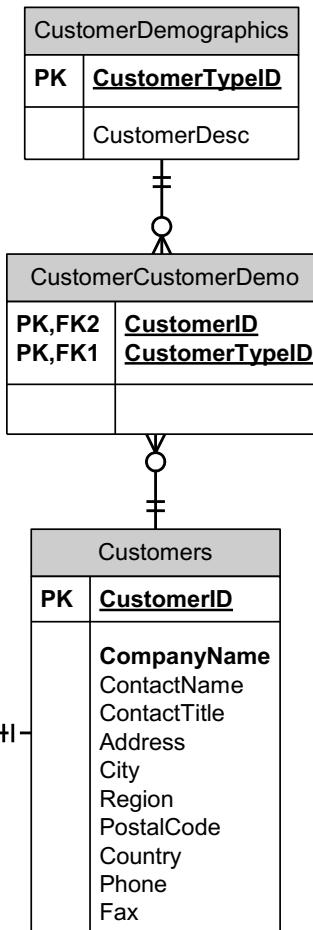
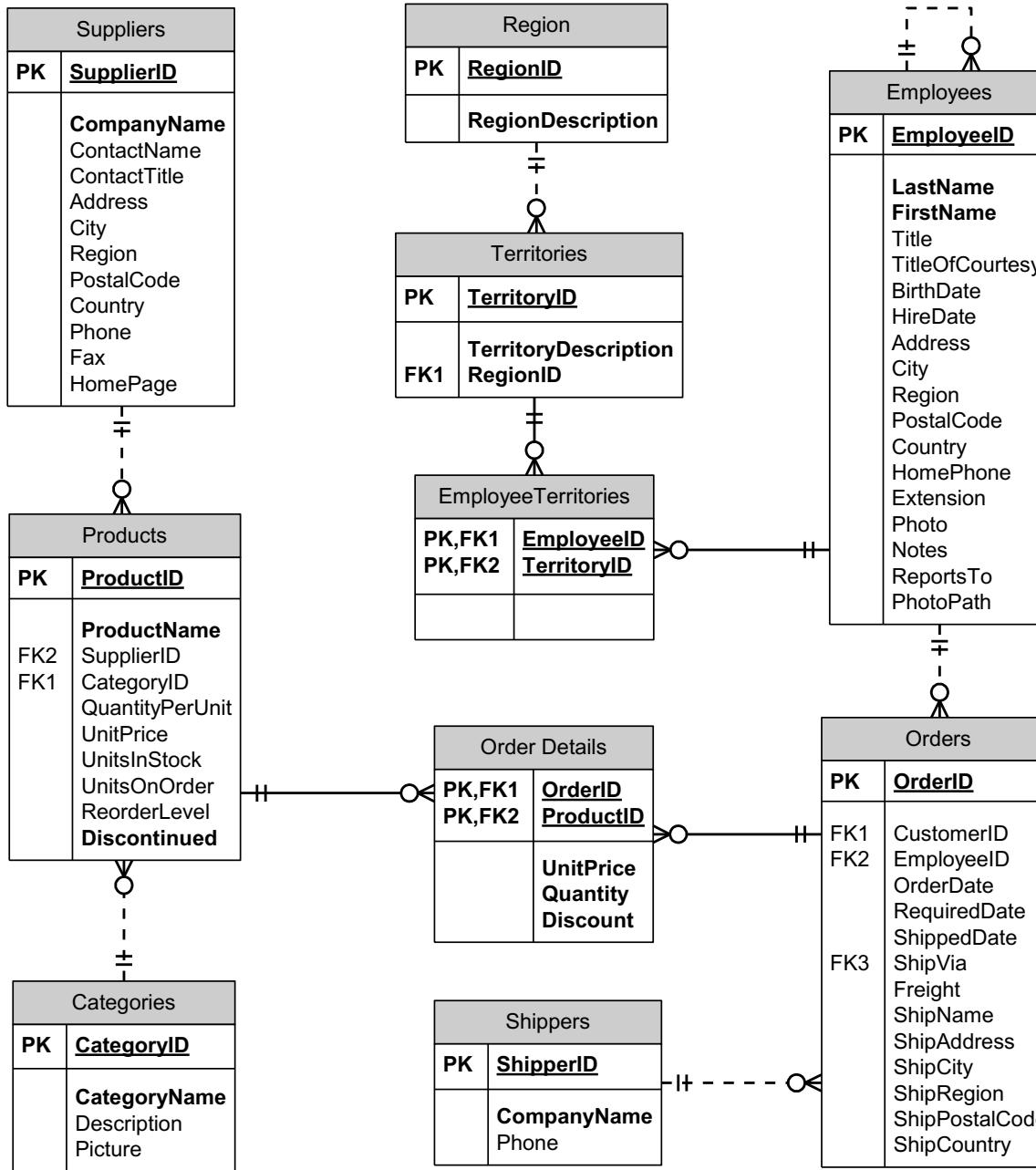


