Database Management System (DBMS)

Lecture-16

Dharmendra Kumar

September 7, 2020

The Union Operation

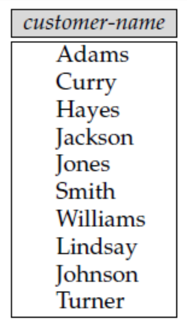
Union operation will be applied if we have to find elements which are belong into either of one relation.
For a union operation r ∪ s to be valid, we require that two conditions hold:

1. The relations r and s must be of the same arity. That is, they must have the same number of attributes.
2. The domains of the i^{th} attribute of r and the i^{th} attribute of s must be the same, for all i.

Relational Algebra

Example: Find the names of all bank customers who have either an account or a loan or both.

Solution: $\Pi_{customer}(depositer) \cup \Pi_{customer}(borrower)$



customer-name
Adams
Curry
Hayes
Jackson
Jones
Smith
Williams
Lindsay
Johnson
Turner

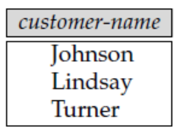
The Set Difference Operation

The set difference operation, denoted by -, allows us to find tuples that are in one relation but are not in another. The expression r - s produces a relation containing those tuples in r but not in s.

Relational Algebra

Example: Find all customers of the bank who have an account but not a loan.

Solution: $\Pi_{customer}(depositer) - \Pi_{customer}(borrower)$



customer-name
Johnson
Lindsay
Turner

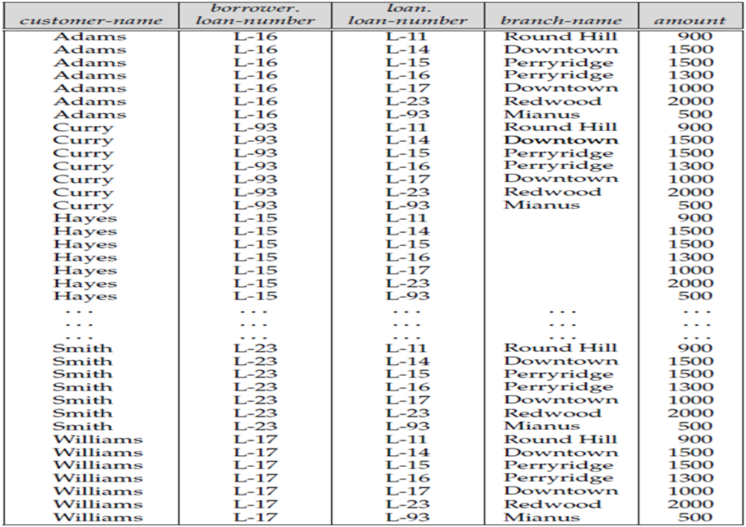
As with the union operation, we must ensure that set differences are taken between compatible relations. Therefore, for a set difference operation r - s to be valid, we require that the relations r and s be of the same arity, and that the domains of the i^{th} attribute of r and the i^{th} attribute of s be the same.

Relational Algebra

The Cartesian-Product Operation

The Cartesian-product operation, denoted by a cross (), allows us to combine information from any two relations.We write the Cartesian product of relations r_1 and r_2 as $r_1 \quad r_2$.

Relational Algebra



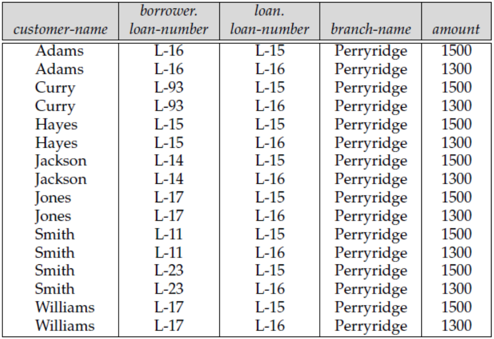
	borrower.	loan.		
customer-name	*loan-number*	*loan-number*	*branch-name*	*amount*
Adams	L-16	L-11	Round Hill	900
Adams	L-16	L-14	Downtown	1500
Adams	L-16	L-15	Perryridge	1500
Adams	L-16	L-16	Perryridge	1300
Adams	L-16	L-17	Downtown	1000
Adams	L-16	L-23	Redwood	2000
Adams	L-16	L-93	Mianus	500
Curry	L-93	L-11	Round Hill	900
Curry	L-93	L-14	Downtown	1500
Curry	L-93	L-15	Perryridge	1500
Curry	L-93	L-16	Perryridge	1300
Curry	L-93	L-17	Downtown	1000
Curry	L-93	L-23	Redwood	2000
Curry	L-93	L-93	Mianus	500
Hayes	L-15	L-11		900
Hayes	L-15	L-14		1500
Hayes	L-15	L-15		1500
Hayes	L-15	L-16		1300
Hayes	L-15	L-17		1000
Hayes	L-15	L-23		2000
Hayes	L-15	L-93		500
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
Smith	L-23	L-11	Round Hill	900
Smith	L-23	L-14	Downtown	1500
Smith	L-23	L-15	Perryridge	1500
Smith	L-23	L-16	Perryridge	1300
Smith	L-23	L-17	Downtown	1000
Smith	L-23	L-23	Redwood	2000
Smith	L-23	L-93	Mianus	500
Williams	L-17	L-11	Round Hill	900
Williams	L-17	L-14	Downtown	1500
Williams	L-17	L-15	Perryridge	1500
Williams	L-17	L-16	Perryridge	1300
Williams	L-17	L-17	Downtown	1000
Williams	L-17	L-23	Redwood	2000
Williams	L-17	L-93	Mianus	500

