

# Group By & Join

Dr. Parimala M

SCORE

# Student Particulars- Sample database

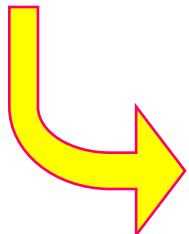
• <u>field</u>	<u>type</u>	<u>width</u>	<u>contents</u>
• <i>id</i>	numeric	4	student id number
• <i>name</i>	character	10	name
• <i>dob</i>	date	8	date of birth
• <i>sex</i>	character	1	sex: M / F
• <i>class</i>	character	2	class
• <i>hcode</i>	character	1	house code: R, Y, B, G
• <i>dcode</i>	character	3	district code
• <i>remission</i>	logical	1	fee remission
• <i>mtest</i>	numeric	2	Math test score

# Student Particulars- Sample database

id	name	dob	sex	class	mtest	hcode	dcode	remission
9801	Peter	06/04/86	M	1A	70	R	SSP	.F.
9802	Mary	01/10/86	F	1A	92	Y	HHM	.F.
9803	Johnny	03/16/86	M	1A	91	G	SSP	.T.
9804	Wendy	07/09/86	F	1B	84	B	YMT	.F.
9805	Tobe	10/17/86	M	1B	88	R	YMT	.F.
:	:	:	:	:	:	:	:	:

# ***Grouping***

**SELECT ..... FROM ..... WHERE *condition* ;  
GROUP BY *groupexpr* [HAVING *requirement*]**



**Group functions:**

**COUNT( ), SUM( ), AVG( ), MAX( ), MIN( )**

- *groupexpr* specifies the related rows to be grouped as one entry. Usually it is a column.
- WHERE *condition* specifies the condition of individual rows before the rows are group. HAVING *requirement* specifies the condition involving the whole group.

# ***Grouping***

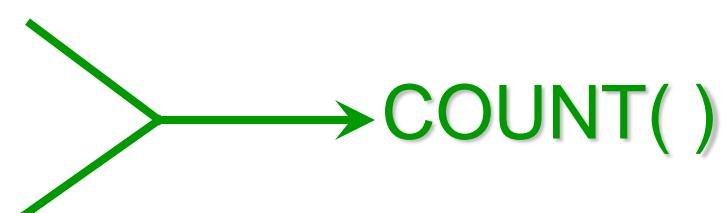
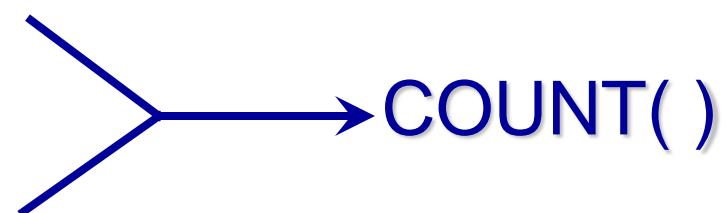
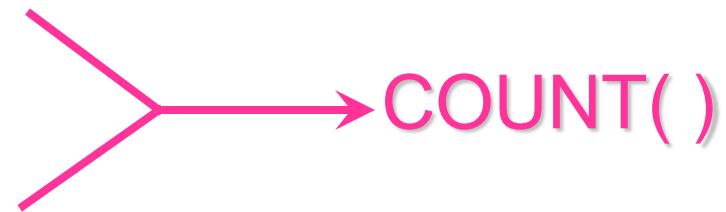
**eg. 11**    List the number of students of each class.

