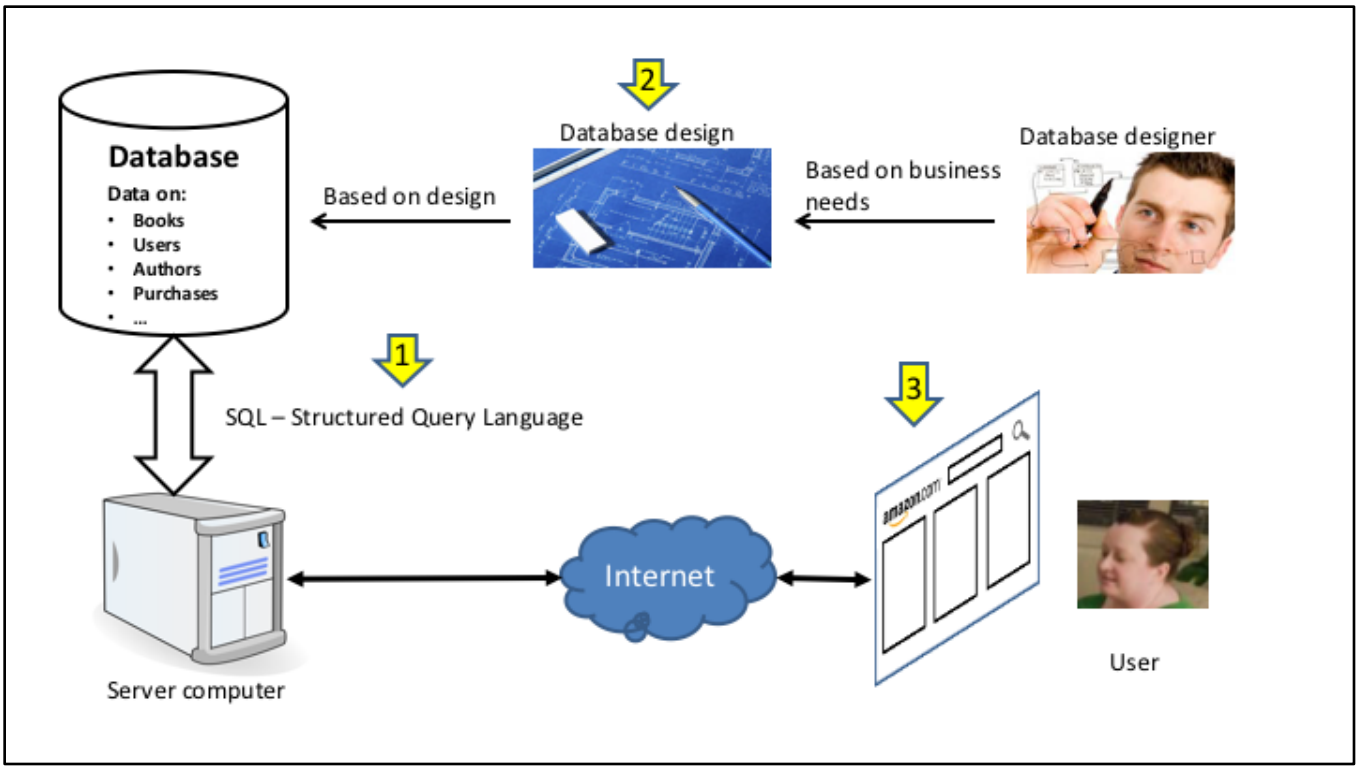
Business Information Modeling

Enterprise-wide Accounting Information Systems -- I

Week 3 – SQL: Part 2


Database
Data on:
• Books
• Users
• Authors
• Purchases
• ...
2
Database design
Based on design
Based on business needs
Database designer
1
SQL – Structured Query Language
3
Internet
Server computer
User

The classic Supplier-Parts database of

Chris J. Date

Suppliers

sno	sname	status	city
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

Parts

pno	pname	color	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

Projects

jno	jname	city
J1	Sorter	Paris
J2	Punch	Rome
J3	Reader	Athens
J4	Console	Athens
J5	Collator	London
J6	Termina	Oslo
J7	Tape	London

Shipments

sno	pno	jno	ship_date	qty
S1	P1	J1	3/3/1999	200
S1	P1	J4	4/4/1988	700
S2	P3	J1	6/6/1988	100
S2	P3	J2	8/8/1999	200
S2	P3	J3	7/7/1977	200
S2	P3	J4	8/9/1999	500
S2	P3	J6	8/7/1998	400
S2	P3	J7	8/8/1988	800
S2	P5	J2	9/9/1999	100
S4	P6	J3	9/9/1999	400
S4	P6	J7	9/8/1988	300

Copy of this provided as PDF document on Blackboard – use as reference as you watch the video

We will use the same set of tables as in the previous lecture.

The database shows the following tables:

- Suppliers
- Parts
- Projects
- Shipments

Query 23: All details of suppliers in London or Paris

sno	sname	status	city
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

sno	sname	status	city
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London

```
SELECT *
FROM suppliers
WHERE city = 'London'
OR city = 'Paris'
```

Using OR logical operator

```
SELECT *
FROM suppliers
WHERE city IN ('London', 'Paris')
```

Better to use IN – especially when we have more values

When we have to check that a specific field equals one of a set of values, we can write it using the OR logical operator, but this can become very unwieldy when we have to check against many values.

A much better and more succinct option is to use the IN operator as shown above.