borrower x loan

Relational Algebra

Example: Find the names of all customers who have a loan at the Perryridge branch.

Solution: $\Pi_{customer-name}(\sigma_{(borrower.loan-no=loan.loan-no)\wedge}$ (branch-name=" Perryridge")(borrower $\times loan$)

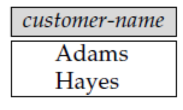


customer-name	borrower. loan-number	loan. loan-number	branch-name	amount
Adams	L-16	L-15	Perryridge	1500
Adams	L-16	L-16	Perryridge	1300
Curry	L-93	L-15	Perryridge	1500
Curry	L-93	L-16	Perryridge	1300
Hayes	L-15	L-15	Perryridge	1500
Hayes	L-15	L-16	Perryridge	1300
Jackson	L-14	L-15	Perryridge	1500
Jackson	L-14	L-16	Perryridge	1300
Jones	L-17	L-15	Perryridge	1500
Jones	L-17	L-16	Perryridge	1300
Smith	L-11	L-15	Perryridge	1500
Smith	L-11	L-16	Perryridge	1300
Smith	L-23	L-15	Perryridge	1500
Smith	L-23	L-16	Perryridge	1300
Williams	L-17	L-15	Perryridge	1500
Williams	L-17	L-16	Perryridge	1300

Result of $\sigma_{branch\text{-}name = \text{“Perryridge”}}$ $(borrower \times loan)$.

Relational Algebra

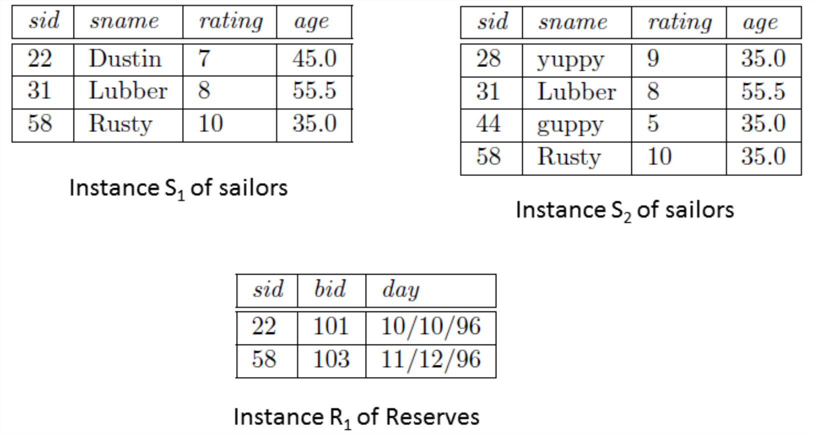
Final answer of the query is



customer-name
Adams
Hayes

Relational Algebra

Example: Consider the following tables S_1, S_2 and R_1:
Compute the following operations. (a) $S_1 \cup S_2$ (b) $S_1 \cap S_2$ (c) $S_1 - S_2$ (d) $S_1 \times S_2$



sid	*sname*	*rating*	*age*
22	Dustin	7	45.0
31	Lubber	8	55.5
58	Rusty	10	35.0

Instance S_1 of sailors

sid	*sname*	*rating*	*age*
28	yuppy	9	35.0
31	Lubber	8	55.5
44	guppy	5	35.0
58	Rusty	10	35.0

