 eМИНИСТЕРСТВО науки и высшего ОБРАЗОВАНИЯ РОссИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(национальный исследовательский университет)»

Институт №3 «Системы управления, информатика и электроэнергетика»

Кафедра № 304 «Вычислительные машины, системы и сети»

Базы данных

Отчет по лабораторной работе № 6

«Операторы модификации»

Выполнил студент группы M3О-311Б-22

Пономарев Н.А.

Проверил доцент, к.т.н., Ткачев О.А.

Москва 2024 г.

Задача 1. Создайте таблицу EMP(employee\_id, first\_name , last\_name ,hire\_date, rating\_e, working, layer) и заполните данными о сотрудниках, работающих в отделе 80. Столбцу working присвоить значение, равное количеству полных лет, которые проработал сотрудник. А значение столбца layer зависит от значения столбца rating\_e. Если rating\_e равен 5 то layer= ‘A’, если rating\_e равен 4 или 3 то layer= ‘B’, у остальных сотрудников layer= ‘C’.

> create table emp as

select

e.employee\_id,

e.first\_name,

e.last\_name,

e.hire\_date,

e.rating\_e,

extract(year from age(now(), e.hire\_date)) as working,

case

when e.rating\_e = 5 then 'A'

when e.rating\_e in (3, 4) then 'B'

else 'C'

end as layer

from

employees e

where e.department\_id = 80

34 row(s) modified.

> select \* from emp

employee\_id|first\_name |last\_name |hire\_date |rating\_e|working|layer|

-----------+-----------+----------+----------+--------+-------+-----+

145|John |Russell |1996-10-01| 3| 28|B |

146|Karen |Partners |1997-01-05| | 27|C |

147|Alberto |Errazuriz |1997-03-10| 2| 27|C |

148|Gerald |Cambrault |1999-10-15| 5| 25|A |

149|Eleni |Zlotkey |2000-01-29| 3| 24|B |

150|Peter |Tucker |1997-01-30| 3| 27|B |

151|David |Bernstein |1997-03-24| 3| 27|B |

153|Christopher|Olsen |1998-03-30| 5| 26|A |

154|Nanette |Cambrault |1998-12-09| 3| 25|B |

155|Oliver |Tuvault |1999-11-23| 5| 24|A |

156|Janette |King |1996-01-30| 1| 28|C |

157|Patrick |Sully |1996-03-04| 2| 28|C |

158|Allan |McEwen |1996-08-01| 2| 28|C |

159|Lindsey |Smith |1997-03-10| 3| 27|B |

160|Louise |Doran |1997-12-15| 5| 26|A |

161|Sarath |Sewall |1998-11-03| 1| 25|C |

162|Clara |Vishney |1997-11-11| 2| 26|C |

163|Danielle |Greene |1999-03-19| 5| 25|A |

164|Mattea |Marvins |2000-01-24| 5| 24|A |

165|David |Lee |2000-02-23| 5| 24|A |

166|Sundar |Ande |2000-03-24| 2| 24|C |

167|Amit |Banda |2000-04-21| 3| 24|B |

168|Lisa |Ozer |1997-03-11| 2| 27|C |

169|Harrison |Bloom |1998-03-23| 3| 26|B |

170|Tayler |Fox |1998-01-24| 3| 26|B |

171|William |Smith |1999-02-23| 2| 25|C |

172|Elizabeth |Bates |1999-03-24| 2| 25|C |

173|Sundita |Kumar |2000-04-21| 2| 24|C |

152|Peter |Hall |1997-08-20| 3| 27|B |

174|Ellen |Abel |1996-05-11| 1| 28|C |

175|Alyssa |Hutton |1997-03-19| 3| 27|B |

176|Jonathon |Taylor |1998-03-24| 3| 26|B |

177|Jack |Livingston|1998-04-23| 2| 26|C |

179|Charles |Johnson |2000-01-04| 4| 24|B |

34 row(s) fetched.

Задача 2. Увеличить на 1 rating\_e сотрудников, которые осуществили продажи на сумму более 1000000 и имеют rating\_e < 5.

