GATE 2025 DBMS

June 2024

Relational Algebra & Tuple Relational Calculus (TRC)

Solutions

Solution. *1*

1. $R1 \cup R2$

   Assumption: *$R1$ and $R2$ have the same schema.*

   Explanation: *The union operation combines all tuples from both $R1$ and $R2$, eliminating duplicates.*

   Minimum size: $\max(N1, N2)$
   This occurs if all tuples in $R1$ are also in $R2$, or vice versa.

   Maximum size: $N1 + N2$
   This occurs if $R1$ and $R2$ are disjoint (no common tuples).

2. $R1 \cap R2$

   Assumption: *$R1$ and $R2$ have the same schema.*

   Explanation: *The intersection operation returns only the tuples that are present in both $R1$ and $R2$.*

   Minimum size: *0*
   This occurs if $R1$ and $R2$ are disjoint (no common tuples).

   Maximum size: $N1$
   This occurs if all tuples in $R1$ are also present in $R2$.

3. $R1 - R2$

   Assumption: *$R1$ and $R2$ have the same schema.*

   Explanation: *The difference operation returns tuples in $R1$ that are not in $R2$.*

   Minimum size: *0*
   This occurs if all tuples in $R1$ are also in $R2$.

   Maximum size: $N1$
   This occurs if none of the tuples in $R1$ are in $R2$.

4. $R1 \times R2$

   Assumption: *No specific assumption about schema other than the tuples can be concatenated.*

   Explanation: *The Cartesian product operation pairs every tuple in $R1$ with every tuple in $R2$.*

   Minimum size: $N1 \cdot N2$
   This is the inherent size of the Cartesian product, as every tuple in $R1$ pairs with every tuple in $R2$.

   Maximum size: $N1 \cdot N2$
   The size is always $N1 \cdot N2$, regardless of the content.

5. $\sigma_{a=5}(R1)$

   Assumption: *Attribute a exists in $R1$.*

   Explanation: *The selection operation returns tuples in $R1$ where attribute a equals 5.*

   Minimum size: *0*
   This occurs if no tuples in $R1$ satisfy the condition $a = 5$.

   Maximum size: $N1$
   This occurs if all tuples in $R1$ satisfy the condition $a = 5$.

6. $\pi_a(R1)$

   Assumption: *Attribute a exists in $R1$.*

   Explanation: *The projection operation returns a relation containing only the attribute a from $R1$, eliminating duplicates.*

   Minimum size: *1*
   This occurs if all tuples in $R1$ have the same value for a.

   Maximum size: $N1$
   This occurs if all tuples in $R1$ have distinct values for a.

7. *$R1/R2$*

   Assumption: *The schema of $R2$ is a subset of the schema of $R1$.*

   Explanation: *The division operation returns tuples in $R1$ that match all tuples in $R2$ for the attributes in $R2$.*

   Minimum size: *0*
   This occurs if there is no tuple in $R1$ that matches all tuples in $R2$.

   Maximum size: $N1$
   This occurs if for every tuple in $R1$, there exists a matching tuple in $R2$ such that the division condition is satisfied.

Solution. *2*
In the answers below, RA refers to Relational Algebra, TRC refers to Tuple Relational Calculus, and DRC refers to Domain Relational Calculus.

1. ***Find the names of suppliers who supply some red part.***
   - *RA:* $\pi_{sname}((\pi_{sid}((\pi_{pid}(\sigma_{color='red'}(Parts))) \bowtie Catalog)) \bowtie Suppliers)$
   - *TRC:* $\{T \mid \exists T_1 \in Suppliers(\exists X \in Parts(X.color = 'red' \wedge \exists Y \in Catalog(Y.pid = X.pid \wedge Y.sid = T_1.sid)) \wedge T.sname = T_1.sname)\}$

2. ***Find the sids of suppliers who supply some red or green part.***
   - *RA:* $\pi_{sid}((\pi_{pid}(\sigma_{color='red'} \vee color = 'green'(Parts))) \bowtie Catalog)$
   - *TRC:* $\{T \mid \exists T_1 \in Catalog(\exists X \in Parts((X.color = 'red' \vee X.color = 'green') \wedge X.pid = T_1.pid) \wedge T.sid = T_1.sid)\}$

3. ***Find the sids of suppliers who supply some red part or are at 221 Packer Ave.***
   - *RA:* $\rho(R_1, \pi_{sid}((\pi_{pid}(\sigma_{color='red'}(Parts))) \bowtie Catalog)) \cup \rho(R_2, \pi_{sid}(\sigma_{address='221\ Packer\ Ave.'}(Suppliers)))$
   - *TRC:* $\{T \mid \exists T_1 \in Catalog(\exists X \in Parts(X.color = 'red' \wedge X.pid = T_1.pid) \wedge T.sid = T_1.sid) \vee \exists T_2 \in Suppliers(T_2.address = '221\ Packer\ Ave.' \wedge T.sid = T_2.sid)\}$

