Querying a Relational Database

Three query languages:

- **SQL** (Structured Query Language) - declarative, used in commercial DBMSs [IBM mid 1970's]
- **Relational Algebra** - procedural, used as a theoretical tool, operations incorporated into SQL [Codd 1970, IBM]
- **QBE** (Query By Example) - graphical, used in MS ACCESS [IBM mid 1970's]

All three query languages are equivalent !

Definition of a computable query:

A *program* which takes a collection of tables as input as returns a table as its output.

Definition of a query:

A computable query which is restricted by disallowing *iteration* and *recursion* programming structures.

Alternative definition of a query:

The set of computable queries that can express *logical operations* on tables.

Types of operation:

- Projection
- Selection
- Join
- Union, Intersection and Difference
- Aggregation

Also,

- Create a table
- Modify a table
- Transaction management

Supplier-Parts-DB (Draw the ERD for the Database)

SNUM	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

Supplier table

PNUM	PNAME	COLOUR	WEIGHT	CITY
P1	Nut	Red	12	London
P2	Bolt	Green	17	Paris
P3	Screw	Blue	17	Rome
P4	Screw	Red	14	London
P5	Cam	Blue	12	Paris
P6	Cog	Red	19	London

Part table

SNUM	PNUM	QTY
S1	P1	900
S2	P3	3100
S2	P5	100
S3	P3	200
S3	P4	500
S4	P6	600
S5	P1	100
S5	P2	300
S5	P3	200
S5	P4	800
S5	P5	1000
S5	P6	700

Supply table

Projection

`Select` SNAME, CITY `From` Supplier

SNUM	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

SNAME	CITY
Smith	London
Jones	Paris
Blake	Paris
Clark	London
Adams	Athens

Projection

`Select` CITY `From` Supplier

SNUM	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

CITY
London
Paris
Athens

Selection

`Select` * `From` Supplier `Where` City = 'London'

SNUM	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

⇓

SNUM	SNAME	STATUS	CITY
S1	Smith	20	London
S4	Clark	20	London

Selection

`Select` SNAME, CITY **`From`** Supplier **`Where`** STATUS $>$ 20

SNUM	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

SNAME	CITY
Blake	Paris
Adams	Athens

Selection

Select SNAME, CITY **From** Supplier
Where STATUS $>$ 20 **AND** CITY $=$ 'Paris'

SNUM	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

SNAME	CITY
Blake	Paris

Selection Condition

A boolean expression may contain the following operators:

(1) `AND`: CITY = 'London' `AND` COLOUR = 'Red'

(2) `NOT`: `NOT` COLOUR = 'Red'
Equivalently: COLOUR <> 'Red'

(3) `OR`: COLOUR = 'Red' `OR` COLOUR = 'Blue'

Join

Select * **From** Small-Supplier, Small-Supply
Where Small-Supplier.SNUM = Small-Supply.SNUM

Small-Supplier Table		
SNUM	SNAME	CITY
S1	Smith	London
S2	Jones	Paris

Small-Supply Table		
SNUM	PNUM	QTY
S1	P1	900
S2	P3	3100
S2	P5	100

⇓

SNUM	SNAME	CITY	PNUM	QTY
S1	Smith	London	P1	900
S2	Jones	Paris	P3	3100
S2	Jones	Paris	P5	100

Join

Select * **From** Small-Part, Small-Supply
Where Small-Part.PNUM = Small-Supply.PNUM

Small-Part Table		
PNUM	PNAME	COLOUR
P1	Nut	Red
P3	Screw	Blue
P5	Cam	Blue

Small-Supply Table		
SNUM	PNUM	QTY
S1	P1	900
S2	P3	3100
S2	P5	100

PNUM	PNAME	COLOUR	SNUM	QTY
P1	Nut	Red	S1	900
P3	Screw	Blue	S2	3100
P5	Cam	Blue	S2	100

Join

Select * **From** Small-Supplier, Small-Part, Small-Supply
Where Small-Supplier.SNUM = Small-Supply.SNUM
AND Small-Part.PNUM = Small-Supply.PNUM

Small-Supplier Table		
SNUM	SNAME	CITY
S1	Smith	London
S2	Jones	Paris

Small-Part Table		
PNUM	PNAME	COLOUR
P1	Nut	Red
P3	Screw	Blue
P5	Cam	Blue

Small-Supply Table		
SNUM	PNUM	QTY
S1	P1	900
S2	P3	3100
S2	P5	100

⇓

SNUM	SNAME	CITY	PNUM	PNAME	COLOUR	QTY
S1	Smith	London	P1	Nut	Red	900
S2	Jones	Paris	P3	Screw	Blue	3100
S2	Jones	Paris	P5	Cam	Blue	100

