Assignment 4: The Relational Algebra

Hien Tu - tun1

November 6, 2021

The requested queries

1. $\sigma_{\text{startdate} < \text{enddate}}(\text{event})$

2. $\pi_{\text{U.uid,E.eid}}(\rho_{\text{U}}(\text{user}) \bowtie_{\text{U.postcode = E.postcode}} \rho_{\text{E}}(\text{event}))$

3. $X := \rho_{\text{E}}(\text{event}) \bowtie_{\text{E.eid = EnoRv.eid}} \rho_{\text{EnoRv}}(\pi_{\text{eid}}(\text{event}) \setminus \pi_{\text{event}}(\text{review}))$
   $\pi_{\text{E.eid,E.title,E.description,E.startdate,E.enddate,E.organizer,E.postcode}}(X)$

4. Select events with at least 3 keywords:
   $X := \rho_{\text{K}_1}(\text{keyword}) \times \rho_{\text{K}_2}(\text{keyword}) \times \rho_{\text{K}_3}(\text{keyword})$
   $Y := \sigma_{K_1.\text{word} \neq K_2.\text{ word} \wedge K_1.\text{word} \neq K_3.\text{ word} \wedge K_2.\text{word} \neq K_3.\text{ word}}(X)$
   $Z := \sigma_{K_1.\text{event} = K_2.\text{event} \wedge K_1.\text{event} = K_3.\text{event}}(Y)$

   Select events with at least 2 keywords:
   $A := \rho_{\text{K}_4}(\text{keyword}) \times \rho_{\text{K}_4}(\text{keyword})$
   $B := \sigma_{K_4.\text{word} \neq K_5.\text{word}}(A)$
   $C := \sigma_{K_4.\text{event} = K_5.\text{event}}(B)$

   Select events with exactly 3 keywords (final query):
   $\pi_{K_4.event}(C) \setminus \pi_{K_1.event}(Z)$

5. (a) $X := \rho_{R_1}(\text{review}) \times \rho_{R_2}(\text{review})$

   Keep reviews from R_1 that are not from latest date:
   $Y := \sigma_{R_1.\text{reviewdate} < R_2.\text{reviewdate} \wedge R_1.\text{user} = R_2.\text{user}}(X)$

   Select user id and event id for which the user wrote a review most recently (final query):
   $Z := \pi_{\text{user,event}}(\text{review}) \setminus \pi_{R_1.\text{user},R_1.\text{event}}(Y)$

(b) $A := \rho_{R_1}(\text{review}) \times \rho_{R_2}(\text{review}) \times \rho_{E_1}(\text{event}) \times \rho_{E_2}(\text{event})$
$B := \sigma_{R_1.\text{user} = R_2.\text{user} \wedge R_1.\text{event} = E_1.\text{eid} \wedge R_2.\text{event} = E_2.\text{eid}}(A)$

Select reviews whose E_1.enddate are not from the latest:
$C := \sigma_{E_1.\text{enddate} < E_2.\text{enddate}(B)}$

Select user id and event id of the most-recent event (according to enddate) for which the user wrote a review (final query):
$D := \pi_{R_1.\text{user}, E_1.\text{eid}}(B \setminus C)$

(c) $\pi_{\text{LR.user,LR.lreview,LE.levent}}(\rho_{\text{LR(user,lreview)}}(Z) \bowtie_{\text{LR.user} = \text{LE.user}} \rho_{\text{LE(user,levent)}}(D))$

6. $A := \rho_{\text{Rv}}(\text{review}) \times \rho_{\text{U}}(\text{user}) \times \rho_{\text{E}}(\text{event})$

Select reviews where the user has the same postal code as the event:
$B := \sigma_{\text{Rv.user=U.uid} \wedge \text{Rv.event=E.eid} \wedge \text{U.postcode=E.postcode}}(A)$

$C := \pi_{\text{Rv.user,Rv.event}}(B)$
$D := \rho_{U_2}(\text{user}) \bowtie_{U_2.\text{postcode} = E_2.\text{postcode}} \rho_E(\text{event})$

