**USER帳號及授權**

**列出所有user帳號**

select user, host from mysql.user

**新增User帳號**

create user username@localhost(不填就是%,也就是user可從任何電腦登入)

identified by '密碼' >>> (可直接跳到下一行,登入時就會直接重設密碼)

password expire >>> (登入即重設密碼,可於上一行先給預設密碼)

**修改user密碼**

alter user username@localhost identified by '新密碼';

**要求user重設密碼**

alter user username@localhost password expire;

**修改user的host**

update mysql.user set host='localhost' where host='%' and user='username';

**刪除User帳號**

drop user username@localhost;

**查詢user權限**

show grants for username;

1. **Global Privileges（全域級授權）**

* 授權/撤銷對**所有資料庫/表格**進行動作
* grant(revoke) create, drop on \*.\* to(from) username;

1. **Database Privileges（資料庫級授權）**

* 授權/撤銷對**某資料庫**進行查詢
* grant(revoke) select on 資料庫名.\* to(from) username;

1. **Table Privileges（表格級授權）**

* 授權/撤銷對**某資料庫內的某表格**進行動作
* grant(revoke) select on 資料庫名.表格名 to(from) username;

grant(revoke) all on 資料庫名.表格名 to(from) username;

* **唯讀 :** 先建立view再授權user只能查詢該view

1. **Column Privileges（欄位級授權）**

* 授權對**指定欄位**進行動作
* grant(revoke) select(欄位1, 欄位2), update(欄位3) on 表格名 to(from) username;

**Table Datatype**

**PK(Primary Key)主鍵**

* 代表一個表格最重要的項目,有唯一性且不可為空值(Not Null)
* 如果想讓系統自動編號(auto\_increment),可在創欄位時勾選”AI”

**FK(Foreign Key)外來鍵**

* 需參照外部表格的欄位
* 被參照的欄位無法隨意更動或刪除,也不能為空值(Not Null)

**數字類型**

* UN(unsigned) : 無負號
* TINYINT(總共幾位數) : 範圍-128~127或0~255
* FLOAT(總共幾位數,小數點有幾位數)
* DOUBLE(總共幾位數,小數點有幾位數) : 雙精確浮點數
* DECIMAL(自訂總共幾位數,小數點有幾位數)

**文字類型**

* CHAR(N) : 固定N個字
* VARCHAR(N) : 字數不確定但不會太長
* TEXT : 極長篇幅
* ENUM('M','FM') : insert的時候只能從中選一個(EX: 'M' OR '1' OR '0'=不填)
* SET('a','b','c','d') : insert的時候可以從中挑選多個組成set(EX: 'a,b,c')
* 不區分大小寫

**時間類型**

* DATE : 年-月-日
* TIME : 時:分:秒
* DATETIME : 年-月-日 時:分:秒
* TIMESTAMP : 年月日時分秒(時區)
* 新增時自動填入系統時間 : 表格最後KEY入”current\_timestamp”
* 修改時自動填入系統時間 : 表格最後KEY入”on update current\_timestamp”

**圖檔類型**

* MEDIUMBLOB

**儲存引擎**

**MyISAM**

* MySQL 5.5之前的預設儲存引擎
* 每個table都有三個檔案

1. .frm : 欄位和表格設定
2. .MYD : data本身
3. .MYI : 索引資訊

* 避免違反檔案大小的限制

**InnoDB**

* MySQL 5.5後的預設儲存引擎
* 提供交易（transaction）、紀錄鎖定（row-level locking）與自動回復（auto-recovery）等功能
* 每個table只有.frm一個檔案，其他存放在共用區
* 避免違反檔案數量的限制

**MEMORY**

* 資料儲存在記憶體中，只要MySQL伺服器關閉就會不見 ，剩下schema

**查詢資料庫資訊**

* 查詢既定資訊 : SHOW character set/collation/engine
* 看有哪些 : SHOW databases/tables (status)/trigger FROM 資料庫名 LIKE '...'
* 看怎麼被建立 : SHOW CREATE database/table/function/procedure/view/trigger 名稱
* 看詳細資訊 : SELECT \* FROM information\_schema.schemata/tables/columns/views/routines/triggers/engines
* 看欄位資訊
* DESC(DESCRIBE) 表格名 欄位 / ’%...’
* SHOW columns/index FROM 表格名 LIKE '...'
* 免登入
* $ mysqlshow -uXXX -pXXX (= SHOW DATABASES)
* $ mysqlshow -uXXX -pXXX db名 (= SHOW TABLES FROM db名)
* $ mysqlshow -uXXX -pXXX db名 表格名 (= SHOW COLUMNS FROM db名.表格名)
* ERROR LOG
* SHOW errors/warnings;
* SELECT @@warning\_count/@@error\_count;

