

```
-- topbuyer

-- For each club, who is the customer who has spent the most via that
-- club's offers on books, and what is the total that he or she has spent
-- via that club?
-- Order by total sales, from highest to lowest, then by customer name.

with
  spending (club, cid, name, amount) as (
    select P.club, P.cid, C.name, sum(P.qty*O.price)
      from yrb_customer C, yrb_purchase P, yrb_offer O
     where C.cid = P.cid and
           P.title = O.title and P.year = O.year and P.club = O.club
     group by P.club, P.cid, C.name
  ),
  top (club, best) as (
    select club, max(amount)
      from spending
     group by club
  )
select S.club, S.name, T.best
  from spending S, top T
 where S.club = T.club and S.amount = T.best
 order by amount desc, name;
```

-- allbooks

-- List customers by name along with category and language such
-- that the customer has bought all the books offered in that
-- category / language group and there is more than one book in that
-- category / language group. Do not have any duplicates.
-- Order by name + category + language.

```
select distinct name, cat as category, language
  from yrb_customer C,
      ((select cid, cat, language
         from yrb_customer,
             (select cat, language
              from yrb_book
              group by cat, language
              having count(*) > 1) as Y)
     except
      (select cid, cat, language
       from
         ((select cid, cat, language, title, year
            from yrb_customer, yrb_book)
        except
         (select D.cid, cat, language, P.title, P.year
          from yrb_customer D, yrb_purchase P, yrb_book B
          where D.cid = P.cid and
                P.title = B.title and P.year = B.year)) as Z))
     as A
 where C.cid = A.cid
 order by name, cat, language;
```

```
-- catlang
```

```
-- List total sales (by sum of price paid) for each category / language pair.  
-- Order by total sales, from highest to lowest.
```

```
select cat as category, language, sum(qnty*price) as total  
  from yrb_purchase P, yrb_offer O, yrb_book B  
  where P.title = O.title and P.year = O.year and  
        P.club = O.club and  
        P.title = B.title and P.year = B.year  
  group by cat, language  
  order by total desc;
```

```
-- like
```

```
-- List all books that have 'like' or 'Like' in the title.  
-- Show the tile, year, and the book's category.  
-- Order by title + year + cat.
```

```
select title, year, cat as category  
  from yrb_book B  
 where B.title like '%like%' or B.title like '%Like%'  
 order by title, year, cat;
```

```
-- meme
```

```
-- List each customer who has bought the same book but on different
-- occasions. List by customer's name, and title and year of the book,
-- and on how many different occasions he or she purchased the book. Do
-- not count cases where a customer bought several copies of a book on one
-- occasion but never again.
-- Order by name + title + year.
```

```
select name, title, year, number
  from yrb_customer C,
       (select distinct cid, title, year, count(when) as number
         from yrb_purchase P
        group by cid, title, year
       having count(when) > 1) as B
 where C.cid = B.cid
 order by name, title, year;
```

```
-- multiple
```

```
-- List each customer who has bought several copies of a book within a  
-- purchase. Show the customer's name, the book's title and year, and how  
-- many copies were purchased.
```

```
-- Order by name + title + year.
```

```
select name, title, year, qty  
  from yrb_purchase P, yrb_customer C  
  where C.cid = P.cid and qty > 1  
  order by name, title, year;
```

```
-- nolang

-- List city / language pairs such that no one in that city
-- has purchased any books in that language.
-- Do not have duplicates in the answer table.
-- Order by city + language.

(select distinct city, language
  from yrb_book B, yrb_customer C)
except
(select city, language
  from yrb_purchase P, yrb_customer C, yrb_book B
  where P.title = B.title and P.year = B.year and
        C.cid = P.cid)
order by city, language;
```

```
-- pairs

-- Find pairs of customers such that the two customers have bought at
-- least three books in common. Print three columns: two with the
-- customers' names and one with the number of books in common. Do not
-- return any duplicates. Furthermore, say 'Mark Dogfurry' and 'Zebulon
-- Zilio' have four books in common, only output ('Mark Dogfurry',
-- 'Zebulon Zilio', 4) and not ('Zebulon Zilio', 'Mark Dogfurry', 4)!

-- If 'Mark Dogfurry' and 'Zebulon Zilio' have each bought the same
-- book three times, this does not count. It has to be at least three
-- different books.

-- Order by the names.

with own (cid, title, year) as
  (select cid, title, year
   from yrb_purchase P
   group by cid, title, year)
select distinct A.name as first, B.name as second, count(*) as number
  from yrb_customer A, yrb_customer B, own P, own Q
  where A.name <= B.name and A.cid <> B.cid and
        A.cid = P.cid and B.cid = Q.cid and
        P.title = Q.title and P.year = Q.year
  group by A.cid, A.name, B.cid, B.name
  having count(*) >= 3
  order by A.name, B.name;
```



```
-- polyuniv
-- List by name and city customers who belong to more than one
-- university club (CNU, UVA, VaTech, and W&M). Do not
-- allow duplicate rows in the answer table.
-- Order by name + city.

select distinct name, city
  from yrb_customer C, yrb_member A, yrb_member B, yrb_club AC, yrb_club BC
 where C.cid = A.cid and C.cid = B.cid and
       A.club = AC.club and B.club = BC.club and
       AC.desc like 'University %' and
       BC.desc like 'University %' and
       A.club < B.club
 order by name, city;
```