# SQL

# Northwind Database



# SELECT

- Select List Specifies the Columns
- WHERE Clause Specifies the Condition Restricting the Query
- FROM Clause Specifies the Table

```
SELECT [ALL | DISTINCT] <select_list>
      FROM {<table_source>} [,...n]
      WHERE <search_condition>
```

# SCHEMA

- **ALTER SESSION SET CURRENT\_SCHEMA = northwind;**

# Specifying Columns

```
SELECT employeeid, lastname, firstname, title  
FROM employees;
```



<i>employeeid</i>	<i>lastname</i>	<i>firstname</i>	<i>title</i>
1	Davolio	Nancy	Sales Representative
2	Fuller	Andrew	Vice President, Sales
3	Leverling	Janet	Sales Representative
4	Peacock	Margaret	Sales Representative
5	Buchanan	Steven	Sales Manager
6	Suyama	Michael	Sales Representative
7	King	Robert	Sales Representative
8	Callahan	Laura	Inside Sales Coordinator
9	Dodsworth	Anne	Sales Representative

# Exercise

1. Select customer names and addresses
2. Select the employeeid, lastname, firstname, and title columns of all employees
3. Select name and unitprice of all products
4. Select name and description of all categories
5. Select name and www address of all suppliers

# Using the WHERE Clause to Specify Rows

```
SELECT employeeid, lastname, firstname, title  
FROM employees  
WHERE employeeid = 5;
```



<i>employeeid</i>	<i>lastname</i>	<i>firstname</i>	<i>title</i>
5	Buchanan	Steven	Sales Manager

# Using Comparison Operators

```
SELECT lastname, city  
FROM employees  
WHERE country = 'USA';
```



<i>lastname</i>	<i>city</i>
Davolio	Seattle
Fuller	Tacoma
Leverling	Kirkland
Peacock	Redmond
Callahan	Seattle

# Using Comparison Operators

- This example retrieves the **orderid** and **customerid** columns with order dates that are older than 8/1/96 from the **orders** table.

```
SELECT orderid, customerid  
FROM orders  
WHERE orderdate < to_date('8/1/1996', 'MM-DD-YYYY');
```

# Exercise

1. Select name and address of customers from London
2. Select names and address of customers from Spain and France
3. Select names of products with price between 20.00 a 30.00
4. Select names of products from the ‘meat ... ’ category
5. Select names of products supplied by ‘Tokyo Traders’

# LIKE

LIKE 'BR%'

LIKE 'Br%'

LIKE '%een'

LIKE '%en%'

LIKE '\_en'

LIKE '[CK]%'

LIKE '[S-V]%'

LIKE 'M[^c]%'

# LIKE

```
SELECT companyname  
FROM customers  
WHERE companyname LIKE '%Restaurant%';
```



<i>companyname</i>
GROSELLA-Restaurante
Lonesome Pine Restaurant
Tortuga Restaurante

