

關聯式資料庫管理系統 結構化查詢語言 (SQL) 使用 MySQL

資料庫物件的維護
(Data Definition Language)
資料表與資料限制條件
(Table & Constraints)

資料庫物件(Database Objects)

- ▶ 資料庫物件是定義在資料庫中用來儲存資料或存取資料的物件
- ▶ 常見的資料庫物件：
 - 資料表(table)、資料限制條件(Constraints)
 - 視觀表(View)、索引(Index)
 - 使用者帳號(User Account)
 - 內儲程式(Stored Program) ...等
- ▶ 維護資料庫物件的命令
 - 新建物件 : CREATE
 - 修改物件 : ALTER
 - 刪除物件 : DROP

資料表(Database Tables)

- ▶ 最主要的資料庫物件(Database Object)
- ▶ 資料庫中儲存資料的物件
- ▶ 主要由欄位(Column)及資料列(Row)所組成
- ▶ 欄位(Column)
 - 每一個欄位都必須命名而且必須設定資料型態和資料長度，用來存放欄位資料。
- ▶ 資料列(Row)
 - 資料列是資料表中的一筆記錄資料

定義資料表(Defining Tables)

▶ 新建資料表時需定義的項目

必要項目(Required Elements)	選擇性項目(Optional Elements)
<ul style="list-style-type: none">• 資料表名稱(Table name)• 欄位名稱(Column names)• 欄位資料型態(Column data types)• 欄位資料長度(column size)	<ul style="list-style-type: none">• 欄位預設值(Default)• 資料檢查條件(Constraints)<ul style="list-style-type: none">• 不允許空值(NOT NULL)• 主鍵(Primary key)• 唯一鍵(Unique key)• 外來鍵(Foreign key)• 資料檢測(Check)

MySQL
不支援

新建資料表

- ▶ 要有建立資料表的權限
- ▶ 必需指定資料表與欄位名稱, 最大長度64 bytes
- ▶ 以字母開頭, 符號可用_, \$
- ▶ 不可使用系統保留字
- ▶ 在同一個資料庫中, 不可有相同名稱的資料表

新建資料表 - CREATE TABLE

```
CREATE TABLE TableName  
( ColumnName type,... );
```

- TableName 表格名稱
- ColumnName 欄位名稱
- Type 欄位資料型態[(長度)]

```
CREATE TABLE DEPT  
( DEPTNO SMALLINT(4),  
  DNAME VARCHAR(14),  
  LOC VARCHAR(13)  
);
```

欄位資料型態(Column Data Types)

▶ 欄位可用的型態

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)		

IF NOT EXISTS 選項

▶ 可判斷資料表是否存在

```
CREATE TABLE [IF NOT EXISTS] TableName  
( ColumnName type,... );
```

- 若資料表已存在，則不建立資料表
- 若資料表不存在，則建立資料表

```
CREATE TABLE IF NOT EXISTS DEPT  
( DEPTNO SMALLINT(4),  
  DNAME VARCHAR(14),  
  LOC VARCHAR(13)  
);
```


資料表儲存引擎

- ▶ MySQL支援多種儲存引擎,以便處理多種類型資料表
- ▶ MySQL支援非交易(速度快)以及交易二種儲存引擎
- ▶ 資料表的儲存引擎類型如下 `SHOW ENGINES;`

儲存引擎類型	支援Tx	說明
ISAM	N	最初的儲存引擎
MyISAM	N	二元可攜式儲存引擎,取代ISAM
BDB	Y	用page locking作安全交易
InnoDB	Y	用row locking作安全交易並支援外來鍵功能
MEMORY/HEAP	N	資料儲存在記憶體中,但關機後立即消失
MERGE/MRG_MyISAM	N	將多個資料表的資料合併 (UNION) 成為一個資料表

註：Tx:Transaction(交易)

設定資料表儲存引擎

```
CREATE TABLE TableName  
( ColumnName type,...)  
  {ENGINE = {BDB|HEAP|ISAM|InnoDB|  
             MERGE|MRG_MYISAM|MYISAM};
```

```
CREATE TABLE IF NOT EXISTS dept  
( deptno smallint(4),  
  dname  varchar(14),  
  loc    varchar(13)  
) ENGINE = InnoDB;
```

