Information Technology in Business and Society - SQL

Pearl Yu

A video: Relational Database

- 7 database paradigms

Queries &
Structured Querying Language

Objective:

Be able to write simple queries in SQL to answer business questions.

Querying relational databases

- Database query
  - Query: A **‘question’** you ask your database
  - Answer: A virtual table (with data coming potentially from multiple database tables) called a **view**.

- How to create queries
  - Most foundational approach:querying using a structured query language (SQL)

Querying relational databases - Math Foundation

NYU STERN SCHOOL OF BUSINESS

- These three simple operations define the whole functionality of SQL.
  - Select: a subset of rows (records)
  - Project: a subset of columns (fields)
  - Join: two tables together

- Every **view** is a result of a combination of select, project and/or join

Not SQL commands! These are math operations!

Relational Algebra represents the operations on relations, an algebra that consists of operations for constructing new relations from given relations.

Querying relational databases

- These three simple operations define the whole functionality of SQL.
  - **Select: a subset of rows (records)**
  - Project: a subset of columns (fields)
  - Join: two tables together

Again, not SQL commands!
These are math operations.

- Example: **Select** (Books costing $30.00 or less)

ISBN	Book Name	Author	Price
#0072952849	MIS in the Information Age	Haag, Stephen	$98.75
#0465039138	Inside Apple	Lessig, Lawrence	$25.00
#0738206083	Database Systems	Rheingold, Howard	$29.95
#0738206679	Alibaba's World	Barabasi, Albert-Laszlo	$34.95
#1234567890	Getting by at Stern	Author, Bookwriter	$25.00
#1573928895	Disruptive Innovation	Litman, Jessica	$55.00

Querying relational databases

- Queries use combinations of query ‘operators’.
  - **Select: a subset of rows (records)**
  - Project: a subset of columns (fields)
  - Join: two tables together

- Example: **Select** (Books costing $30.00 or less)

ISBN	Book Name	Author	Price
#0465039138	Inside Apple	Lessig, Lawrence	$25.00
#0738206083	Database Systems	Rheingold, Howard	$29.95
#1234567890	Getting by at Stern	Author, Bookwriter	$25.00

Querying relational databases

- Queries use combinations of query ‘operators’.
  - Select: a subset of rows (records)
  - **Project: a subset of columns (fields)**
  - Join: two tables together

Again, not SQL commands!
These are math operations.

- Example: **Project** (Book Name and Price)

Book Name
MIS in the Information Age
Inside Apple
Database Systems
Alibaba’s World
Getting by at Stern
Disruptive Innovation

Price
$98.75
$25.00
$29.95
$34.95
$25.00
$55.00

Querying relational databases

- Queries use combinations of query ‘operators’.
  - **Select: a subset of rows (records)**
  - **Project: a subset of columns (fields)**
  - Join: two tables together

- Example: **Select and Project** (Book Name and Price for books under $30.00)

Book Name	Price
Inside Apple	$25.00
Database Systems	$29.95
Getting by at Stern	$25.00

Querying relational databases

- Queries use combinations of query ‘operators’.
  - **Select: a subset of rows (records)**
  - **Project: a subset of columns (fields)**
  - Join: two tables together

- Example: **Select and Project** (Book Name and Price for books under $30.00)

Book Name	Price
Inside Apple	$25.00
Database Systems	$29.95
Getting by at Stern	$25.00

Querying relational databases

- Queries use combinations of query ‘operators’.
  - Select: a subset of rows (records)
  - Project: a subset of columns (fields)
  - **Join**: two tables together

- Example: **Join (**Order and Book**)**

Order

Order#	Customer ID	ISBN	Payment
1	C1001	#0465039138	Credit
2	C1004	#1573928895	Credit
3	C1002	#0072952849	Cash
4	C1003	#0738206679	Cash
5	C1003	#0738206083	Cash
6	C1001	#0738206083	Credit
7	C1002	#1573928895	Credit
8	C1001	#0738206679	Credit

