

關聯式資料庫管理系統 結構化查詢語言(SQL) 使用MySQL

資料查詢(SELECT)

MySQL資料型態(Data Types)

內建函數(Built-in Functions)

MySQL的資料型態

▶ 基本資料型態

◦ 文數字資料 (Character Types)

- CHAR and VARCHAR
- BLOB and TEXT

◦ 數值資料 (Numeric Types)

- 整數
 - INT
- 實數
 - DECIMAL[(p[, s])], NUMERIC[(p[, s])]
 - Float

▶ 導出型資料型態

◦ 日期/時間資料 (Date and Time Types)

- Date
- DateTime

欄位資料型態(Column Data Types)

► MySQL 的資料型態

數值資料	日期時間資料	文數字資料
TINYINT	DATE	CHAR(M)
SMALLINT	DATETIME	VARCHAR(M)
MEDIUMINT	TIMESTAMP	TINYBLOB, TINYTEXT
INT, INTEGER	TIME	BLOB, TEXT
BIGINT	YEAR	MEDIUMBLOB, MEDIUMTEXT
FLOAT(p)		LOBLOB, LONGTEXT
FLOAT		
DOUBLE [PRECISION], item REAL		
DECIMAL(M,D), NUMERIC(M,D)		

文數字資料 (Character Types)

類型	儲存方式	儲存大小	範圍
CHAR(N)	固定size 不足部份以空白填入 顯示時將空白截斷	字元數	0~255
VARCHAR(N)	變動size(空白移除) 空白部份自動截斷 顯示時將空白截斷	字元數+ 1byte	0~255
TINYBLOB BLOB MEDIUMBLOB LONGBLOB	Binary 不分大小寫	L+1 bytes L+2 bytes L+3 bytes L+4 bytes	L < 2 ⁸ L < 2 ¹⁶ L < 2 ¹⁶ L < 2 ¹⁶
TINYTEXT TEXT MEDIUMTEXT LONGTEXT	Character 分大小寫	L+1 bytes L+2 bytes L+3 bytes L+4 bytes	L < 2 ⁸ L < 2 ¹⁶ L < 2 ¹⁶ L < 2 ¹⁶

數值資料(Numeric Types)

- ▶ 整數(Integer)
 - TINYINT, SMALLINT, MEDIUMINT
 - INT/INTEGER, BIGINT
- ▶ 整數資料-資料長度與有效範圍

Type	Bytes	Minimum Value (Signed)	Maximum Value (Signed)
TINYINT	1	-128	127
SMALLINT	2	-32768	32767
MEDIUMINT	3	-8388608	8388607
INT	4	-2147483648	2147483647
BIGINT	8	-9223372036854775808	9223372036854775807

數值資料(Numeric Types)

▶ 實數

- **DECIMAL(p, s), NUMERIC(p, s)**
- FLOAT, DOUBLE PRECISION(p, s), DOUBLE

▶ 精準實數(DECIMAL & NUMERIC) – 商業應用

- DECIMAL與NUMERIC型態是相同的作法
- 儲存時以字串儲存, 而不是二元浮點方式
- 最大的範圍同DOUBLE
- 可以指定精確度及小數位數

DECIMAL[(p[, s])]

NUMERIC[(p[, s])]

- 總長度(p)=整數+小數(s) 預設值為10
- 範例：DECIMAL(5, 2) 範圍999.99 ~ -999.99

數值資料(Numeric Types)

▶ FLOAT & DOUBLE – 科學計算

- 使用數字逼近法

FLOAT[(p)]

DOUBLE[(p)]

- p 指定精確長度, 決定儲存的大小
- 資料長度與有效範圍
 - 單精確數 4bytes p=0~23
 - 雙精確數 8bytes p=24~53

日期/時間資料(Date and Time Types)

▶ 日期與時間資料

- DATE
- DATETIME
- TIMESTAMP
- TIME
- YEAR

型 態	資 訊	格 式	範 圍	儲存大小
DATETIME	DATE & TIME	'YYYY-MM-DD HH:MM:SS'	'1000-01-01 00:00:00' 至 '9999-12-31 23:59:59'	8 bytes
TIMESTAMP	DATE & TIME	'YYYY-MM-DD HH:MM:SS'	'1000-01-01 00:00:00' 至 '9999-12-31 23:59:59'	4 bytes
DATE	DATE	'YYYY-MM-DD'	'1000-01-01' ~ '9999-12-31'	3 bytes
TIME	TIME	'HH:MM:SS'	'-838:59:59' ~ '838:59:59'	3 bytes
YEAR	YEAR	'YYYY'	1901~2155	1 byte

