

8. MySQL

- 1) Why Database?
- 2) What is DBMS?
- 3) Create & Drop Database
- 4) Create & Drop Table
- 5) SELECT Statement
- 6) WHERE Clause
- 7) AND,OR Operators
- 8) DISTINCT,LIMIT,LIKE,IN,NOT Keywords
- 9) INSERT Statement
- 10) UPDATE Statement
- 11) DELETE Statement
- 12) ORDER BY
- 13) GROUP BY
- 14) Aggregate Function
- 15) IS NULL,NOT NULL
- 16) JOIN Queries

1) Why Database?

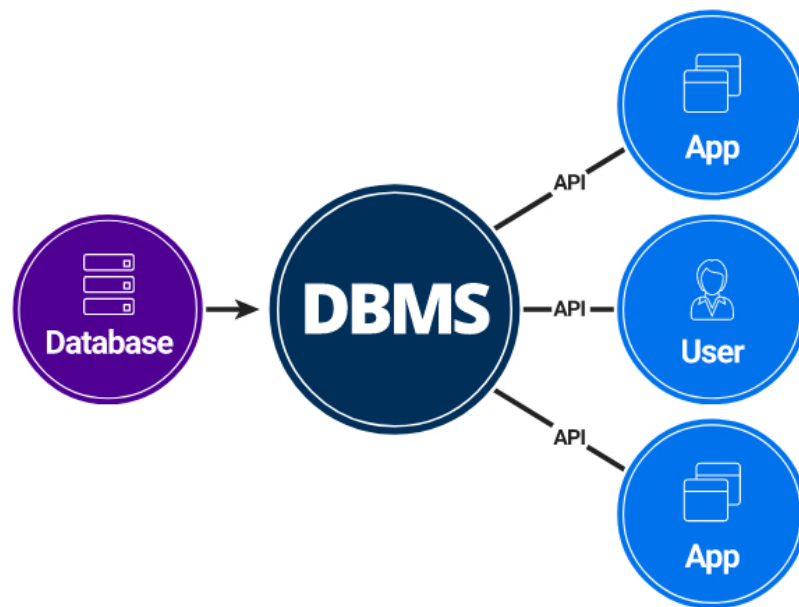
Database ဆိုတာကတော့ data တွေကို သိမ်းဆည်းထားနိုင်၊ဖန်တီးနိုင်၊ပြောင်းလဲနိုင်၊ ဖျက်ဆီးနိုင်တဲ့ electric system နဲ့ထိန်းသိမ်းထားနိုင်တဲ့ အစုအဝေးတခုဖြစ်ပါတယ်။ ကုမ္ပဏီတခုရဲ့ process မှာဆိုရင် Customerရဲ့Record တွေ၊ Employee ရဲ့ data တွေ၊ bank transactions တွေစတဲ့ လုပ်ငန်းတွေကို မှတ်သားဖို့အတွက် ဖိုင်ပေါင်းများစွာ လိုအပ်ပါတယ်။ On paper ပေါ်မှာမှတ်သားလိုက်ခြင်းဟာ human fault ကြောင့် သော်လည်းကောင်း၊အကြောင်း အမျိုးမျိုးကြောင့်ပျက်စီးသွားစေနိုင်ပါတယ်။ Securityအရလုံခြုံမှုကိုမရရှိစေနိုင်ပါဘူး။ large amount data record တွေကို သိမ်းဆည်းထားနိုင်ဖို့ space problem တွေဖြစ်ပေါ်လာစေပါတယ်။

ဒီလိုပြဿနာတွေကြောင့် တနေရာထဲမှာ တစုတဝေးထဲ data တွေကို collect လုပ်နိုင်ဖို့၊ လိုအပ်လာတဲ့အခါမှာလည်း လွယ်လွယ်ကူကူ ထုတ်ပြန်နိုင်ဖို့၊ပြန်လည်ပြင်ဆင်နိုင်ဖို့ Data Store အဖြစ် Database ကိုသုံးကြပါတယ်။

DATABASE မှာတော့ MySQL အပြင် တခြားသော Database system တွေများစွာရှိပါတယ်။ MSSQL,PostgreSQL,Oracle...

