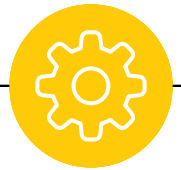
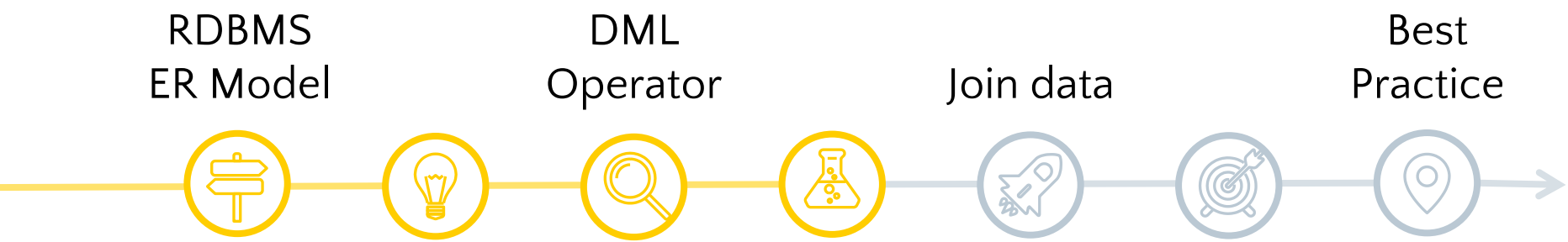
Welcome back



SQL Essentials

Roadmap


RDBMS
ER Model
SQL Server
DDL
DML
Operator
Function
SQL Clause
Join data
Sub Query
Best
Practice

Previous lecture

DML

- INSERT
- UPDATE
- DELETE

Select

- SELECT Syntax
- TOP & PERCENT
- ALIAS
- DISTINCT
- FROM
- WHERE
- VIEW
- SELECT INTO

Operator

- Authentic
- Compare
- Logical

What we will explore today?

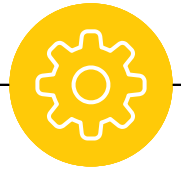
SQL built-in Function

- String funtions
- Datetime functions
- Aggregate functions
- Others

SQL Clause

- GROUP BY
- HAVING

SQL Built-in Functions



String Functions

Prepare data

```
CREATE TABLE Student (
  ID INT PRIMARY KEY AUTO_INCREMENT,
  FirstName VARCHAR(10),
  MiddleName VARCHAR(10),
  LastName VARCHAR(10),
  Math INT,
  Physic INT,
  Chemical INT,
  DateOfBirth DATE
);
```

```
INSERT INTO Student(FirstName, MiddleName, LastName, Math, Physic,
Chemical, DateOfBirth) VALUES
                ('Nguyễn', 'Văn', 'Huấn', 7, 8, 9, '2000-10-15'),
                ('Võ', 'Văn', 'Hiếu', 3, 4, 5, '2005-10-15'),
                ('Nguyễn', 'Thị', 'Huệ', 2, 5, 7, '2008-10-15'),
                ('Nguyễn', NULL, 'Trương', NULL, 5, 7, '1999-10-15');
```

Prepare data

	ID	FirstName	MiddleName	LastName	Math	Physic	Chemical	DateOfBirth
▶	1	Nguyễn	Văn	Huấn	7	8	9	2000-10-15
	2	Võ	Văn	Hiếu	3	4	5	2005-10-15
	3	Nguyễn	Thị	Huệ	2	5	7	2008-10-15
	4	Nguyễn	NULL	Trương	NULL	5	7	1999-10-15
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

LOWER & UPPER

```
SELECT LOWER('HELLO') AS 'lower function';
```

	lower function
▶	hello

```
SELECT UPPER('Hi there') AS 'UPPER FUNCTION';
```

	UPPER FUNCTION
▶	HI THERE

Practice

Base Data

ID	FirstName	MiddleName	LastName	Math	Physic	Chemical	DateOfBirth
1	Nguyễn	Văn	Huấn	7	8	9	2000-10-15
2	Võ	Văn	Hiếu	3	4	5	2005-10-15
3	Nguyễn	Thị	Huệ	2	5	7	2008-10-15
4	Nguyễn	NULL	Trương	NULL	5	7	1999-10-15
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
SELECT UPPER(FirstName) AS 'UPPER FirstName',
       LOWER(LastName) AS 'LOWER LastName'
FROM Student;
```

UPPER FirstName	LOWER LastName
NGUYỄN	huấn
VÕ	hiếu
NGUYỄN	huệ
NGUYỄN	trương

CHAR_LENGTH & REVERSE

