

資料庫期末（作業版）

資料庫管理系統

第二次練習

1. --新增記錄 單筆輸入 整批輸入

INSERT INTO dbo.估價

VALUES (7,10,'2017/11/12')

2. --更新的欄位不只一個請使用逗號分隔

UPDATE dbo.估價 SET 數量=4, 日期= getdate()

where 數量 = 3

3. --將資料表符合條件的記錄刪除掉

DELETE FROM dbo.估價 WHERE 日期 is null

1. SELECT column1, column2 FROM table WHERE conditions

<ex>查詢 所有客戶的名稱、聯絡人、連絡人職稱和電話資料

SELECT 客戶.公司名稱, 客戶.連絡人, 客戶.連絡人職稱, 客戶.電話

FROM 客戶;

2. 欄位（單一欄位）中的資料重複，擁有相同值，在SELECT指令只需加上DISTINCT指令，就只會顯示其中一筆記錄資料。

<ex>查詢 產品的成本有多少種

SELECT DISTINCT 產品資料.單價

FROM 產品資料;

3. WHERE條件子句

WHERE條件的欄位如果是字串需要加上單引號或雙引號

日期/時間字串需要使用「#」符號括起

5. NOT運算子可以搭配前述子句，取得與條件相反的查詢結果 (優先順序以括號為優先)

查詢產品資訊的產品名稱有"Nano"子字串或入庫日期有"7"子字串，這2個條件使用括號括起，而且定價需要大於等於5000的記錄

SELECT * FROM [dbo].[產品資料]

WHERE ([產品] LIKE '%Nano%' OR [庫存量] LIKE '%7%') AND [單價]>=5

Like

<ex> 查詢產品名稱擁有子字串"起司"

SELECT * FROM [dbo].[產品資料]

WHERE 產品 LIKE '%起司%'

<ex> 查詢產品名稱是以任何字串開頭，結尾為"起司"

SELECT DISTINCT 產品資料.*

FROM 產品資料

WHERE (((產品資料.[產品]) Like '%起司%'));

<ex>查詢產品的單位數量是以子字串"12"

SELECT * FROM [dbo].[產品資料]

WHERE 單位數量 LIKE '%12%'

<ex>查詢單位數量不是以子字串"12"

SELECT * FROM 產品資料

WHERE 單位數量 not LIKE '%12%'

4. AND與OR多條件查詢

<ex>查詢產品資訊的單位數量是以子字串"12"且產品名稱有"油"子字串

SELECT 產品資料.產品, *

FROM 產品資料

WHERE (((產品資料.[單位數量]) Like '%12%') AND ((產品資料.產品) Like '%油%'))

資料庫期末（作業版）

6. ORDER BY子句指定排序欄位，就可以由小到大進行排序

<ex>查詢定價大於等於3500元的記錄，並且使用定價欄位由小到大進行排序

```
SELECT * FROM [dbo].[產品資料]
```

```
WHERE [單價]>=30 ORDER BY [單價] ASC
```

7. BETWEEN AND子句可以定義SELECT指令WHERE條件子句的範圍，範圍值可以使用文字、數值或日期/時間

<ex> 查詢1992年1月1日到12月31日雇用的員工資料

```
SELECT 員工.*
```

```
FROM 員工
```

```
WHERE (((員工.雇用日期) Between '1/1/1992' And '12/31/1992'));
```

8. IN運算子可以設定一串文字或數值清單

<ex>查詢單價(成本)為38, 43.9, 和123.79的產品資料

```
SELECT * FROM 產品資料
```

```
WHERE 單價 IN (38,43.9,123.79)
```

問題

【a】 產品單價有多少種?欄位不要重複

```
SELECT DISTINCT 產品資料.單價
```

```
FROM 產品資料
```

```
ORDER BY 產品資料.單價;
```

