**Name: Mihaela McReynolds**

**Relational Algebra**

* The Relational Model consists of the elements: relations, which are made up of attributes.
* A relation is a set of attributes with values for each attribute such that:
  1. Each attribute value must be a single value only (atomic).
  2. All values for a given attribute must be of the same type (or domain).
  3. Each attribute name must be unique.
  4. The order of attributes is insignificant
  5. No two rows (tuples) in a relation can be identical.
  6. The order of the rows (tuples) is insignificant.
* Relational Algebra is a collection of operations on Relations.
* Relations are operands and the result of an operation is another relation.
* Two main collections of relational operators:
  1. Set theory operations:  
     Union, Intersection, Difference and Cartesian product.
  2. Specific Relational Operations:  
     Selection, Projection, Join, Division

## Set Theoretic Operations

Consider the following relations **R** and **S** (in Oracle’s RDBMS implementation, we think of these as tables)  
**R**

|  |  |  |
| --- | --- | --- |
| **First** | **Last** | **Age** |
| Bill | Smith | 22 |
| Sally | Green | 28 |
| Mary | Keen | 23 |
| Tony | Jones | 32 |

**S**

|  |  |  |
| --- | --- | --- |
| **First** | **Last** | **Age** |
| Forrest | Gump | 36 |
| Sally | Green | 28 |
| DonJuan | DeMarco | 27 |

* Union: R http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/union.gif S   
  Result: Relation with tuples from R and S with duplicates removed.
* Difference: R - S   
  Result: Relation with tuples from R excluding those that are also in S
* Intersection: R http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/intersect.gif S   
  Result: Relation with tuples that appear in both R and S.

**R http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/union.gif S**

|  |  |  |
| --- | --- | --- |
| **First** | **Last** | **Age** |
| Bill | Smith | 22 |
| Sally | Green | 28 |
| Mary | Keen | 23 |
| Tony | Jones | 32 |
| Forrest | Gump | 36 |
| DonJuan | DeMarco | 27 |

**R - S**

|  |  |  |
| --- | --- | --- |
| **First** | **Last** | **Age** |
| Bill | Smith | 22 |
| Mary | Keen | 23 |
| Tony | Jones | 32 |

**R http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/intersect.gif S**

|  |  |  |
| --- | --- | --- |
| **First** | **Last** | **Age** |
| Sally | Green | 28 |

## Union Compatible Relations

* Attributes of relations need not be identical to perform union, intersection and difference operations.
* However, they must have the same number of attributes and the *domains* for corresponding attributes must be identical.
* **Domain** is the datatype and size of an attribute.
* The **degree** of relation R is the number of attributes it contains.
* Definition: Two relations R and S are *union compatible* if and only if they have the same degree and the domains of the corresponding attributes are the same.
* Some additional properties:
  + Union, Intersection and difference operators may only be applied to Union Compatible relations.
  + Union and Intersection are commutative operations  
    R http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/union.gif S = S http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/union.gif R  
    R http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/intersect.gif S = S http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/intersect.gif R
  + Difference operation is NOT commutative.  
    R - S not equal to S - R
  + The resulting relations may not have meaningful names for the attributes. Convention is to use the attribute names from the first relation.

### Exercise A

* Given relation **T**

|  |  |  |
| --- | --- | --- |
| **fName** | **lName** | **Score** |
| William | Smith | 44 |
| Sally | Green | 28 |
| Mary | Kontrary | 27 |

1. Compute R http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/union.gif T:

|  |  |  |
| --- | --- | --- |
| **First** | **Last** | **Age** |
| Bill | Smith | 22 |
| Sally | Green | 28 |
| Mary | Keen | 23 |
| Tony | Jones | 32 |
| William | Smith | 44 |
| Mary | Kontrary | 27 |

1. Compute R http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/intersect.gif T:

|  |  |  |
| --- | --- | --- |
| First | Last | Age |
| Sally | Green | 28 |

1. Show that R - T is not equal to T - R

R – T:

|  |  |  |
| --- | --- | --- |
| **First** | **Last** | **Age** |
| Bill | Smith | 22 |
| Mary | Keen | 23 |
| Tony | Jones | 32 |

T – R

|  |  |  |
| --- | --- | --- |
| **fName** | **lName** | **Score** |
| William | Smith | 44 |
| Mary | Kontrary | 27 |

## Cartesian Product

* Produce all combinations of tuples from two relations.