**資料庫**

**建立資料庫**

create database (schema)資料庫名稱 if not exists;

create database (schema)資料庫名稱 CHARACTER SET utf8 COLLATE utf8\_unicode\_ci; => 要指定字元集時

**更改資料庫字元集**

ALTER DATABASE DB名 CHARACTER SET utf8 COLLATE utf8\_unicode\_ci

**刪除資料庫**

drop database (schema)資料庫名稱if exists;

**表格**

**建立表格**

* use 資料庫名稱 ;

create table表格名稱 (

欄位一名 datatype unsigned zerofill(左邊用0填滿) primary key not null auto\_increment ,

欄位二名 datatype not null default 'XXX'(如果沒填就自動填入XXX) ,

欄位三名 timestamp default current\_timestamp(新增時自動填入系統時間) on update current\_timestamp(修改時自動填入系統時間),

constraint fk\_外來鍵名稱 \*\*(如果有的話),

foreign key (外來鍵欄位名稱) references 參照表格名 (參照欄位名稱),

(UNIQUE/FULLTEXT/SPATIAL) INDEX 索引名(可不取) USING (索引欄位(索引長度) ASC)

) AUTO\_INCREMENT=1000(從1000開始編號) CHARSET=latin1 ENGINE= MyISAM

※ AUTO\_INCREMENT = SERIAL

※ 也可CREATE TEMPORARY TABLE : 用戶端專屬表格，離線後自動刪除

* 將select的資料建立到一個新的表格

create table 新表格名 (可新增或重新定義COLUMN)

select....

(primary key、engine那些設定都不會保留!!!!!!!)

**刪除表格**

drop table表格名;

**修改表格**

* + **改表格名**
* alter table 表格名 rename to 新表格名;
* rename table 表格名 to 新表格名;
  + **新增欄位**

alter table表格名 add column 欄位名 datatype;

* + **欄位更名**

alter table 表格名 rename column 欄位名 to 新欄位名;

* + **修改欄位預設值**

alter table表格名 alter column欄位名 set default '新預設值';

* + **刪除欄位預設值**

alter table表格名 alter column欄位名 drop default;

* + **修改欄位類型**

alter table表格名 modify欄位名 新datatype;

* + **刪除欄位**

alter table表格名 drop column 欄位名;

* + **建立fk**

alter table 表格名 add constraint fk\_外來鍵名稱

foreign key (外來鍵欄位名稱) references 參照表格名 (參照欄位名稱) ;

* + **移除fk**

alter table 表格名 drop foreign key fk\_外來鍵名稱;

* + **欄位改為不可/可為空值**

alter table表格名 modify 欄位名 datatype not null;

(視覺化刪除資料只會將資料設為空字串而不會設為空值，需對欄位右鍵選set field to null)

**欄位**

**索引(具指標性的意義ex : PK,方便日後查詢)**

1. 分為 :
   * 1. **主索引 :** 主鍵/外來鍵(不可為空值)
     2. **唯一索引 :** 有獨特性**(unique)**,但可為空值
     3. **非唯一索引**
     4. **複合型索引 :** 顯示時有優先順序,但移除時會一起
2. 建立索引
   * + 1. 創建表格時
       2. 複合型索引

create table ...

unique(索引欄1,索引欄2)

* + - 1. create index索引名 on 表格名 (索引欄1,索引欄2…)

※ 這裡一定要取索引名!!

* + - 1. alter table表格名add index 索引名 (索引欄1,索引欄2…)

1. 移除索引

drop index 索引名(or索引欄名) on 表格名

1. 評估執行效能(查詢速度)

explain select ...

* rows數字小 : 代表查詢速度快(有索引幫忙)
* TYPE=ALL : 代表查詢速度慢(沒用索引,全部都查)
* Possible Keys=Null : 代表索引沒發揮功效

**新增資料**

* Insert IGNORE (into) 表格名 (欄位1, 欄位2) values

('第1個物件值1', '第1個物件值2') ,

('第2個物件值1', defalut)

ON DUPLICATE KEY UPDATE …..(如果主鍵重複的話就….);

※IGNORE代表如果主鍵會重複就先忽略這筆資料,並繼續往下執行

* 將select的結果插入到另一個表格

insert into 表格名

select ....