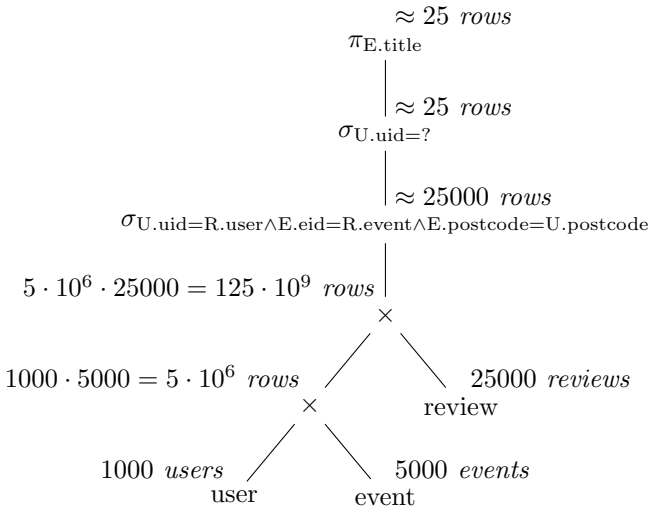
Select *uid* and *eid* where user has the same postal code as the event:
$E := \pi_{U_2.\text{uid}, E_2.\text{eid}}(D)$

Select *uid* and *eid* where the user has the same postal code as the event but haven't written review for that event:
$F := E \setminus C$

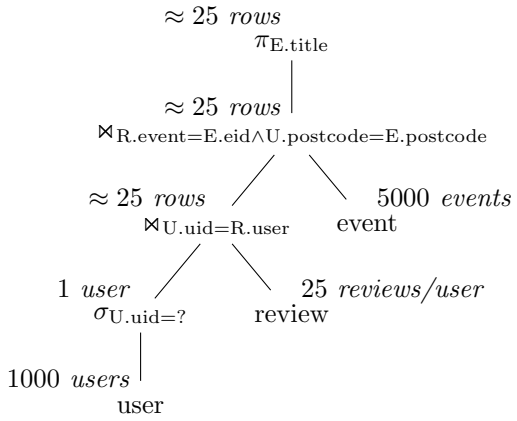
Final query:
$\pi_{\text{uid}}(user) \setminus \pi_{U_2.\text{uid}}(F)$

Effiency of queries

7.



8.



From the query execution plan in question 7, we can see that the result of the cross product of the 3 tables, *user*, *review* and *event* would have $125 \cdot 10^9$ rows, which is enormous. We can reduce this significantly by first selecting the row from the table *user* where the *uid* is equal to the required user id (in this case, denoted as ?). This will result in only 1 rows since each *uid* is unique. After that, we will join the result with the *review*. Although the table *review* has 25000 rows, each user only has 25 reviews and since we have selected only 1 user, the result of this join would have approximately 25 rows. Then, we join this result with the table *event*. Again, although the table *event* has 5000 rows, since the previous result only has approximately 25 rows, there are only approximately 25 different event id. Thus, the result of this join only contains approximately 25 rows.

Projecting just takes the specified column so the result of the whole query execution plan would have approximately 25 rows. This execution plan is good since the returned table in each intermediate step is significantly small.