Database ရဲ့data တွေကိုသိမ်းဆည်းပုံကတော့ Table တွေနဲ့Group လိုက် သိမ်းဆည်းတဲ့ပုံစံမျိုးနဲ့ သိမ်းထားတာဖြစ်ပါတယ်။ Table တွေအချင်းချင်း ချိတ်နိုင်ပြီး collect လုပ်နိုင်ပြီး၊လိုချင်တဲ့ data တွေဖြစ်အောင် condition တွေစစ်ပြီးထုတ်နိုင်ပါတယ်။

2) What is DBMS?



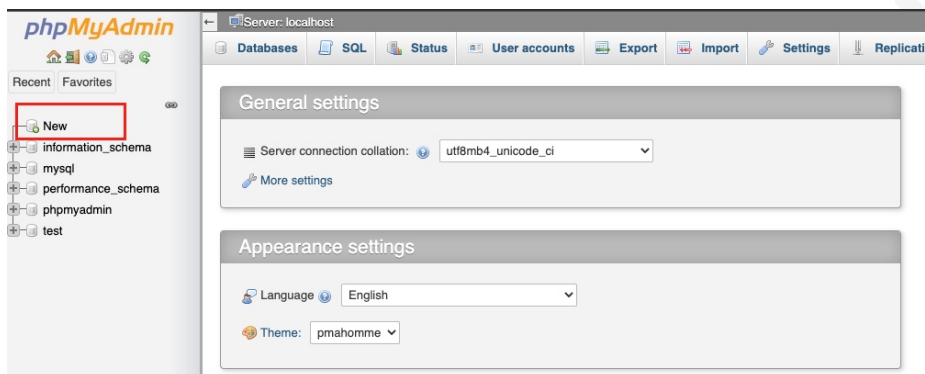
Database က data တွေကို electronic system နဲ့သိမ်းဆည်းထားတာဖြစ်လို့ အသုံးပြုသူ end user တွေအနေနဲ့ management တွေလုပ်နိုင်ဖို့ ကြားခံ interface အနေနဲ့ DBMS ကို အသုံးပြုကြပါတယ်။ DBMS ဟာ software တခုဖြစ်ပါတယ်။ end user တွေနဲ့ တခြား third-party software တွေနဲ့ database ကို communication လုပ်နိုင်ဖို့အတွက် အသုံးပြုပါတယ်။ Software based (MySQL Workbench, MSSQL Server...) နဲ့ web based interface (phpmyadmin) စသည်ဖြင့်ခွဲခြားထားပါတယ်။

3) Create & Drop Database

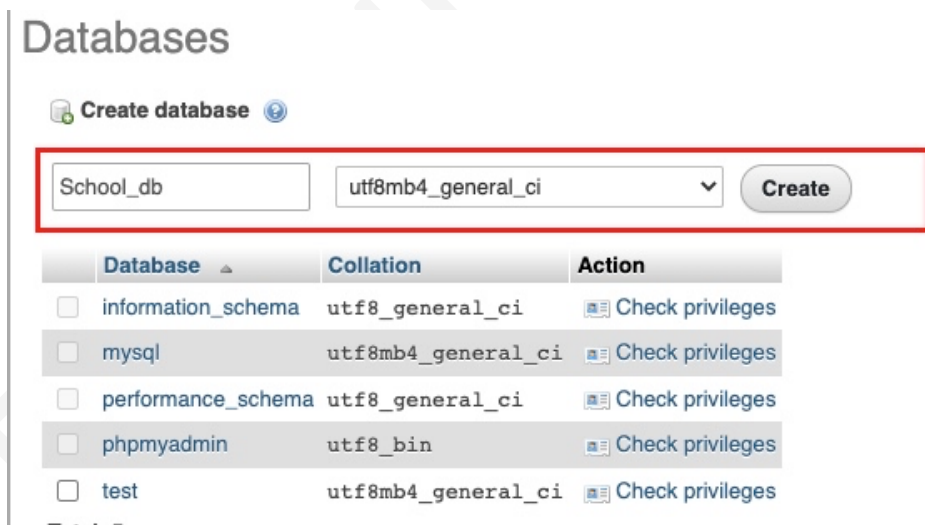
အသုံးပြုမယ့် Database တစ်ခုကိုတည်ဆောက်ဖို့အရင်ဆုံးလိုအပ်ပါတယ်။
Manual သို့မဟုတ် Query နဲ့တည်ဆောက်နိုင်ပါတယ်။

