MySQL定義欄位內容型態：

**數字類型**

|  |  |
| --- | --- |
| **INT** | * 正常大小的整數，可以帶符號。如果是有符號的，它允許的範圍是從   -2147483648到2147483647。如果是無符號，允許的範圍是從0到4294967295。 可以指定多達11位的寬度。 |
| **TINYINT** | * 一個非常小的整數，可以帶符號。如果是有符號，它允許的範圍是從-128到127。如果是無符號，允許的範圍是從0到255，可以指定多達4位數的寬度。 |
| **SMALLINT** | 一個小的整數，可以帶符號。如果有符號，允許範圍為-32768至32767。如果無符號，允許的範圍是從0到65535，可以指定最多5位的寬度。 |
| **MEDIUMINT** | 一個中等大小的整數，可以帶符號。如果有符號，允許範圍為-8388608至8388607。 如果無符號，允許的範圍是從0到16777215，可以指定最多9位的寬度。 |
| **BIGINT** | 一個大的整數，可以帶符號。如果有符號，允許範圍為-223372036854775808到9223372036854775807。如果無符號，允許的範圍是從0到18446744073709551615. 可以指定最多20位的寬度。 |
| **FLOAT(M,D)** | 不能使用無符號的浮點數字。可以定義顯示長度(M)和小數位數(D)。這不是必需的，並且默認為10,2。其中2是小數的位數，10是數字(包括小數)的總數。小數精度可以到24個浮點。 |
| **DOUBLE(M,D)** | 不能使用無符號的雙精度浮點數。可以定義顯示長度(M)和小數位數(D)。 這不是必需的，默認為16,4，其中4是小數的位數。小數精度可以達到53位的DOUBLE。 REAL是DOUBLE同義詞。 |
| **DECIMAL(M,D)** | 非壓縮浮點數不能是無符號的。在解包小數，每個小數對應於一個字節。定義顯示長度(M)和小數(D)的數量是必需的。 NUMERIC是DECIMAL的同義詞。 |

**日期和時間類型**

|  |  |
| --- | --- |
| **DATE** | 以YYYY-MM-DD格式的日期，在1000-01-01和9999-12-31之間。 例如，1973年12月30日將被存儲為1973-12-30。 |
| **DATETIME** | 日期和時間組合以YYYY-MM-DD HH:MM:SS格式，在1000-01-01 00:00:00 到9999-12-31 23:59:59之間。例如，1973年12月30日下午3:30，會被存儲為1973-12-30 15:30:00。 |
| **TIMESTAMP** | 1970年1月1日午夜之間的時間戳，到2037的某個時候。這看起來像前麵的DATETIME格式，無需隻是數字之間的連字符; 1973年12月30日下午3點30分將被存儲為19731230153000(YYYYMMDDHHMMSS)。 |
| **TIME** | 存儲時間在HH:MM:SS格式。 |
| **YEAR(M)** | 以2位或4位數字格式來存儲年份。如果長度指定為2(例如YEAR(2))，年份就可以為1970至2069(70〜69)。如果長度指定為4，年份範圍是1901-2155，默認長度為4。 |

**字符串類型**

|  |  |
| --- | --- |
| **CHAR(M)** | 固定長度的字符串是以長度為1到255之間個字符長度(例如：CHAR(5))，存儲右空格填充到指定的長度。 限定長度不是必需的，它會默認為1。 |
| **VARCHAR(M)** | 可變長度的字符串是以長度為1到255之間字符數(高版本的MySQL超過255); 例如： VARCHAR(25). 創建VARCHAR類型字段時，必須定義長度。 |
| **BLOB 或 TEXT** | 字段的最大長度是65535個字符。 BLOB是“二進製大對象”，並用來存儲大的二進製數據，如圖像或其他類型的文件。定義為TEXT文本字段還持有大量的數據; 兩者之間的區彆是，排序和比較上存儲的數據，BLOB大小寫敏感，而TEXT字段不區分大小寫。不用指定BLOB或TEXT的長度。 |
| **TINYBLOB 或 TINYTEXT** | BLOB或TEXT列用255個字符的最大長度。不指定TINYBLOB或TINYTEXT的長度。 |
| **MEDIUMBLOB 或 MEDIUMTEXT** | BLOB或TEXT列具有16777215字符的最大長度。不指定MEDIUMBLOB或MEDIUMTEXT的長度。 |
| **LONGBLOB 或 LONGTEXT** | BLOB或TEXT列具有4294967295字符的最大長度。不指定LONGBLOB或LONGTEXT的長度。 |
| **ENUM** | 這是一個奇特的術語列表。當定義一個ENUM，要創建它的值的列表，這些是必須用於選擇的項(也可以是NULL)。例如，如果想要字段包含“A”或“B”或“C”，那麼可以定義為ENUM為 ENUM(“A”，“B”，“C”)也隻有這些值(或NULL)才能用來填充這個字段。 |