日期/時間資料(Date and Time Types)

▶ DateTime資料以字串型態輸入格式

格式	範例	日期時間
'YYYY-MM-DD HH:MM:SS'	'1998-12-31 11:30:45'	1998-12-31 11:30:45
'YY-MM-DD HH:MM:SS'	'98-12-31 11:30:45'	1998-12-31 11:30:45
'YYYY-MM-DD'	'1998-12-31'	1998-12-31
'YY-MM-DD'	'98-12-31'	1998-12-31
'YYYYMMDDHHMMSS'	'19970523091528'	1997-05-23 09:15:28
'YYMMDDHHMMSS'	'970523091528'	1997-05-23 09:15:28
'YYYYMMDD'	'19981231'	1998-12-31
'YYMMDD'	'981231'	1998-12-31
YYYYMMDDHHMMSS	19830905132800	1983-09-05 13:28:00
YYMMDDHHMMSS	830905132800	1983-09-05 13:28:00
YYYYMMDD	19830905	1983-09-05
YYMMDD	830905	1983-09-05

DateTime資料以字串型態輸入

▶ 'YYMMDD'

```
mysql> SELECT empno, ename, job, hiredate, sal  
-> FROM emp  
-> WHERE hiredate='811203';
```

empno	ename	job	hiredate	sal
7900	JAMES	CLERK	1981-12-03 00:00:00	950.00
7902	FORD	ANALYST	1981-12-03 00:00:00	3000.00

▶ 'YYYYMMDD'

```
mysql> SELECT empno, ename, job, hiredate, sal  
-> FROM emp  
-> WHERE hiredate='19811203';
```

empno	ename	job	hiredate	sal
7900	JAMES	CLERK	1981-12-03 00:00:00	950.00
7902	FORD	ANALYST	1981-12-03 00:00:00	3000.00

DateTime資料以字串型態輸入

- ▶ 你可將-或:符號改成其他任何符號

字串格式	日期/時間
'98-12-31 11:30:45'	1998-12-31 11:30:45
'98.12.31 11+30+45'	1998-12-31 11:30:45
'98/12/31 11*30*45'	1998-12-31 11:30:45
'98@12@31 11^30^45'	1998-12-31 11:30:45

- ▶ 所有不合法的格式, 格式均以0表示

資料型態	日期/時間
DATE	0000-00-00
DATETIME	0000-00-00 00:00:00
TIMESTAMP	0000000000000000

DateTime資料以字串型態輸入

▶ 'YY-MM-DD'

```
mysql> SELECT empno, ename, job, hiredate, sal  
-> FROM emp  
-> WHERE hiredate='81-12-03';
```

empno	ename	job	hiredate	sal
7900	JAMES	CLERK	1981-12-03 00:00:00	950.00
7902	FORD	ANALYST	1981-12-03 00:00:00	3000.00

▶ 'YY#MM#DD'

```
mysql> SELECT empno, ename, job, hiredate, sal  
-> FROM emp  
-> WHERE hiredate='81#12#03';
```

empno	ename	job	hiredate	sal
7900	JAMES	CLERK	1981-12-03 00:00:00	950.00
7902	FORD	ANALYST	1981-12-03 00:00:00	3000.00

DateTime資料以字串型態輸入

- ▶ 可用一位數表示月, 日, 時, 分, 秒
 - '1979-6-9' = '1979-06-09'
 - '1979-10-30 1:2:3' = '1979-10-30 01:02:03'
- ▶ 若年為2位數時
 - 00-69 轉成 2000~2069
 - 70-99 轉成 1970~1999
 - 可用6, 8, 12, 14等長度自動判斷

長度	判斷格式	判斷長度(年)
6	YYMMDD	2
8	YYYYMMDD	4
12	YYMMDDHHMMSS	2
14	YYYYMMDDHHMMSS	4

- ▶ 型態互換注意事項
 - DATE 轉 DATETIME, TIMESTAMP 時間單位為 '00:00:00'
 - DATETIME, TIMESTAMP 轉 DATE 截斷時間單位

TIME 資料型態

▶ 顯示格式及範圍

- 'HH:MM:SS'
- '-838:59:59' ~ '838:59:59'

▶ 輸入字串格式

- 'D HH:MM:SS.fraction'
- 'D HH:MM:SS'
- 'D HH:MM'
- 'D HH'
- 'HH:MM:SS.fraction' 或 'HHMMSS.fraction'
- 'HH:MM:SS' 或 'HHMMSS'
- 'HH:MM' 或 'HHMM'
- 'SS'