> with order\_totals as (

select

o.salesman\_id,

sum(oi.quantity \* oi.unit\_price) as total\_sales

from orders o

join order\_items oi on o.order\_id = oi.order\_id

group by o.salesman\_id

),

employee\_sales as (

select

e.employee\_id,

coalesce(ot.total\_sales, 0) as total\_sales

from employees e

left join order\_totals ot on e.employee\_id = ot.salesman\_id

)

update emp

set rating\_e = rating\_e + 1

from employee\_sales es

where emp.employee\_id = es.employee\_id

and

es.total\_sales > 1000000

and

emp.rating\_e < 5

returning \*

employee\_id|first\_name|last\_name|hire\_date |rating\_e|working|layer|employee\_id|total\_sales|

-----------+----------+---------+----------+--------+-------+-----+-----------+-----------+

162|Clara |Vishney |1997-11-11| 3| 26|C | 162| 1763220.00|

152|Peter |Hall |1997-08-20| 4| 27|B | 152| 1440050.00|

179|Charles |Johnson |2000-01-04| 5| 24|B | 179| 1199100.00|

3 row(s) fetched.

> update emp

set layer = CASE

WHEN rating\_e = 5 THEN 'A'

WHEN rating\_e IN (3, 4) THEN 'B'

ELSE 'C'

end

where emp.employee\_id != 0

returning \*

employee\_id|first\_name |last\_name |hire\_date |rating\_e|working|layer|

-----------+-----------+----------+----------+--------+-------+-----+

145|John |Russell |1996-10-01| 3| 28|B |

146|Karen |Partners |1997-01-05| | 27|C |

147|Alberto |Errazuriz |1997-03-10| 2| 27|C |

148|Gerald |Cambrault |1999-10-15| 5| 25|A |

149|Eleni |Zlotkey |2000-01-29| 3| 24|B |

150|Peter |Tucker |1997-01-30| 3| 27|B |

151|David |Bernstein |1997-03-24| 3| 27|B |

153|Christopher|Olsen |1998-03-30| 5| 26|A |

154|Nanette |Cambrault |1998-12-09| 3| 25|B |

155|Oliver |Tuvault |1999-11-23| 5| 24|A |

156|Janette |King |1996-01-30| 1| 28|C |

157|Patrick |Sully |1996-03-04| 2| 28|C |

158|Allan |McEwen |1996-08-01| 2| 28|C |

159|Lindsey |Smith |1997-03-10| 3| 27|B |

160|Louise |Doran |1997-12-15| 5| 26|A |

161|Sarath |Sewall |1998-11-03| 1| 25|C |

163|Danielle |Greene |1999-03-19| 5| 25|A |

164|Mattea |Marvins |2000-01-24| 5| 24|A |

165|David |Lee |2000-02-23| 5| 24|A |

166|Sundar |Ande |2000-03-24| 2| 24|C |

167|Amit |Banda |2000-04-21| 3| 24|B |

168|Lisa |Ozer |1997-03-11| 2| 27|C |

169|Harrison |Bloom |1998-03-23| 3| 26|B |

170|Tayler |Fox |1998-01-24| 3| 26|B |

171|William |Smith |1999-02-23| 2| 25|C |

172|Elizabeth |Bates |1999-03-24| 2| 25|C |

173|Sundita |Kumar |2000-04-21| 2| 24|C |

174|Ellen |Abel |1996-05-11| 1| 28|C |

175|Alyssa |Hutton |1997-03-19| 3| 27|B |

176|Jonathon |Taylor |1998-03-24| 3| 26|B |

177|Jack |Livingston|1998-04-23| 2| 26|C |

162|Clara |Vishney |1997-11-11| 3| 26|B |

152|Peter |Hall |1997-08-20| 4| 27|B |

179|Charles |Johnson |2000-01-04| 5| 24|A |

34 row(s) fetched.

Задача 3. Добавить в таблицу Employees\_Copy столбец emp\_sales и присвоить ему значение общей стоимости продаж осуществленных каждым сотрудником.