9. The answer for question 9 is given in the answer for question 8.

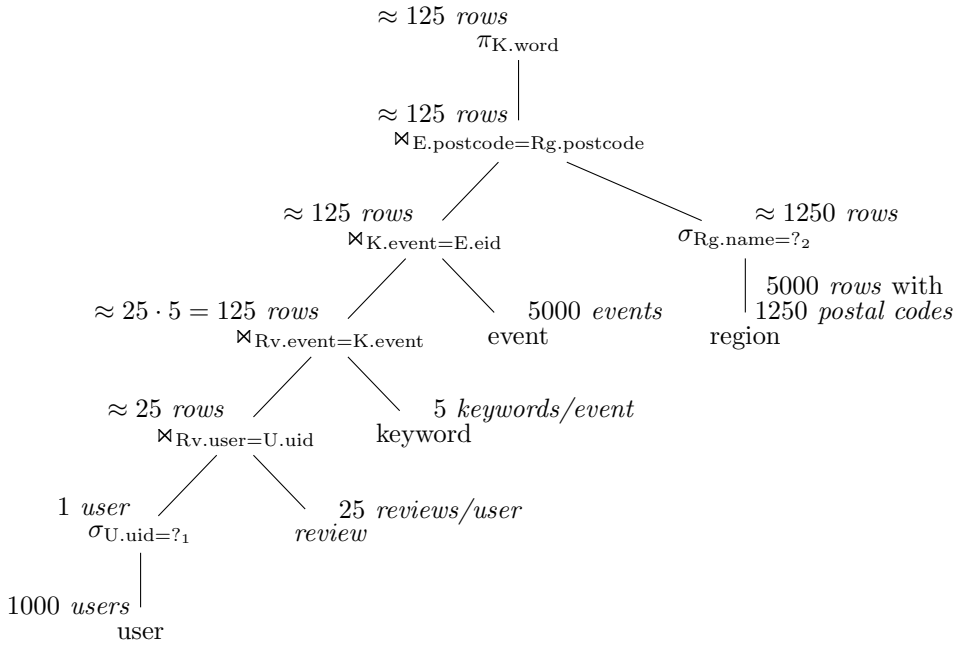
10.
```
SELECT DISTINCT E.title
FROM user U, event E, review R
WHERE U.uid = ? AND
        R.user = U.uid AND
        R.event = E.eid AND
        U.postcode = E.postcode;
```

11. The given SQL query can be simplified to

```
SELECT DISTINCT K.word
FROM user U, review Rv, keyword K, event E, region
Rg
WHERE U.uid = ?1 AND
        Rv.user = U.uid AND
        Rv.event = K.eid AND
        E.postcode = Rg.postcode AND
        Rg.name = ?2 AND
        K.event = E.eid;
```

Thus, the relational algebra for the SQL query is
$X := \rho_{\mathrm{U}}(\text{user}) \times \rho_{\mathrm{Rv}}(\text{review}) \times \rho_{\mathrm{K}}(\text{keyword}) \times \rho_{\mathrm{E}}(\text{event}) \times \rho_{\mathrm{Rg}}(\text{region})$
$Y := \sigma_{\mathrm{U.uid}=?_1 \wedge \mathrm{Rv.user}=\mathrm{U.uid} \wedge \mathrm{Rv.event}=\mathrm{K.event} \wedge \mathrm{E.postcode}=\mathrm{Rg.postcode} \wedge \mathrm{Rg.name}=?_2 \wedge \mathrm{K.event}=\mathrm{E.eid}}(X)$
$\pi_{\mathrm{K.word}}(Y)$

12.



This query execution plan is good since the result in each intermediate step is significantly small comparing to the size of the original tables. First, from the table *user*, we select the user whose *uid* is the same as the required user (in this case, denoted as $?_1$). Since each *uid* is unique and we only select one user, the result only contains 1 rows. Then, we will join (conditional join) the result with the *review* table. Since the previous result returns 1 user and it is estimated that each user writes around 25 reviews, the result of this only contains approximately 25 rows, that is 25 reviews on 25 events that the specified user writes. After that, we will perform conditional join the result with the table *keyword*. Since each event that the specified user writes has 5 keywords associated with it and there are 25 events that the user writes review about, the result table would have $25 \cdot 5 = 125$ rows. Then, we will conditional join with the table *event*. Since there are only 125 rows in our above query, we only have 125 rows in the resulting query (each row in the above query will be augmented with the event info from the *event* table). Let's call this query (A) to use later. Separately, we will select the from the *region* table whose region name is the same as the required region name (in this case, denoted as $?_2$). In the worst cases, the required region is associated with all 1250 distinct postal codes. Then, the result of the select would contain 1250 rows. Let's call this result (B). Now, we will conditional join the query (A) with the query (B). Since the result of query (A) only contain 125 rows, there are only 125 postal codes in it. Thus, there will only be around 125 postal codes from (B) that matches the ones in (A). Therefore, the result

would have around 125 rows. Since projecting does not change the rows, our final result for this query execution plan would have approximately 125 rows.

13. The answer for question 13 is given in the answer for question 12 (above).