▶ 注意：MySQL 不會儲存 fraction 部份

TIME 資料型態

► 輸入字串格式轉換原則

D : 0 ~ 34 days (1 D = 24 hours)

格式	範例	時間
'D HH:MM:SS.fraction'	'1 06:10:10.123000'	30:10:10.123000
'D HH:MM:SS'	'1 06:10:10'	30:10:10
'D HH:MM'	'1 06:10'	30:10:00
'D HH'	'1 06'	30:00:00
'HH:MM:SS.fraction'	'06:10:10.123000'	06:10:10.123000
'HHMMSS.fraction'	'061010.123000'	06:10:10.123000
'HH:MM:SS'	'06:10:10'	06:10:10
'HHMMSS'	'061010'	06:10:10
'HH:MM'	'06:10'	06:10:00
'MMSS'	'0610'	00:06:10 (注意)
'SS'	'10'	00:00:10 (注意)

所有無法自動轉換的字串, 皆轉成00:00:00

YEAR資料型態

▶ 顯示格式及範圍

- 'YYYY'
- 1901~2155

▶ 自動轉換原則

格式	範例	時間
'YYYY'	'1901'~'2155' '0000'	1901~2155 0000
'YY'	'00'~'69' '70'~'99'	2000~2069 1970~1999
YYYY	1901~2155 1700	1901~2155 0000
YY	1~69 70~99 0	2001~2069 1970~1999 0000

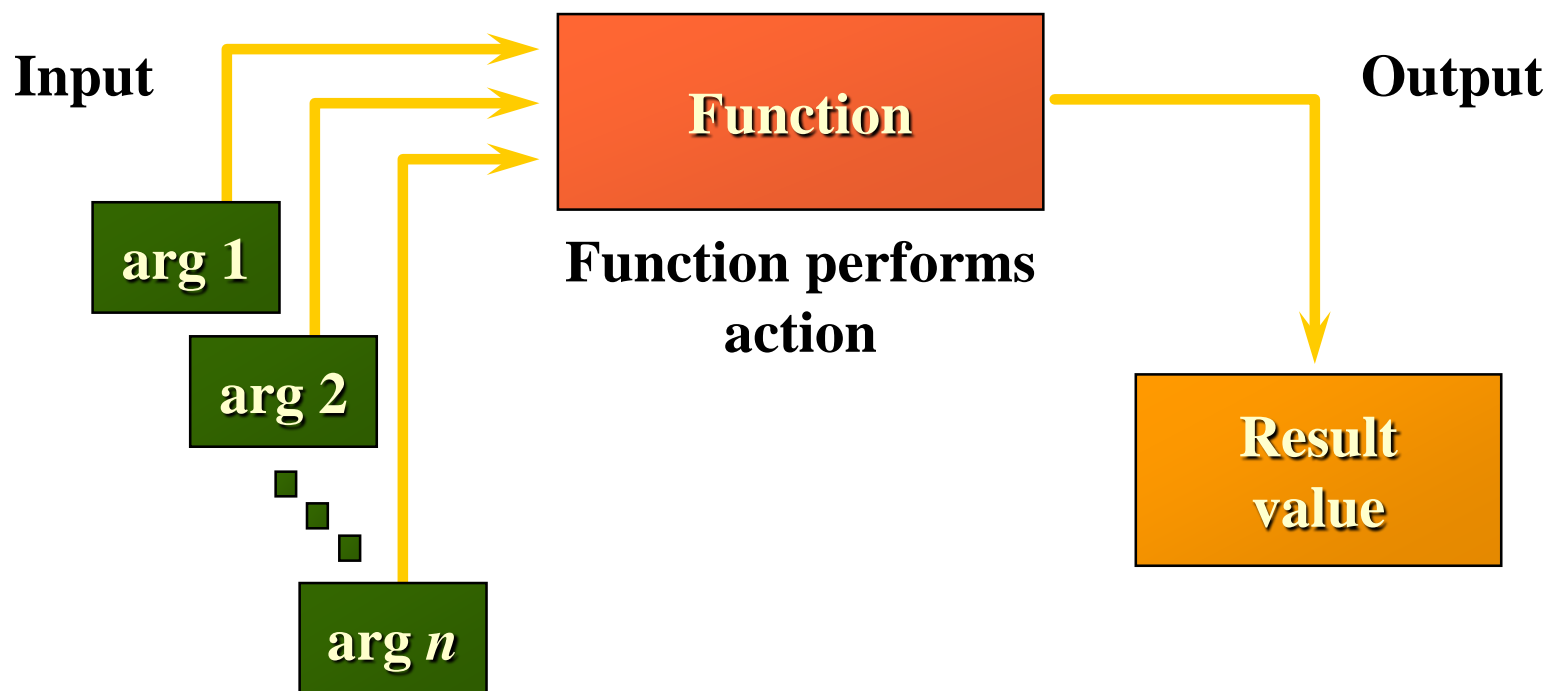
對於無法自動轉換的字串，皆轉成0000

SQL 函數 (SQL Functions)