**MySQL Command 指令**

登入MySQL使用介面

先在windows使用cmd呼叫出指令介面，後續再來下相關的Command指令。

**登入MySQL指令**

|  |
| --- |
| mysql -u root -p |

**創建資料庫**

|  |
| --- |
| create database yiibai; |

**刪除資料庫**

|  |
| --- |
| drop database yiibai; |

**選擇資料庫為 test :**

|  |
| --- |
| Show Databases; //顯示所有資料庫  use test; |

**建立資料表CREATE TABLE table\_name (column\_name column\_type);**

|  |
| --- |
| create table tutorials\_tbl(  tutorial\_id INT NOT NULL AUTO\_INCREMENT,  tutorial\_title VARCHAR(100) NOT NULL,  tutorial\_author VARCHAR(40) NOT NULL,  submission\_date DATE,  PRIMARY KEY ( tutorial\_id )  ); |

**Drop Table刪除表**

|  |
| --- |
| DROP TABLE tutorials\_tbl; |

**Insert插入資料INSERT INTO table\_name ( field1, field2,...fieldN ) VALUES ( value1, value2,...valueN );**

|  |
| --- |
| INSERT INTO tutorials\_tbl (tutorial\_title, tutorial\_author, submission\_date)  VALUES  ("Learn MySQL", "Yiibai", NOW());  INSERT INTO tutorials\_tbl  (tutorial\_title, tutorial\_author, submission\_date)  VALUES  ("JAVA Tutorial", "Sanjay", '2022-09-06');  INSERT INTO tutorials\_tbl  (tutorial\_title, tutorial\_author, submission\_date)  VALUES  ("Learn PHP", "John Poul", '2022-09-18'); |

**Select 查詢資料**

|  |
| --- |
| SELECT \* from tutorials\_tbl; |

**Where 條件判斷式**

|  |  |  |
| --- | --- | --- |
| **符號** | **描述** | **範例** |
| = | 檢查兩個操作數的值是否相等，如果是，則條件變為真。 | (A = B) 不為 true. |
| != | 檢查兩個操作數的值是否相等，如果值不相等，則條件變為真。 | (A != B) 為 true. |
| > | 檢查左操作數的值是否大於右操作數的值，如果是，則條件為真。 | (A > B) 不為 true. |
| < | 檢查左操作數的值是否小於右操作數的值，如果是，則條件為真。 | (A < B) 為 true. |
| >= | 檢查左操作數的值是否大於或等於右操作數的值，如果是，則條件為真。 | (A >= B) 不為 true. |
| <= | 檢查左操作數的值是否小於或等於右操作數的值，如果是，則條件變為真。 | (A <= B) 為 true |
| between | 取得介於起始值與指定末值的資料 | Between A and B |
| like | 檢查資料字串內，有符合指定字串資料。  %字符来表示任意字符 | Like ‘abc%’ |
| left | 從左開始擷取字串，left(被擷取字串， 擷取長度) | left(’abcdefg’, 3) |
| right | 從右開始擷取字串，right(被擷取字串， 擷取長度) | right(’abcdefg’, 3) |
| substring | 擷取特定長度的字串，substring(被擷取字串，從第幾位開始擷取，擷取長度) | SUBSTRING('www.yuanrengu.com', 9, 3) |

|  |
| --- |
| SELECT \* from tutorials\_tbl WHERE tutorial\_author='yiibai';  SELECT \* from tutorials\_tbl WHERE submission\_date between '2022-10-01' and '2022-10-31';  SELECT \* from tutorials\_tbl WHERE tutorial\_title like 'JAVA %'; |

**Update更新資料**

|  |
| --- |
| UPDATE tutorials\_tbl SET tutorial\_title='Learning JAVA' WHERE tutorial\_title = “Learn PHP”; |