Query 24: All details of Green or Blue parts – your turn

pno	pname	color	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

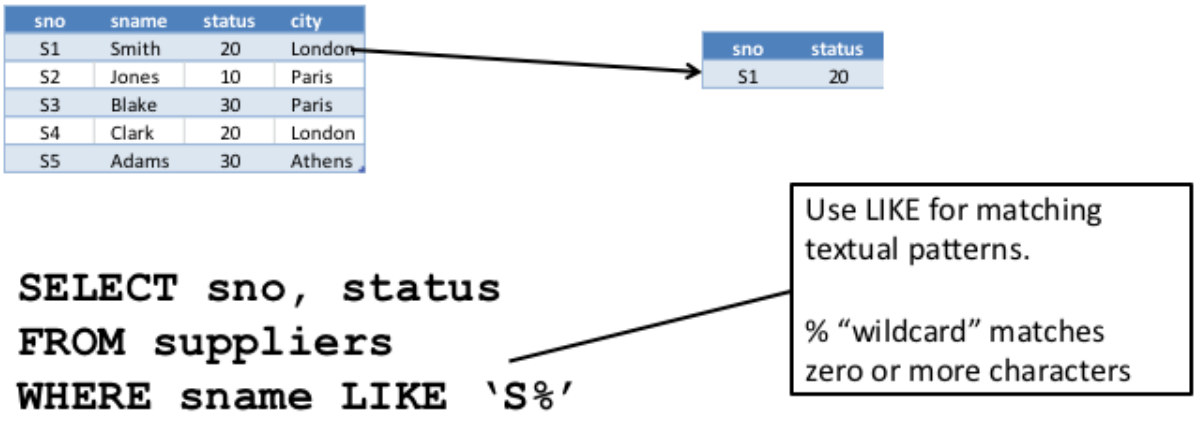
pno	pname	color	weight	city
P2	Bolt	Green	17	Paris
P3	Screw	Blue	17	Rome
P5	Cam	Blue	12	Paris

```
SELECT *
FROM parts
WHERE color = 'Green'
OR color = 'Blue'
```

```
SELECT *
FROM parts
WHERE color IN ('Green', 'Blue')
```

Preferred approach

Query 25: Supplier number and status of suppliers whose name begins with "S" (Not very meaningful in this context, but very useful in many other situations.)

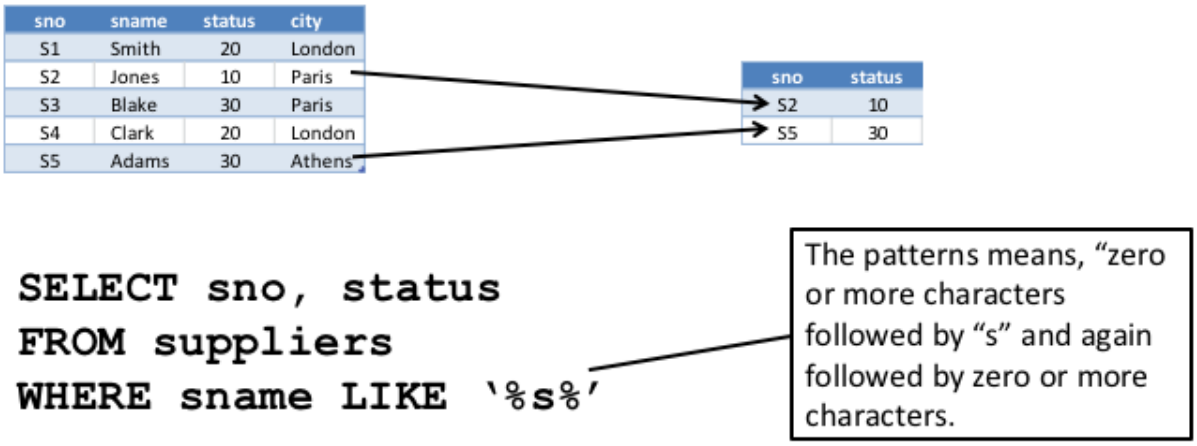


Use LIKE for textual pattern matching.

% wildcard matches zero or more characters.

The above example will match even a supplier with name "S" alone because the % also matches zero characters and none appear after "S".

Query 26: Supplier number and status of suppliers whose name has "s" (lowercase) anywhere in it.



sno	sname	status	city
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

sno	status
S2	10
S5	30

```
SELECT sno, status
FROM suppliers
WHERE sname LIKE '%s%'
```

The patterns means, "zero or more characters followed by "s" and again followed by zero or more characters.

Use LIKE for textual pattern matching.

% wildcard matches zero or more characters.

The above example will match even a supplier with name "S" alone because the % also matches zero characters and none appear after "S".

Query 27: Supplier number and status of suppliers whose name has "l" ("el", lowercase) as the second character.

sno	sname	status	city
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

sno	status
S3	30
S4	20

```
SELECT sno, status
FROM suppliers
WHERE sname LIKE '_l%'
```

The "_" matches a single character.

This pattern means, "exactly one character followed by "l" and followed by zero or more characters.

Query 28: Part number and color for parts whose names start with "C" (your turn)

pno	pname	color	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

pno	color
P5	Blue
P6	Red

```
SELECT pno, color
FROM parts
WHERE pname LIKE 'C%'
```

Query 29: Part number and color for parts whose names have "o" anywhere in them. (your turn)

pno	pname	color	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

pno	color
P2	Green
P6	Red

```
SELECT pno, color
FROM parts
WHERE pname LIKE '%o%'
```

Query 30: Part number and color for parts whose names have exactly three characters in them. (your turn)

pno	pname	color	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

pno	color
P1	Red
P5	Blue
P6	Red

```
SELECT pno, color
FROM parts
WHERE pname LIKE '___'
```

Three underscore "_" characters

Query 31: Part number and color for parts whose names have at least four characters in them. (your turn)

pno	pname	color	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

pno	color
P2	Green
P3	Blue
P4	Red

```
SELECT pno, color
FROM parts
WHERE pname LIKE '____%'
```

Four underscore "_" characters followed by "%"

Query 32: For every shipment, get the supplier name, part number and shipment quantity.

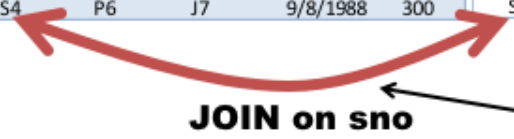
Shipments table

sno	pno	jno	ship_date	qty
S1	P1	J1	3/3/1999	200
S1	P1	J4	4/4/1988	700
S2	P3	J1	6/6/1988	100
S2	P3	J2	8/8/1999	200
S2	P3	J3	7/7/1977	200
S2	P3	J4	8/9/1999	500
S2	P3	J6	8/7/1998	400
S2	P3	J7	8/8/1988	800
S2	P5	J2	9/9/1999	100
S4	P6	J3	9/9/1999	400
S4	P6	J7	9/8/1988	300

Appended supplier information

sno	sname	status	city
S1	Smith	20	London
S1	Smith	20	London
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S4	Clark	20	London
S4	Clark	20	London

Conceptually, join creates a "super table" from which the system selects the needed columns



JOIN on sno

Join the shipments table and the suppliers table on sno

This query requires the system to retrieve data from two tables. The supplier names have to come from the suppliers table and the rest of the data has to come from the shipments table.

In SQL, when we need information from multiple tables, we need to JOIN the tables on the appropriate common column.

From the slide, we can see that this process creates (conceptually) a larger joined table.

We can then get the necessary columns from this joined table.

