

## 資訊及通訊科技

### 試卷二

本試卷必須用中文作答

一小時三十分完卷 (上午十一時十五分至下午十二時四十五分)

#### 考生須知

- (一) 本試卷共有甲、乙和丙**三部**。考生須選答任何**兩部**中的**全部**試題。
- (二) 答案須寫在所提供的 **DSE(B)** 答題簿內，每題(非指分題)必須另起新頁作答。
- (三) 本試題答題簿末頁附有 SQL 指令及實體關係圖所採用的符號以供參考。

撥款資助  
FUNDED BY



創新計劃項目  
INNOVATIVE PROGRAMME



協創機構  
INTERMEDIARY



印刷贊助  
PRINTING SPONSOR



© 迴享有限公司 共享創意授權 (By 智乃、Willis)

Recurso Limited 2025

CC BY-NC 4.0

考試結束前不可  
將試卷攜離試場

## 甲部 數據庫

1. 一個大學學系正在建立一個數據庫來管理其開辦的課程。數據庫的要求如下：

每個課程必須由一位教授任教，但一位教授任教至少一個課程。有些課程不需要實驗課，而有些課程有多個實驗課。每個實驗課只與一個課程相關。

下圖顯示了該數據庫的部分實體關係圖。繪製此數據庫的實體關係圖。圖內不用畫上屬性。(4分)



2. 一間娛樂公司在香港舉辦演唱會。他們使用數據庫來管理門票銷售。目前，所有數據都存儲在名為 TICKET\_SALES 的數據表中。

TICKET\_SALES

欄位名稱	描述	例子
TicketID	門票編號	T2001
ConcertID	音樂會編號	C001
ConcertName	音樂會名稱	20週年香港巡迴演唱會
Date	音樂會舉辦日期	2025/10/07
VenueID	場地編號	V001
VenueAddress	場地位置	會議展覽中心
SeatNum	分配給門票的具體座位編號	A110
Performer	在音樂會表演的藝人名稱	48 組

下面草擬了符合第三範式的新數據庫模式。數據表 Ticket 儲存門票資料，而數據表 Concert 存儲每個演唱會的資料。

完成以下模式未填寫的部分。如果表格沒有任何外鍵碼，請填上「N/A」。 (5分)

Ticket (  )

外鍵碼：

Concert (  )

外鍵碼：

Venue (VenueID, VenueAddress)

外鍵碼：

3. 一所學校圖書館使用數據庫來管理書籍和借閱記錄。數據庫包含以下數據表：

LOAN

欄位名稱	數據類型	描述	例子
LoanID	字符	借閱記錄編號	L1005
BookID	字符	書籍編號	B0777
StudentID	字符	學生編號	S202501
LoanDate	日期	書籍借出日期	2025/12/01
DueDate	日期	書籍到期歸還日期	2025/12/15

為以下任務寫出 SQL 語句。

- (a) 列出學生編號為「S202501」且借閱了書籍編號為「B0777」的借閱記錄。(1分)
- (b) 為了改善題 (a) 中搜尋的效率，圖書館管理員建立了一個索引。由於圖書館管理員需要經常檢查特定學生的借閱記錄，利用數據表 LOAN 的 BookID 和 StudentID 建立一個索引。(2分)

4. 一所學校管理兩個課外活動學會的會員資格：棋藝學會和橋牌學會。數據庫包含兩個表格，CHESS 和 BRIDGE。

CHESS

StudentID	Name
S101	Amy
S102	Ben
S103	Cat
S106	Fay

BRIDGE

StudentID	Name
S102	Ben
S104	Dan
S105	Eva
S106	Fay

- (a) 考慮以下兩個 SQL 語句：

語句 1：

```
DELETE FROM CHESS
```

語句 2：

```
DROP TABLE CHESS
```

這兩個 SQL 語句有什麼分別？

(1 分)

- (b) 學校希望製作一份參與以上兩個學會的所有學生名單。

語句 3：

```
SELECT * FROM CHESS  
UNION  
SELECT * FROM BRIDGE
```

語句 4：

```
SELECT * FROM CHESS  
UNION ALL  
SELECT * FROM BRIDGE
```

這兩個 SQL 語句有什麼分別？

(1 分)

- (c) 寫出執行以下 SQL 語句後的輸出。

(2 分)

```
SELECT StudentID FROM CHESS  
MINUS  
SELECT StudentID FROM BRIDGE
```

5. 一間網上書店「BookWorld」使用數據庫來管理其銷售。數據庫包含以下表格：

CUSTOMER

欄位	數據類型	描述	例子
CustID	字符	客戶編號	C9001
CustName	字符	客戶名稱	大文
RegDate	日期	客戶註冊日期	2025/10/31

主關鍵碼： CustID

外鍵碼： N/A

PRODUCT

欄位	數據類型	描述	例子
ProdID	字符	產品編號	P001
ProdName	字符	書籍標題	Python 簡介
Category	字符	書籍類別	非小說
Price	數字	書籍價格	120

主關鍵碼： ProdID

外鍵碼： N/A

SALES

欄位	數據類型	描述	例子
SaleID	字符	銷售編號	S1001
CustID	字符	客戶編號	C9001
ProdID	字符	產品編號	P001
Quantity	整數	是次交易售出的數量	2

