The CAP (Customers-Agents-Products) database

CUSTOMERS	A table containing information about customers
cid	Unique identifier for a customer/row—note there is no customer corresponding to cid = 'c005'
cname	Name of a customer
city	City where the customer (headquarters) is located
discnt	Each customer has a negotiated discount on prices
AGENTS	A table containing information about agent employees
aid	Unique identifier for an agent/row
aname	Last name of agent
city	City where agent is based
percent	Percentage commission each agent receives on each sale
PRODUCTS	A table containing information about products for sale
pid	Unique identifier for a product/row
pname	Descriptive name of product
city	City where this product is warehoused
quantity	Quantity on hand for sale, in standard units
price	Wholesale price of each unit product

Note that the same column name, city, appears in all three tables defined so far. This is not a coincidence.

ORDERS	A table containing information about orders
ordno	Unique identifier for this order
month	Month the order was placed; assume that orders started in January of this year
cid	This customer . . .
aid	. . . purchased through this agent . . .
pid	. . . this specific product . . .
qty	. . . in this total quantity. . .
dollars	. . . at this dollar cost

Queries from O'Neil and O'Neil's book:

(a)• Find all (cid, aid, pid) triples for customer, agent, product combinations that are all in the same city. Nothing about orders is involved in this selection.

(b) Find all (cid, aid, pid) triples for customer, agent, product combinations that are *not* all in the same city (any two may be).

(c)• Find all (cid, aid, pid) triples for customer, agent, product combinations, *no two of which* are in the same city.

(d) Get cities of agents booking an order from customer c002.

(e)• Get product names ordered by at least one customer based in Dallas through an agent based in Tokyo.

(f) Get pids of products ordered through any agent who makes at least one order for a customer in Kyoto. NOTE: The request posed here is not the same as asking for pids of products ordered by a customer in Kyoto.

(g)• Display all pairs of aids for agents who live in the same city.

(h) Find cids of customers who did not place an order through agent a03.

(i)• Find cids of customers who have the largest discount; separately, find those who have the smallest discount. NOTE: This is quite hard with the operations provided in relational algebra.

(j) Find cids of customers who order all products.

(k)• Find pids of products ordered through agent a03 but not through agent a06.

(l) Get pnames and pids of products that are stored in the same city as one of the agents who sold these products.

(m)• Get aids and anames of agents with aname beginning with the letter "N" who do not place orders for any product in Newark.

(n) Get cids of customers who order both product p01 and product p07.

(o)• Get names of agents who place orders for all products ordered by customer c002.

(p) Get names of agents who place orders for all products that are ordered by any customer at all. (Hint: The phrase "any customer at all" means the same as "some customer.")

(q)• Get (cid, aid, pid) triples for customer, agent, product combinations so that *at most two* of them are in the same city. (Is this equivalent to any of the first three queries of this exercise, (a), (b), or (c)?)

(r) Get pids of products ordered by all customers who place any order through agent a03.

(s)• Get aids of agents who place individual orders in dollar value greater than $500 for customers living in Kyoto.

(t) Give all (cname, aname) pairs where the customer places an order through the agent.

(u)• [HARD] Get cids of customers who order all their products through only one agent.

Consider the CAP described as:

`CUSTOMERS (cid, cname, city, discnt)  C`
`AGENTS (aid, aname, city, percent)  A`
`PRODUCTS (pid, pname, city, quantity, price)  P`
`ORDERS (ordno, month, cid, aid, pid, qty, dollars)  O`

(1) RA expressions for the queries c, e, I, k, m, o, p, q, r and u.

(c) $\Pi_{\text{cid,aid,pid}}(\sigma_{3\neq 7 \wedge 7\neq 11 \wedge 3\neq 11}(\text{CUSTOMERS} \times \text{AGENTS} \times \text{PRODUCTS}))$

(e) $\Pi_{\text{pname}}(\text{PRODUCTS} \infty \Pi_{\text{pid}}(\sigma_{\text{city=”Dallas”}}(\text{CUSTOMERS}) \infty \text{ ORDERS} \infty$
$\sigma_{\text{city=”Tokyo”}}(\text{AGENTS}))$

(i) Max: $\Pi_1(\text{CUSTOMERS}) - \Pi_1(\sigma_{4<8}(\text{CUSTOMERS} \times \text{CUSTOMERS}))$