Nested Queries

```
SELECT SNUM, QTY
FROM Supply
WHERE PNUM in
      (SELECT PNUM FROM Part WHERE PNAME = 'Screw')
```

Part and Supply Tables

⇓

SNUM	QTY
S2	3100
S3	200
S3	500
S5	200
S5	800

```
SELECT SNUM, QTY
FROM Parts p, Supply sp
WHERE p.PNUM = sp.PNUM and PNAME = 'Screw'
```

Not the lowest weight

```
SELECT PNAME
FROM Part
WHERE WEIGHT > any
      (SELECT WEIGHT FROM Part)
```

Part Table

⇓

PNAME
Bolt
Screw
Cog

The lowest weight

```
SELECT PNAME
FROM Part
WHERE WEIGHT <= all
      (SELECT WEIGHT FROM Part)
```

Part Table

PNAME
Nut
Cam

Union

```
Select * From London-Supplier
          Union
Select * From Paris-Supplier
```

London-Supplier		
SNUM	SNAME	STATUS
S1	Smith	20
S4	Clark	20

Paris-Supplier		
SNUM	SNAME	STATUS
S2	Jones	10
S3	Blake	30

London-Union-Paris		
SNUM	SNAME	STATUS
S1	Smith	20
S4	Clark	20
S2	Jones	10
S3	Blake	30

Intersect

```
Select * From London-Rome-Colour-Part
          Intersect
Select * From Paris-Colour-Part
```

London-Rome-Colour-Paris
COLOUR
Red
Blue

Paris-Colour-part
COLOUR
Green
Blue

London-Rome-Intersect-Paris
COLOUR
Blue

Difference

```
Select * From London-Rome-Colour-Part
          Minus
Select * From Paris-Colour-Part
```

London-Rome-Colour-Paris
COLOUR
Red Blue

Paris-Colour-part
COLOUR
Green Blue

London-Rome-Minus-Paris
COLOUR
Red

Symmetric Difference

(**Select** * **From** London-Rome-Colour-Part
Minus
Select * **From** Paris-Colour-Part)
Union
(**Select** * **From** Paris-Colour-Part
Minus
Select * **From** London-Rome-Colour-Part)

London-Rome-Colour-Paris
COLOUR
Red
Blue

Paris-Colour-part
COLOUR
Green
Blue

London-Rome-Symmetric-Paris
COLOUR
Red
Green

Aggregation

Q1 How many suppliers are listed?

Q2 How many suppliers are listed in each city?

Q3 What is the overall average and standard deviation of the weight of parts?

Q4 What is the average weight of parts of a given colour?

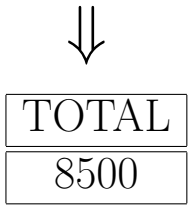
Q5 What is the maximum, respectively minimum, quantity a supplier has in stock?

Q6 What is the sum of quantities of parts each supplier has in stock?

The most common aggregate functions are:
`min`, `max`, `sum`, `avg`, `stddev` and `count`

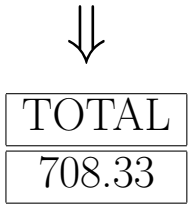
⋆ Beware of `null`s in numeric aggregates !

```
SELECT sum(QTY) TOTAL
FROM Supply
```



TOTAL
8500

```
SELECT avg(QTY) TOTAL
FROM Supply
```



TOTAL
708.33

```
SELECT count(*)
FROM Supply
```

⇓

count(*)
12

```
SELECT count(SNUM)
FROM Supply
```

⇓

count(SNUM)
5

```
SELECT sum(QTY * WEIGHT) TOT-WEIGHT
FROM Part, Supply
WHERE Part.PNUM = Supply.PNUM
```

TOT-WEIGHT
132700

```
SELECT PNAME
FROM Part
WHERE WEIGHT >
      (SELECT avg(WEIGHT) FROM Part)
```

Part Table

$\Downarrow$ avg $= 15.2$

PNAME
Bolt
Screw
Cog

```
SELECT PNUM, sum(QTY) TOTAL
FROM Supply
GROUP BY PNUM
```

PNUM	TOTAL
P1	1000
P2	300
P3	3500
P4	1300
P5	1100
P6	1300

```
SELECT PNUM, sum(QTY) TOTAL
FROM Supply
GROUP BY PNUM
HAVING sum(QTY) >= 1000
```

⇓

PNUM	TOTAL
P3	3500
P4	1300
P5	1100
P6	1300

```
SELECT PNUM, sum(QTY) TOTAL
FROM Supply
GROUP BY PNUM
HAVING sum(QTY) >
        (SELECT avg(QTY) FROM Supply)
ORDER BY sum(QTY)
```

avg(QTY)

⇓

PNUM	TOTAL
P1	1000
P5	1100
P4	1300
P6	1300
P3	3500