PRACTICE PROBLEMS

Consider the following schema:

SALE (CUST_NAME, ITEM_NAME, WHEN)

CUST (CUST_NAME, INDUSTRY)

ITEM (ITEM_NAME, CATEGORY, PRICE)

Write both relational algebra and relational calculus expressions which will answer the following questions:

(1) Which customers bought something on 12-18-04?

(2) Which customers from the airline industry bought something on 12-18-04?

(3) Which customers have bought something from a category other than "transportation"?

(4) Which customers bought something costing more than $1,000,000 that was not in the category "doohicky"?

(5) Which customers from the airline industry did not buy something on 12-18-04?

(6) Which customers from the airline industry bought an item from the category "spare part" on 12-18-04?

(7) What categories has the customer "Widgets 'R'Us" never purchased an item in? (8) Which customers are associated with more than one industry?

(9) What category has the most expensive item?

(10) Which customers have bought every item in the "spare part" category?

(11) Which customers have bought exactly the same set of items as "Chris' Lobster Shack"?

ANSWERS FOR PRACTICE PROBLEMS

1

RA:
PROJECT (CUST_NAME) (SELECT (WHEN = 12-18-04) (SALE))

RC:
{s.CUST_NAME | SALE(s) and s.WHEN = 12-18-04}

2

RA:
PROJECT (CUST_NAME) (SELECT (WHEN = 12-18-04 and INDUSTRY = 'airline') (SALE * CUST))

RC:
{s.CUST_NAME | SALE(s) and s.WHEN = 12-18-04 and EXISTS (c)(CUST(c) and c.CUST_NAME = s.CUST_NAME and c.INDUSTRY = 'airline')}

3

RA:
PROJECT (CUST_NAME) (SELECT (CATEGORY != 'transportation') (SALE * ITEM))

RC:
{s.CUST_NAME | SALE(s) and EXISTS (i)(ITEM(i) and i.ITEM_NAME = s.ITEM_NAME and i.CATEGORY != 'transportation'}

4

RA:
PROJECT (CUST_NAME) (SELECT (CATEGORY != 'doohicky' and PRICE > 1000000) (SALE * ITEM))

RC:
{s.CUST_NAME | SALE(s) and EXISTS (i)(ITEM(i) and i.ITEM_NAME = s.ITEM_NAME and i.CATEGORY != 'dohicky' and i.PRICE > 1000000}

5

RA:
THOSE_WHO_DID <- PROJECT (CUST_NAME) (SELECT (WHEN = 12-18-04 and INDUSTRY = 'airline') (SALE * CUST))

ANSWER <- PROJECT (CUST_NAME) (SELECT (INDUSTRY = 'AIRLINE') (CUST)) -THOSE_WHO_DID

RC:
{c.CUST_NAME | CUST(c) and c.INDUSTRY = 'airline'
and NOT EXISTS(s)(SALE(s) and s.CUST_NAME = c.CUST_NAME and s.WHEN = 12-18-04)}

6

RA:
PROJECT (CUST_NAME) (SELECT (CATEGORY = 'spare part' and INDUSTRY = 'airline' and WHEN = 12-18-04) (SALE * CUST * ITEM)

RC:
{c.CUST_NAME | CUST(c) and c.INDUSTRY = 'airline' and EXISTS (s)(SALE(S) and s.CUST_NAME = c.CUST_NAME and s.WHEN = 12-18-04 and EXISTS (i)(ITEM(i) and i.ITEM_NAME = s.ITEM_NAME and i.CATEGORY = 'spare part'))}

7

RA:
WRU_CATS <- PROJECT (CATEGORY) (SELECT (CUST_NAME = 'Widgets R Us') (SALE * ITEM))

ANSWER <- PROJECT (CATEGORY) (ITEM) - WRU_CATS

RC:
{i.CATEGORY | ITEM(i) AND FORALL(s)(SALE (s) and s.CUST_NAME = 'Widgets R Us' => NOT EXISTS (ii)(ITEM(ii) and ii.CATEGORY = i.CATEGORY and s.ITEM_NAME = ii.ITEM_NAME))}

Note that this is the same as:
{i.CATEGORY | ITEM(i) AND FORALL (s)(SALE (s) and s.CUST_NAME = 'Widgets R Us' => "the item purchased in the sale s was not in the same category as i")}

which is the same as:

{i.CATEGORY | ITEM(i) AND FORALL (s)("if s was a sale to the customer 'Widegets R Us' then the item purchased in the sale s was not in the same category as i")}

