introduction to SCRIPTING, DATABASES, SYSTEM ARCHITECTURE

SQL II: Joins, Indices, and Group-By



Claus Brabrand

(((brabrand@itu.dk)))

Associate Professor, Ph.D. (((Software and Systems))) IT University of Copenhagen

ITU Online Course Evaluation



Remember to fill in the ITU online course evaluation this week

(see email about this from study administration)

Claus Brabrand, ITU, Denmark

SCRIPTING, DATABASES, & SYSTEM ARCHITECTURE

[2] Nov 02, 2012

Agenda

More SQL commands

- Join (samkøring af data)
- Indices
- Group by
- Left and Right Join

SQL: Main Commands

Data definition:

CREATE TABLE

DROP TABLE

// creates a new table// deletes a table

Data manipulation:

INSERT INTO .. VALUES (..)

SELECT .. FROM .. WHERE

DELETE FROM .. WHERE

UPDATE .. SET .. WHERE

// insert record(s)

// retrieves information

// delete record(s)

// changes record(s)

SHOW TABLES & DESCRIBE

Show tables in database:

SHOW TABLES ;	Tables_in_my_database
	mailing_list
	phone_numbers
	students

Show information about a particular table:

DESCRIBE students ;	Field	Туре	Null	Key	Default	Extra
	name	varchar(80)	NO		NULL	
	age	int(11)	NO		NULL	

Quit:

OUIT

Claus Brabrand, ITU, Denmark

LIMIT and ORDER-BY

We can limit the number of records displayed:

Retrieve a hundred recipes:

SELECT * FROM recipes LIMIT 100 ;

Retrieve the youngest student:

SELECT * FROM students ORDER BY age LIMIT 1 ;

Retrieve three (alphabetically) last courses:

SELECT * FROM courses ORDER BY name DESC LIMIT 3 ;

COUNT

Given the table:		name	age
	1234	Anna	20
SELECT * FROM students ;	5678	Brian	25
	9999	Claire	23

COUNT (will count number of records):

SELECT COUNT(*) FROM students ;

COUNT AS (gives the count a name):

SELECT COUNT (*)	AS num_courses	FROM students	;	num courses
				3

COUNT(*)

3

MIN, MAX, SUM, AVG

	stud_id	name	age
	1234	Anna	20
1	5678	Brian	25
	9999	Claire	23

MIN (minimum):

SELECT MIN(age) FROM students ;



SELECT MAX(age) AS oldest FROM students ;

oldest 25

MIN(age)

20

SUM:

 SELECT SUM(age) FROM students ;
 SUM(age)

 68

AVG (average):

SELECT AVG(age) AS average FROM students ;

22.6667

average

Agenda

More SQL commands

Join (samkøring af data)

Indices

Group by

Left and Right Join

Claus Brabrand, ITU, Denmark

Join (samkøring af data)

Given the tables (entities):

_list:
email
brabrand@itu.dk
obama@hotmail.com
jdoe@notmail.com

phone_numbers:

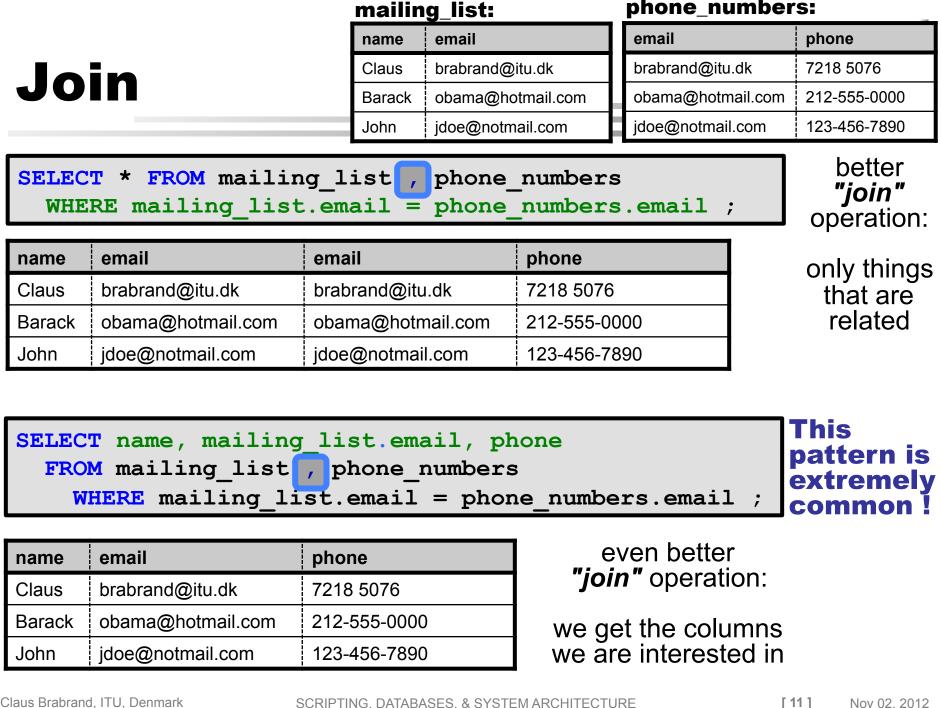
email	phone
brabrand@itu.dk	7218 5076
obama@hotmail.com	212-555-0000
jdoe@notmail.com	123-456-7890

