**Relational Algebra**

Relational algebra is a procedural query language, which takes instances of relations as input and yields instances of relations as output. It uses operators to perform queries. An operator can be either unary or binary. They accept relations as their input and yield relations as their output. Relational algebra is performed recursively on a relation and intermediate results are also considered relations.

There are two types of relational operators:

1. Basic operators

* Projection
* Select
* Union
* Set Different
* Cross Product
* Rename

1. Derived operators

* Join

**Projection**

This operation shows the list of those attributes that we wish to appear in the result. Rest of the attributes are eliminated from the table. It is denoted by ∏

Notation: ∏ A1, A2, An (r)

**Where**

**A1, A2, A3**is used as an attribute name of relation **r**.

.

Student

|  |  |  |  |
| --- | --- | --- | --- |
| Roll | Name | Address | Age |
| 1 | Anjit | lamjung | 22 |
| 2 | Ayush | jhapa | 21 |
| 3 | Krishna | kathmandu | 23 |
| 4 | Ayush | pokhara | 20 |

Retrieve the roll number from table student.

∏ Roll (Student)

|  |
| --- |
| Roll |
| 1 |
| 2 |
| 3 |
| 4 |

Retrieve the name from table student.

∏ Name (Student)

|  |
| --- |
| Name |
| Anjit |
| Ayush |
| Krishna |

**Projection only works on columns and by default it retrieves distinct data from the entire database.**

**Select**

The select operation selects tuples that satisfy a given predicate. It is denoted by sigma (σ).

Notation:  σ p(r)

Where:

σ is used for selection prediction  
r is used for relation  
p is used as a propositional logic formula which may use connectors like: AND OR and NOT. These relational can use as relational operators like =, ≠, ≥, <, >, ≤.

|  |  |  |  |
| --- | --- | --- | --- |
| Roll | Name | Address | Age |
| 1 | Anjit | lamjung | 22 |
| 2 | Ayush | jhapa | 21 |
| 3 | Krishna | kathmandu | 23 |
| 4 | Ayush | pokhara | 20 |

Retrieve the name of the student whose roll number is 3.

∏ Name (σ Roll = ‘ 3 ‘ (Student))

|  |
| --- |
| Name |
| Krishna |

**Select works only on rows.**

**Cartesian product**

The Cartesian product is used to combine each row in one table with each row in the other table. It is also known as a cross product.It is denoted by X.

Notation: R1 X R2

R1

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **C** |
| **1** | **2** | **3** |
| **4** | **5** | **6** |
| **7** | **8** | **9** |

R2

|  |  |
| --- | --- |
| **C** | **D** |
| **3** | **4** |
| **5** | **9** |

R1 X R2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **C** | **D** |
| **1** | **2** | **3** | **3** | **4** |
| **1** | **2** | **3** | **5** | **9** |
| **4** | **5** | **6** | **3** | **4** |
| **4** | **5** | **6** | **5** | **9** |
| **7** | **8** | **9** | **3** | **4** |
| **7** | **8** | **9** | **5** | **9** |

**Rename**

The rename operation is used to rename the output relation. It is denoted by rho (ρ).

Example: We can use the rename operator to rename STUDENT relation to STUDENT1.

ρ (STUDENT1, STUDENT)

**Union**

Suppose there are two tuples R and S. The union operation contains all the tuples that are either in R or S or both in R & S.It eliminates the duplicate tuples. It is denoted by ∪.

Notation: R1 ∪ R2

A union operation must hold the following condition:

* R1 and R2 must have the attribute of the same number.
* Duplicate tuples are eliminated automatically.

Student

|  |  |
| --- | --- |
| Std - ID | Name |
| 1 | Anjit |
| 2 | Pabitra |
| 3 | Rabina |
| 4 | Saugat |

Employee

|  |  |
| --- | --- |
| Emp - ID | Name |
| 2 | Pabitra |
| 7 | Ayush |
| 8 | Sonam |

( Student ) U ( Employee )

|  |  |
| --- | --- |
| Std - ID | Name |
| 1 | Anjit |
| 2 | Pabitra |
| 3 | Rabina |
| 4 | Saugat |
| 7 | Ayush |
| 8 | Sonam |

Retrieve the name of the student who can be belongs to employee table also.

∏ NAME (Student) ∪ ∏ NAME (Employee)

|  |
| --- |
| Name |
| Anjit |
| Pabitra |
| Rabina |
| Saugat |
| Ayush |
| Sonam |

**Set Difference**

Suppose there are two tuples R and S. The set intersection operation contains all tuples that are in both R & S. It is denoted by intersection ∩.

Notation: R1 - R2

A set difference operation must hold the following condition:

* R1 and R2 must have the attribute of the same number.
* Duplicate tuples are eliminated automatically.

Student

|  |  |
| --- | --- |
| Std - ID | Name |
| 1 | Anjit |
| 2 | Pabitra |
| 3 | Rabina |
| 4 | Saugat |

Employee

|  |  |
| --- | --- |
| Emp - ID | Name |
| 2 | Pabitra |
| 7 | Ayush |
| 8 | Sonam |

( Student ) - ( Employee )

|  |  |
| --- | --- |
| Std - ID | Name |
| 1 | Anjit |
| 3 | Rabina |
| 4 | Saugat |

Retrieve the name of the student who is not employee.

∏ NAME (Student) - ∏ NAME (Employee)

|  |
| --- |
| Name |
| Anjit |
| Rabina |
| Saugat |