Databases - 4

Other relational operations and DDL

How to write RA expressions for dummies

Step 1: Identify the relations required and CP them together

Step 2: Add required selections to make the CP into an appropriate Join

Step 3: Add any other selections required for the query

Step 4: Add appropriate projections to get the required attributes for the query

---

RA example

Show the name, job and location for all staff with a salary greater than £25,000

Step 1: Identify the relations required and CP them together

Requires 2 tables EMP and DEPT

$$emp \times dept$$

RA example

Show the name, job and location for all staff with a salary greater than £25,000

Step 2: Add required selections to make the CP into an appropriate Join

$$\sigma_{predicate} (emp \times dept)$$

$$\sigma_{emp.deptno=dept.deptno} (emp \times dept)$$

RA example

Show the name, job and location for all staff with a salary greater than £25,000

Step 3: Add any other selections required for the query

$$\sigma_{emp.deptno=dept.deptno} (emp \times dept)$$

$$\sigma_{sal>25000} (\sigma_{emp.deptno=dept.deptno} (emp \times dept))$$

Note

Alternatives here: could simply use an AND

$$\sigma_{sal>25000} (\sigma_{emp.deptno=dept.deptno} (emp \times dept))$$

Would be the same as

$$\sigma_{(emp.deptno=dept.deptno) AND sal>25000} (emp \times dept)$$

RA example

Show the **name, job** and **location** for all staff with a salary greater than £25,000

Step 4: Add appropriate projections to get the required attributes for the query

$$\sigma_{sal>25000} (\sigma_{emp.deptno=dept.deptno} (emp \times dept))$$

$$\pi_{ename, job, loc} (\sigma_{sal>25000} (\sigma_{emp.deptno=dept.deptno} (emp \times dept)))$$

Important

Watch out for projections BEFORE selections - check they still work

$$\sigma_{sal>25000} (\pi_{ename, job, loc} (\sigma_{emp.deptno=dept.deptno} (emp \times dept)))$$

This produces an empty set (or an error result). Why?

How to write SQL expressions for dummies

Step 1: Identify the tables required and CP them together

Step 2: Add required conditions to make the CP into an appropriate Join

Step 3: Add any other conditions required for the query

Step 4: Add appropriate projections to get the required columns for the query

SQL example

Show the name, job and location for all staff with a salary greater than £25,000

Step 1: Identify the tables required and CP them together

Requires 2 tables EMP and DEPT

```
select * or expression          projections here
from emp, dept                   CPs here
[where expression]              selections here
```

SQL example

Show the name, job and location for all staff with a salary greater than £25,000

Step 2: Add required conditions to make the CP into an appropriate Join

```
select * or expression              projections here
from emp, dept                       CPs here
where emp.deptno=dept.deptno   selections here
```

SQL example

Show the name, job and location for all staff with a salary greater than £25,000

Step 3: Add any other conditions required for the query

```
select * or expression              projections here
from emp, dept                       CPs here
where emp.deptno=dept.deptno   selections here
    and sal > 25000
```

SQL example

Show the name, job and location for all staff with a salary greater than £25,000

Step 4: Add appropriate projections to get the required columns for the query

```
select ename, job, loc           projections here
from emp, dept                    CPs here
where emp.deptno=dept.deptno      selections here
    and sal > 25000
```

Relational Algebra operations

Selection	σ
Projection	π
Cartesian Product	×
Union	∪
Set Difference	-
Join	⋈
Intersection	∩
Division	÷

Find all values

UNION

R ∪ S

A set, every member of which is an element of one or another of two or more given sets.

All the distinct values from the 'first' set along with all the distinct values in the 'second' set

UNION

$R \cup S$
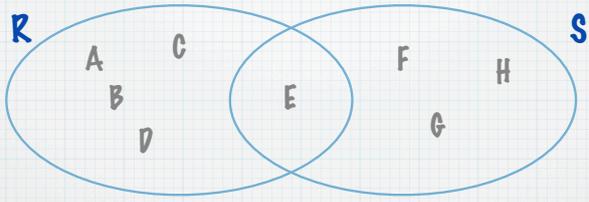
$R \cup S = \{ A, B, C, D, E, F, G, H \}$

RA example

Create a list of all the hisals and losals

list of all the losals

$$\pi_{losals} (grade)$$

losal
17000
22000
24000
30000
50000

list of all the hisals

$$\pi_{hisals} (grade)$$

hisal
21999
23999
29999
49999
99999

RA example

Create a list of all the hisals and losals

$$\pi_{losals} (grade)$$
$$\cup$$
$$\pi_{hisals} (grade)$$

21999
23999
29999
49999
99999
17000
22000
24000
30000
50000

SQL example

Create a list of all the **hisals and losals**

select hisal from grade

union

select losal from grade

21999
23999
29999
49999
99999
17000
22000
24000
30000
50000

Relational Algebra operations

Selection	σ
Projection	π
Cartesian Product	×
Union	∪
Set Difference	-
Join	⋈
Intersection	∩
Division	÷

