COMPSCI 3DB3

Assignment 4

Mingzhe Wang
McMaster University

November 7, 2021

1.

$$\sigma_{enddate>startdate}(event)$$

2.

$$\rho_{R(\text{user},\text{event})}(\pi_{U.\text{uid},E.\text{eid}}(\sigma_{U.\text{postcode}=E.\text{postcode}}(\rho_U(user) \times \rho_E(event))))$$

3.

$$\begin{aligned}
X &:= (event \ltimes_{event.eid=review.event} review)\\
Y &:= \pi_{eid}(event) \setminus \pi_{event.eid}(X)\\
Z &:= event \bowtie Y
\end{aligned}$$

4.

$$\begin{aligned}
X4 &:= \rho_{k_1}(keyword) \times \rho_{k_2}(keyword) \times \rho_{k_3}(keyword) \times \rho_{k_4}(keyword)\\
Y4 &:= \sigma_{k_1.evnet \neq k_2.event \wedge k_1.evnet \neq k_3.event \wedge k_1.evnet \neq k_4.event \wedge k_2.evnet \neq k_3.event \wedge k_2.evnet \neq k_4.event \wedge k_3.evnet \neq k_4.event}(X4)\\
Z4 &:= \pi_{k_1.word=k_2.word \wedge k_2.word=k_3.word \wedge k_3.word=k_4.word}(Y4)\\
X &:= \sigma_{k_1.event,k_1.word}(Z4)\\
X3 &:= \rho_{k_1}(keyword) \times \rho_{k_2}(keyword) \times \rho_{k_3}(keyword)\\
Y3 &:= \sigma_{k_1.evnet \neq k_2.event \wedge k_1.evnet \neq k_3.event \wedge k_2.evnet \neq k_3.event}(X3)\\
Z3 &:= \pi_{k_1.word=k_2.word \wedge k_2.word=k_3.word}(Y3)\\
Y &:= \sigma_{k_1.event,k_1.word}(Z3)\\
Y &\setminus X
\end{aligned}$$

5(a).

$$\begin{aligned}
X &:= \rho_{r_1}(review) \times \rho_{r_2}(review)\\
Y &:= \sigma_{r_1.reviewdata<r_2.reviewdate}(X)\\
Z &:= \pi_{r_3.event}(\rho_{r_3}(review)) \setminus \pi_{r_1.event}(Y)\\
RA &:= \rho_{R(user,event)}(\pi_{review.user,review.event}(review \bowtie Z))
\end{aligned}$$

5(b).

$$
\begin{aligned}
X &:= \rho_{e_1}(event) \times \rho_{e_2}(event) \\
Y &:= \sigma_{e_1.enddate < e_2.enddate}(X) \\
Z &:= \pi_{eid}(event) \setminus \pi_{e_1.eid}(Y) \\
RB &:= \rho_{R(user,event)}(\pi_{r.user,r.event}(\rho_r(review) \bowtie_{r.event=e.eid} \rho_e(Z)))
\end{aligned}
$$

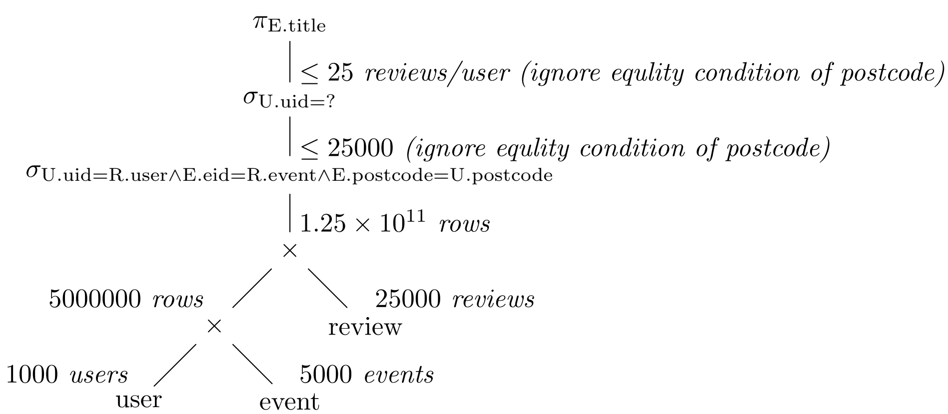
5(c).