This is how the SQL processor ***conceptually*** handles the retrieval. Internally, it will find much more efficient ways to handle the operation, but being fans of non-procedural queries, we will not need to bother about the details.

Query 32 (continued): For every shipment, get the supplier name, part number and shipment quantity.

Shipments table

sno	pno	jno	ship_date	qty
S1	P1	J1	3/3/1999	200
S1	P1	J4	4/4/1988	700
S2	P3	J1	6/6/1988	100
S2	P3	J2	8/8/1999	200
S2	P3	J3	7/7/1977	200
S2	P3	J4	8/9/1999	500
S2	P3	J6	8/7/1998	400
S2	P3	J7	8/8/1988	800
S2	P5	J2	9/9/1999	100
S4	P6	J3	9/9/1999	400
S4	P6	J7	9/8/1988	300

Appended supplier information

sno	sname	status	city
S1	Smith	20	London
S1	Smith	20	London
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S4	Clark	20	London
S4	Clark	20	London

Result

sname	pno	qty
Smith	P1	200
Smith	P1	700
Jones	P3	100
Jones	P3	200
Jones	P3	200
Jones	P3	500
Jones	P3	400
Jones	P3	800
Jones	P5	100
Clark	P6	400
Clark	P6	300

This query requires the system to retrieve data from two tables. The supplier names have to come from the suppliers table and the rest of the data has to come from the shipments table.

Query 32 (continued): For every shipment, get the supplier name, part number and shipment quantity.

Shipments table

sno	pno	jno	ship_date	qty
S1	P1	J1	3/3/1999	200
S1	P1	J4	4/4/1988	700
S2	P3	J1	6/6/1988	100
S2	P3	J2	8/8/1999	200
S2	P3	J3	7/7/1977	200
S2	P3	J4	8/9/1999	500
S2	P3	J6	8/7/1998	400
S2	P3	J7	8/8/1988	800
S2	P5	J2	9/9/1999	100
S4	P6	J3	9/9/1999	400
S4	P6	J7	9/8/1988	300

Appended supplier information

sno	sname	status	city
S1	Smith	20	London
S1	Smith	20	London
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S4	Clark	20	London
S4	Clark	20	London

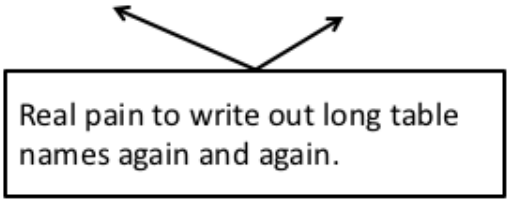
Result

sname	pno	qty
Smith	P1	200
Smith	P1	700
Jones	P3	100
Jones	P3	200
Jones	P3	200
Jones	P3	500
Jones	P3	400
Jones	P3	800
Jones	P5	100
Clark	P6	400
Clark	P6	300

```
SELECT sname, pno, qty
FROM suppliers JOIN shipments ON suppliers.sno = shipments.sno
```

Query 32 (continued): For every shipment, get the supplier name, part number and shipment quantity.

```
SELECT sname, pno, qty
FROM suppliers JOIN shipments ON suppliers.sno = shipments.sno
```




```
SELECT sname, pno, qty
FROM suppliers s JOIN shipments sp   ON s.sno = sp.sno
```

Create "aliases " for the names of tables joined and use the aliases for subsequent references.

Query 32 (continued): For every shipment, get the supplier name, part number and shipment quantity.

Shipments table

sno	pno	jno	ship_date	qty
S1	P1	J1	3/3/1999	200
S1	P1	J4	4/4/1988	700
S2	P3	J1	6/6/1988	100
S2	P3	J2	8/8/1999	200
S2	P3	J3	7/7/1977	200
S2	P3	J4	8/9/1999	500
S2	P3	J6	8/7/1998	400
S2	P3	J7	8/8/1988	800
S2	P5	J2	9/9/1999	100
S4	P6	J3	9/9/1999	400
S4	P6	J7	9/8/1988	300

Appended supplier information

sno	sname	status	city
S1	Smith	20	London
S1	Smith	20	London
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S4	Clark	20	London
S4	Clark	20	London

```
SELECT sname, pno, qty
FROM suppliers s
JOIN shipments sp
ON s.sno = sp.sno
```

INNER JOIN

Why no S3 or S5?

Because of this, the system only appends information for supplier numbers that appear in the shipments table.

INNER JOIN

Default method when we do not specify anything else.

Query 33 : For every shipment, get the part name, project number and shipment quantity. (Your turn)

Shipments table

sno	pno	jno	ship_date	qty
S1	P1	J1	3/3/1999	200
S1	P1	J4	4/4/1988	700
S2	P3	J1	6/6/1988	100
S2	P3	J2	8/8/1999	200
S2	P3	J3	7/7/1977	200
S2	P3	J4	8/9/1999	500
S2	P3	J6	8/7/1998	400
S2	P3	J7	8/8/1988	800
S2	P5	J2	9/9/1999	100
S4	P6	J3	9/9/1999	400
S4	P6	J7	9/8/1988	300

Joined parts

P1 Nut	Red	12	London
P1 Nut	Red	12	London
P3 Screw	Blue	17	Rome
P3 Screw	Blue	17	Rome
P3 Screw	Blue	17	Rome
P3 Screw	Blue	17	Rome
P3 Screw	Blue	17	Rome
P3 Screw	Blue	17	Rome
P5 Cam	Blue	12	Paris
P6 Cog	Red	19	London
P6 Cog	Red	19	London

Result

pname	jno	qty
Nut	J1	200
Nut	J4	700
Screw	J1	100
Screw	J2	200
Screw	J3	200
Screw	J4	500
Screw	J6	400
Screw	J7	800
Cam	J2	100
Cog	J3	400
Cog	J7	300

JOIN on pno

```
SELECT pname, jno, qty
FROM parts p JOIN shipments sp ON p.pno = sp.pno
```

Query 34 : For every shipment, get the part name, project name, supplier name and shipment quantity.

