Database Systems

Lecture

Contents

- Introduction
- Selection
- Projection
- Set operations (Union, intersection, difference)
- Cartesian Product (also called Cross product)
- Join
    - Theta Join
    - Equi Join
    - Natural Join
    - Outer join
    - Left Outer join
    - Right Outer join
    - Full outer join

Relational Algebra

- So we have lots of data stored in relations, but what do we do with this data?
  - We use this data to get information
    - But first, we need to retrieve data
  - Relational algebra is a set of operations for retrieving data
  - Relational operations consist of relations as operands along with a set of operators
  - Every relational operator takes as input one or more relations and produces a relation as output
- A sequence of relational algebra operators is called a relational algebra expression

Selection

- Selects tuples (rows) from a relation which match the given criteria
    - Criteria can be applicable on one or more attributes
- The resulting relation has the same attributes as the input relation
- Represented as σ<selection condition>(R), where
    - R is the input relation
    - Selection condition determines which tuples to retrieve
- Note that R can be one relation, or the result of another relational operation…

Selection

- Select all those employees who live in Street 1
  - $\sigma_{\text{Address=Street 1}}$ (EMPLOYEE)
- Result:

$\sigma_{\text{Address=Street 1}}$ (EMPLOYEE)

EmpID	DoJ	Name	Address	Dept
E_112	12-2-2020	John	Street 1	D_2
E_144	8-3-2020	Mark	Street 1	D_4

EMPLOYEE

EmpID	DoJ	Name	Address	Dept
E_112	12-2-2020	John	Street 1	D_2
E_134	8-3-2020	Andy	Street 2	NULL
E_144	8-3-2020	Mark	Street 1	D_4
E_149	8-3-2020	Bill	Street 9	D_3
E_152	4-5-2020	Charles	Street 4	D_2
E_155	6-5-2020	James	Street 3	D_1
E_167	12-6-2020	Chris	Street 7	D_1
E_168	12-6-2020	Shaun	Street 2	D_3
E_172	19-6-2020	David	Street 5	NULL

Selection

- Select all those employees who joined after 31-3-2020
    - $\sigma_{DoJ>31\text{-}3\text{-}2020}$ (EMPLOYEE)

EMPLOYEE

<u>EmpID</u>	DoJ	Name	Address	Dept
E_112	12-2-2020	John	Street 1	D_2
E_134	8-3-2020	Andy	Street 2	NULL
E_144	8-3-2020	Mark	Street 1	D_4
E_149	8-3-2020	Bill	Street 9	D_3
E_152	4-5-2020	Charles	Street 4	D_2
E_155	6-5-2020	James	Street 3	D_1
E_167	12-6-2020	Chris	Street 7	D_1
E_168	12-6-2020	Shaun	Street 2	D_3
E_172	19-6-2020	David	Street 5	NULL

- Result:

$\sigma_{DoJ>31\text{-}3\text{-}2020}$ (EMPLOYEE)

<u>EmpID</u>	DoJ	Name	Address	Dept
E_152	4-5-2020	Charles	Street 4	D_2
E_155	6-5-2020	James	Street 3	D_1
E_167	12-6-2020	Chris	Street 7	D_1
E_168	12-6-2020	Shaun	Street 2	D_3
E_172	19-6-2020	David	Street 5	NULL

Selection

- Select all those employees who joined after 31-3-2020 and who work in D_2
  - $\sigma_{(\text{DoJ}>31\text{-}3\text{-}2020 \text{ AND } \text{Dept}=\text{D\_2})}$ (EMPLOYEE)

- Result:

$\sigma_{(\text{DoJ}>31\text{-}3\text{-}2020 \text{ AND } \text{Dept}=\text{D\_2})}$ (EMPLOYEE)

<u>EmpID</u>	DoJ	Name	Address	Dept
E_152	4-5-2020	Charles	Street 4	D_2

EMPLOYEE

<u>EmpID</u>	DoJ	Name	Address	Dept
E_112	12-2-2020	John	Street 1	D_2
E_134	8-3-2020	Andy	Street 2	NULL
E_144	8-3-2020	Mark	Street 1	D_4
E_149	8-3-2020	Bill	Street 9	D_3
E_152	4-5-2020	Charles	Street 4	D_2
E_155	6-5-2020	James	Street 3	D_1
E_167	12-6-2020	Chris	Street 7	D_1
E_168	12-6-2020	Shaun	Street 2	D_3
E_172	19-6-2020	David	Street 5	NULL

Selection

- Select all those employees who joined after 31-3-2020 or who work in D_2
  - ○ $\sigma_{(DoJ>31\text{-}3\text{-}2020 \text{ OR } Dept=D\_2)}$ (EMPLOYEE)

