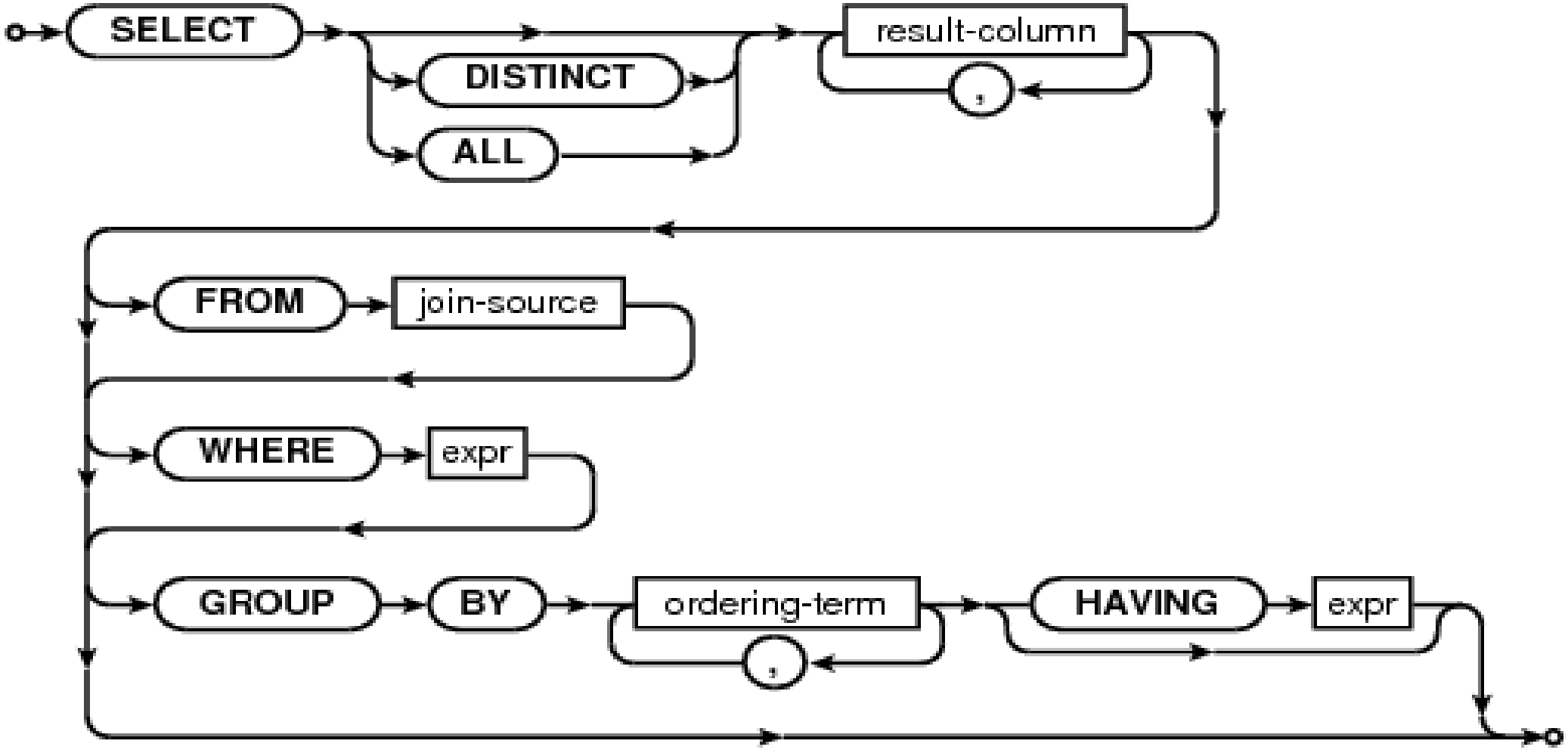
Data Warehouse

Sirojul Munir S.SI, M.KOM – Semester Genap TA 20182

SQL for Data Warehouse

SQL: SELECT Syntax



SQL SELECT: Basic

CUSTOMER
Id
FirstName
LastName
City
Country
Phone

SQL SELECT Examples

Problem: List all customers

```
1. SELECT *
2.   FROM Customer
```

Results: 91 records

Id	FirstName	LastName	City	Country	Phone
1	Maria	Anders	Berlin	Germany	030-0074321
2	Ana	Trujillo	México D.F.	Mexico	(5) 555-4729
3	Antonio	Moreno	México D.F.	Mexico	(5) 555-3932
4	Thomas	Hardy	London	UK	(171) 555-7788
5	Christina	Berglund	Luleå	Sweden	0921-12 34 65