註：沒有指定 ENGINE 時，其預定的類型由參數 default-storage-engine 控制

資料檢查條件 (Database Constratins)

- ▶ 用來檢查資料表的資料 (data validation)
 - 欄位預設值(Default)
 - 不允許空值(NOT NULL)
 - 主鍵(Primary key)
 - 唯一鍵(Unique key)
 - 外來鍵(Foreign key)
 - 資料檢測(Check) - (目前MySQL不支援)
- ▶ 可在建立資料表時指定或於資料表建立後加入
 - 欄位階層(Column Level)
 - 資料表階層(Table Level)

欄位預設值(DEFAULT option)

▶ 設定欄位預設值

```
ColumnName type [NOT NULL] | [NULL] [DEFAULT expr]
```

- Expr 需是常值(Literal), 不可以是函數或運算式
- MySQL資料庫中, 所有欄位皆有預設值, 如果不設定其預設值為空值(NULL)

```
CREATE TABLE EMP
(
    ...,
    BRITHDATE DATE,
    SEX CHAR(1) DEFAULT 'M',
    ...
);
```

初值為NULL

初值為 'M'

不允許空值(NOT NULL constratin)

- ▶ 不為空值檢查(一定要輸入資料)

ColumnName type [NOT NULL] | [NULL]

- NOT NULL: 該column所有的rows均要輸入Data
- 只能以欄位階層(Column Level)來設定

```
CREATE TABLE EMP
(  EMPNO      SMALLINT      NOT NULL,
   ENAME      VARCHAR(10)   NOT NULL,
   JOB        VARCHAR(9) ,
   MGR        SMALLINT,
   HIREDATE   DATE,
   SAL        INT,
   COMM       INT,
   DEPTNO     SMALLINT
);
```

主鍵 (PRIMARY KEY constraint)

▶ 主鍵(PK)

ColumnName type **[PRIMARY KEY]**

- 一個資料表中只能有一個主鍵
- 可以欄位階層(Column Level)或資料表階層(Table Level)來設定

```
CREATE TABLE DEPT122
```

欄位階層(Column Level)

```
(  
    DEPTNO SMALLINT(4) PRIMARY KEY,  
    DNAME  VARCHAR(14),  
    LOC    VARCHAR(13)  
);
```

```
CREATE TABLE DEPT122
```

資料表階層(Table Level)

```
(  
    DEPTNO SMALLINT(4),  
    DNAME  VARCHAR(14),  
    LOC    VARCHAR(13),  
    CONSTRAINT PK_DEPT122_DEPTNO PRIMARY KEY (DEPTNO)  
);
```

主鍵 (PRIMARY KEY constraint)

▶ 複合主鍵(由多個欄位構成)

```
CREATE TABLE table
( ...,
  ColumnName type NOT NULL,
  ...,
  [CONSTRAINT const_name] PRIMARY KEY(column, ...)
);
```

- 需以資料表階層(Table Level)來設定
 - **CONSTRAINT** const_name: 為主鍵命名

```
CREATE TABLE ITEM12
(  ORCID          INT          NOT NULL,
   ITEMID         SMALLINT NOT NULL,
   . . .
   CONSTRAINT PK_ITEM_ORCID_ITEMID PRIMARY KEY(ORCID, ITEMID)
);
```

AUTO_INCREMENT 選項

▶ 欄位資料自動編號

ColumnName type **AUTO_INCREMENT**

- 欄位的資料型態必需是數值型態
- 預設起始值:0
- 需與Primary key或Unique key配合使用
- LAST_INSERT_ID() : 取得最近一次自動產生的編號

```
CREATE TABLE ORD2
(  ORCID      INT AUTO_INCREMENT PRIMARY KEY,
  ORD_DATE    DATE,
  ...
);
```


AUTO_INCREMENT option

▶ 欄位自動編號(設定起始號碼)

```
CREATE TABLE TableName  
(ColumnName type AUTO_INCREMENT,  
    ...,  
) AUTO_INCREMENT = start ;
```

