6.1 Significance of Relational Operators

- Relational database supports basic database operations in order to provide useful means for retrieving or manipulating data in tables.
- Because the relational model has its mathematical basis upon the relational theory (by thinking tables as sets or relations), the supported database operators conform to existing operators in relational algebra.
- In fact, a relational database software implementation, called DBMS, is said to have higher degree of relational completeness depending upon the extent to which relational algebra operators are supported.
- In total there are eight operators are found in relational theory, namely SELECT, PROJECT, JOIN, INTERSECT, UNION, DIFFERENCE, PRODUCT and DIVIDE.
- Minimally speaking, a DBMS implementation is said to be relational if it supports at least the key relational operators, namely SELECT, PROJECT, and JOIN.
- Very few DBMSs are capable of supporting all eight relational operators.
- Use of relational algebra operators on existing tables (relations) results in outcomes look like new relations. This characteristic lets the user recursively applying the operators among the operator outcomes. Figure

6.2 Operator: UNION:

- Needs two tables as its operands
- Combines all rows from two tables, excluding duplicate rows.
- Tables, used as operands, must be UNION compatible with each other. (Tables must have the same attribute characteristics for all columns i.e. the columns and domains must be identical).
- Figure-6.1 demonstrates an example of UNION operation.
Popped up Questions

Choose the right answer for the UNION Operation shown in the figure.

6.3 Operator: INTERSECT:
- Needs two tables as its operands
- Yields only the rows that appear in both the tables
- Operand tables must be UNION compatible with each other
- Figure-6.2 demonstrates an example of INTERSECT operation.

![INTERSECT](image)

Figure 6.2: Demonstrating INTERSECT Operator

Popped up Questions

Choose the right answer for the INTERSECT Operation shown in the figure.

6.4 Operator: DIFFERENCE
- Needs two tables as its operands
- Yields all rows in one table not found in the other table—that is, it subtracts one table from the other.
- Requires the UNION compatibility of the operand tables.
- Figure-6.3 demonstrates an example of DIFFERENCE operation.

![DIFFERENCE](image)

Figure6.3: Demonstrating DIFFERENCE Operator

Popped up Questions

Choose the right answer for the DIFFERENCE Operation shown in the figure.

6.5 Operator: PRODUCT
- Needs two tables as its operands
- Yields all possible pairs of rows from the two tables.
The operand tables are not necessarily UNION compatible with each other.
The yielded result is also known as the Cartesian product.
Figure-6.4 demonstrates an example of PRODUCT operation.

**Figure6.4: Demonstrating PRODUCT Operator**

<table>
<thead>
<tr>
<th>P_CODE</th>
<th>P_DESCRIP</th>
<th>PRICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>123455</td>
<td>Flashlight</td>
<td>$5.26</td>
</tr>
<tr>
<td>123457</td>
<td>Lamp</td>
<td>$25.15</td>
</tr>
<tr>
<td>123458</td>
<td>Box Fan</td>
<td>$10.99</td>
</tr>
<tr>
<td>213345</td>
<td>9v battery</td>
<td>$1.92</td>
</tr>
<tr>
<td>254467</td>
<td>100W bulb</td>
<td>$1.47</td>
</tr>
<tr>
<td>311452</td>
<td>Powerdrill</td>
<td>$34.99</td>
</tr>
</tbody>
</table>

**PRODUCT**

<table>
<thead>
<tr>
<th>STORE</th>
<th>AISLE</th>
<th>SHELF</th>
</tr>
</thead>
<tbody>
<tr>
<td>w</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>K</td>
<td>9</td>
</tr>
<tr>
<td>25</td>
<td>Z</td>
<td>6</td>
</tr>
</tbody>
</table>

Popped up Questions

Choose the right answer for the SELECT Operation shown in the figure.

**6.6 Operator: SELECT**

- Needs a single table as its operand
- Yields values for all rows found in the table
- Can be used to list either all row values or it can yield only those row values that match a specified criterion
- Yields a horizontal subset of a table
- Figure-6.5 demonstrates an example of SELECT operation.
Popped up Questions

Choose the right answer for the SELECT Operation shown in the figure.

6.7 Operator: PROJECT

- Uses a single table as its operand
- Yields all values for selected attributes
- Yields a vertical subset of a table
- Figure-6.6 demonstrates an example of PROJECT operation.
Choose the right answer for the PROJECT Operation shown in the figure.

6.8 Operator: JOIN

- Allows us to combine information from two tables
- Uses two table having a common attribute as its operands
- Join operation is considered as the real power behind the relational database implementations (RDBMS)
- JOIN allows the use of independent tables, linked by common attributes, resulting in minimal redundancy possible.
- Possible types of JOIN includes: NATURAL JOIN, EQUIJOIN, THETA JOIN, LEFT OUTER JOIN, and RIGHT OUTER JOIN
- Figure6.7 shows the table examples used in the forth coming discussion of different JOIN operators:
6.8.1 JOIN Subtype: NATURAL JOIN

- The operands must have common attribute(s) in order to yield the join outcome. Such common attributes are called join column(s).
- Links two tables by selecting only rows with common values in their join column(s).
- Works in a three-step process:
  
  Step 1. PRODUCT of the tables is created.

Step 2. SELECT is performed on Step 1 output to yield only the rows for which the join column values are equal.
Step 3. PROJECT is performed on Step 2 results to yield a single copy of each attribute, thereby eliminating duplicate columns.

6.8.2 JOIN Subtype: EQUIJOIN
- Links tables on the basis of an equality condition that compares specified columns (join columns) of each table.
- Outcome does not eliminate duplicate columns. In other words, the result will show the join column more than once. Figure
- In equijoin, the condition used to join the tables is equal to (=), thus the name equijoin.
6.8.3 JOIN Subtype: THETA JOIN

- While joining the tables using join condition having any other comparison operator other than equal to (=), the join is called theta join.

### Popped up Questions
Choose the right answer for the EQUIJOIN/ THETAJOIN Operation shown in the figure.

6.8.4 JOIN Subtype: LEFT/ RIGHT OUTER JOIN
In normal join operations, the outcome shows only the rows of the tables having matching values in the join columns while the rows having unmatched values are discarded.

In case, if unmatched rows are required to be shown in the outcome (some columns in the result will be having NULL values), the resultant join is called an Outer Join.

While taking and outer join between two tables A and B, there could be two possible outcomes.

- If the result shows all the matching rows and the unmatched rows of A with the NULL values in the columns of B, the join is called **Left outer join** (allows the unmatched rows of left table to come). Figure 6.13 is demonstrating LEFT OUTER JOIN using tables of Figure 6.7.

- If the result shows all the matching rows and the unmatched rows of B with the NULL values in the columns of A, the join is called **Right outer join** (allows the unmatched rows of right table to come). Figure 6.14 is demonstrating RIGHT OUTER JOIN using tables of Figure 6.7.

### Popped up Questions

Choose the right answer for the LEFT/RIGHT OUTER JOIN Operation shown in the figure.

#### 6.9 Operator: DIVIDE

- DIVIDE requires the use of one single-column table and one two-column table
- DIVIDE requires the use of one single-column table and one two-column table
Consider an example of ‘TABLE1 DIVIDE TABLE2 yields TABLE2’
- TABLE1 & TABLE2 must be sharing one common column
- To be included in the result (TABLE3), a value in the nonshared column must be associated (in the dividing TABLE2) with every value in TABLE1.

**Figure 6.15: Demonstrating DIVIDE Operator**

![Diagram of DIVIDE operation]

**Popped up Questions**

Choose the right answer for the DIVIDE Operation shown in the figure.