

SQL Visual Lexicon
Peter Whitehouse (wonko@wonko.info)

Why bother with a new metaphor?

- Standard way to deal with databases
- Simple querying easy to learn
- More complex queries need planning

>= 1000 words

A Sample Database

- The "twitter" database

TABLE STRUCTURE

```

  Sycophants: 00 1
  |
  | PratlId PK | follower PK
  |
  | Prats :
  | PratlId PK | name | password | bio | homepage | icon
  | PK | UQ | ma | op | op | ma
  |
  | Prattles:
  | prattleId PK | pratlId | prattle | prattleDate
  | PK | ma | ma | ma
  
```

<http://www.wonko.info/prattler/>

Simple Symbol Set

- Data flow
- Join
- Order by
- Group by
- Count

and so on ... make it up (so long as you are consistent and the symbology is not too elaborate - it has to make sense to you), it is a PLANNING method, not prescriptive

Single Table Queries

- Projection (column subsetting)
- Selection (row subsetting)
- Derivation (working stuff out)

Projection

- List real name, bio and homepage of all prats

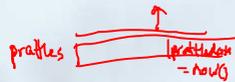


Select name, bio, homepage
from prats

- Column sub-setting, a select * is trivial:

Selection (using the "where" clause, and a little projection)

- List all details of prattles said today



Select *
from prattles
where prattleid = now()

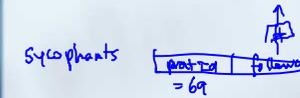
- List the names of people without bios



Select name
from prats
where bio null

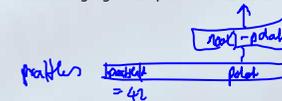
Derivation (working stuff out)

- List the number of followers prattler #69 has



Select count(*)
from sycophants
where prattler = 69

- How long ago was prattle #42 said?



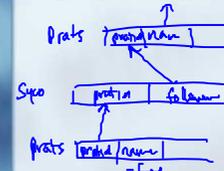
Select now()-prattleid
from prattles
where prattleid = 42

Multi-Table Queries

- Sub-Queries
 - Intersection
 - Difference
- Joins
- Unions
- Correlation

Intersection ("in" queries)

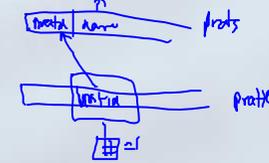
- List the names of all the people who follow Fred



```

Select name
from prats
where
  prat_id in
  (select follower
   from sycophants
   where pratt_id in
    (select pratt_id
     from prats
     where name = 'Fred'))
    
```

- Name prats who lurk (ie. Those who prattled only once)



```

Select name
from prats
where prat_id in
  (select pratt_id
   from prattles
   group by pratt_id
   having count(*) = 1)
    
```

Difference ("not in" queries)

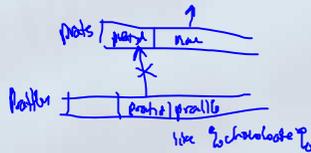
- Name prats who have never prattled



```

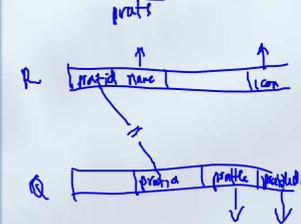
Select name
from prats
where prat_id not in
  (select pratt_id
   from prattles)
    
```

- List prats NOT interested in chocolate



Joins (where data is required from more than ONE table)

- List the name, icon along with what they said and when

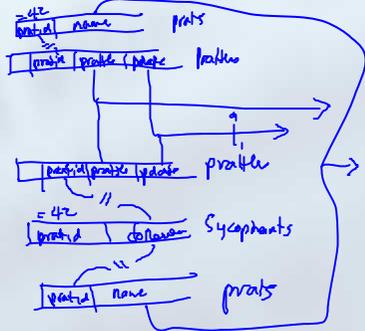


```

select R.name, R.icon, Q.prattile, Q.prattile_date
from prats R, prattles Q
where R.pratt_id = Q.pratt_id
    
```

Union (merging data streams)

- Construct the "feed" for prat #42 (his and his followers posts) in reverse chrono order

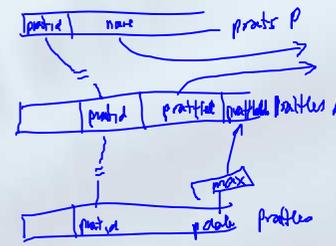


select name, prattle, update
 from prats, prattles
 where prats.prattid = prattles.prattid
 and prattid = 42

union
 select name, prattle, update
 from prats P, Sycophants S, prattles R
 where S.prattid = P.prattid
 and S.prattid = 42
 order by 3 desc

Correlation (join and subq)

- List the name of each Prat along with their most recent prattle



Select P.name, A.prattle
 from prats P, prattles A
 where P.prattid = A.prattid
 and prattle.date in
 (select max(prattle.date)
 from prattles
 where prattid = A.prattid)



Want more?

<http://www.wonko.info/stateconf99/index.html>