【b】 查詢所有客戶的名稱、聯絡人和電話資料，其中連絡人職稱為董事長或業務

```
SELECT 客戶.公司名稱, 客戶.連絡人, 客戶.電話
```

```
FROM 客戶
```

```
WHERE (((客戶.連絡人職稱)='董事長' Or (客戶.連絡人職稱)='業務'));
```

```
SELECT 客戶.公司名稱, 客戶.連絡人, 客戶.電話
```

```
FROM 客戶
```

```
WHERE (客戶.連絡人職稱) in ('董事長','業務');
```

【c】 查詢所有客戶的名稱、聯絡人和電話資料，其中公司名稱有”中”的資料

資料庫期末（作業版）

SELECT 客戶.公司名稱, 客戶.連絡人, 客戶.電話, 客戶.公司名稱

FROM 客戶

where 公司名稱 like '%中%'

【d】 查詢貢獻最好的顧客

SELECT 客戶.公司名稱, Sum([訂貨明細].[單價]*[訂貨明細].[數量]) AS total

FROM (客戶 INNER JOIN 訂貨主檔 ON 客戶.客戶編號 = 訂貨主檔.客戶編號) INNER JOIN 訂貨明細 ON 訂貨主檔.訂單號碼 = 訂貨明細.訂單號碼

GROUP BY 客戶.公司名稱

ORDER BY Sum([訂貨明細].[單價]*[訂貨明細].[數量]) DESC;

【e】 查詢販賣顧客所購買最多的產品

SELECT 客戶.公司名稱, 產品資料.產品, Sum([訂貨明細].[單價]*[訂貨明細].[數量]) AS total

FROM 產品資料 INNER JOIN ((客戶 INNER JOIN 訂貨主檔 ON 客戶.客戶編號 = 訂貨主檔.客戶編號) INNER JOIN 訂貨明細 ON 訂貨主檔.訂單號碼 = 訂貨明細.訂單號碼) ON 產品資料.產品編號 = 訂貨明細.產品編號

GROUP BY 客戶.公司名稱, 產品資料.產品

ORDER BY 客戶.公司名稱, Sum([訂貨明細].[單價]*[訂貨明細].[數量]) DESC;

作業 2 十個查詢 自訂問題 下達SQL查詢語言 結果

查詢 所有員工的名稱、聘用日期

SELECT [dbo].[employee].[fname],[lname],[hire_date]

from [dbo].[employee];

查詢 出版商來自哪幾個國家（不重複）

SELECT DISTINCT [country]

FROM [dbo].[publishers];

查詢 JOBS的max_lvl為大於"200"

SELECT * FROM [dbo].[jobs]

WHERE [max_lvl] >= '200'

查詢 JOBS的max_lvl為小於"150" 或 min_lvl為大於"100"

SELECT [job_desc], [min_lvl],[max_lvl]

資料庫期末（作業版）

```
FROM [dbo].[jobs]

WHERE ((([job_desc] Like '%Manager%') AND ([min_lvl] <= '100'))
```

查詢1990年1月1日到1992年12月31日雇用的員工資料

```
SELECT [dbo].[employee].*
FROM [dbo].[employee]
WHERE ([hire_date] Between '1/1/1990' And '12/31/1992');
```

9. WHERE條件子句

WHERE條件的欄位如果是字串需要加上單引號或雙引號

日期/時間字串需要使用「#」符號括起

Like

<ex> 查詢產品名稱擁有子字串"起司"

```
SELECT * FROM 產品資訊
WHERE 產品 LIKE '*起司'
```

<ex> 查詢產品名稱是以任何字串開頭，結尾為"起司"

```
SELECT DISTINCT 產品資料.*
FROM 產品資料
WHERE ((([產品資料].[產品]) Like '*起司'));
```