```
SELECT CHAR_LENGTH('Test Length') AS 'LEN FUNCTION';
```

	LEN FUNCTION
▶	11

```
SELECT REVERSE('123456') AS 'REVERSE FUNCTION';
```

	REVERSE FUNCTION
▶	654321

Try it

Base Data

ID	FirstName	MiddleName	LastName	Math	Physic	Chemical	DateOfBirth
1	Nguyễn	Văn	Huấn	7	8	9	2000-10-15
2	Võ	Văn	Hiếu	3	4	5	2005-10-15
3	Nguyễn	Thị	Huệ	2	5	7	2008-10-15
4	Nguyễn	NULL	Trương	NULL	5	7	1999-10-15
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
SELECT REVERSE(LastName) AS 'REVERSE LastName',
 CHAR_LENGTH(LastName) AS 'Length of LastName'
FROM Student
```

REVERSE LastName	Length of LastName
nấuH	4
uếiH	4
ệuH	3
gnơưrT	6

CONCAT & SUBSTRING

```
SELECT CONCAT('He','llo') AS 'CONCAT FUNCTION';
```

	CONCAT FUNCTION
▶	Hello

```
SELECT SUBSTRING('1234567', 2, 3) AS 'SUBSTRING FUNCTION';
```

	SUBSTRING FUNCTION
▶	234

Practice

Base Data

	ID	FirstName	MiddleName	LastName	Math	Physic	Chemical	DateOfBirth
▶	1	Nguyễn	Văn	Huấn	7	8	9	2000-10-15
	2	Võ	Văn	Hiếu	3	4	5	2005-10-15
	3	Nguyễn	Thị	Huệ	2	5	7	2008-10-15
	4	Nguyễn	NULL	Trương	NULL	5	7	1999-10-15
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
SELECT CONCAT(FirstName, ' ', MiddleName, ' ', LastName) AS 'FullName'
FROM Student
```

	FullName
▶	Nguyễn Văn Huấn
	Võ Văn Hiếu
	Nguyễn Thị Huệ
	NULL

TRIM

```
SELECT TRIM('  HI  ') AS 'TRIM FUNCTION';
```

	TRIM FUNCTION
▶	HI

Datetime Functions

MONTH, DAY, YEAR

```
SELECT MONTH('2022-11-03') AS 'MONTH FUNCTION';
```

	MONTH FUNCTION
▶	11

```
SELECT DAY('2022-11-03') AS 'DAY FUNCTION';
```

	DAY FUNCTION
▶	3

```
SELECT YEAR('2022-11-03') AS 'YEAR FUNCTION';
```

	YEAR FUNCTION
▶	2022

Practice

Base Data

ID	FirstName	MiddleName	LastName	Math	Physic	Chemical	DateOfBirth
1	Nguyễn	Văn	Huấn	7	8	9	2000-10-15
2	Võ	Văn	Hiếu	3	4	5	2005-10-15
3	Nguyễn	Thị	Huệ	2	5	7	2008-10-15
4	Nguyễn	NULL	Trương	NULL	5	7	1999-10-15
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

LastName	DateOfBirth	Birth Year
Huấn	2000-10-15	2000
Hiếu	2005-10-15	2005
Huệ	2008-10-15	2008
Trương	1999-10-15	1999

```
SELECT LastName, DateOfBirth, YEAR(DateOfBirth) AS 'Birth Year'
FROM Student
```

NOW, DATE_ADD, DATE_SUB

```
SELECT NOW() AS current_date;
```

	current_datetime
▶	2023-07-12 07:19:24

```
SELECT DATE_ADD(NOW(), INTERVAL 1 WEEK) AS future_date;
SELECT DATE_SUB(NOW(), INTERVAL 1 HOUR) AS past_date;
```

	future_date
▶	2023-07-19 07:28:47

	past_date
▶	2023-07-12 06:29:10

STR_TO_DATE