SQL SELECT: Basic

Problem: List the first name, last name, and city of all customers

CUSTOMER
Id
FirstName
LastName
City
Country
Phone

```
1.  SELECT FirstName, LastName, City
2.    FROM Customer
```

Results: 91 records

FirstName	LastName	City
Maria	Anders	Berlin
Ana	Trujillo	México D.F.
Antonio	Moreno	México D.F.
Thomas	Hardy	London
Christina	Berglund	Luleå
...		

SQL SELECT: WHERE

SQL WHERE Clause Examples

CUSTOMER
Id
FirstName
LastName
City
Country
Phone

Problem: List the customers in Sweden

```
1.  SELECT Id, FirstName, LastName, City, Country, Phone
2.    FROM Customer
3.   WHERE Country = 'Sweden'
```

Results: 2 records

Id	FirstName	LastName	City	Country	Phone
5	Christina	Berglund	Luleå	Sweden	0921-12 34 65
24	Maria	Larsson	Bräcke	Sweden	0695-34 67 21

SQL SELECT: UPDATE , DELETE

A WHERE clause with an UPDATE statement:

```
UPDATE table-name
   SET column-name = value
 WHERE condition
```

A WHERE clause with a DELETE statement:

```
DELETE table-name
 WHERE condition
```

Problem: Update the city to Sydney for supplier Pavlova, Ltd.

```
UPDATE Supplier
   SET City = 'Sydney'
 WHERE Name = 'Pavlova, Ltd.'
```

Results: 1 record updated.

SUPPLIER
Id
CompanyName
ContactName
City
Country
Phone
Fax

Problem: Delete all products with unit price higher than $50.

```
DELETE FROM Product
 WHERE UnitPrice > 50
```

Results: 7 records deleted.

PRODUCT
Id
ProductName
SupplierId
UnitPrice
Package
IsDiscontinued

SQL SELECT: ORDER BY

CUSTOMER
Id
FirstName
LastName
City
Country
Phone

SQL ORDER BY Examples

Problem: List all suppliers in alphabetical order

```
SELECT CompanyName, ContactName, City, Country
  FROM Supplier
 ORDER BY CompanyName
```

The default sort order is ascending, that is, low-high or a-z.

Results: 29 records

Id	CompanyName	ContactName	City	Country
18	Aux joyeux ecclésiastiques	Guylène Nodier	Paris	France
16	Bigfoot Breweries	Cheryl Saylor	Bend	USA
5	Cooperativa de Quesos 'Las Cabras'	Antonio del Valle Saavedra	Oviedo	Spain
27	Escargots Nouveaux	Marie Delamare	Montceau	France
1	Exotic Liquids	Charlotte Cooper	London	UK
...				

SQL SELECT: ORDER BY

CUSTOMER
Id
FirstName
LastName
City
Country
Phone

SQL ORDER BY Examples

Problem: List all suppliers in alphabetical order

```
SELECT CompanyName, ContactName, City, Country
  FROM Supplier
 ORDER BY CompanyName
```

The default sort order is ascending, that is, low-high or a-z.

Results: 29 records

Id	CompanyName	ContactName	City	Country
18	Aux joyeux ecclésiastiques	Guylène Nodier	Paris	France
16	Bigfoot Breweries	Cheryl Saylor	Bend	USA
5	Cooperativa de Quesos 'Las Cabras'	Antonio del Valle Saavedra	Oviedo	Spain
27	Escargots Nouveaux	Marie Delamare	Montceau	France
1	Exotic Liquids	Charlotte Cooper	London	UK
...				