Shipments

sno	pno	jno	ship_date	qty
S1	P1	J1	3/3/1999	200
S1	P1	J4	4/4/1988	700
S2	P3	J1	6/6/1988	100
S2	P3	J2	8/8/1999	200
S2	P3	J3	7/7/1977	200
S2	P3	J4	8/9/1999	500
S2	P3	J6	8/7/1998	400
S2	P3	J7	8/8/1988	800
S2	P5	J2	9/9/1999	100
S4	P6	J3	9/9/1999	400
S4	P6	J7	9/8/1988	300

Parts

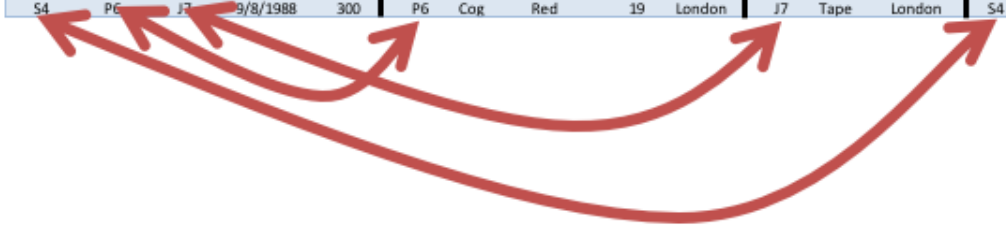
pno	pname	color	weight	city
P1	Nut	Red	12	London
P1	Nut	Red	12	London
P3	Screw	Blue	17	Rome
P3	Screw	Blue	17	Rome
P3	Screw	Blue	17	Rome
P3	Screw	Blue	17	Rome
P3	Screw	Blue	17	Rome
P3	Screw	Blue	17	Rome
P5	Cam	Blue	12	Paris
P6	Cog	Red	19	London
P6	Cog	Red	19	London

Projects

jno	jname	city
J1	Sorter	Paris
J4	Console	Athens
J1	Sorter	Paris
J2	Punch	Rome
J3	Reader	Athens
J4	Console	Athens
J6	Terminal	Oslo
J7	Tape	London
J2	Punch	Rome
J3	Reader	Athens
J7	Tape	London

Suppliers

sno	sname	status	city
S1	Smith	20	London
S1	Smith	20	London
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S4	Clark	20	London
S4	Clark	20	London



Query 34 (continued) : For every shipment, get the part name, project name, supplier name and shipment quantity.

Shipments

sno	pno	jno	ship	date	qty
S1	P1	J1	3/3	999	200
S1	P1	J4	4/4	988	700
S2	P3	J1	6/6	988	100
S2	P3	J2	8/8	999	200
S2	P3	J3	7/7	977	200
S2	P3	J4	8/9	999	500
S2	P3	J6	8/7	998	400
S2	P3	J7	8/8	988	800
S2	P5	J2	9/9	999	100
S4	P6	J3	9/9	999	400
S4	P6	J7	9/8	988	300

Parts

pno	pname	color	weight	city
P1	Nut	Red	12	London
P1	Nut	Red	12	London
P3	Screw	Blue	17	Rome
P3	Screw	Blue	17	Rome
P3	Screw	Blue	17	Rome
P3	Screw	Blue	17	Rome
P3	Screw	Blue	17	Rome
P3	Screw	Blue	17	Rome
P5	Cam	Blue	12	Paris
P6	Cog	Red	19	London
P6	Cog	Red	19	London

Projects

jno	jname	city
J1	Sorter	Paris
J4	Console	Athens
J1	Sorter	Paris
J2	Punch	Rome
J3	Reader	Athens
J4	Console	Athens
J6	Termina	Oslo
J7	Tape	London
J2	Punch	Rome
J3	Reader	Athens
J7	Tape	London

Suppliers

sno	sname	status	city
S1	Smith	20	London
S1	Smith	20	London
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S4	Clark	20	London
S4	Clark	20	London

pname	jname	sname	qty
Nut	Sorter	Smith	200
Nut	Console	Smith	700
Screw	Sorter	Jones	100
Screw	Punch	Jones	200
Screw	Reader	Jones	200
Screw	Console	Jones	500
Screw	Terminal	Jones	400
Screw	Tape	Jones	800
Cam	Punch	Jones	100
Cog	Reader	Clark	400
Cog	Tape	Clark	300

Query 34 (continued) : For every shipment, get the part name, project name, supplier name and shipment quantity.

Shipments

sno	pno	jno	ship_date	qty
S1	P1	J1	3/3/1999	200
S1	P1	J4	4/4/1988	700
S2	P3	J1	6/6/1988	100
S2	P3	J2	8/8/1999	200
S2	P3	J3	7/7/1977	200
S2	P3	J4	8/9/1999	500
S2	P3	J6	8/7/1998	400
S2	P3	J7	8/8/1988	800
S2	P5	J2	9/9/1999	100
S4	P6	J3	9/9/1999	400
S4	P6	J7	9/8/1988	300

Parts

pno	pname	color	weight	city
P1	Nut	Red	12	London
P1	Nut	Red	12	London
P3	Screw	Blue	17	Rome
P3	Screw	Blue	17	Rome
P3	Screw	Blue	17	Rome
P3	Screw	Blue	17	Rome
P3	Screw	Blue	17	Rome
P3	Screw	Blue	17	Rome
P5	Cam	Blue	12	Paris
P6	Cog	Red	19	London
P6	Cog	Red	19	London

Projects

jno	jname	city
J1	Sorter	Paris
J4	Console	Athens
J1	Sorter	Paris
J2	Punch	Rome
J3	Reader	Athens
J4	Console	Athens
J6	Terminal	Oslo
J7	Tape	London
J2	Punch	Rome
J3	Reader	Athens
J7	Tape	London

Suppliers

sno	sname	status	city
S1	Smith	20	London
S1	Smith	20	London
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S4	Clark	20	London
S4	Clark	20	London

pname	jname	sname	qty
Nut	Sorter	Smith	200
Nut	Console	Smith	700
Screw	Sorter	Jones	100
Screw	Punch	Jones	200
Screw	Reader	Jones	200
Screw	Console	Jones	500
Screw	Terminal	Jones	400
Screw	Tape	Jones	800
Cam	Punch	Jones	100
Cog	Reader	Clark	400
Cog	Tape	Clark	300

```
SELECT pname, jname, sname, qty
FROM shipments sp
JOIN parts p ON sp.pno = p.pno
JOIN projects j ON sp.jno = j.jno
JOIN suppliers s ON sp.sno = s.sno
```

Query 35: List the name and city for every supplier who has made at least one shipment.