- ▶ SQL 函數是強化 SQL 命令功能的副程式
 - 執行計算(Perform calculations on data)
 - 修改個別資料項(Modify individual data items)
 - 格式化輸出結果(Format dates and numbers for display)
 - 資料型態轉換(Convert column data types)
 - 群組資料的彙總(Manipulate output for groups of rows)

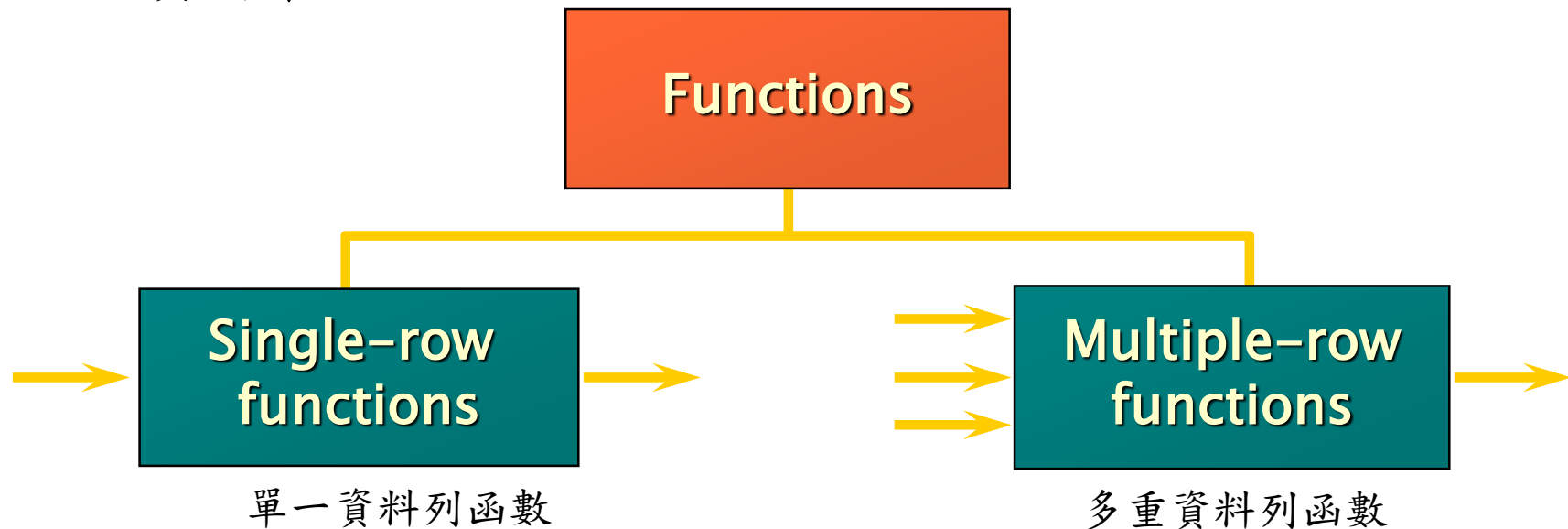
函數(Functions)

- ▶ 強化資料處理與運算的能力
- ▶ 可接受使用者輸入參數並回傳結果



SQL 函數的類別

- ▶ 單一資料列函數(Single-Row Functions)
 - 每筆記錄(row)執行一次, 傳回一個結果
- ▶ 多重資料列函數(Multiple-Row Functions)
 - 多筆記錄(rows)執行一次, 傳回一個結果
 - 資料彙總



單一資料列函數(Single-Row Functions)

- ▶ 一般用來做單一資料項的運算
- ▶ 接受一個或多個參數
- ▶ 每筆記錄(row)執行一次, 傳回一個結果

```
function_name (column|expression, [arg1, arg2,...])
```

- ▶ 參數種類:
 - 欄位名稱 (Column name)
 - 運算式(Expression)
 - 常值(User-supplied constant)
 - 變數值(Variable value)
 - 函數(Function)

單一資料列函數(Single-Row Functions)

- ▶ 字串函數(String functions)
 - 處理字串資料:接受字串資料當參數，執行後傳回字串或數值資料
- ▶ 數值函數(Numeric functions)
 - 處理數值資料:接受數值資料當參數，執行後傳回數值資料
- ▶ 日期/時間函數(Date and Time functions)
 - 處理日期/時間資料
- ▶ 資料型態轉換函數(Conversion functions)
 - 做資料型態轉換
- ▶ 通用函數(General functions)
 - 流程控制與獲取系統資訊