8

RA:
PAIRS (C1, I1, C2, I2) <- CUST X CUST
ANSWER <- PROJECT (C1) (SELECT (C1 = C2 and I1 != I2) (PAIRS))

RC:
{c.CUST_NAME | CUST (c) and EXISTS (cc) (CUST(cc) and c.CUST_NAME = cc.CUST_NAME and c.INDUSTRY != cc.INDUSTRY)}

9

RA:
PAIRS (I1, C1, P1, I2, C2, P2) <- ITEM X ITEM

This next step gives us all items with someone who costs more:
ONES_WITH_LARGER (ITEM_NAME, CATEGORY) <- PROJECT (I1, C1) (SELECT (P1 < P2) (PAIRS))

And all items minus all items with someone who costs more gives our answer:

ANSWER <- PROJECT (CATEGORY) (PROJECT (ITEM_NAME, CATEOGRY) (ITEM) - ONES_WITH_LARGER)

RC:

{i.CATEGORY | ITEM(i) and NOT EXISTS (ii)(ITEM (ii) and i.PRICE < ii.PRICE)}

10

RA:
First, we get all person/part combos for every type of spare part:
ALL_PAIRS <- PROJECT (CUST_NAME) (CUST) X PROJECT (ITEM_NAME) (SELECT (CATEGORY = 'Spare Part') (ITEM))

Now, we use that to get everyone who has NOT bought every spare part
NOT_ALL <- PROJECT (CUST_NAME) (ALL_PAIRS - PROJECT (CUST_NAME, ITEM_NAME) (SALE))

And the answer is everyone else!
ANSWER <- PROJECT (CUST_NAME) (CUST) - NOT_ALL

RC:
It is easier in RC:
{c.CUST_NAME | CUST(c) and forall (i)(ITEM(i) and i.CATEGORY = 'spare part' => EXISTS (s)(SALE(s) and s.CUST_NAME = c.CUST_NAME and i.ITEM_NAME = s.ITEM_NAME))}

This is the same as:
{c.CUST_NAME | CUST(c) and forall (i)(ITEM(i) and i.CATEGORY = 'spare part' => "the customer c.CUST_NAME has bought i.ITEM_NAME")}

Or
{c.CUST_NAME | CUST(c) and forall (i)("if i is a spare part from the ITEM table, then the customer c.CUST_NAME has bought i.ITEM_NAME")}

11

RA:
This is not too different from #10, but it's even harder.

First, we need to get all person/item combos, where the item is one from Chris' Lobster Shack:
ALL_COMBOS <- PROJECT (CUST_NAME) (CUST) X PROJECT (ITEM_NAME) (SELECT (CUST_NAME = 'Chris' Lobster Shack') (SALE))

We use that to find the people who are missing one of the items from Chris' Lobster Shack:
NOT_ALL <- PROJECT (CUST_NAME) (ALL_PAIRS - PROJECT (CUST_NAME, ITEM_NAME) (SALE))

Now we can get everyone who has bought all of those items:
GOT_EM_ALL <- PROJECT (CUST_NAME) (CUST) - NOT_ALL

But, we're not done. We may have people who bought some extra items.
So we need everyone who bought something that Chris did not.
CHRIS_MISSED <- PROJECT (ITEM_NAME) (ITEM) - PROJECT (ITEM_NAME) (SELECT (CUST_NAME = 'Chris' Lobster Shack') (SALE)))

EXTRA <- PROJECT (CUST_NAME) (SALE * CHRIS_MISSED)

And subtract out those people to get the answer:
ANSWER <- GOT_EM_ALL - EXTRA

RC:
We want something like:
{c.CUST_NAME | CUST (c) and "Chris has bought everything that c.CUST_NAME has and c.CUST_NAME has bought everything that Chris has"}

The RC for this is:
{c.CUST_NAME | CUST (c) and
FORALL (s)(SALE(s) and s.CUST_NAME = c.CUST_NAME =>
EXISTS (ss)(SALE(ss) and ss.ITEM_NAME = s.ITEM_NAME and
ss.CUST_NAME = 'Chris' Lobster Shack'))

and

FORALL (s)(SALE(s) and s.CUST_NAME = 'Chris' Lobster Shack' =>
EXISTS (ss)(SALE(ss) and ss.ITEM_NAME = s.ITEM_NAME and
ss.CUST_NAME = s.CUST_NAME))}