Shipments table

sno	pno	jno	ship_date	qty
S1	P1	J1	3/3/1999	200
S1	P1	J4	4/4/1988	700
S2	P3	J1	6/6/1988	100
S2	P3	J2	8/8/1999	200
S2	P3	J3	7/7/1977	200
S2	P3	J4	8/9/1999	500
S2	P3	J6	8/7/1998	400
S2	P3	J7	8/8/1988	800
S2	P5	J2	9/9/1999	100
S4	P6	J3	9/9/1999	400
S4	P6	J7	9/8/1988	300

Appended supplier information

sno	sname	status	city
S1	Smith	20	London
S1	Smith	20	London
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S4	Clark	20	London
S4	Clark	20	London

Result

sname	city
Smith	London
Smith	London
Jones	Paris
Jones	Paris
Jones	Paris
Jones	Paris
Jones	Paris
Jones	Paris
Jones	Paris
Clark	London
Clark	London

Unwanted duplicates!

```
SELECT sname, city
FROM shipments sp
JOIN suppliers s on sp.sno = s.sno
```

We are

Query 35 (continued): List the name and city for every supplier who has made at least one shipment.

Shipments table

sno	pno	jno	ship_date	qty
S1	P1	J1	3/3/1999	200
S1	P1	J4	4/4/1988	700
S2	P3	J1	6/6/1988	100
S2	P3	J2	8/8/1999	200
S2	P3	J3	7/7/1977	200
S2	P3	J4	8/9/1999	500
S2	P3	J6	8/7/1998	400
S2	P3	J7	8/8/1988	800
S2	P5	J2	9/9/1999	100
S4	P6	J3	9/9/1999	400
S4	P6	J7	9/8/1988	300

Appended supplier information

sno	sname	status	city
S1	Smith	20	London
S1	Smith	20	London
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S4	Clark	20	London
S4	Clark	20	London

Result

Smith London
Jones Paris
Clark London

Duplicates gone !

```
SELECT DISTINCT sname, city
FROM shipments sp
JOIN suppliers s on sp.sno = s.sno
```

Query 36: List the name and city for every part for which we have at least one shipment. (Your turn)

Shipments

sno	pno	jno	ship_date	qty
S1	P1	J1	3/3/1999	200
S1	P1	J4	4/4/1988	700
S2	P3	J1	6/6/1988	100
S2	P3	J2	8/8/1999	200
S2	P3	J3	7/7/1977	200
S2	P3	J4	8/9/1999	500
S2	P3	J6	8/7/1998	400
S2	P3	J7	8/8/1988	800
S2	P5	J2	9/9/1999	100
S4	P6	J3	9/9/1999	400
S4	P6	J7	9/8/1988	300

Parts

pno	pname	color	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

Result

Nut	London
Screw	Rome
Cam	Paris
Cog	London

```
SELECT DISTINCT pname, city
FROM shipments sp
JOIN parts p on sp.pno = p.pno
```

Query 37 : List the supplier number, name and number of shipments for every supplier who has made at least one shipment.

Shipments table

sno	pno	jno	ship_date	qty
S1	P1	J1	3/3/1999	200
S1	P1	J4	4/4/1988	700
S2	P3	J1	6/6/1988	100
S2	P3	J2	8/8/1999	200
S2	P3	J3	7/7/1977	200
S2	P3	J4	8/9/1999	500
S2	P3	J6	8/7/1998	400
S2	P3	J7	8/8/1988	800
S2	P5	J2	9/9/1999	100
S4	P6	J3	9/9/1999	400
S4	P6	J7	9/8/1988	300

Appended supplier information

sno	sname	status	city
S1	Smith	20	London
S1	Smith	20	London
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S4	Clark	20	London
S4	Clark	20	London

Result

S1	Smith	2
S2	Jones	7
S4	Clark	2

```
SELECT s.sno, sname, count(*) as "No. of shipments"
FROM suppliers s JOIN shipments sp on sp.sno = s.sno
GROUP BY sno, sname
```

Query 38 : List the part number, name and number of shipments for every part that has at least one shipment. (Your turn)

```
SELECT p.pno, pname, count(*) as "No. of shipments"
FROM parts p JOIN shipments sp on p.pno = sp.pno
GROUP BY p.pno, pname
```

Query 39 : List the supplier names and the corresponding total quantity and average shipment quantities.

What will the output look like?

sname	sum(qty)	avg(qty)
Clark	...	...
Jones	...	...
Smith	...	...

We want one row per supplier, showing the aggregated values. So we need to GROUP BY sname

```
SELECT sname, sum(qty), avg(qty)
FROM suppliers s JOIN shipments sp ON s.sno = sp.sno
GROUP BY sname
```

Query 40 : List the part names and the corresponding maximum, minimum and average shipment quantities. Your turn.

What will the output look like?

pname	max(qty)	min(qty)	avg(qty)
Cam	...	...	...
Cog	...	...	...
Nut	...	...	...
Screw	...	...	...

We want one row per part, showing the aggregated values. So we need to GROUP BY pname

```
SELECT pname, max(qty), min(qty), avg(qty)
FROM parts p JOIN shipments sp ON p.pno = sp.pno
GROUP BY pname
```

Query 41 : List the supplier name, part number, project number, and quantity for every supplier who has made at least one shipment. For suppliers who have not made any shipments, include only the supplier name.

Shipments table

sno	pno	jno	ship_date	qty
S1	P1	J1	3/3/1999	200
S1	P1	J4	4/4/1988	700
S2	P3	J1	6/6/1988	100
S2	P3	J2	8/8/1999	200
S2	P3	J3	7/7/1977	200
S2	P3	J4	8/9/1999	500
S2	P3	J6	8/7/1998	400
S2	P3	J7	8/8/1988	800
S2	P5	J2	9/9/1999	100
S4	P6	J3	9/9/1999	400
S4	P6	J7	9/8/1988	300

Appended supplier information

sno	sname	status	city
S1	Smith	20	London
S1	Smith	20	London
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S4	Clark	20	London
S4	Clark	20	London

Result

sname	pno	jno	qty
Smith	P1	J1	200
Smith	P1	J4	700
Jones	P3	J1	100
Jones	P3	J2	200
Jones	P3	J3	200
Jones	P3	J4	500
Jones	P3	J6	400
Jones	P3	J7	800
Jones	P5	J2	100
Clark	P6	J3	400
Clark	P6	J7	300

Blake and Adams missing … INNER JOIN and no matching shipments

```
SELECT sname, pno, jno, qty
FROM suppliers s JOIN shipments sp on s.sno = sp.sno
```

Query 41 (continued) : List the supplier name, part number, project number, and quantity for every supplier who has made at least one shipment. For suppliers who have not made any shipments, include only the supplier name.

