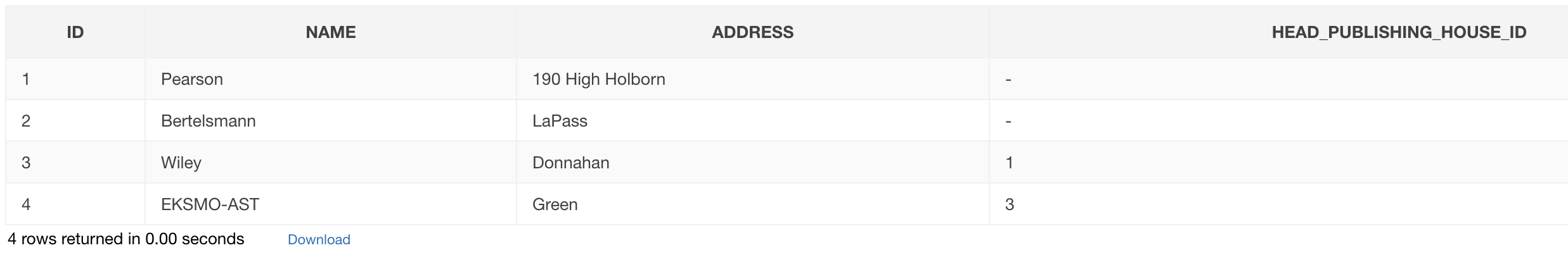
**--1) Составьте на выборку данных с использованием рефлексивного соединения для таблицы из задания 5 лабораторной работы №2 (academy.oracle.com\iLearning\2013-2014 Oracle Academy Database Programming with SQL – Student\Section 3 Executing Database Joins\Self Joins and Hierarchical Queries).**

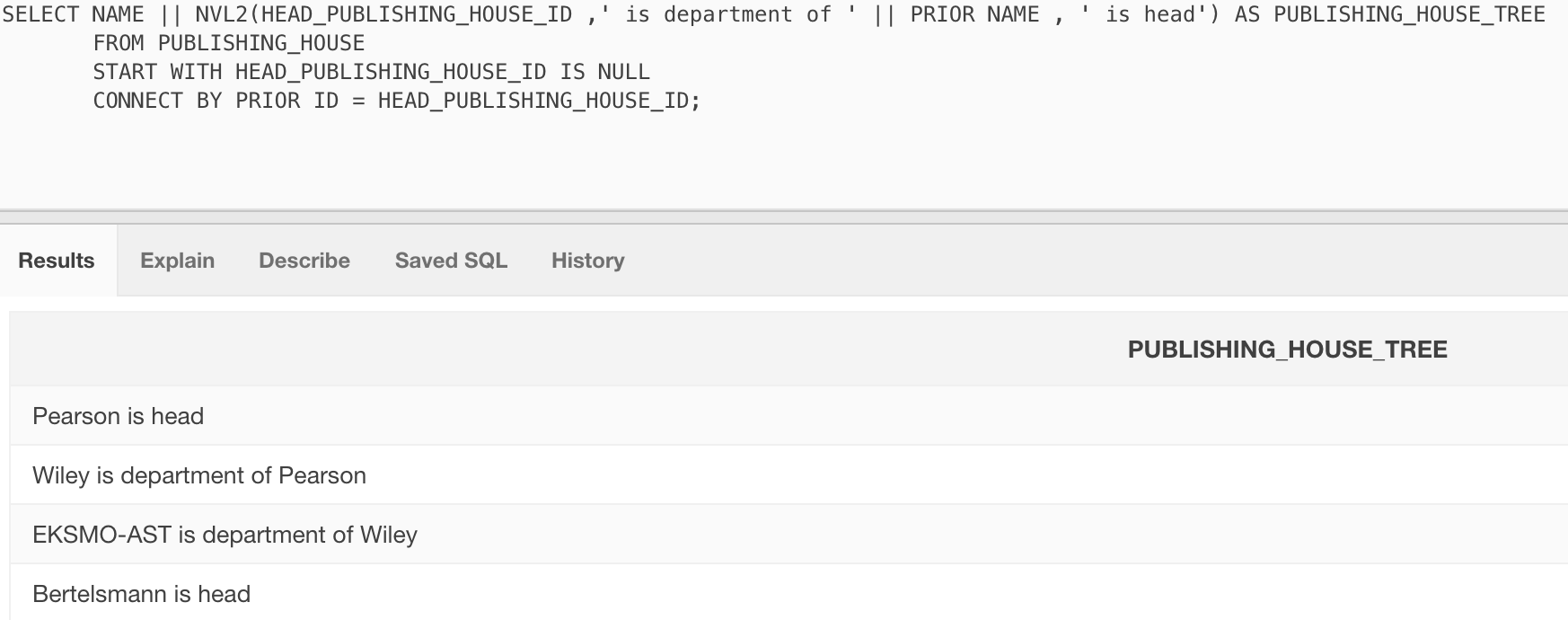


SELECT NAME || NVL2(HEAD\_PUBLISHING\_HOUSE\_ID ,' is department of ' || PRIOR NAME , ' is head') AS PUBLISHING\_HOUSE\_TREE