```
SELECT STR_TO_DATE('15/10/2020', '%d/%m/%Y') Convert103;
SELECT STR_TO_DATE('15.10.2020', '%d.%m.%Y') Convert104;
SELECT STR_TO_DATE('15-10-2020', '%d-%m-%Y') Convert105;
```

	Convert103
▶	2020-10-15

	Convert104
▶	2020-10-15

	Convert105
▶	2020-10-15

style	input/output	default
100	mon dd yyyy hh:miAM/PM	Default
101	mm/dd/yyyy	US
102	yyyy.mm.dd	ANSI
103	dd/mm/yyyy	British/French
104	dd.mm.yyyy	German
105	dd-mm-yyyy	Italian
106	dd mon yyyy	-
107	Mon dd, yyyy	-
108	hh:mm:ss	-

Aggregrate Functions

SUM

Base Data

	ID	FirstName	MiddleName	LastName	Math	Physic	Chemical	DateOfBirth
▶	1	Nguyễn	Văn	Huấn	7	8	9	2000-10-15
	2	Võ	Văn	Hiếu	3	4	5	2005-10-15
	3	Nguyễn	Thị	Huệ	2	5	7	2008-10-15
	4	Nguyễn	NULL	Trương	NULL	5	7	1999-10-15
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
SELECT SUM(Physic) AS 'Sum of Physic'
FROM Student
```

	Sum of Physic
▶	22

Practice

Base Data

	ID	FirstName	MiddleName	LastName	Math	Physic	Chemical	DateOfBirth
▶	1	Nguyễn	Văn	Huấn	7	8	9	2000-10-15
	2	Võ	Văn	Hiếu	3	4	5	2005-10-15
	3	Nguyễn	Thị	Huệ	2	5	7	2008-10-15
	4	Nguyễn	NULL	Trương	NULL	5	7	1999-10-15
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
SELECT SUM(Math) AS 'Sum of Math'
FROM Student
```

	Sum of Math
▶	12

AVG

Base Data

ID	FirstName	MiddleName	LastName	Math	Physic	Chemical	DateOfBirth
1	Nguyễn	Văn	Huấn	7	8	9	2000-10-15
2	Võ	Văn	Hiếu	3	4	5	2005-10-15
3	Nguyễn	Thị	Huệ	2	5	7	2008-10-15
4	Nguyễn	NULL	Trương	NULL	5	7	1999-10-15
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
SELECT AVG(Physic) AS 'AVG of Physic'
FROM Student
```

AVG of Physic
5.5000

Practice

Base Data

ID	FirstName	MiddleName	LastName	Math	Physic	Chemical	DateOfBirth
1	Nguyễn	Văn	Huấn	7	8	9	2000-10-15
2	Võ	Văn	Hiếu	3	4	5	2005-10-15
3	Nguyễn	Thị	Huệ	2	5	7	2008-10-15
4	Nguyễn	NULL	Trương	NULL	5	7	1999-10-15
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
SELECT AVG(Math) AS 'AVG of Math'
FROM Student
```

AVG of Math
4.0000

MIN & MAX

Base Data

ID	FirstName	MiddleName	LastName	Math	Physic	Chemical	DateOfBirth
1	Nguyễn	Văn	Huấn	7	8	9	2000-10-15
2	Võ	Văn	Hiếu	3	4	5	2005-10-15
3	Nguyễn	Thị	Huệ	2	5	7	2008-10-15
4	Nguyễn	NULL	Trương	NULL	5	7	1999-10-15
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
SELECT MIN(Math) AS 'MIN of Math',
       MAX(Math) AS 'MAX of Math'
FROM Student
```

MIN of Math	MAX of Math
2	7

Practice

Base Data

	ID	FirstName	MiddleName	LastName	Math	Physic	Chemical	DateOfBirth
▶	1	Nguyễn	Văn	Huấn	7	8	9	2000-10-15
	2	Võ	Văn	Hiếu	3	4	5	2005-10-15
	3	Nguyễn	Thị	Huệ	2	5	7	2008-10-15
	4	Nguyễn	NULL	Trương	NULL	5	7	1999-10-15
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
SELECT MIN(Physic) AS 'MIN of Physic',
       MAX(Physic) AS 'MAX of Physic'
FROM Student
```

	MIN of Physic	MAX of Physic
▶	4	8

COUNT

Base Data

ID	FirstName	MiddleName	LastName	Math	Physic	Chemical	DateOfBirth
1	Nguyễn	Văn	Huấn	7	8	9	2000-10-15
2	Võ	Văn	Hiếu	3	4	5	2005-10-15
3	Nguyễn	Thị	Huệ	2	5	7	2008-10-15
4	Nguyễn	NULL	Trương	NULL	5	7	1999-10-15
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
SELECT COUNT(ID) AS 'COUNT ID'
FROM Student
```