* INSERT時不知道下一個ID是多少
* 如果是AUTO\_INCREMENT : 用0/NULL/不打
* SELECT **LAST\_INSERT\_ID()** 查詢
* select auto\_increment from information\_schema.tables where table\_schema = '資料庫名稱' and table\_name = '表格名稱' ;
* show table status from 資料庫名稱 where name like '表格名稱';
* 如果database年久失修,欲重新編號 :
* set @@SESSION.information\_schema\_stats\_expiry = 0;

**刪除資料**

* delete IGNORE from 表格名 where 欄位 = '值';
* 清空整個TABLE
* delete from 表格名
* truncate table 表格名

**修改資料**

* update IGNORE表格名 set 欄位1 = '新值' where 欄位2 = '值';
* replace (into) 表格名 (欄位1, 欄位2) values ('第1個物件值1', '第1個物件值2');

**查詢資料**

* select \* from 資料庫名.表格名;
* select欄位1, 欄位2 \* 0.9 as 新名字, 欄位3 from表格名
* select還可用在**函式**
* Where用法
  + 1. where欄位 is null
    2. where not欄位 is null
    3. where 欄位 between 值 and 值
    4. where欄位 in (A, B, C) : 有A或B或C都會列出,不可接’=’等符號
    5. where欄位 = any/some (A, B, C) : 有A或B或C都會列出,可接符號
    6. where欄位 = all (A, B, C) : ABC都符合才列出
    7. where 欄位 like '\_XXX%' : 字串符合這種格式都可以列

(\_代表一個任意字 %代表可以是0~無限多個字)

* + 1. && = AND

|| = OR

<> : 不等於

<=> : 等於,只能用在NULL(=is null)

* **子查詢**
  + 1. select…. from (select…from..) as 子查詢名稱
    2. select…. where … (select…from..)
    3. select….from 表格 b where exists (select \* from… where欄位名 = b.欄位名 and 其他條件)

>>> exists運用pk鍵有索引的特性，跟in比起來比對速度更快。

* **View(把select的內容直接存成view,會顯示在左欄,簡潔且可以直接當表格呼叫)**
* 建立：CREATE OR REPLACE ALGORITHM = MERGE/TEMPTABLE view db名.view名稱 (欄位名稱,….) AS

SELECT ...

WITH (LOCAL/CASCADED) CHECK OPTION ;

* 檢查View的正確性 : CHECK TABLE VIEW名
* 查詢：SELECT \* FROM view名稱
* 更改：ALTER VIEW VIEW名 AS SELECT ...
* 修改檢查
* LOCAL : UPDATE時違反當下VIEW的初衷會報錯
* CASCADED : UPDATE時違反FROM的那個VIEW的初衷會報錯
* ALGORITHM
* MERGE(預設)
* TEMPTABLE(暫時) : 跟source table之間已無關聯
* 限制
* 不可以用計算欄位、函式、子查詢、TEMPORARY表格、Triggers、自定義變數、SP參數、多結合查詢
* 「ALGORITHM」設定為「TEMPTABLE」無法JOIN
* 讓查出的值不重複

SELECT DISTINCT 欄位 FROM 表格名;

* 查出來的結果升冪(asc)或降冪(desc)呈現

SELECT \* FROM 表格名 ORDER BY欄位1(或直接寫1) asc或不寫, 欄位2 desc,

(結果會依欄位1升冪呈現,再來才是欄位2) , 欄位3 rand() (隨機排);

* 讓null的欄位排在最後

SELECT isnull(null欄位) FROM 表格名 ORDER BY isnull(null欄位);

(isnull會為1，其餘為0)

**UNION語法(垂直合併兩個表格，欄位數要一樣)**

SELECT 欄位1 AS 新欄位名FROM 表格1

UNION ALL (不加ALL重複的資料就不會顯示兩次)

SELECT 欄位2 AS 新欄位名 FROM 表格2

合併後以第一個欄位名稱為主

**JOIN語法(水平合併兩個表格同一個欄位資料的其他欄位)**

**INNER JOIN**

* + **2個表格在同個欄位名稱都有這筆資料才會呈現,其他資料一律不顯示**
  + SELECT表格1縮寫(EX:B).共同的欄位名, 表格縮寫.其他想列的欄位