- Result:

$\sigma_{(DoJ>31\text{-}3\text{-}2020 \text{ OR } Dept=D\_2)}$ (EMPLOYEE)

EmpID	DoJ	Name	Address	Dept
E_152	4-5-2020	Charles	Street 4	D_2
E_155	6-5-2020	James	Street 3	D_1
E_167	12-6-2020	Chris	Street 7	D_1
E_168	12-6-2020	Shaun	Street 2	D_3
E_172	19-6-2020	David	Street 5	NULL
E_112	12-2-2020	John	Street 1	D_2

EMPLOYEE

EmpID	DoJ	Name	Address	Dept
E_112	12-2-2020	John	Street 1	D_2
E_134	8-3-2020	Andy	Street 2	NULL
E_144	8-3-2020	Mark	Street 1	D_4
E_149	8-3-2020	Bill	Street 9	D_3
E_152	4-5-2020	Charles	Street 4	D_2
E_155	6-5-2020	James	Street 3	D_1
E_167	12-6-2020	Chris	Street 7	D_1
E_168	12-6-2020	Shaun	Street 2	D_3
E_172	19-6-2020	David	Street 5	NULL

Projection

- Retrieves attributes (columns) from a relation
- The duplicates in resulting relation are removed
- Represented as $\pi_{A1,\ A2,..An}$ (*R*), where
  - R is the input relation
  - A1, A2, .. An are the attributes to be retrieved, or projected

Projection

- Project Emp ID and Name from EMPLOYEE
  - ○ $\pi_{\text{EmpID,Name}}$ (EMPLOYEE)
- Result:

$\pi_{\text{EmpID,Name}}$ (EMPLOYEE)

<u>EmpID</u>	Name
E_112	John
E_134	Andy
E_144	Mark
E_149	Bill
E_152	Charles
E_155	James
E_167	Chris
E_168	Shaun
E_172	David

EMPLOYEE

<u>EmpID</u>	DoJ	Name	Address	Dept
E_112	12-2-2020	John	Street 1	D_2
E_134	8-3-2020	Andy	Street 2	NULL
E_144	8-3-2020	Mark	Street 1	D_4
E_149	8-3-2020	Bill	Street 9	D_3
E_152	4-5-2020	Charles	Street 4	D_2
E_155	6-5-2020	James	Street 3	D_1
E_167	12-6-2020	Chris	Street 7	D_1
E_168	12-6-2020	Shaun	Street 2	D_3
E_172	19-6-2020	David	Street 5	NULL

Projection

- Project Date of Joining from EMPLOYEE
  - ○ π_{DoJ} (EMPLOYEE)

- Result: π_{DoJ} (EMPLOYEE)

DoJ
12-2-2020
8-3-2020
4-5-2020
6-5-2020
12-6-2020
19-6-2020

Duplicates removed!

EMPLOYEE

EmpID	DoJ	Name	Address	Dept
E_112	12-2-2020	John	Street 1	D_2
E_134	8-3-2020	Andy	Street 2	NULL
E_144	8-3-2020	Mark	Street 1	D_4
E_149	8-3-2020	Bill	Street 9	D_3
E_152	4-5-2020	Charles	Street 4	D_2
E_155	6-5-2020	James	Street 3	D_1
E_167	12-6-2020	Chris	Street 7	D_1
E_168	12-6-2020	Shaun	Street 2	D_3
E_172	19-6-2020	David	Street 5	NULL

Set operations: Union

- Union is a binary operation that takes two relations as input, and returns a relation that contains all the tuples which are either in first or second or both relations
- Duplicates are removed
- The two input relations must be union-compatible
  - The relations must have same attributes with same domain
- Represented as R1 U R2

Set operations: Union

- Find CNIC of all people who are either employees or students
  - ○ π_{CNIC} (EMPLOYEE) U π_{CNIC} (STUDENT)

EMPLOYEE

<u>CNIC</u>	Salary
1234	50k
1123	100k
1343	70k
1315	90k

STUDENT

<u>CNIC</u>	Major
1213	CS
1315	Mgmt

- Result:

  π_{CNIC} (EMPLOYEE) U π_{CNIC} (STUDENT)

CNIC
1234
1123
1343
1315
1213

Set operations: Intersection

- Intersection is a binary operation that takes two relations as input, and returns a relation that contains all the tuples which are common in both relations
- The two input relations must be union-compatible
    - The relations must have same attributes with same domain
- Represented as R1 ∩ R2