- start 設定起始號碼

```
CREATE TABLE ORD2  
(  ORCID      INT    AUTO_INCREMENT PRIMARY KEY,  
  ORD_DATE    DATE,  
  ...,  
) AUTO_INCREMENT = 101 ;
```

唯一鍵 (UNIQUE constraint)

▶ 唯一鍵

```
ColumnName type UNIQUE
```

- 一個資料表中可以多組唯一鍵
- 可以欄位階層(Column Level)或資料表階層(Table Level)來設定

```
CREATE TABLE EMP
```

欄位階層(Column Level)

```
( ...,  
  EMAIL          VARCHAR(200) UNIQUE,  
  ...  
)
```

```
CREATE TABLE EMP
```

資料表階層(Table Level)

```
( ...,  
  EMAIL          VARCHAR(200),  
  ...  
  CONSTRAINT UK_EMP_EMAIL UNIQUE (EMAIL)  
)
```

唯一鍵 (UNIQUE constraint)

▶ 複合唯一鍵(由多個欄位構成)

```
CREATE TABLE TableName
( ...,
  ColumnName type ,
  ...,
  [CONSTRAINT const_name] UNIQUE(column,...)
);
```

- 需以資料表階層(Table Level)來設定
 - const_name 為唯一鍵命名

```
CREATE TABLE ITEM
(  ORDID          INT          NOT NULL,
   ITEMID         SMALLINT NOT NULL,
   PRODID         SMALLINT,
   CONSTRAINT UK_ITEM_ORDID_PRODID UNIQUE (ORDID, PRODID)
);
```

外來鍵(Foreign key constraint)

▶ 外來鍵

```
CREATE TABLE table  
( ...,  
  [CONSTRAINT const_name]  
    FOREIGN KEY(fk_column,...) REFERENCES parent(column,...)  
) ENGINE=InnoDB;
```

- 外來鍵與參考的資料表須是InnoDB 儲存引擎類型
- 外來鍵的欄位數需和 parent table的主鍵或唯一鍵欄位數量相同
- ▶ 可以欄位階層(Column Level)或資料表階層(Table Level)來設定
 - REFERENCES parent(column,...)
 - FOREIGN KEY(fk_column,...) REFERENCES parent(column,...)

外來鍵(Foreign key constraint)

```
CREATE TABLE DEPT
(
  DEPTNO          INT NOT NULL,
  DNAME           VARCHAR(14),
  LOC             VARCHAR(13),
  CONSTRAINT DEPT_PRIMARY_KEY PRIMARY KEY(DEPTNO)
) ENGINE = INNODB;
```

```
CREATE TABLE EMP
(
  EMPNO           INT NOT NULL,
  ENAME           VARCHAR(10),
  JOB             VARCHAR(9),
  MGR             INT,
  HIREDATE        DATETIME,
  SAL             DECIMAL(7,2),
  COMM            DECIMAL(7,2),
  DEPTNO          INT NOT NULL,
  CONSTRAINT EMP_DEPTNO_FK FOREIGN KEY(DEPTNO) REFERENCES DEPT(DEPTNO),
  CONSTRAINT EMP_MGR_FK FOREIGN KEY(MGR) REFERENCES EMP(EMPNO),
  CONSTRAINT EMP_EMPNO_PK PRIMARY KEY(EMPNO)
) ENGINE = INNODB;
```

參考另一個資料表

參考自己資料表

外來鍵(Foreign key constraint)

▶ 外來鍵資料的刪除與修改- PARENT TABLE

```
CREATE TABLE table
(
    ...,
    fk_definition
    [ON DELETE {RESTRICT | CASCADE | SET NULL}] |
    [ON UPDATE {RESTRICT | CASCADE | SET NULL}]
);
```

- ON DELETE 資料刪除時
- ON UPDATE 資料更新時

- RESTRICT 限制
- CASCADE 相依性
- SET NULL 設成空值

外來鍵(Foreign key constraint)

▶ ON DELETE CASCADE

◦ 相依性刪除

```
CREATE TABLE T1
( PK    SMALLINT NOT NULL PRIMARY KEY
) ENGINE = INNODB;

CREATE TABLE T2
( FK    SMALLINT,
  C2    CHAR(2),
  FOREIGN KEY(FK) REFERENCES T1(PK) ON DELETE CASCADE
) ENGINE = INNODB;

INSERT INTO T1 VALUES(1),(2),(3);
INSERT INTO T2 VALUES(1,'A1'),(2,'A2'),(2,'A3'),(3,'A3');
```