Manual Create Database

Step 1: New ကိုနှိပ်ပါ။



Step 2: Database နာမည်၊ Collation ကိုထည့်ပြီး Create ကိုနှိပ်ပါ။



Query Create Database

Databases SQL Status User accounts Export Import Settings More

Run SQL query/queries on server "localhost":

```
CREATE DATABASE School_db COLLATE utf8_general_ci;
```

Clear Format Get auto-saved query

☐ Bind parameters

Bookmark this SQL query:

Delimiter ; ☐ Show this query here again ☐ Retain query box ☐ Rollback when finished

☒ Enable foreign key checks

Go

တည်ဆောက်ထားတဲ့ Database ကို ဖျက်ဖို့အတွက် Drop ကိုသုံးရပါတယ်။

Manual Drop Database

Databases

Create database

Database name utf8mb4_general_ci Create

Database	Collation	Action
<input type="checkbox"/> information_schema	utf8_general_ci	Check privileges
<input type="checkbox"/> mysql	utf8mb4_general_ci	Check privileges
<input type="checkbox"/> performance_schema	utf8_general_ci	Check privileges
<input type="checkbox"/> phpmyadmin	utf8_bin	Check privileges
<input checked="" type="checkbox"/> School_db	utf8_general_ci	Check privileges
<input type="checkbox"/> test	utf8mb4_general_ci	Check privileges

Total: 6

Check all With selected: Drop

Query Drop Database

Databases SQL Status User accounts Export Import Settings More

Run SQL query/queries on server "localhost":

```
1 DROP DATABASE School_db;
```

Clear Format Get auto-saved query

☐ Bind parameters

Bookmark this SQL query:

Delimiter ; ☐ Show this query here again ☐ Retain query box ☐ Rollback when finished

☒ Enable foreign key checks

Go

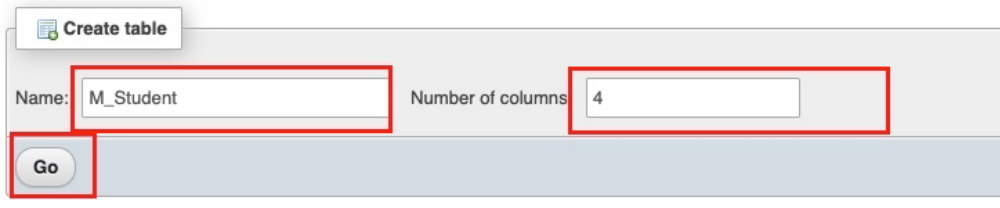
4) Create & Drop Table

Database တခုကိုတည်ဆောက်ပြီးတဲ့အခါ Data record တခုချင်းစီကို group အလိုက်ထည့်နိုင်ဖို့ Table တွေထည်ဆောက်ရပါတယ်။

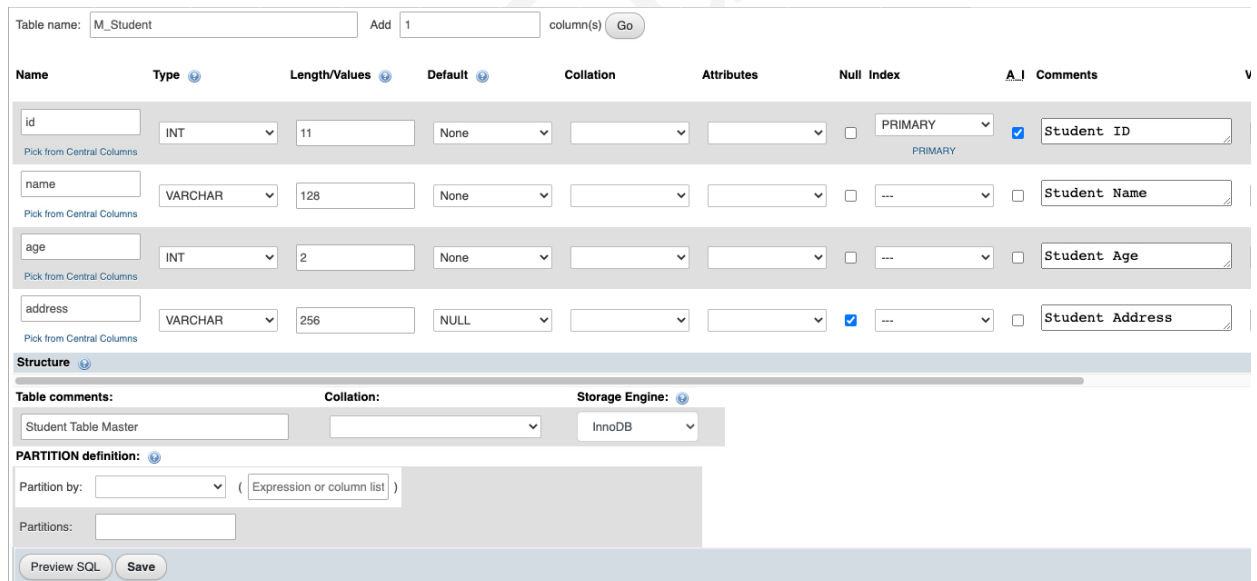
Manual နဲ့ Create လုပ်နိုင်သလို Query နဲ့လည်း Create လုပ်နိုင်ပါတယ်။