Shipments table

sno	pno	jno	ship_date	qty
S1	P1	J1	3/3/1999	200
S1	P1	J4	4/4/1988	700
S2	P3	J1	6/6/1988	100
S2	P3	J2	8/8/1999	200
S2	P3	J3	7/7/1977	200
S2	P3	J4	8/9/1999	500
S2	P3	J6	8/7/1998	400
S2	P3	J7	8/8/1988	800
S2	P5	J2	9/9/1999	100
S4	P6	J3	9/9/1999	400
S4	P6	J7	9/8/1988	300

Appended supplier information

sno	sname	status	city
S1	Smith	20	London
S1	Smith	20	London
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S4	Clark	20	London
S4	Clark	20	London
S3	Blake	30	Paris
S5	Adams	30	Athens

What we would like …

Join the matching rows

For unmatched rows include only supplier information

Query 41 (continued) : List the supplier name, part number, project number, and quantity for every supplier who has made at least one shipment. For suppliers who have not made any shipments, include only the supplier name.

Shipments table

sno	pno	jno	ship_date	qty
S1	P1	J1	3/3/1999	200
S1	P1	J4	4/4/1988	700
S2	P3	J1	6/6/1988	100
S2	P3	J2	8/8/1999	200
S2	P3	J3	7/7/1977	200
S2	P3	J4	8/9/1999	500
S2	P3	J6	8/7/1998	400
S2	P3	J7	8/8/1988	800
S2	P5	J2	9/9/1999	100
S4	P6	J3	9/9/1999	400
S4	P6	J7	9/8/1988	300

We want this

sname	pno	jno	qty
Smith	P1	J1	200
Smith	P1	J4	700
Jones	P3	J1	100
Jones	P3	J2	200
Jones	P3	J3	200
Jones	P3	J4	500
Jones	P3	J6	400
Jones	P3	J7	800
Jones	P5	J2	100
Clark	P6	J3	400
Clark	P6	J7	300
Blake	NULL	NULL	NULL
Adams	NULL	NULL	NULL

INNER JOIN will not produce these two rows because S3 and S5 have made no shipments

OUTER JOIN

Allows us to JOIN matched records and also include non-matching information from one table in the pair involved in the join.

`a JOIN b` `ON a.x = b.y`	JOIN only matching rows from a and b	**INNER JOIN**
`a LEFT JOIN b` `ON a.x = b.y`	JOIN matching rows from a and b and also rows from a that have no match in b	**LEFT OUTER JOIN**
`a RIGHT JOIN b` `ON a.x = b.y`	JOIN matching rows from a and b and also rows from b that have no match in a	**RIGHT OUTER JOIN**

Query 41 (continued) : List the supplier name, part number, project number, and quantity for every supplier who has made at least one shipment. For suppliers who have not made any shipments, include only the supplier name.

Appended supplier information

sno	sname	status	city
S1	Smith	20	London
S1	Smith	20	London
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S4	Clark	20	London
S4	Clark	20	London
S3	Blake	30	Paris
S5	Adams	30	Athens

Shipments table

sno	pno	jno	ship_date	qty
S1	P1	J1	3/3/1999	200
S1	P1	J4	4/4/1988	700
S2	P3	J1	6/6/1988	100
S2	P3	J2	8/8/1999	200
S2	P3	J3	7/7/1977	200
S2	P3	J4	8/9/1999	500
S2	P3	J6	8/7/1998	400
S2	P3	J7	8/8/1988	800
S2	P5	J2	9/9/1999	100
S4	P6	J3	9/9/1999	400
S4	P6	J7	9/8/1988	300
NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL

We want this

sname	pno	jno	qty
Smith	P1	J1	200
Smith	P1	J4	700
Jones	P3	J1	100
Jones	P3	J2	200
Jones	P3	J3	200
Jones	P3	J4	500
Jones	P3	J6	400
Jones	P3	J7	800
Jones	P5	J2	100
Clark	P6	J3	400
Clark	P6	J7	300
Blake	NULL	NULL	NULL
Adams	NULL	NULL	NULL

```
SELECT sname, pno, jno, qty
FROM suppliers s
LEFT JOIN shipments sp ON s.sno = sp.sno
```

LEFT JOIN is doing the trick for us – explanation follows.

NULL

… means that a field **does not have a value**

… Different from "Blank" or zero value.

Query 41 (continued) : List the supplier name, part number, project number, and quantity for every supplier who has made at least one shipment. For suppliers who have not made any shipments, include only the supplier name.

```
SELECT sname, pno, jno, qty
FROM suppliers s
LEFT JOIN shipments sp ON s.sno = sp.sno
```

```
suppliers s LEFT JOIN shipments sp on s.sno = sp.sno
```

Table on the left in the join

Table on the right in the join

With **LEFT JOIN**, rows on the left table with no matches in the table on the right will be included. Hence Blake and Adams were included.

Query 41 (continued) : List the supplier name, part number, project number, and quantity for every supplier who has made at least one shipment. For suppliers who have not made any shipments, include only the supplier name.

```
SELECT sname, pno, jno, qty
FROM suppliers s
LEFT JOIN shipments sp
ON s.sno = sp.sno
```

sname	pno	jno	qty
Smith	P1	J1	200
Smith	P1	J4	700
Jones	P3	J1	100
Jones	P3	J2	200
Jones	P3	J3	200
Jones	P3	J4	500
Jones	P3	J6	400
Jones	P3	J7	800
Jones	P5	J2	100
Clark	P6	J3	400
Clark	P6	J7	300
Blake	NULL	NULL	NULL
Adams	NULL	NULL	NULL

For suppliers who do not appear in the shipments table, pno, jno and qty all have "NULL" values

Query 41 (continued) : List the supplier name, part number, project number, and quantity for every supplier who has made at least one shipment. For suppliers who have not made any shipments, include only the supplier name.

```
SELECT sname, pno, jno, qty
FROM suppliers s
LEFT JOIN shipments sp on s.sno = sp.sno
```

```
SELECT sname, pno, jno, qty
FROM shipments sp
RIGHT JOIN suppliers s on s.sno = sp.sno
```

Same results

Query 41 (continued) : List the supplier name, part number, project number, and quantity for every supplier who has made at least one shipment. For suppliers who have not made any shipments, include only the supplier name.

```
SELECT sname, pno, jno, qty
FROM shipments sp
RIGHT JOIN suppliers s on s.sno = sp.sno
```

```
FROM shipments RIGHT JOIN suppliers
```

On left | On right

With **RIGHT JOIN**, rows on the right table with no matches in the table on the left will be included. Hence Blake and Adams will appear.

