

Module 09

Partha Pratim Das

Outline

Additional Basi Operations

Cartesian Product

Rename AS

String Values

Order By Clause Select Top / Fetch

Where Claus Predicates

Module Summar

Database Management Systems

Module 09: Introduction to SQL/2

Partha Pratim Das

Department of Computer Science and Engineering Indian Institute of Technology, Kharagpur

ppd@cse.iitkgp.ac.in

Module Recap

Module 09

Partha Pratii Das

Objectives & Outline

Additional Bas Operations Cartesian Produc

String Values Order By Clause

Clause
Where Clause

Module Summar

- Introduced relational query language
- Familiarized with data definition and basic query structure

Module Objectives

Module 09

Partha Pratir Das

Objectives & Outline

Additional Base Operations Cartesian Produc

Rename AS

Operation String Values

Order By Clause Select Top / Feb

Where Clause Predicates

Module Summary

• To complete the understanding of basic query structure

Module Outline

Module 09

Partha Pratir Das

Objectives & Outline

Additional Basi Operations

Cartesian Product

Rename AS

String Values
Order By Clause
Select Top / Fetch

Where Clause Predicates Duplicates

Module Summar

Additional Basic Operations

- o Cartesian Product
- Rename AS Operation
- String Values
- o Order By
- Select Top / Fetch
- Where Clause Predicate
- o Duplicates

Additional Basic Operations

Module 09

Partha Pration

Objectives Outline

Additional Basic Operations

Cartesian Product

Rename AS Operation

String Values

Select Top / Feto

Where Clause Predicates

Module Summar

Additional Basic Operations

Database Management Systems Partha Pratim Das 09.5



Cartesian Product

Module 09

Partha Pratin Das

Objectives Outline

Additional Basic Operations Cartesian Product Rename AS

String Values Order By Clause Select Top / Fetch Clause Where Clause Predicates

Module Summar

• Find the Cartesian product *instructor X teaches*

select *

from instructor, teaches

- generates every possible instructor-teaches pair, with all attributes from both relations
- For common attributes (for example, *ID*), the attributes in the resulting table are renamed using the relation name (for example, *instructor.ID*)
- Cartesian product not very useful directly, but useful combined with where-clause condition (selection operation in relational algebra)



Cartesian Product

Module 09

Partha Pratir Das

Objectives Outline

Additional Bas Operations

Cartesian Product

Danama AC

String Values Order By Clause

Where Clause Predicates

Module Summar

				teaches												
ID	name	dept_name		-	salary		IE)	course_id		sec_ia	l se	semester		year	
10101	Sriniv	asan		np. Sci.		65000		101	01	CS-101		1	F	Fall		2009
12121	Wu			ance		90000		101	01	CS-31	5	1	S	pring		2010
15151	Moza		Mu			40000		10101		CS-347		1	1 F			2009
22222	Einste	in	Phy	sics	cs 95000			121	21	FIN-201		1	S	pring		2010
32343	El Said	t		tory	60000			151	51	MU-199		1		pring		2010
33456	Gold		Physics		87000			222	22	PHY-101		1	F	all		2009
45565	Katz		Comp. Sci.			75000		323	43	HIS-351		1	S	pring		2010
58583	Calific	eri		tory		62000		455	65	CS-10	1	1	S	pring		2010
76543	Singh			ance		80000		455	65	CS-31	9	1		pring		2010
76766	Crick		Biology		ı	72000		76766 I		BIO-1	01	1	S	ummei	r	2009
83821	Inst.ID	nan	ue.	dept_na	1110	ealary	teach	oc ID	co	urse_id	cec i	d 60	mecter	year	П	2010
98345	Inotal				_					_	500_I	_			Н	2009
	10101			Comp.			1010			-101	1	Fa		2009	Н	2009
	10101			Comp.			1010			-315	1		ring	2010		2010
	10101	Srini	vasan	Comp.	Sci.	65000	1010			-347	1	Fa		2009		2009
	10101			Comp.			1212			V-201	1		ring	2010	_	
	10101	Srini	vasan	Comp.	Sci.	65000	1515			J-199	1		ring	2010		
	10101	Srini	vasan	Comp.	Sci.	65000	2222	22	PE	IY-101	1	Fa	11	2009		
		***		***							101			- 91		
											177					
	12121	Wu		Finance		90000	1010			-101	1	Fa		2009		
	12121	Wu		Finance		90000	1010			-315	1		ring	2010		
	12121	Wu		Pinance		90000	1010			-347	1	Fa		2009		
	12121	Wu		Pinance		90000	1212			V-201	1		ring	2010		
	12121	Wu		Finance		90000	1515			J-199	1		ring	2010		
	12121	Wu		Pinance	2	90000	2222	22	PH	IY-101	1	Fa	11	2009		
		l				i						Ι.				

