Union Operation

- The union operation allows us to combine two relations
- Notation: $r \cup s$

- For $r \cup s$ to be valid.
  1. *r, s* must have the *same* **arity** (same number of attributes)
  2. The attribute domains must be **compatible** (example: 2nd column of *r* deals with the same type of values as does the 2nd column of *s*)

Union Operation

- Example: to find all courses taught in the Fall 2017 semester, or in the Spring 2018 semester, or in both

$$\Pi_{course\_id}\left(\sigma_{semester=\text{“Fall”} \wedge year=2017}(section)\right) \cup$$

$$\Pi_{course\_id}\left(\sigma_{semester=\text{“Spring”} \wedge year=2018}(section)\right)$$

Union Operation (Cont.)

- Result of:

$$\Pi_{course\_id}(\sigma_{semester=\text{“Fall”} \wedge year=2017}(section)) \cup \Pi_{course\_id}(\sigma_{semester=\text{“Spring”} \wedge year=2018}(section))$$

course_id
CS-101
CS-315
CS-319
CS-347
FIN-201
HIS-351
MU-199
PHY-101

Set-Intersection Operation

- The set-intersection operation allows us to find tuples that are in both the input relations.
- Notation: $r \cap s$
- Assume:
  - *r*, *s* have the *same arity*
  - attributes of *r* and *s* are compatible

Set-Intersection Operation

- Example: Find the set of all courses taught in both the Fall 2017 and the Spring 2018 semesters.

$$\prod_{course\_id} (\sigma_{semester=\text{“Fall”} \wedge year=2017} (section)) \cap \prod_{course\_id} (\sigma_{semester=\text{“Spring”} \wedge year=2018} (section))$$

  - Result

course_id
CS-101

Set Difference Operation

- The set-difference operation allows us to find tuples that are in one relation but are not in another.

- Notation $r - s$

- Set differences must be taken between **compatible** relations.
  - r and s must have the same arity
  - attribute domains of r and s must be compatible

Set Difference Operation

- Example: to find all courses taught in the Fall 2017 semester, but not in the Spring 2018 semester

- $\Pi_{course\_id}(\sigma_{semester=\text{“Fall”} \wedge year=2017}(section)) - \Pi_{course\_id}(\sigma_{semester=\text{“Spring”} \wedge year=2018}(section))$

course_id
CS-347
PHY-101

The Rename Operation

- The results of relational-algebra expressions do not have a name that we can use to refer to them. The rename operator, ρ, is provided for that purpose
- The expression:

$$\rho_x (E)$$

returns the result of expression *E* under the name *x*

- Another form of the rename operation:

$$\rho_{x(A1,A2, .. An)} (E)$$

The Assignment Operation

- It is convenient at times to write a relational-algebra expression by assigning parts of it to temporary relation variables.
- The assignment operation is denoted by $\leftarrow$ and works like assignment in a programming language.
- Example: Find all instructor in the "Physics" and Music department.

$$Physics \leftarrow \sigma_{dept\_name= \text{“}Physics\text{”}}(instructor)$$
$$Music \leftarrow \sigma_{dept\_name= \text{“}Music\text{”}}(instructor)$$
$$Physics \cup Music$$

- With the assignment operation, a query can be written as a sequential program consisting of a series of assignments followed by an expression whose value is displayed as the result of the query.

Operations on relations

- Selection : σ (unary)
  - Select tuples that meet given condition
- Projection : Π (unary)
  - Select only given columns and remove duplicate rows
- Product : x (binary)
  - Combine every tuple of relation R with those of relation S
  - Size of result ?

Algebra ...

- Compatible tables : same schema
- Union : $\cup$
  - Merge two tables as sets, remove duplicates
- Intersection : $\cap$
  - Gives tuples present in both
- Difference : –
  - R – S : Gives Tuples of R not present in S

Joining two relations – Natural Join

- Combine relations on ‘common’ columns
- Join student with dept on dno
  - Gives dept data with student data
- Keep data from both tables without repeating columns
- Matching rows combined
- Un-matching rows dropped
- Denoted by

Algebra query

- Expressions using algebra operations
- Possible since each operation produces result as a relation

Π (σ (student ⋈ dept))

- Nesting possible

- Write query for

List departments having at least one student in hostel 8 and budget < 5 lakhs

Next : Introduction to SQL

- a large language with many features
- many modules