Min: $\Pi_1(\text{CUSTOMERS}) - \Pi_1(\sigma_{4>8}(\text{CUSTOMERS} \times \text{CUSTOMERS}))$

(k) $\Pi_{\text{pid}}(\sigma_{\text{aid=”a03”}}(\text{ORDERS})) - \Pi_{\text{pid}}(\sigma_{\text{aid=”a06”}}(\text{ORDERS}))$

(m) $\Pi_{\text{aid,aname}}(\sigma_{\text{aname=”N*”}}(\text{AGENTS})) - \Pi_{\text{aid,aname}}(\text{AGENTS} \infty \text{ ORDERS} \infty$
$\sigma_{\text{city=”Nework”}}(\text{PRODUCTS}))$

(o) $\Pi_{\text{aname}}(\text{AGENTS} \infty (\Pi_{\text{aid,pid}}(\text{ORDERS}) \div \Pi_{\text{pid}}(\sigma_{\text{cid=”c002”}}(\text{ORDERS}))))$

(p) $\Pi_{\text{aname}}(\text{AGENTS} \infty (\Pi_{\text{aid,pid}}(\text{ORDERS}) \div \Pi_{\text{pid}}(\text{ORDERS})))$

(q) $\Pi_{\text{cid,aid,pid}}(\text{CUSTOMERS} \times \text{AGENTS} \times \text{PRODUCTS}) -$

$\Pi_{\text{cid,aid,pid}}(\text{CUSTOMERS} \infty \text{ AGENTS} \infty \text{ PRODUCTS})$

Equivalent to query (b)

(r) $\Pi_{\text{pid,cid}}(\text{ORDERS}) \div \Pi_{\text{cid}}(\sigma_{\text{aid=”a03”}}(\text{ORDERS}))$

(u) $\Pi_3(\text{ORDERS}) - \Pi_3(\sigma_{3=10 \wedge 4\neq 11}(\text{ORDERS} \times \text{ORDERS}))$

(2) TRC for a, c, e, k, o, p, q, r and u

(a) $\{ t^{(3)} \mid (\exists c)(\exists a)(\exists p)(\text{CUSTOMERS}(c) \wedge \text{AGENTS}(a) \wedge \text{PRODUCTS}(p) \wedge t[1]=c[1] \wedge t[2]=a[1] \wedge t[3]=p[1] \wedge c[3]=a[3] \wedge c[3]=p[3])\}$

(k) $\{ t^{(1)} \mid (\exists o1)(\text{ORDERS}(o1) \wedge t[1]=o1[5] \wedge o1[4]=\text{”a03”} \wedge \text{NOT}(\exists o2)(\text{ORDERS}(o2) \wedge o1[5]=o2[5] \wedge o2[4]=\text{”a06”}))\}$

(o) $\{ t^{(1)} \mid (\exists a)(\text{AGENTS}(a) \wedge t[1]=a[2] \wedge (\forall o1)(\text{ORDERS}(o1) \wedge o1[3]=\text{”c002”} \longrightarrow (\exists o2)(\text{ORDERS}(o2) \wedge o1[5]=o2[5] \wedge o2[4]=a[1]))))\}$

(p) $\{ t^{(1)} \mid (\exists a)(\text{AGENTS}(a) \wedge t[1]=a[2] \wedge (\forall o1)(\text{ORDERS}(o1) \longrightarrow (\exists o2)(\text{ORDERS}(o2) \wedge o1[5]=o2[5] \wedge o2[4]=a[1]))))\}$

(q) $\{ t^{(3)} \mid (\exists c)(\exists a)(\exists p)(\text{CUSTOMERS}(c) \wedge \text{AGENTS}(a) \wedge \text{PRODUCTS}(p) \wedge t[1]=c[1] \wedge t[2]=a[1] \wedge t[3]=p[1] \wedge (c[3]\neq a[3] \vee c[3]\neq p[3] \vee a[3]\neq p[3]))\}$

(r) $\{ t^{(1)} \mid (\exists p)(\text{PRODUCTS}(p) \wedge t[1]=p[1] \wedge (\forall o1)(\text{ORDERS}(o1) \wedge o1[4]=\text{”a03”} \longrightarrow (\exists o2)(\text{ORDERS}(o2) \wedge o1[3]=o2[3] \wedge o2[5]=p[1]))))\}$