Query 42 : List the project name, part number, supplier number, and quantity for every project that received at least one shipment. For projects without any shipments, include only the project name. Your turn.

```
SELECT jname, pno, sno, qty
FROM shipments sp
RIGHT JOIN projects j on j.jno = sp.jno
```

```
SELECT jname, pno, sno, qty
FROM projects j
LEFT JOIN shipments sp on j.jno = sp.jno
```

Query 43 : List the names of suppliers who have made no shipments.

Shipments table

sno	pno	jno	ship_date	qty
S1	P1	J1	3/3/1999	200
S1	P1	J4	4/4/1988	700
S2	P3	J1	6/6/1988	100
S2	P3	J2	8/8/1999	200
S2	P3	J3	7/7/1977	200
S2	P3	J4	8/9/1999	500
S2	P3	J6	8/7/1998	400
S2	P3	J7	8/8/1988	800
S2	P5	J2	9/9/1999	100
S4	P6	J3	9/9/1999	400
S4	P6	J7	9/8/1988	300
NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL

Appended supplier information

sno	sname	status	city
S1	Smith	20	London
S1	Smith	20	London
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S4	Clark	20	London
S4	Clark	20	London
S3	Blake	30	Paris
S5	Adams	30	Athens

Use OUTER JOIN ...

... but select only rows from suppliers table that have no match in the shipments table

```
SELECT sname
FROM shipments sp
RIGHT JOIN suppliers s on s.sno = sp.sno
WHERE sp.sno IS NULL
```

Query 44 : List the names of suppliers who have made no shipments – use LEFT JOIN. (Your turn)

Shipments table

sno	pno	jno	ship_date	qty
S1	P1	J1	3/3/1999	200
S1	P1	J4	4/4/1988	700
S2	P3	J1	6/6/1988	100
S2	P3	J2	8/8/1999	200
S2	P3	J3	7/7/1977	200
S2	P3	J4	8/9/1999	500
S2	P3	J6	8/7/1998	400
S2	P3	J7	8/8/1988	800
S2	P5	J2	9/9/1999	100
S4	P6	J3	9/9/1999	400
S4	P6	J7	9/8/1988	300
NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL

Appended supplier information

sno	sname	status	city
S1	Smith	20	London
S1	Smith	20	London
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S2	Jones	10	Paris
S4	Clark	20	London
S4	Clark	20	London
S3	Blake	30	Paris
S5	Adams	30	Athens

Use OUTER JOIN .....

... but select only rows from suppliers table that have no match in the shipments table

```
SELECT sname
FROM suppliers s
LEFT JOIN shipments sp on s.sno = sp.sno
WHERE sp.sno IS NULL
```

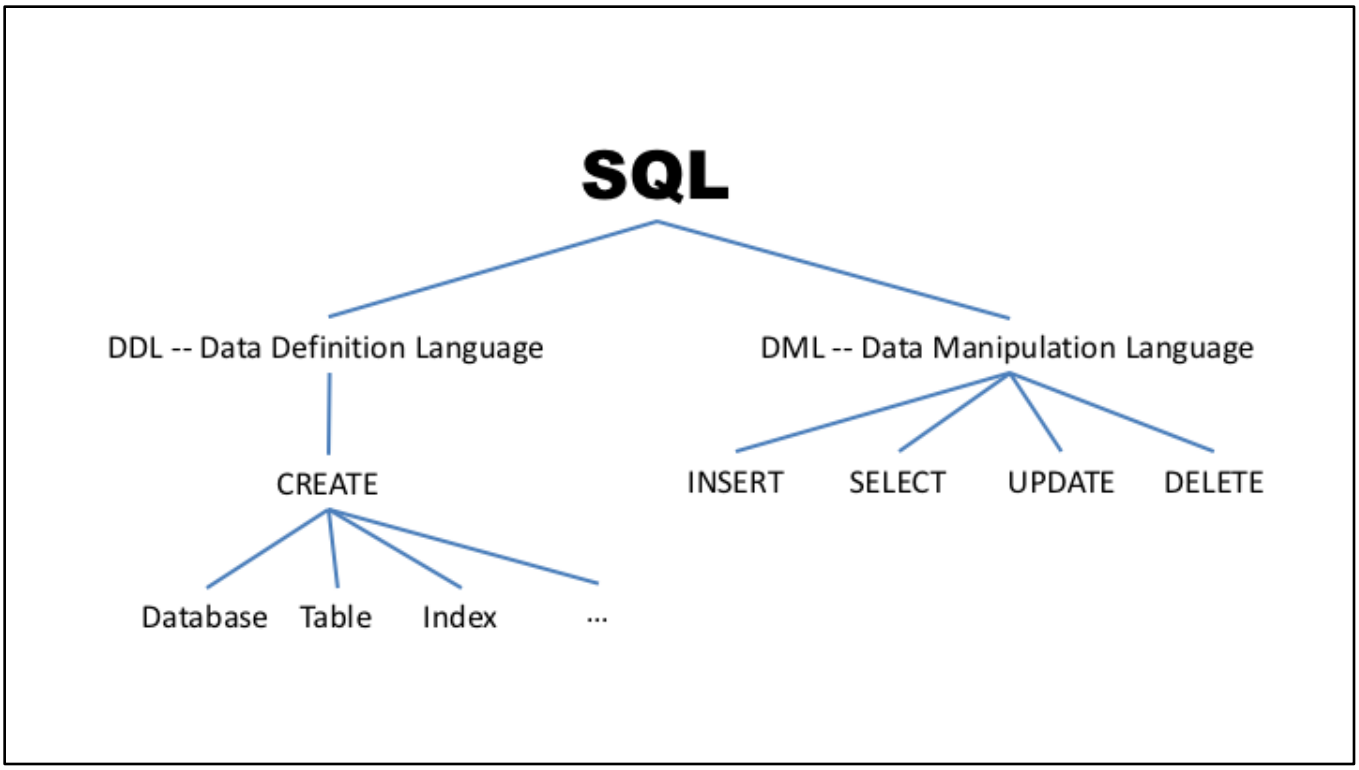
Query 45 : List the city and the names of suppliers, parts and projects and quantity for all cases when a shipment has all three from the same city

```
SELECT s.city, sname, pname, jname, qty
FROM shipments sp
JOIN suppliers s ON sp.sno = s.sno
JOIN parts p ON sp.pno = p.pno
JOIN projects j ON sp.jno = j.jno
WHERE s.city = p.city AND p.city = j.city
```

JOIN the four tables ...