**Delete刪除資料**

|  |
| --- |
| DELETE FROM tutorials\_tbl WHERE tutorial\_title ='Learning JAVA'; |

**Order By排序結果**

**ASC 為遞增; DESC為遞減;**

|  |
| --- |
| SELECT \* from tutorials\_tbl ORDER BY tutorial\_author ASC |

**Join聯接**

**可以在單個SQL查詢中使用多個表。用一行指令連接兩個或多個資料表查詢資料。**

**假設我們有兩個表 tcount\_tbl 和 tutorials\_tbl，在數據庫：test ，完整列表如下：**

|  |
| --- |
| create table tcount\_tbl(  tutorial\_author varchar(40) NOT NULL,  tutorial\_count INT  );  INSERT INTO tcount\_tbl (tutorial\_author, tutorial\_count ) values ('Yiibai', 2);  INSERT INTO tcount\_tbl (tutorial\_author, tutorial\_count) values ('mahran', 20);  INSERT INTO tcount\_tbl (tutorial\_author, tutorial\_count) values ('Gill', 20);  SELECT a.tutorial\_id, a.tutorial\_author, b.tutorial\_count  FROM tutorials\_tbl a, tcount\_tbl b  WHERE a.tutorial\_author = b.tutorial\_author;  SELECT a.tutorial\_id, a.tutorial\_author, b.tutorial\_count  FROM tutorials\_tbl a  LEFT JOIN tcount\_tbl b ON a.tutorial\_author = b.tutorial\_author; |

**備註：如果使用LEFT JOIN，得到的所有記錄的匹配方式相同，在左邊表中得到的每個記錄不匹配也會有一個額外的記錄。 從而確保(在本例子)，每次作者信息都會列出。**

**使用NULL值**

|  |
| --- |
| INSERT INTO tcount\_tbl (tutorial\_author, tutorial\_count) values ('mahnaz', NULL);  INSERT INTO tcount\_tbl (tutorial\_author, tutorial\_count) values ('Jen', NULL);  SELECT \* FROM tcount\_tbl WHERE tutorial\_count is NULL;  SELECT \* FROM tcount\_tbl WHERE tutorial\_count is not NULL; |

**修改欄位資料**

|  |
| --- |
| create table testalter\_tbl ( i INT, c CHAR(1) );  SHOW COLUMNS FROM testalter\_tbl;  ALTER TABLE testalter\_tbl DROP i; //刪除欄位i  ALTER TABLE testalter\_tbl ADD i INT; //增加i的欄位，型態為數值  ALTER TABLE testalter\_tbl DROP i;  ALTER TABLE testalter\_tbl ADD i INT AFTER c; //增加一欄在C之後  ALTER TABLE testalter\_tbl MODIFY c CHAR(10); //要改變字段 c 從 CHAR(1) 修改為 CHAR(10)  ALTER TABLE testalter\_tbl CHANGE i j BIGINT; //修改i名稱為j 並將型態改為BIGINT  ALTER TABLE testalter\_tbl  MODIFY j BIGINT NOT NULL DEFAULT 100; //更改j欄位並預設為100  SHOW COLUMNS FROM testalter\_tbl;  ALTER TABLE testalter\_tbl RENAME TO alter\_tbl; //更新資料表名稱 |

**備份指令**

|  |
| --- |
| SELECT \* FROM tutorials\_tbl INTO OUTFILE 'C:\tutorials.txt';  mysqldump -u root -p test tutorials\_tbl > dump.txt; // test資料庫內的tutorials\_tbl資料表  mysqldump -u root -p test > database\_dump.txt; //導出test資料庫 |

**匯入指令**

|  |
| --- |
| mysql -u root -p database\_name < dump.txt |

**MySQL相關函數**

|  |  |
| --- | --- |
| Group By子句 | GROUP BY語句以及SQL聚合函數，用於類似SUM提供某些資料表內相同的列來產生分組結果 |
| MySQL IN子句 | 這是一個子句，它可以用來連同任何MySQL查詢語句以指定條件 |
| BETWEEN子句 | 這是一個子句，它可以用來與任何MySQL查詢來指定條件 |
| UNION關鍵字 | 使用UNION操作多個結果集組合成一個結果集 |
| COUNT()函數 | COUNT聚合函數用於計算一個資料表中的行數 |
| MAX() 函數 | MAX聚合函數允許我們選擇某些列的最高(最大)值 |
| MIN()函數 | MIN聚合函數允許我們選擇某些列的最低(最小)值 |
| AVG()函數 | AVG聚合函數是用來對某些表的列求它的平均值 |
| SUM()函數 | SUM聚合函數允許選擇某列的總和 |
| SQRT函數 | 這是用來生成給定數的平方根 |
| RAND()函數 | 使用MySQL命令產生一個隨機數 |
| CONCAT()函數 | 這是用來連接MySQL命令中的任何字符串 |
| DATE 和 Time 時間日期函數 | 日期和時間相關的函數完整列表 |