**create** **table** Employees\_Copy **as**

**select**

\*

**from**

employees e;

**ALTER** **TABLE** Employees\_Copy

**ADD** **COLUMN** emp\_sales **NUMERIC**(15, 2) **DEFAULT** 0;

**with** order\_totals **as** (

**select**

o.salesman\_id,

**sum**(oi.quantity \* oi.unit\_price) **as** total\_sales

**from** orders o

**join** order\_items oi **on** o.order\_id = oi.order\_id

**group** **by** o.salesman\_id

),

employee\_sales **as** (

**select**

e.employee\_id,

**coalesce**(ot.total\_sales, 0) **as** total\_sales

**from** employees e

**left** **join** order\_totals ot **on** e.employee\_id = ot.salesman\_id

)

**UPDATE** Employees\_Copy *ec*

**SET** emp\_sales = *es*.total\_sales

**FROM** employee\_sales *es*

**WHERE** *ec*.employee\_id = *es*.employee\_id

**returning** ec.employee\_id, ec.emp\_sales;

> with order\_totals as (

select

o.salesman\_id,

sum(oi.quantity \* oi.unit\_price) as total\_sales

from orders o

join order\_items oi on o.order\_id = oi.order\_id

group by o.salesman\_id

),

employee\_sales as (

select

e.employee\_id,

coalesce(ot.total\_sales, 0) as total\_sales

from employees e

left join order\_totals ot on e.employee\_id = ot.salesman\_id

)

UPDATE Employees\_Copy ec

SET emp\_sales = es.total\_sales

FROM employee\_sales es

WHERE ec.employee\_id = es.employee\_id

returning ec.employee\_id, ec.emp\_sales

employee\_id|emp\_sales |

-----------+----------+

100| 0.00|

101| 0.00|

102| 0.00|

103| 0.00|

104| 0.00|

106| 0.00|

107| 0.00|

108| 0.00|

109| 0.00|

110| 0.00|

111| 0.00|

112| 0.00|

113| 0.00|

115| 0.00|

118| 0.00|

119| 0.00|

120| 0.00|

121| 0.00|

122| 0.00|

123| 0.00|

124| 0.00|

128| 0.00|

129| 0.00|

130| 0.00|

131| 0.00|

133| 0.00|

134| 0.00|

135| 0.00|

136| 0.00|

137| 0.00|

138| 0.00|

139| 0.00|

140| 0.00|

141| 0.00|

142| 0.00|

143| 0.00|

144| 0.00|

145| 407170.00|

146| 0.00|

147| 0.00|

148| 63280.00|

149| 379180.00|

150| 292980.00|

151| 0.00|

153| 973650.00|

154| 364540.00|

155| 526380.00|

156| 0.00|

157| 191100.00|

158| 60200.00|

159| 105980.00|

160| 544000.00|

161| 504730.00|

162|1763220.00|

163| 0.00|

164|2776310.00|

165| 42500.00|

166| 248020.00|

167| 0.00|

168| 0.00|

169| 0.00|

170| 82680.00|

171| 577600.00|

172| 0.00|

173| 0.00|

114| 0.00|

116| 0.00|

117| 0.00|

126| 0.00|

127| 0.00|

132| 0.00|

152|1440050.00|

174| 368890.00|

175| 100060.00|

176| 237250.00|

177| 447750.00|

178| 0.00|

179|1199100.00|

180| 0.00|

181| 0.00|

182| 0.00|

183| 0.00|

184| 0.00|

185| 0.00|

186| 0.00|

187| 0.00|

188| 0.00|

189| 0.00|

190| 0.00|

191| 0.00|

192| 0.00|

193| 0.00|

194| 0.00|

195| 0.00|

196| 0.00|

197| 0.00|

198| 0.00|

199| 0.00|

200| 0.00|

201| 0.00|

202| 0.00|

203| 0.00|

204| 0.00|

205| 0.00|

206| 0.00|

105| 0.00|

106 row(s) fetched.

Задача 4. Выполните слияние таблицы Orders1 только с теми строками таблицы Orders2, в которых заказы находятся в состоянии 'Shipped'.

