1. 範例

|  |  |
| --- | --- |
| 功能 | 範例 |
| 建立資料庫(DDL) | CREATE DATABASE 學校  ON PRIMARY**--檔案群組**  **--建立DB**  (NAME=’學校’,**--db名稱**  FILENAME=’C:\DATA\學校.mdf’,**--存取路徑**  SIZE=8MB,  MAXSIZE=10MB,  FILEGROWTH=1MB  ),  **--建立FileGroup**  FILEGROUP 檔案群組1  (  NAME=’學校\_檔案群組一’,  FILENAME=’C:\DATA\學校\_檔案群組一.ndf’,  SIZE=8MB,  MAXSIZE=10MB,  FILEGROWTH=1MB  ),  (  NAME=’學校\_檔案群組二’,  FILENAME=’C:\DATA\學校\_檔案群組二.ndf’,  SIZE=8MB,  MAXSIZE=10MB,  FILEGROWTH=1MB  )  **--建立log檔**  LOG ON  (NAME=’學校\_log’,  FILENAME=’C:\DATA\學校.ldf’,  SIZE=8MB,  MAXSIZE=10MB,  FILEGROWTH=10%  )  [COLLATE] 定序名稱**--資料排列方式**  [FOR ATTACH] |
| 新增(DDL)-ADD | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*新增檔案\*\*\*\*\*\*\*\*\*\*\*\*  ALERT DATABASE 學校 ADD FILE  (NAME=’學校’,  FILENAME=’C:\DATA\學校.mdf’,  SIZE=8MB,  MAXSIZE=10MB,  FILEGROWTH=1MB  )  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*新增LOG檔\*\*\*\*\*\*\*\*\*\*\*\*  ALERT DATABASE 學校 ADD LOG FILE  (NAME=’學校’,  FILENAME=’C:\DATA\學校.ldf’,  SIZE=8MB,  MAXSIZE=10MB,  FILEGROWTH=1MB  ) |
| 修改(DDL)-ALERT | \*\*\*\*\*\*\*\*\*\*\*\*\*\*修改檔案空間\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ALERT DATABASE 學校 MODIFY FILE  (NAME=’學校’,  SIZE=5MB  …  )  \*\*\*\*\*\*\*\*\*\*\*\*\*\*修改檔案群組\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ALERT DATABASE 學校 MODIFY FILEGROUP 群組二 |
| 刪除(DDL)-DROP | DROP DATABASE 學校,產品 |
| 卸載(DDL)-detach | EXEC sp\_detach\_db ‘圖書’ |
| 上載(DDL)-attach | EXEC sp\_attach\_db ‘圖書’,’c:\.....’ |
| 修改名稱(DDL) | EXEC sp\_rename ‘訂單’, ‘學校訂單’ |
| 建立資料表(DDL) | CREATE TABLE 教授(  教授編號 INT PRIMARY KEYIDENTITY(1,1),  **--主鍵、自動生成(start=1,step=1)**  教授ID varchar(10) NOT NULL UNIQUE,**--unique**  體重 INT DEFAULT 80,**--預設值**  身高 INT,  BMI AS 體重/身高\*身高**--計算欄位,**  **小孩數 int SPARSE --鬆散欄位**  ) |
| 欄位限制(DDL) | CREATE TABLE XXX(  … CHECK(…>…),  …,  …,  PRIMARY KEY(欄1,欄2)**—不用,**  CONSTRAINT 條件約束名**—設定條件約束名，不用,**  CHECK( (欄1>5) AND (欄2<4) OR (欄3>0) ),  FOREIGN KEY(欄1) REFERENCES 表2 (欄2),  FOREIGN KEY(欄2) REFERENCES 表3 (欄1)  ) |
| 修改欄位(DDL) | ALERT TABLE 表1  ADD教授ID varchar(10) NOT NULL UNIQUE**—新**  DROP COLUMN 教授**—刪欄位**  MODIFY 教授ID int**—修型別**  CHANGE 教授ID IDnum int**—修欄位名稱** |
| 清空表格資料(DDL) | TRUNCATE TABLE 表格名 |
