3-3-2020

UNIT-II

The Relational Model and Relational Algebra

- Course Learning objective 2:

To gain the knowledge of relational algebra and gain the knowledge SQL and PL/SQL

Data Abstraction:

It is a mechanism by which the details of the database structure and the storage details are hidden from the end user

Data Model: means for achieving data abstract^n

a set of concepts for defining db structure
- datatypes
- relationships
- constraints

retrieval and update operation

Types of Model:
1. Network model
2. Hierarchical model
3. Relational Model

Relational Algebra :

The basic set of operations for the relational model is the relational algebra. These operations enable a user to specify retrieval requests. The result of a retrieval is a new relation which may have been formed from one or more relations.

Operations

Operat^{n}s borrowed from the SET theory	Specifically available for use with relational model
• UNION	• SELECT } UNARY OPERATION
• INTERSECTION	• PROJECT } UNARY OPERATION
• DIFFERENCE	• JOIN - BINARY OPERATION
• CARTESIAN PRODUCT	

* SELECT :

- Unary operation.
- Retrieves a subset of tuples from a relation that satisfies a <u>selection condition</u>

- filter / partition

syntax

$\sigma_{\langle condition \rangle}(R)$ ← name of a relation / algebraic expression

notation: σ

<condition> ↓ boolean expression

<attribute><comparison op><constant>

<attribute><comparison op><attribute>

comparison op ↓ $=, \neq, <, <=, >, >=$

can be applied on ordered values (numeric, dates)

for strings : $=, \neq$

3 multiple clauses can be combined using logical operators AND, OR, NOT

$$R' \leftarrow \sigma_{\langle condition \rangle}(R)$$

- degree of relation R' ← same as R i.e. n
  no. of attributes, horizontal partition
- Selectivity of relation R' ← in resulting relation <= (R)
  no. of rows that satisfies the condition.

4-3-2020

Properties of Selection operation

1. Unary
2. horizontal partition of tuples in a relation
4. degree of new relation is same as R
5. Selection operation is commutative in nature

$$\sigma_{<cond1>}(\sigma_{<cond2>}(R)) = \sigma_{<cond2>}(\sigma_{<cond1>}(R))$$

6. A sequence of selects can be combined into a single select operation

$$\sigma_{<cond1>}(\sigma_{<cond2>}(\sigma_{<cond3>} \ldots (\sigma_{<cond\,n>}(R))))$$

$$= \sigma_{<cond1> AND <cond2> \ldots AND <cond\,n>}(R)$$

PROJECTION :

- Unary operation.
- Vertical partition of relation
- retrieve a subset of columns/attributes.

Syntax

$\Pi_{\underbrace{\langle col^n \text{ or attribute list} \rangle}_{\text{a comma separated list}}} (R)$ ← algebraic exp / Relation.

↗ operator

- generate a new relation (R').
- R' ← deg(R') <= deg(R)

  or no. of attribute listed in the attribute list
- no. of tuples in R' = no. of tuples in R

* attribute list

  Key attributes / non-key attribute

Eg: $\Pi_{SSN, name}(Student)$ $\Pi_{gender, salary}(Emp)$

- No duplicates
- automatically delete / eliminate duplicates (duplicate elimination)

* Properties :

Eg: Consider the following schema

Sailors (<u>Sid</u>, Sname, rating, age)

Boats (<u>Bid</u>, Bname, Bcolor)

Reserves (<u>Sid</u>, <u>Bid</u>, day)

1. List id, name and age of sailors with rating > 8
2. List details of sailors with age bet^n 20 and 30
3. List details of all boats
4. List details of only red boats.
5. List details of boats which are either red/blue
6. List age of all sailors.
7. List all distinct boat names
8. List id's of boats reserved on 4th Mar 2020
9. List id's of boats reserved by sailor with id 10 or 20

1. $\pi_{sid, name, age}(\sigma_{rating>8}(sailor))$

$R' = \sigma_{rating>8}(Sailors)$

$R'' = \pi_{sid, name, age}(R')$

3. $\pi_{Bid, Bname, Bcolor}(Boats)$

2. $\pi_{sid, sname, rating, age}(\sigma_{age>=20 \,\&\&\, age<=30}(Sailors))$

Operations from Set Theory:

1. Union
2. Intersection
3. Set Difference
4. Cartesian product.

- These all are binary operations
- Pre-requisite for 1, 2, 3 is "union compatible"

$R(A_1, A_2, \dots A_n)$

$S(B_1, B_2, \dots B_n)$

1. $deg(R) = deg(S)$
2. for $1 \leq i \leq n$

$domain(A_i) = domain(B_i)$

$\therefore$ R and S are union compatible.

Cartesian Product:

$R \rightarrow n$ tuples $(A_1, A_2, \dots A_n)$

$S \rightarrow m$ tuples $(B_1, B_2, \dots B_m)$

The cartesian product $R \times S$ will combine each tuple in R with every tuple in S with attributes of R appearing first followed by attributes of S.

$deg(R \times S) = n + m$,

no. of tuples in $R \times S = n \times m$

5-3-2020

* Handling violation of referential integrity constraint

1 On delete set NULL
2. On delete Cascade

Find the union, intersection and set difference for the given

Student :

fname	lastname
Suresh	Rao
Ramesh	Shah
Dale	Jones
Amar	Narayan
Jimmy	John

Professor

fname	lname
John	Smith
Richard	Brown
Suresh	Rao
Benjamin	Brown
Ramesh	shah

Union :

fname	lname.
Suresh	Rao
Ramesh	shah
Dale	Jones
Amar	Narayan
Jimmy	John
John	Smith
Richard	Brown
Benjamin	Brown.

