**Relational Algebra**

The relational model of the data and relational algebra were introduced by Codd (1970).

Concepts:

Domain - a set of atomic values for an attribute Relation schema - a set of attributes

Relation (relation instance) -a set of tuples Tuple Attribute

Degree - number of attributes in a relation schema

Cardinality - a number of tuples (at specific point of time) in a relation (relation instance)

Note: the absence of data is represented as special **NULL** value (means no data)

Key attributes of the relation

A relation is defined as a set of tuples; therefore, all tuples in a relation must be distinct.

Candidate key - potential unique identifiers for the tuples

Primary key - one of the candidate keys designated as a the primary key

Relational Database Schema

A relational database schema *is a set of relation schemas and a set of integrity constraints*.

Integrity constrains:

* Key constraints (specify candidate keys for each relation)
* Entity integrity constraint - no primary key value can be NULL
* Referential integrity constraint (associations between relations expressed by foreign keys). Informally,

the referential constraint states that a foreign key in one relation must refer to an existing primary key.

Update operations on relations

* Insert a tuple or tuples
* Delete a tuple or tuples
* Modify a value or values of attributes

**Relational Algebra**

Relational algebra is a collection of operations that are used to manipulate entire relations. The result of each operation is also a relation.

Relational operations are divided into set-theory operations and relational operations developed for database model.

Set-theory operations: UNION

INTERSECTION

DIFFERENCE

CARTESIAN PRODUCT

Relational Algebra operations PROJECTION (unary operation)

SELECTION (unary operation)

JOIN (THETA JOIN, EQUI JOIN, NATURAL JOIN)

Relational Algebra Operations

|  |  |  |
| --- | --- | --- |
| **Operation** | **Purpose** | **Notation** |
| SELECTION | Select all tuples that satisfy the selection condition from a relation R | <selection condition> (R) |
| PROJECTION | Produce a result with specified list of attributes from relation R | <attribute list> ( R) |
| THETA JOIN (JOIN) | Produce all combinations of tuples from R and S that satisfy join condition | R join <join condition> S |
| EQUI JOIN | Join condition is equality comparison | R join <equi join condition> S |
| NATURAL JOIN | Same as equi join except that the join attribute from S is not included | R natural join S |
| UNION | All the tuples in R and S.  Relations must be UNION COMPATIBLE | R ∪ S |
| INTERSECTION | Tuples that are common to R and S. Relations must be UNION compatible. | R ∩ S |
| DIFFERENCE | Tuples that belong to R , that are not in S  Relations must be UNION COMPATIBLE | R - S |
| CARTESIAN PRODUCT | A relation that has all the attributes of R and S. The tuples are the all possible combinations of tuples in R and S | R x S |

**Relational Algebra Exercise**

Given two relations (tables) **R**  and **S:**

**R S**

|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **C** | **D** |
| a1 | b1 | c2 | d1 |
| a3 | b1 | c1 | d2 |
| a2 | b2 | c4 | d5 |

|  |  |
| --- | --- |
| **E** | **A** |
| e1 | a3 |
| e3 | a2 |

Answer the following questions:

1. What is the result of a Selection **σ**  over **R** :**σ** B <> b2 (**R**) =

**B <> b**2 means that *the value in column B is not equal to b2*

1. What is the result of a Selection over R: **σ** B <> b2 and B<> b1 (**R**) =
2. What is the result of a Projection over **R**: **π** B (**R**) =
3. Is Union **R ∪ S** a valid operation? Explain why.
4. What is the **degree** (number of columns) of the result **R x S** (**x** – Cartesian Product)?\_\_\_ What is the **cardinality** of the result?\_\_\_\_
5. Calculate natural join between R and S. What is the degree of the resulting relation?
6. Calculate Left Outer Join between R and S.
7. What is the result of **R ∩ S** ?