

# Databases - 3

## Null, Cartesian Product and Join

### Null

Null is a value that we use when

Something will never have a value

Something will have a value in the future

Something had a value but doesn't at the moment

### Null

Null is a reserved word in SQL and has special meaning

It is NOT the same as zero...

...consider the COMM attribute in the emp relation

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
405	MARCH	ADMIN	938	13/06/1997	18000		2
535	BYRNE	SALES	734	15/08/1997	26000	300	3
557	BELL	SALES	734	26/03/2000	22500	500	3
602	BIRD	MANAGER	875	31/10/1997	39750		2
690	AHMAD	SALES	734	05/12/1997	22500	1400	3

## Null

Access uses a blank cell to indicate a NULL

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
405	MARCH	ADMIN	938	13/06/1997	18000		2
535	BYRNE	SALES	734	15/08/1997	26000	300	3
557	BELL	SALES	734	26/03/2000	22500	500	3
602	BIRD	MANAGER	875	31/10/1997	39750		2
690	AHMAD	SALES	734	05/12/1997	22500	1400	3
734	COX	MANAGER	875	11/06/2002	38500		3
818	POLLARD	MANAGER	875	14/05/2000	34500		1
824	REES	ANALYST	602	05/03/2000	40000		2
875	PARKER	PRESIDENT		09/07/2002	60000		1
880	TURNER	SALES	734	04/06/2001	25000	0	
912	HAYES	ADMIN	824	04/06/2001	21000		2
936	CASSY	ADMIN	734	23/07/2002	19500		3
938	GIBSON	ANALYST	602	05/12/1997	40000		2
970	BLACK	ADMIN	818	21/11/1997	23000		1

Note how **Turner** is allowed to earn commission, but hasn't yet

## Null

Problems arise when we try to perform mathematical operations on NULL in SQL

Find the total earnings (salary + commission) for staff

```
select ename, sal, comm, sal+comm  
from emp
```

table name goes here

projection expressions go here

## Null

Find the total earnings (salary + commission) for staff

```
select ename, sal, comm, sal+comm  
from emp
```

Problem:

any value + null returns null  
18000 + null returns null

ename	sal	comm	Expr1003
MARCH	18000		
BYRNE	26000	300	26300
BELL	22500	500	23000
BIRD	39750		
AHMAD	22500	1400	23900
COX	38500		
POLLARD	34500		
REES	40000		
PARKER	60000		
TURNER	25000	0	25000
HAYES	21000		
CASSY	19500		
GIBSON	40000		
BLACK	23000		

## Null

Solution: `nz(attribute)` function

Returns either **zero** or a **value** from the corresponding attribute

Use this on an attribute that may contain nulls before performing calculations

## Null

Solution: `nz(attribute)` function

`nz(null)`  
→ 0

Returns **zero**, which we can then use in a calculation

`nz(14)`  
→ 14

Returns **14**, which we can then use in a calculation

`nz(comm)`  
→ **comm values**

Returns **comm** values, replacing **nulls** with **0**

## Null

Find the total earnings (salary + commission) for staff

```
select ename, sal, comm, sal+nz(comm)
from emp
```

ename	sal	comm	Expr1003
MARCH	18000		18000
BYRNE	26000	300	26300
BELL	22500	500	23000
BIRD	39750		39750
AHMAD	22500	1400	23900
COX	38500		38500
POLLARD	34500		34500
REES	40000		40000
PARKER	60000		60000
TURNER	25000	0	25000
HAYES	21000		21000
CASSY	19500		19500
GIBSON	40000		40000
BLACK	23000		23000

## PROJECTION

Gives us particular columns of interest

## SELECTION

Gives us particular rows of interest

problem ...

What if the information required in a query is in more than one relation/table?

List all staff along with the location where they work

Requires information in two relations tables to be combined

emp table

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
405	MARCH	ADMIN	938	13/06/1997	18000		2
535	BYRNE	SALES	734	15/08/1997	26000	300	3
557	BELL	SALES	734	26/03/2000	22500	500	3
602	BIRD	MANAGER	875	31/10/1997	39750		2
690	AHMAD	SALES	734	05/12/1997	22500	1400	3
734	COX	MANAGER	875	11/06/2002	38500		3
818	POLLARD	MANAGER	875	14/05/2000	34500		1
824	REES	ANALYST	602	05/03/2000	40000		2
875	PARKER	PRESIDENT		09/07/2002	60000		1
880	TURNER	SALES	734	04/06/2001	25000	0	3
912	HAYES	ADMIN	824	04/06/2001	21000		2
936	CASSY	ADMIN	734	23/07/2002	19500		3
938	GIBSON	ANALYST	602	05/03/2000	40000		2
970	BLACK	ADMIN	818				1