<ex>查詢單位數量不是以子字串"12"

```
SELECT * FROM 產品資料
WHERE 單位數量 not LIKE '*12'
```

10. AND與OR多條件查詢

11. NOT運算子可以搭配前述子句，取得與條件相反的查詢結果 (優先順序以括號為優先)

查詢產品資訊的產品名稱有"Nano"子字串或入庫日期有"7"子字串，這2個條件使用括號括起，而且定價需要大於等於5000的記錄

資料庫期末（作業版）

SELECT * FROM 產品資訊

WHERE (產品名稱 LIKE '*Nano*' OR 入庫日期 LIKE '*7*') AND 定價>=5000

12. ORDER BY子句指定排序欄位，就可以由小到大進行排序

<ex>查詢定價大於等於3500元的記錄，並且使用定價欄位由小到大進行排序

SELECT * FROM 產品資訊

WHERE 定價>=3500 ORDER BY 定價 ASC

13. BETWEEN AND子句可以定義SELECT指令WHERE條件子句的範圍，範圍值可以使用文字、數值或日期/時間

<ex> 查詢1992年1月1日到12月31日雇用的員工資料

SELECT 員工.*

FROM 員工

WHERE (((員工.雇用日期) Between #1/1/1992# And #12/31/1992#));

14. IN運算子可以設定一串文字或數值清單

<ex>查詢單價(成本)為38, 43.9, 和123.79的產品資料

SELECT * FROM 產品資料

WHERE 單價 IN (38,43.9,123.79)

問題

【f】 產品單價有多少種?欄位不要重複

SELECT DISTINCT 產品資料.單價

FROM 產品資料

ORDER BY 產品資料.單價;

【g】 查詢所有客戶的名稱、聯絡人和電話資料，其中連絡人職稱為董事長或業務

SELECT 客戶.公司名稱, 客戶.連絡人, 客戶.電話

FROM 客戶

WHERE (((客戶.連絡人職稱)="董事長" Or (客戶.連絡人職稱)="業務"));

SELECT 客戶.公司名稱, 客戶.連絡人, 客戶.電話

FROM 客戶

WHERE (客戶.連絡人職稱) in ("董事長","業務");

【h】 查詢所有客戶的名稱、聯絡人和電話資料，其中公司名稱有”中”的資料

資料庫期末（作業版）

SELECT 客戶.公司名稱, 客戶.連絡人, 客戶.電話, 客戶.公司名稱

FROM 客戶

where 公司名稱 like "**中**"

聚合函數

【i】查詢貢獻最好的顧客

SELECT 客戶.公司名稱, Sum([訂貨明細].[單價]*[訂貨明細].[數量]) AS total

FROM (客戶 INNER JOIN 訂貨主檔 ON 客戶.客戶編號 = 訂貨主檔.客戶編號) INNER JOIN 訂貨明細 ON 訂貨主檔.訂單號碼 = 訂貨明細.訂單號碼

GROUP BY 客戶.公司名稱

ORDER BY Sum([訂貨明細].[單價]*[訂貨明細].[數量]) DESC;

【j】查詢販賣顧客所購買最多的產品

SELECT 客戶.公司名稱, 產品資料.產品, Sum([訂貨明細].[單價]*[訂貨明細].[數量]) AS total

FROM 產品資料 INNER JOIN ((客戶 INNER JOIN 訂貨主檔 ON 客戶.客戶編號 = 訂貨主檔.客戶編號) INNER JOIN 訂貨明細 ON 訂貨主檔.訂單號碼 = 訂貨明細.訂單號碼) ON 產品資料.產品編號 = 訂貨明細.產品編號

GROUP BY 客戶.公司名稱, 產品資料.產品

ORDER BY 客戶.公司名稱, Sum([訂貨明細].[單價]*[訂貨明細].[數量]) DESC;

【k】查詢販賣最好的產品

SELECT 產品資料.產品, Sum([訂貨明細].[單價]*[訂貨明細].[數量]) AS total

FROM 產品資料 INNER JOIN (訂貨主檔 INNER JOIN 訂貨明細 ON 訂貨主檔.訂單號碼 = 訂貨明細.訂單號碼) ON 產品資料.產品編號 = 訂貨明細.產品編號

GROUP BY 產品資料.產品

ORDER BY Sum([訂貨明細].[單價]*[訂貨明細].[數量]) DESC;