字串函數(String functions)

函數	功能
LENGTH(str)	字串長度
CHAR_LENGTH(str)	字元個數
LCASE(str) LOWER(str)	將字串轉成小寫字母
UCASE(str) UPPER(str)	將字串轉成大寫字母
ASCII(int)	將 int 轉成字元
CONCAT(str1, str2, ...)	字串連結
FIELD(str, str1, str2, str3, ...)	傳回 str 在 str list 的位置
INSERT(str, pos, len, newstr)	在指定區間插入一字串
LEFT(str, len)	由左至右傳回指定字元數
RIGHT(str, len)	由右至左傳回指定字元數
REVERSE(str)	將字串反轉
LPAD(str, len, padstr)	將字串不足指定長度以指定字元向左補齊
RPAD(str, len, padstr)	將字串不足指定長度以指定字元向右補齊

字串函數(String functions)

函數	功能
<code>SUBSTRING(str, pos)</code> <code>SUBSTRING(str FROM pos)</code> <code>SUBSTRING(str, pos, len)</code> <code>SUBSTRING(str FROM pos FOR len)</code>	截取字串部份字元
<code>REPEAT(str, count)</code>	將指定字串重複指定次數
<code>SPACE(N)</code>	將空白字元重複指定次數
<code>INSTR(str, substr)</code> <code>LOCATE(substr, str[, pos])</code>	傳回指定字元在字串中出現的位置
<code>REPLACE(str, from_str, to_str)</code>	將新字串替代舊字串
<code>LTRIM(str)</code>	去除字串左邊空白字元
<code>RTRIM(str)</code>	去除字串右邊空白字元
<code>TRIM([[BOTH LEADING TRAILING] [remstr] FROM] str)</code>	去除字串二端空白字元
<code>STRCMP(expr1, expr2)</code>	<code>expr1=expr2</code> return 0 <code>expr1>expr2</code> return 1 <code>expr1<expr2</code> return -1

字串函數(String functions)

Function	Result
CONCAT(' Good ', ' String ')	GoodString
SUBSTRING(' String ',1,3)	Str
LENGTH(' String ')	6
INSTR(' String ', 'r')	3
LPAD(sal,10,'*')	*****5000
TRIM('S' FROM 'SSMITH')	MITH

```
mysql> select ename, sal, concat(job,' in department ',deptno) Job_dep  
-> from emp  
-> where substr(ename,1,1)='M';
```

```
+-----+-----+-----+  
| ename  | sal      | Job_dep                               |  
+-----+-----+-----+  
| MARTIN | 1250.00  | SALESMAN in department 30          |  
| MILLER | 1300.00  | CLERK in department 10             |  
+-----+-----+-----+
```


字串函數(String functions)

▶ LPAD, RPAD

```
mysql> select ename, sal, lpad(sal,10,'#'), rpad(sal,10,'#')
-> from emp;
```

ename	sal	lpad(sal,10,'#')	rpadd(sal,10,'#')
SMITH	800.00	####800.00	800.00####
ALLEN	1600.00	###1600.00	1600.00###
WARD	1250.00	###1250.00	1250.00###
JONES	2975.00	###2975.00	2975.00###
MARTIN	1250.00	###1250.00	1250.00###
BLAKE	2850.00	###2850.00	2850.00###
CLARK	2450.00	###2450.00	2450.00###
SCOTT	3000.00	###3000.00	3000.00###
KING	5000.00	###5000.00	5000.00###
TURNER	1500.00	###1500.00	1500.00###
ADAMS	1100.00	###1100.00	1100.00###
JAMES	950.00	####950.00	950.00####
FORD	3000.00	###3000.00	3000.00###
MILLER	1300.00	###1300.00	1300.00###

數值函數(Numeric functions)

函數	功能
ROUND(X, D)	四捨五入
TRUNCATE(X, D)	無條件捨去
MOD(N, M)	求餘數
CEIL(X)	傳回不小於 X 的最小整數值
FLOOR(X)	傳回不大於 X 的最大整數值
POWER(X, Y)	X的Y次方
SQRT(X)	平方根
ABS(X)	絕對值
SIGN(X)	正負數
RAND() RAND(N)	亂數
PI()	圓周率
RADIANS(X)	度數轉徑值
DEGREES(X)	徑值轉度數