T1		T2	
PK		FK	C2
1		1	A1
2		2	A2
3		2	A3
		3	A3

外來鍵(Foreign key constraint)

- ON DELETE SET NULL
 - 相依性刪除

```
CREATE TABLE T1
( PK    SMALLINT NOT NULL PRIMARY KEY
) ENGINE = INNODB;
CREATE TABLE T2
( FK    SMALLINT,
  C2    CHAR(2),
  FOREIGN KEY(FK) REFERENCES T1(PK) ON DELETE SET NULL
) ENGINE = INNODB;
INSERT INTO T1 VALUES(1), (2), (3);
INSERT INTO T2 VALUES(1, 'A1'), (2, 'A2'), (2, 'A3'), (3, 'A3');
```

T1		T2	
PK		FK	C2
1		1	A1
2		NULL	A2
3		NULL	A3
		3	A3

外來鍵(Foreign key constraint)

▶ ON UPDATE CASCADE

◦ 相依性修改

```
CREATE TABLE T1
( PK    SMALLINT NOT NULL PRIMARY KEY
) ENGINE = INNODB;
CREATE TABLE T2
( FK    SMALLINT,
  C2    CHAR(2),
  FOREIGN KEY(FK) REFERENCES T1(PK) ON UPDATE CASCADE
) ENGINE = INNODB;

INSERT INTO T1 VALUES(1), (2), (3);
INSERT INTO T2 VALUES(1, 'A1'), (2, 'A2'), (2, 'A3'), (3, 'A3');
```

T1	
PK	
1	
5	
3	

T2	
FK	C2
1	A1
5	A2
5	A3
3	A3

外來鍵(Foreign key constraint)

▶ ON UPDATE SET NULL

- 外來鍵修改時設為空值

```
CREATE TABLE T1
( PK    SMALLINT NOT NULL PRIMARY KEY
) ENGINE = INNODB;

CREATE TABLE T2
( FK    SMALLINT,
  C2    CHAR(2),
  FOREIGN KEY(FK) REFERENCES T1(PK) ON UPDATE SET NULL
) ENGINE = INNODB;
INSERT INTO T1 VALUES(1), (2), (3);
INSERT INTO T2 VALUES(1, 'A1'), (2, 'A2'), (2, 'A3'), (3, 'A3');
```

T1	
PK	
1	
5	
3	

T2	
FK	C2
1	A1
NULL	A2
NULL	A3
3	A3

外來鍵(Foreign key constraint)

▶ 範例

```
CREATE TABLE DEPARTMENT
(  DEPTNO          SMALLINT NOT NULL PRIMARY KEY,
   DNAME           VARCHAR(14) ,
   LOC             VARCHAR(13)
) ENGINE = INNODB;

CREATE TABLE EMPLOYEE
(  EMPNO           SMALLINT NOT NULL PRIMARY KEY,
   ENAME           VARCHAR(14) ,
   JOB             VARCHAR(13) ,
   MGR             SMALLINT ,
   HIREDATE        DATE ,
   SAL             INT ,
   COMM            INT ,
   DEPTNO          SMALLINT ,
   EMAL            VARCHAR(200) UNIQUE,
   CONSTRAINT FK_EMP_MGR FOREIGN KEY(MGR) REFERENCES EMPLOYEE(EMPNO) ,
   CONSTRAINT FK_EMP_DEPTNO FOREIGN KEY(DEPTNO) REFERENCES DEPARTMENT(DEPTNO)
) ENGINE = INNODB;
```

使用現有資料表來建立新的資料表

▶ 使用子查詢建立新的資料表

```
CREATE TABLE TableName[ (ColumnName, ...) ]  
AS  
SELECT ColumnName, ...  
FROM TableName  
WHERE ... Sub-Query
```

- 可在資料表後指定欄位名稱
 - 若不指定, 則以子查詢中的欄位來命名
 - 亦可在子查詢中使用欄位別名來命名(運算式/函數)
- ## ▶ 子查詢的資料會新增到新建的資料表中
- 只有不為空值的檢查條件會被保留在新的資料表中, 其餘的 Constraints 均不存在