↓ Group By Class

class

		1A		
		1A		
		1A		
		1B		
		1C		
		1C		
		1C		

Student



# ***Grouping***

**eg. 11** List the number of students of each class.

```
SELECT class, COUNT(*) FROM student ;  
GROUP BY class
```

Result

class	cnt
1A	10
1B	9
1C	9
2A	8
2B	8
2C	6

# ***Grouping***

**eg. 12**    List the average Math test score of each class.

↓ Group By Class

class

		1A		
		1A		
		1A		
		1B		
		1C		
		1C		
		1C		

1A {

1B {

1C {

Student

AVG()

AVG()

AVG()

# ***Grouping***

**eg. 12** List the average Math test score of each class.

```
SELECT class, AVG(mtest) FROM student ;  
GROUP BY class
```

Result

class	avg_mtest
1A	85.90
1B	70.33
1C	37.89
2A	89.38
2B	53.13
2C	32.67

# ***Grouping***

**eg. 13** List the number of girls of each district.

Count()

Group by

```
SELECT dcode, COUNT(*) FROM student ;  
WHERE sex="F" GROUP BY dcode
```



dcode	cnt
HHM	6
KWC	1
MKK	1
SSP	5
TST	4
YMT	8

Count() girls

1. Group by –district
2. Stname, district, gender
3. Condition- where clause
4. Count()
5. Condition->3 -----having clause

# ***Grouping***

**eg. 14** List the max. and min. test score of Form 1 students of each district.

```
SELECT MAX(mtest), MIN(mtest), dcode  
FROM student  
WHERE class LIKE "1_" GROUP BY dcode
```

**Result**

max_mtest	min_mtest	dcode
92	36	HHM
91	19	MKK
91	31	SSP
92	36	TST
75	75	TSW
88	38	YMT

# ***Grouping***

**eg. 15** List the average Math test score of the boys in each class. The list should not contain class with less than 3 boys.

```
SELECT AVG(mtest), class FROM student ;  
WHERE sex="M" GROUP BY class ;  
HAVING COUNT(*) >= 3
```



avg_mtest	class
86.00	1A
77.75	1B
35.60	1C
86.50	2A
56.50	2B

Dr. Parimala M, SCORE

# ***Display Order***

```
SELECT ..... FROM ..... WHERE .....
GROUP BY .... ;
ORDER BY colname ASC / DESC
```

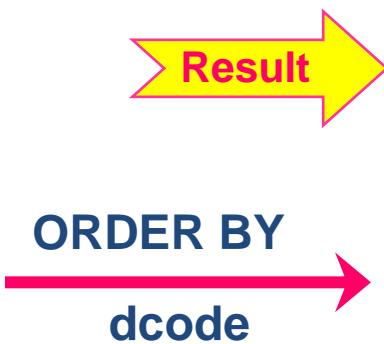
# ***Display Order***

**eg. 16** List the boys of class 1A, order by their names.

SELECT name, id FROM student ;

WHERE sex="M" AND class="1A" ORDER BY name

<b>name</b>	<b>id</b>
Peter	9801
Johnny	9803
Luke	9810
Bobby	9811
Aaron	9812
Ron	9813



<b>name</b>	<b>id</b>
Aaron	9812
Bobby	9811
Johnny	9803
Luke	9810
Peter	9801
Ron	9813

# ***Display Order***

**eg. 17** List the 2A students by their residential district.

```
SELECT name, id, class, dcode FROM student ;  
WHERE class="2A" ORDER BY dcode
```



<b>name</b>	<b>id</b>	<b>class</b>	<b>dcode</b>
Jimmy	9712	2A	HHM
Tim	9713	2A	HHM
Samual	9714	2A	SHT
Rosa	9703	2A	SSP
Helen	9702	2A	TST
Joseph	9715	2A	TSW
Paula	9701	2A	YMT
Susan	9704	2A	YMT

# ***Display Order***

**eg. 18** List the number of students of each district  
(in desc. order).

```
SELECT COUNT(*) AS cnt, dcode FROM student ;  
GROUP BY dcode ORDER BY cnt DESC
```



cnt	dcode
11	YMT
10	HHM
10	SSP
9	MKK
5	TST
2	TSW
1	KWC
1	MMK
1	SHT

## ***Display Order***

**eg. 19** List the boys of each house order by the classes. (2-level ordering)

```
SELECT name, class, hcode FROM student ;  
WHERE sex="M" ORDER BY hcode, class
```