... but select only rows where all three cities are the same

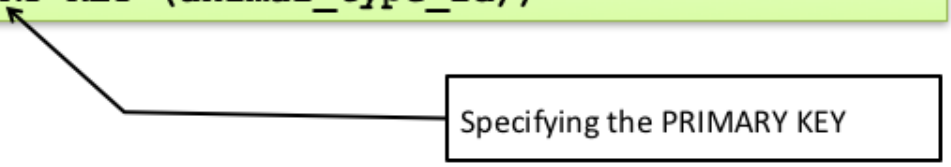
London	Clark	Cog	Tape	300

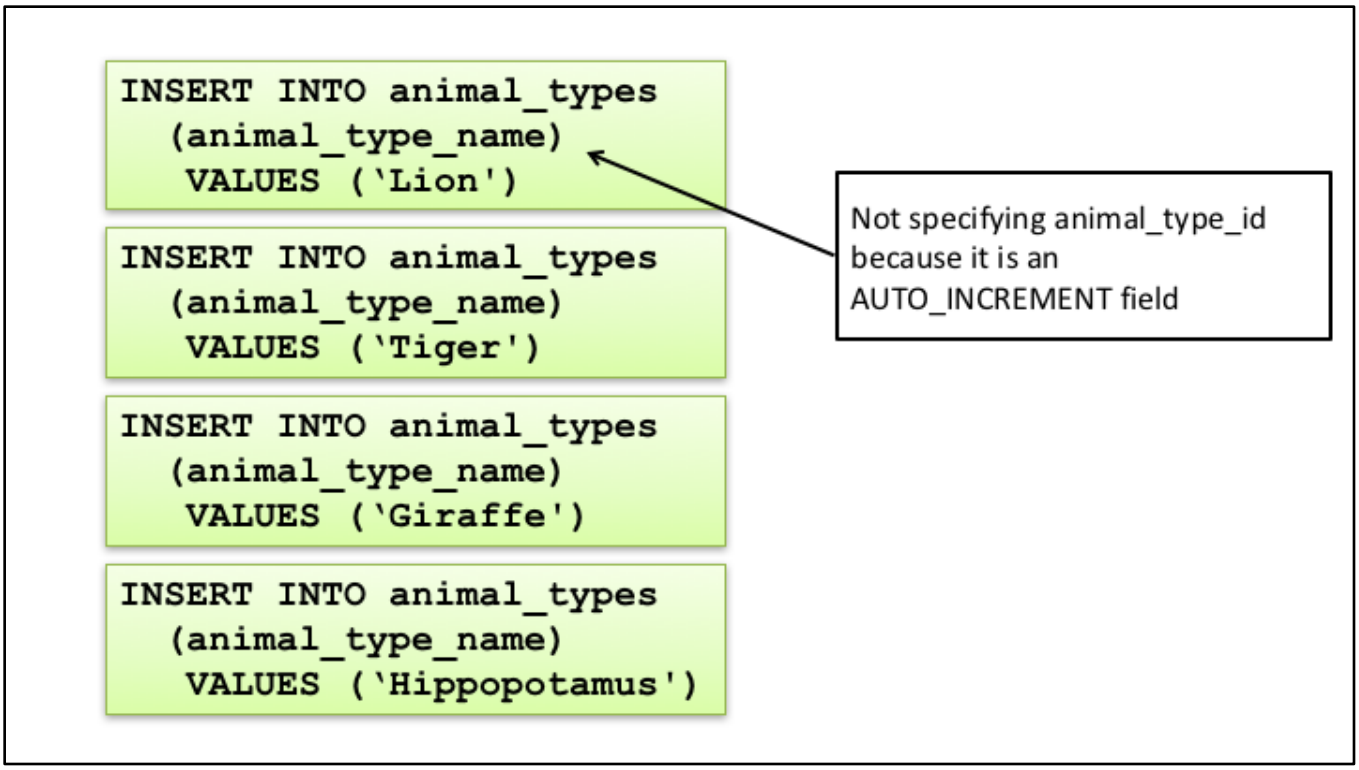

SQL
DDL -- Data Definition Language
DML -- Data Manipulation Language
CREATE
INSERT
SELECT
UPDATE
DELETE
Database
Table
Index
…

```
CREATE DATABASE zoo
```

```
USE zoo
```

```
CREATE TABLE animal_types
  (animal_type_id INT(5) NOT NULL AUTO_INCREMENT,
   animal_type_name VARCHAR(50) NOT NULL,
   PRIMARY KEY (animal_type_id))
```




INSERT INTO animal_types
(animal_type_name)
VALUES ('Lion')
INSERT INTO animal_types
(animal_type_name)
VALUES ('Tiger')
INSERT INTO animal_types
(animal_type_name)
VALUES ('Giraffe')
INSERT INTO animal_types
(animal_type_name)
VALUES ('Hippopotamus')
Not specifying animal_type_id because it is an AUTO_INCREMENT field

```
SELECT *
FROM animal_types
```

animal_type_id	animal_type_name
1	Lion
2	Tiger
3	Giraffe
4	Hippopotamus

```
CREATE TABLE animals
   (animal_id INT(5) NOT NULL AUTO_INCREMENT,
    animal_type_id INT(5) NOT NULL,
    animal_name VARCHAR(50) NOT NULL,
    animal_weight FLOAT,
    PRIMARY KEY(animal_id),
    CONSTRAINT fk1_animal_types
      FOREIGN KEY (animal_type_id)
      REFERENCES animal_types (animal_type_id))
```

```
INSERT INTO animals
(animal_type_id, animal_name, animal_weight)
VALUES (1, 'Lisa', 200)
```

```
INSERT INTO animals
(animal_type_id, animal_name, animal_weight)
VALUES (1, `Mona', 220)
```

```
INSERT INTO animals
(animal_type_id, animal_name, animal_weight)
VALUES (3, `Albert', 500)
```

```
INSERT INTO animals
(animal_type_id, animal_name, animal_weight)
VALUES (4, `Hippy', 1200)
```

```
SELECT *
FROM animals
```

animal_id	animal_type_id	animal_name	animal_weight
1	1	Lisa	200
2	1	Mona	220
3	3	Albert	500
4	4	Hippy	1200

```
SELECT animal_id, animal_type_name,
       animal_name, animal_weight
FROM animals a
JOIN animal_types at
ON a.animal_type_id = at.animal_type_id
```

animal_id	animal_type_name	animal_name	animal_weight
1	Lion	Lisa	200
2	Lion	Mona	220
3	Giraffe	Albert	500
4	Hippopotamus	Hippy	1200