> select \* from orders\_1\_copy oc

order\_id|customer\_id|status |salesman\_id|order\_date|

--------+-----------+-------+-----------+----------+

8| 28|Pending| 156|2019-09-09|

5| 5|Pending| 156|2019-09-09|

2| 4|Pending| 145|2019-09-26|

3| 5|Pending| 146|2019-09-26|

4 row(s) fetched.

> select \* from orders\_2\_copy oc

order\_id|customer\_id|status |salesman\_id|order\_date|

--------+-----------+-------+-----------+----------+

2| 4|Shipped| 145|2019-09-26|

3| 5|Shipped| 146|2019-09-26|

6| 6|Pending| 145|2019-10-15|

7| 7|Pending| 147|2019-10-15|

9| 8|Shipped| 147|2019-10-05|

5 row(s) fetched.

> merge into orders\_1\_copy ord1

using (

select \*

from orders\_2\_copy

where status = 'Shipped'

) ord2

on (ord1.order\_id = ord2.order\_id)

when matched then

update set status = ord2.status

when not matched then

insert (order\_id, customer\_id, status, salesman\_id, order\_date)

values (ord2.order\_id, ord2.customer\_id, ord2.status, ord2.salesman\_id, ord2.order\_date)

3 row(s) modified.

> select \* from orders\_1\_copy oc

order\_id|customer\_id|status |salesman\_id|order\_date|

--------+-----------+-------+-----------+----------+

8| 28|Pending| 156|2019-09-09|

5| 5|Pending| 156|2019-09-09|

2| 4|Shipped| 145|2019-09-26|

3| 5|Shipped| 146|2019-09-26|

9| 8|Shipped| 147|2019-10-05|

5 row(s) fetched.

Задача 5. Создайте таблицу Order\_Items\_New, которая содержит данные о новых продажах, и заполните ее данными. Выполните слияние таблицы Order\_Items\_Copy с таблицей Order\_Items\_New. Алгоритм слияния: если в таблице Order\_Items\_Copy существует строка, у которой значения столбцов order\_id, product\_id совпадают со значениями этих столбцов в добавляемой строке из таблицы Order\_Items\_New, то обновить значение столбца quantity, в противном случае вставить новую строку.

> select \* from order\_items\_new oin

order\_id|item\_id|product\_id|quantity|unit\_price|

--------+-------+----------+--------+----------+

8| 1| 34| 144| 150.00|

6| 7| 7| 119| 2380.00|

6| 1| 19| 67| 1850.00|

11| 1| 66| 113| 560.00|

12| 1| 11| 41| 2120.00|

999| 999| 999| 100000| 100000.00|

1000| 1000| 1000| 100000| 100000.00|

7 row(s) fetched.

> MERGE INTO Order\_Items\_Copy AS copy

USING Order\_Items\_New AS new

ON (copy.order\_id = new.order\_id AND copy.product\_id = new.product\_id)

WHEN MATCHED THEN

UPDATE SET

quantity = new.quantity + copy.quantity

WHEN NOT MATCHED THEN

INSERT (order\_id, item\_id, product\_id, quantity, unit\_price)

VALUES (new.order\_id, new.item\_id, new.product\_id, new.quantity, new.unit\_price)

returning copy.\*

order\_id|item\_id|product\_id|quantity|unit\_price|rating\_p|

--------+-------+----------+--------+----------+--------+

6| 7| 7| 238| 2380.00| 4|

6| 1| 19| 134| 1850.00| 4|

8| 1| 34| 288| 150.00| 5|

11| 1| 66| 226| 560.00| 3|

12| 1| 11| 82| 2120.00| 4|

999| 999| 999| 100000| 100000.00| |

1000| 1000| 1000| 100000| 100000.00| |

Задача 6. Удалить данные об отмененных заказах (status = ‘Canceled’), с даты оформления которых прошло более 5 лет.

> delete from orders\_copy

where

status = 'Canceled'

and

extract(year from age(now(), order\_date)) > 5

returning \*

order\_id|customer\_id|status |salesman\_id|order\_date|

--------+-----------+--------+-----------+----------+

5| 5|Canceled| 156|2018-03-15|

22| 22|Canceled| 152|2018-07-26|

27| 43|Canceled| 152|2018-08-16|

31| 46|Canceled| 153|2018-08-12|

42| 56|Canceled| 164|2018-06-03|

47| 59|Canceled| 162|2018-10-10|

48| 60|Canceled| 164|2018-10-10|

56| 67|Canceled| 175|2018-09-29|

69| 44|Canceled| 154|2017-03-17|

70| 45|Canceled| 161|2017-02-21|

97| 47|Canceled| 170|2017-07-12|

11 row(s) fetched.