<pre>SELECT * FROM mailing_list , phone_numbers ;</pre>			
name	email	email	phone
Claus	brabrand@itu.dk	brabrand@itu.dk	7218 5076
Barack	obama@hotmail.com	brabrand@itu.dk	7218 5076
John	jdoe@notmail.com	brabrand@itu.dk	7218 5076
Claus	brabrand@itu.dk	obama@hotmail.com	212-555-0000
Barack	obama@hotmail.com	obama@hotmail.com	212-555-0000
John	jdoe@notmail.com	obama@hotmail.com	212-555-0000
Claus	brabrand@itu.dk	jdoe@notmail.com	123-456-7890
Barack	obama@hotmail.com	jdoe@notmail.com	123-456-7890
John	jdoe@notmail.com	jdoe@notmail.com	123-456-7890

"join" operation:

gives all combos!

Nov 02, 2012



Claus Brabrand, ITU, Denmark

SCRIPTING, DATABASES, & SYSTEM ARCHITECTURE

Nov 02, 2012

EXERCISE: Join

Database:

suspects:

name	dna_profile
Jack the Ripper	ACTGC
Jason	ACGTC
Amagermanden	ACTTC

crimescenes:

place	item	dna_found
Living Room	Table	ACCTC
Living Room	Lamp	AGTGC
Hallway	Chainsaw	ACCTC
Back Alley	Knife	ACTGC
Bathroom	Shampoo	ACCTC
Kitchen	Milk Carton	ACTTC

Join to determine (possibly multiple) perpetrators?

SELECT name, place, item
FROM suspects, crimescenes
WHERE dna_profile = dna_found ;



name	place	item
Jack the Ripper	Back Alley	Knife
Amagermanden	Kitchen	Milk Carton

EXERCISE: Join (cont'd)

Database:

suspects:

name	dna_profile
Jack the Ripper	ACTGC
Jason	ACGTC
Amagermanden	ACTTC

crimescene:

place	item	dna_found
Living Room	Table	ACCTC
Living Room	Lamp	AGTGC
Hallway	Chainsaw	ACCTC
Back Alley	Knife	ACTGC
Bathroom	Shampoo	ACCTC
Kitchen	Milk Carton	ACTTC

...now also with "dna" in result:

SELECT name, pla	ace, item, dna_profile AS dna	
FROM suspects	, crimescene	
WHERE dna_profile = dna_found ;		



name	place	item	dna
Jack the Ripper	Back Alley	Knife	ACTGC
Amagermanden	Kitchen	Milk Carton	ACTTC

Claus Brabrand, ITU, Denmark

EXERCISE: Join

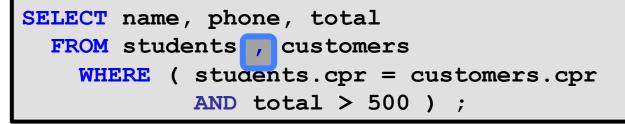
students:

cpr	name	phone	address
321085-1111	Anna	212-555-7755	Somewhere 4
221188-2222	Brian	212-555-5050	Anystreet 7
010190-3333	Claire	212-555-0707	Some Other Place 15
020290-4444	Danny	212-555-9999	Anywhere 1a

customers:

cpr	total
010190-3333	1,525.24
321085-1111	195.56
040587-5555	1,020,30
221188-2222	897.20

Find names and phone numbers of students who are "good customers" (spent more than 500 kr):



	name	phone	total
\rightarrow	Brian	212-555-5050	897.20
	Claire	212-555-0707	1,525.24

Agenda

More SQL commands

Join (samkøring af data)

Indices

Group byLeft and Right Join

Claus Brabrand, ITU, Denmark

How fast is a query?