$$\rho_{u.uid \mapsto user, a.event \mapsto lreview, b.event \mapsto levent}(\sigma_{a.user=b.user \wedge b.user=u.uid}(\rho_a(RA) \times \rho_b(RB) \times \rho_u(user)))$$

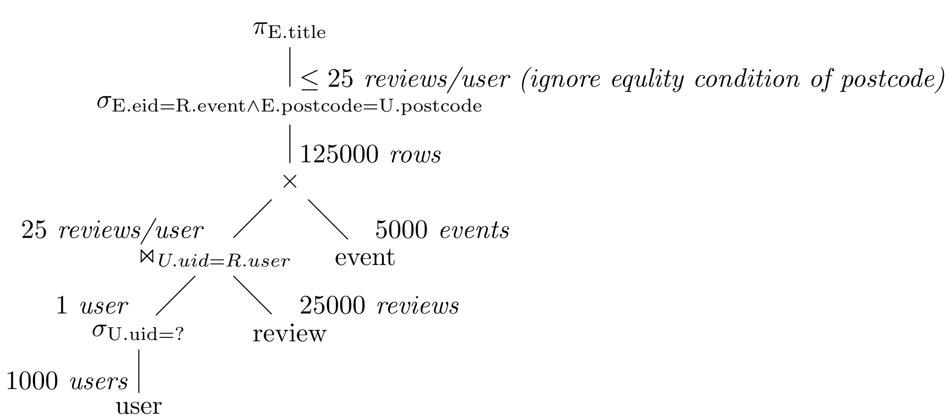
6.

$$
\begin{aligned}
X &:= \sigma_{U.postcode=E.postcode}(\rho_U(user) \times \rho_E(event)) \\
Y &:= \pi_{U.uid,E.eid}(X) \setminus \rho_{user \mapsto uid, event \mapsto eid}(review) \\
&\pi_{uid}(user) \setminus \pi_{U.uid}(Y)
\end{aligned}
$$

7.



8.



Why is good?
We push down selectors to filter the student earlier, thus reducing the number of rows of intermediate tables. Note the maximum of row number of the original execution plan is 1.25×10^{11}, while that of this execution plan is 125000. Assume look up one row of the table takes constant time t, the execution time is reduced significantly.

9. As marked in 8.

There is 1000 users for *user* table, after filter by $\sigma_{\text{U.uid=?}}$, only one row left because the *uid* in *user* table is unique. Then the result is further joined with *review* table, after cartisian product, the result should have 25000 rows, however, when filtered by the join condition, there should be approximately 25 rows left, because we estimate 25 reviews/user. Then we do the cartisian product gain with events, which has 5000 rows, the rows before filter should be 125000, after filter, the row number will be ≤ 25, if we take into account the equality condition of postcode.

10.

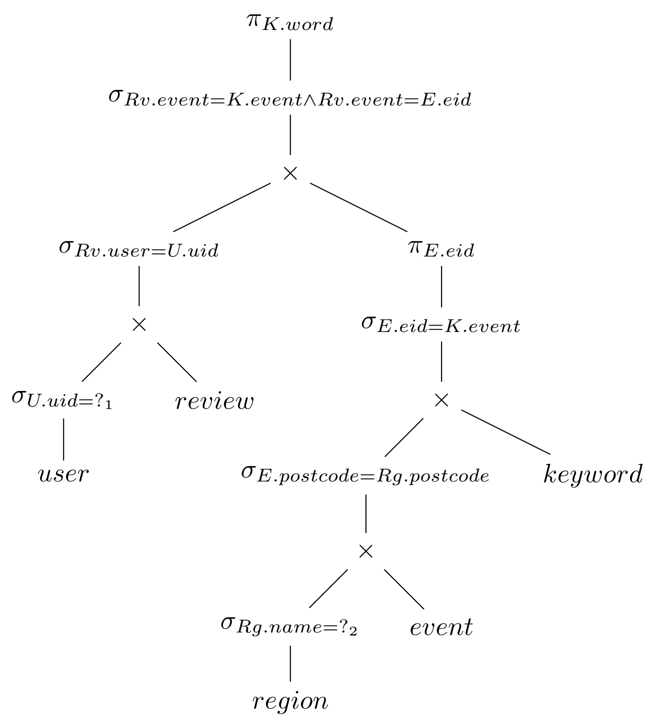
```
SELECT e.title
FROM
  user AS u
  INNER JOIN event AS e ON u.postcode = e.postcode
  INNER JOIN review AS r ON u.uid = r.user e.eid = r.event
WHERE u.uid = ?; -- specificial user
```

11.