# ***Display Order***

**Result**

**Order  
by  
*hcod*  
e**

**Blue  
Hous  
e**

**Gree  
n  
Hous  
e :**

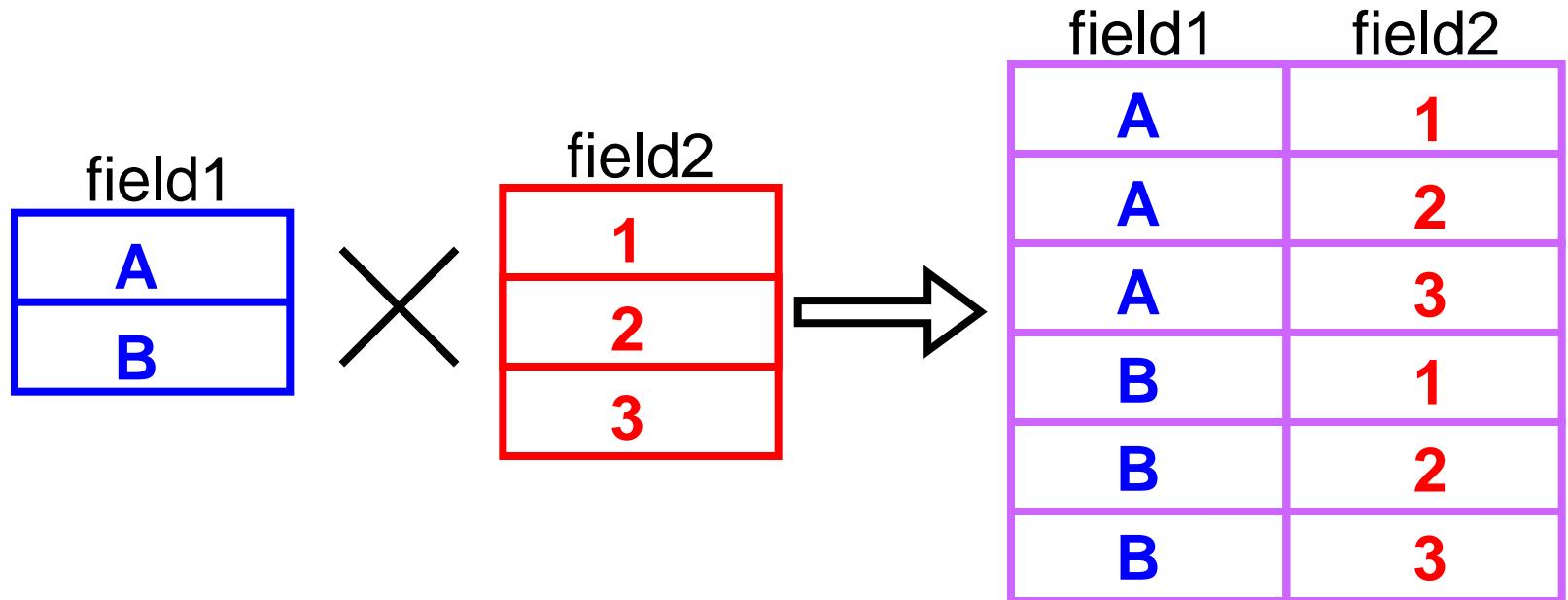
name	hcode	class
Bobby	B	1A
Teddy	B	1B
Joseph	B	2A
Zion	B	2B
Leslie	B	2C
Johnny	G	1A
Luke	G	1A
Kevin	G	1C
George	G	1C
:	:	:

**Order  
by  
*class***

# ***Multiple Tables:***

- SQL provides a convenient operation to retrieve information from multiple tables.
- This operation is called **join**.
- The join operation will **combine** the tables into one large table with all possible combinations (Math: Cartesian Product), and then it will filter the rows of this combined table to yield useful information.

# ***Multiple Tables:***



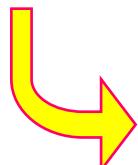
# The Situation: Music Lesson

Each student should learn a musical instrument.  
Music database consists of two fields student id  
and the type of music instrument they learn.