A join of two tables can take a very long time!

every record in one table needs to be compared to every record in the other table

SELECT * FROM table1 , table2 WHERE table.id1 = table.id2 ;

If both tables have 10,000 records (not uncommon!), we get 10,000 x 10,000 = 100,000,000 comparisons

This can easily take 20 seconds !

However, if id1 and id2 have been indexed

- This only takes 0.05 seconds !
- (Note: the bigger the table, the more important with indexing)

Claus Brabrand, ITU, Denmark

Types of Indices (= Index'es)

Three types of indices:

type	needs to be unique?	is allowed to be null?
PRIMARY KEY	yes	no
UNIQUE	yes	yes
INDEX	no	yes

(Note: only one PRIMARY KEY allowed per table)

Example:

CREATE TABLE students (students:		
stud id INT PRIMARY KEY,		name	age	
name VARCHAR(50) NOT NULL,		Anna	20	
age INT NOT NULL,	5678	Brian	25	
)	9999	Claire	23	

When to use Indices

Indices are best used on columns that are...:

• ... frequently used in the **WHERE** part:

SELECT * FROM table WHERE age > 20 ;

• ... frequently used in the **ORDER BY** part:

SELECT * FROM table ORDER BY name ;

...used as part of joins:

Agenda

More SQL commands

- Join (samkøring af data)
- Indices

Group by

Left and Right Join

Claus Brabrand, ITU, Denmark

GROUP BY

Given table: CREATE TABLE INSERT INTO

	ех	pe	ns	es:	
--	----	----	----	-----	--

expense	dept	year	amount
salary	research	2001	490000
salary	sales	2002	1500000
salary	research	2002	500000
coffee	research	2003	800
coffee	sales	2003	300
salary	sales	2003	1600000
salary	research	2003	510000

Let's calculate total expenses per dept.:

SELECT dept, SUM(amount) FROM expenses GROUP BY dept ;

"GROUP BY" causes the records to be sorted and grouped wrt. their dept value:

expense	expense dept		amount
salary	research	2001	490000
salary	research	2002	500000
coffee	research	2003	800
salary	research	2003	510000
salary	sales	2002	1500000
coffee	sales	2003	300
salary	sales	2003	1600000

THEN the result is produced which involves calculating the 'SUM(amount)' values:

dept	SUM(amount)
research	1500800
sales	3100300

0000 G, DATABASES, & SYSTEM ARCHITECTURE

							expens	<u>53.</u>		
							expense	dept	year	amount
G	RO	ID	RV	7			salary	research	2001	490000
V							salary	sales	2002	1500000
					salary	research	2002	500000		
-	The s	um	ofex	nens	ses		coffee	research	2003	800
-	aroun			non	60		coffee	sales	2003	300
Ĺ	group			hell	うて		salary	sales	2003	1600000
	and d	lepa	artme	ent:			salary	research	2003	510000
SELECT expense, dept, SUM(amount) AS to GROUP BY expense, dept ;The table is first grouped by expense:THEN, the table is sub-grouped by de						AS tc	otal FRO	M expe	enses	
G The t	ROUP BY	expe irst	ense, d	ept; THEN	, the tab	le is		M expe	enses	
G The t	ROUP BY	expe irst	ense, d	ept; THEN	, the tab	le is	pt.:	FINALL	Y, the r	result
G The t grou	ROUP BY able is f bed by e	expe irst xpen	ense, d SC:	ept ; THEN <u>sub-g</u>	, the tab rouped	le is by de	pt.:		Y, the r	result
G The t grou exp.	ROUP BY able is f bed by e dept	expe irst xpen year	ense, de S <i>E:</i> amount	ept ; THEN <u>sub-g</u> exp.	, the tab rouped dept	le is by de year	p <i>t.:</i> amount	- FINALL is produ	Y, the r	result total
G The t grou exp. coffee	ROUP BY	experimental exper	se: amount 800	ept ; THEN sub-g exp. coffee coffee	, the tab rouped dept research sales	<i>le is</i> <i>by de</i> year 2003 2003	pt.: amount 800 300	FINALL is produ	Y, the r uced:	1
G The t grou exp. coffee coffee	ROUP BY able is f bed by e dept research sales	experies for star star star star star star star sta	ense, d Se: amount 800 300	ept ; THEN sub-g exp. Coffee coffee salary	, the tab rouped dept research sales research	le is by de year 2003 2003	Dt.: amount 800 300 490000	FINALL is produ	Y, the r uced: dept research	total
G The t grou exp. coffee coffee salary	ROUP BY	expe irst xpen 2003 2003 2001	ense, de SE: amount 800 300 490000	ept ; THEN sub-g exp. Coffee Coffee salary salary	, the tab rouped dept research sales research research	le is by de year 2003 2003 2001 2002	Dt.: amount 800 300 490000 500000	FINALL is produ exp. c coffee r	Y, the r uced: dept research sales	total 800
G The f grou exp. coffee coffee salary salary	ROUP BY able is find oed by e dept research sales research sales	expe irst xpen 2003 2003 2001 2002	ense, de SE: amount 800 300 490000 1500000	ept ; THEN sub-g exp. Coffee coffee salary	, the tab rouped dept research sales research	le is by de year 2003 2003	Dt.: amount 800 300 490000	FINALL is produ exp. coffee r coffee s salary r	Y, the r uced: dept research sales	total 800 300

Claus Brabrand, ITU, Denmark

							<u>expens</u>	es:		
							expense	dept	year	amount
C	POI	ID	BV	7			salary	research	2001	490000
G	GROUP BY				salary	sales	2002	1500000		
							salary	research	2002	500000
-	The s	ıım	ofex	nens	Ses		coffee	research	2003	800
-	aroun	od		non			coffee	sales	2003	300
Ĺ	group		Dy ex		5E		salary	sales	2003	1600000
ć	and d	lepa	artme	ent 2	002-0	J3 :	salary	research	2003	510000
SEL		WHERE 2002 <= year AND year <= 2003 GROUP BY expense, dept ;								
The t	able is f	irst	_	THEN	, the tab	le is		exper	nse, d	ept ;
The t	able is f	irst	_	THEN	, the tab	le is	p <i>t.:</i> amount	- FINALL	.Y, the r	-
The t grou	able is f bed by e	irst xpen	se:	THEN sub-g	, the tab rouped	le is by de	p <i>t.:</i> amount	_	.Y, the r	-
The t group exp.	able is f bed by e dept	irst xpen year	S <i>C:</i> amount	THEN sub-g	, the tab rouped dept	le is by de year	p <i>t.:</i> amount	- FINALL is prod	.Y, the r	-
The t group exp. coffee	able is fi bed by e dept research	irst xpen year 2003	Se: amount 800	THEN sub-g exp. coffee coffee	, the tab rouped i dept research sales	le is by de year 2003	pt.: amount 800 300	FINALL is prod exp.	.Y, the r uced:	result
The t group exp. coffee coffee	able is f bed by e dept research sales	irst xpen year 2003 2003	SE: amount 800 300	exp. coffee salary	, the tab rouped dept research sales	le is by de year 2003 2003	Dt.: amount 800 300	FINALL is prod exp.	.Y, the r uced: dept	result
The t group exp. coffee coffee	able is fi bed by e dept research sales research	irst year 2003 2003 2001	Se: amount 800 300 490000	THEN sub-g exp. coffee coffee salary salary	, the tab rouped i dept research sales research research	le is by de year 2003 2003 2001 2002	pt.: amount 800 300 490000 500000	FINALL is prod exp. (coffee (coffee (.Y, the r uced: dept research	result total 800
The t group exp. coffee coffee salary salary	able is fi bed by e dept research sales research sales	irst year 2003 2003 2001 2002	Se: amount 800 300 400000 1500000	exp. coffee salary	, the tab rouped dept research sales	le is by de year 2003 2003	Dt.: amount 800 300	FINALL is prod exp. (coffee (salary (. <i>Y, the r</i> uced: dept research sales	total 800 300