| 修改限制條件(DDL) | ALERT TABLE 學生  \*\*\*\*\*\*\*\*\*\*增加限制條件\*\*\*\*\*\*\*\*\*\*\*  ADD CONSTRAINT XXX  CHECK(A<B)  \*\*\*\*\*\*\*\*\*增加PK、FK\*\*\*\*\*\*\*\*\*\*\*\*  ADD PRIMARY KEY(欄位名)  ADD FOREIGN KEY(欄2) REFERENCES 表3 (欄1)  \*\*\*\*\*\*\*\*\*\*刪除限制條件\*\*\*\*\*\*\*\*\*\*\*  DROP CONSTRAINT 條件名稱 |
| 刪除資料表 | DROP TABLE XXX |
| 建立區域暫存資料表(DDL) | CREATE TABLE #學校( **--名稱以#開頭**  ) |
| 建立全域暫存資料表(DDL) | CREATE TABLE ##學校( **--名稱以##開頭**  ) |
| SELECT(DML) | SELECT \*  FROM table name  [WHERE search condition]  [GROUP BY column name]  [HAVING search condition]**—二階段篩選**  [ORDER BY 欄一 [ASC, DESC],欄二 [ASC, DESC]]  **—(多欄)排序依據**  **可用欄位名或順序**  **{ASC}-小到大**  **DESC-大到小** |
| SELECT (DML)  -SELECT | SELECT TOP 3 WITH TIES \* FROM 課程 ORDER BY 學分  **--取出前三個，但學分相同的一併取出**  **SELECT \*/**欄名/欄名 AS 自訂欄名/函式 AS 欄名/DISTINCT 欄名/TOP 100 欄名  /TOP 25 PERCENT 欄名/COUNT(DISTINCT 欄名)  /AVG(欄名)/MAX(欄名)/MIN(欄名)/SUM(欄名)  **--AS後表顯示欄名**  **--DISTINCT表重複值只取一個** |
| SELECT (DML)  -WHERE | WHERE (NOT) LIKE/BETWEEN AND/IN  **--BETWEEN AND可接日期、數字**  BETWEEN ‘1994-06-28’ AND ‘2017-06-29’  **--IN後面接VECTOR**  IN(‘PLMA2266’,’PLMA5533’,’CISDF2583’)  **--LIKE後符號**  **%:任何長度字串**  **\_:一個任何字元**  **[AGB]:符合括號中任一字元**  **[A-Z]:符合括號中範圍內字元**  **[^]:符合不在括號中範圍內字元** |
| SELECT (DML)  -GROUP BY | SELECT 課程編號,COUNT(\*) AS 學生數  FROM 班級 GROUP BY課程編號  --每個課程有多少學生 |
| 子查詢 | SELECT \*  FROM (SELECT ……) 產品表A LEFTJOINT 產品表B  ON A.產品代碼=B產品代碼 |
| 擷取字串 | --SQL、MYSQL--SUBSTRING(obj,start,length)  --ORACLE---SUBSTR(obj,start,length) |
| 黏接字串 | MYSQL:CONCAT()---CONCAT(A,” ”,B)  ORACLE:CONCAT、||  SQL:+---A+” ”+B |
| 去空值 | LTRIM(字串)---去字首空字串(leading)  RTRIM(字串)---去字尾空字串(trailing) |
| 取長度 | ORACLE、MYSQL---LENGTH(字串)  SQL---LEN(字串) |
| 取代 | REPLACE(字串,被取代字,取代字) |
| 建立索引 | CREATE INDEX "索引名" ON TABLE名(欄1,欄2) |
| INSERT | \*\*\*\*\*\*\*\*\*\*\*\*\*\*單筆匯入\*\*\*\*\*\*\*\*\*\*\*\*  INSERT INTO 表格名 (欄1,欄2) VALUES(‘A’,’B’)  \*\*\*\*\*\*\*\*\*\*\*\*\*\*別資料表匯入\*\*\*\*\*\*\*\*\*  INSERT INTO 表格名 (欄1,欄2)  SELECT 匯入的欄位 |
| UPDATE | UPDATE 表格名  SET 欄位名=”值”  WHERE ….  若篩出多筆資料，該欄資料都被修改 |
| UNION-聯集 | SELECT……  UNION  SELECT……  將SELECT出來的值上下組合，欄位型別需相同，相同值**會**被剔除 |
| UNION ALL-聯集 | SELECT……  UNION ALL  SELECT……  將SELECT出來的值上下組合，欄位型別需相同，相同值**不會**被剔除  B  A |
| LEFT JOIN(差集) | SELECT OO  FROM XX A LEFTJOIN XXXX B  ON A.FF=B.FF  WHERE UUUU  C:\Users\Z00040868\Desktop\下載 (1).png |
| INNER JOIN(交集)—AND | C:\Users\Z00040868\Desktop\下載.png |
| OUTTER JOIN(聯集)—OR | C:\Users\Z00040868\Desktop\images.png |
