Relational Algebra and SQL

Please see the associated file: RelationalAlgebraAndSQL.sql

Why Relational Algebra?

Adapted From Pritam Mitra
(https://www.doc.ic.ac.uk/~pjm/teaching/student_projects/pm105_report.pdf)

Often lecturers introduce Relational Algebra as conceptual operations which can be performed by actual query languages. Such presentations lead students to believe that Relational Algebra plays no important role in the actual implementation of database applications. Students may fail to realize that SQL queries are declarative and only tell the Database Management Systems (DBMS) what they want and not how to get it.
Relational Algebra on the other hand is procedural since it is concerned with operations on relations. It is very important that students understand Relational Algebra as it allows them to understand and compare database operations

Examples of Relational Algebra and SQL

Operand

Relational Algebra Operation	Relational Algebra Operator Symbol	Relational Algebra Example	SQL Example	Result
		R	SELECT * FROM R	C1 C2 1 B 2 A 3 B 4 A
		S	SELECT * FROM S	C1 C2 2 A 4 B 6 C
		T	SELECT * FROM T	C1 1 3

Operations

Relational Algebra Operation	Relational Algebra Operator Symbol	Relational Algebra Example	SQL Example	Result
Union	∪	R ∪ S	SELECT * FROM R **UNION** SELECT * FROM S	C1 C2 1 B 2 A 3 B 4 A 4 B 6 C
Intersect	∩	R ∩ S	SELECT * FROM R **INTERSECT** SELECT * FROM S	C1 C2 2 A
Difference	-	R - S	SELECT * FROM R **EXCEPT** SELECT * FROM S	C1 C2 1 B 3 B 4 A
Rename	ρ	$\rho_{K1/C1}(R)$	SELECT C1 **AS** K1, C2 FROM R	K1 C2 1 B 2 A 3 B 4 A
Select	σ_{φ}	$\sigma_{C2='B'}(R)$	SELECT * FROM R **WHERE C2 = 'B'**	C1 C2 1 B 3 B
Project	π	$\pi_{C1}(R)$	SELECT **C1** FROM R	C1 1 2 3 4