Note: this is a simplified version. (The method following this article: Translating SQL into the Relational Algebra.)

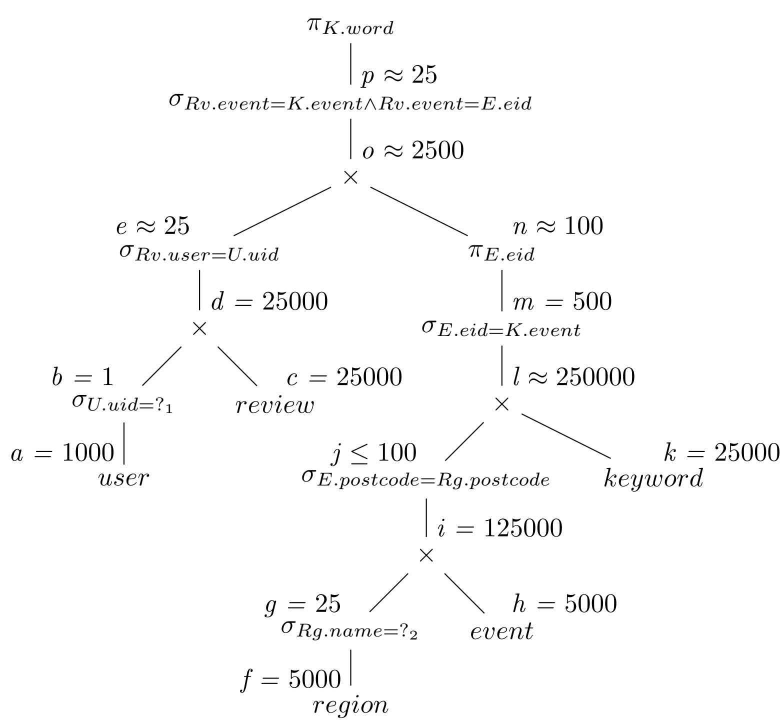
$$Subquery := \pi_{E.eid}(\sigma_{E.postcode=Rg.postcode \wedge Rg.name=?_2 \wedge E.eid=K.event}(\rho_E(event) \times \rho_{Rg}(region) \wedge \rho_K(keyword)))$$
$$X := \rho_U(user) \times \rho_{Rv}(review)$$
$$Y := X \bowtie_{K.event=E.eid} subquery$$
$$\pi_{K.word}(\sigma_{U.uid=?_1 \wedge Rv.user=U.uid \wedge Rv.event=K.event}\, Y)$$

12.



Why is good?
We push down selectors and projectors to filter all intermediate tables earlier, thus reducing the number of rows of intermediate tables by pushing down selectors and reducing the numbers of attributes we need to store by pushing down projectors.

13.



Reason

a 1000 users

b uid is unique

c 25000 reviews

d $|T \times U| = mn$.

e the result should be equivalent to user natural join review – 25 review/user

f 5000 rows in region table

g $num.of.postcode \div num.of.region = 1250/50 = 25\, postcodes/region$

h 5000 events

i $|T \times U| = mn$.

j $num.of.event \div num.of.region = 5000/50 = 100$, when taking into account the postcode equality condition, the result should be ≤ 100.

k 25000 keywords.

l $|T \times U| = mn$.

m there are 100 rows in the left branch of this subtree, and we know 5 keywords/event, so totally there should be 500 rows.

n because 5 keywords/event, and we are using set semantics, so 500 / 5 = 100 rows.

o $|T \times U| = mn$.

p because 25 reviews / user and 5 reviews / event, we have 5 events / user, then by 5 keywords / event, we know 25 keywords / user.