Module 09

Partha Pratio

Objectives Outline

Additional Basic Operations Cartesian Product

Rename AS
Operation
String Values
Order By Clause

Select Top / Fetcl Clause Where Clause Predicates

Module Summai

• Find the names of all instructors who have taught some course and the course_id

select name, course_id

from instructor, teaches

where instructor.ID = teaches.ID

Equi-Join, Natural Join

instructor									teaches									
ID	name	name dept		лате	ne salary			ID		course_id		sec_id		semester			year	
10101		Srinivasan Con		np. Sci.	np. Sci. 65000		٦.	10101		CS-101			1 F		Fall		2009	
12121	Wu		Fin	ance	ce 90000			10101		CS-315			1 8		pring		2010	
15151	Moza	Mozart Mus		sic 400		40000		10101		CS-347					Fall		2009	
22222	Einste	in	Phy	rsics 95000			12121		FIN-201			1 2		Spring		2010		
32343				tory 60000				15151		MU-199		1			Spring		2010	
33456	Gold	Gold Phy		sics 87000			22222		PHY-101					Fall		2009		
45565	Katz			np. Sci.	75000			32343		HIS-351					pring		2010	
58583	Califie	ri	His	tory		62000		455	65	CS-10	1		1		oring		2010	
76543	Singh		Fin	ance		80000		455	65	CS-31			1		oring		2010	
76766	Crick			logy	72000			76766			BIO-101		î	Summer		r	2009	
83821	Inst.ID	nan		dept_no		calani	teach	er ID		urse id		14			year	H	2010	
98345	InstaD	nan	w	aepr_m	me	smary	tenen	es.ID	00.	urse_na	Set_	iei	semi	ster	yew	ш	2009	
	10101	Srini	vasan	Comp.	Sci.	65000	1010)1	CS	-101	1	Т	Fall		2009	Ш	2009	
				Comp.			1010)1	CS	-315	1		Spri	ing	2010	Ш	2010	
	10101	Srini	vasan	Comp.	Sci.	65000	1010)1	CS	-347	1		Fall		2009	Ш	2009	
	10101			Comp.			121	21	FIR	V-201	-1	+	Spri	ne	2010	-		
				Comp.			151	51	MI	J-199	1	4	Spr		2010	H		
				Comp.			222	22	PF	IY-101	-1	4	Fall		2009	L		
				'														
		Wu		Finance	,	90000	101)1	CS	-101	1	-	Fall		2009	H	-	
	12121	Wu		Finance	_	90000	101)1	CS	315	-1	4	Spri	ne	2010			
		Wu		Pinano		90000	101)1	CS	347	1	4	Fall		2009			
	12121	Wu		Pinano		90000	1213	21	FII	N-201	1		Spri	ine	2010			
		Wu		Finance		90000	151			I-199	l î	4	Spri		2010			
_		Wu		Pinance		90000	222	22	PI	IY-101	î	4	Fall	0	2009	L		

Database Management Systems Partha Pratim Das 09.8



Examples

Module 09

Partha Pratii Das

Objectives Outline

Additional Base Operations

Cartesian Product

Operation
String Values
Order By Clause
Select Top / Fetcl
Clause

Where Clause Predicates

Module Summar

 Find the names of all instructors in the Art department who have taught some course and the course_id

select name, course_id

from *instructor*, *teaches*

where *instructor.ID* = *teaches.ID* **and** *instructor.dept_name* = 'Art'



Rename AS Operation

Module 09

Partha Pratim Das

Objectives Outline

Additional Basi Operations Cartesian Product

Operation String Values Order By Clause Select Top / Fetcl Clause

Where Clause Predicates Duplicates

Module Summai

- The SQL allows renaming relations and attributes using the as clause:
 old_name as new_name
- Find the names of all instructors who have a higher salary than some instructor in 'Comp. Sci'.

```
select distinct T.name
from instructor as T, instructor as S,
where T.salary > S.salary and S.dept_name = 'Comp. Sci'
```

• Keyword **as** is optional and may be omitted instructor **as** $T \equiv instructor T$



Cartesian Product Example

Module 09

Rename AS Operation

• Relation *emp_super*

person	supervisor
Bob	Alice
Mary	Susan
Alice	David
David	Mary

- Find the supervisor of "Bob"
- Find the supervisor of the supervisor of "Bob"
- Find ALL the supervisors (direct and indirect) of "Bob"



String Operations

Module 09

String Values

- SQL includes a string-matching operator for comparisons on character strings. The operator like uses patterns that are described using two special characters:
 - o percent (%). The % character matches any substring
 - o underscore (_). The _ character matches any character
- Find the names of all instructors whose name includes the substring "dar"
 - select name
 - from instructor
 - where name like '%dar%'
- Match the string "100%"
 - like '100%' escape '\'
- in that above we use backslash (\) as the escape character



String Operations (2)

Module 09

Partha Pratir Das

Objectives Outline

Operations

Cartesian Produc

String Values
Order By Clause

Select Top / Feti Clause Where Clause Predicates

Module Summar

- Patterns are case sensitive
- Pattern matching examples:
 - o 'Intro%' matches any string beginning with "Intro"
 - o '%Comp%' matches any string containing "Comp" as a substring
 - o '_ _ _ ' matches any string of exactly three characters
 - \circ '_ _ _ %' matches any string of at least three characters
- SQL supports a variety of string operations such as
 - concatenation (using "||")
 - converting from upper to lower case (and vice versa)
 - o finding string length, extracting substrings, etc.