數值函數(Numeric functions)

```
mysql> select ename, sal, repeat('*',round(sal/100,0))
-> from emp
-> where deptno=10;
```

ename	sal	repeat('*',round(sal/100,0))
CLARK	2450.00	*****
KING	5000.00	*****
MILLER	1300.00	*****

```
mysql> select ename, sal, sal+floor(rand()*1000)
-> from emp
-> where deptno=20;
```

ename	sal	sal+floor(rand()*1000)
SMITH	800.00	1775.00
JONES	2975.00	3716.00
SCOTT	3000.00	3780.00
ADAMS	1100.00	1780.00
FORD	3000.00	3060.00

日期/時間函數(Date and Time functions)

函數	功能
CURDATE()	現在日期
CURTIME()	現在時間
CURRENT_TIMESTAMP()	現在日期與時間
NOW()	現在日期與時間
UTC_DATE()	格林威治日期
UTC_TIME()	格林威治時間
UTC_TIMESTAMP()	格林威治日期與時
YEAR(date)	傳回指定日期的年份
MONTH(date)	傳回指定日期的月份
DAY(date)	傳回指定日期的日份
HOUR(time)	傳回指定日期的時
MINUTE(time)	傳回指定日期的分
SECOND(time)	傳回指定日期的秒

日期/時間函數(Date and Time functions)

函數	功能
TIME(expr)	傳回時間單位
MICROSECOND(expr)	傳回指定日期的微秒
EXTRACT(type FROM date)	傳回指定的日期單位
DAYNAME(date)	日期名稱
MONTHNAME(date)	月份名稱
DAYOFWEEK(date)	一週中的第幾天
DAYOFMONTH(date)	月份中的第幾天
DAYOFYEAR(date)	一年中的第幾天
WEEK(date[, mode])	傳回周數0~52
WEEKDAY(date)	傳回星期索引(0, 一, 6, 日)
WEEKOFYEAR(date)	一年中的第幾週 MySQL 4.1.1
YEARWEEK(date[, start])	Returns year and week

日期/時間函數(Date and Time functions)

```
mysql> SELECT curdate(), DAYOFWEEK(curdate()),  
-> weekday(curdate()), dayofmonth(curdate()), dayofyear(curdate());
```

curdate()	DAYOFWEEK(curdate())	weekday(curdate())	dayofmonth(curdate())	dayofyear(curdate())
2015-03-15	1	6	15	74

```
Mysql> SELECT curdate(), YEARWEEK(curdate());
```

curdate()	YEARWEEK(curdate())
2015-03-15	201511

1 row in set (0.00 sec)

日期/時間函數(Date and Time functions)

函數	功能
DATEDIFF(expr1, expr2)	二個日期相減
ADDDATE(date, INTERVAL expr type)	日期加法
SUBDATE(date, INTERVAL expr type)	日期減法
ADDTIME(expr1, expr2)	時間加法
SUBTIME(expr1, expr2)	時間減法
TIMEDIFF(expr1, expr2)	時間減法
TIMESTAMP(expr1, expr2)	時間加法
TIMESTAMPADD(interval, int_expr, datetime_expr)	時間加法
TIMESTAMPDIFF(interval, datetime_expr1, datetime_expr2)	時間減法
PERIOD_DIFF(P1, P2)	月份(年月-年月)

日期/時間函數(Date and Time functions)

▶ 查詢現在系統日期時間

```
mysql> select now(),DAYNAME(now()),  
              EXTRACT(month FROM curdate()), DATEDIFF(curdate(),'20150317') ;
```

now()	DAYNAME(now())	EXTRACT(month FROM curdate())	DATEDIFF(curdate(),'20150317')
2015-03-15 11:54:45	Sunday	3	-2

▶ 日期時間計算

```
mysql> SELECT curdate(), adddate(curdate(), INTERVAL 3 DAY),  
              adddate(curdate(), INTERVAL 3 month), adddate(curdate(), INTERVAL 3 year);
```

curdate()	adddate(curdate(), INTERVAL 3 DAY)	adddate(curdate(), INTERVAL 3 month)	adddate(curdate(), INTERVAL 3 year)
2015-03-14	2015-03-17	2015-06-14	2018-03-14

日期/時間函數(Date and Time functions)