Book

ISBN	Book Name	Author	Price
#0072952849	MIS in the Information Age	Haag, Stephen	$98.75
#0465039138	Inside Apple	Lessig, Lawrence	$25.00
#0738206083	Database Systems	Rheingold, How..	$29.95
#0738206679	Alibaba's World	Barabasi, Albert-	$34.95
#1234567890	Getting by at Stern	Author, Bookwri..	$25.00
#1573928895	Disruptive Innovation	Litman, Jessica	$55.00

Querying relational databases

- Example: **Join (**Order and Book**)**

Order

Order#	Customer ID	ISBN	Payment	Book Name	Author	Price
1	C1001	#0465039138	Credit	Inside Apple	Lessig, Lawrence	$25.00
2	C1004	#1573928895	Credit	Disruptive Innovation	Litman, Jessica	$55.00
3	C1002	#0072952849	Cash	MIS in the Information Age	Haag, Stephen	$98.75
4	C1003	#0738206679	Cash	Alibaba's World	Barabasi, Albert-	$34.95
5	C1003	#0738206083	Cash	Database Systems	Rheingold, How..	$29.95
6	C1001	#0738206083	Credit	Database Systems	Rheingold, How..	$29.95
7	C1002	#1573928895	Credit	Digital Copyright:..	Litman, Jessica	$55.00
8	C1001	#0738206679	Credit	Alibaba's World	Barabasi, Albert-	$34.95

Book

ISBN	Book Name	Author	Price
#0072952849	MIS in the Information Age	Haag, Stephen	$98.75
#0465039138	Inside Apple	Lessig, Lawrence	$25.00
#0738206083	Database Systems	Rheingold, How..	$29.95
#0738206679	Alibaba's World	Barabasi, Albert-	$34.95
#1234567890	Getting by at Stern	Author, Bookwri..	$25.00
#1573928895	Disruptive Innovation	Litman, Jessica	$55.00

Querying relational databases - Math Foundation

NYU STERN SCHOOL OF BUSINESS

- These three simple operations define the whole functionality of SQL.
  - **Select**: a subset of rows (records)
  - **Project**: a subset of columns (fields)
  - **Join**: two tables together

- Every **view** is a result of a combination of select, project and/or join

Not SQL commands! These are math operations!

Relational Algebra represents the operations on relations, an algebra that consists of operations for constructing new relations from given relations.

Querying relational databases

- **S**tructured **Q**uery **L**anguage
- Storing, manipulating, retrieving data in database
- SQL uses combinations of keywords and symbols

Querying relational databases

- All queries follow the same basic pattern:

```
SELECT [columns] FROM [table]
```

SQL keyword indicating from which table we will be selecting observations.

The database table we're selecting observations from

```
SELECT * FROM Customers
```

SQL keyword indicating a query in which we will be 'selecting' data from a table

Here we list which 'fields' we want from the table.
* indicates that we want all of them.
We can also list columns we want by name.

Querying relational databases

SQL Tryit Editor v1.6

http://goo.gl/iBpPLO

SQL

- **What are in the Customers table?**

```
SELECT * FROM Customers
```

SQL Practice

- Let's try:
- What are in the Products table?

```
SELECT *
FROM Products;
```

All of the columns and rows are returned.

NYU STERN SCHOOL OF BUSINESS

SQL Statement:

```
SELECT * FROM Products
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 77

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18
2	Chang	1	1	24 - 12 oz bottles	19

SQL

- Let's try:
- **Select particular columns:**
  For example, if we only want to see the list of product ID and price

```
SELECT ProductID, Price
FROM Products;
```

These two columns and all the rows are returned.

NYU STERN SCHOOL OF BUSINESS

SQL Statement:

```
SELECT * FROM Products
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 77

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18
2	Chang	1	1	24 - 12 oz bottles	19

SQL Practice

- Let's try:
- Select Customer name and address from Customers table

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country

```
SELECT CustomerName, Address
FROM Customers;
```

