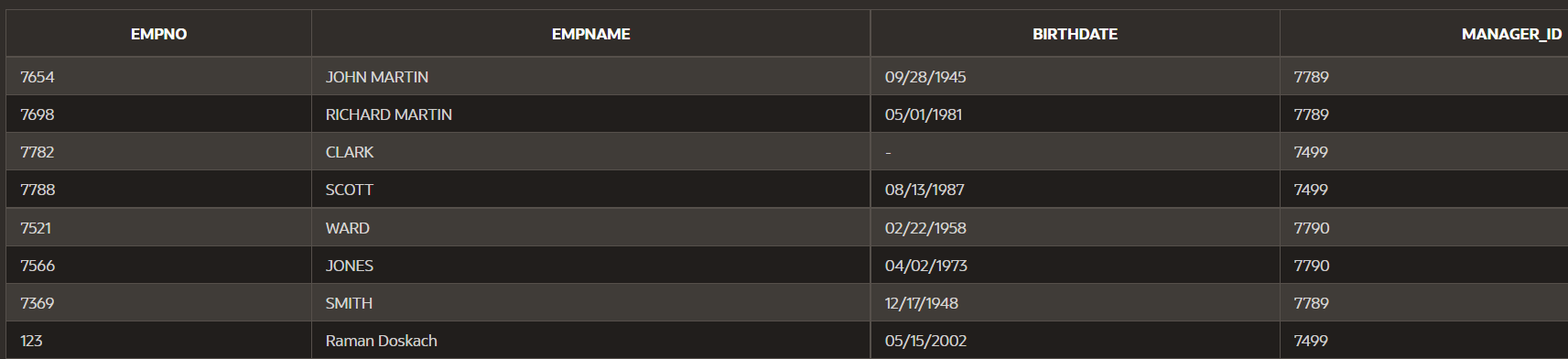
INSERT INTO TMP\_EMP

SELECT \* FROM EMP WHERE EMPNO NOT IN (SELECT EMPNO FROM CAREER);

**До**

****

**После**

****

**14. Удалите все записи из таблицы TMP\_JOB и добавьте в нее информацию по тем**

**специальностям, которые не используются в настоящий момент на предприятии.**

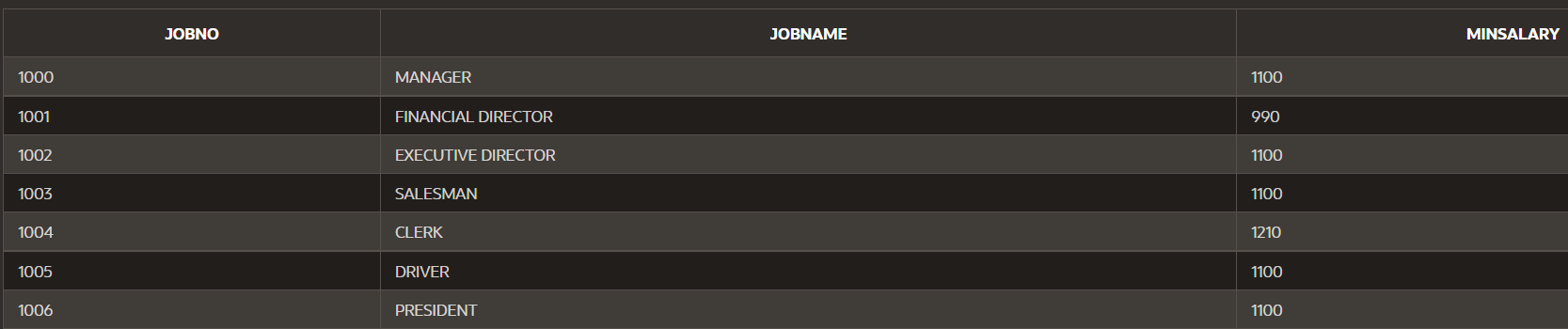
CREATE TABLE TMP\_JOB AS SELECT \* FROM JOB;

DELETE FROM TMP\_JOB;

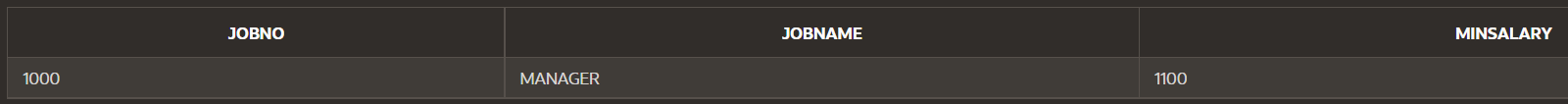
INSERT INTO TMP\_JOB SELECT \* FROM JOB

WHERE JOBNO NOT IN (SELECT JOBNO FROM CAREER);