SQL SELECT: ORDER BY DESC

CUSTOMER
Id
FirstName
LastName
City
Country
Phone

Problem: List all suppliers in **reverse** alphabetical order

```
1.  SELECT CompanyName, ContactName, City, Country
2.    FROM Supplier
3.   ORDER BY CompanyName DESC
```

The keyword DESC denotes descending, i.e., reverse order.

Results: 29 records

Id	CompanyName	ContactName	City	Country
22	Zaanse Snoepfabriek	Dirk Luchte	Zaandam	Netherlands
4	Tokyo Traders	Yoshi Nagase	Tokyo	Japan
17	Svensk Sjöföda AB	Michael Björn	Stockholm	Sweden
8	Specialty Biscuits, Ltd.	Peter Wilson	Manchester	UK
10	Refrescos Americanas LTDA	Carlos Diaz	Sao Paulo	Brazil

SQL SELECT: ORDER BY DESC

SUPPLIER
Id
CompanyName
ContactName
City
Country
Phone
Fax

Problem: List all suppliers in the USA, Japan, and Germany, ordered by city, then by company name in reverse order

```
SELECT Id, CompanyName, City, Country
  FROM Supplier
 WHERE Country IN ('USA', 'Japan', 'Germany')
 ORDER BY Country ASC, CompanyName DESC
```

This shows that you can order by more than one column.
ASC denotes ascending, but is optional as it is the default sort order.

Results: 9 records

Id	CompanyName	City	Country
12	Plutzer Lebensmittelgroßmärkte AG	Frankfurt	Germany
13	Nord-Ost-Fisch Handelsgesellschaft mbH	Cuxhaven	Germany
11	Heli Süßwaren GmbH & Co. KG	Berlin	Germany
4	Tokyo Traders	Tokyo	Japan
6	Mayumi's	Osaka	Japan

SQL SELECT: DISTINCT

SUPPLIER
Id
CompanyName
ContactName
City
Country
Phone
Fax

SQL SELECT Examples

Problem: List all supplier countries in alphabetical order.

```
1. SELECT DISTINCT Country
2.   FROM Supplier
3. ORDER BY COUNTRY
```

Results: 16 rows

Country
Australia
Brazil
Canada
Denmark

SQL SELECT: MIN,MAX

The general MIN syntax is:

```
SELECT MIN(column-name)
  FROM table-name
```

The general MAX syntax is:

```
SELECT MAX(column-name)
  FROM table-name
```

Problem: Find the cheapest product

```
SELECT MIN(UnitPrice)
  FROM Product
```

Results:

UnitPrice
2.50

PRODUCT
Id
ProductName
SupplierId
UnitPrice
Package
IsDiscontinued

Problem: Find the largest order placed in 2014

```
SELECT MAX(TotalAmount)
  FROM [Order]
 WHERE YEAR(OrderDate) = 2014
```

Results:

TotalAmount
17250.00

ORDER
Id
OrderDate
OrderNumber
CustomerId
TotalAmount

SQL SELECT: AGGREGATE FUNCTION

SQL SELECT COUNT, SUM, and AVG Examples

Problem: Find the number of customers

```
SELECT COUNT(Id)
  FROM Customer
```

Results:

Count
91

Problem: Compute the total amount sold in 2013

```
SELECT SUM(TotalAmount)
  FROM [Order]
 WHERE YEAR(OrderDate) = 2013
```

Results:

Sum
658388.75

CUSTOMER
Id
FirstName
LastName
City
Country
Phone

Problem: Compute the average size of all orders

```
SELECT AVG(TotalAmount)
  FROM [Order]
```

Results:

Average
1631.877819

ORDER
Id
OrderDate
OrderNumber
CustomerId
TotalAmount

SELECT WHERE: AND, OR , NOT

A WHERE clause with AND:

```
SELECT column-names
  FROM table-name
 WHERE condition1 AND condition2
```

A WHERE clause with OR:

```
UPDATE table-name
   SET column-name = value
 WHERE condition1 OR condition2
```

A WHERE clause with NOT:

```
DELETE table-name
 WHERE NOT condition
```

SQL WHERE with AND, OR, and NOT Examples

CUSTOMER
Id
FirstName
LastName
City
Country
Phone

Problem: Get customer named Thomas Hardy

```
SELECT Id, FirstName, LastName, City, Country
  FROM Customer
 WHERE FirstName = 'Thomas' AND LastName = 'Hardy'
```

Results: 1 record.