# LIKE

```
SELECT * FROM products  
WHERE QuantityPerUnit LIKE '%bottle%'
```

# Logical operators

```
SELECT productid, productname, supplierid, unitprice  
FROM products  
WHERE (productname LIKE 'T%' OR productid = 46)  
AND (unitprice > 16.00) ;
```



<i>productid</i>	<i>productname</i>	<i>supplierid</i>	<i>unitprice</i>
14	Tofu	6	23.25
29	Thüringer Rostbratwurst	12	123.79
62	Tarte au sucre	29	49.3

# Logical operators

- The following example retrieves all products with product names that begin with the letter T or have a product identification number of 46, and that have a price greater than \$16.00.

```
SELECT productid, productname, supplierid, unitprice  
FROM products  
WHERE (productname LIKE 'T%' OR productid = 46)  
AND (unitprice > 16.00) ;
```

# Logical operators

- The following example retrieves products with product names that begin with the letter T, or that have a product identification number of 46 and a price greater than \$16.00

```
SELECT productid, productname, supplierid, unitprice  
FROM products  
WHERE (productname LIKE 'T%')  
OR (productid = 46 AND unitprice > 16.00);
```

# Between

```
SELECT productname, unitprice  
FROM products  
WHERE unitprice BETWEEN 10 AND 20;
```



<i>productname</i>	<i>unitprice</i>
Chai	18
Chang	19
Aniseed Syrup	10
Genen Shouyu	15.5
Pavlova	17.45
Sir Rodney's Scones	10
...	...

# Exercise

```
SELECT productname, unitprice  
FROM products  
WHERE unitprice BETWEEN 10 AND 20
```

- Write with <, > ,<=,>=

# Between

```
SELECT productname, unitprice  
FROM products  
WHERE unitprice NOT BETWEEN 10 AND 20;
```

In

```
SELECT companynname, country  
FROM suppliers  
WHERE country IN ('Japan', 'Italy')
```



<i>companynname</i>	<i>country</i>
Tokyo Traders	Japan
Mayumi's	Japan
Formaggi Fortini s.r.l.	Italy
Pasta Buttini s.r.l.	Italy

OR

```
SELECT companynname, country  
FROM suppliers  
WHERE country = 'Japan' OR country = 'Italy'
```



<i>companynname</i>	<i>country</i>
Tokyo Traders	Japan
Mayumi's	Japan
Formaggi Fortini s.r.l.	Italy
Pasta Buttini s.r.l.	Italy

# NULL

```
SELECT companynname, fax  
FROM suppliers  
WHERE fax IS NULL
```



<i>companynname</i>	<i>fax</i>
Exotic Liquids	NULL
New Orleans Cajun Delights	NULL
Tokyo Traders	NULL
Cooperativa de Quesos 'Las Cabras'	NULL
...	...

# ORDER BY

```
SELECT productid, productname, unitprice  
FROM products  
ORDER BY unitprice
```

```
SELECT productid, productname, unitprice  
FROM products  
ORDER BY unitprice DESC
```

# ORDER BY

```
SELECT productid, productname, categoryid, unitprice  
FROM products  
ORDER BY categoryid, unitprice DESC
```



<i>productid</i>	<i>productname</i>	<i>categoryid</i>	<i>unitprice</i>
38	Cote de Blaye	1	263.5000
43	Ipoх Coffee	1	46.0000
2	Chang	1	19.0000
...	...	...	...
63	Vegie-spread	2	43.9000
8	Northwoods Cranberry Sauce	2	40.0000
61	Sirop d'érable	2	28.5000
...	...	...	...