Manual Create Table

Step 1: Table နာမည်၊ ပါဝင်မည့် Columns အရေအတွက်ကိုထည့်ပြီး Go ကိုနှိပ်ပါ။



Step 2: လိုအပ်တဲ့ information တွေဖြည့်ပါ။



Name : ပေးချင်တဲ့ Column နာမည်

Type: Datatype ပုံစံ String ဆို Varchar၊ Integer ဆို int စသည်ဖြင့် ဝင်လာမည့် data ရဲ့ type

Length/Values: ထည့်နိုင်တဲ့ data length

Default: data record တွေအသစ်တွေထည့်တဲ့အခါ လက်ရှိcolumn ရဲ့ တန်ဖိုးပါမလာတဲ့အခါ Default အနေနဲ့ ထည့်ထားချင်တဲ့ တန်ဖိုး။
Null: လက်ရှိcolumn က NULL တန်ဖိုးလက်ခံမခံ။

Index:

Index :data record တွေဟာ million နဲ့ချီပြီ တည်ရှိနေတဲ့အခါ data search ရှာဖွေတဲ့အခါ အချိန်အများကြီး ကြာပါတယ်။ Data collect or searching speed မြန်နိုင်ဖို့အတွက် index တွေကို အသုံးပြုရပါတယ်။

Primary :ကတော့ data တွေ တခုနဲ့တခု မထပ်နိုင် primaryအနေနဲ့ထားတာ ဖြစ်ပါတယ်။

Unique : Primary နဲ့ လုပ်ဆောင်ပုံခြင်းတူပါတယ်။သို့ပေမယ့် primary က NULL value ကို လက်မခံပါဘူး။Unique ကတော့ NULL ကို လက်ခံပါတယ်။

AI: Data record တခုထည့်လိုက်တိုင်း Auto Increment ဖြစ်နေစေချင်တဲ့အခါ သုံးပါတယ်။

Comment: Column ရဲ့ information ကိုမှတ်ထားတဲ့အနေနဲ့ သုံးပါတယ်။

Query Create Table

```
CREATE TABLE `School_db`.(  
  `id` INT(11) NOT NULL AUTO_INCREMENT COMMENT 'Student ID',  
  `name` VARCHAR(128) NOT NULL COMMENT 'Student Name',  
  `age` INT(2) NOT NULL COMMENT 'Student Age',  
  `address` INT(256) NULL DEFAULT NULL COMMENT 'Student Address',  
  PRIMARY KEY(`id`)  
) ENGINE = INNODB COMMENT = 'Student Table Master';
```


5) SELECT Statement

Syntax

SELECT *column* **FROM** *table_name*;

SELECT Query ကတော့ database base ထဲက data records တွေကို ထုတ်ပြတဲ့အနေနဲ့ သုံးပါတယ်။ column နေရာမှာတော့ table ရဲ့ column name တွေ ထည့်နိုင်သလို table ရဲ့ column တွေအားလုံးထုတ်ချင်တဲ့အခါ * လိုမျိုး keyword နဲ့လည်းထုတ်လို့ရပါတယ်။