Id	FirstName	LastName	City	Country
4	Thomas	Hardy	London	UK

SELECT WHERE: AND, OR , NOT

A WHERE clause with AND:

```
SELECT column-names
  FROM table-name
 WHERE condition1 AND condition2
```

A WHERE clause with OR:

```
UPDATE table-name
   SET column-name = value
 WHERE condition1 OR condition2
```

A WHERE clause with NOT:

```
DELETE table-name
 WHERE NOT condition
```

Problem: List all customers from Spain or France

```
SELECT Id, FirstName, LastName, City, Country
  FROM Customer
 WHERE Country = 'Spain' OR Country = 'France'
```

CUSTOMER
Id
FirstName
LastName
City
Country
Phone

Results: 16 records.

Id	FirstName	LastName	City	Country
7	Frédérique	Citeaux	Strasbourg	France
8	Martín	Sommer	Madrid	Spain
9	Laurence	Lebihan	Marseille	France
18	Janine	Labrune	Nantes	France
22	Diego	Roel	Madrid	Spain
23	Martine	Rancé	Lille	France

SELECT WHERE: AND, OR , NOT

A WHERE clause with AND:

```
SELECT column-names
  FROM table-name
 WHERE condition1 AND condition2
```

A WHERE clause with OR:

```
UPDATE table-name
   SET column-name = value
 WHERE condition1 OR condition2
```

A WHERE clause with NOT:

```
DELETE table-name
 WHERE NOT condition
```

Problem: List all customers that are not from the USA

```
SELECT Id, FirstName, LastName, City, Country
  FROM Customer
 WHERE NOT Country = 'USA'
```

CUSTOMER
Id
FirstName
LastName
City
Country
Phone

Results: 78 records.

Id	FirstName	LastName	City	Country
1	Maria	Anders	Berlin	Germany
2	Ana	Trujillo	México D.F.	Mexico
3	Antonio	Moreno	México D.F.	Mexico
4	Thomas	Hardy	London	UK
5	Christina	Berglund	Luleå	Sweden
6	Hanna	Moos	Mannheim	Germany
7	Frédérique	Citeaux	Strasbourg	France

SELECT WHERE: AND, OR , NOT

A WHERE clause with AND:

```
SELECT column-names
  FROM table-name
 WHERE condition1 AND condition2
```

A WHERE clause with OR:

```
UPDATE table-name
   SET column-name = value
 WHERE condition1 OR condition2
```

A WHERE clause with NOT:

```
DELETE table-name
 WHERE NOT condition
```

Problem: List all orders that not between $50 and $15000

```
SELECT Id, OrderDate, CustomerId, TotalAmount
  FROM [Order]
 WHERE NOT (TotalAmount >= 50 AND TotalAmount <= 15000)
 ORDER BY TotalAmount DESC
```

ORDER
Id
OrderDate
OrderNumber
CustomerId
TotalAmount

Results: 16 records.

Id	OrderDate	CustomerId	TotalAmount
618	2/2/2014 12:00:00 AM	63	17250.00
783	4/17/2014 12:00:00 AM	71	16321.90
734	3/27/2014 12:00:00 AM	34	15810.00
175	1/22/2013 12:00:00 AM	27	49.80
24	8/1/2012 12:00:00 AM	75	48.00

SQL SELECT: BETWEEN

The general syntax is:

```
SELECT column-names
  FROM table-name
 WHERE column-name BETWEEN value1 AND value2
```

PRODUCT
Id
ProductName
SupplierId
UnitPrice
Package
IsDiscontinued

SQL WHERE BETWEEN Examples

Problem: List all products between $10 and $20

```
SELECT Id, ProductName, UnitPrice
  FROM Product
 WHERE UnitPrice BETWEEN 10 AND 20
 ORDER BY UnitPrice
```

Results: 29 records.