使用現有資料表來建立新的資料表

```
mysql> CREATE TABLE EMP10
-> AS
-> SELECT EMPNO, ENAME, JOB, SAL
-> FROM EMP
-> WHERE DEPTNO = 10;
Query OK, 3 rows affected (0.26 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

```
mysql> desc emp10;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| EMPNO | int(11)       | NO   |     | NULL    |       |
| ENAME | varchar(10)   | YES  |     | NULL    |       |
| JOB   | varchar(9)    | YES  |     | NULL    |       |
| SAL   | decimal(7,2)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

NOT NULL Constraint

使用LIKE子句建立資料表

▶ 使用LIKE子句建立資料表

```
CREATE TABLE TableName LIKE Source_TableName;
```

- 使用現有的資料表結構來建立新的資料表
- 此種方法不會有資料的新增, 只有結構定義部份
- 無法指定欄位名稱, 需和來源資料表欄位同名
- 只有主鍵與不為空值會被保留, 其餘的Constraints均不保留
- 要新增資料請再使用Insert into with Sub-Query方法

使用LIKE子句建立資料表

```
mysql> create table emp1 like emp;  
Query OK, 0 rows affected (0.27 sec)
```

```
mysql> desc emp1;
```

Field	Type	Null	Key	Default	Extra
EMPNO	int(11)	NO	PRI	NULL	
ENAME	varchar(10)	YES		NULL	
JOB	varchar(9)	YES		NULL	
MGR	int(11)	YES	MUL	NULL	
HIREDATE	datetime	YES		NULL	
SAL	decimal(7,2)	YES		NULL	
COMM	decimal(7,2)	YES		NULL	
DEPTNO	int(11)	NO	MUL	NULL	

NOT NULL & PRIMARY KEY

```
mysql> select * from emp1;  
Empty set (0.00 sec)
```

資料表的維護(Managing Tables)

- ▶ 可以修改資料表的結構
 - 新增, 修改, 刪除欄位
 - 新增, 刪除資料檢查條件
- ▶ 修改資料表名稱或欄位名與註解
 - Rename
 - Comment
- ▶ 可以刪除資料表的結構或資料
 - 刪除資料
 - DELETE , TRUNCATE
 - 刪除資料及結構
 - DROP TABLE

新增一個新欄位(Adding a Column)

▶ 加入新欄位

```
ALTER TABLE TableName  
ADD [COLUMN] Column_Definition [FIRST|AFTER column];  
OR  
ALTER TABLE TableName  
ADD [COLUMN] (Column_Definition,...);
```

- Column_Definition 欄位的定義
- FIRST 新增在第一個欄位
- AFTER column 新增在指定欄位後
- 預設為新增在最後一個欄位之後

新增一個新欄位(Adding a Column)

```
mysql> desc emp10;
```

Field	Type	Null	Key	Default	Extra
EMPNO	int(11)	NO		NULL	
ENAME	varchar(10)	YES		NULL	
JOB	varchar(9)	YES		NULL	
SAL	decimal(7,2)	YES		NULL	

4 rows in set (0.00 sec)

```
mysql> ALTER TABLE EMP10
```

```
-> ADD COLUMN MGR SMALLINT;
```

```
Query OK, 0 rows affected (0.36 sec)
```

```
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> desc emp10;
```

Field	Type	Null	Key	Default	Extra
EMPNO	int(11)	NO		NULL	
ENAME	varchar(10)	YES		NULL	
JOB	varchar(9)	YES		NULL	
SAL	decimal(7,2)	YES		NULL	
MGR	smallint(6)	YES		NULL	

5 rows in set (0.00 sec)

新增在
最後一個欄位之後



新增一個新欄位(Adding a Column)

```
mysql> desc emp10;
```

Field	Type	Null	Key	Default	Extra
EMPNO	int(11)	NO		NULL	
ENAME	varchar(10)	YES		NULL	
JOB	varchar(9)	YES		NULL	
SAL	decimal(7,2)	YES		NULL	
MGR	smallint(6)	YES		NULL	

```
mysql> ALTER TABLE EMP10
```

```
-> ADD COLUMN HIREDATE DATE AFTER JOB;
```

```
Query OK, 0 rows affected (0.34 sec)
```

```
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> desc emp10;
```

