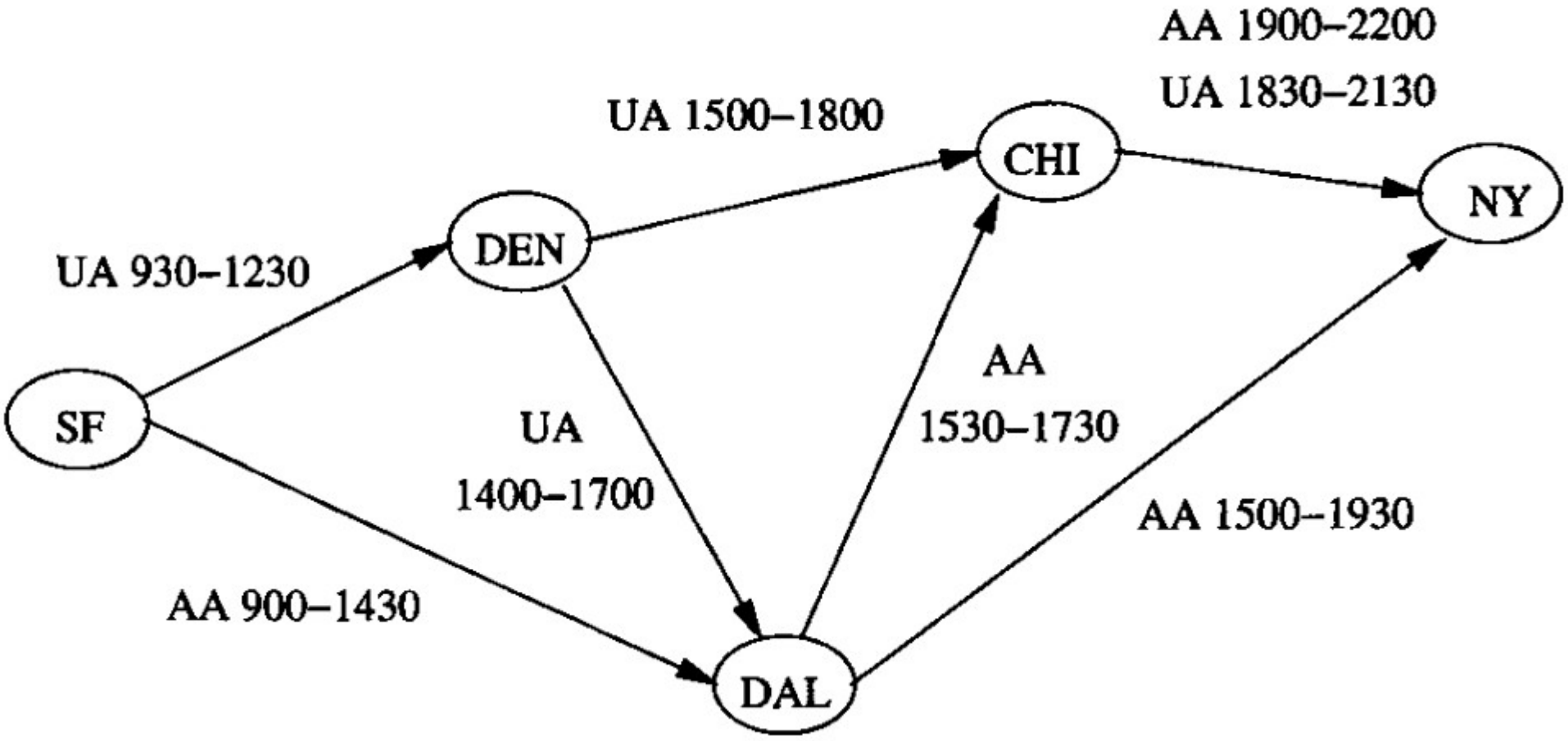
Recursion in SQL

SQL WITH Clause

- create view PC_Maker(model, speed, ram, hd, price, maker) as
  select PC.model, speed, ram, hd, price, maker
  from PC, Product P
  where PC.model = P.model
- **with PC_Maker**(model, speed, ram, hd, price, maker) as
  (select PC.model, speed, ram, hd, price, maker
  from PC, Product P
  where PC.model = P.model)
  select * from **PC_Maker**
- *https://sqlite.org/lang_with.html*

Graph Reachability



Graph Reachability with Recursion

- Flights(orig, dest, depart, arrive)
- Reaches(orig, dest)
  - Reaches(o,d) <= Flights(o,d)
    - base case
  - Reaches(o,d) <= Reaches(o,x) AND Flights(x,d)
    - recursive case
  - Reaches(o,d) <= Reaches(o,x) AND Reaches(x,d)

SQL Recursion

- **with recursive** **Reaches**(orig, dest) as

  (select orig, dest

  from Flights **base case**

  union

  select r.orig, f.dest

  from **Reaches** r, Flights f **recursive case**

  where r.dest = f.orig)

  select * from **Reaches**

Graph Reachability with Constraints

- Reaches(orig, dest, depart, arrive)
  - Reaches(o,d dep, arr) <= Flights(o,d, dep, arr)
    - base case
  - Reaches(o,d, d1, a2) <= Reaches(o,x, d1, a1) AND Flights(x,d, d2, a2) AND d2-a1 > 100
    - recursive case

SQL Query

- with recursive Reaches(orig, dest, depart, arrive) as
  (select orig, dest, depart, arrive
  from Flights
  union
  select r.orig, f.dest, r.depart, f.arrive
  from Reaches r, Flights f
  where r.dest = f.orig
  and f.depart-r.arrive > 100)
  select * from Reaches

Median PC Price with LIMIT

- SELECT AVG(price)

  FROM (

  SELECT price

  FROM PC

  ORDER BY price

  LIMIT 2 - (SELECT COUNT(*) FROM PC) % 2 -- odd 1, even 2

  OFFSET (SELECT (COUNT(*) - 1) / 2 FROM PC)

)

- *https://stackoverflow.com/questions/15763965/how-can-i-calculate-the-median-of-values-in-sqlite*

Median PC Price with Ranking

- **with recursive**
- G(model_1, model_2, diff) as
  (select p1.model, p2.model, p2.price-p1.price
  from PC p1, PC p2
  where p1.price <= p2.price),
- **Hops**(model_1, model_2, hop) as
  (select model_1, model_2, 0
  from G
  union
  select h.model_1, G.model_2, h.hop+1
  from **Hops** h, G
  where h.model_2 = G.model_1
  and G.diff > 0),
- Rank(model, rnk) as
  (select model_2, max(hop)+1 as rank
  from Hops h
  group by model_2
  order by rank)
- select *
  from Rank
  where rnk = (select count(*)/2 from Rank)