dept table

DEPTNO	DNAME	LOC
1	ACCOUNTING	LONDON
2	RESEARCH	YORK
3	SALES	BIRMINGHAM
4	OPERATIONS	LEEDS



## Relational Algebra operations

Selection	$\sigma$
Projection	$\pi$
Cartesian Product	$\times$
Union	$\cup$
Set Difference	$-$
Join	$\bowtie$
Intersection	$\cap$
Division	$\div$

Combine relations

## CARTESIAN PRODUCT

$R \times S$

The Cartesian product operation defines a relation that is the concatenation of every tuple of relation  $R$  with every tuple of relation  $S$

Everything in the 'first' table is combined with everything in the 'second' table

## CARTESIAN PRODUCT example

Combine the **emp** relation with the **dept** relation

**emp** $\times$ **dept**

The first relation  
here

The second  
relation here

.. but how many tuples in the result?

## CARTESIAN PRODUCT example

$emp \times dept$  not really what we were looking for ...

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	emp.DEPTN	dept.DEPTN	DNAME	LOC
405	MARCH	ADMIN	938	13/06/1997	18000			2	1 ACCOUNTING	LONDON
405	MARCH	ADMIN	938	13/06/1997	18000			2	2 RESEARCH	YORK
405	MARCH	ADMIN	938	13/06/1997	18000			2	3 SALES	BIRMINGHAM
405	MARCH	ADMIN	938	13/06/1997	18000			2	4 OPERATIONS	LEEDS
535	BYRNE	SALES	734	15/08/1997	26000	300		3	1 ACCOUNTING	LONDON
535	BYRNE	SALES	734	15/08/1997	26000	300		3	2 RESEARCH	YORK
535	BYRNE	SALES	734	15/08/1997	26000	300		3	3 SALES	BIRMINGHAM
535	BYRNE	SALES	734	15/08/1997	26000	300		3	4 OPERATIONS	LEEDS
557	BELL	SALES	734	26/03/2000	22500	500		3	1 ACCOUNTING	LONDON
557	BELL	SALES	734	26/03/2000	22500	500		3	2 RESEARCH	YORK
557	BELL	SALES	734	26/03/2000	22500	500		3	3 SALES	BIRMINGHAM
557	BELL	SALES	734	26/03/2000	22500	500		3	4 OPERATIONS	LEEDS
602	BIRD	MANAGER	875	31/10/1997	39750			2	1 ACCOUNTING	LONDON
602	BIRD	MANAGER	875	31/10/1997	39750			2	2 RESEARCH	YORK
602	BIRD	MANAGER	875	31/10/1997	39750			2	3 SALES	BIRMINGHAM
602	BIRD	MANAGER	875	31/10/1997	39750			2	4 OPERATIONS	LEEDS
690	AHMAD	SALES	734	05/12/1997	22500	1400		3	1 ACCOUNTING	LONDON
690	AHMAD	SALES	734	05/12/1997	22500	1400		3	2 RESEARCH	YORK
690	AHMAD	SALES	734	05/12/1997	22500	1400		3	3 SALES	BIRMINGHAM
690	AHMAD	SALES	734	05/12/1997	22500	1400		3	4 OPERATIONS	LEEDS
734	COX	MANAGER	875	11/06/2002	38500			3	1 ACCOUNTING	LONDON
734	COX	MANAGER	875	11/06/2002	38500			3	2 RESEARCH	YORK
734	COX	MANAGER	875	11/06/2002	38500			3	3 SALES	BIRMINGHAM
734	COX	MANAGER	875	11/06/2002	38500			3	4 OPERATIONS	LEEDS
818	POLLARD	MANAGER								
818	POLLARD	MANAGER								

.. with many more rows not shown here...