Задача 7. Удалить из таблицы Order\_Items\_Copy данные о продаже товаров, которые нарушают правило: рейтинг продавца должен больше или равен рейтингу товара.

**create** **table** Order\_Items\_Copy2 **as**

**select**

\*

**from**

order\_items oi;

**ALTER** **TABLE** Order\_Items\_Copy2

**ADD** **Column** rating\_p **integer**;

**UPDATE** Order\_Items\_Copy2 **as** *oic*

**set** rating\_p = (**SELECT** *p*.rating\_p **FROM** Products *p*

**WHERE** *p*.product\_id=*oic*.product\_id)

**returning** \*;

**with** emp\_rating **as** (

**select**

o.order\_id,

e.rating\_e

**from** orders o

**join** employees e **on** o.salesman\_id = e.employee\_id

)

**delete** **from** order\_items\_copy2 *oic*

**using** emp\_rating er

**where** oic.order\_id = er.order\_id

**and** oic.rating\_p > er.rating\_e

**returning** oic.\*;

> with emp\_rating as (

select

o.order\_id,

e.rating\_e

from orders o

join employees e on o.salesman\_id = e.employee\_id

)

delete from order\_items\_copy2 oic

using emp\_rating er

where oic.order\_id = er.order\_id

and oic.rating\_p > er.rating\_e

returning oic.\*

order\_id|item\_id|product\_id|quantity|unit\_price|rating\_p|

--------+-------+----------+--------+----------+--------+

32| 1| 14| 86| 700.00| 4|

60| 1| 15| 36| 280.00| 4|

87| 1| 1| 57| 640.00| 4|

107| 2| 66| 126| 400.00| 3|

107| 1| 63| 31| 1900.00| 3|

87| 9| 21| 142| 720.00| 4|

70| 7| 32| 32| 2800.00| 5|

73| 5| 12| 124| 2060.00| 4|

84| 6| 31| 34| 3000.00| 5|

6| 7| 7| 119| 2380.00| 4|

6| 1| 19| 67| 1850.00| 4|

12| 1| 11| 41| 2120.00| 4|

14| 1| 29| 98| 3190.00| 5|

15| 1| 28| 60| 3250.00| 5|

23| 1| 22| 104| 1760.00| 4|

24| 1| 12| 129| 2060.00| 4|

25| 1| 43| 19| 1000.00| 5|

26| 1| 20| 114| 1840.00| 4|

28| 1| 31| 61| 3000.00| 5|

37| 1| 35| 150| 2020.00| 5|

39| 1| 18| 116| 1900.00| 4|

43| 1| 67| 74| 550.00| 3|

51| 1| 21| 34| 1810.00| 4|

53| 1| 10| 130| 2200.00| 4|

54| 1| 28| 73| 3250.00| 5|

57| 1| 24| 75| 1710.00| 4|

58| 1| 64| 55| 570.00| 3|

62| 1| 18| 130| 1900.00| 4|

70| 1| 18| 146| 1900.00| 4|

71| 1| 19| 146| 1850.00| 4|

81| 1| 28| 133| 3250.00| 5|

84| 1| 16| 137| 1910.00| 4|

96| 1| 26| 34| 5500.00| 5|

97| 1| 11| 39| 2120.00| 4|

102| 1| 19| 69| 1850.00| 4|

58| 9| 36| 134| 2000.00| 5|

81| 6| 65| 61| 570.00| 3|

22| 11| 23| 144| 1750.00| 4|

23| 9| 14| 86| 2010.00| 4|

24| 6| 27| 99| 4140.00| 5|

27| 11| 16| 67| 1910.00| 4|

43| 7| 1| 57| 3410.00| 4|

58| 10| 63| 31| 650.00| 3|

43 row(s) fetched.