6) WHERE Clause

SQL statement တွေကို ကိုယ်လိုချင်တဲ့ condition တွေသတ်မှတ်ပြီး Data record တွေကို SELECT လုပ်တာ၊ UPDATE လုပ်တဲ့နေရာတွေမှာသုံးပါတယ်။

SELECT Query

SELECT *column* **FROM** *table_name* **WHERE** *condition*

UPDATE Query

UPDATE
table_name **SET**
column1=value,
column2=value2,... **WHERE** *condition*

7) AND,OR Operators

Expression တခုနဲ့တခု ဆက်ပေးတဲ့အခါ အသုံးပြုပါတယ်။

SELECT *column* **FROM** *table_name* **WHERE** *expression1* **AND** *expression2*

SELECT *column* **FROM** *table_name* **WHERE** *expression1* **OR** *expression2*

8) DISTINCT,LIMIT,LIKE,IN NOT Keywords

DISTINCT Clause ကတော့ data record တွေထုတ်တဲ့အခါ တူညီနေတဲ့ record တွေကို ဖယ်ထုတ်ပေးပြီး တစ်ခုတည်းအနေနဲ့ပြသပေးပါတယ်။

```
SELECT DISTINCT expression FROM table_name WHERE condition ;
```

LIMIT Clause ကတော့ data record တွေကို range သတ်မှတ်ပြီးထုတ်ချင်တဲ့အခါ သုံးပါတယ်။

```
LIMIT row_count  
LIMIT offset,row_count
```

LIKE Clause ကတော့ WHERE Clause နဲ့တွဲအသုံးပြုပါတယ်။LIKE မှာအသုံးပြုနိုင်တဲ့patter တွေရရှိပါတယ်။

```
WHERE column LIKE pattern
```

Pattern 1

```
WHERE column LIKE '%a'
```

Pattern 2

```
WHERE column LIKE 'a%'
```

Pattern 3

```
WHERE column LIKE '%a%'
```

IN Clause ကတော့ record တွေကို filter စစ်ထုတ်တဲ့အခါ အသုံးပြုပါတယ်။

```
SELECT column_name FROM table_name  
WHERE column_name IN (value1, value2, ...);
```

NOT ကတော့ Negative နေရာမှာအသုံးပြုပါတယ်။

```
NOT IN (value1, value2, ...)
```

9) INSERT Statement

Syntax

```
INSERT INTO table_name (column1,column2,...)  
VALUES (value1,value2,...)
```

INSERT Query ကတော့ database ထဲ data အသစ်တွေ ထည့်တဲ့သုံးတဲ့ keyword တခုဖြစ်ပါတယ်။ ထည့်ချင်တဲ့ column name တွေကိုသတ်မှတ်ပြီး Values မှာတော့ သတ်မှတ်ထားမယ် တန်ဖိုးတွေထည့်ပေးရမှာဖြစ်ပါတယ်။

10) UPDATE Statement

Syntax

```
UPDATE table_name SET  
column1=value,  
column2=value2,... WHERE condition
```

ထည့်ပြီးသား data record တွေကို ပြန်ပြင်ချင်တဲ့အခါ Update Statement ကိုအသုံးပြုပါတယ်။ WHERE ကတော့ Target ထားမယ့် column ကသတ်မှတ်ထားဖြစ်လို့ ထည့်ပေးရပါတယ်။

11) DELETE Statement

Syntax

```
DELETE FROM table_name WHERE condition
```

ထည့်ပြီးသား data records တွေကို ဖျက်ပစ်ချင်တဲ့အခါ သုံးပါတယ်။ ဖျက်ချင်တဲ့ record ကို target ထားပြီး ဖျက်တာကြောင့် WHERE Condition လိုအပ်ပါတယ်။

12) ORDER BY

Data record တွေကို sorting လုပ်တဲ့နေရာမှာအသုံးပြုပါတယ်။ASC နဲ့ sorting စီနိုင်သလို DESC နဲ့လည်းစီနိုင်ပါတယ်။