▶ 日期時間計算

```
mysql> SELECT ename, hiredate, ADDDATE(hiredate,INTERVAL 6 Month)
-> FROM emp;
```

ename	hiredate	ADDDATE(hiredate,INTERVAL 6 Month)
ALLEN	1981-02-20 00:00:00	1981-08-20 00:00:00
JONES	1981-04-02 00:00:00	1981-10-02 00:00:00
MARTIN	1981-09-28 00:00:00	1982-03-28 00:00:00
BLAKE	1981-05-01 00:00:00	1981-11-01 00:00:00
CLARK	1981-06-09 00:00:00	1981-12-09 00:00:00
SCOTT	1982-12-09 00:00:00	1983-06-09 00:00:00
KING	1981-11-17 00:00:00	1982-05-17 00:00:00
TURNER	1981-09-08 00:00:00	1982-03-08 00:00:00
ADAMS	1983-01-12 00:00:00	1983-07-12 00:00:00
MARY	1981-12-03 00:00:00	1982-06-03 00:00:00
MILLER	1982-01-23 00:00:00	1982-07-23 00:00:00

11 rows in set (0.00 sec)

DATE_FORMAT(date, format)

▶ 將日期格式化成字串

- `DATE_FORMAT(curdate(), ' %D of %M , %Y');`
- Format 格式

Specifier	Meaning
%a	Three-characters abbreviated weekday name e.g., Mon, Tue, Wed, etc.
%b	Three-characters abbreviated month name e.g., Jan, Feb, Mar, etc.
%c	Month in numeric e.g., 1, 2, 3...12
%D	Day of the month with English suffix e.g., 0th, 1st, 2nd, etc.
%d	Day of the month with leading zero if it is 1 number e.g., 00, 01,02, ...31
%e	Day of the month without leading zero e.g., 1,2,...31
%f	Microseconds in the range of 000000..999999
%H	Hour with 24-hour format with leading zero e.g., 00..23
%h	Hour with 12-hour format with leading zero e.g., 01, 02...12
%l	Same as %h
%i	Minutes with leading zero e.g., 00, 01,...59
%j	Day of year with leading zero e.g., 001,002,...366
%k	Hour in 24-hour format without leading zero e.g., 0,1,2...23
%l	Hour in 12-hour format without leading zero e.g., 1,2...12

DATE_FORMAT(date, format)

Specifier	Meaning
%M	Full month name e.g., January, February,...December
%m	Month name with leading zero e.g., 00,01,02,...12
%p	AM or PM, depending on other time specifiers
%r	Time in 12-hour format hh:mm:ss AM or PM
%S	Seconds with leading zero 00,01,...59
%s	Same as %S
%T	Time in 24-hour format hh:mm:ss
%U	Week number with leading zero when the first day of week is Sunday e.g., 00,01,02...53
%u	Week number with leading zero when the first day of week is Monday e.g., 00,01,02...53
%V	Same as %U; it is used with %X
%v	Same as %u; it is used with %x
%W	Full name of weekday e.g., Sunday, Monday,..., Saturday
%w	Weekday in number (0=Sunday, 1 = Monday,etc.)
%X	Year for the week in four digits where the first day of the week is Sunday; often used with %V
%x	Year for the week, where the first day of the week is Monday, four digits; used with %v
%Y	Four digits year e.g., 2000, 2001,...etc.
%y	Two digits year e.g., 10,11,12, etc.
%%	Add percentage (%) character to the output

DATE_FORMAT(date, format)

▶ 格式化日期以字串顯示

```
mysql> SELECT curdate(),Date_format(curdate(),' %W %D %M %Y');
+-----+-----+
| curdate() | Date_format(curdate(),' %W %D %M %Y') |
+-----+-----+
| 2015-03-14 | Saturday 14th March 2015 |
+-----+-----+
```

▶ 加入字串常數

```
mysql> SELECT curdate(),Date_format(curdate(),' %W ,the %D of %M, %Y');
+-----+-----+
| curdate() | Date_format(curdate(),' %W ,the %D of %M, %Y') |
+-----+-----+
| 2015-03-14 | Saturday ,the 14th of March, 2015 |
+-----+-----+
```

資料型態轉換函數(Conversion functions)

- ▶ 資料型態轉換函數
 - CAST(expr AS type)
 - CONVERT(expr, type)
- ▶ 資料型態
 - BINARY
 - CHAR
 - DATE
 - DATETIME
 - SIGNED [INTEGER]
 - TIME
 - UNSIGNED [INTEGER]

資料型態轉換函數(Conversion functions)

▶ 使用 Cast 函數

```
mysql> SELECT CAST(NOW() AS DATE) x1,  
->          CAST(CAST(1-2 AS UNSIGNED) AS SIGNED) x2,  
->          CAST(1-2 AS UNSIGNED) x3,  
->          CAST(1 AS UNSIGNED) - 2.0 x4;  
  
+-----+-----+-----+-----+  
| x1          | x2 | x3          | x4      |  
+-----+-----+-----+-----+  
| 2004-06-27 | -1 | 18446744073709551615 | -1.0 |  
+-----+-----+-----+-----+
```