FROM PUBLISHING\_HOUSE

START WITH HEAD\_PUBLISHING\_HOUSE\_ID IS NULL

CONNECT BY PRIOR ID = HEAD\_PUBLISHING\_HOUSE\_ID;



*Составьте запросы на выборку данных с использованием следующих операторов, конструкций функций языка SQL:*

**--2) Простого оператора CASE ();**

SELECT NAME,

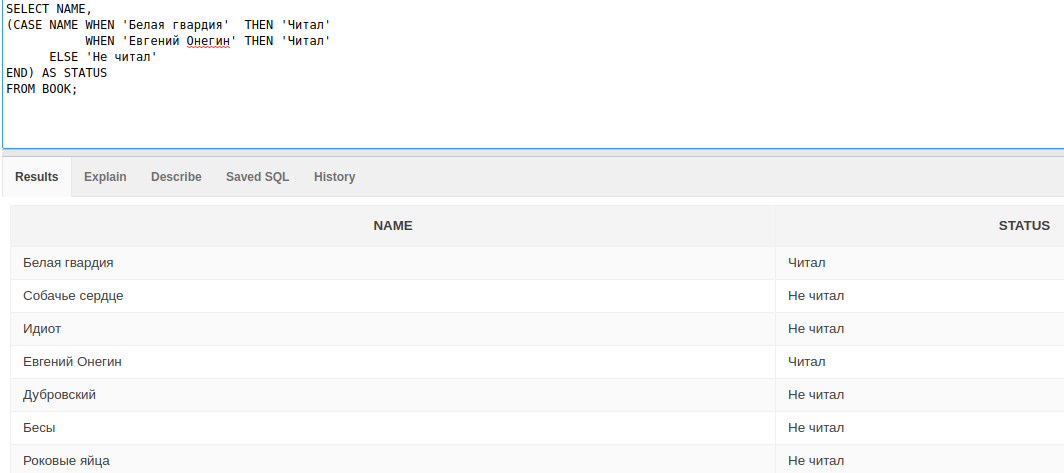
(CASE NAME WHEN 'Белая гвардия' THEN 'Читал'

WHEN 'Евгений Онегин' THEN 'Читал'

ELSE 'Не читал'

END) AS STATUS

FROM BOOK;



**--3) Поискового оператора CASE();**

SELECT b.NAME,

(CASE

WHEN SUM(o.AMOUNT) <= 3 THEN 'Мало, купите еще'

WHEN SUM(o.AMOUNT) >= 4 AND SUM(o.AMOUNT) < 20 THEN 'Достаточно'

ELSE 'Очень много, книг'

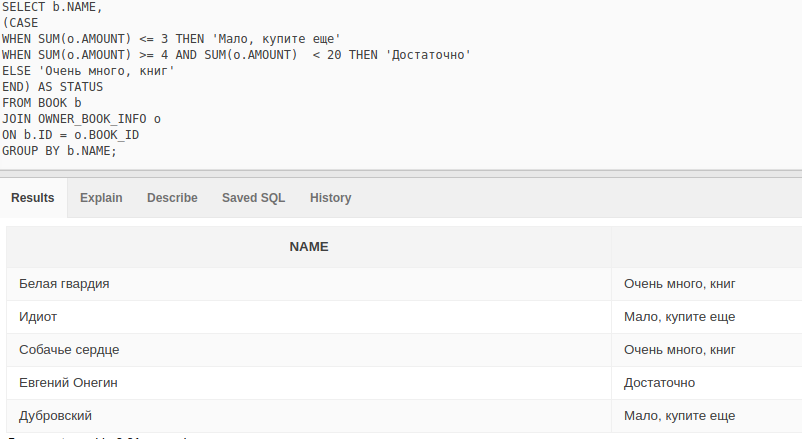
END) AS STATUS

FROM BOOK b

JOIN OWNER\_BOOK\_INFO o

ON b.ID = o.BOOK\_ID

GROUP BY b.NAME;