`ORDER BY column1, column2, ... ASC|DESC;`

13) GROUP BY

Data record တွေကို အုပ်စုလိုက်တစုတည်းစုစည်းပြီးထုတ်ပြတဲ့အခါသုံးပါတယ်။

GROUP BY ကိုတော့ Aggregate function တွေနဲ့တွဲသုံးတာများပါတယ်။

`GROUP BY column_name;`

14) Aggregate Function

id	name	category	initPrice	salePrice	stock	del_flg	create_date	update_date
1	Mens Shirt	Shirt	500	600	100	0	2021/11/10	2021/11/10
2	Kid Shoe	Shoe	50	80	100	1	2021/11/10	NULL
3	Womens	Pants	150	200	10	1	2021/11/10	2021/11/10
4	Men	Pants	500	600	5	0	2021/11/10	NULL
5	Kid Shoe	Shoe	50	80	100	0	2021/11/10	NULL

▪ Count()

Data record ရဲ့အရေအတွက်ကို လိုချင်တဲ့အခါအသုံးပြုပါတယ်။

```
SELECT COUNT(id) FROM `M_Product` WHERE del_flg = 0;  
//3
```

▪ Sum()

Data record(integer) တွေရဲ့အားလုံးပေါင်းခြင်းကိုလိုချင်တဲ့အခါ အသုံးပြုပါတယ်။

```
SELECT SUM(initPrice) FROM `M_Product` WHERE del_flg = 0;  
//1050
```

- **Average()**

Data record(integer) တွေရဲ့ပျမ်းမျှတန်ဖိုးကိုလိုချင်တဲ့အခါ အသုံးပြုပါတယ်။

```
SELECT AVG(initPrice) FROM `M_Product` WHERE del_flg = 0;  
//350.00
```

- **Min()**

Data record(integer) တွေထဲက အနည်းဆုံးတန်ဖိုးကို လိုချင်တဲ့အခါ အသုံးပြုပါတယ်။

```
SELECT MIN(initPrice) FROM `M_Product` WHERE del_flg = 0;  
//50
```

- **Max()**

Data record(integer) တွေထဲက အများဆုံးတန်ဖိုးကို လိုချင်တဲ့အခါ အသုံးပြုပါတယ်။

```
SELECT MAX(initPrice) FROM `M_Product` WHERE del_flg = 0;  
//500
```

15) **IS NULL,NOT NULL**

IS NULL ကတော့ column ရဲ့တန်ဖိုးတွေက NULL ဖြစ်နေတဲ့ record တွေကိုလိုချင်တဲ့အခါသုံးပါတယ်။

NOT NULL ကတော့ IS NULL ဆန့်ကျင်ဘက်ဖြစ်ပြီးတော့ NULL ဖြစ်မနေတဲ့ တန်ဖိုးတွေကို ထုတ်ပြချင်တဲ့အခါသုံးပါတယ်။

16) JOIN Queries

Table တစ်ခုနဲ့တစ်ခု ချိတ်ဆက်ပြီး data record တွေထုတ်ပြချင်တဲ့အခါ join query တွေကိုအသုံးပြုပါတယ်။ Join မှာ ၄မျိုး ရှိပါတယ်။

Example:

M_Customer

id	firstname	lastname	address	membership
1	John	Smith	No 3 Apple Street Yangon ,Myanmar	0
2	Mary	Lynn	No 12 Apple Street Mandalay ,Myanmar	0
3	Brad	David	014-1235 Tokyo Japan	1
4	Yuri	Hana	123-3435 Hokkaido Japan	1

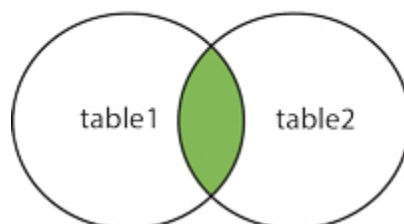
M_Order

id	customer_id	product_id	quantity	buy_date
1	1	2	2	2021/11/01
2	2	4	3	2021/11/02
3	3	4	5	2021/11/05
4	5	1	6	2021/11/06