Intersection :

fname	last name.
Suresh	Rao
Ramesh	Shah

R - S
⇒ S - P

fname	lname
Dales	Jones
Amar	Narayan
Jimmy	John

* To retrieve data or combine related tuple from 2 relations into single tuples, we use the join operation.

* This operation is imp. because it allows us to process relationship among relation

Syntax :

$$R \bowtie_{\theta} S$$

R, S - name of relations
θ - join condition.

Join Condition :

$\theta \leftarrow$ join condition

• involves attributes of R and S

<attribute name><comparison op.><attribute name>

$=, <>, <, >, <=, >=$.

Types of Join

Inner join:

1 Theta join : Combines tuples from diff. relations provided they satisfy the ~~join~~ theta condition

- The θ condition is denoted by symbol θ

2 Equi Join :

- A join operation where the only operator used is equal to i.e. equality operator is called an equi join.
- An equi join is a variation of θ join
- The notation for equi join $R \bowtie_{\langle join\ condit^n \rangle} S$

Consider the following relations:

Student

Sid	Name	Std
101	Alex	10
102	Maria	11

Subjects

Class	Subject
10	Maths
10	English
11	Music
11	Sport

Determine the results of following operations

1. Student × Subjects
2. Student $\bowtie_{\text{Student.std} < \text{subject.Class}}$ Subjects
3. Student $\bowtie_{\text{student.Std} = \text{Subject.class}}$ Subjects

2.

Sid	Name	Std	Class	Subject
101	Alex	10	11	Music
101	Alex	10	11	Sport

3.

Sid	Name	std	class	Subject
101	Alex	10	10	Maths
101	Alex	10	10	English
102	Maria	11	11	Music
102	Maria	11	11	Sports.

Natural Join:

- It eliminates redundant attributes appearing in the output of equi join
- Syntax: $R \bowtie S$
- The natural join does not use any comparison Operator
- It can perform a natural join only if there is atleast 1 common attribute that exists betn 2 relations
- In addition the attributes must have the same name and domain

7-3-2020

θ join, equi join and the natural joins are called inner joins because only those combinatⁿ of tuples which satisfy the join conditⁿ are retained in the result of the join operation. Those that do not satisfy the join conditⁿ are discarded

Outer Joins :

Left Outer

All the tuples from the left relation are included in resulting relation.

If there are tuples in R without any matching tuple in the right relation S then the S attributes of the resulting relation are made NULL

R ⟕ S

2 Right Join

R ⟖ S

All the tuples from the right relation S are including in the relation.

- If there are tuples in S with no matching tuples in R, then the R attributes of the resulting relation are made NULL.

3. Full Outer Join

R ⟗

All the tuples from both participating relations are included in the resulting relation.

- If there are no matching tuples for both relations, ~~they are~~ their respective unmatching attributes are made NULL

For the given relation determine the results of

i. Left ii. Right iii Full OJ.s.

COURSES : (R)

id	Name
100	Database
101	Mechanics
102	Electronics

HOD. (S)

id	Name
100	Alex
102	Maya
104	Mira

i. R ⟕ S

R.id = S.id

id	Name	id	Name
100	Database	100	Alex
101	Mechanics	NULL	NULL
102	Electronics	102	Maya

ii. R ⟖ S

R.id : S.id

id	Name	id	Name
100	Database	100	Alex
102	Electronics	102	Maya
NULL	NULL	104	Mira

iii. $R \bowtie_{R.id = S.id} S$

id	Name	id	Name
100	Database	100	Alex
101	Mechanics	NULL	NULL
102	Electronics	102	Maya
NULL	NULL	104	Mira

Consider the 2 tables T_1 and T_2 shown in fig. below.
Show the results of following operatⁿs

T_1

P	Q	R
10	a	5
15	b	8
25	a	6.

T_2

A	B	C
10	b	6
25	c	3
10	b	5

1. $T_1 \underset{T_1.P = T_2.A}{\bowtie} T_2$

2. $T_1 \underset{T_1.Q = T_2.B}{\bowtie} T_2$

3. $T_1 \underset{T_1.P = T_2.A}{⟕} T_2$

4. $T_1 \underset{T_1.Q = T_2.B}{⟖} T_2$

5. $T_1 \cup T_2$

6. $T_1 \underset{(T_1.P = T_2.A \text{ AND } T_1.R = T_2.C)}{\bowtie} T_2$

1. $T_1 \underset{T_1.p = T_2.A}{⟕} T_2$

~~id~~ P	Q	R	A	B	C.
10	a	5	10	b	6
10	a	5	10	b	5
25	a	6	25	c	3.

2 $T_1 \bowtie T_2$

$T_1.Q = T_2.B$.

P	Q	R	A	B	C.
15	b	8	10	b	6
15	b	8	10	b	5

3.

P	Q	R	A	B	C.
10	a	5	10	b	6.
15	b	8			
25	a	6			

Find the results of these ~~ope~~ expressions for relation schema.

R

A	B	C	D
1	2	3	4
2	5	1	
3	4	2	6
4	2	5	3

S

C	D	E
1	2	4
3	4	1
5	1	6
4	2	3

1. $R \cup S$
2. $R \cap S$
3. $R \bowtie S$
4. $R \bowtie_{R.C = S.C} S$
5. $R \bowtie_{R.A = S.C} S$
6. $R \bowtie_{R.A = S.E} S$