(u) $\{ t^{(1)} \mid (\exists c)(\text{CUSTOMERS}(c) \wedge t[1]=c[1] \wedge (\forall o1)\ (\forall o2)\ (\text{ORDERS}(o1) \wedge \text{ORDERS}(o2) \wedge o1[3]=o2[3]=c[1]) \longrightarrow o1[4]=o2[4]))\}$

SQL Queries:

(a)

```
SELECT C.cid, A.aid, P.pid
FROM CUSTOMERS C, AGENTS A, PRODUCTS P
WHERE C.city=A.city AND C.city=P.city
```

(b)

```
SELECT C.cid, A.aid, P.pid
FROM CUSTOMERS C, AGENTS A, PRODUCTS P
WHERE C.city≠A.city OR C.city≠P.city OR A.city≠P.city
```

(c)

```
SELECT C.cid, A.aid, P.pid
FROM CUSTOMERS C, AGENTS A, PRODUCTS P
WHERE C.city≠A.city AND C.city≠P.city AND A.city≠P.city
```

(d)

```
SELECT DISTINCT A.city
FROM ORDERS O, AGENTS A
WHERE O.cid=’c002’ AND A.aid=O.aid
```

(f)

```
SELECT O.pid
FROM ORDERS O
WHERE O.aid IN
     {
     SELECT O1.aid
     FROM CUSTOMERS C, ORDERS O1
     WHERE C.cid=O1.cid AND C.city=’Kyoto’
     }
```

(g)

```
SELECT DISTINCT A1.aid, A2.aid
FROM AGENTS A1, AGENTS A2
WHERE A1.aid≠A2.aid AND A1.city=A2.city
```

(h)

```
SELECT C.cid
FROM CUSTOMERS C
WHERE NOT EXISTS
     {
     SELECT *
     FROM ORDERS O
     WHERE O.cid=C.cid AND O.aid=’a03’
     }
```

(i)

```
MAX:
SELECT C.cid
FROM CUSTOMERS C
WHERE C.discnt=
     {
     SELECT MAX(C1.discnt)
     FROM CUSTOMERS C1
     }
MIN:
SELECT C.cid
FROM CUSTOMERS C
WHERE C.discnt=
     {
     SELECT MIN(C1.discnt)
     FROM CUSTOMERS C1
     }
```

(j) We give two solutions for this query.

Solution1:

```
SELECT C.cid
     FROM CUSTOMERS C
     WHERE NOT EXISTS
          (SELECT *
           FROM PRODUCT P
           WHERE NOT EXISTS
                  (SELECT *
                   FROM ORDERS O
                   WHERE O.cid = C.cid AND O.pid = P.pid))
```

Solution2:

```
SELECT C.cid
FROM CUSTOMERS C
WHERE NOT EXISTS
{
SELECT P.pid
FROM PRODUCTS P

EXCEPT

SELECT DISTINCT O.pid
FROM ORDERS O
WHERE O.cid=C.cid
}
```

(k)

```
SELECT O.pid
FROM ORDERS O
WHERE O.aid='a03' AND
      NOT EXISTS
        (SELECT O1.cid
         FROM ORDERS O1
         WHERE O1.pid=O.pid AND O1.aid='a06'
```

(p)

```
SELECT A.aname
FROM AGENTS A
WHERE NOT EXIST
     {
     SELECT DISTINCT O.pid
     FROM ORDERS O

     EXCEPT

     SELECT DISTINCT O1.pid
     FROM ORDERS O1
     WHERE O1.aid=A.aid
     }
```

(q)

```
SELECT C.cid, A.aid, P.pid
FROM CUSTOMERS C, AGENTS A, PRODUCTS P
WHERE C.city≠A.city OR C.city≠P.city OR A.city≠P.city
```

(r)

```
SELECT P.pid
FROM PRODUCTS P
WHERE NOT EXISTS
     {
     SELECT DISTINCT O.cid
     FROM ORDERS O
     WHERE O.pid=P.pid

     EXCEPT

     SELECT DISTINCT C.cid
     FROM CUSTOMERS C, ORDERS O
     WHERE C.cid=O.cid AND O.aid='a03'
     }
```

(u)

```
SELECT C.cid
FROM CUSTOMERS C
WHERE NOT EXISTS
{
SELECT O1
FROM ORDERS O1, ORDERS O2
WHERE O1.cid=C.cid AND O1.cid=O2.cid AND O1.aid≠O2.aid}
```