

# summary revision

🕒 Created	@October 2, 2023 9:52 PM
☑ Reviewed	<input type="checkbox"/>
🕒 Last edited time	@October 4, 2023 1:12 PM

Data type:

date: 'YYYY-MM-DD'

integer/decimal(10,2): 90.00 **#no need to ' 90.00 ' no cotation mark**

GROUP BY need to mention in the SELECT

exp:

SELECT Name, \* FROM Book

GROUP BY Name

(will display all the detail because of\* and need the Name use to group by)

the sort of group() order() having():

GROUP BY

HAVING

ORDER BY

---

Lab3

FOREIGN KEY REFERENCES tablename(columnname)

## DISTINCH

only show the same data once however duplicate how many time

SELECT DISTINCH Address FROM Supplier

exp: johor, selangor, penang, selangor, negeri sembilan

will display johor, penang, selangor, negeri sembilan

---

#### Lab 4

ORDER BY more than 1 condition using , to store the two condition

exp:

ORDER BY Name ASC, Price DESC

BETWEEN and NOT BETWEEN

question: except something

exp: except book publisher by PublisherID ' P01'

use: !=

WHERE PublisherID != 'P01'

---

#### Lab 5

##### LIKE

use to find the data using keyword

SELECT \* FROM table WHERE column LIKE 'condition'

'K%' start with k/K after k is anything ( no matter how many k) without limit word

'%K' end with k/K

'%K%' the k must be middle, cannot be first or last letter no matter how many k

'\_k%' second word should be k/K end with any

exp:

SELECT \* FROM Book WHERE Name LIKE 'B%'

the total letter in the word

SELECT \* FROM *table* WHERE *condition 1* AND LEN(*column*) > = *number*

exp:

SELECT \* FROM Book WHERE Name LIKE 'B%' AND LEN(Name) > = 3

\*\* the total letter are lest and equal 3

### IN operator with OR

use to simplify the query compare more data in easier way

SELECT \* FROM table WHERE column IN ('condition1' , 'condition2' , 'condition3')

exactly with : WHERE column = 'condition1' OR column = 'condition2' OR column = 'condition3'

exp:

SELECT \* FROM Publisher WHERE Address IN ('Puchong' , 'Selangor' , 'Penang')

### ALTER TABLE

use to change the properties(data types) of the column(not add)

ALTER TABLE *tablename*

ALTER COLUMN *columnname* *dataproperties(datatypes)*

exp:

ALTER TABLE Book

ALTER COLUMN Name varchar(50)

### **REMARK**

will error if the previous data types is nvarchar(50) and change to integer(or any data types might conflict)

(cannot change word to decimal)

ALTER TABLE *tablename*

ADD *columnname datatype*

exp:

ALTER TABLE Publisher

ADD Telephone int

## REMARK

the telephone cannot put IS NOT NULL (will conflict)

because the telephone 100% IS NULL (we just add new column, dont have add data)

## DELETE

use to delete record/data only

DELETE FROM *tablename*

WHERE *condition*

exp:

DELETE FROM Book

WHERE Name = 'Biology' AND Author = 'J.Wenton'

---

## Lab 7

### NULL

if the data is allow NULL

create table should add NULL after data type

exp:

Price nvarchar(50) NULL

PublisherID (50) FOREIGN KEY REFERENCES Publisher(PublisherID) NULL

### SUB-query:

where foreignkey = (SELECT primary key FROM ...)

or any unique data exist in the both table(mostly is primary key)

exp:

```
SELECT * FROM Book
```

```
WHERE PublisherID = (SELECT PublisherID FROM Publisher WHERE Name =  
'Pearson')
```

### **REMARK**

ORDER BY Name.Book

(cannot use because it have belong to the book table so no need to mention,  
this mostly or maybe only use in the JOIN)

### Update

```
UPDATE table name
```

```
SET column name = 'newdata'
```

```
WHERE primarykey = (SELECT primarykey FROM tablename
```

```
WHERE tablename = 'datadetail')
```

exp:

```
UPDATE Publisher
```

```
SET Address = 'Shah Alam'
```

```
WHERE PublisherID = (SELECT PublisherID FROM Book WHERE Name =  
'Biology')
```

or the direct column in the same table

WHERE PublisherID = 'P01'

### ORDER BY

sort order of the column by selected column

ASC: ORDER BY Name #in ascending order: a-z, 0-9

DESC: ORDER BY Price #in descending order: z-a, 9-0

### GROUP BY

#### HAVING

#### COUNT()

use to count how many time exist in the table

can be COUNT(), as counting total number of the column

or COUNT(PublisherID), as counting the total of publisherID exist in table

exp: P01 got published 2 book, P02 published 3 book, P03 published 2 book

result: 7

### **EXTRA:**

SELECT COUNT(Author) FROM Book

GROUP BY Author

result:

J.Denzin | 1

K.Vince | 2

S.hanson | 1

#will show the total existing group by author name

(know how many book written by who)

### SUM()

SUM(*must have specific column*)

cannot be SUM(\*) because use to total up the number

exp:

SUM(Price)

AVG()

basically similar to sum()

name ... AS

use to name the column name to customize column name mostly use in  
COUNT(), SUM(), AVG()

NOT EXIST

skip

NOT IN/ IN

= is use to compare only 1 data

IN use to compare the data more than 1

replace '='

# the = only use to compare 1 data

exp:

SELECT \* FROM Publisher

WHERE PublisherID NOT IN (SELECT PublisherID FROM Book...)

---

Lab 8

JOIN = INNER JOIN

use to join two table(mostly use)

LEFT JOIN

use to show the data that is only in one table (will not display in join)

exp:

Results		Messages	
	PublisherID	Name	Address
1	P01	Pearson	Bukit Jalil
2	P02	Deitel	Puchong
3	P03	Rainbow	Subang
4	P04	MacHill	Kuala Lumpur

  

	BookID	Name	Author	Price	PublishedDate	PublisherID
1	B01	Maths	J.Wenton	50.60	2016-01-10	P01
2	B02	Science	S.Hanson	100.00	2016-02-12	P01
3	B03	English	K.Vince	89.30	2016-03-09	P02
4	B04	Boilogy	K.Vince	150.80	2016-04-24	P03
5	B05	Computing	J.Denzin	NULL	NULL	NULL

the B05 will not display in JOIN because the PublisherID is NULL

need to use left join if want to display all book join with publisher detail

(full outer join oso can)



```
--q14
SELECT * FROM Book b
LEFT JOIN Publisher p ON p.PublisherID = b.PublisherID
WHERE Author != 'K.Vince'

--q15
SELECT * FROM Book b
FULL OUTER JOIN Publisher p ON p.PublisherID = b.PublisherID
WHERE Author != 'K.Vince'
--q15
```

98 %

Results Messages

	BookID	Name	Author	Price	PublishedDate	PublisherID	PublisherID	Name	Address
1	B01	Maths	J.Wenton	50.60	2016-01-10	P01	P01	Pearson	Bukit Jalil
2	B02	Science	S.Hanson	100.00	2016-02-12	P01	P01	Pearson	Bukit Jalil
3	B05	Computing	J.Denzin	NULL	NULL	NULL	NULL	NULL	NULL

  

	BookID	Name	Author	Price	PublishedDate	PublisherID	PublisherID	Name	Address
1	B01	Maths	J.Wenton	50.60	2016-01-10	P01	P01	Pearson	Bukit Jalil
2	B02	Science	S.Hanson	100.00	2016-02-12	P01	P01	Pearson	Bukit Jalil
3	B05	Computing	J.Denzin	NULL	NULL	NULL	NULL	NULL	NULL

## RIGHT JOIN

same to left join

## FULL OUTER JOIN

## Lab 9

### GETDATE()

use to get the current date at the run moment (will same with computer time)

### DATEDIFF()

*DATEDIFF(year/month/day, columnA, columnB)*

use to get the date different based on year, month, day between column A and column B

column a and column b can use table column or column b use GETDATE() to calculate the different between date and today

exp:

DATEDIFF(year, s.DOB, GETDATE())

### TEMP

SELECT \* FROM (SELECT \* *count()*, *sum()*, *datediff* or any thing AS *customizenam* FROM *table*) AS TEMP

WHERE *customizenam* with any condition

exp:

SELECT \* FROM (SELECT datediff(year, DOB, getdate()) AS Age FROM Student)  
AS TEMP

WHERE Age > AVG(AGE)

### JOIN(multiple table)

using the common column (mostly primary key and foreign key) to combine multiple table

should base on the table, A combine B, B combine C

\*\*no A combine C, D combine B should have common data between 2 table

SELECT \* FROM Table1 T1

JOIN Table2 T2 ON T1.PRIMARYKEY = T2.PRIMARYKEY

JOIN Table3 T3 ON T2.PRIMARYKEY = T3.PRIMARYKEY

exp:

SELECT \* FROM Course c

JOIN Student s ON c.CourseID = s.CourseID

JOIN Subject sj ON s.COURSEID = sj.CourseID

### WITHOUT JOIN(to combine table)

```
SELECT * FROM Table1 T1, Table 2 T2  
WHERE T1.PRIMARYKEY = T2.FOREIGNKEY  
AND othercondition
```