| 取出同時出現在兩張表的值-交集 | SELECT……  INTERSECT  SELECT……  只選出相同值 |
| 取出只有第一張表有的-差集 | SELECT……  MINUS  SELECT……  只選出第一張表有，而第二張表沒有的 |
| 欄1查詢有結果才觸發WHERE條件 | SELECT "欄1" FROM "表格1" WHERE **EXISTS** (SELECT \*  FROM "表格2" WHERE "條件"); |
| CASE—流程判斷  --製造一個可判斷式欄位，可以兩種CASE方法做判斷，並在END後加名稱 | SELECT Store\_Name,  -CASE Store\_Name WHEN 'Los Angeles' THEN Sales \* 2   WHEN 'San Diego' THEN Sales \* 1.5  -CASE WHEN StorePrice=5 THEN … ELSE Sales END "New Sales", Txn\_Date FROM Store\_Information; |
| 建立檢視表 | CREATE VIEW 檢視表名稱 (學號,學生姓名,學生電話)  AS  SELECT num,name,phone From Student  GO  SELECT \* FROM檢視表名稱  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  CREATE VIEW 高薪人員\_VIEW  AS  SELECT \* FROM 員工  WHERE 薪資>50000  GO  SELECT \* FROM高薪人員\_VIEW  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  CREATE VIEW 高薪三欄\_VIEW (身分證字號,姓名,電話)  AS  SELECT IDno,Name,Phone FROM 員工  WHERE Salary>50000  GO  SELECT \* FROM高薪三欄\_VIEW |
| 更改檢視表 | ALERT VIEW 學生\_VIEW  AS  …(SELECT語法) |
| 建立預存程序 | CREATE PROCEDURE 課程資料報表  AS  BEGIN  預存程序  END  GO |
| 執行預存程序 | EXEC 預存程序名稱 |
| 以變數執行預存程序 | DECLARE @pro\_name char(20)  SET @pro\_name=’pro名稱’  EXECUTE @pro\_name |
| 暫存預存程序 | pro名以#開頭，其他照舊 |
| 建立含有參數的預存程序 | CREATE PROCEDURE 課程查詢  @c\_no char(5)  AS  BEGIN  SELECT \*  FROM 課程  WHERE 課程編號=@c\_no  END  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  CREATE PROCEDURE 員工查詢  @salary money =-1,--可指定預設值  @tax money  AS  BEGIN  IF @salary<=0  SET @salary=30000  IF @tax<=0  SET @tax=300  SELECT…  …  RETURN 回傳值  END |
| 執行含有參數的預存程序 | EXEC 員工查詢 10000,100 |
| 修改PROCEDURE  -等於重新定義PRO | ALERT PROCEDURE 預存程式名  AS  … |
| PROC執行完後回傳內部參數--OUTPUT | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*建立PROC\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  CREATE PROC AAA  @test int OUTPUT,  @name char(20)  AS  …  \*\*\*\*\*\*\*\*\*\*\*\*\*\*抓取Parameter\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  DECLARE @test int;  EXEC AAA '1',’John’,@test output;  PRINT @test; |
| 基礎Debug | PRINT @@ERROR🡪印出錯誤碼  CAST(@@NESTLEVEL AS char)🡪將巢狀層數轉為字串 |
| 系統與擴充預存程序 | sp\_help [名稱]🡪回傳資料庫物件、自訂類型、SQL內建資訊…詳細資料  sp\_helptext 名稱🡪傳回參數預存程序、自訂函數、觸發程序、檢視表內容  sp\_dbhelp [DB名稱]🡪傳回資料庫資訊，沒指定就是所有資料庫資訊  sp\_columns Table名稱🡪回傳檢視表或資料表欄位資訊  … |