【l】1995/1/1~1995/5/31銷售最好產品

SELECT 產品資料.產品, Sum([訂貨明細].[單價]*[訂貨明細].[數量]) AS total

FROM 訂貨主檔 INNER JOIN (產品資料 INNER JOIN 訂貨明細 ON 產品資料.產品編號 = 訂貨明細.產品編號) ON 訂貨主檔.訂單號碼 = 訂貨明細.訂單號碼

WHERE (((訂貨主檔.訂單日期)>=#1/1/1995#) AND ((訂貨主檔.訂單日期)<=#5/31/1995#))

GROUP BY 產品資料.產品

ORDER BY Sum([訂貨明細].[單價]*[訂貨明細].[數量]) DESC;

資料庫期末（作業版）

【m】 查詢業績最好的員工

```
SELECT 員工.姓名, Sum([訂貨明細].[單價]*[訂貨明細].[數量]) AS total  
  
FROM 員工 INNER JOIN (訂貨主檔 INNER JOIN 訂貨明細 ON 訂貨主檔.訂單號碼 = 訂貨明細.訂單號碼) ON 員工.員工編號 = 訂貨主檔.員工編號  
  
GROUP BY 員工.姓名  
  
ORDER BY Sum([訂貨明細].[單價]*[訂貨明細].[數量]) DESC;
```

【n】 查詢販賣最好的類別

```
SELECT 產品類別.類別名稱, Sum([訂貨明細].[單價]*[訂貨明細].[數量]) AS total  
  
FROM 產品類別 INNER JOIN (產品資料 INNER JOIN 訂貨明細 ON 產品資料.產品編號 = 訂貨明細.產品編號) ON 產品類別.類別編號 = 產品資料.類別編號  
  
GROUP BY 產品類別.類別名稱  
  
ORDER BY Sum([訂貨明細].[單價]*[訂貨明細].[數量]) DESC;
```

【o】 查詢販賣最好產品的供應商

```
SELECT 供應商.供應商, Sum([訂貨明細].[單價]*[訂貨明細].[數量]) AS total  
  
FROM 供應商 INNER JOIN (產品資料 INNER JOIN 訂貨明細 ON 產品資料.產品編號 = 訂貨明細.產品編號) ON 供應商.供應商編號 = 產品資料.供應商編號  
  
GROUP BY 供應商.供應商  
  
ORDER BY Sum([訂貨明細].[單價]*[訂貨明細].[數量]) DESC;
```

【p】 查詢販賣最好產品的貨運公司

```
SELECT 貨運公司.貨運公司名稱, Sum([訂貨明細].[單價]*[訂貨明細].[數量]) AS total  
  
FROM (貨運公司 INNER JOIN 訂貨主檔 ON 貨運公司.貨運公司編號 = 訂貨主檔.送貨方式) INNER JOIN 訂貨明細 ON 訂貨主檔.訂單號碼 = 訂貨明細.訂單號碼  
  
GROUP BY 貨運公司.貨運公司名稱  
  
ORDER BY Sum([訂貨明細].[單價]*[訂貨明細].[數量]) DESC;
```