Field	Type	Null	Key	Default	Extra
EMPNO	int(11)	NO		NULL	
ENAME	varchar(10)	YES		NULL	
JOB	varchar(9)	YES		NULL	
HIREDATE	date	YES		NULL	
SAL	decimal(7,2)	YES		NULL	
MGR	smallint(6)	YES		NULL	

新增在 JOB
欄位之後

```
6 rows in set (0.00 sec)
```

新增一個新欄位(Adding a Column)

▶ 新增為第一個欄位

```
mysql> ALTER TABLE emp10  
-> ADD PHONE varchar(12) DEFAULT '02-66316710' FIRST;  
Query OK, 0 rows affected (0.39 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> desc emp10;
```

Field	Type	Null	Key	Default	Extra
PHONE	varchar(12)	YES		02-66316710	
EMPNO	int(11)	NO		NULL	
ENAME	varchar(10)	YES		NULL	
JOB	varchar(9)	YES		NULL	
HIREDATE	date	YES		NULL	
SAL	decimal(7,2)	YES		NULL	
MGR	smallint(6)	YES		NULL	
COMM	int(11)	YES		NULL	
DEPTNO	smallint(6)	YES		NULL	
EMAIL	varchar(200)	YES		NULL	

```
10 rows in set (0.00 sec)
```

新增多個新欄位(Adding Columns)

▶ 新增多個欄位

```
mysql> ALTER TABLE EMP10  
      -> ADD COLUMN(COMM INT, DEPTNO SMALLINT, EMAIL VARCHAR(200));  
Query OK, 0 rows affected (0.37 sec)  
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> desc emp10;
```

Field	Type	Null	Key	Default	Extra
EMPNO	int(11)	NO		NULL	
ENAME	varchar(10)	YES		NULL	
JOB	varchar(9)	YES		NULL	
HIREDATE	date	YES		NULL	
SAL	decimal(7,2)	YES		NULL	
MGR	smallint(6)	YES		NULL	
COMM	int(11)	YES		NULL	
DEPTNO	smallint(6)	YES		NULL	
EMAIL	varchar(200)	YES		NULL	

9 rows in set (0.00 sec)

修改欄位定義 (Modifying a column)

- ▶ 修改欄位定義
 - 修改欄位預設值

```
ALTER TABLE TableName  
ALTER [COLUMN] column {SET DEFAULT literal|DROP DEFAULT};
```

- 修改欄位資料型態
- 修改欄位順序

```
ALTER TABLE TableName  
MODIFY [COLUMN] column_definition [FIRST|AFTER column];
```

修改欄位定義 (Modifying a column)

▶ 修改欄位預設值 (取消預設值設定)

```
mysql> ALTER TABLE EMP10  
      -> ALTER PHONE DROP DEFAULT;  
Query OK, 0 rows affected (0.07 sec)  
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> desc emp10;
```

Field	Type	Null	Key	Default	Extra
PHONE	varchar(12)	YES		NULL	
EMPNO	int(11)	NO		NULL	
ENAME	varchar(10)	YES		NULL	
JOB	varchar(9)	YES		NULL	
HIREDATE	date	YES		NULL	
SAL	decimal(7,2)	YES		NULL	
MGR	smallint(6)	YES		NULL	
COMM	int(11)	YES		NULL	
DEPTNO	smallint(6)	YES		NULL	
EMAIL	varchar(200)	YES		NULL	

```
10 rows in set (0.00 sec)
```

修改欄位定義 (Modifying a column)

▶ 修改欄位資料型態/順序

```
mysql> ALTER TABLE EMP10  
      -> MODIFY COLUMN MGR INT AFTER JOB;  
Query OK, 3 rows affected (0.54 sec)  
Records: 3  Duplicates: 0  Warnings: 0
```

```
mysql> desc emp10;
```