**R**

|  |  |  |
| --- | --- | --- |
| **First** | **Last** | **Age** |
| Bill | Smith | 22 |
| Mary | Keen | 23 |
| Tony | Jones | 32 |

**S**

|  |  |
| --- | --- |
| **Dinner** | **Dessert** |
| Steak | Ice Cream |
| Lobster | Cheesecake |

**R X S**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **First** | **Last** | **Age** | **Dinner** | **Dessert** |
| Bill | Smith | 22 | Steak | Ice Cream |
| Bill | Smith | 22 | Lobster | Cheesecake |
| Mary | Keen | 23 | Steak | Ice Cream |
| Mary | Keen | 23 | Lobster | Cheesecake |
| Tony | Jones | 32 | Steak | Ice Cream |
| Tony | Jones | 32 | Lobster | Cheesecake |

### Selection Operator

* Selection and Projection are *unary* operators.
* The selection operator is sigma: http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/sigma.gif
* The selection operation acts like a *filter* on a relation by returning only a certain number of tuples. We think of tuples as rows in our table.
* The resulting relation will have the same degree as the original relation (i.e. same # of attributes)
* The resulting relation may have fewer tuples than the original relation.
* The tuples to be returned are dependent on a *condition* that is part of the selection operator.
* http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/sigma.gifC (R) Returns only those tuples in R that satisfy condition **C**
* A condition C can be made up of any combination of comparison or logical operators that operate on the attributes of R.
  + Comparison operators: http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/comparison_ops.gif
  + Logical operators: http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/logical_ops.gif (these symbols mean AND OR NOT respectively)

### Selection Examples

Assume the following relation EMP has the following *tuples*:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Office** | **Dept** | **Rank** |
| Smith | 400 | CS | Assistant |
| Jones | 220 | Econ | Adjunct |
| Green | 160 | Econ | Assistant |
| Brown | 420 | CS | Associate |
| Smith | 500 | Fin | Associate |

* Select only those Employees in the CS department:  
  http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/sigma.gifDept = 'CS' (EMP)  
  Result:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Office** | **Dept** | **Rank** |
| Smith | 400 | CS | Assistant |
| Brown | 420 | CS | Associate |

* Select only those Employees with last name Smith who are assistant professors:  
  http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/sigma.gifName = 'Smith' http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/and.gifRank = 'Assistant' (EMP)  
  Result:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Office** | **Dept** | **Rank** |
| Smith | 400 | CS | Assistant |

* Select only those Employees who are either Assistant Professors or in the Economics department:  
  http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/sigma.gifRank = 'Assistant' http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/or.gifDept = 'Econ' (EMP)  
  Result:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Office** | **Dept** | **Rank** |
| Smith | 400 | CS | Assistant |
| Jones | 220 | Econ | Adjunct |
| Green | 160 | Econ | Assistant |

* Select only those Employees who are not in the CS department or Adjuncts:  
  http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/sigma.gifhttp://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/not.gif(Rank = 'Adjunct' http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/or.gifDept = 'CS') (EMP)  
  Result:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Office** | **Dept** | **Rank** |
| Green | 160 | Econ | Assistant |
| Smith | 500 | Fin | Associate |

### Exercise B

* Evaluate the following expressions:
  1. http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/sigma.gifhttp://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/not.gif(Rank = 'Adjunct' http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/and.gifDept = 'CS') (EMP)

Select only those Employees in the CS department who are not Adjuncts

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Office** | **Dept** | **Rank** |
| Smith | 400 | CS | Assistant |
| Brown | 420 | CS | Associate |

* 1. http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/sigma.gifRank = 'Associate' ( http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/sigma.gifDept = 'CS' EMP )

Select only the Associates out of the Employees in the CS department

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Office** | **Dept** | **Rank** |
| Brown | 420 | CS | Associate |

* 1. http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/sigma.gifDept = 'CS' ( http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/sigma.gifRank = 'Associate' EMP )

Select only those from the CS department out of the Employees who are Associates

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Office** | **Dept** | **Rank** |
| Brown | 420 | CS | Associate |

* 1. http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/sigma.gifRank = 'Associate' http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/and.gifDept = 'CS' (EMP)

Select only the Employees who are Associates in the CS department

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Office** | **Dept** | **Rank** |
| Brown | 420 | CS | Associate |

* 1. http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/sigma.gifAge > 26 (R http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/union.gif S)

For this expression, use R and S from the Set Theoretic Operations section above.

Select only those over 26 in age from R union S (tuples from R and S with duplicates removed). The result set includes:

|  |  |  |
| --- | --- | --- |
| **First** | **Last** | **Age** |
| Sally | Green | 28 |
| Tony | Jones | 32 |
| Forrest | Gump | 36 |
| DonJuan | DeMarco | 27 |

Do expressions 2, 3 and 4 above all evaluate to the same thing?

Yes, they do.

### Projection Operator

* Projection is also a *Unary* operator.
* The Projection operator is pi: http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/pi.gif
* Projection limits the *attributes* that will be returned from the original relation.
* The general syntax is: http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/pi.gifattribute-list R  
  Where *attributes* is the list of attributes to be displayed and R is the relation.
* The resulting relation will have the same number of *tuples* (a.k.a. rows) as the original relation (unless there are duplicate tuples produced).
* The *degree* of the resulting relation may be equal to or less than that of the original relation.