**-4) Оператора WITH();**

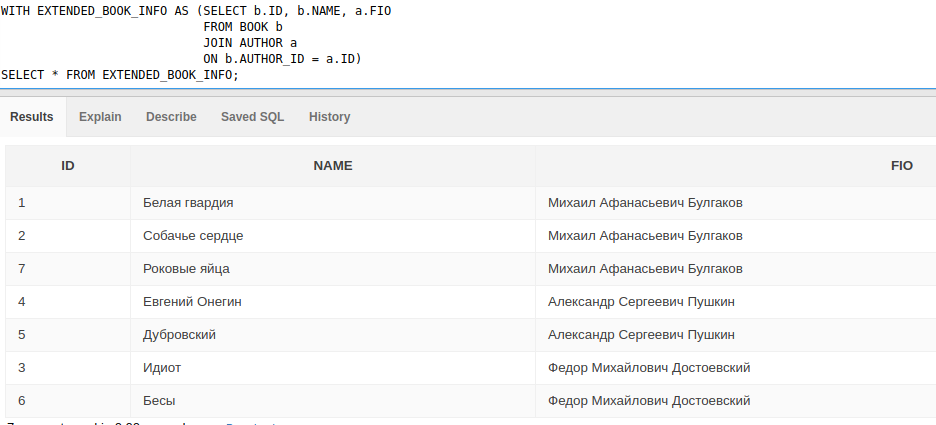
WITH EXTENDED\_BOOK\_INFO AS (SELECT b.ID, b.NAME, a.FIO

FROM BOOK b

JOIN AUTHOR a

ON b.AUTHOR\_ID = a.ID)

SELECT \* FROM EXTENDED\_BOOK\_INFO;



**--5) встроенного представления();**

CREATE OR REPLACE VIEW BOOK\_INFO AS

(SELECT b.ID, b.NAME, a.FIO AS AUTHOR, pub.NAME AS PUBLISHING\_HOUSE, pub.ADDRESS as PUBLISHING\_HOUSE\_ADDRESS

FROM BOOK b

JOIN AUTHOR a

ON b.AUTHOR\_ID = a.ID

JOIN PUBLISHING\_HOUSE pub

on b.PUBLISHING\_HOUSE\_ID = pub.ID);