第三次練習

Create database

create database s3a632093

use s3a632093

Create table

本範例將示範已建立在s3a632093資料庫中的三個資料表 (job、employee及publishers)的完整資料表定義，包含所有的條件約束定義

建立jobs

CREATE TABLE jobs

資料庫期末（作業版）

```

(
    job_id smallint not null
        IDENTITY(1,1)
        PRIMARY KEY,
    job_desc varchar(50) NOT NULL
        DEFAULT 'New Position - title not formalized yet',
    min_lvl tinyint NOT NULL
        CHECK (min_lvl >= 10),
    max_lvl tinyint NOT NULL
        check (max_lvl >= 100)
)

-- 匯入資料
insert into publishers select *from pubs.. publishers

-- 建立employee
sp_addtype empid, 'char(9)', 'null'

CREATE TABLE employee
(
    emp_id empid
        CONSTRAINT PK_emp_id PRIMARY KEY
        CONSTRAINT CK_emp_id CHECK (emp_id LIKE
            '[A-Z][A-Z][A-Z][1-9][0-9][0-9][0-9][FM]' or
            emp_id LIKE '[A-Z]-[A-Z][1-9][0-9][0-9][0-9][FM]'),
    /* Each employee ID consists of three characters that
       represent the employee's initials, followed by a five
       digit number ranging from 10000 through 99999 and
       then the
       employee's gender (M or F). A (hyphen) - is acceptable
       for the middle initial. */
    fname varchar(20) NOT NULL,
    minit char(1) NULL,
    lname varchar(30) NOT NULL,
    job_id smallint NOT NULL
        DEFAULT 1
    /* Entry job_id for new hires. */
    REFERENCES jobs(job_id),
    job_lvl tinyint
        DEFAULT 10,
    /* Entry job_lvl for new hires. */
    pub_id char(4) NOT NULL
        DEFAULT ('9952')
)

```

匯入資料

輸入資料庫pubs中的jobs

```

set IDENTITY_INSERT jobs on

go

insert jobs (job_id, job_desc, min_lvl, max_lvl) select * from
pubs..jobs

```

建立publishers

```

CREATE TABLE publishers
(
    pub_id char(4) NOT NULL
        CONSTRAINT UPKCL_pubind PRIMARY KEY
        CHECK (pub_id IN ('1389', '0736', '0877', '1622',
            '1756')
            OR pub_id LIKE '99[0-9][0-9]'),
    pub_name varchar(40) NULL,
    city varchar(20) NULL,
    state char(2) NULL,
    country varchar(30) NULL
        DEFAULT('USA')
)

```


資料庫期末 (作業版)

```
REFERENCES publishers(pub_id),

/* By default, the Parent Company Publisher is the
company

to whom each employee reports. */

hire_date    datetime    NOT NULL

DEFAULT (getdate())

/* By default, the current system date is entered. */

)
```

匯入資料

```
insert into employee select *from pubs.. employee
```

在employee中練習輸入新進員工

手動鍵入

insert into輸入

```
insert into employee([emp_id],[fname],[lname]) values
('AAA54321M','王','翔')
```

建立計算欄位資料表

```
CREATE TABLE mytable
```

```
(
[low] int,
high int,
myavg AS ([low] + high)/2
)
```

建立計算欄位資料表

```
CREATE TABLE 估價
```

```
( 編號 int IDENTITY,
```

```
單價 money,
```

```
數量 int,
```

```
總價 AS 單價 * 數量
```

```
)
```

移除主鍵

```
ALTER TABLE [dbo].[publishers]
```

```
drop [UPKCL_pubind]
```

增加主鍵

```
ALTER TABLE dbo.publishers
```

```
ADD PRIMARY KEY (pub_id)
```

資料庫管理系統

第四次練習

班級:資一乙 學號:3a632093 姓名:黃名廷

缺少: 錯誤:

內容

建立外來鍵，例：employee中pub_id 4

改變資料型態 5

新增欄位 6

加入限制 7

刪除限制 8

刪除欄位 9

增加預設值 11

--試試預設 12

刪除預設 13

建立規則 14

--新增記錄 單筆輸入 整批輸入 14

--更新的欄位不只一個請使用逗號分隔 15

--將資料表符合條件的記錄刪除掉 16

Alter table

物件 動作 範例

主鍵 刪除 例1：publishers中pub_id

建立 例2：publishers中pub_id

資料庫期末 (作業版)