SQL Statement:

```
SELECT CustomerName, Address FROM Customers;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 91

CustomerName	Address
Alfreds Futterkiste	Obere Str. 57
Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222
Antonio Moreno Taquería	Mataderos 2312
Around the Horn	120 Hanover Sq.
Berglunds snabbköp	Berguvsvägen 8

SQL

- **Adding in some conditions**

```
SELECT *
FROM Customers
WHERE [condition]
```

This is called, a WHERE clause. This statement will only return **rows that meet your criteria**.

SQL

- Let's try:
- Find the cheaper products (price < $10)

```
SELECT *
FROM Products
WHERE Price < 10;
```

Eleven rows and all the columns are returned.

NYU STERN SCHOOL OF BUSINESS

SQL Statement:

```
SELECT CustomerName, Address FROM Customers;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 91

CustomerName	Address
Alfreds Futterkiste	Obere Str. 57
Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222
Antonio Moreno Taquería	Mataderos 2312
Around the Horn	120 Hanover Sq.
Berglunds snabbköp	Berguvsvägen 8

SQL Practice

- Let's try:
- What does this mean?

```
SELECT *
FROM Products
WHERE Price <> 10;
```

Does not equal in SQL grammar

NYU STERN SCHOOL OF BUSINESS

SQL Statement:

```
SELECT *
FROM Products
WHERE Price <> 10;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

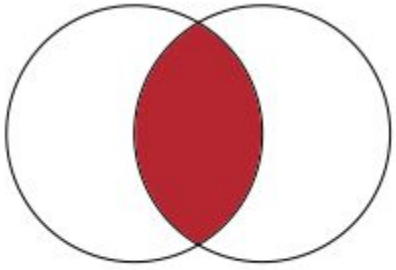
Number of Records: 74

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18
2	Chang	1	1	24 - 12 oz bottles	19
4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	22
5	Chef Anton's Gumbo Mix	2	2	36 boxes	21.35
6	Grandma's Boysenberry Spread	3	2	12 - 8 oz jars	25

SQL

- Now let's **combine conditions.**
- **AND**:
  - The AND Operator returns a row if for that row, all the conditions separated by AND are true.
- Let's try:

```
SELECT *
FROM Products
WHERE Price < 10 AND Price > 5;
```



NYU STERN SCHOOL OF BUSINESS

SQL Statement:

```
SELECT *
FROM Products
WHERE Price < 10 AND Price > 5;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

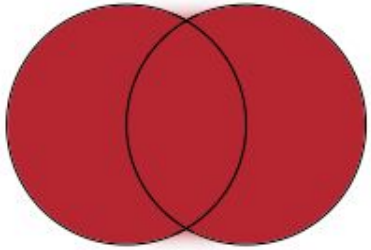
Result:

Number of Records: 9

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
13	Konbu	6	8	2 kg box	6
19	Teatime Chocolate Biscuits	8	3	10 boxes x 12 pieces	9.2
23	Tunnbröd	9	5	12 - 250 g pkgs.	9
41	Jack's New England Clam Chowder	19	8	12 - 12 oz cans	9.65
45	Røgede sild	21	8	1k pkg.	9.5
47	Zaanse koeken	22	3	10 - 4 oz boxes	9.5

SQL

- Now let's **combine conditions.**
- **OR**:
  - The OR Operator returns a row if for that row, any of the conditions separated by OR is true.



OR

- Let's try:

```
SELECT *
FROM Products
WHERE Price < 10 OR Price > 100;
```

SQL Statement:

```
SELECT *
FROM Products
WHERE Price < 10 OR Price > 100;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 13

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
13	Konbu	6	8	2 kg box	6
19	Teatime Chocolate Biscuits	8	3	10 boxes x 12 pieces	9.2
23	Tunnbröd	9	5	12 - 250 g pkgs.	9
24	Guaraná Fantástica	10	1	12 - 355 ml cans	4.5
29	Thüringer Rostbratwurst	12	6	50 bags x 30 sausgs.	123.79

SQL

- Now let's **combine AND, OR operators.**
- Let's try:

```
SELECT *
FROM Products
WHERE (Price >2 AND Price <10) OR Price = 97;
```

NYU STERN SCHOOL OF BUSINESS

SQL Statement:

```
SELECT *
FROM Products
WHERE (Price >2 AND Price <10) OR Price = 97;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 12

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
9	Mishi Kobe Niku	4	6	18 - 500 g pkgs.	97
13	Konbu	6	8	2 kg box	6
19	Teatime Chocolate Biscuits	8	3	10 boxes x 12 pieces	9.2
23	Tunnbröd	9	5	12 - 250 g pkgs.	9

SQL Practice

- Let's try:
- We want to find the suppliers located in 'Boston'

SupplierID	SupplierName	ContactName	Address	City	PostalCode	Country	Phone

```
SELECT *
FROM Suppliers
WHERE City = 'Boston';
```

A string needs to be in quotation marks.

Only one supplier is in Boston.

NYU STERN SCHOOL OF BUSINESS

SQL Statement:

```
SELECT *
FROM Suppliers
WHERE City = 'Boston';
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 1

SupplierID	SupplierName	ContactName	Address	City	PostalCode	Country	Phone
19	New England Seafood Cannery	Robb Merchant	Order Processing Dept. 2100 Paul Revere Blvd.	Boston	02134	USA	(617) 555-3267

SQL Practice

- Please come up with some questions (that you want to ask the database and we can write the query with things learnt so far), and try them.

- For example:
  - What’re the names of the customers?
  - What is the background of the employee Nancy?
  - …

Type in your question, your query on the jamboard. jamboard

SQL

- Now let's look at **functions.**
  - Function is a computational activity.
- For example, how many products have a price less than $10?

```
SELECT COUNT(ProductId)
FROM Products
WHERE Price < 10;
```

COUNT is a function that counts the number of rows that satisfy the criteria specified with the WHERE clause.

NYU STERN SCHOOL OF BUSINESS

SQL Statement:

```
SELECT COUNT(ProductId)
FROM Products
WHERE Price < 10;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 1

COUNT(ProductId)
11

SQL

- Now let's look at **functions.**
  - Function is a computational activity.
- For example, how many products have a price less than $10?

```
SELECT COUNT(ProductId)
FROM Products
WHERE Price < 10;
```

- Other functions: sum, avg, min, max…

COUNT is a function that counts the number of rows that satisfy the criteria specified with the WHERE clause.

SQL

- Let's try:
- How many products have a price between 20 and 30?

```
SELECT COUNT(ProductId)
FROM Products
WHERE Price >20 AND Price <30;
```

- Output:

Number of Records: 1

COUNT (ProductID)
12

COUNT (ProductId) is not a very intuitive name.

SQL

- We can also **rename the column**.
- **AS**:

```
SELECT COUNT(ProductId) AS NumProducts
FROM Products
WHERE Price < 10;
```

- Output:

NumProducts
14

SQL

- Let's try.
- In the output table, create a new column taking 10% off each price?

```
SELECT Price,Price * 0.9 AS SalePrice
FROM Products;
```

SQL

- Output:

Number of Records: 77

Price	SalePrice
18	16.2
19	17.1
10	9
22	19.8
21.35	19.215000000000003
25	22.5
30	27
40	36
97	87.3

SQL

- Let's try another function, and the rename operator.
- Find average price of the products that have price greater than $20?
  - Rename a field as AvgPrice

```
SELECT AVG(Price) AS AvgPrice
FROM Products
WHERE Price > 20;
```

- Output:

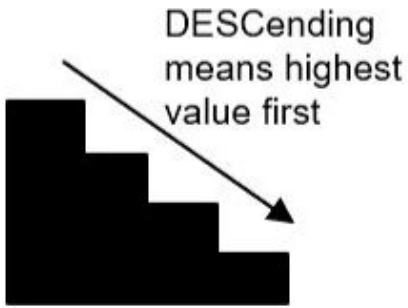
Number of Records: 1

AvgPrice
45.813783783783784

SQL

- Let's try the **ORDER BY** command.
- How can we **sort** the products by price?

```
SELECT *
FROM Products
ORDER BY Price DESC;
```



The field to order the results.