<table>
<tr><td>Cartesian Product</td><td>X</td><td>R X S</td><td>SELECT * FROM <u>R, S</u></td><td><table><tr><th>R.C1</th><th>R.C2</th><th>S.C1</th><th>S.C2</th></tr><tr><td>1</td><td>B</td><td>2</td><td>A</td></tr><tr><td>1</td><td>B</td><td>4</td><td>B</td></tr><tr><td>1</td><td>B</td><td>6</td><td>C</td></tr><tr><td>2</td><td>A</td><td>2</td><td>A</td></tr><tr><td>2</td><td>A</td><td>4</td><td>B</td></tr><tr><td>2</td><td>A</td><td>6</td><td>C</td></tr><tr><td>3</td><td>B</td><td>2</td><td>A</td></tr><tr><td>3</td><td>B</td><td>4</td><td>B</td></tr><tr><td>3</td><td>B</td><td>6</td><td>C</td></tr><tr><td>4</td><td>A</td><td>2</td><td>A</td></tr><tr><td>4</td><td>A</td><td>4</td><td>B</td></tr><tr><td>4</td><td>A</td><td>6</td><td>C</td></tr></table></td></tr>
<tr><td>Join (theta Join)</td><td>$\bowtie_{\varphi}$</td><td>R $\bowtie_{R.C1=S.C1}$ S</td><td>SELECT * FROM R
<u>JOIN</u> S <u>ON R.C1=S.C1</u></td><td><table><tr><th>R.C1</th><th>R.C2</th><th>S.C1</th><th>S.C2</th></tr><tr><td>2</td><td>A</td><td>2</td><td>A</td></tr><tr><td>4</td><td>A</td><td>4</td><td>B</td></tr></table></td></tr>
<tr><td>Natural Join</td><td>$\bowtie$</td><td>R $\bowtie$ S</td><td>SELECT
<u>R.C1 AS C1, R.C2 AS C2</u>
FROM R
<u>JOIN</u> S <u>ON R.C1=S.C1 AND R.C2=S.C2</u></td><td><table><tr><th>C1</th><th>C2</th></tr><tr><td>2</td><td>A</td></tr></table></td></tr>
<tr><td>Left Join</td><td>$⟕_{\varphi}$</td><td>R $⟕_{R.C1=S.C1}$ S</td><td>SELECT * FROM R
<u>LEFT JOIN</u> S <u>ON R.C1=S.C1</u></td><td><table><tr><th>R.C1</th><th>R.C2</th><th>S.C1</th><th>S.C2</th></tr><tr><td>2</td><td>A</td><td>2</td><td>A</td></tr><tr><td>4</td><td>A</td><td>4</td><td>B</td></tr><tr><td>1</td><td>B</td><td>NULL</td><td>NULL</td></tr><tr><td>3</td><td>B</td><td>NULL</td><td>NULL</td></tr></table></td></tr>
<tr><td>Right Join</td><td>$⟖_{\varphi}$</td><td>R $⟖_{R.C1=S.C1}$ S</td><td>SELECT * FROM R
<u>RIGHT JOIN</u> S <u>ON R.C1=S.C1</u></td><td><table><tr><th>R.C1</th><th>R.C2</th><th>S.C1</th><th>S.C2</th></tr><tr><td>2</td><td>A</td><td>2</td><td>A</td></tr><tr><td>4</td><td>A</td><td>4</td><td>B</td></tr><tr><td>NULL</td><td>NULL</td><td>6</td><td>C</td></tr></table></td></tr>
<tr><td>Full Join</td><td>$⟗_{\varphi}$</td><td>R $⟗_{R.C1=S.C1}$ S</td><td>SELECT * FROM R
<u>FULL JOIN</u> S <u>ON R.C1=S.C1</u></td><td><table><tr><th>R.C1</th><th>R.C2</th><th>S.C1</th><th>S.C2</th></tr><tr><td>2</td><td>A</td><td>2</td><td>A</td></tr><tr><td>4</td><td>A</td><td>4</td><td>B</td></tr><tr><td>1</td><td>B</td><td>NULL</td><td>NULL</td></tr><tr><td>3</td><td>B</td><td>NULL</td><td>NULL</td></tr><tr><td>NULL</td><td>NULL</td><td>6</td><td>C</td></tr></table></td></tr>
</table>

Left Semi Join*	$\ltimes_{\varphi}$	$R \ltimes_{R.C1=S.C1} S$	SELECT **R.C1, R.C2** FROM R **JOIN** S **ON R.C1=S.C1**	**R.C1** | **R.C2** **2** | **A** **4** | **A**
Right Semi Join*	$\rtimes_{\varphi}$	$R \ltimes_{R.C1=S.C1} S$	SELECT **S.C1, S.C2** FROM R **JOIN** S **ON R.C1=S.C1**	**S.C1** | **S.C2** 2 | **A** 4 | **B**
Anti-Join*	$\rhd_{\varphi}$	$R \rhd_{R.C1=S.C1} S$	SELECT * FROM R **EXCEPT** **SELECT R.C1, R.C2 FROM R JOIN S ON R.C1=S.C1**	**R.C1** | **R.C2** **1** | **B** **3** | **B**
Self-Join		$\rho_{R/S}(S) \bowtie_{2=S.C1} S$	SELECT * FROM **S AS R JOIN S ON 2 = S.C1**	**C1** | **C2** | **C1** | **C2** **2** | **A** | **2** | **A** **4** | **B** | **2** | **A** **6** | **C** | **2** | **A**
Division	÷	R ÷ T	SELECT **C2** FROM R **EXCEPT** SELECT **C2** FROM (SELECT * FROM **T, (SELECT C2 FROM R)** AS U **EXCEPT** SELECT * FROM R) AS U	**C2** **B**

Here, the definitions of Left Semi Join, Right Semi Join, and Anti Join are variations of the Theta-Join. Left Semi Join, Right Semi Join, and Anti Join are actually variations on the rarely-used natural join.

Some Links:
Relational Database derives from relational algebra: http://en.wikipedia.org/wiki/Relational_algebra, http://en.wikipedia.org/wiki/Relational_database
Self join: http://stackoverflow.com/questions/3362038/what-is-self-join-and-when-would-you-use-it
Joins: http://en.wikipedia.org/wiki/Join_(SQL)