<u>field</u>	<u>type</u>	<u>width</u>	<u>contents</u>
id	numeric	4	student id number
type	character	10	type of the music instrument

## ***Natural Join***

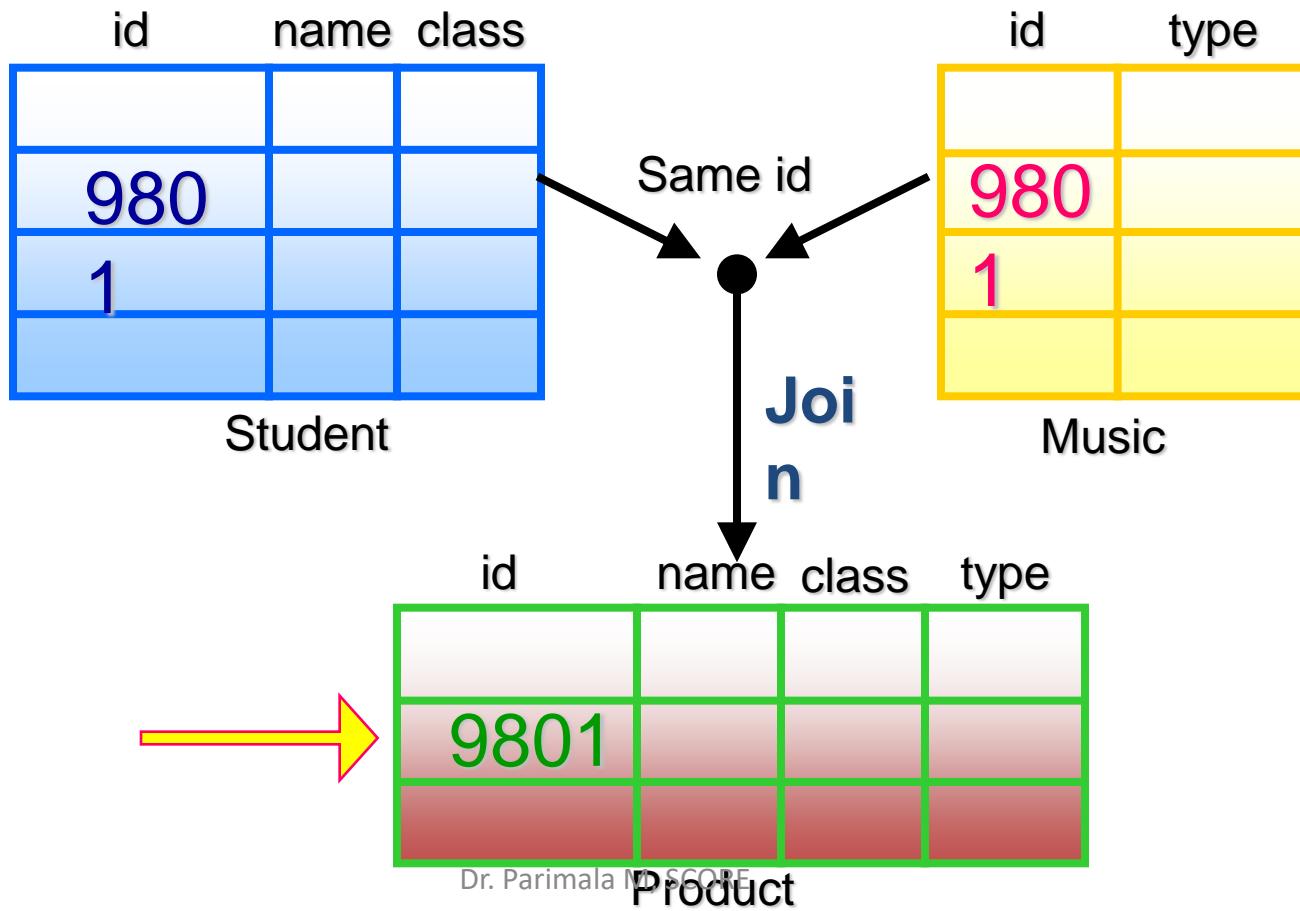
A **Natural Join** is a join operation that joins two tables by their common column. This operation is similar to the setting relation of two tables.



```
SELECT a.comcol, a.col1, b.col2, expr1, expr2 ;  
FROM table1 a, table2 b ;  
WHERE a.comcol = b.comcol
```