**Group By子句 and Count函數**

|  |
| --- |
| create table employee\_tabl(  id int(10) not null primary key auto\_increment,  name varchar(32) not null default '',  work\_date date,  daily\_typing\_pages int(10) default 0  );  INSERT INTO employee\_tabl(name, work\_date,daily\_typing\_pages) VALUES('John', '2015-01-24', 150);  INSERT INTO employee\_tabl(name, work\_date,daily\_typing\_pages) VALUES('Ram', '2015-07-27', 220);  INSERT INTO employee\_tabl(name, work\_date,daily\_typing\_pages) VALUES('Jack', '2015-05-06', 170);  INSERT INTO employee\_tabl(name, work\_date,daily\_typing\_pages) VALUES('Jack', '2015-01-24', 100);  INSERT INTO employee\_tabl(name, work\_date,daily\_typing\_pages) VALUES('Jill', '2015-01-24', 220);  INSERT INTO employee\_tabl(name, work\_date,daily\_typing\_pages) VALUES('Zara', '2015-01-26', 300);  INSERT INTO employee\_tabl(name, work\_date,daily\_typing\_pages) VALUES('Zara', '2015-02-24', 350);  SELECT COUNT(\*) FROM employee\_tabl; //查詢所有筆數  SELECT name, COUNT(\*)  FROM employee\_tabl  GROUP BY name; //根據name分組取筆數 |

**Between子句**

|  |
| --- |
| SELECT \* FROM employee\_tabl WHERE  daily\_typing\_pages >= 170 AND daily\_typing\_pages <= 300;  SELECT \* FROM employee\_tabl WHERE daily\_typing\_pages BETWEEN 170 AND 300; |

**In子句**

|  |
| --- |
| SELECT \* FROM employee\_tabl WHERE  daily\_typing\_pages= 250 OR daily\_typing\_pages= 220 OR daily\_typing\_pages= 170;  SELECT \* FROM employee\_tabl WHERE daily\_typing\_pages IN ( 250, 220, 170 ); |

**Max()函數**

|  |
| --- |
| SELECT MAX(daily\_typing\_pages) FROM employee\_tabl; |

**Min()函數**

|  |
| --- |
| SELECT MIN(daily\_typing\_pages) least, MAX(daily\_typing\_pages) max FROM employee\_tabl; |

**Avg()函數**

|  |
| --- |
| SELECT AVG(daily\_typing\_pages) FROM employee\_tabl; |

**Sum()函數**

|  |
| --- |
| SELECT SUM(daily\_typing\_pages) FROM employee\_tabl; |

**Sqrt ()函數**

|  |
| --- |
| SELECT name, SQRT(daily\_typing\_pages) FROM employee\_tabl; |

**Rand()函數：以產生一個在 0 和 1 之間的隨機數：**

|  |
| --- |
| SELECT \* FROM employee\_tabl ORDER BY RAND(); |

**CONCAT()函數**

|  |
| --- |
| SELECT CONCAT("SQL ", "Runoob ", "Gooogle ", "Facebook") AS ConcatenatedString; |

**ADDDATE(d,n)**

|  |
| --- |
| SELECT ADDDATE("2017-06-15", INTERVAL 10 DAY); |

**DATE\_FORMAT(d,f)：日期格式化**

|  |
| --- |
| SELECT DATE\_FORMAT('2011-11-11 11:11:11','%Y-%m-%d %r'); |

**DATEDIFF(d1,d2)：計算兩個日期的間隔天數**

|  |
| --- |
| SELECT DATEDIFF('2001-01-01','2001-02-02'); |

**If 判斷式**

|  |
| --- |
| SELECT IF(1 > 0,'正確','錯誤') ; |

**IFNULL(v1,v2)判斷式**

|  |
| --- |
| SELECT IFNULL(null,'Hello Word'); |