4. ***Find the sids of suppliers who supply some red part and some green part.***
   - *RA:* $\rho(R_1, \pi_{sid}((\pi_{pid}(\sigma_{color='red'}(Parts))) \bowtie Catalog)) \cap \rho(R_2, \pi_{sid}((\pi_{pid}(\sigma_{color='green'}(Parts))) \bowtie Catalog))$
   - *TRC:* $\{T \mid \exists T_1 \in Catalog(\exists X \in Parts(X.color = 'red' \wedge X.pid = T_1.pid) \wedge \exists T_2 \in Catalog(\exists Y \in Parts(Y.color = 'green' \wedge Y.pid = T_2.pid) \wedge T_2.sid = T_1.sid) \wedge T.sid = T_1.sid)\}$

5. ***Find the sids of suppliers who supply every part.***
   - *RA:* $(\pi_{sid,pid}(Catalog)) \div (\pi_{pid}(Parts))$
   - *TRC:* $\{T \mid \exists T_1 \in Catalog(\forall X \in Parts(\exists T_2 \in Catalog(T_2.pid = X.pid \wedge T_2.sid = T_1.sid)) \wedge T.sid = T_1.sid)\}$

6. ***Find the sids of suppliers who supply every red part.***
   - *RA:* $(\pi_{sid,pid}(Catalog)) \div (\pi_{pid}(\sigma_{color='red'}(Parts)))$
   - *TRC:* $\{T \mid \exists T_1 \in Catalog(\forall X \in Parts(X.color \neq 'red' \vee \exists T_2 \in Catalog(T_2.pid = X.pid \wedge T_2.sid = T_1.sid)) \wedge T.sid = T_1.sid)\}$

7. ***Find the sids of suppliers who supply every red or green part.***
   - *RA:* $(\pi_{sid,pid}(Catalog)) \div (\pi_{pid}(\sigma_{color='red'} \vee color = 'green'(Parts)))$
   - *TRC:* $\{T \mid \exists T_1 \in Catalog(\forall X \in Parts((X.color \neq 'red' \wedge X.color \neq 'green') \vee \exists T_2 \in Catalog(T_2.pid = X.pid \wedge T_2.sid = T_1.sid)) \wedge T.sid = T_1.sid)\}$

8. ***Find the sids of suppliers who supply every red part or supply every green part.***
   - *RA:* $\rho(R_1, (\pi_{sid,pid}(Catalog)) \div (\pi_{pid}(\sigma_{color='red'}(Parts)))) \cup \rho(R_2, (\pi_{sid,pid}(Catalog)) \div (\pi_{pid}(\sigma_{color='green'}(Parts)))$

- *TRC:* $\{T \mid \exists T_1 \in Catalog((\forall X \in Parts(X.color \neq \text{'red'} \vee \exists Y \in Catalog(Y.pid = X.pid \wedge Y.sid = T_1.sid)) \vee \forall Z \in Parts(Z.color \neq \text{'green'} \exists P \in Catalog(P.pid = Z.pid \wedge P.sid = T_1.sid))) \wedge T.sid = T_1.sid)\}$

9. ***Find pairs of sids such that the supplier with the first sid charges more for some part than the supplier with the second sid.***
   - *RA:* $\rho(R_1, Catalog)\rho(R_2, Catalog)\pi_{R1.sid,R2.sid}(\sigma_{R1.pid=R2.pid \wedge R1.sid \neq R2.sid \wedge R1.cost > R2.cost}(R_1 \times R_2))$
   - *TRC:* $\{T \mid \exists T_1 \in Catalog(\exists T_2 \in Catalog(T_2.pid = T_1.pid \wedge T_2.sid \neq T_1.sid \wedge T_2.cost < T_1.cost \wedge T.sid2 = T_2.sid) \wedge T.sid1 = T_1.sid)\}$
   - *DRC:* $\{\langle X, P\rangle \mid \langle X, Y, Z\rangle \in Catalog \wedge \exists P, Q, R(\langle P, Q, R\rangle \in Catalog \wedge Q = Y \wedge P \neq X \wedge R < Z)\}$

10. ***Find the pids of parts that are supplied by at least two different suppliers.***
    - *RA:* $\rho(R_1, Catalog)\rho(R_2, Catalog)\pi_{R1.pid}(\sigma_{R1.pid=R2.pid \wedge R1.sid \neq R2.sid}(R_1 \times R_2))$
    - *TRC:* $\{T \mid \exists T_1 \in Catalog(\exists T_2 \in Catalog(T_2.pid = T_1.pid \wedge T_2.sid \neq T_1.sid) \wedge T.pid = T_1.pid)\}$

11. ***Find the pids of the most expensive parts supplied by suppliers named Yosemite Sham.***
    - *RA:* $\rho(R_1, \pi_{sid}(\sigma_{sname=\text{'Yosemite Sham'}}(Suppliers)))\rho(R_2, R_1 \bowtie Catalog)\rho(R_3, R_2)\rho(R_4(1 \to sid, 2 \to pid, 3 \to cost), \sigma_{R3.cost<R2.cost}(R_3 \times R_2))\pi_{pid}(R_2 - \pi_{sid,pid,cost}(R_4))$
    - *TRC:* $\{T \mid \exists T_1 \in Catalog(\exists X \in Suppliers(X.sname = \text{'Yosemite Sham'} \wedge X.sid = T_1.sid) \wedge \neg(\exists S \in Suppliers(S.sname = \text{'Yosemite Sham'} \wedge \exists Z \in Catalog(Z.sid = S.sid \wedge Z.cost > T_1.cost))) \wedge T.pid = T_1.pid)\}$