COUNT ID
4

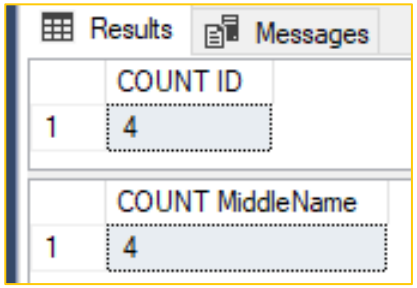
```
SELECT COUNT(MiddleName) AS 'COUNT MiddleName'
FROM Student
```

COUNT MiddleName
3

COUNT(*) vs COUNT(1)

```
SELECT COUNT(*) AS 'COUNT ID'
FROM Student
```

```
SELECT COUNT(1) AS 'COUNT MiddleName'
FROM Student
```



COUNT with DISTINCT

Base Data

ID	FirstName	MiddleName	LastName	Math	Physic	Chemical	DateOfBirth
1	Nguyễn	Văn	Huấn	7	8	9	2000-10-15
2	Võ	Văn	Hiếu	3	4	5	2005-10-15
3	Nguyễn	Thị	Huệ	2	5	7	2008-10-15
4	Nguyễn	NULL	Trương	NULL	5	7	1999-10-15
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
SELECT COUNT(DISTINCT MiddleName) AS 'COUNT DISTINCT'
FROM Student
```

COUNT DISTINCT
2

CEILING & FLOOR

```
SELECT CEILING(1.00001) AS 'LÀM TRÒN LÊN'
```

	LÀM TRÒN LÊN
▶	2

```
SELECT FLOOR(1.99999) AS 'LÀM TRÒN XUỐNG'
```

	LÀM TRÒN XUỐNG
▶	1

PI & ROUND

```
SELECT PI() AS 'PI NUMBER'
```

	PI NUMBER
▶	3.141593

```
SELECT ROUND(PI(), 4) AS 'ROUND PI NUMBER'
SELECT ROUND(PI(), 2) AS 'ROUND PI NUMBER'
```

	ROUND PI NUMBER
▶	3.1416

	ROUND PI NUMBER
▶	3.14

POWER & SQRT

```
SELECT POWER(2, 8) AS 'POWER LÀ SỨC MẠNH'
```

	POWER LÀ SỨC MẠNH
▶	256

```
SELECT SQRT(16) AS 'SQRT LÀ CĂN BẬC 2'
```

	SQRT LÀ CĂN BẬC 2
▶	4

Others Function

Just like if-else statement

```
SELECT IF(1>0, 'Điều kiện đúng',
'Điều kiện sai');
```

	IF(1>0, 'Điều kiện đúng', 'Điều kiện sai')
▶	Điều kiện đúng

```
SELECT IF(1=0, 'Điều kiện đúng',
'Điều kiện sai');
```

	IF(1=0, 'Điều kiện đúng', 'Điều kiện sai')
▶	Điều kiện sai

Math > 5

Base Data

	ID	FirstName	MiddleName	LastName	Math	Physic	Chemical	DateOfBirth
▶	1	Nguyễn	Văn	Huấn	7	8	9	2000-10-15
	2	Võ	Văn	Hiếu	3	4	5	2005-10-15
	3	Nguyễn	Thị	Huệ	2	5	7	2008-10-15
	4	Nguyễn	NULL	Trương	NULL	5	7	1999-10-15
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
SELECT LastName, IF(Math>5,'Tạm ổn','Chưa ổn lắm') AS 'Học Lực'
FROM Student
```

	LastName	Học Lực
▶	Huấn	Tạm ổn
	Hiếu	Chưa ổn lắm
	Huệ	Chưa ổn lắm
	Trương	Chưa ổn lắm

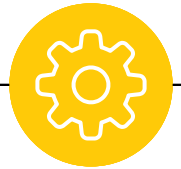
CAST

```
SELECT CAST(25.65 AS char) CastToChar;
SELECT CAST('2022-11-25' AS datetime) CastToDatetime;
```