Set operations: Intersection

- Find CNIC of all people who are EMPLOPYEEs as well as STUDENTS
    - π_{CNIC} (EMPLOYEE) ∩ π_{CNIC} (STUDENT)

- Result:

π_{CNIC} (EMPLOYEE) ∩ π_{CNIC} (STUDENT)

CNIC
1234
1315

EMPLOYEE

CNIC	Salary
1234	50k
1123	100k
1343	70k
1315	90k

ALUMNUS

CNIC	Degree
1213	BSc
1315	BBA

STUDENT

CNIC	Major
1234	CS
1315	Mgmt

Set operations: Difference

- Difference is a binary operation that takes two relations as input, and returns a relation that contains all the tuples of first relation which are not in second relation
- The two input relations must be union-compatible
  - The relations must have same attributes with same domain
- Represented as R1 - R2

Set operations: Difference

- Find CNIC of all people who are EMPLOPYEEs but not STUDENTS
  - ○ π_{CNIC} (EMPLOYEE) - π_{CNIC} (STUDENT)

- Result:

π_{CNIC} (EMPLOYEE) - π_{CNIC} (STUDENT)

CNIC
1123
1343

EMPLOYEE

CNIC	**Salary**
1234	50k
1123	100k
1343	70k
1315	90k

ALUMNUS

CNIC	**Degree**
1213	BSc
1315	BBA

STUDENT

CNIC	**Major**
1234	CS
1315	Mgmt

Cartesian Product (or Cross product)

- Cartesian Product of two relations contains all the combinations of tuples from both relations
- The two input relations are not required to be union-compatible
- If m and n are degrees (no. of attributes) of R1 and R2 respectively, then the degree of their Cartesian product will be m + n
- If p and q are cardinalities (no. of tuples) of R1 and R2 respectively, then the cardinality of their Cartesian product will be p x q
- Represented as R1 x R2

Cartesian Product

EMPLOYEE

EmpID	DoJ	Name	Address
E_112	12-2-2020	John	Street 1
E_134	8-3-2020	Andy	Street 2
E_144	8-3-2020	Mark	Street 1

DEPARTMENT

DepID	Name	Location
D_1	Sales	Site 1
D_2	Marketing	Site 1
D_3	Production	Site 2
D_4	HR	Site 3

EMPLOYEE X DEPARTMENT

EmpID	DoJ	Name	Address	DepID	D_Name	Location
E_112	12-2-2020	John	Street 1	D_1	Sales	Site 1
E_112	12-2-2020	John	Street 1	D_2	Marketing	Site 1
E_112	12-2-2020	John	Street 1	D_3	Production	Site 2
E_112	12-2-2020	John	Street 1	D_4	HR	Site 3
E_134	8-3-2020	Andy	Street 2	D_1	Sales	Site 1
E_134	8-3-2020	Andy	Street 2	D_2	Marketing	Site 1
E_134	8-3-2020	Andy	Street 2	D_3	Production	Site 2
E_134	8-3-2020	Andy	Street 2	D_4	HR	Site 3
E_144	8-3-2020	Mark	Street 1	D_1	Sales	Site 1
E_144	8-3-2020	Mark	Street 1	D_2	Marketing	Site 1
E_144	8-3-2020	Mark	Street 1	D_3	Production	Site 2
E_144	8-3-2020	Mark	Street 1	D_4	HR	Site 3

Join

- The example of Cartesian Product that we just discussed involves two relations that do not seem to have any relationship with each other!
    - Remember, when we converted ERD into relational model, we connected two related entity types by using one’s PK as FK in the other
- So let’s see a Cartesian Product of two relations which are related with each other

Join

EMPLOYEE

EmpID	DoJ	Name	Address
E_112	12-2-2020	John	Street 1
E_134	8-3-2020	Andy	Street 2
E_144	8-3-2020	Mark	Street 1
E_155	10-5-2021	James	Street 5

DEPARTMENT

DepID	Name	Location	HoD
D_1	Sales	Site 1	E_112
D_2	Marketing	Site 1	E-134
D_3	Production	Site 2	E-155

