

①

- Selection Demo --
- General Format for naming a Database object --
- Server.database.schema.Object --
- To refer to Employees Table in Northwind DataBase
- Select * from Employees
- Select * from Northwind.dbo.Employees.
- Dont specify server name for local system...
- Use it in a Network

- Use Northwind;
- Go
- Difference between Go and ;
- ; is used for terminating a SQL Statement } Separator

-- GO is used to Terminate or Close a SQL Batch
(All declared local variables will retain their values within a batch) (2)

```
SELECT FirstName, LastName from dbo.Employees;  
Select * from dbo.employees;
```

-- Concatenating

```
SELECT LastName + ', ' + FirstName FROM dbo.Employees;
```

-- Aliasing column names

```
SELECT LastName + ', ' + FirstName AS [Full Name]  
FROM dbo.Employees;
```

-- AS is optional

```
SELECT LastName + ', ' + FirstName Full Name  
FROM dbo.Employees;
```

-- Another aliasing option

```
SELECT FullName = LastName + ', ' + FirstName  
FROM dbo.Employees;
```

-- This is deprecated;

```
SELECT 'Fullname' = LastName + ', ' + FirstName  
FROM dbo.Employees;
```

-- SELECT and SELECT DISTINCT

```
SELECT Title FROM dbo.Employees;
```

```
SELECT DISTINCT Title FROM dbo.Employees;
```

(3)

-- WHERE

```
SELECT CompanyName, City  
FROM dbo.Customers  
WHERE City = 'Paris';
```

-- LIKE and wildcard characters

```
SELECT CompanyName  
FROM dbo.Customers  
WHERE CompanyName LIKE 'S%';
```

or
'S%' (or) '%S%'

-- Matching single characters.

```
SELECT CustomerID  
FROM dbo.Customers  
WHERE CustomerID LIKE 'B---P';
```

-- Matching from a list

```
SELECT CustomerID  
FROM dbo.Customers  
WHERE CustomerID like 'FRAN[RK]';
```

LIKE 'FRAN [A-S]';

-- Not containing → LIKE 'FRAN[^R]';

-- BETWEEN

(sortable - Ordinable data)

```
SELECT LastName, FirstName, PostalCode  
From dbo.Employees  
WHERE PostalCode BETWEEN '98103' AND '98999';
```

-- Testing for Null

④

```
SELECT LastName, FirstName, Region  
FROM dbo.Employees  
WHERE Region = NULL;
```

-- Three level logic (True, False, NULL)

```
SELECT LastName, FirstName, Region  
FROM dbo.Employees  
WHERE Region IS NULL;
```

-- AND requires both conditions to be true

```
SELECT LastName, City, PostalCode  
FROM dbo.Employees  
WHERE City = 'Seattle' AND PostalCode LIKE '9%';
```

-- OR Only requires one condition to be true

```
SELECT LastName, City, PostalCode  
FROM dbo.Employees  
WHERE City = 'Seattle' OR PostalCode LIKE '9%';
```

-- NOT negates the expression

```
SELECT LastName, City, PostalCode  
FROM dbo.Employees  
WHERE City NOT LIKE 'Seattle';
```

-- Operator Precedence : NOT, AND, OR

(5)

```
SELECT LastName, FirstName, City  
FROM dbo.Employees  
WHERE LastName LIKE '%.S.%'  
      AND City NOT LIKE 'Seattle';
```

```
SELECT LastName, FirstName, City  
FROM dbo.Employees  
WHERE (LastName LIKE '%.S.%')  
      AND (City NOT LIKE 'Seattle');
```

-- IN Clause

-- One way to do it :

```
SELECT CustomerID, Country  
FROM dbo.Customers  
WHERE Country = 'France' OR Country = 'Spain'
```

```
SELECT CustomerID, Country  
FROM dbo.Customers  
WHERE Country like 'U%'
```

-- Using IN -- to match in a list of elements

```
SELECT CustomerID, Country  
FROM dbo.Customers  
WHERE Country IN ('France', 'Spain');
```

-- NESTED SUBQUERY

(6)

```
SELECT CustomerID, Country
FROM dbo.Customers
WHERE Country IN (SELECT DISTINCT Country
FROM dbo.Customers
WHERE Country like 'UX%');
```

-- IN with a Subquery (that returns a list of elements)

```
SELECT CustomerID
FROM dbo.Customers
WHERE CustomerID NOT IN (SELECT DISTINCT CustomerID
                           FROM dbo.Orders);
```

Join

→ Create table Customers (cid int primary key, name varchar(50),
cphone varchar(50))

→ Create table Orders (oid int primary key, cid int,
product varchar(50), quantity int)

→ insert into Customers values (1, 'anand', '12345')

:
: (1, 'Vimal', '32125')
(2, 'Niru', '54321')

→ Select * from Customers

→ insert into orders values (1, 1, 'pencil', 100)

⑦

(10,)

→ select * from orders

-- cross join

select * from customers c cross join orders o

-- cross join differently

select * from customers, orders

-- cross join with condition (a inner join)

select * from customers c cross join orders o where
c.cid = o.cid

-- a simple inner join

select * from customers join orders on customers.cid =
orders.cid

-- another way for inner join without join keyword

select * from customers, orders where customers.cid =
orders.cid

-- another way for inner join without join keyword

select * from customers, orders where customers.cid =
orders.cid

-- another way for inner join

select * from customers c inner join orders o on
c.cid = o.cid

- Show Only Selective columns using a join (8)

Select. c.cid as CustId, c.cname as Name, c.cphone as phone, o.oid as OrderId, o.product as product, o.quantity as Quantity from Customers c full join orders o on c.cid = o.cid where c.cid is null

Right join, Left join

where o.oid is null.

SQL - 3 class:

- Sorting Demo
- ORDER By

SELECT LastName, City
FROM dbo.Employees
ORDER By City \rightarrow asc (or) DESC.

- Sorting on multiple columns (ASC optional)

SELECT LastName, City
FROM dbo.Employees
ORDER By City DESC, LastName ASC;

Region \rightarrow Specific
columns
- I can't count with