Why? Its in the definition of cartesian product

## CARTESIAN PRODUCT

$R \times S$

The Cartesian product operation defines a relation that is the concatenation of every tuple of relation  $R$  with every tuple of relation  $S$

Everything in the 'first' table is combined with everything in the 'second' table

$emp$  table 14 rows

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
405	MARCH	ADMIN	938	13/06/1997	18000		2
535	BYRNE	SALES	734	15/08/1997	26000	300	3
557	BELL	SALES	734	26/03/2000	22500	500	3
602	BIRD	MANAGER	875	31/10/1997	39750		2
690	AHMAD	SALES	734	05/12/1997	22500	1400	3
734	COX	MANAGER	875	11/06/2002	38500		3
818	POLLARD	MANAGER	875	14/05/2000	34500		1
824	REES	ANALYST	602	05/03/2000	40000		2
875	PARKER	PRESIDENT		09/07/2002	60000		1
880	TURNER	SALES	734	04/06/2001	25000	0	3
912	HAYES	ADMIN	824	04/06/2001	21000		2
936	CASSY	ADMIN	734	23/07/2002	19500		3
938	GIBSON	ANALYST	602	05/03/2000	40000		2
970	BLACK	ADMIN	818				1

$dept$  table 4 rows

DEPTNO	DNAME	LOC
1	ACCOUNTING	LONDON
2	RESEARCH	YORK
3	SALES	BIRMINGHAM
4	OPERATIONS	LEEDS

$14 * 4 = 56$  rows in the result

emp x dept

## Many incorrectly matched rows

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	emp.DEPTNO	dept.DEPTNO	DNAME	LOC
405	MARCH	ADMIN	938	13/06/1997	18000		2	1	ACCOUNTING	LONDON
405	MARCH	ADMIN	938	13/06/1997	18000		2	2	RESEARCH	YORK
405	MARCH	ADMIN	938	13/06/1997	18000		2	3	SALES	BIRMINGHAM
405	MARCH	ADMIN	938	13/06/1997	18000		2	4	OPERATIONS	LEEDS
535	BYRNE	SALES	734	15/08/1997	26000	500	3	1	ACCOUNTING	LONDON
535	BYRNE	SALES	734	15/08/1997	26000	500	3	2	RESEARCH	YORK
535	BYRNE	SALES	734	15/08/1997	26000	500	3	3	SALES	BIRMINGHAM
535	BYRNE	SALES	734	15/08/1997	26000	500	3	4	OPERATIONS	LEEDS
557	BELL	SALES	734	26/03/2000	22500	500	3	1	ACCOUNTING	LONDON
557	BELL	SALES	734	26/03/2000	22500	500	3	2	RESEARCH	YORK
557	BELL	SALES	734	26/03/2000	22500	500	3	3	SALES	BIRMINGHAM
557	BELL	SALES	734	26/03/2000	22500	500	3	4	OPERATIONS	LEEDS
602	BIRD	MANAGER	875	31/10/1997	39750		2	1	ACCOUNTING	LONDON
602	BIRD	MANAGER	875	31/10/1997	39750		2	2	RESEARCH	YORK
602	BIRD	MANAGER	875	31/10/1997	39750		2	3	SALES	BIRMINGHAM
602	BIRD	MANAGER	875	31/10/1997	39750		2	4	OPERATIONS	LEEDS
690	AHMAD	SALES	734	05/12/1997	22500	500	3	1	ACCOUNTING	LONDON
690	AHMAD	SALES	734	05/12/1997	22500	500	3	2	RESEARCH	YORK
690	AHMAD	SALES	734	05/12/1997	22500	500	3	3	SALES	BIRMINGHAM
690	AHMAD	SALES	734	05/12/1997	22500	500	3	4	OPERATIONS	LEEDS
734	COX	MANAGER	875	11/06/2002	38500		3	1	ACCOUNTING	LONDON
734	COX	MANAGER	875	11/06/2002	38500		3	2	RESEARCH	YORK
734	COX	MANAGER	875	11/06/2002	38500		3	3	SALES	BIRMINGHAM
734	COX	MANAGER	875	11/06/2002	38500		3	4	OPERATIONS	LEEDS
818	POLLARD	MANAGER	875	14/05/2000	34500		1	1	ACCOUNTING	LONDON
818	POLLARD	MANAGER	875	14/05/2000	34500		1	2	RESEARCH	YORK

## Joining tables

We have to indicate how the tables should be correctly connected

There may be many choices

...consider the deptno attributes in the result set

emp.DEPTNO	dept.DEPTNO	
2	1	AC
2	2	RE
2	3	SA
2	4	OF
300	3	AC
300	3	RE
300	3	SA
300	3	OF
500	3	AC
500	3	RE
500	3	SA
500	3	OF
2	1	AC
2	2	RE
2	3	SA
2	4	OF