EMPLOYEE X DEPARTMENT

EmpID	DoJ	Name	Address	DepID	D_Name	Location	HoD
E_112	12-2-2020	John	Street 1	D_1	Sales	Site 1	E_112
E_112	12-2-2020	John	Street 1	D_2	Marketing	Site 1	E-134
E_112	12-2-2020	John	Street 1	D_3	Production	Site 2	E-155
E_134	8-3-2020	Andy	Street 2	D_1	Sales	Site 1	E_112
E_134	8-3-2020	Andy	Street 2	D_2	Marketing	Site 1	E-134
E_134	8-3-2020	Andy	Street 2	D_3	Production	Site 2	E-155
E_144	8-3-2020	Mark	Street 1	D_1	Sales	Site 1	E_112
E_144	8-3-2020	Mark	Street 1	D_2	Marketing	Site 1	E-134
E_144	8-3-2020	Mark	Street 1	D_3	Production	Site 2	E-155
E_155	10-5-2021	James	Street 5	D_1	Sales	Site 1	E_112
E_155	10-5-2021	James	Street 5	D_2	Marketing	Site 1	E-134
E_155	10-5-2021	James	Street 5	D_3	Production	Site 2	E-155

Join

- EMPLOYEE and DEPT are now connected in a relationship
    - EMPLOYEE heads a DEPARTMENT
- But still we see a lot of strange tuples in this Cartesian Product

Join

EMPLOYEE

EmpID	DoJ	Name	Address
E_112	12-2-2020	John	Street 1
E_134	8-3-2020	Andy	Street 2
E_144	8-3-2020	Mark	Street 1
E_155	10-5-2021	James	Street 5

DEPARTMENT

DepID	Name	Location	HoD
D_1	Sales	Site 1	E_112
D_2	Marketing	Site 1	E-134
D_3	Production	Site 2	E-155

EMPLOYEE X DEPARTMENT

EmpID	DoJ	Name	Address	DepID	D_Name	Location	HoD
E_112	12-2-2020	John	Street 1	D_1	Sales	Site 1	E_112
E_112	12-2-2020	John	Street 1	D_2	Marketing	Site 1	E-134
E_112	12-2-2020	John	Street 1	D_3	Production	Site 2	E-155
E_134	8-3-2020	Andy	Street 2	D_1	Sales	Site 1	E_112
E_134	8-3-2020	Andy	Street 2	D_2	Marketing	Site 1	E-134
E_134	8-3-2020	Andy	Street 2	D_3	Production	Site 2	E-155
E_144	8-3-2020	Mark	Street 1	D_1	Sales	Site 1	E_112
E_144	8-3-2020	Mark	Street 1	D_2	Marketing	Site 1	E-134
E_144	8-3-2020	Mark	Street 1	D_3	Production	Site 2	E-155
E_155	10-5-2021	James	Street 5	D_1	Sales	Site 1	E_112
E_155	10-5-2021	James	Street 5	D_2	Marketing	Site 1	E-134
E_155	10-5-2021	James	Street 5	D_3	Production	Site 2	E-155

Join

- EMPLOYEE and DEPT are now connected in a relationship
  - EMPLOYEE heads a DEPARTMENT
- But still we see a lot of strange tuples in this Cartesian Product
- What does the highlighted tuple tells us?
  - Nothing!
- We need to apply certain conditions on the Cartesian Product to make this data meaningful
- What will be the output of the following operation?
  - $\sigma_{(\text{DEPARETMENT.HoD=EMPLOYEE.EmpID})}$ (EMPLOYEE X DEPARTMENT)

Join

EMPLOYEE

EmpID	DoJ	Name	Address
E_112	12-2-2020	John	Street 1
E_134	8-3-2020	Andy	Street 2
E_144	8-3-2020	Mark	Street 1
E_155	10-5-2021	James	Street 5

DEPARTMENT

DepID	Name	Location	HoD
D_1	Sales	Site 1	E_112
D_2	Marketing	Site 1	E-134
D_3	Production	Site 2	E-155

$\sigma_{(\text{DEPARETMENT.HoD=EMPLOYEE.EmpID})}$ (EMPLOYEE X DEPARTMENT)

EmpID	DoJ	Name	Address	DepID	D_Name	Location	HoD
E_112	12-2-2020	John	Street 1	D_1	Sales	Site 1	E_112
E_134	8-3-2020	Andy	Street 2	D_2	Marketing	Site 1	E-134
E_155	10-5-2021	James	Street 5	D_3	Production	Site 2	E-155

We have data of all those employees who are heads of departments!

This information was not available from individual relations… DEPARTMENT does not include HoD’s name or address!

