Database and its Applications

Data Models and Mathematical Foundations

Pooja T S

Computer Applications

Database and its Applications

Data Models and Mathematical Foundations

Relational Algebra: Difference, Joins, Product

Pooja T S

Computer Applications

Relational Algebra – Difference ()

- The Difference operation returns tuples present in one relation but not in the other.
- Notation: $R - S$

  Requirement: R and S must be union-compatible.
- Example (Cartoon Dataset):
  - DisneyCharacters = {Mickey, Goofy}
  - LooneyCharacters = {Bugs, Daffy}
  - Query: Characters in Disney but not in Looney.
  - Expression: DisneyCharacters LooneyCharacters
  - Result: {Mickey, Goofy}
- SQL Equivalent: `SELECT * FROM DisneyCharacters EXCEPT SELECT * FROM LooneyCharacters;`

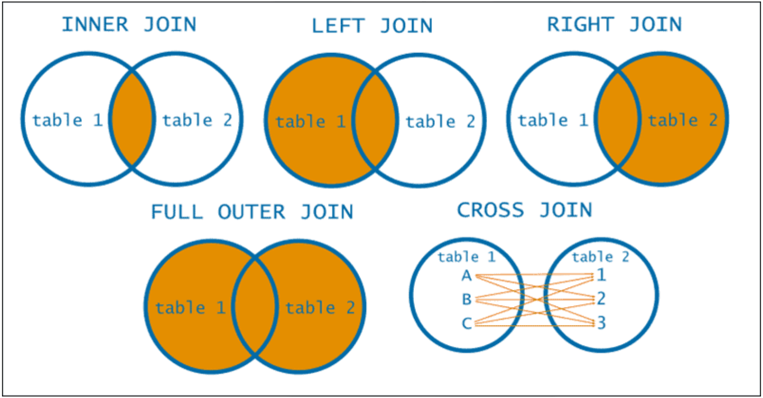
Relational Algebra – Cartesian Product ($\times$)

- The Cartesian Product pairs each tuple of one relation with every tuple of another.
- Notation: $R \times S$
- Basis for JOIN operations.
- Example (Cartoon Dataset):
  - Characters = {Mickey, Bugs}
  - Shows = {Clubhouse, Looney Tunes}
  - Result: {(Mickey, Clubhouse), (Mickey, Looney Tunes), (Bugs, Clubhouse), (Bugs, Looney Tunes)}
- SQL Equivalent: `SELECT * FROM Characters CROSS JOIN Shows;`

Relational Algebra Joins

- Joins combine tuples from two relations based on a related attribute.
- Relational Algebra Notation:
  - $\bowtie$ denotes join.
  - σ (selection) + $\times$ (Cartesian product) can express joins.
- Example Dataset:
  - CartoonCharacter(`name`, `show`)
  - ShowInfo(`show`, `year`)

Joins


INNER JOIN
table 1
table 2
LEFT JOIN
table 1
table 2
RIGHT JOIN
table 1
table 2
FULL OUTER JOIN
table 1
table 2
CROSS JOIN
table 1
A
B
C
table 2
1
2
3

Inner Join

- Returns tuples that match on a given condition.
- Relational Algebra:

$$R \bowtie_{R.show=S.show} S$$

- SQL:

```
SELECT C.name, S.year
FROM CartoonCharacter C
INNER JOIN ShowInfo S
ON C.show = S.show;
```

- Example Output:
  - (Tom, 1940), (Jerry, 1940), (SpongeBob, 1999)
- Use Case: Retrieve only characters with known show details.

Left Outer Join

- Returns all tuples from left relation, matched tuples from right; unmatched → NULL.
- Relational Algebra:

$$R \quad ⟕ \quad S$$

- SQL:

```
SELECT C.name, S.year
FROM CartoonCharacter C
LEFT JOIN ShowInfo S
ON C.show = S.show;
```

- Example Output:
  - (Tom, 1940), (Jerry, 1940), (SpongeBob, 1999), (Doraemon, NULL)
- Use Case: Keep all characters, even if show info is missing.

Right Outer Join

- Returns all tuples from right relation, matched tuples from left; unmatched → NULL.
- Relational Algebra:

$$R \;⟖\; S$$

- SQL:

```
SELECT C.name, S.year
FROM CartoonCharacter C
RIGHT JOIN ShowInfo S
ON C.show = S.show;
```

- Example Output:
  - (Tom, 1940), (Jerry, 1940), (NULL, 2001)
- Use Case: Keep all shows, even if no characters are linked.

Full Outer Join

- Returns all tuples from both relations. Matches combined, unmatched → NULL.
- Relational Algebra:

$$R \;⟗\; S$$

- SQL:

```
SELECT C.name, S.year
FROM CartoonCharacter C
FULL OUTER JOIN ShowInfo S
ON C.show = S.show;
```

- Example Output:
  - (Tom, 1940), (Jerry, 1940), (SpongeBob, 1999), (Doraemon, NULL), (NULL, 2001)
- Use Case: Build a complete dataset from both sides.

Join Types – Summary

- Summary Table:
  - **Inner Join:** $R \bowtie S$ — Only common tuples.
  - **Left Outer Join:** $R ⟕ S$ — All left + matched right.
  - **Right Outer Join:** $R ⟖ S$ — All right + matched left.
  - **Full Outer Join:** $R ⟗ S$ — All tuples from both.

Thank You

Pooja T S

Assistant Professor

Department of Computer Applications

poojats@pes.edu

080-26721983 Extn: 233