# ***Natural Join***

**eg. 25** Make a list of students and the instruments they learn. (Natural Join)



## ***Natural Join***

**eg. 25** Make a list of students and the instruments they learn. (Natural Join)

```
SELECT s.class, s.name, s.id, m.type ;
```

```
FROM student s, music m ;
```

```
WHERE s.id=m.id ORDER BY class, name
```

**Result**

class	name	id	type
1A	Aaron	9812	Piano
1A	Bobby	9811	Flute
1A	Gigi	9824	Recorder
1A	Jill	9820	Piano
1A	Johnny	9803	Violin
1A	Luke	9810	Piano
1A	Mary	9802	Flute
:	:	:	:

# ***Natural Join***

**eg. 26** Find the number of students learning piano in each class.

**Three Parts :**

**(1) Natural Join.**

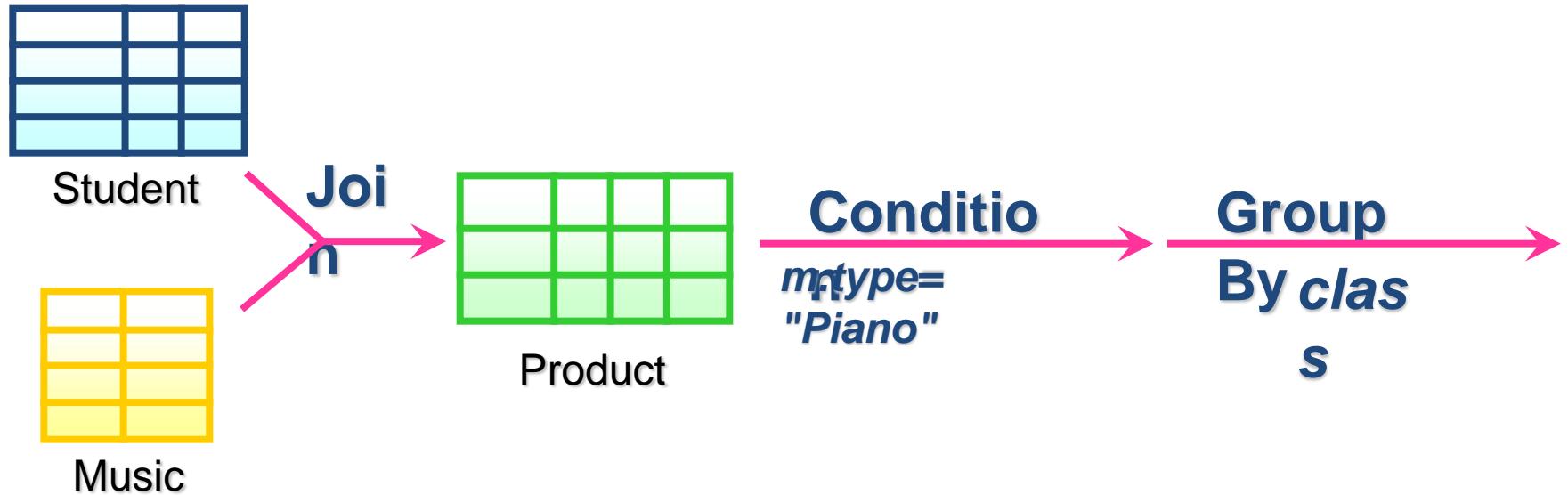
**(2) Condition:**

***m.type="Piano"***

**(3) GROUP BY class**

# ***Natural Join***

eg. 26



# ***Natural Join***

**eg. 26** Find the number of students learning piano in each class.

```
SELECT s.class, COUNT(*) ;  
FROM student s, music m ;  
WHERE s.id=m.id AND m.type="Piano" ;  
GROUP BY class ORDER BY class
```

**Result**

class	cnt
1A	4
1B	2
1C	1