Ordering the Display of Tuples

Module 09

Partha Pratin Das

Objectives Outline

Additional Basic Operations Cartesian Product Rename AS Operation

String Values

Order By Clause

Select Top / Fetch Clause

Where Clause Predicates Duplicates

∕lodule Summar

• List in alphabetic order the names of all instructors

select distinct name from instructor

order by name

order by name

 We may specify desc for descending order or asc for ascending order, for each attribute; ascending order is the default.

Example: order by name desc

Can sort on multiple attributes

Example: order by dept_name, name



Selecting Number of Tuples in Output

Module 09

Partha Pratin Das

Objectives Outline

Additional Basic Operations Cartesian Product Rename AS Operation String Values

String Values
Order By Clause
Select Top / Fetch
Clause

Where Clause Predicates Duplicates

Module Summai

- The **Select Top** clause is used to specify the number of records to return
- The **Select Top** clause is useful on large tables with thousands of records. Returning a large number of records can impact performance

select top 10 distinct name from instructor

- Not all database systems support the SELECT TOP clause.
 - SQL Server & MS Access support select top
 - MySQL supports the limit clause
 - Oracle uses fetch first n rows only and rownum

select distinct name from instructor order by name fetch first 10 rows only



Where Clause Predicates

Module 09

Partha Pratin Das

Objectives Outline

Operations

Cartesian Product

Operation
String Values
Order By Clause
Select Top / Fetch

Where Clause Predicates Duplicates

Module Summai

SQL includes a between comparison operator

• Example: Find the names of all instructors with salary between \$90,000 and \$100,000 (that is, \geq \$90,000 and \leq \$100,000)

select name from instructor where salary between 90000 and 100000

Tuple comparison

select *name, course_id* **from** *instructor*, *teaches*

where (instructor.ID, dept_name) = (teaches.ID, 'Biology');



In Operator

Module 09

Partha Pratin Das

Objectives Outline

Additional Bas Operations Cartesian Produc Rename AS

String Values
Order By Clause
Select Top / Feto

Where Clause Predicates

Module Summai

- The in operator allows you to specify multiple values in a where clause
- The in operator is a shorthand for multiple or conditions
 - select name
 - **from** *instructor*
 - where dept_name in ('Comp. Sci.', 'Biology')



Duplicates

Module 09

Partha Pratir Das

Objectives Outline

Additional Bas Operations Cartesian Produc Rename AS Operation

String Values
Order By Clause
Select Top / Fetch
Clause
Where Clause

Duplicates

Module Summai

- In relations with duplicates, SQL can define how many copies of tuples appear in the result
- Multiset versions of some of the relational algebra operators given multiset relations r_1 and r_2 :
 - a) $\sigma_{\theta}(r_1)$: If there are c_1 copies of tuple t_1 in r_1 , and t_1 satisfies selections σ_{θ} , then there are c_1 copies of t_1 in $\sigma_{\theta}(r_1)$
 - b) $\Pi_A(r)$: For each copy of tuple t_1 in r_1 , there is a copy of tuple $\Pi_A(t_1)$ in $\Pi_A(r_1)$ where $\Pi_A(t_1)$ denotes the projection of the single tuple t_1
 - c) $r_1 \times r_2$: If there are c_1 copies of tuple t_1 in r_1 and c_2 copies of tuple t_2 in r_2 , there are $c_1 \times c_2$ copies of the tuple $t_1.t_2$ in $r_1 \times r_2$

Duplicates (2)

Module 09

Partha Pratim Das

Objectives Outline

Additional Bas Operations Cartesian Produc

Rename AS
Operation
String Values
Order By Clause
Select Top / Fetch

Clause
Where Clause
Predicates

Duplicates

Module Summar

• Example: Suppose multiset relations $r_1(A, B)$ and $r_2(C)$ are as follows:

$$r_1 = \{(1, a)(2, a)\}$$
 $r_2 = \{(2), (3), (3)\}$

- Then $\Pi_B(r_1)$ would be $\{(a), (a)\}$, while $\Pi_B(r_1) \times r_2$ would be $\{(a, 2), (a, 2), (a, 3), (a, 3), (a, 3), (a, 3)\}$
- SQL duplicate semantics:

select
$$A_1, A_2, \ldots, A_n$$

from r_1, r_2, \ldots, r_m
where P

is equivalent to the *multiset* version of the expression:

$$\Pi_{A_1,A_2,\ldots,A_n}(\sigma_P(r_1\times r_2\times\ldots\times r_m))$$



Module Summary

Module 09

Partha Pratio

Objective: Outline

> Operations Cartesian Produc

Rename AS Operation String Values Order By Clause Select Top / Fetch

Clause
Where Clause
Predicates
Duplicates

Module Summary

• Completed the understanding of basic query structure

Slides used in this presentation are borrowed from http://db-book.com/ with kind permission of the authors.

Edited and new slides are marked with "PPD".