

1 YRB 2005 Queries

-- ancient

-- List the customers who made a purchase before January 1 2002
-- ('2002-1-1'). Show customer's name and city and the date (not
-- timestamp!) of purchase. Eliminate duplicates.
-- Order by name + city + date

```
select distinct name, city, cast(when as date) as date
  from yrb_customer C, yrb_purchase P
  where C.cid = P.cid and
         cast(when as date) < cast('2002-1-1' as date);
```

2 YRB 2005 Queries

```
-- repeat

-- List each customer who has bought more than one copy of the same book
-- over time. 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, sum(qnty) as total
  from yrb_purchase P, yrb_customer C
  where C.cid = P.cid
  group by P.cid, name, title, year
  having sum(qnty) > 1
  order by name, title, year;
```

3 YRB 2005 Queries

```
-- clubreport

-- For each club, what are the total sales, the number of distinct book
-- titles (title + year) bought via that club, and the number of customers
-- who belong to that club?
-- Order by total sales, from highest to lowest,
-- and then by club name in cases of ties.
```

with

```
Sales (club, sales) as (
  select P.club, sum(P.qty*O.price)
    from yrb_purchase P, yrb_offer O
   where P.title = O.title
        and P.year = O.year
        and P.club = O.club
   group by P.club
 union
  select club, 0
    from yrb_club
   where club not in (select club from yrb_purchase)
),
```

```
ClubBooks (club, title, year) as (
  select distinct club, title, year
    from yrb_purchase
),
```

```
BookCount (club, titles) as (
  select club, count(*)
    from ClubBooks
   group by club
 union
  select club, 0
    from yrb_club
   where club not in (select club from yrb_purchase)
),
```

```
ClubMembers (club, membership) as (
  select club, count(*)
    from yrb_member
   group by club
 union
  select club, 0
    from yrb_club
   where club not in (select club from yrb_member)
)
```

```
select S.club, S.sales, B.titles, C.membership
  from Sales S, BookCount B, ClubMembers C
 where S.club = B.club and S.club = C.club
 order by S.sales desc, S.club;
```

4 YRB 2005 Queries

```
-- 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;
```

5 YRB 2005 Queries

-- uniclub

-- To how many university clubs (for example, CNU, UVA, VaTech, and W&M)
-- does each customer belong?
-- Order by name + city.

with

```
Uniclub (cid, name, city, #uniclubs) as (  
  select A.cid, A.name, A.city, count(*)  
    from yrb_customer A, yrb_member M, yrb_club C  
   where A.cid = M.cid and  
         M.club = C.club and  
         C.desc like 'University club%'  
   group by A.cid, A.name, A.city  
  )
```

```
select name, city, #uniclubs  
  from Uniclub  
 union  
select name, city, 0 as #uniclubs  
  from yrb_customer  
 where cid not in (select cid from Uniclub)  
 order by name, city;
```

```

-- 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;

```

7 YRB 2005 Queries

```
-- orders

-- All the books a customer orders at the same time (when) are considered
-- to be part of the same "order". Those books are shipped together to
-- the customer and the customer is billed for the entire order.

-- Calculate the bill for each "order" Show the customer's name and city,
-- the date and time of the order (not the timestamp!), and the bill.
-- Order by name + city + when.

select name, city, cast(when as date) as day, cast(when as time) as time,
       cast(sum(price * qty) as decimal(5,2)) as bill
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.cid, name, city, when
order by name, city, when;
```