# ORDER BY

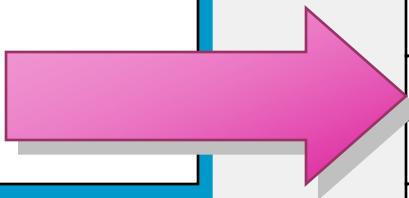
```
SELECT productid, productname, categoryid, unitprice  
FROM products  
ORDER BY 3,4 DESC
```

# DUPLICATES

```
SELECT country  
FROM suppliers  
ORDER BY country
```

# DISTINCT

```
SELECT DISTINCT country  
FROM suppliers  
ORDER BY country
```



country
Australia
Brazil
Canada
Denmark
Finland
France
Germany
Italy
Japan
Netherlands
Norway
Singapore
Spain
Sweden
UK
USA

# Changing column names

```
SELECT  firstname AS First, lastname AS Last  
       ,employeeid AS 'Employee ID:'  
FROM employees
```



<i>First</i>	<i>Last</i>	<i>Employee ID:</i>
Nancy	Davolio	1
Andrew	Fuller	2
Janet	Leverling	3
Margaret	Peacock	4
Steven	Buchanan	5
Michael	Suyama	6
Robert	King	7
Laura	Callahan	8
Anne	Dodsworth	9

# Literals

```
SELECT  firstname, lastname  
      , 'Identification number:', employeeid  
FROM employees
```



<i>First</i>	<i>Last</i>	<i>Employee ID:</i>
Nancy	Davolio	Identification Number: 1
Andrew	Fuller	Identification Number: 2
Janet	Leverling	Identification Number: 3
Margaret	Peacock	Identification Number: 4
Steven	Buchanan	Identification Number: 5
Michael	Suyama	Identification Number: 6
Robert	King	Identification Number: 7
Laura	Callahan	Identification Number: 8
Anne	Dodsworth	Identification Number: 9

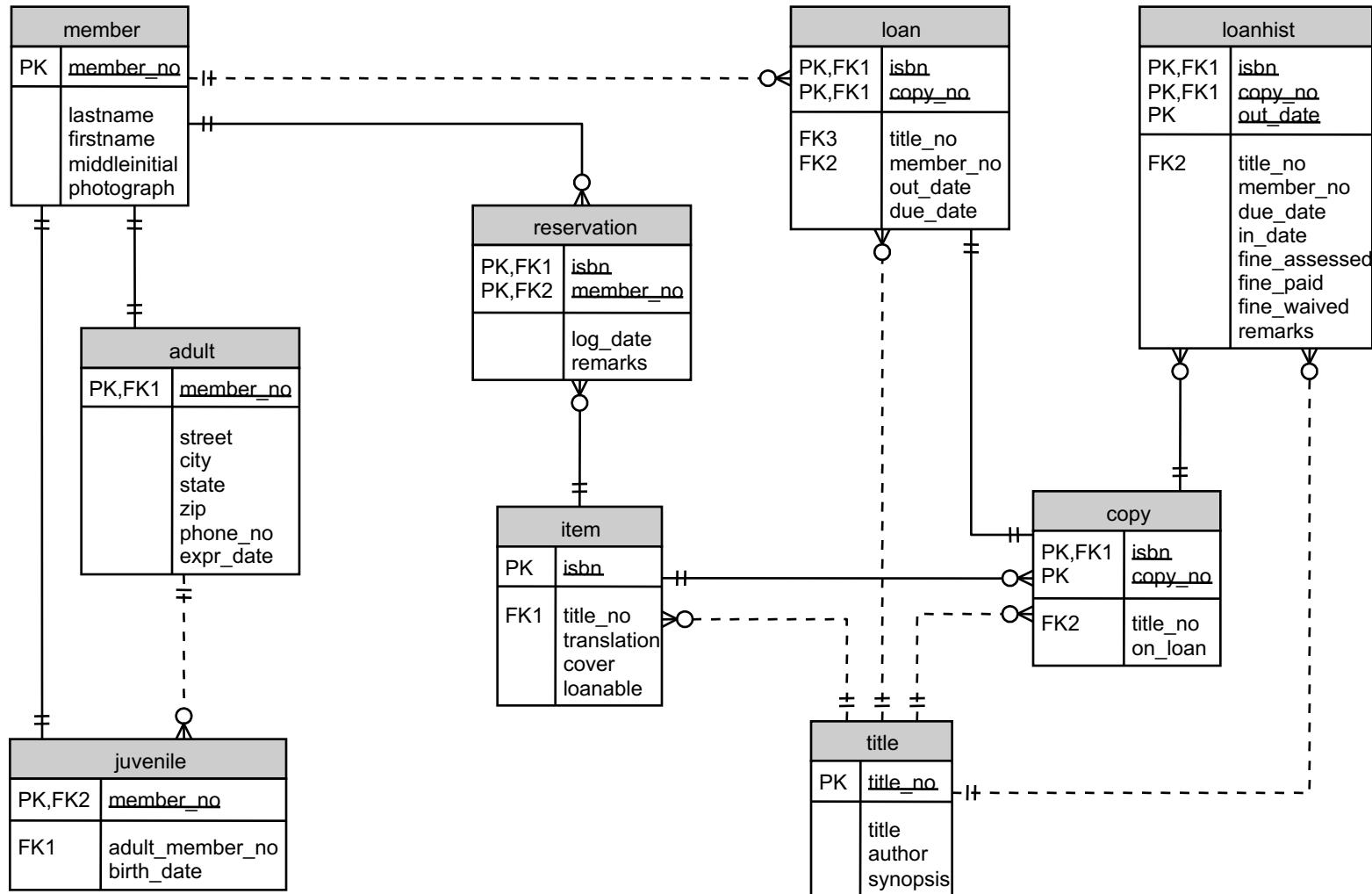
# Calculations

```
SELECT orderid, unitprice * 1.05 as newunitprice  
FROM orderdetails
```