	CastToChar
▶	25.65

	CastToDatetime
▶	2022-11-25 00:00:00

SQL CLAUSE



Prepare data

```
CREATE TABLE StudentScore(
        ID int PRIMARY KEY AUTO_INCREMENT,
        StudentID int,
        FullName VARCHAR(20),
        LearnSubject VARCHAR(20),
        Score int
);
```

```
INSERT INTO StudentScore(StudentID, FullName,
LearnSubject, Score)
VALUES (1,'Châu Tinh Trì', 'Toán', 3),
      (2,'Châu Kiệt Luân', 'Toán', 3),
      (3, 'Châu Nhuận Phát', 'Toán', 5),
      (4,'Ngôn Nhật Phi', 'Toán', 5),
      (1,'Châu Tinh Trì', 'Văn', 6),
      (2,'Châu Kiệt Luân', 'Văn', 6),
      (3, 'Châu Nhuận Phát', 'Văn', 8),
      (4,'Ngôn Nhật Phi', 'Văn', 8),
      (1,'Châu Tinh Trì', 'Anh', 8),
      (2,'Châu Kiệt Luân', 'Anh', 10),
      (3, 'Châu Nhuận Phát', 'Anh', 10),
      (4,'Ngôn Nhật Phi', 'Anh', 8);
```

Some data overview

ID	StudentID	FullName	LearnSubject	Score
2	2	Châu Kiệt Luân	Toán	3
6	2	Châu Kiệt Luân	Văn	6
10	2	Châu Kiệt Luân	Anh	10
3	3	Châu Nhuận Phát	Toán	5
7	3	Châu Nhuận Phát	Văn	8
11	3	Châu Nhuận Phát	Anh	10
1	1	Châu Tinh Trì	Toán	3
5	1	Châu Tinh Trì	Văn	6
9	1	Châu Tinh Trì	Anh	8
4	4	Ngôn Nhật Phi	Toán	5
8	4	Ngôn Nhật Phi	Văn	8
12	4	Ngôn Nhật Phi	Anh	8
NULL	NULL	NULL	NULL	NULL

```
SELECT SUM(Score) SUMScore
FROM StudentScore
```

SUMScore
80

```
SELECT AVG(Score) AVGScore
FROM StudentScore
```

AVGScore
6.6667

GROUP BY

- The GROUP BY statement groups rows that have the same values into summary rows
- The GROUP BY statement is often used with aggregate functions (COUNT(), MAX(), MIN(), SUM(), AVG()) to group the result-set by one or more columns.

Big picture

```
SELECT column_data
FROM source
    JOIN source2
WHERE condition
GROUP BY
HAVING condition
ORDER BY sort [ASC|DESC]
```

GROUP BY

```
SELECT AVG(Score) AVGScore
FROM StudentScore;
```

AVGScore
6.6667

ID	StudentID	FullName	LearnSubject	Score
2	2	Châu Kiệt Luân	Toán	3
6	2	Châu Kiệt Luân	Văn	6
10	2	Châu Kiệt Luân	Anh	10
3	3	Châu Nhuận Phát	Toán	5
7	3	Châu Nhuận Phát	Văn	8
11	3	Châu Nhuận Phát	Anh	10
1	1	Châu Tinh Trì	Toán	3
5	1	Châu Tinh Trì	Văn	6
9	1	Châu Tinh Trì	Anh	8
4	4	Ngôn Nhật Phi	Toán	5
8	4	Ngôn Nhật Phi	Văn	8
12	4	Ngôn Nhật Phi	Anh	8
NULL	NULL	NULL	NULL	NULL

```
SELECT LearnSubject, AVG(Score) AVGScore
FROM StudentScore
GROUP BY LearnSubject;
```

LearnSubject	AVGScore
Toán	4.0000
Văn	7.0000
Anh	9.0000

Practice 1

```
SELECT StudentID, FullName, AVG(Score)
AVGScore
FROM StudentScore
GROUP BY StudentID, FullName
ORDER BY StudentID
```

StudentID	FullName	AVGScore
1	Châu Tinh Trì	5.6667
2	Châu Kiệt Luân	6.3333
3	Châu Nhuận Phát	7.6667
4	Ngôn Nhật Phi	7.0000

ID	StudentID	FullName	LearnSubject	Score
2	2	Châu Kiệt Luân	Toán	3
6	2	Châu Kiệt Luân	Văn	6
10	2	Châu Kiệt Luân	Anh	10
3	3	Châu Nhuận Phát	Toán	5
7	3	Châu Nhuận Phát	Văn	8
11	3	Châu Nhuận Phát	Anh	10
1	1	Châu Tinh Trì	Toán	3
5	1	Châu Tinh Trì	Văn	6
9	1	Châu Tinh Trì	Anh	8
4	4	Ngôn Nhật Phi	Toán	5
8	4	Ngôn Nhật Phi	Văn	8
12	4	Ngôn Nhật Phi	Anh	8
NULL	NULL	NULL	NULL	NULL