**До**

****

**После**

****

**15. Начислите зарплату в размере 120% минимального должностного оклада всем**

**сотрудникам, работающим на предприятии. Зарплату начислять по должности,**

**занимаемой сотрудником в настоящий момент и отнести ее на прошлый месяц**

**относительно текущей даты.**

INSERT INTO TMP\_SALARY

(SELECT EMPNO,

EXTRACT(MONTH FROM ADD\_MONTH(SYSDATE, - 1)) AS MONTH,

EXTRACT(YEAR FROM ADD\_MONTH(SYSDATE,-1)) AS YEAR,

MINSALARY \* 1.2 AS SALVALUE

FROM CAREER NATURAL JOIN JOB WHERE ENDDATE IS NULL);

**До**

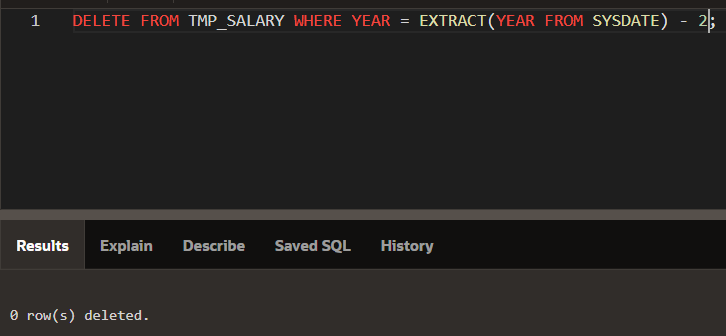
****

**После**

****

**16. Удалите данные о зарплате за прошлый год.**

DELETE FROM TMP\_SALARY WHERE YEAR = EXTRACT(YEAR FROM SYSDATE) - 1;

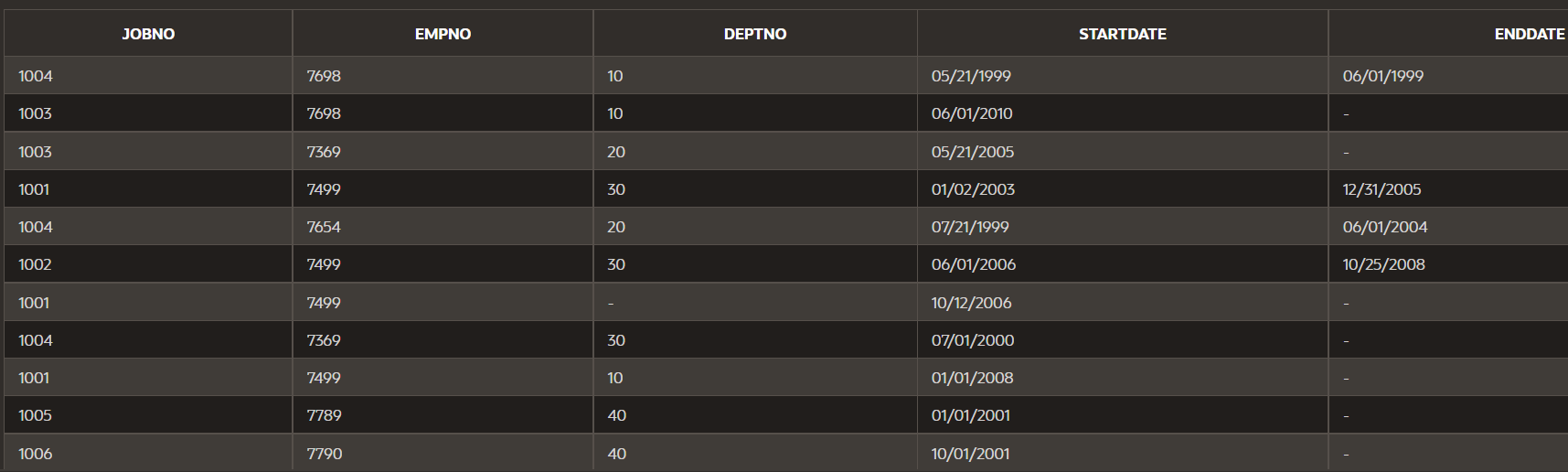
****

**17. Удалите информацию о карьере сотрудников, которые в настоящий момент уже не**

**работают на предприятии, но когда-то работали.**

DELETE FROM TMP\_CAREER WHERE ENDDATE IS NOT NULL;

**До**

****

**После**

****

**18. Удалите информацию о начисленной зарплате сотрудников, которые в настоящий**

**момент уже не работают на предприятии (можно использовать результаты работы**

**предыдущего запроса)**

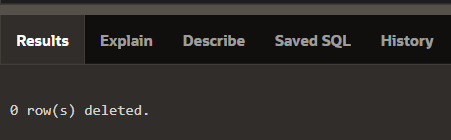
DELETE FROM TMP\_SALARY

WHERE EMPNO IN (SELECT EMPNO FROM CAREER WHERE ENDDATE IS NOT NULL);