Claus Brabrand, ITU, Denmark

	<u>expens</u>	<u>es:</u>		
	expense	dept	year	amount
EXERCISE 1	salary	research	2001	490000
EXERCISE	salary	sales	2002	1500000
	salary	research	2002	500000
- M/hot is the total amount	coffee	research	2003	800
What is the total amount	coffee	sales	2003	300
spent per expense?	salary	sales	2003	1600000
	salary	research	2003	510000

SELECT expense, SUM(amount) FROM expenses GROUP BY expense ;

The table is grouped by expense:

exp.	dept	year	amount
coffee	research	2003	800
coffee	sales	2003	300
salary	research	2001	490000
salary	sales	2002	1500000
salary	research	2002	500000
salary	sales	2003	1600000
salary	research	2003	510000

THEN the result is produced which involves calculating the 'SUM(amount)' values:

expense	SUM(amount)
coffee	1100
salary	4600000

		<u>expens</u>	es:		
		expense	dept	year	amount
EXERCISE 2		salary	research	2001	490000
LALAUJL Z	Н	salary	sales	2002	1500000
	Я	salary	research	2002	500000
- Mhot is the average		coffee	research	2003	800
What is the average		coffee	sales	2003	300
amount per expense?		salary	sales	2003	1600000
		salary	research	2003	510000
	-				

SELECT expense, AVG (amount) FROM expenses GROUP BY expense ;

The table is grouped

by expense:

exp.	dept	year	amount
coffee	research	2003	800
coffee	sales	2003	300
salary	research	2001	490000
salary	sales	2002	1500000
salary	research	2002	500000
salary	sales	2003	1600000
salary	research	2003	510000

THEN the result is produced which involves calculating the 'AVG(amount)' values:

expense	AVG(amount)
coffeee	550
salary	920000

	<u>expens</u>	ses:		
	expense	dept	year	amount
EXERCISE 3	salary	research	2001	490000
EXENCISE J	salary	sales	2002	1500000
	salary	research	2002	500000
- M/hat are the tatal	coffee	research	2003	800
What are the total	coffee	sales	2003	300
expenses each year?	salary	sales	2003	1600000
	salary	research	2003	510000
SELECT year, SUM(amount) FROM expenses	GROUP	BY year	;	

The table is grouped by year:

expense	dept	year	amount
salary	research	2001	490000
salary	sales	2002	1500000
salary	research	2002	500000
coffee	research	2003	800
coffee	sales	2003	300
salary	sales	2003	1600000
salary	research	2003	510000

THEN the result is produced which involves calculating the 'SUM(amount)' values:

year	SUM(amount)
2001	490000
2002	2000000
2003	2111100

expenses:						
	expense	dept	year	amount		
EXERCISE 4	salary	research	2001	490000		
	salary	sales	2002	1500000		
	salary	research	2002	500000		
- M/bat are the average	coffee	research	2003	800		
What are the expenses	coffee	sales	2003	300		
per year per department?	salary	sales	2003	1600000		
	salary	research	2003	510000		
SELECT year, dept, SUM(amount) AS total FROM expenses						

GROUP BY year, dept ;

The table is first grouped by year:

expense	dept	year	amount
salary	research	2001	490000
salary	sales	2002	1500000
salary	research	2002	500000
coffee	research	2003	800
coffee	sales	2003	300
salary	sales	2003	1600000
salary	research	2003	510000

THEN, subgrouped by department

FINALLY, result...:

year	dept	total
2001	research	490000
2002	sales	1500000
2002	research	500000
2003	research	510800
2003	sales	1600300

Claus Brabrand, ITU, Denmark

	<u>expens</u>	<u>es:</u>	-	
	expense	dept	year	amount
EXERCISE 5	salary	research	2001	490000
EXENCISE J	salary	sales	2002	1500000
	salary	research	2002	500000
What are the expenses each year in 2002-03?	coffee	research	2003	800
	coffee	sales	2003	300
	salary	sales	2003	1600000
<i>J J J J J J J J J J</i>	salary	research	2003	510000
SELECT year, SUM(amount) FROM expenses				