Id	ProductName	UnitPrice
3	Aniseed Syrup	10.00
21	Sir Rodney's Scones	10.00
74	Longlife Tofu	10.00
46	Spegesild	12.00
31	Gorgonzola Telino	12.50

SQL SELECT: BETWEEN

The general syntax is:

```
SELECT column-names
  FROM table-name
 WHERE column-name BETWEEN value1 AND value2
```

Problem: List all products not between $10 and $100 sorted by price.

```
SELECT Id, ProductName, UnitPrice
  FROM Product
 WHERE UnitPrice NOT BETWEEN 5 AND 100
 ORDER BY UnitPrice
```

PRODUCT
Id
ProductName
SupplierId
UnitPrice
Package
IsDiscontinued

Results: 4 records.

Id	ProductName	UnitPrice
33	Geitost	2.50
24	Guaraná Fantástica	4.50
29	Thüringer Rostbratwurst	123.79
38	Côte de Blaye	263.50

SQL SELECT: BETWEEN

The general syntax is:

```
SELECT column-names
  FROM table-name
 WHERE column-name BETWEEN value1 AND value2
```

Problem: Get the number of orders and amount sold between Jan 1, 2013 and Jan 31, 2013.

```
SELECT COUNT(Id), SUM(TotalAmount)
  FROM [Order]
 WHERE OrderDate BETWEEN '1/1/2013' AND '1/31/2013'
```

PRODUCT
Id
ProductName
SupplierId
UnitPrice
Package
IsDiscontinued

Results:

Count	TotalAmount
33	66692.80

SQL SELECT: IN

The general syntax is:

```
SELECT column-names
  FROM table-name
 WHERE column-name IN (values)
```

SQL WHERE IN Examples

PRODUCT
Id
ProductName
SupplierId
UnitPrice
Package
IsDiscontinued

Problem: List all suppliers from the USA, UK, OR Japan

```
SELECT Id, CompanyName, City, Country
  FROM Supplier
 WHERE Country IN ('USA', 'UK', 'Japan')
```

Results: 8 records.

Id	CompanyName	City	Country
1	Exotic Liquids	London	UK
2	New Orleans Cajun Delights	New Orleans	USA
3	Grandma Kelly's Homestead	Ann Arbor	USA
4	Tokyo Traders	Tokyo	Japan
6	Mayumi's	Osaka	Japan

SQL SELECT: IN

The general syntax is:

```
SELECT column-names
  FROM table-name
 WHERE column-name IN (values)
```

PRODUCT
Id
ProductName
SupplierId
UnitPrice
Package
IsDiscontinued

Problem: List all products that are not exactly $10, $20, $30, $40, or $50

```
SELECT Id, ProductName, UnitPrice
  FROM Product
 WHERE UnitPrice NOT IN (10,20,30,40,50)
```

Results: 72 records.

Id	ProductName	UnitPrice
1	Chai	18.00
2	Chang	19.00
4	Chef Anton's Cajun Seasoning	22.00
5	Chef Anton's Gumbo Mix	21.35
6	Grandma's Boysenberry Spread	25.00

SQL SELECT: IN

The general syntax is:

```
SELECT column-names
  FROM table-name
 WHERE column-name IN (values)
```

Problem: List all customers that are from the same countries as the suppliers.

```
SELECT Id, FirstName, LastName, Country
  FROM Customer
 WHERE Country IN
       (SELECT Country
          FROM Supplier)
```

CUSTOMER
Id
FirstName
LastName
City
Country
Phone

SUPPLIER
Id
CompanyName
ContactName
City
Country
Phone
Fax

Results: 91 records.

Id	FirstName	LastName	Country
1	Maria	Anders	Germany
4	Thomas	Hardy	UK
5	Christina	Berglund	Sweden
6	Hanna	Moos	Germany

SQL SELECT: LIKE

The general syntax is:

```
SELECT column-names
  FROM table-name
 WHERE column-name LIKE value
```

Optional *Wildcard* characters allowed in 'value' are % (percent) and _ (underscore).

- A % matches any string with zero or more characters.
- An _ matches any single character.

SQL WHERE LIKE Examples

PRODUCT
Id
ProductName
SupplierId
UnitPrice
Package
IsDiscontinued

Problem: List all products with names that start with 'Ca'

```
SELECT Id, ProductName, UnitPrice, Package
  FROM Product
 WHERE ProductName LIKE 'Ca%'
```

Results: 2 records.