▶ 使用 CONVERT 函數

```
mysql> SELECT CONVERT(NOW(),DATE) x1,  
->          CONVERT(CONVERT(1-2,UNSIGNED),SIGNED) x2,  
->          CONVERT(1-2,UNSIGNED) x3,  
->          CONVERT(1,UNSIGNED)-2.0 x4;  
  
+-----+-----+-----+-----+  
| x1          | x2 | x3          | x4      |  
+-----+-----+-----+-----+  
| 2004-06-27 | -1 | 18446744073709551615 | -1.0 |  
+-----+-----+-----+-----+
```

通用函數(General functions)

▶ 系統資訊函數(Information Functions)

函數	功能
USER()	連線的使用者
VERSION()	資料庫版本
DATABASE()	連線的資料庫
CONNECTION_ID()	連線的Connection ID
CHARSET(str)	字串所屬的字元集

```
mysql> SELECT USER(),VERSION(),DATABASE(),  
->          CONNECTION_ID(), CHARSET('abc');
```

```
+-----+-----+-----+-----+-----+  
| USER() | VERSION() | DATABASE() | CONNECTION_ID() | CHARSET('abc') |  
+-----+-----+-----+-----+-----+  
| jih@localhost | 5.6.22-log | exp | 3 | utf8 |  
+-----+-----+-----+-----+-----+
```

通用函數(General functions)

- ▶ 流程控制函數(Control Flow Functions)
 - IFNULL(expr1, expr2)
 - IF(expr1, expr2, expr3)
 - NULLIF(expr1, expr2)

IFNULL() Function

IFNULL(expr1,expr2)

If expr1 is not NULL,
Then

returns expr1

else

returns expr2

- Return type : numeric or string

```
mysql> SELECT IFNULL(1,0),  
->          IFNULL(NULL,10),  
->          IFNULL(1/0,10),  
->          IFNULL(1/0,'yes');
```

IFNULL(1,0)	IFNULL(NULL,10)	IFNULL(1/0,10)	IFNULL(1/0,'yes')
1	10	10.00	yes

IF() Function

IF(expr1,expr2,expr3)

If expr1 is TRUE

then

returns expr2,

else

returns expr3.

```
mysql> SELECT IF(1>2,2,3) x1,  
->          IF(1<2,'yes','no') x2,  
->          IF(STRCMP('test','test1'),'no','yes') x3;
```

+	+	+	+	+	+
	x1		x2		x3
+	+	+	+	+	+
	3		yes		no
+	+	+	+	+	+

NULLIF () Function

IFNULL (expr1,expr2)

If expr1=expr2 ,

Then

returns NULL

else

returns expr1

- Return type : numeric or string

```
mysql> SELECT  NULLIF(1,1), NULLIF(1,0), NULLIF(0,1);
```

+	-----+	-----+	-----+			
	NULLIF(1,1)		NULLIF(1,0)		NULLIF(0,1)	
+	-----+	-----+	-----+			
	NULL		1		0	
+	-----+	-----+	-----+			

作業練習

建立查詢指令以顯示下列各題描述之資料：

1. 顯示系統目前的日期並將表頭命名為”系統日期”。
2. 顯示所有員工之員工編號, 姓名, 薪資及將薪資增加15%並且以整數表示, 並將表頭命名為” New Salary”。
3. 接續第二題, 增加一個資料項表頭命名為Increase (將New Salary 減掉 salary 的值)。
4. 顯示所有員工之員工編號, 姓名, 進公司日期, 並自訂日期格式為: Sunday, the Seventh of September。
5. 顯示每位員工的姓名, 資料項(MONTHS_WORKED):計算到今天為止工作了幾個月(將月數四捨五入到整數)
6. 顯示如下格式: <員工姓名> earns <薪水> monthly but wants <3倍的薪水>. 並將表頭顯示為Dream Salaries。
7. 顯示所有員工之姓名和薪資, 設定薪資長度為15個字元並且在左邊加上\$符號, 將表頭命名為SALARY。
8. 顯示員工之姓名, 進公司日期, 資料項(DAY):顯示員工被雇用的那天為星期幾, 並以星期一作為一週的起始日, 依星期排序。
9. 顯示員工的姓名和名為COMM的欄位:顯示佣金額, 如果該員工沒有賺取佣金則顯示"No Commission."
10. 顯示資料項命名為 EMPLOYEE_AND_THEIR_SALARIES 的資料來顯示所有員工之名字和薪資, 且用星號來表示他們的薪資, 每一個星號表示100元, 並以薪資由高到低來顯示。