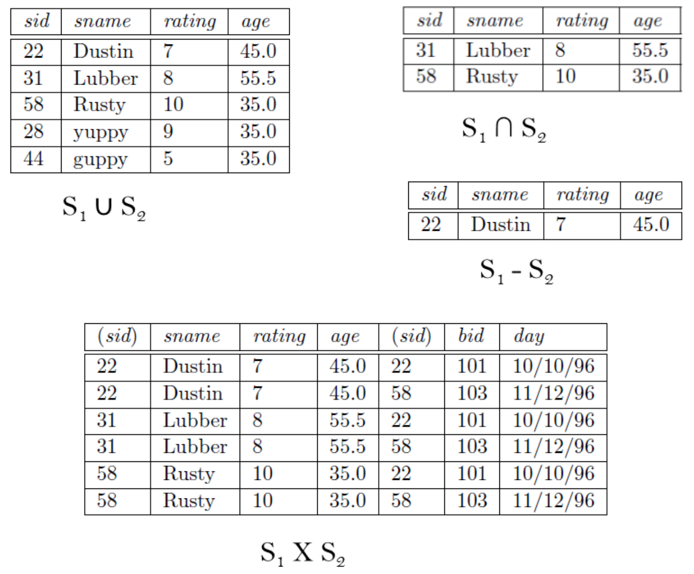
Instance S_2 of sailors

sid	*bid*	*day*
22	101	10/10/96
58	103	11/12/96

Instance R_1 of Reserves

Relational Algebra

Solution: Result of all the operations are shown as the following:-



sid	*sname*	*rating*	*age*
22	Dustin	7	45.0
31	Lubber	8	55.5
58	Rusty	10	35.0
28	yuppy	9	35.0
44	guppy	5	35.0

$S_1 \cup S_2$

sid	*sname*	*rating*	*age*
31	Lubber	8	55.5
58	Rusty	10	35.0

$S_1 \cap S_2$

sid	*sname*	*rating*	*age*
22	Dustin	7	45.0

$S_1 - S_2$

(*sid*)	*sname*	*rating*	*age*	(*sid*)	*bid*	*day*
22	Dustin	7	45.0	22	101	10/10/96
22	Dustin	7	45.0	58	103	11/12/96
31	Lubber	8	55.5	22	101	10/10/96
31	Lubber	8	55.5	58	103	11/12/96
58	Rusty	10	35.0	22	101	10/10/96
58	Rusty	10	35.0	58	103	11/12/96

$S_1 X S_2$

The Rename Operation

Unlike relations in the database, the results of relational-algebra expressions do not have a name that we can use to refer to them. It is useful to be able to give them names; the rename operator, denoted by the lowercase Greek letter rho (ρ).

Given a relational-algebra expression E, the expression

$$\rho_x(E)$$

returns the result of expression E under the name x.

A second form of the rename operation is as follows. Assume that a relational algebra expression E has arity n. Then, the expression

$$\rho_{x(A_1, A_2, ..., A_n)}(E)$$

returns the result of expression E under the name x, and with the attributes renamed to $A_1, A_2, ..., A_n$.

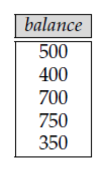
Relational Algebra

Example: Find the largest account balance in the bank.

Solution: Our strategy is to (1) compute first a temporary relation consisting of those balances that are not the largest and (2) take the set difference between the relation $\Pi_{balance}(account)$ and the temporary relation just computed, to obtain the result.

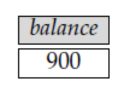
Relational Algebra

$$\Pi_{balance}(account) - \Pi_{account.balance}(\sigma_{account.balance<d.balance}(\text{account} \times \rho_d(account)))$$



balance
500
400
700
750
350

Figure 1: Result of $\Pi_{account.balance}(\sigma_{account.balance<d.balance}(account \times \rho_d(account)))$

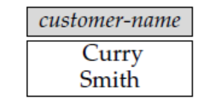


balance
900

Figure 2: Largest balance

Relational Algebra

Example: Find the names of all customers who live on the same street and in the same city as Smith.
Solution: In this query, first we find Smith's street and city. Second, we match Smith's street and city with other customer street and city. If match found then we select that customer.
$\Pi_{customer-name}(\sigma_{customer.customer-street=smith.street \wedge customer.customer-city=smith.city}$
$(\text{customer} \times \rho_{smith(street,city)}(\Pi_{customer-street,customer-city}$
$(\sigma_{customer-name=''Smith''}(customer)))))$



customer-name
Curry
Smith