

Relational Algebra

Query 2: Write a query that returns the Subject attributes of all subjects for which no book has been written by any author.

$\pi_{\text{Subject}}(\text{Subjects}) - \pi_{\text{Subject}}(\text{Books} \bowtie \text{Subjects})$

(note: Authors table can be included in join statement)

Query 3: Write a query that finds the ISBNs of all editions of books written by Agatha Christie.

$\pi_{\text{ISBN}}(\text{Editions} \bowtie \text{Books} \bowtie \sigma_{\text{First_Name}='Agatha' \wedge \text{Last_Name}='Christie'}(\text{Authors}))$

Query 4: Write a query that finds the first and last names of all authors who have written at least one children's/young adult book (subject: "Children/YA") and at least one book of fiction (subject: "Fiction").

$\rho(\text{C_Author}, \pi_{\text{Author_ID}, \text{First_Name}, \text{Last_Name}}(\text{Authors} \bowtie \text{Books} \bowtie \sigma_{\text{Subject}='Children/YA'}(\text{Subjects})))$

$\rho(\text{F_Author}, \pi_{\text{Author_ID}, \text{First_Name}, \text{Last_Name}}(\text{Authors} \bowtie \text{Books} \bowtie \sigma_{\text{Subject}='Fiction'}(\text{Subjects})))$

$\pi_{\text{First_Name}, \text{Last_Name}}(\text{C_Author} \cap \text{F_Author})$

(note: renaming operation not necessary)

Query 5: Write a query that finds the IDs, first names, and last names of all authors who have written at least one book in every subject for which J. K. Rowling has written at least one book, including J.K. Rowling.

$\rho(\text{Like_JK}, (\pi_{\text{Author_ID}, \text{Subject_ID}}(\text{Books}) / \pi_{\text{Subject_ID}}(\text{Books} \bowtie \sigma_{\text{First_Name}='J.K.' \wedge \text{Last_Name}='Rowling'}(\text{Authors}))))$

$\pi_{\text{Author_ID}, \text{First_Name}, \text{Last_Name}}(\text{Authors} \bowtie \text{Like_JK})$

(note: renaming operation not necessary)

Relational Calculus

Query 2: Write a query that returns the Subject attributes of all subjects for which no book has been written by any author.

$\{T | \exists S \in \text{Subjects}, \forall B \in \text{Books} (S.\text{Subject ID} \neq B.\text{Subject ID}) \wedge (T.\text{Subject ID} = S.\text{Subject ID})\}$

Query 3: Write a query that finds the ISBNs of all editions of books written by Agatha Christie.

$\{T | \exists E \in \text{Editions}, \exists B \in \text{Books}, \exists A \in \text{Authors} (T.\text{ISBN} = E.\text{ISBN} \wedge E.\text{Book ID} = B.\text{Book ID} \wedge B.\text{Author ID} = A.\text{Author ID} \wedge A.\text{First name} = \text{'Agatha'} \wedge A.\text{Last name} = \text{'Christie'})\}$

Query 4: Write a query that finds the first and last names of all authors who have written at least one children's/young adult book (subject: "Children/YA") and at least one book of fiction (subject: "Fiction").

$\{T | (\exists A \in \text{Authors}, \exists B \in \text{Books}, \exists S_1 \in \text{Subjects} (T.\text{First name} = A.\text{First name} \wedge T.\text{Last name} = A.\text{Last name} \wedge T.\text{Author ID} = A.\text{Author ID} \wedge A.\text{Author ID} = B.\text{Author ID} \wedge B.\text{Subject ID} = S_1.\text{Subject ID} \wedge S_1.\text{Subject} = \text{'Children/YA'})) \wedge (\exists A \in \text{Authors}, \exists B \in \text{Books}, \exists S_2 \in \text{Subjects} (T.\text{First name} = A.\text{First name} \wedge T.\text{Last name} = A.\text{Last name} \wedge T.\text{Author ID} = A.\text{Author ID} \wedge A.\text{Author ID} = B.\text{Author ID} \wedge B.\text{Subject ID} = S_2.\text{Subject ID} \wedge S_2.\text{Subject} = \text{'Fiction'}))\}$

Normalization

1. {id}, {name, school}, {name, major}, {name, GPA}
2. 3NF
3. Not BCNF
4. {GPA → major}, {school → major}