WHERE 2002 <= year AND year <= 2003 GROUP BY year ;

The table is grouped by year:

expense	dept	year	amount
salary	research	2001	490000
salary	sales	2002	1500000
salary	research	2002	500000
coffee	research	2003	800
coffee	sales	2003	300
salary	sales	2003	1600000
salary	research	2003	510000

Result...:

year	SUM(amount)		
2002	2000000		
2003	2111100		

		<u>expens</u>	es:	-	
		expense	dept	year	amount
EXERCISE 6		salary	research	2001	490000
		salary	sales	2002	1500000
		salary	research	2002	500000
\//batia tha biabaat		coffee	research	2003	800
What is the <i>highest</i>		coffee	sales	2003	300
expenditure for each dept?	2	salary	sales	2003	1600000
		salary	research	2003	510000
<pre>SELECT dept, MAX(amount) FROM expenses GROUP BY dept ;</pre>					

The table is grouped by dept:

expense	dept	year	amount
salary	research	2001	490000
salary	research	2002	500000
coffee	research	2003	800
salary	research	2003	510000
salary	sales	2002	1500000
coffee	sales	2003	300
salary	sales	2003	1600000

Result...:

dept	MAX(amount)
research	510000
sales	1600000

EXERCISE 7	
------------	--

What is the *highest exp.*

for each dept per year?

expenses:

_	CAPCII3C3						
	expense dept		year	amount			
	salary	research	2001	490000			
ļ	salary	sales	2002	1500000			
J	salary	research	2002	500000			
	coffee	research	2003	800			
	coffee	sales	2003	300			
	salary	sales	2003	1600000			
	salary	research	2003	510000			

SELECT dept, year, MAX(amount) FROM expenses
GROUP BY dept, year ;

The table is first grouped by dept:

expense	dept	year	amount
salary	research	2001	490000
salary	research	2002	500000
coffee	research	2003	800
salary	research	2003	510000
salary	sales	2002	1500000
coffee	sales	2003	300
salary	sales	2003	1600000

THEN, subgrouped by year

Result...:

dept	year	MAX(amount)
research	2001	490000
research	2002	500000
research	2003	510000
sales	2002	1500000
sales	2003	1600000

Claus Brabrand, ITU, Denmark

HAVING

9	expenses:					
	expense	dept	year	amount		
	salary	research	2001	490000		
	salary	sales	2002	1500000		
	salary	research	2002	500000		
	coffee	research	2003	800		
	coffee	sales	2003	300		
	salary	sales	2003	1600000		
	salary	research	2003	510000		

What is the highest exp. for each dept per year?

SELECT dept, year, MAX(amount) FROM expenses
GROUP BY dept, year HAVING MAX(amount) > 500000 ;

The table is first grouped by dept:

expense	dept	year	amount
salary	research	2001	490000
salary	research	2002	500000
coffee	research	2003	800
salary	research	2003	510000
salary	sales	2002	1500000
coffee	sales	2003	300
salary	sales	2003	1600000

THEN, subgrouped by year

Re	C		1		
110		И	L .		

dept	year	MAX(amount)
-research	2001	490000
research	2002	500000
research	2003	510000
sales	2002	1500000
sales	2003	1600000

Claus Brabrand, ITU, Denmark

Abbreviations (aliases)

The query...:

SELECT students.id, students.name
FROM students, courses, enrollment
WHERE students.id = enrollment.id
AND enrollment.id = courses.id ;

....can be abbeviated to (via aliases)...:

SELECT s.id, s.name
FROM students AS s, courses AS c, enrollment AS e
WHERE s.id = e.id
AND e.id = c.id ;

Claus Brabrand, ITU, Denmark

Agenda

More SQL commands

- Join (samkøring af data)
- Indices
- Group by

Left and Right Join

Recall "Join"

Database:

suspects:

name	dna_profile	
Jack the Ripper	ACTGC	
Jason	ACGTC	
Amagermanden	ACTTC	

crimescenes:

place	item	dna_found
Living Room	Table	ACCTC
Living Room	Lamp	AGTGC
Hallway	Chainsaw	ACCTC
Back Alley	Knife	ACTGC
Bathroom	Shampoo	ACCTC
Kitchen	Milk Carton	ACTTC

Join to determine (possibly multiple) perpetrators?