Id	ProductName	UnitPrice	Package
18	Carnarvon Tigers	62.50	16 kg pkg.
60	Camembert Pierrot	34.00	15-300 g rounds

SQL SELECT: LIKE

The general syntax is:

```
SELECT column-names
  FROM table-name
 WHERE column-name LIKE value
```

PRODUCT
Id
ProductName
SupplierId
UnitPrice
Package
IsDiscontinued

Problem: List all products that start with 'Cha' or 'Chan' and have one more character.

```
SELECT Id, ProductName, UnitPrice, Package
  FROM Product
 WHERE ProductName LIKE 'Cha_' OR ProductName LIKE 'Chan_'
```

Optional *Wildcard* characters allowed in 'value' are % (percent) and _ (underscore).

- A % matches any string with zero or more characters.
- An _ matches any single character.

Results: 2 records.

Id	ProductName	UnitPrice	Package
1	Chai	18.00	10 boxes x 20 bags
2	Chang	19.00	24 - 12 oz bottles

SQL SELECT: NULL

```
SELECT column-names
  FROM table-name
 WHERE column-name IS NULL
```

The general not null syntax is:

```
SELECT column-names
  FROM table-name
  WHERE column-name IS NOT NULL
```

SQL WHERE IS NULL Examples

SUPPLIER
Id
CompanyName
ContactName
City
Country
Phone
Fax

Problem: List all suppliers that have no fax number

```
SELECT Id, CompanyName, Phone, Fax
  FROM Supplier
 WHERE Fax IS NULL
```

Results: 16 records

Id	CompanyName	Phone	Fax
1	Exotic Liquids	(171) 555-2222	NULL
2	New Orleans Cajun Delights	(100) 555-4822	NULL
4	Tokyo Traders	(03) 3555-5011	NULL

SQL SELECT: NULL

```
1. SELECT column-names
2.   FROM table-name
3.  WHERE column-name IS NULL
```

The general not null syntax is:

```
1. SELECT column-names
2.   FROM table-name
3.   WHERE column-name IS NOT NULL
```

Problem: List all suppliers that do have a fax number

```
1. SELECT Id, CompanyName, Phone, Fax
2.   FROM Supplier
3.  WHERE Fax IS NOT NULL
```

SUPPLIER
Id
CompanyName
ContactName
City
Country
Phone
Fax

Results: 13 records

Id	CompanyName	Phone	Fax
3	Grandma Kelly's Homestead	(313) 555-5735	(313) 555-3349
7	Pavlova, Ltd.	(03) 444-2343	(03) 444-6588
9	PB Knäckebröd AB	031-987 65 43	031-987 65 91
13	Nord-Ost-Fisch Handelsgesellschaft mbH	(04721) 8713	(04721) 8714

SQL SELECT: GROUP BY

The general syntax is:

```
SELECT column-names
  FROM table-name
 WHERE condition
 GROUP BY column-names
```

The general syntax with ORDER BY is:

```
SELECT column-names
  FROM table-name
 WHERE condition
 GROUP BY column-names
 ORDER BY column-names
```

SQL GROUP BY Examples

Problem: List the number of customers in each country

```
SELECT COUNT(Id), Country
  FROM Customer
 GROUP BY Country
```

Results: 21 records.

Count	Country
3	Argentina
2	Austria
2	Belgium
9	Brazil

CUSTOMER
Id
FirstName
LastName
City
Country
Phone

SQL SELECT: GROUP BY

The general syntax is:

```
SELECT column-names
  FROM table-name
 WHERE condition
 GROUP BY column-names
```

The general syntax with ORDER BY is:

```
SELECT column-names
  FROM table-name
 WHERE condition
 GROUP BY column-names
 ORDER BY column-names
```

Problem: List the number of customers in each country sorted high to low

```
SELECT COUNT(Id), Country
  FROM Customer
 GROUP BY Country
 ORDER BY COUNT(Id) DESC
```

Results: 21 records.