We only want to include results where the deptno is the same in both relations/tables

emp.deptno = dept.deptno

This can be done with a selection

emp.DEPTNO	dept.DEPTNO	
2	1	AC
2	2	RE
2	3	SA
2	4	OF
300	3	AC
300	3	RE
300	3	SA
300	3	OF
500	3	AC
500	3	RE
500	3	SA
500	3	OF
2	1	AC
2	2	RE
2	3	SA
2	4	OF

## Combine the tables using a cartesian product

$emp \times dept$

## Add an appropriate condition

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

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

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	emp.DEPTNO	dept.DEPTNO	DNAME	LOC
405	MARCH	ADMIN	938	13/06/1997	18000		2	2	RESEARCH	YORK
535	BYRNE	SALES	734	15/08/1997	26000	300	3	3	SALES	BIRMINGHAM
557	BELL	SALES	734	26/03/2000	22500	500	3	3	SALES	BIRMINGHAM
602	BIRD	MANAGER	875	31/10/1997	39750		2	2	RESEARCH	YORK
690	AHMAD	SALES	734	05/12/1997	22500	1400	3	3	SALES	BIRMINGHAM
734	COX	MANAGER	875	11/06/2002	38500		3	3	SALES	BIRMINGHAM
818	POLLARD	MANAGER	875	14/05/2000	34500		1	1	ACCOUNTING	LONDON
824	REES	ANALYST	602	05/03/2000	40000		2	2	RESEARCH	YORK
875	PARKER	PRESIDENT		09/07/2002	60000		1	1	ACCOUNTING	LONDON
880	TURNER	SALES	734	04/06/2001	25000	0	3	3	SALES	BIRMINGHAM
912	HAYES	ADMIN	824	04/06/2001	21000		2	2	RESEARCH	YORK
936	CASSY	ADMIN	734	23/07/2002	19500		3	3	SALES	BIRMINGHAM
938	GIBSON	ANALYST	602	05/12/1997	40000		2	2	RESEARCH	YORK
970	BLACK	ADMIN	818	21/11/1997	23000		1	1	ACCOUNTING	LONDON

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

Note how we refer to the two deptno attributes using **tablename.columnname**

This operation is performed so frequently that it has its own name - an **Inner Join**

## So: how do we do cartesian product in SQL?

SQL always looks like this:

select \* or expression  
from relations  
where expression

← add tables here  
separated by  
commas



## Cartesian Product/Inner Join example

List staff along with where they work

```
select *  
from emp, dept  
where emp.deptno=dept.deptno
```

the particular columns we require go here

table names go here

the predicate goes here

We add a condition to turn the CP into a Join

List staff along with where they work

```
select *  
from emp, dept  
where emp.deptno=dept.deptno
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	emp.DEPTNO	dept.DEPTNO	DNAME	LOC
405	MARCH	ADMIN	938	13/06/1997	18000		2	2	RESEARCH	YORK
535	BYRNE	SALES	734	15/08/1997	26000	300	3	3	SALES	BIRMINGHAM
557	BELL	SALES	734	26/03/2000	22500	500	3	3	SALES	BIRMINGHAM
602	BIRD	MANAGER	875	31/10/1997	39750		2	2	RESEARCH	YORK
690	AHMAD	SALES	734	05/12/1997	22500	1400	3	3	SALES	BIRMINGHAM
734	COX	MANAGER	875	11/06/2002	38500		3	3	SALES	BIRMINGHAM
818	POLLARD	MANAGER	875	14/05/2000	34500		1	1	ACCOUNTING	LONDON
824	REES	ANALYST	602	05/03/2000	40000		2	2	RESEARCH	YORK
875	PARKER	PRESIDENT		09/07/2002	60000		1	1	ACCOUNTING	LONDON
880	TURNER	SALES	734	04/06/2001	25000	0	3	3	SALES	BIRMINGHAM
912	HAYES	ADMIN	824	04/06/2001	21000		2	2	RESEARCH	YORK
936	CASSY	ADMIN	734	23/07/2002	19500		3	3	SALES	BIRMINGHAM
938	GIBSON	ANALYST	602	05/12/1997	40000		2	2	RESEARCH	YORK
970	BLACK	ADMIN	818	21/11/1997	23000		1	1	ACCOUNTING	LONDON

Extra projections or selections can be added as necessary

```
select *  
from emp, dept  
where emp.deptno=dept.deptno
```

the particular columns we require go here

extra conditions go here with an AND or OR

## Example 2: Cartesian Product/Inner Join

List staffname, dname and location for staff who earn more than £30000

```
select ename, dname, loc  
from emp, dept  
where emp.deptno=dept.deptno  
and sal>30000
```