FROM 表格1 表格1縮寫

JOIN 表格2 表格2縮寫 ON 表格1縮寫.共同的欄位名 = 表格2縮寫.共同的欄位名

(如果共同欄位原本名字一樣, “ON後面那段”可以換成”**USING (共同欄位名)**”)

**LEFT (OUTER) JOIN**

* + **會顯示表格1的所有值，表格2無對應者會顯示null**
  + 語法同INNER JOIN ,只是”join”改成**”left join”**

**RIGHT (OUTER) JOIN**

* + **會顯示表格2的所有值，表格1無對應者會顯示null**
  + 語法同INNER JOIN ,只是”join”改成**”right join”**

**FULL OUTER JOIN**

* + **2個表格全顯示，無對應者會顯示null**
  + MySQL不支援，可將LEFT JOIN與RIGHT JOIN用UNION結合以達目的

LEFT JOIN 語法

UNION

RIGHT JOIN 語法

**SELF JOIN**

* + **同表格的某欄位參照另一個欄位**
  + 語法善用上面方式,注意縮寫即可

**GROUP BY(用來分組呈現資料)**

* + SELECT列出來的欄位，除了聚集函數的欄位外，其餘都要列在GROUP BY後
  + SELECT 分組依據的欄位1, 分組依據的欄位2, 聚集函數(要統計的欄位) 分組完的統計欄位名 FROM 表格名

(JOIN….)

GROUP BY 分組依據的欄位1, 分組依據的欄位2 desc ,分組依據的欄位3(可以當分組依據但不呈現)

(HAVING….) >>>跟WHERE用法一樣,只是要用在GROUP完的結果

(ORDER BY….)

(WITH ROLLUP) >>>最後顯示總計

**LIMIT(只顯示前幾筆資料)**

* + 顯示前10筆 : SELECT …. ORDER BY…. LIMIT 10(放最後)
  + 從第四列開始顯示10筆 : SELECT …. ORDER BY…. LIMIT 4,10
  + 顯示第二頁的10筆 : SELECT …. LIMIT 10 OFFSET 10

**函式(需用select開頭)**

**字串函式**

* + **char\_length(str)**: 回傳str的長度
  + **length(str):** 回傳str長度(以byte計算)

(中文字一個算3 bt,英文單字一個算1 bt)

* + **lower(str):** 將str轉成小寫字
  + **repeat(str, count):** 將str重複count次
  + **reverse(str):** 將str內容反轉
  + **concat(str1, str2, …):** 將str1, str2...等各個字串串接起來
  + **concat\_ws(“分隔符號” ,str1, str2, …):** 使用分隔符號串接各個字串
  + **group\_concat(欄位ORDER BY 欄位 SEPARATOR '|') :** 把每一列用','隔開(或用**'|'**),串成一句
  + **field(str, str1, str2, str3, …):** 在str1, str2, ...內搜尋是否存在str字串，有則回傳所在位置，無則回傳0
  + **find\_in\_set(str, strlist):** 在字串列表 strlist 內搜尋是否存在str字串
  + **instr(str, substr)= locate(substr,str) :** 回傳**substr**在**str**第幾個位置出現
  + **insert(str, pos, len, newstr):** 將str從第pos位開始len長度的字替換成newstr
  + **replace(str, from\_str, to\_str):** 將str文字中所有from\_str都替換成to\_str (區別大小寫)
  + **left(str, len):** 回傳str從左開始len個字
  + **substring(str,pos,len):** 回傳str從第pos位開始的len個字元，沒有len代表從pos到最後
  + **substring\_index(str, “分隔符號”, count):** 回傳第count個分隔符號前的字串。Count若為負代表右算到左，回傳分隔符號後面的字串。
  + **ltrim(str) :** 移除左邊的空白
  + **trim(both / leading / trailing “x” from str):** 去除左右端的空白字元或“x”
  + **lpad(str1, 長度, str2) :** 如果str1的長度小於**長度**，就在str1左邊用str2補滿

**數學函式**

* + **abs(x):** 絕對值
  + **pow(x, y):** x的y次方
  + **sqrt(x):**平方
  + **ceil(x):** 回傳大於x的最小整數
  + **floor(x):** 回傳小於x的最大整數
  + **round(x, d)= format(x, d) :** -- 將x四捨五入到小數第d位，省略d則代表到整數位
  + **truncate(x, d):** 保留x到小數第d位，其餘無條件捨去
  + **pi():** 回傳圓周率
  + **radians(x):** 將角度轉成弧度
  + **degrees(x):** 將弧度轉成角度
  + **rand([n]):** 回傳一個介於0(含)~1.0(不含)的亂數。如果加上n則可重複產生相同亂數