**До**

****

**После**

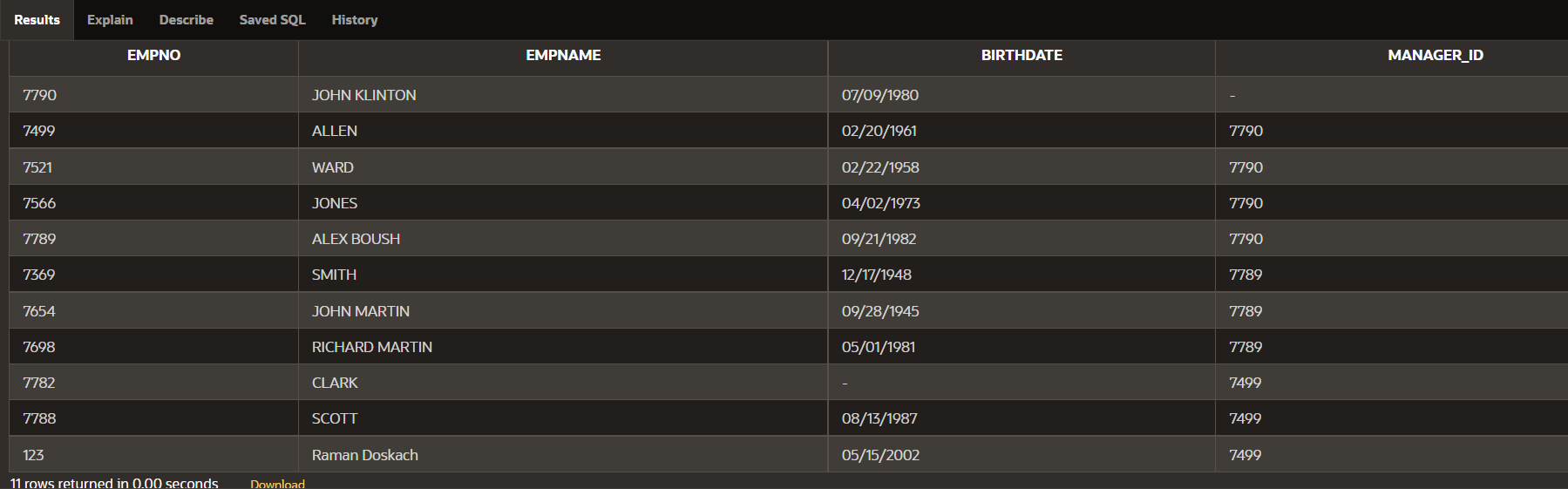
****

**19. Удалите записи из таблицы EMP для тех сотрудников, которые никогда не**

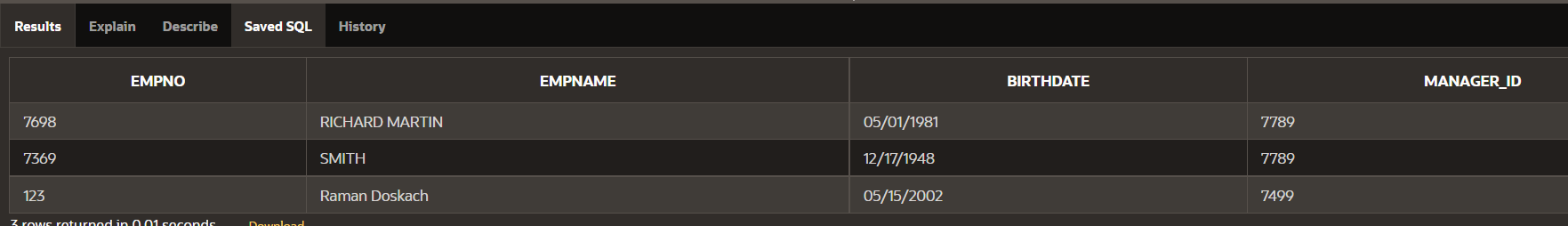
**приступали к работе на предприятии.**

DELETE FROM TMP\_EMP WHERE EMPNO NOT IN (SELECT EMPNO FROM CAREER);

**До**

****

**После**

****

**2 часть**

**15. Расписание занятий**

**1. Добавить новую аудиторию. Тип аудитории определить по наименьшему**

**количеству аудиторий данного типа.**

INSERT INTO CLASSROOM  
VALUES (1, 'BSU', 100,  
 (SELECT Class\_Type FROM CLASSROOM  
 GROUP BY Class\_Type  
 HAVING COUNT(Class\_Type) =  
 (SELECT MIN(*CNT*) FROM

(SELECT COUNT(Class\_Type) AS *CNT* FROM CLASSROOM GROUP BY Class\_Type)  
 )  
 ));

**2. В новую аудиторию перенести занятия из аудиторий с максимальной загрузкой.**

UPDATE CLASSROOM  
SET lesson = (SELECT lesson FROM CLASSROOM WHERE session\_count\_per\_week = MAX(session\_count))  
WHERE clussroom\_number = 1;

**3. Удалить сведения о предметах, по которым не проводятся аудиторные занятия. В**

**архиве сохранить информацию: название предмета, преподаватель, вид занятий.**

DELETE FROM SUBJECT WHERE 'time' IS NULL;