Find values in one set but not in another

SET DIFFERENCE

R - S

A set, every member of which is in one particular set but not another

All the distinct values from the 'first' set without all the distinct values in the 'second' set

SET DIFFERENCE

R - S
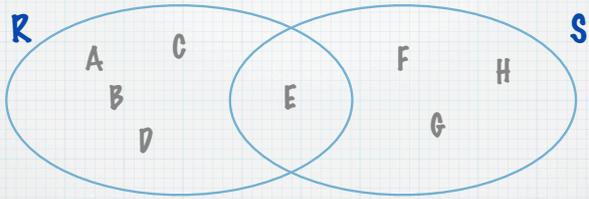
R - S = { A, B, C, D }

RA example

Create a list of actual salaries excluding any that are earning the highest value for the grade

list of all the **hisals**

π_{hisals} (grade)

hisal
21999
23999
29999
49999
99999

RA example

Create a list of actual salaries excluding any that are earning the highest value for the grade

list of all the **salaries**

π_{sal} (emp)

SAL
18000
26000
22500
39750
22500
38500
34500
40000
60000
25000
21000
19500
40000
23000

RA example

Create a list of actual salaries excluding any that are earning the highest value for the grade

$$\pi_{sal}(emp) - \pi_{hisals}(grade)$$

Done using the SQL keyword minus - unfortunately not many PC databases support this syntax

---

Relational Algebra operations

Selection	σ
Projection	π
Cartesian Product	×
Union	∪
Set Difference	-
Join	⋈
Intersection	∩
Division	÷

Done using CP and selection

Not really implemented in databases as can be done by selection

Not really implemented in databases as hard to implement

Data Query Language (DQL)

The SQL shown so far is for writing queries - the DQL part of the language

    select * or expression
    from relations
    [where expression]

DDL and DML

SQL also has a syntax for creating tables, altering the structure of tables and deleting tables - called the Data Definition Language (DDL)

SQL also has a syntax for inserting rows, updating and deleting rows - called the Data Manipulation Language (DML)

Data Definition Language (DDL)

Creating a table

```
CREATE TABLE tablename
    (column_name    type [NULL/NOT NULL],
     column_name    type [NULL/NOT NULL],
     ..)
```

Data Definition Language (DDL)

type(s)

|	
CHAR (size)	Character data, maximum of 'size' characters upto 240
DATE	Dates (which include time)
LONG	Character data up to 65535 (some restrictions may apply on the use of this field in a select statement)
NUMBER	Maximum of 40 digits (will accept scientific notation)

Data Definition Language (DDL)

Creating a table

```
CREATE TABLE tablename
    (column_name   type [NULL/NOT NULL],
     column_name   type [NULL/NOT NULL],
      ..)
```

```
create table myemp
    (ename  char,
     sal number,
     deptno number)
```

☐ myemp			_ ☐ X
ename ▾	sal ▾	deptno ▾	
*			

Record: ◄ ◄ 1 of 1 ► ►I ►⋈ ☒ No Filter | Search

Data Manipulation Language (DML)

Inserting a record

```
insert into table [(columnname, columnname, ...)]
    values (value, value,...)
```

```
insert into table
    values (value, value,...)
```

Data Manipulation Language (DML)

Inserting a record

```
insert into myemp (ename, sal, deptno)
values ("smith", 25000, 3)
```

```
insert into myemp
values ("jones", 24000, 2)
```

☐ myemp			_ ☐ X
ename ▾	sal ▾	deptno ▾	
Smith	25000	3	
Jones	24000	2	
*			

Record: ◄ ◄ 3 of 3 ► ►I ►⋈ ☒ No Filter | Search
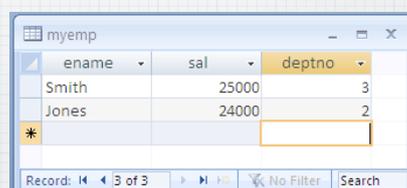
Data Manipulation Language (DML)

Inserting records from other tables

```
insert into table [(columnname, columnname, ...)]
    select  expression
```

Data Manipulation Language (DML)

Copies records from another table

```
insert into myemp (ename,
sal, deptno)
select ename,sal,deptno
from emp
```
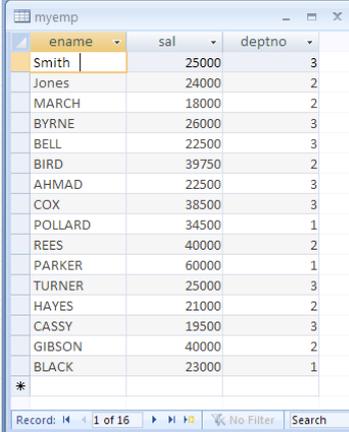
ename	sal	deptno
Smith	25000	3
Jones	24000	2
MARCH	18000	2
BYRNE	26000	3
BELL	22500	3
BIRD	39750	2
AHMAD	22500	3
COX	38500	3
POLLARD	34500	1
REES	40000	2
PARKER	60000	1
TURNER	25000	3
HAYES	21000	2
CASSY	19500	3
GIBSON	40000	2
BLACK	23000	1