SELECT b.ID,

b.NAME,

b.AUTHOR,

b.PUBLISHING\_HOUSE\_ADDRESS,

i\_d.AMOUNT,

i.SELLING\_PRICE,

i.BUYING\_PRICE,

i.OPERATION\_DESCRIPTION AS TYPE

FROM BOOK\_INFO b

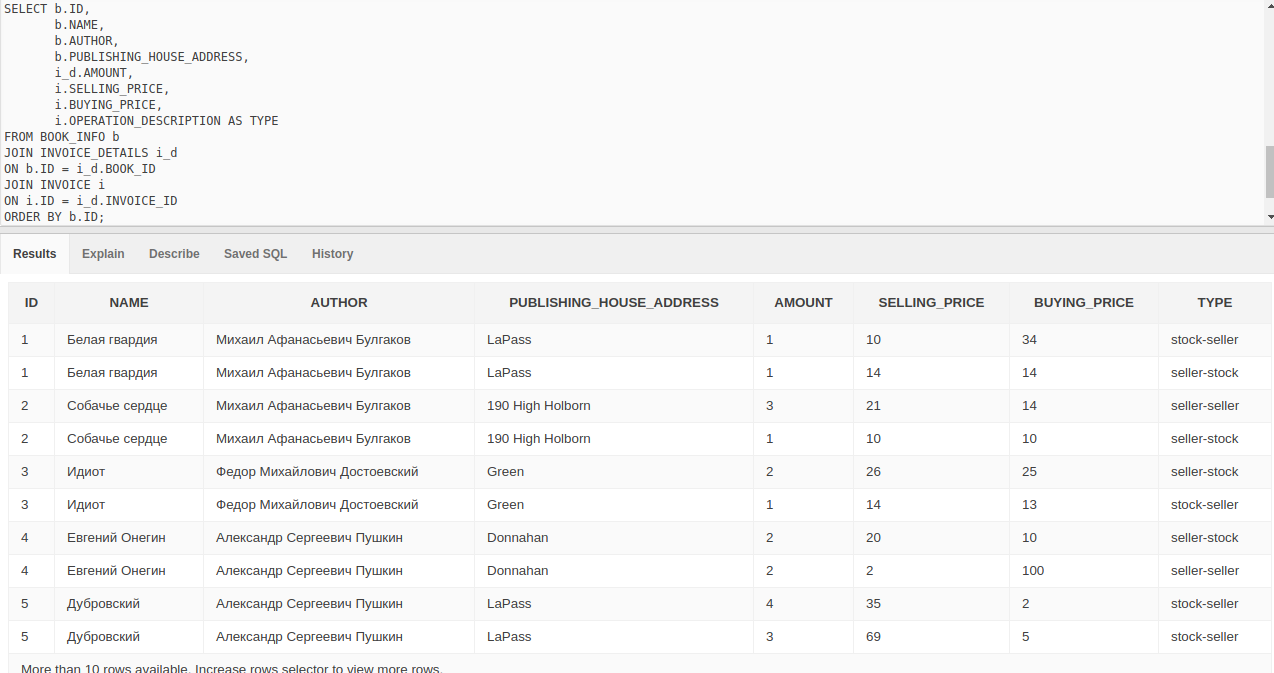
JOIN INVOICE\_DETAILS i\_d

ON b.ID = i\_d.BOOK\_ID

JOIN INVOICE i

ON i.ID = i\_d.INVOICE\_ID

ORDER BY b.ID;

****

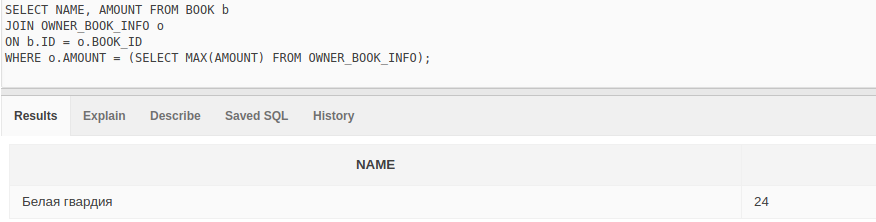
**--6) некоррелированного запроса(academy.oracle.com\iLearning\2013-2014 Oracle Academy Database Programming with SQL – Student\Section 6 Creating Subqueries);**

SELECT NAME, AMOUNT FROM BOOK b

JOIN OWNER\_BOOK\_INFO o

ON b.ID = o.BOOK\_ID

WHERE o.AMOUNT = (SELECT MAX(AMOUNT) FROM OWNER\_BOOK\_INFO);



**--7) коррелированного запроса(academy.oracle.com\iLearning\2013-2014 Oracle Academy Database Programming with SQL – Student\Section 6 Creating Subqueries).);**

SELECT \* FROM OWNER o

JOIN OWNER\_INVOICE o\_i

ON o.ID = o\_i.OWNER\_ID

JOIN INVOICE i

ON i.ID = o\_i.INVOICE\_ID

WHERE o\_i.INVOICE\_ID IN (SELECT i1.ID FROM INVOICE i1

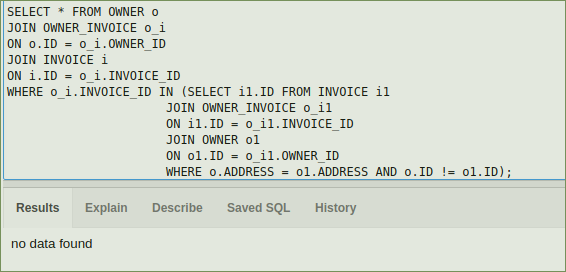
JOIN OWNER\_INVOICE o\_i1

ON i1.ID = o\_i1.INVOICE\_ID

JOIN OWNER o1

ON o1.ID = o\_i1.OWNER\_ID

WHERE o.ADDRESS = o1.ADDRESS AND o.ID != o1.ID)



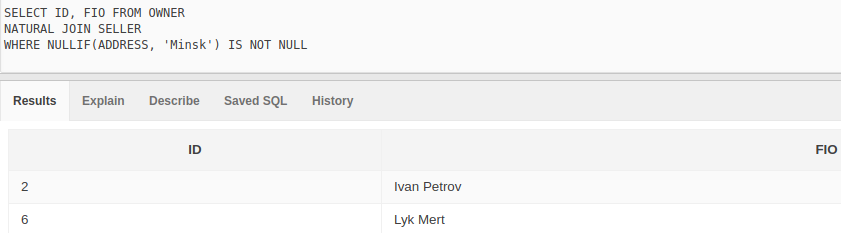
из одного города

**--8) функции NULLIF (academy.oracle.com\iLearning\2013-2014 Oracle Academy Database Programming with SQL – Student\Section 2 Using Single-Row Functions);**