SELECT name, place, item
FROM suspects, crimescenes
WHERE dna profile = dna found ;



name	place	item		
Jack the Ripper	Back Alley	Knife		
Amagermanden	Kitchen	Milk Carton		

Claus Brabrand, ITU, Denmark

One problem with joins

Database:

suspects:

name	dna_profile
Jack the Ripper	ACTGC
Jason	ACGTC
Amagermanden	ACTTC

crimescenes:

place	item	dna_found
Living Room	Table	ACCTC
Living Room	Lamp	AGTGC
Hallway	Chainsaw	ACCTC
Back Alley	Knife	ACTGC
Bathroom	Shampoo	ACCTC
Kitchen	Milk Carton	ACTTC

Suppose we instead would like to generate:

That is always constants a reast	place	item	name (if match)
That is, <u>always</u> generate a record from the right table (not just when we happen to have a match)	Living Room	Table	NULL
we happen to have a match)	Living Room	Lamp	NULL
SELECT place, item, name FROM	Hallway	Chainsaw	NULL
	Back Alley	Knife	Jack the Ripper
<pre>suspects RIGHT JOIN crimescenes ON dna profile = dna found ;</pre>	Bathroom	Shampoo	NULL
	Kitchen	Milk Carton	Amagermanden

Claus Brabrand, ITU, Denmark

SCRIPTING, DATABASES, & SYSTEM ARCHITECTURE

[34] Nov 02, 2012

LEFT JOIN vs. RIGHT JOIN

Database:

suspects:

name	dna_profile		
Jack the Ripper	ACTGC		
Jason	ACGTC		
Amagermanden	ACTTC		

crimescenes:

place	item	dna_found
Living Room	Table	ACCTC
Living Room	Lamp	AGTGC
Hallway	Chainsaw	ACCTC
Back Alley	Knife	ACTGC
Bathroom	Shampoo	ACCTC
Kitchen	Milk Carton	ACTTC

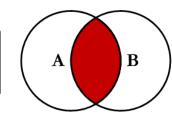
Same thing for LEFT JOIN (just swap arguments):

SELECT place, item, name FROM crimescenes LEFT JOIN suspects	place	item	name (if match)
ON suspects.dna_profile =	Living Room	Table	NULL
<pre>crimescene.dna_found ;</pre>	Living Room	Lamp	NULL
	Hallway	Chainsaw	NULL
<pre>SELECT place, item, name FROM suspects <u>RIGHT</u> JOIN crimescenes ON suspects.dna profile =</pre>	Back Alley	Knife	Jack the Ripper
	Bathroom	Shampoo	NULL
crimescene.dna_found ;	Kitchen	Milk Carton	Amagermanden
Claus Brabrand, ITU, Denmark SCRIPTING, DATABASE	BASES, & SYSTEM ARCHITECTURE		[35] Nov 02, 2012

	a:		b:	
	name	id	id	course
Join (normal, left, right)	Anna	1	1	DSDS
	Brian	2	2	GSD
	Claire	3	4	IWJX

Normal join:

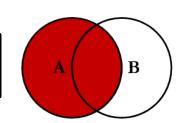
SELECT name, a.id, course
FROM a , b WHERE a.id = b.id ;



name	id	course
Anna	1	DSDS
Brian	2	GSD

Left join:

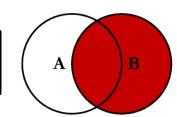
SELECT name, a.id, course
FROM a LEFT JOIN b ON a.id = b.id ;



name	id	course
Anna	1	DSDS
Brian	2	GSD
Claire	3	NULL

Right join: SELECT name, b.id, course

FROM a <u>RIGHT</u> JOIN b ON a.id = b.id ;



name	id	course
Anna	1	DSDS
Brian	2	GSD
NULL	4	IWJX

Assignment 7

Exercise 7.1:

Create and fill in a course database

Exercise 7.2:

Query the database using increasingly complex queries (7.2.1 + 7.2.2 + 7.2.3 + 7.2.4)

Exercise 7.3:

Answer five questions on SQL

Any Questions?

(Have a nice weekend)

Claus Brabrand, ITU, Denmark

SCRIPTING, DATABASES, & SYSTEM ARCHITECTURE

Nov 02, 2012