Record: 1 of 16   No Filter   Search

Data Manipulation Language (DML)

Updating record(s)

```
update tablename set name = value [(,name=value)]
    [where expression]
```

Data Manipulation Language (DML)

Updating record(s)

update myemp set deptno = 4
where ename = "Cox"
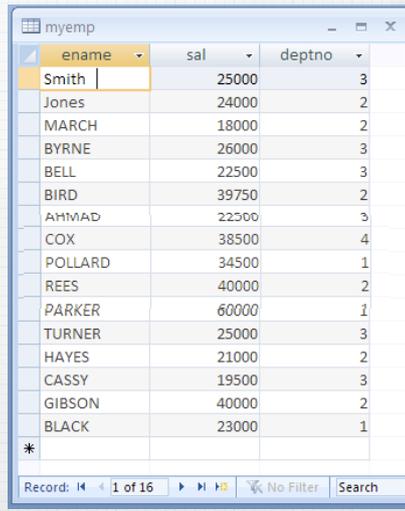
ename	sal	deptno
Smith	25000	3
Jones	24000	2
MARCH	18000	2
BYRNE	26000	3
BELL	22500	3
BIRD	39750	2
AHMAD	22500	3
COX	38500	4
POLLARD	34500	1
REES	40000	2
PARKER	60000	1
TURNER	25000	3
HAYES	21000	2
CASSY	19500	3
GIBSON	40000	2
BLACK	23000	1

Record: 1 of 16   No Filter   Search

Data Manipulation Language (DML)

Updating record(s)

update myemp set sal = sal * 1.1
where ename = "Pollard"
or ename = "Rees"
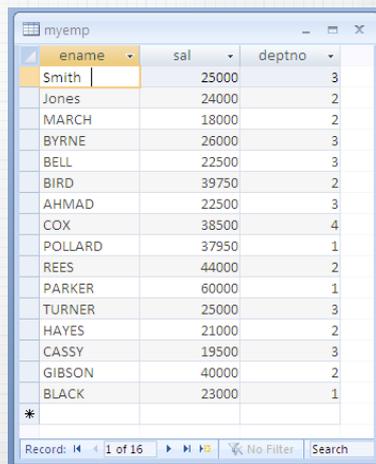
ename	sal	deptno
Smith	25000	3
Jones	24000	2
MARCH	18000	2
BYRNE	26000	3
BELL	22500	3
BIRD	39750	2
AHMAD	22500	3
COX	38500	4
POLLARD	37950	1
REES	44000	2
PARKER	60000	1
TURNER	25000	3
HAYES	21000	2
CASSY	19500	3
GIBSON	40000	2
BLACK	23000	1

Record: 1 of 16   No Filter   Search

Data Manipulation Language (DML)

Deleting record(s)

delete * from tablename
[where expression]

Data Manipulation Language (DML)

Deleting Pollard

```
delete * from myemp
where ename = "Cox"
```
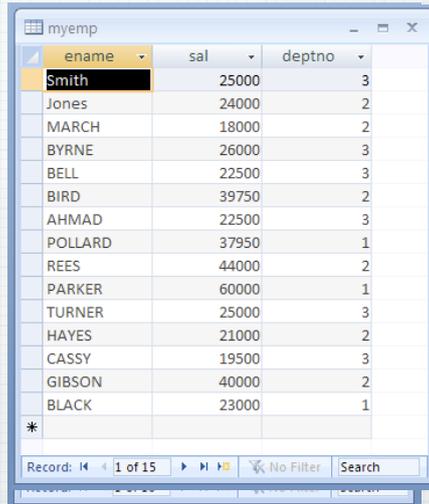
ename	sal	deptno
Smith	25000	3
Jones	24000	2
MARCH	18000	2
BYRNE	26000	3
BELL	22500	3
BIRD	39750	2
AHMAD	22500	3
POLLARD	37950	1
REES	44000	2
PARKER	60000	1
TURNER	25000	3
HAYES	21000	2
CASSY	19500	3
GIBSON	40000	2
BLACK	23000	1

Record: 1 of 15   No Filter   Search

Data Manipulation Language (DML)

Watch out:
Deleting everything

```
delete * from myemp
```

ename	sal	deptno

Record: 1 of 1   No Filter   Search

Data Definition Language (DDL)

Deleting a table - we say "dropping a table"

```
DROP TABLE tablename
```

```
drop table myemp
```

ename	sal	deptno

Record: 1 of 1   No Filter   Search