| 建立順序物件 | CREATE SEQUENCE 編號順序 AS INT  START WITH 1🡪起始值  INCREASE BY 1🡪增加值  NO MINVALUE🡪沒有最小值  MAXVALUE 100🡪最大值為100  CYCLE🡪超過邊界值即從起始值重新  NO CATCH🡪不儲存CATCH來增加效率 |
| 使用順序物件 | INSERT INTO 客戶 (編號,名稱,ID)  VALUES (NEXT VALUE FOR 編號順序,’王大明’,’F124578854’)  GO |
| 更改順序物件 | ALERT SEQUENCE 編號順序  RESTART WITH 1  其餘格式與CREATE一樣，ALERT無法更動物件型別 |
| 純量值函數 | \*\*\*\*\*\*\*\*\*\*\*\*\*建立FUN\*\*\*\*\*\*\*\*\*\*\*  CREATE FUNCTION fnGETSALARY  (@name varchar(5))  RETURNS money🡪指定回傳型別  AS  BEGIN  DECLARE @salary money  SELECT @salary=(薪水-保險-稅)  FROM 員工  WHERE 姓名=@name  IF @@ROWCOUNT=0🡪沒有當筆資料  RETURN 0  RETURN @salary🡪回傳值  END  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*使用資料\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  PRINT ‘薪水: ’ + CONVERT(varchar,dbo.fnGETSALARY(‘阿呆’)) |
| 嵌入資料表值函數 | \*\*\*\*\*\*\*\*\*\*\*\*\*建立FUN\*\*\*\*\*\*\*\*\*\*\*  CREATE FUNCTION AA  (@salary int)  RETURNS TABLE  RETURN(  SELECT…  )  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*使用資料\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  SELECT \* FROM AA(50000) |
| 多陳述式資料表值函數 | CREATE FUNCTION fnEmployee  (@m int,@n int)  RETURNS @outTable Table  (編號 int IDENTITY(1,1),  身分證字號 char(10),  姓名 varchar(12),  地址 varchar(20),  電話 char(12),  薪水淨額 money  )  BEGIN  INSERT @outTable  SELECT 編號,身分證字號,姓名,地址,(電話,薪水-保險-稅) FROM 員工  DELETE @outTable WHERE 編號<@m OR 編號>@n  RETURN  END |
| 運用FUN做型別驗證 | CREATE FUNCTION fnValidCode  (@p\_no char(5))  RETURN bit  BEGIN  DECLARE @valid bit,@number int  SET @valid=0  IF @p\_no LIKE’[A-Z][0-9] [0-9] [0-9] [0-9]’  驗證程序  IF… valid=1  END  RETURN @valid  GO  CREATE Table 代銷產品(  編號 char(5),  名稱 varchar(5),  定價 money,  CHECK (dbo.fnValidCode (編號)=1)  ) |
| 修改FUNCTION | ALERT FUNCTION …  照舊 |
| 替相同資料編序號 | ROW\_NUMBER() over (PARTITION by [案件編號],[借款科目],[申貸金額(仟元)],[償還方式代碼],[償還方式],[借款用途代碼],[借款用途],[金流],[送保] order by QRType desc)  as rownumber from  -- ROW\_NUMBER() over產生序號  -- PARTITION by 劃分群體不同，若沒有則等於在最後一列加列數  --order by排列群體 |
| 日期加減計算 | SELECT DATEADD(日期區間<日月年>,起始日期,增減期間)  -- SELECT DATEADD(MONTH,GETDATE(),2)🡪兩個月後 |
| 計算日期區間 | SELECT DATEDIFF(日期區間<日月年>,起始日期,結束日期)  =結束日期-起始日期(單位為第一個參數)  -- SELECT DATEDIFF(DAY,’2017-05-01’,’2017-05-02’)🡪1  **\*\*\*\*\*日期計算為無條件捨去**  **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*取當月一號\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***  --SELECT DATEADD(MONTH,DATEDIFF(MONTH,GETDATE(),0),0) |
|  |  |