Field	Type	Null	Key	Default	Extra
PHONE	varchar(12)	YES		NULL	
EMPNO	int(11)	NO		NULL	
ENAME	varchar(10)	YES		NULL	
JOB	varchar(9)	YES		NULL	
MGR	int(11)	YES		NULL	
HIREDATE	date	YES		NULL	
SAL	decimal(7,2)	YES		NULL	
COMM	int(11)	YES		NULL	
DEPTNO	smallint(6)	YES		NULL	
EMAIL	varchar(200)	YES		NULL	

```
10 rows in set (0.00 sec)
```


修改欄位定義 (Modifying a column)

▶ 修改欄位資料型態 / NOT NULL

```
mysql> ALTER TABLE EMP10  
      -> MODIFY COLUMN ENAME VARCHAR(20) NOT NULL;  
Query OK, 3 rows affected (0.54 sec)  
Records: 3  Duplicates: 0  Warnings: 0
```

```
mysql> desc emp10;  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type          | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| PHONE      | varchar(12)   | YES  |     | NULL    |      |  
| EMPNO      | int(11)       | NO   |     | NULL    |      |  
| ENAME      | varchar(20)  | NO  |     | NULL   |      |  
| JOB        | varchar(9)    | YES  |     | NULL    |      |  
| MGR        | int(11)       | YES  |     | NULL    |      |  
| HIREDATE   | date         | YES  |     | NULL    |      |  
| SAL        | decimal(7,2)  | YES  |     | NULL    |      |  
| COMM       | int(11)       | YES  |     | NULL    |      |  
| DEPTNO     | smallint(6)   | YES  |     | NULL    |      |  
| EMAIL      | varchar(200)  | YES  |     | NULL    |      |  
+-----+-----+-----+-----+-----+-----+  
10 rows in set (0.00 sec)
```

更改欄位名稱(Rename the Name of Column)

▶ 更改欄位名稱/定義

```
ALTER TABLE TableName  
CHANGE [COLUMN] old_column New_column_definition;
```

- old_column 舊欄位名稱
- New_column_definition 新欄位定義

更改欄位名稱(Rename the Name of Column)

► 修改欄位型態 / NOT NULL

```
mysql> ALTER TABLE EMP10  
      -> CHANGE COLUMN SAL SALARY SMALLINT;  
Query OK, 3 rows affected (0.54 sec)  
Records: 3  Duplicates: 0  Warnings: 0
```

```
mysql> desc emp10;
```

Field	Type	Null	Key	Default	Extra
PHONE	varchar(12)	YES		NULL	
EMPNO	int(11)	NO		NULL	
ENAME	varchar(20)	NO		NULL	
JOB	varchar(9)	YES		NULL	
MGR	int(11)	YES		NULL	
HIREDATE	date	YES		NULL	
SALARY	smallint(6)	YES		NULL	
COMM	int(11)	YES		NULL	
DEPTNO	smallint(6)	YES		NULL	
EMAIL	varchar(200)	YES		NULL	

```
10 rows in set (0.00 sec)
```

刪除欄位(Dropping a column)

▶ 刪除欄位

```
ALTER TABLE TableName  
DROP [COLUMN] ColumnName;
```

- 欄位有無資料皆可刪除, 資料表最少留一個欄位

```
mysql> ALTER TABLE dept1  
-> DROP COLUMN loc;  
Query OK, 0 rows affected (0.46 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> desc dept1;  
+-----+-----+-----+-----+-----+-----+  
| Field  | Type          | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| DEPTNO | int(11)       | NO   | PRI | NULL    |      |  
| DNAME  | varchar(14)   | YES  |     | NULL    |      |  
+-----+-----+-----+-----+-----+-----+  
2 rows in set (0.00 sec)
```

新增資料檢查條件(Adding a constraint)

▶ 加入資料檢查條件

```
ALTER TABLE TableName  
  ADD [CONSTRAINT const_name] constraint_type;
```

- const_name 限制條件名稱
- constraint_type 限制條件種類
- NOT NULL constraint 請用MODIFY句子

```
/* 新增主鍵 */  
ALTER TABLE EMP10  
  ADD CONSTRAINT PK_EMP10_EMPNO PRIMARY KEY(EMPNO);  
/* 新增外來鍵 */  
ALTER TABLE EMP10  
  ADD CONSTRAINT FK_EMP10_MGR FOREIGN KEY(MGR)  
                                     REFERENCES EMP10(EMPNO);  
/* 新增唯一鍵 */  
ALTER TABLE EMP10  
  ADD CONSTRAINT UK_EMP10_EMAIL UNIQUE(EMAIL);
```