**聚集函式**

* + **sum()：**加總
  + **avg()：**平均
  + **count()：**計算數量
  + **max()：**最大值
  + **min()：**最小值
  + **rank() :** 排名

rank() over(order by欄位)

**日期時間函式**

日期時間單位參看：<https://dev.mysql.com/doc/refman/8.0/en/expressions.html#temporal-intervals>

格式化符號參看：

https://dev.mysql.com/doc/refman/8.0/en/date-and-time-functions.html#function\_date-format

* + **date\_format(date, format):** 將date使用format來格式化
  + **str\_to\_date(str, format):** 將字串str解析回日期時間
  + **extract(unit from date):** 回傳date的unit部分
  + **date(date):** 回傳日期部分
  + **time(time):** 回傳時間部分
  + **month(date):** 回傳月
  + **quarter(date) :** 回傳季度
  + **monthname(date) :** 回傳月份名
  + **dayname(date) :** 回傳星期名
  + **extract(單位 from 時間) :** 回傳時間的單位
  + **dayofmonth(date):** 回傳date是一個月的第幾日
  + **dayofweek(date):** 回傳date的weekday(1 = Sunday, 2 = Monday…)
  + **dayofyear(date):** 回傳date是一年的第幾日
  + **week(date):** 回傳date是一年的第幾週
  + **last\_day(date):** 回傳date月份中的最後一天
  + **now():** 回傳sql語法準備執行時的時間，同句呼叫多次也是回傳相同時間
  + **curdate() :** 今天"日期"
  + **curtime() :** 現在"時間"
  + **sysdate():** 回傳執行當下的時間，同句呼叫多次會是不同時間
  + **adddate(date, interval value unit):** 將date加上多少value的什麼unit
  + **datediff(date1, date2):** 回傳2個日期相差的天數(date1 - date2)
  + **timediff(date1, date2):** 回傳2個日期相差的時間(date1 - date2)
  + **timestampdiff(unit, date1, date2):** 回傳2個日期相差的時間(date2 - date1)，差值以unit計算

**資料類型轉換函式**

* + **convert(value, datatype):** 將value轉成指定datatype

(binary, char, date, datetime, time, signed, unsigned)

**流程控制函式**

* + **if(條件, 回饋1, 回饋2):** 條件為true回傳回饋1，否則回傳回饋2
  + **ifnull(資料1, 資料2):** 如果資料是null則回傳資料2，否則回傳資料1
  + **CASE 欄位**

**WHEN 條件 THEN 回傳值**

**ELSE 回傳值**

**END**

**系統資訊函式**

* + **charset(str):** 回傳str的字元集
  + **database():** 回傳使用中的資料庫
  + **user():** 回傳當下user帳號與主機名稱
  + **version():** 回傳資料庫版本

**進階**

**SQL\_MODE(嚴謹模式)**

* 設定

SET (SESSION/GLOBAL) sql\_mode = '...'

* ' ' : **Non-Strict模式**(改資料比較容錯)
* 'STRICT\_ALL\_TABLES' : **Strict模式**
* 'ALLOW\_INVALID\_DATES'
* 'NO\_ZERO\_DATE' (0000-00-00)
* 'NO\_ZERO\_IN\_DATE'
* 'ERROR\_FOR\_DIVISION\_BY\_ZERO' (正常會產生NULL)
* 查詢
* SHOW VARIABLES LIKE 'sql\_mode'
* SELECT @@sql\_mode

**AUTO-COMMIT&ROLLBACK&SAVEPOINT**

* Workbench > Query > Auto-Commit Transactions 預設為開啟

下完任何指令後都無法還原

* 查詢AUTO-COMMIT狀態

SELECT @@AUTOCOMMIT;

* 修改AUTO-COMMIT狀態（0 - 關閉；1 - 開啟）

SET AUTOCOMMIT = 0;

* AUTO-COMMIT關閉狀態下,打完某個段落後可自訂讓指令回復(ROLLBACK)

…..

ROLLBACK;

* AUTO-COMMIT關閉狀態下,打完某個段落後可自訂讓指令無法還原(COMMIT)

…..

COMMIT;

* AUTO-COMMIT關閉狀態下,設下儲存點即可回溯到某儲存點

…..

SAVEPOINT POINT01;

…..

SAVEPOINT POINT02;

ROLLBACK TO POINT01;