### 

### Projection Examples

Assume the same EMP relation above is used.

* Project only the names and departments of the employees:  
  http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/pi.gifname, dept (EMP)  
  Results:

|  |  |
| --- | --- |
| **Name** | **Dept** |
| Smith | CS |
| Jones | Econ |
| Green | Econ |
| Brown | CS |
| Smith | Fin |

### Combining Selection and Projection

* The selection and projection operators can be combined to perform both operations.
* Show the names of all employees working in the CS department:  
  http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/pi.gifname ( http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/sigma.gifDept = 'CS' (EMP) )  
  Results:

|  |
| --- |
| **Name** |
| Smith |
| Brown |

* Show the name and rank of those Employees who are not in the CS department or Adjuncts:  
  http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/pi.gifname, rank ( http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/sigma.gifhttp://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/not.gif(Rank = 'Adjunct' http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/or.gifDept = 'CS') (EMP) )  
    
  Result:

|  |  |
| --- | --- |
| **Name** | **Rank** |
| Green | Assistant |
| Smith | Associate |

### Exercise C

* Evaluate the following expressions:
  1. http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/pi.gifname, rank ( http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/sigma.gifhttp://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/not.gif(Rank = 'Adjunct' http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/and.gifDept = 'CS') (EMP) )

Show the name and rank of the employees who are not Adjuncts in the CS department

|  |  |
| --- | --- |
| **Name** | **Rank** |

* 1. http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/pi.giffname, age (http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/sigma.gif Age > 22 (R http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/union.gif S) )

For this expression, use R and S from the Set Theoretic Operations section above.

Show the first name and age of those over age 22 from R union S (all rows from R and S, with duplicates removed)

|  |  |
| --- | --- |
| **First** | **Age** |
| Sally | 28 |
| Mary | 23 |
| Tony | 32 |
| Forrest | 36 |
| DonJuan | 27 |

* 1. http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/sigma.gifoffice > 300 ( http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/pi.gifname, rank (EMP))

Select those employees in an office number higher than 300 and show their name and rank

|  |  |  |
| --- | --- | --- |
| **Name** | **Dept** | **Rank** |
| Smith | CS | Assistant |
| Brown | CS | Associate |
| Smith | Fin | Associate |

### Aggregate Functions

* We can also apply *Aggregate functions* to attributes and tuples:
  + SUM
  + MINIMUM
  + MAXIMUM
  + AVERAGE, MEAN, MEDIAN
  + COUNT
* Aggregate functions are sometimes written using the *Projection* operator or the *Script F* character: http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/scriptf.gif

Aggregate Function Examples

Assume the relation EMP has the following tuples:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Office** | **Dept** | **Salary** |
| Smith | 400 | CS | 45000 |
| Jones | 220 | Econ | 35000 |
| Green | 160 | Econ | 50000 |
| Brown | 420 | CS | 65000 |
| Smith | 500 | Fin | 60000 |

* Find the minimum Salary: http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/scriptf.gifMIN (salary) (EMP)  
  Results:

|  |
| --- |
| **MIN(salary)** |
| 35000 |

* Find the average Salary: http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/scriptf.gifAVG (salary) (EMP)  
  Results:

|  |
| --- |
| **AVG(salary)** |
| 51000 |

* Count the number of employees in the CS department: http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/scriptf.gifCOUNT (name) ( http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/sigma.gifDept = 'CS' (EMP) )  
  Results:

|  |
| --- |
| **COUNT(name)** |
| 2 |

* Find the total payroll for the Economics department: http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/scriptf.gifSUM (salary) ( http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/sigma.gifDept = 'Econ' (EMP) )  
  Results:

|  |
| --- |
| **SUM(salary)** |
| 85000 |

## Join Operation

* Join operations bring together two relations and combine their attributes and tuples in a specific fashion.
* The generic join operator (called the *Theta Join* is: http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/joinop_big.gif
* It takes as arguments the attributes from the two relations that are to be joined.
* For example assume we have the EMP relation as above and a separate DEPART relation with (Dept, MainOffice, Phone) :  
  EMP http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/joinop.gifEMP.Dept = DEPART.Dept DEPART
* The join condition can be http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/comparison_ops.gif
* When the join condition operator is   **=**   then we call this an *Equijoin*
* Note that the attributes in common are repeated.