These operations (selection on a cross product) are combined into a single JOIN operation, represented as:

$R_1 \bowtie_{\text{(join condition)}} R_2$

Theta Join

- The previous example will be represented as
  - EMPLOYEE $\bowtie_{\text{EMPLOYEE.EmpID = DEPARTMENT.HoD}}$ DEPARTMENT
- Join allows us to retrieve information by combining data from different but related relations
- This general Join operation, which specifies a condition is called a theta join
  - In theta join, we can specify any condition on any attribute
  - e.g., EMPLOYEE $\bowtie_{\text{EMPLOYEE.DoJ > 31-3-2020}}$ DEPARTMENT

Equi Join

- A more specialized join operation is Equi Join, which only allows equality operator in the join condition
  - $\text{EMPLOYEE} \bowtie_{\text{EMPLOYEE.EmpID} = \text{DEPARTMENT.HoD}} \text{DEPARTMENT}$
- The example we just studied is a case of Equi Join

Natural Join

- A type of equi join which combines two relations on the basis of a common attribute (or set of attributes)
- The extra copies of join attributes are removed!
- Let’s find complete details of all persons who are employees of university
- PERSON $\bowtie$ EMPLOYEE

CNIC	**Name**	**Address**	**BirthDate**	**Salary**
1234	ABC	St 1	1-1-2000	50k
1123	XYZ	St 2	1-2-2000	100k
1343	SDT	St 5	4-3-2005	70k
1315	YYT	St 2	9-2-2001	90k

PERSON

CNIC	**Name**	**Address**	**BirthDate**
1234	ABC	St 1	1-1-2000
1123	XYZ	St 2	1-2-2000
1213	ABX	St 1	3-2-2003
1343	SDT	St 5	4-3-2005
1315	YYT	St 2	9-2-2001
1420	STD	St 7	3-1-2008

EMPLOYEE

CNIC	**Salary**
1234	50k
1123	100k
1343	70k
1315	90k

Outer Joins

- The Join operations that we studied contain only those tuples from both relations which satisfy the join criteria
- Sometimes we need to see all tuples from first, or second, or both relations even if some of these tuples don’t meet the join condition
- This is achieved through outer join operations
  - Three variations:
  - Left outer join
  - Right outer join
  - Full outer join

Left outer Join

- The Join operation contains all tuples that satisfy the join condition, but also includes all tuples of left relation
  - The attributes pertaining to right relation are padded with NULL
- $\text{PERSON} ⟕_{\text{PERSON.CNIC=EMPLOYEE.CNIC}} \text{EMPLOYEE}$

CNIC	Name	Address	BirthDate	Salary
1234	ABC	St 1	1-1-2000	50k
1123	XYZ	St 2	1-2-2000	100k
1213	ABX	St 1	3-2-2003	NULL
1343	SDT	St 5	4-3-2005	70k

PERSON

CNIC	Name	Address	BirthDate
1234	ABC	St 1	1-1-2000
1123	XYZ	St 2	1-2-2000
1213	ABX	St 1	3-2-2003
1343	SDT	St 5	4-3-2005

EMPLOYEE

CNIC	Salary
1234	50k
1123	100k
1343	70k
1315	90k

Right outer Join

- The Join operation contains all tuples that satisfy the join condition, but also includes all tuples of right relation
  - The attributes pertaining to left relation are padded with NULL
- PERSON ⟖$_{\text{PERSON.CNIC=EMPLOYEE.CNIC}}$ EMPLOYEE

<u>CNIC</u>	Name	Address	BirthDate	Salary
1234	ABC	St 1	1-1-2000	50k
1123	XYZ	St 2	1-2-2000	100k
1343	SDT	St 5	4-3-2005	70k
1315	NULL	NULL	NULL	90k

PERSON

<u>CNIC</u>	Name	Address	BirthDate
1234	ABC	St 1	1-1-2000
1123	XYZ	St 2	1-2-2000
1213	ABX	St 1	3-2-2003
1343	SDT	St 5	4-3-2005

EMPLOYEE

<u>CNIC</u>	Salary
1234	50k
1123	100k
1343	70k
1315	90k

Full outer Join

- The Join operation contains all tuples that satisfy the join condition, but also includes all tuples of both right and left relations
  - Basically a union of left outer join and right outer join
- $\text{PERSON} ⟗_{\text{PERSON.CNIC=EMPLOYEE.CNIC}} \text{EMPLOYEE}$

CNIC	Name	Address	BirthDate	Salary
1234	ABC	St 1	1-1-2000	50k
1123	XYZ	St 2	1-2-2000	100k
1213	ABX	St 1	3-2-2003	NULL
1343	SDT	St 5	4-3-2005	70k
1315	NULL	NULL	NULL	90k

PERSON

CNIC	Name	Address	BirthDate
1234	ABC	St 1	1-1-2000
1123	XYZ	St 2	1-2-2000
1213	ABX	St 1	3-2-2003
1343	SDT	St 5	4-3-2005

EMPLOYEE

CNIC	Salary
1234	50k
1123	100k
1343	70k
1315	90k

Thanks a lot