SELECT ID, FIO FROM OWNER

NATURAL JOIN SELLER

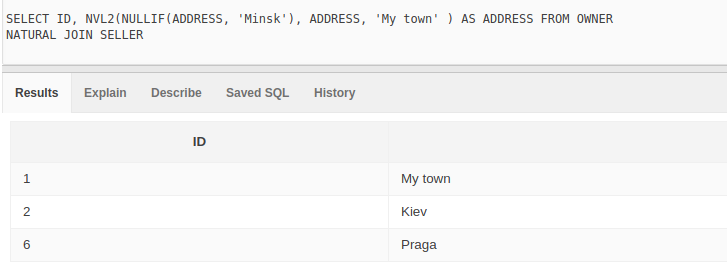
WHERE NULLIF(ADDRESS, 'Minsk') IS NOT NULL



**--9) функции NVL2 (academy.oracle.com\iLearning\2013-2014 Oracle Academy Database Programming with SQL – Student\Section 2 Using Single-Row Functions);**

SELECT ID, NVL2(NULLIF(ADDRESS, 'Minsk'), ADDRESS, 'My town' ) AS ADDRESS FROM OWNER

NATURAL JOIN SELLER



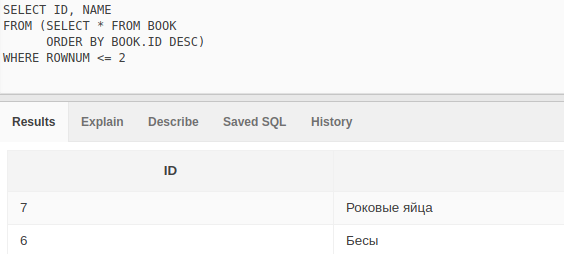
**--10) TOP-N анализа();**

SELECT ID, NAME

FROM (SELECT \* FROM BOOK

ORDER BY BOOK.ID DESC)

WHERE ROWNUM <= 2



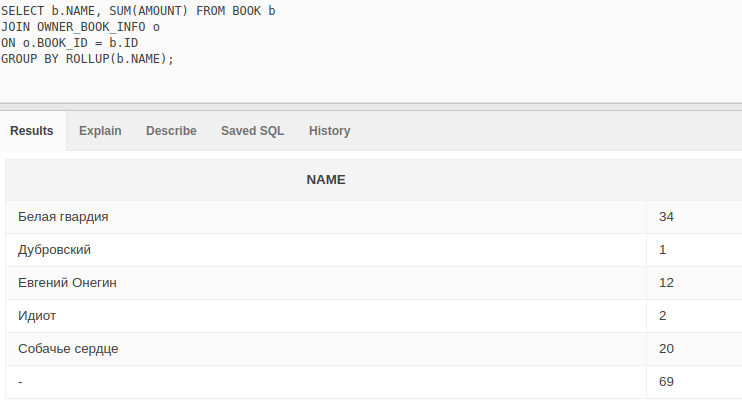
**-11) функции ROLLUP().**

SELECT b.NAME, SUM(AMOUNT) FROM BOOK b

JOIN OWNER\_BOOK\_INFO o

ON o.BOOK\_ID = b.ID

GROUP BY ROLLUP(b.NAME);



**--12) Составьте запрос на использование оператора MERGE языка манипулирования данными.**

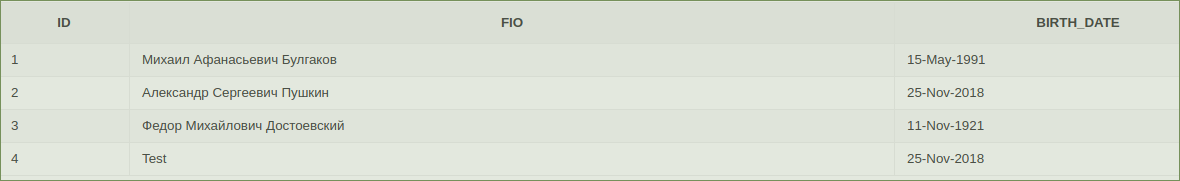
CREATE TABLE AUTHOR\_TMP AS SELECT \* FROM AUTHOR

SELECT \* FROM AUTHOR\_TMP

UPDATE AUTHOR\_TMP

SET BIRTH\_DATE = SYSDATE

WHERE ID = 2



MERGE INTO AUTHOR a

USING ( SELECT \* FROM AUTHOR\_TMP) a1

ON (a.ID = a1.ID)

WHEN MATCHED THEN UPDATE SET a.BIRTH\_DATE = a1.BIRTH\_DATE

WHEN NOT MATCHED THEN INSERT (a.ID, a.FIO, a.BIRTH\_DATE)

VALUES (a1.ID, a1.FIO, a1.BIRTH\_DATE)