12. ***Find the pids of parts supplied by every supplier at less than \$200.***
    - *RA:* $\rho(R_1, \sigma_{cost<200}(Catalog))\pi_{pid}(\rho(R_2, \pi_{sid,pid}(Catalog)) \div \pi_{sid,pid}(R_1))$
    - *TRC:* $\{T \mid \forall T_1 \in Suppliers(\exists T_2 \in Catalog(T_2.sid = T_1.sid \wedge T_2.pid = T.pid \wedge T_2.cost < 200))\}$

Solution. *3*

1. $\pi_{sname}(\pi_{sid}(\sigma_{color=\text{'red'}}(Parts)) \bowtie (\sigma_{cost<100}(Catalog)) \bowtie Suppliers)$

   Solution: *This query finds the names of suppliers who supply any red part for less than \$100. First, it selects red parts from the Parts table, then finds entries in the Catalog where the cost is less than \$100, and finally joins with the Suppliers table to project the supplier names.*

2. $\pi_{sname}(\pi_{sid}((\sigma_{color=\text{'red'}}(Parts)) \bowtie (\sigma_{cost<100}(Catalog)) \bowtie Suppliers))$

   Solution: *This query finds the names of suppliers who supply red parts for less than \$100. The main difference from the previous query is the order of projection. Here, the sid projection happens after the joins, resulting in the same output due to the nature of the relational operations.*

3. $(\pi_{sname}((\sigma_{color=\text{'red'}}(Parts)) \bowtie (\sigma_{cost<100}(Catalog)) \bowtie Suppliers)) \cap (\pi_{sname}((\sigma_{color=\text{'green'}}(Parts)) \bowtie (\sigma_{cost<100}(Catalog)) \bowtie Suppliers))$

   Solution: *This query finds the names of suppliers who supply both red and green parts, each for less than \$100. It performs two separate selections for red and green parts, respectively, and then computes the intersection of supplier names from both results.*

4. $(\pi_{sid}((\sigma_{color=\text{'red'}}(Parts)) \bowtie (\sigma_{cost<100}(Catalog)) \bowtie Suppliers)) \cap (\pi_{sid}((\sigma_{color=\text{'green'}}(Parts)) \bowtie (\sigma_{cost<100}(Catalog)) \bowtie Suppliers))$

   Solution: *This query finds the sids of suppliers who supply both red and green parts, each for less than \$100. It selects red and green parts separately, then finds the intersection of sids from both results.*

5. $\pi_{sname}((\pi_{sid,sname}((\sigma_{color=\text{'red'}}(Parts)) \bowtie (\sigma_{cost<100}(Catalog)) \bowtie Suppliers)) \cap (\pi_{sid,sname}((\sigma_{color=\text{'green'}}(Parts)) \bowtie (\sigma_{cost<100}(Catalog)) \bowtie Suppliers)))$

   Solution: *This query finds the names of suppliers who supply both red and green parts for less than \$100. It first joins for red parts and green parts, respectively, to get a relation of sids and snames, and then takes the intersection of these relations, followed by a projection on sname.*

Solution. *4*

1. *Express the following English queries in TRC:*

   (a) ***What are the department abbreviations and titles of all courses numbered 101?***

   $$\{c.Dept, c.Title \mid Course(c) \land c.Num = 101\}$$

   Explanation: *This query retrieves the department abbreviations and titles of all courses that have the course number 101 by selecting tuples from the Course relation where the Num attribute is equal to 101.*

   (b) ***What are the IDs of the MATH students taking "Discrete Structures?"***

   $$\{s.ID \mid Student(s) \land s.Major =' MATH' \land (\exists e)(Enroll(e) \land s.ID = e.SID \land (\exists c)(Course(c) \land e.Dept = c.Dept \land$$

   Explanation: *This query finds the IDs of students who are majoring in MATH and are enrolled in a course titled "Discrete Structures." It selects tuples from the Student relation where the Major attribute is 'MATH,' and checks for corresponding enrollments and courses matching the title 'Discrete Structures.'*

2. *For each of these TRC queries, what is the corresponding query in conversational English?*

   (a) $\{$**s.Name, s.ID** $\mid Student(s) s.Major =' CSC'\}$
   Answer: *What are the names and IDs of the CSC majors?*
   Explanation: *This query retrieves the names and IDs of all students whose major is 'CSC' by selecting tuples from the Student relation where the Major attribute is 'CSC.'*

   (b) $\{$**e.Date** $\mid Enroll(e)(c)(Course(c)e.Dept = c.Depte.Num = c.Numc.Title =' DatabaseDesign')\}$
   Answer: *On which dates did students enroll in "Database Design?"*
   Explanation: *This query retrieves the dates on which students enrolled in the course titled "Database Design" by selecting tuples from the Enroll relation where there is a corresponding course in the Course relation with the title 'Database Design.'*