DESC means descending order.
ASC means ascending order.

SQL

- Let's try.
- Order product name alphabetically (ASC order)?

```
SELECT *
FROM Products
ORDER BY ProductName ASC;
```

SQL

- Output:

Number of Records: 77

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
17	Alice Mutton	7	6	20 - 1 kg tins	39
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10
40	Boston Crab Meat	19	8	24 - 4 oz tins	18.4
60	Camembert Pierrot	28	4	15 - 300 g rounds	34

SQL

- Let's try the **LIKE** function.
- How to **search** for the rows that contain particular words in a column?

```
SELECT *
FROM [table]
WHERE [column] LIKE [pattern];
```

This allows you to search a pattern within fields.

SQL

- Let's try the **LIKE** function.
- Let's find the rows that contains data ending with 'bottles' in Unit column.

```
SELECT *
FROM Products
WHERE Unit LIKE '%bottles';
```

Inside the quotation marks, **%** is a **wildcard** character. It can represent anything (zero, one,multiple characters, or nothing).

Pay attention to the **location** of the **%**:

x%: anything can appear after x (xtra, x, xtreme..)

%x%: anything before or after x (piexi,xlevel, tjmaxx)

SQL

- Output:

Number of Records: 11

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
2	Chang	1	1	24 - 12 oz bottles	19
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10
15	Genen Shouyu	6	2	24 - 250 ml bottles	15.5
34	Sasquatch Ale	16	1	24 - 12 oz bottles	14
35	Steeleye Stout	16	1	24 - 12 oz bottles	18
38	Côte de Blaye	18	1	12 - 75 cl bottles	263.5
61	Sirop d'érable	29	2	24 - 500 ml bottles	28.5
65	Louisiana Fiery Hot Pepper Sauce	2	2	32 - 8 oz bottles	21.05
67	Laughing Lumberjack Lager	16	1	24 - 12 oz bottles	14
70	Outback Lager	7	1	24 - 355 ml bottles	15
75	Rhönbräu Klosterbier	12	1	24 - 0.5 l bottles	7.75

SQL Practice

- Let's try it with the Customers table.
- Find the ID and PostalCode of Customers whose PostalCode includes '5'.

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country

```
SELECT CustomerID, PostalCode
FROM Customers
WHERE PostalCode LIKE '%5%';
```

SQL

- Output:

Number of Records: 30

CustomerID	PostalCode
2	05021
3	05023
5	S-958 22
11	EC2 5NT
13	05022
15	05432-043
17	52066
21	05442-030
23	59000
25	80805
28	1675
34	05454-876
35	5022
44	60528
46	3508
55	99508
56	50739

SQL Practice

- Let's try it with **GROUP BY** command.
  - GROUP BY groups rows with the same values into a summary row.
- What're the different countries where the customers are located?

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country

```
SELECT Country
FROM Customers
GROUP BY Country;
```

SQL

- Output:

Number of Records: 21

Country
Argentina
Austria
Belgium
Brazil
Canada
Denmark
Finland
France
Germany
Ireland
Italy
Mexico
Norway
Poland
Portugal
Spain
Sweden
Switzerland
UK
USA
Venezuela

SQL

- Let's combine **COUNT** and **GROUP BY**.
- How many customers from each country?

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country

```
SELECT Country, COUNT(CustomerID)
FROM Customers
GROUP BY Country;
```

First GROUP BY group rows based on country. Then the function operates on the data underlying the aggregation.

SQL

- Output:

Number of Records: 21

Country	COUNT(CustomerName)
Argentina	3
Austria	2
Belgium	2
Brazil	9
Canada	3
Denmark	2
Finland	2
France	11
Germany	11
Ireland	1
Italy	3
Mexico	5
Norway	1
Poland	1
Portugal	2
Spain	5
Sweden	2
Switzerland	2
UK	7
USA	13
Venezuela	4