# String concatenation

```
SELECT firstname || ‘ ’ || lastname as name  
FROM employees
```

# Library Database



# Exercise

- Write and execute a SELECT statement that retrieves the **title and title\_no** columns from the **title table**
- Write and execute a SELECT statement that retrieves the title of title number 10 from the **title table**
- Write and execute a SELECT statement that retrieves the member numbers and assessed fines from the **loanhist table for all members who have had** fines between \$8.00 and \$9.00.

# Exercise

- Write and execute a SELECT statement that retrieves the title number and author from the **title table for all books authored by Charles Dickens or Jane Austen**. Use the IN operator as part of the SELECT statement
- Write and execute a SELECT statement that retrieves the title number and title from the **title table for all rows that contain the character string “adventures” in the title**. Use the LIKE operator in your query

# Exercise

- Write and execute a SELECT statement that retrieves the member number, assessed fine, and fine that is paid for loans that have unpaid fines from the **loanhist** table.  
**Retrieve rows that have fines entered in the fine\_assessed column and that have null values for the fine\_paid column**
- Write and execute a query that retrieves all of the **unique pairs of cities and states** from the **adult** table. **You should receive only one row in the result set** for each city and state pair

# Exercise

- Write and execute a query that retrieves a sorted list of all titles from the **title** table

# Exercise

- Write and execute a query that retrieves the **member\_no**, **isbn**, and **fine\_assessed** columns from the **loanhist** table of all archived loans a nonnull value in the **fine\_assessed** column.
- Create a computed column that contains the value of the **fine\_assessed** column multiplied by two.
- Use the column alias ‘double fine’ for the computed column.

# Exercise

- Write and execute a query that generates a single column that contains the **firstname**, **middleinitial**, and **lastname** columns from the **member** table for all members with the last name Anderson.
- Use the column alias `email_name`.
- Modify the query to return a list of e-mail names with the member's first name, middle initial, and first two letters of the last name in lowercase characters. Use the `SUBSTRING` function to retrieve part of a string column. Use the `LOWER` function to return the result in lowercase characters. Use the addition (+) operator to concatenate the character strings.

# Exercise

- Write and execute a query that retrieves the **title** and **title\_no** columns from the **title** table. Your result set should be a single column with the following format:
  - The title is: Poems, title number 7
- This query returns a single column based on an expression that concatenates four elements:
  - The title is: string constant
  - title.title column
  - title number string constant
  - title.title\_no column