試試	刪除	例：jobs中job_id	FOREIGN KEY ([pub_id])
	建立	例：jobs中job_id	REFERENCES [dbo].[publishers]([pub_id])
外來鍵	刪除	例3：employee中pub_id	
	建立	例4：employee中pub_id	改變資料型態
試試	刪除	例：employee中job_id	例5：publishers中Country char(30) [?] char(20)
	建立	例：employee中job_id	ALTER TABLE [dbo].[publishers]
資料型態 改變		例5：publishers中Country char(30) [?] char(20)	ALTER COLUMN [country] varchar(20) NULL
試試	改變	例：jobs中job_desc (50) [?] char(40)	新增欄位
欄位	建立	例6：因為要報稅, employee 中增加一欄ssn char(10)	因為要報稅, employee 中增加一欄ssn char(10)
	加入限制	例7：增加ssn 身分證字號限制	ALTER TABLE [dbo].[employee]
	刪除限制	例8：刪除ssn 身分證字號限制	ADD ssn char(10) null
	刪除	例9：刪除ssn	加入限制
預設	建立	例10：估價缺少資料新增數量欄位的預設值為1	增加ssn 身分證字號限制
試試	預設	例：估價缺少資料新增估價日期的預設值為今天	ALTER TABLE [dbo].[employee]
	刪除	例11：刪除預設	add CONSTRAINT check_ssn
規則	建立	例12：增加數量>=1	CHECK (SSN like '[A-Z][12][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')
	刪除	例13	刪除限制
jobs中job_id			刪除ssn 身分證字號限制
			ALTER TABLE [dbo].[employee]
			drop CONSTRAINT [check_ssn]
修改前			刪除欄位
		ALTER TABLE [dbo].[jobs]	刪除ssn
		drop [PK__jobs__6E32B6A58D0E0F1C]	ALTER TABLE [dbo].[employee]
			drop column ssn
		ALTER TABLE [dbo].[jobs]	
		ADD PRIMARY KEY (job_id)	試試 建立 例6：employee 中增加一欄連絡人職稱 char(30)
建立外來鍵，例：employee中pub_id			ALTER TABLE [dbo].[employee]
ALTER TABLE [dbo].[employee]			ADD 連絡人職稱char(30) null
add CONSTRAINT fk_pubs_restrict			

資料庫期末 (作業版)

加入限制 例7：增加連絡人職稱限制 in (業務, 業務經理, 助理)

```
ALTER TABLE [dbo].[employee]
```

```
add CONSTRAINT check_連絡人職稱
```

```
CHECK (連絡人職稱 like '[業務],[業務經理],[助理]')
```

刪除限制 例8：刪除連絡人職稱限制

```
ALTER TABLE [dbo].[employee]
```

```
drop CONSTRAINT [check_連絡人職稱]
```

刪除 例9：刪除連絡人職稱

```
ALTER TABLE [dbo].[employee]
```

```
drop column 連絡人職稱
```

增加預設值

例10：估價缺少資料數量欄位的預設值為1

```
--估價缺少資料數量欄位的預設值為1
```

```
/* 10 */ALTER TABLE [dbo].[估價]
```

```
ADD CONSTRAINT default_qty
```

```
DEFAULT 1 FOR [數量]
```

```
--試試預設
```

例：估價缺少資料新增估價日期的預設值為今天

輸入資料試試

```
ALTER TABLE [dbo].[估價]
```

```
ADD 估價日期 datetime
```

```
DEFAULT (getdate ())
```

驗證

刪除預設

```
--刪除數量欄位的預設值為1 的預設
```

```
ALTER TABLE [dbo].[估價]
```

```
Drop CONSTRAINT default_qty
```

建立規則

```
--估價增加數量>=1
```

```
ALTER TABLE [dbo].[估價]
```

```
ADD CONSTRAINT rule_qty
```

```
check ([數量]>= 1)
```

```
--新增記錄 單筆輸入 整批輸入
```

```
INSERT INTO dbo.估價
```

```
VALUES (7,10,'2017/11/12')
```

```
--更新的欄位不只一個請使用逗號分隔
```

```
UPDATE dbo.估價 SET 數量=4, 估價日期= getdate()
```

```
where 數量 = 3
```

```
--將資料表符合條件的記錄刪除掉
```

```
DELETE FROM dbo.估價 WHERE 估價日期 is null
```