8 YRB 2005 Queries

```
-- weights

-- Calculate the total weight of every customer's order.
-- Order by weight, descending.

select name, city, cast(when as date) as day, cast(when as time) as time,
       sum(weight * qty) as grams
from yrb_customer C, yrb_purchase P, yrb_book B
where C.cid = P.cid and
      P.title = B.title and P.year = B.year
group by name, city, when
order by sum(weight * qty) desc, name, city, when;
```



```
-- billing

-- All the books a customer orders at the same time (when) are considered
-- to be part of the same "order". Those books are shipped together to
-- the customer and the customer is billed for the entire order.

-- Calculate the bill for each "order" with the shipping cost added.
-- The shipping cost is as follows: The weight of the order is looked up
-- in the 'yrb_shipping' table. If the weight is X grams, the entry just
-- higher than X is found in the shipping table and the associated
-- shipping price is added. For instance, if the order's weight is 1447
-- grams, the entry '1500 5.00' is found, and so the cost is $5.00.

-- Show the customer's name and city, the date and time of the order (not
-- the timestamp!), the bill without the shipping charge, and the total
-- bill (with shipping).
-- Order by name + city + when.
```

```
with
  orders (cid, name, city, when, bill, grams) as
    (select P.cid, name, city, when,
      cast(sum(price * qty) as decimal(5,2)),
      sum(weight * qty)
    from yrb_customer C, yrb_purchase P, yrb_offer O, yrb_book B
    where C.cid = P.cid and
      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 P.cid, name, city, when),
  rounded (cid, name, city, when, bill, weighin) as
    (select cid, name, city, when, bill, min(weight)
    from orders O, yrb_shipping S
    where weight > grams
    group by cid, name, city, when, bill)
select name, city, cast(when as date) as day, cast(when as time) as time,
  bill, (bill + cost) as total
from rounded R, yrb_shipping S
where R.weighin = S.weight
order by name, city, when;
```

```
-- droppable
-- A club is droppable if all the same purchases in the database could still
-- have been made by the customers, just using the remaining clubs instead.
-- Report each droppable club along with how much more money (or less!) YRB
-- would have made if that club had never existed. Assume that a club
-- "reassignment" for each purchase involving the dropped club replaces it
-- will a best offer (across the remaining clubs) for that customer.
```

```
with
```

```
redo (cid, title, year, when, diff, old, new) as (
  select P.cid, P.title, P.year, P.when,
         (N.price - O.price) * P.qnty,
         O.club, N.club
  from yrb_purchase P, yrb_offer O, yrb_offer N, yrb_member M
  where P.title = O.title and P.year = O.year and
         P.club = O.club and
         P.cid = M.cid and M.club <> P.club and
         P.title = N.title and P.year = N.year and
         M.club = N.club
),
```

```
min_redo (cid, title, year, when, diff, old) as (
  select cid, title, year, when, min(diff), old
  from redo
  group by cid, title, year, when, old
)
```

```
select R.old as club, sum(R.diff) as savings
  from min_redo R
  group by R.old
  having count(*) =
         (select count(*)
          from yrb_purchase P
          where R.old = P.club)
```

```
union
```

```
select club, 0 as savings
  from yrb_club
  where club not in (select club from yrb_purchase)
  order by club;
```

11 YRB 2005 Queries

```
-- 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;
```

12 YRB 2005 Queries

```
-- 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;
```

13 YRB 2005 Queries

```
-- 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;
```

14 YRB 2005 Queries

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

15 YRB 2005 Queries

```
-- nofrench

-- List cities such that no one in that city has purchased any
-- books in French. Do not have duplicates in the answer table.
-- Order by city.

(select distinct city
  from yrb_customer C)
except
(select city
  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 and
        language = 'French')
order by city;
```

```
-- 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;

```

```

-- percentage

-- List each language with the city that has the largest
-- percentage of book sales in that language across
-- cities which have had at least 20 books in total sold.
-- Measure book sales in number of books.

-- For example, say 44% of the books sold in Montreal are
-- French, 22% in Vancouver are French, 18% in Toronto, and so
-- forth, "French Montreal 44" would be in the output.

-- Show language, city, percentage, and number of books
-- (sold in that city in that language) for the output rows.
-- Order by language + city.

with
  Market (language, city, sales) as
    (select language, city, sum(qnty)
     from yrb_customer C, yrb_purchase P, yrb_book B
     where C.cid = P.cid and
           P.title = B.title and P.year = B.year
     group by language, city),
  Total (city, total) as
    (select city, sum(sales)
     from Market M
     group by city
     having sum(sales) >= 20),
  Percent (language, city, percentage) as
    (select language, M.city, ((100 * sales) / total)
     from Market M, Total T
     where M.city = T.city),
  Best (language, high) as
    (select language, max(percentage)
     from Percent P
     group by language)
select P.language, P.city, percentage, sales
from Percent P, Best B, Market M
where P.language = B.language and P.percentage = B.high and
      P.language = M.language and P.city = M.city
order by P.language, P.city;

```

```
-- 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;
```

```
-- overcharge
```

```
with
  best (cid, title, year, lowest) as
    (select distinct M.cid, O.title, O.year, min(price)
     from yrb_member M, yrb_purchase P, yrb_offer O
     where M.club = O.club and
           M.cid = P.cid and
           P.title = O.title and P.year = O.year
     group by M.cid, O.title, O.year)
select C.name, P.title, P.year, qnty, price, lowest
from yrb_customer C, yrb_purchase P, Best B, yrb_offer O
where P.cid = B.cid and P.title = B.title and P.year = B.year and
      P.title = O.title and P.year = O.year and P.club = O.club and
      C.cid = P.cid and
      O.price > B.lowest
order by C.name, P.title, P.year;
```

-- Refunds

```
with
  best (cid, title, year, lowest) as
    (select distinct M.cid, O.title, O.year, min(price)
     from yrb_member M, yrb_purchase P, yrb_offer O
     where M.club = O.club and
           M.cid = P.cid and
           P.title = O.title and P.year = O.year
     group by M.cid, O.title, O.year)
select C.name, C.cid, C.city,
       cast(sum(qnty *(price -lowest)) as decimal(6,2)) as refund
  from yrb_customer C, yrb_purchase P, Best B, yrb_offer O
 where P.cid = B.cid and P.title = B.title and P.year = B.year
       and P.title = O.title and P.year = O.year and P.club = O.club
       and C.cid = P.cid
       and O.price > B.lowest
  group by C.cid, C.name, C.city
 order by C.name;
```