### Join Examples

Assume we have the EMP relation from above and the following DEPART relation:

|  |  |  |
| --- | --- | --- |
| **Dept** | **MainOffice** | **Phone** |
| CS | 404 | 555-1212 |
| Econ | 200 | 555-1234 |
| Fin | 501 | 555-4321 |
| Hist | 100 | 555-9876 |

* Find all information on every employee including their department info:  
  EMP http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/joinop.gifemp.Dept = depart.Dept DEPART
* Results:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Office** | **EMP.Dept** | **Salary** | **DEPART.Dept** | **MainOffice** | **Phone** |
| Smith | 400 | CS | 45000 | CS | 404 | 555-1212 |
| Jones | 220 | Econ | 35000 | Econ | 200 | 555-1234 |
| Green | 160 | Econ | 50000 | Econ | 200 | 555-1234 |
| Brown | 420 | CS | 65000 | CS | 404 | 555-1212 |
| Smith | 500 | Fin | 60000 | Fin | 501 | 555-4321 |

* Find all information on every employee including their department info where the employee works in an office numbered less than the department main office:  
  EMP http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/joinop.gif(emp.office < depart.mainoffice) http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/and.gif(emp.dept = depart.dept) DEPART  
  Results:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Office** | **EMP.Dept** | **Salary** | **DEPART.Dept** | **MainOffice** | **Phone** |
| Smith | 400 | CS | 45000 | CS | 404 | 555-1212 |
| Green | 160 | Econ | 50000 | Econ | 200 | 555-1234 |
| Smith | 500 | Fin | 60000 | Fin | 501 | 555-4321 |

### Natural Join

* Notice in the generic (Theta) join operation, any attributes in common (such as dept above) are repeated.
* The *Natural Join* operation removes these duplicate attributes.
* The natural join operator is: **\***
* We can also assume using **\*** that the join condition will be **=** on the two attributes in common.
* Example: EMP **\*** DEPART  
  Results:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Office** | **Dept** | **Salary** | **MainOffice** | **Phone** |
| Smith | 400 | CS | 45000 | 404 | 555-1212 |
| Jones | 220 | Econ | 35000 | 200 | 555-1234 |
| Green | 160 | Econ | 50000 | 200 | 555-1234 |
| Brown | 420 | CS | 65000 | 404 | 555-1212 |
| Smith | 500 | Fin | 60000 | 501 | 555-4321 |

### 

### Outer Join

* In the Join operations so far, only those tuples from both relations that satisfy the join condition are included in the output relation.
* The *Outer join* includes other tuples as well according to a few rules.
* Three types of outer joins:
  1. Left Outer Join http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/loj.gifincludes all tuples in the left hand relation and includes only those matching tuples from the right hand relation.
  2. Right Outer Join http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/roj.gifincludes all tuples in the right hand relation and includes ony those matching tuples from the left hand relation.
  3. Full Outer Join http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/foj.gifincludes all tuples in the left hand relation and from the right hand relation.
* Examples:

Assume we have two relations: PEOPLE and MENU:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PEOPLE:   |  |  |  | | --- | --- | --- | | **Name** | **Age** | **Food** | | Alice | 21 | Hamburger | | Bill | 24 | Pizza | | Carl | 23 | Beer | | Dina | 19 | Shrimp | | MENU:   |  |  | | --- | --- | | **Food** | **Day** | | Pizza | Monday | | Hamburger | Tuesday | | Chicken | Wednesday | | Pasta | Thursday | | Tacos | Friday | |

* PEOPLE http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/loj.gifpeople.food = menu.food MENU

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Age** | **people.Food** | **menu.Food** | **Day** |
| Alice | 21 | Hamburger | Hamburger | Tuesday |
| Bill | 24 | Pizza | Pizza | Monday |
| Carl | 23 | Beer | *NULL* | *NULL* |
| Dina | 19 | Shrimp | *NULL* | *NULL* |

* PEOPLE http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/roj.gifpeople.food = menu.food MENU

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Age** | **people.Food** | **menu.Food** | **Day** |
| Bill | 24 | Pizza | Pizza | Monday |
| Alice | 21 | Hamburger | Hamburger | Tuesday |
| *NULL* | *NULL* | *NULL* | Chicken | Wednesday |
| *NULL* | *NULL* | *NULL* | Pasta | Thursday |
| *NULL* | *NULL* | *NULL* | Tacos | Friday |

* PEOPLE http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/foj.gifpeople.food = menu.food MENU

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Age** | **people.Food** | **menu.Food** | **Day** |
| Alice | 21 | Hamburger | Hamburger | Tuesday |
| Bill | 24 | Pizza | Pizza | Monday |
| Carl | 23 | Beer | *NULL* | *NULL* |
| Dina | 19 | Shrimp | *NULL* | *NULL* |
| *NULL* | *NULL* | *NULL* | Chicken | Wednesday |
| *NULL* | *NULL* | *NULL* | Pasta | Thursday |
| *NULL* | *NULL* | *NULL* | Tacos | Friday |