

# COMP 7810/4096

## Business Intelligence

FOK WAI YEE  
INSTRUCTOR, CS, HKBU

# SSMS vs SSIS

- ▶ SQL Server Management Studio (SSMS)
  - ▶ creating databases, design tables, adding data, updating data and running queries.
- ▶ SQL Server Integration Services (SSIS)
  - ▶ extraction, transformation, and loading (ETL) of data for data warehousing

# SQL

- ▶ Structured Query Language (**SQL**) is a language for querying databases

## SQL SELECT Syntax

```
SELECT column_name, column_name  
FROM table_name;
```

and

```
SELECT * FROM table_name;
```

# Select clause

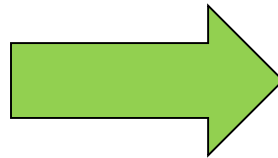
```
SELECT ProductID, ProductName, Price, ProductDescription  
FROM Products
```

```
SELECT *  
FROM Products
```

```
SELECT ProductID, ProductName, Price, ProductDescription  
FROM Products  
WHERE ProductID < 60
```

# Where Clause

SalesOrderID	Line Total
1	100
2	200
1	300
3	550
2	500

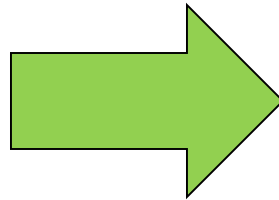


SalesOrderID	Line Total
1	400
1	300

```
SELECT SalesOrderID, [Line Total]
FROM SalesOrderDetails
Where SalesOrderID = 1
```

# Where Clause

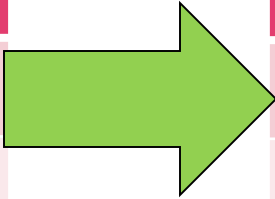
Salesperson	Line Total
Chan	100
Wong	200
Chan	300
Lee	550
Wong	500



Salesperson	Line Total
Chan	400
Chan	300

```
SELECT Salesperson, [Line Total]
FROM SalesOrderDetails
WHERE Salesperson='Chan';
```

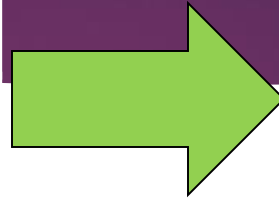
# Group by Clause

SalesOrderID	Line Total		SalesOrderID	SubTotal
1	100		1	400
2	200		2	700
1	300		3	550
3	550			
2	500			

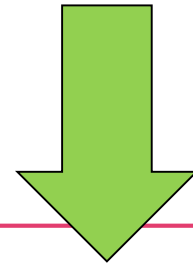
```
SELECT SalesOrderID, SUM([Line Total]) AS SubTotal
FROM SalesOrderDetails
GROUP BY SalesOrderID
```

# Having, Order by

SalesOrderID	Line Total
1	100
2	200
1	300
3	550
2	500



SalesOrderID	SubTotal
1	400
2	700
3	550



```
SELECT SalesOrderID, SUM([Line Total]) AS SubTotal
FROM SalesOrderDetails
GROUP BY SalesOrderID
HAVING SUM([Line Total]) > 500
ORDER BY SubTotal DESC;
```

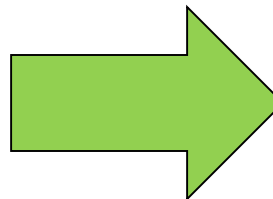
SalesOrderID	SubTotal
2	700
3	550



# Top n Values

```
SELECT Top 1 SalesOrderID, SUM([Line Total]) AS SubTotal  
FROM SalesOrderDetails  
GROUP BY SalesOrderID  
ORDER BY SubTotal DESC;
```

SalesOrderID	Line Total
1	100
2	200
1	300
3	550
2	500



SalesOrderID	SubTotal
2	700

# Cannot retrieve individual item after grouping

- ▶ Error occurs if you type:

```
SELECT SalesOrderID, OrderDate  
FROM SalesOrderDetails  
GROUP BY SalesOrderID
```



SalesOrderID	Line Total	OrderDate
1	100	3/18/2015
2	200	3/19/2015
1	300	3/20/2015
3	550	3/21/2015
2	500	3/22/2015



No individual item → OK

```
select salesperson, MAX([order amount]) as max_amount  
from tbl_Sales group by salesperson
```

Results Messages

	salesperson	max_amount
1	Buchanan	9210.9
2	Callahan	4825
3	Davolio	15810
4	Dodsworth	11380
5	Fuller	16387.5
6	King	12615.05
7	Leverling	10952.84
8	Peacock	11188.4
9	Suyama	4707.54



# Subquery

- ▶ A **subquery** is a SQL query within a query

```
SELECT * FROM SalesOrderDetails
where [Line Total] =
(select MAX([Line Total]) from SalesOrderDetails)
```

SalesOrderID	Line Total	OrderDate
1	100	3/18/2015
2	200	3/19/2015
1	300	3/20/2015
3	550	3/21/2015
2	500	3/22/2015



SalesOrderID	Line Total	OrderDate
3	550	3/21/2015

## Try

- ▶ How to select the `country` field and `salesperson` field from `tbl_sales` for the salesperson having the `maximum` order amount?
- ▶ Hint: use `max()` to calculate maximum order amount in subquery

# Other functions

- ▶ AVG()
- ▶ MAX()
- ▶ MIN()
- ▶ COUNT()

# More references

- ▶ <https://www.w3schools.com/sql/default.asp>
- ▶ [http://docs.oracle.com/html/A95915\\_01/sqfunc.htm](http://docs.oracle.com/html/A95915_01/sqfunc.htm)