Practice 2

```
SELECT StudentID, FullName, AVG(Score)
AVGScore
FROM StudentScore
WHERE StudentID = 1
GROUP BY StudentID, FullName
ORDER BY StudentID
```

StudentID	FullName	AVGScore
1	Châu Tinh Trì	5.6667

ID	StudentID	FullName	LearnSubject	Score
2	2	Châu Kiệt Luân	Toán	3
6	2	Châu Kiệt Luân	Văn	6
10	2	Châu Kiệt Luân	Anh	10
3	3	Châu Nhuận Phát	Toán	5
7	3	Châu Nhuận Phát	Văn	8
11	3	Châu Nhuận Phát	Anh	10
1	1	Châu Tinh Trì	Toán	3
5	1	Châu Tinh Trì	Văn	6
9	1	Châu Tinh Trì	Anh	8
4	4	Ngôn Nhật Phi	Toán	5
8	4	Ngôn Nhật Phi	Văn	8
12	4	Ngôn Nhật Phi	Anh	8
NULL	NULL	NULL	NULL	NULL

Careful with group by

```
CREATE TABLE HocSinh
(
		HocSinhID int primary key
AUTO_INCREMENT,
		HoTen varchar(20)	NOT NULL,
		DiemMonToan int
);
```

```
INSERT INTO HocSinh(HoTen, DiemMonToan)
VALUES
		('Huy', 90),
		('Việt', 100),
		('Huy', 50);
```

Results | Messages

	HocSinhID	HoTen	DiemMonToan
1	1	Huy	90
2	2	Việt	100
3	3	Huy	50

The Different

	HocSinhID	HoTen	DiemMonToan
▶	1	Huy	90
	2	Việt	100
	3	Huy	50

```
SELECT HoTen, AVG(DiemMonToan) 'Group by HoTen'
FROM HocSinh
GROUP BY HoTen;
```

	HoTen	Group by HoTen
▶	Huy	70.0000
	Việt	100.0000

```
SELECT HoTen, AVG(DiemMonToan) 'Group by HocSinhID, HoTen'
FROM HocSinh
GROUP BY HocSinhID, HoTen;
```

	HoTen	Group by HocSinhID, HoTen
▶	Huy	90.0000
	Việt	100.0000
	Huy	50.0000

HAVING

- The HAVING clause was added to SQL because the WHERE keyword cannot be used with Aggregate functions.

HAVING

```
SELECT LearnSubject, AVG(Score) AVGScore
FROM StudentScore
GROUP BY LearnSubject
```

	LearnSubject	AVGScore
▶	Toán	4.0000
	Văn	7.0000
	Anh	9.0000

```
SELECT LearnSubject, AVG(Score) AVGScore
FROM StudentScore
GROUP BY LearnSubject
HAVING AVG(Score) > 5
```

	LearnSubject	AVGScore
▶	Văn	7.0000
	Anh	9.0000

Practice

```
SELECT StudentID, FullName, AVG(Score) AVGScore
FROM StudentScore
GROUP BY StudentID, FullName
ORDER BY StudentID
```

StudentID	FullName	AVGScore
1	Châu Tinh Trì	5.6667
2	Châu Kiệt Luân	6.3333
3	Châu Nhuận Phát	7.6667
4	Ngôn Nhật Phi	7.0000

StudentID	FullName	AVGScore
2	Châu Kiệt Luân	6.3333
3	Châu Nhuận Phát	7.6667
4	Ngôn Nhật Phi	7.0000

```
SELECT StudentID, FullName, AVG(Score) AVGScore
FROM StudentScore
GROUP BY StudentID, FullName
HAVING AVG(Score) > 6
ORDER BY StudentID
```

Practice Time

```
CREATE TABLE CustomerOrder(
                ID INT PRIMARY KEY AUTO_INCREMENT,
                CustomerID INT,
                FullName VARCHAR(20) NOT NULL,
                DeliveryCity VARCHAR(20),
                DateDelivery DATE,
                OrderDate DATE,
                TotalAmount DECIMAL(10, 2),
                OrderStatus VARCHAR(20)
);
```

Thank you!

*Any **questions** ?*

Extra Resources

Name	Link
became SQL god?	https://www.w3schools.com/sql/default.asp