INNER JOIN

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```

INNER JOIN

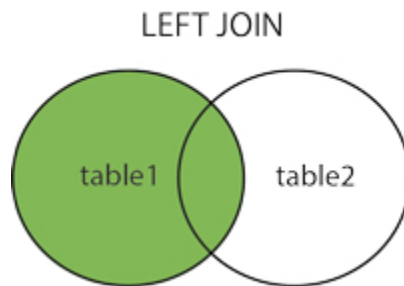


```
SELECT * FROM M_Order
INNER JOIN M_Customer
ON M_Order.customer_id = M_Customer.id;
```

id	customer_id	product_id	quantity	buy_date	id	firstname	lastname	address	membership
1	1	2	2	2021/11/01	1	John	Smith	No 3 Apple Street Yangon ,Myanmar	0
2	2	4	3	2021/11/02	2	Mary	Lynn	No 12 Apple Street Mandalay ,Myanmar	0
3	3	4	5	2021/11/05	3	Brad	David	014-1235 Tokyo Japan	1

LEFT JOIN

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name = table2.column_name;
```

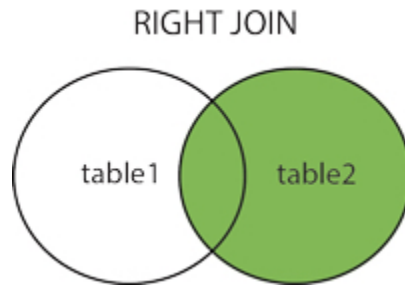


```
SELECT * FROM M_Order
LEFT JOIN M_Customer
ON M_Order.customer_id = M_Customer.id;
```

id	customer_id	product_id	quantity	buy_date	id	firstname	lastname	address	membership
1	1	2	2	2021/11/01	1	John	Smith	No 3 Apple Street Yangon ,Myanmar	0
2	2	4	3	2021/11/02	2	Mary	Lynn	No 12 Apple Street Mandalay ,Myanmar	0
3	3	4	5	2021/11/05	3	Brad	David	014-1235 Tokyo Japan	1
4	5	1	6	2021/11/06	NULL	NULL	NULL	NULL	NULL

RIGHT JOIN

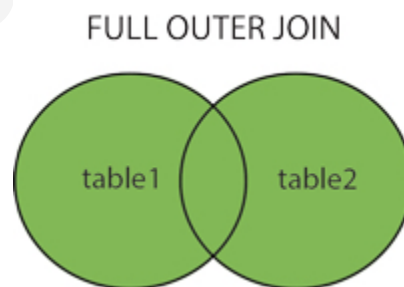
```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;
```



```
SELECT * FROM M_Order
RIGHT JOIN M_Customer
ON M_Order.customer_id = M_Customer.id;
```

id	customer_id	product_id	quantity	buy_date	id	firstname	lastname	address	membership
1	1	2	2	2021/11/01	1	John	Smith	No 3 Apple Street Yangon ,Myanmar	0
2	2	4	3	2021/11/02	2	Mary	Lynn	No 12 Apple Street Mandalay ,Myanmar	0
3	3	4	5	2021/11/05	3	Brad	David	014-1235 Tokyo Japan	1
NULL	NULL	NULL	NULL	NULL	4	Yuri	Hana	123-3435 Hokkaido Japan	1

FULL JOIN (UNION ALL)



```
SELECT * FROM M_Order
LEFT JOIN M_Customer ON M_Order.customer_id = M_Customer.id
UNION ALL
SELECT * FROM M_Order
RIGHT JOIN M_Customer ON M_Order.customer_id = M_Customer.id;
```


id	customer_id	product_id	quantity	buy_date	id	firstname	lastname	address	membership
1	1	2	2	2021/11/01	1	John	Smith	No 3 Apple Street Yangon ,Myanmar	0
2	2	4	3	2021/11/02	2	Mary	Lynn	No 12 Apple Street Mandalay ,Myanmar	0
3	3	4	5	2021/11/05	3	Brad	David	014-1235 Tokyo Japan	1
4	5	1	6	2021/11/06	NULL	NULL	NULL	NULL	NULL
1	1	2	2	2021/11/01	1	John	Smith	No 3 Apple Street Yangon ,Myanmar	0
2	2	4	3	2021/11/02	2	Mary	Lynn	No 12 Apple Street Mandalay ,Myanmar	0
3	3	4	5	2021/11/05	3	Brad	David	014-1235 Tokyo Japan	1
NULL	NULL	NULL	NULL	NULL	4	Yuri	Hana	123-3435 Hokkaido Japan	1