Count	Country
13	USA
11	France
11	Germany
9	Brazil
7	UK

CUSTOMER
Id
FirstName
LastName
City
Country
Phone

SQL SELECT: GROUP BY

The general syntax is:

```
SELECT column-names
  FROM table-name
 WHERE condition
 GROUP BY column-names
```

The general syntax with ORDER BY is:

```
SELECT column-names
  FROM table-name
 WHERE condition
 GROUP BY column-names
 ORDER BY column-names
```

Problem: List the total amount ordered for each customer

```
SELECT SUM(O.TotalPrice), C.FirstName, C.LastName
  FROM [Order] O JOIN Customer C
    ON O.CustomerId = C.Id
 GROUP BY C.FirstName, C.LastName
 ORDER BY SUM(O.TotalPrice) DESC
```

ORDER
Id
OrderDate
OrderNumber
CustomerId
TotalAmount

CUSTOMER
Id
FirstName
LastName
City
Country
Phone

This query uses a JOIN with Customer to obtain customer names

Results: 89 records.

Sum	FirstName	LastName
117483.39	Horst	Kloss
115673.39	Jose	Pavarotti
113236.68	Roland	Mendel
57317.39	Patricia	McKenna
52245.90	Paula	Wilson

SQL SELECT: HAVING

The general syntax is:

```
SELECT column-names
  FROM table-name
 WHERE condition
 GROUP BY column-names
HAVING condition
```

The general syntax with ORDER BY is:

```
SELECT column-names
  FROM table-name
 WHERE condition
 GROUP BY column-names
HAVING condition
 ORDER BY column-names
```

CUSTOMER
Id
FirstName
LastName
City
Country
Phone

SQL GROUP BY Examples

Problem: List the number of customers in each country. Only include countries with more than 10 customers.

```
SELECT COUNT(Id), Country
  FROM Customer
 GROUP BY Country
HAVING COUNT(Id) > 10
```

Results: 3 records

Count	Country
11	France
11	Germany
13	USA

SQL SELECT: HAVING

The general syntax is:

```
SELECT column-names
  FROM table-name
 WHERE condition
 GROUP BY column-names
HAVING condition
```

The general syntax with ORDER BY is:

```
SELECT column-names
  FROM table-name
 WHERE condition
 GROUP BY column-names
HAVING condition
 ORDER BY column-names
```

Problem: List the number of customers in each country, except the USA, sorted high to low. Only include countries with 9 or more customers.

```
SELECT COUNT(Id), Country
  FROM Customer
 WHERE Country <> 'USA'
 GROUP BY Country
HAVING COUNT(Id) >= 9
 ORDER BY COUNT(Id) DESC
```

Results: 3 records

Count	Country
11	France
11	Germany
9	Brazil

CUSTOMER
Id
FirstName
LastName
City
Country
Phone

SQL SELECT: HAVING

The general syntax is:

```
SELECT column-names
  FROM table-name
 WHERE condition
 GROUP BY column-names
HAVING condition
```

The general syntax with ORDER BY is:

```
SELECT column-names
  FROM table-name
 WHERE condition
 GROUP BY column-names
HAVING condition
 ORDER BY column-names
```

Problem: List all customer with average orders between $1000 and $1200.

ORDER
Id
OrderDate
OrderNumber
CustomerId
TotalAmount

```
SELECT AVG(TotalAmount), FirstName, LastName
  FROM [Order] O JOIN Customer C ON O.CustomerId = C.Id
 GROUP BY FirstName, LastName
HAVING AVG(TotalAmount) BETWEEN 1000 AND 1200
```

Results: 10 records

Average	FirstName	LastName
1081.215000	Miguel	Angel Paolino
1063.420000	Isabel	de Castro
1008.440000	Alexander	Feuer
1062.038461	Thomas	Hardy

SQL SELECT: ALIAS

The general syntax is:

```
SELECT column-name AS alias-name
  FROM table-name alias-name
 WHERE condition
```

SQL Alias Examples

CUSTOMER
Id
FirstName
LastName
City
Country
Phone

Problem: List total customers in each country.
Display results with easy to understand column headers.

```
SELECT COUNT(C.Id) AS TotalCustomers, C.Country AS Nation
  FROM Customer C
 GROUP BY C.Country
```

TotalCustomers and Nation are column aliases.
The table alias (C) in this example is not particularly useful.