刪除資料檢查條件(Drop a constraint)

```
ALTER TABLE table  
DROP [PRIMARY KEY|FOREIGN KEY ConstraintName |  
      INDEX ConstraintName];
```

```
/* 刪除外來鍵 */  
ALTER TABLE EMP10,  
      DROP FOREIGN KEY FK_EMP10_MGR;
```

```
/* 刪除主鍵 */  
ALTER TABLE EMP10  
      DROP PRIMARY KEY;
```

```
/* 刪除唯一鍵 */  
ALTER TABLE EMP10  
      DROP INDEX UK_EMP10_EMAIL;
```

刪除資料檢查條件(Drop a constraint)

▶ 刪除UNIQUE key

```
mysql> ALTER TABLE EMP10
  -> ADD CONSTRAINT EMP10_ENAME_UK UNIQUE(ENAME);
Query OK, 0 rows affected (0.60 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> UPDATE EMP10
  -> SET ENAME='CLARK'
  -> WHERE EMPNO=7934;
ERROR 1062 (23000): Duplicate entry 'CLARK' for key 'EMP10_ENAME_UK'

mysql> ALTER TABLE EMP10
  -> DROP INDEX EMP10_ENAME_UK;
Query OK, 3 rows affected (0.59 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> UPDATE EMP10
  -> SET ENAME='CLARK'
  -> WHERE EMPNO=7934;
Query OK, 1 row affected (0.04 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

資料表改名 (Managing the Name of Table)

▶ 資料表重新命名

```
ALTER TABLE TableName  
RENAME [TO] new_table;
```

▶ 將 EMP10 改名為 EMP10A

```
ALTER TABLE EMP10  
RENAME TO EMP10A;
```


更改資料表儲存引擎類型

▶ 命令

```
ALTER TABLE TableName  
ENGINE = type;
```

```
ALTER TABLE EMP10A  
ENGINE = INNODB;
```

截斷資料表中的資料(Truncating a Table)

- ▶ 刪除資料表中的所有資料
- ▶ 釋出所使用的磁碟空間
- ▶ 保留資料表的結構

```
TRUNCATE TABLE TableName;
```

```
TRUNCATE TABLE EMP10A;
```

刪除資料表(Dropping a Table)

- ▶ 刪除資料表物件
 - 刪除資料表的所有資料及結構

```
DROP TABLE TableName;
```

```
DROP TABLE EMP10A;
```

作業練習

1. 利用下列資料來新建 DEPARTMENT 資料表

Column Name	Null?	Data Type
id	NOT NULL	NUMERIC(7)
name	NOT NULL	VARCHAR(24)

2. 利用DEPT資料表中的資料, 將資料新增至DEPARTMENT 資料表中(只新增相對的資料欄)

3. 利用下列資料來新建 EMPLOYEE資料表

Column Name	Null?	Data Type
id	NOT NULL	NUMERIC(7)
last_name	NOT NULL	VARCHAR(24)
first_name		VARCHAR(24)
dept_id		NUMERIC(7)

4. 將EMPLOYEE資料表中last_name欄位的資料型態更改為 varchar(40)

作業練習

5. 使用EMP資料表的結構中之EMPNO, ENAME, and DEPTNO之定義來新建EMPLOYEE2資料表並將欄位名稱設定為 ID, LAST_NAME, and DEPT_ID
6. 刪除整個EMPLOYEE資料表
7. 將EMPLOYEE2資料表改名為EMPLOYEE
8. 將EMPLOYEE資料表中的LAST_NAME欄位刪除
9. 修改EMPLOYEE資料表, 新增一個欄位 SALARY 資料型態為 NUMERIC datatype, precision 7.
10. 修改EMPLOYEE資料表, 使用ID 欄位新增一個Primary key 的限制條件(constraint), 並為他命名
11. 在EMPLOYEE資料表新增一個外部鍵(foreign key)以確保員工不會被分派到一個不存在的部門