SQL SELECT: ALIAS

The general syntax is:

```
SELECT column-name AS alias-name
  FROM table-name alias-name
 WHERE condition
```

ORDER
Id
OrderDate
OrderNumber
CustomerId
TotalAmount

CUSTOMER
Id
FirstName
LastName
City
Country
Phone

Problem: List the total amount ordered by customer with easy to read column headers

```
SELECT C.Id AS Identifier, C.LastName + ', ' + C.FirstName AS CustomerName,
       SUM(O.TotalAmount) AS TotalSpent
  FROM [Order] O JOIN Customer C ON O.CustomerId = C.Id
 GROUP BY C.Id, C.LastName + ', ' + C.FirstName
 ORDER BY TotalSpent DESC
```

The aliases significantly simplify writing the JOIN and ORDER BY clauses.
The C alias in C.Id helps identify the Customer Id rather then the Order Id.

Results: 89 records

Identifier	CustomerName	TotalSpent
63	Kloss, Horst	117483.39
71	Pavarotti, Jose	115673.39
20	Mendel, Roland	113236.68

CHEAT SHEET : SQL

SQL CHEAT SHEET http://www.sqltutorial.org

QUERYING DATA FROM A TABLE

```
SELECT c1, c2 FROM t;
```
Query data in columns c1, c2 from a table

```
SELECT * FROM t;
```
Query all rows and columns from a table

```
SELECT c1, c2 FROM t
WHERE condition;
```
Query data and filter rows with a condition

```
SELECT DISTINCT c1 FROM t
WHERE condition;
```
Query distinct rows from a table

```
SELECT c1, c2 FROM t
ORDER BY c1 ASC [DESC];
```
Sort the result set in ascending or descending order

```
SELECT c1, c2 FROM t
ORDER BY c1
LIMIT n OFFSET offset;
```
Skip *offset* of rows and return the next n rows

```
SELECT c1, aggregate(c2)
FROM t
GROUP BY c1;
```
Group rows using an aggregate function

```
SELECT c1, aggregate(c2)
FROM t
GROUP BY c1
HAVING condition;
```
Filter groups using HAVING clause

QUERYING FROM MULTIPLE TABLES

```
SELECT c1, c2
FROM t1
INNER JOIN t2 ON condition;
```
Inner join t1 and t2

```
SELECT c1, c2
FROM t1
LEFT JOIN t2 ON condition;
```
Left join t1 and t1

```
SELECT c1, c2
FROM t1
RIGHT JOIN t2 ON condition;
```
Right join t1 and t2

```
SELECT c1, c2
FROM t1
FULL OUTER JOIN t2 ON condition;
```
Perform full outer join

```
SELECT c1, c2
FROM t1
CROSS JOIN t2;
```
Produce a Cartesian product of rows in tables

```
SELECT c1, c2
FROM t1, t2;
```
Another way to perform cross join

```
SELECT c1, c2
FROM t1 A
INNER JOIN t2 B ON condition;
```
Join t1 to itself using INNER JOIN clause

USING SQL OPERATORS

```
SELECT c1, c2 FROM t1
UNION [ALL]
SELECT c1, c2 FROM t2;
```
Combine rows from two queries

```
SELECT c1, c2 FROM t1
INTERSECT
SELECT c1, c2 FROM t2;
```
Return the intersection of two queries

```
SELECT c1, c2 FROM t1
MINUS
SELECT c1, c2 FROM t2;
```
Subtract a result set from another result set

```
SELECT c1, c2 FROM t1
WHERE c1 [NOT] LIKE pattern;
```
Query rows using pattern matching %, _

```
SELECT c1, c2 FROM t
WHERE c1 [NOT] IN value_list;
```
Query rows in a list

```
SELECT c1, c2 FROM t
WHERE c1 BETWEEN low AND high;
```
Query rows between two values

```
SELECT c1, c2 FROM t
WHERE c1 IS [NOT] NULL;
```
Check if values in a table is NULL or not

Referensi

- https://www.dofactory.com/