主關鍵碼： SalesID

外鍵碼： CustID, ProdID

為下列任務 (a) 至 (d) 寫出 SQL 指令。

- (a) 列出在 2023 年在「BookWorld」註冊的所有客戶名稱。 (2 分)
- (b) 列出客戶編號為「C1024」的客戶購買的所有書籍的產品名稱。 (2 分)
- (c) 列出從未售出過的產品名稱。 (3 分)
- (d) 銷售團隊希望了解每個書籍類別的受歡迎程度。對於每個書籍類別，列出類別名稱和售出的書籍總數。只包括售出書籍總數大於 500 的類別。 (3 分)
- (e) 簡略描述以下 SQL 語句的目的。 (2 分)

```
SELECT ProdName, Price
FROM PRODUCT P1
WHERE Price > (
    SELECT AVG(Price)
    FROM PRODUCT P2
    WHERE P2.Category = P1.Category)
```

- (f) 一位 CustID 為「C9002」的新客戶剛剛註冊。系統嘗試添加相關客戶的記錄 (語句 1) 和他們的購買記錄 (語句 2)。

語句 1:

```
INSERT INTO CUSTOMER
VALUES ('C9002', 'Chris Wong', '2025-12-08')
```

語句 2:

```
INSERT INTO SALES
VALUES ('S1211', 'C9002', 'P1023', 10)
```

以下哪個執行次序會導致數據完整性問題？簡略解釋。

次序 A:

語句 1 → 語句 2

次序 B:

語句 2 → 語句 1

(2 分)

**甲部完**

## 乙部 網絡應用程式開發

6. ABC 公司向客戶銷售五種類型的電器，包括冷氣機、風扇、智能手機、平板電腦和智能家居裝置。公司使用網頁讓客戶搜尋他們想要的產品。

草擬一個網頁，讓客戶無需輸入產品名稱即可搜尋產品。網頁中應顯示搜尋結果。搜尋結果應包含兩種類型的多媒體元素。如適合，請為你的設計加上註釋。(3 分)

7. 約翰使用 DHCP 伺服器分配 IP 位址給辦公室內的裝置。

(a) 舉出這種做法的一個優點和一個限制。(2 分)

(b) 除了 IP 位址外，寫出兩種 DHCP 伺服器會提供給裝置的資訊。(2 分)

8. 一個會議廳配備了一些無線網絡接達點 (無線 AP)，每個同時支援最多 50 個連接。

(a) 雖然 3 個 AP 已足夠應付超過 170 個同時連接，但網絡設計師安裝了總共 6 個 AP。舉出這種做法的一個優點。(1 分)

(b) 連接到 AP 時會涉及 WPA2。舉出 WPA2 的一個功能。(1 分)

(c) 為了提供即時視像串流，用戶可以使用 TCP 和 UDP。分別舉出使用 TCP 和 UDP 的一個優點。(2 分)

9. 寶琳建立了一個網頁，讓兒童學習不同的職業。

(a) 她在一張有人穿著不同制服的圖像上使用了影像地圖。

(i) 指出使用影像地圖的一個限制。(1 分)

(ii) JavaScript 可以達到同樣的目的嗎？簡略說明。(1 分)

(b) 寶琳可以購買伺服器來寄存網站，或訂閱網頁寄存服務。舉出每種做法的一個優點。(2 分)

10. 一家戲院正建立一個網站，允許顧客在線購買電影票。以下是檔案 index.php 的前 20 行：

行號	代碼
1	<!DOCTYPE html>
2	<html>
3	<head>
4	<style>
5	body {
6	margin: 0;
7	width: 65ch;
8	font-family: "Arial", sans-serif;
9	padding: 1rem;
10	}
11	h1 {
12	font-size: 16px;
13	}
14	</style>
15	<title>電影門票系統</title>
16	
17	<link href="/style.css" rel="stylesheet" />
18	</head>
19	<body>
20	<h1>電影門票系統</h1>
:	
:	

- (a) 簡略描述 index.php 第 1 行的目的。 (1 分)
- (b) 除了網頁的標題外，寫出另外**兩項**該公司可以包括在網頁中的元數據。 (2 分)
- (c) 雖然 index.php 的第 12 行將字體大小設置為 16px，但並非所有網頁瀏覽器都會以 16px 呈現 h1。寫出造成此現象的一個可能原因。 (1 分)

index.php 的第 21-58 行如下：

行號	代碼
21	<code>&lt;hr /&gt;</code>
22	<code>&lt;form action="/process.php" method="POST"&gt;</code>
23	<code>&lt;label for="customer-name"&gt;顧客名稱：&lt;/label&gt;</code>
24	<code>&lt;input name="customer_name"</code> <code>id="customerName" type="text" /&gt;</code>
25	
26	<code>&lt;label for="number-adult"&gt;成人數目：&lt;/label&gt;</code>
27	<code>&lt;input name="number-adult"</code> <code>id="numberOfAdult" type="text" min="0" max="10" /&gt;</code>
28	
29	<code>&lt;label for="number-concession"&gt;</code> <code>兒童／長者數目：&lt;/label&gt;</code>
30	<code>&lt;input name="number-concession" id="numberOfConcession"</code> <code>type="text" min="0" max="10" /&gt;</code>
31	
32	<code>&lt;hr /&gt;</code>
33	
34	<code>&lt;label for="movie-selection"&gt;Movie Name:&lt;/label&gt;</code>
35	<code>&lt;select name="movie" id="movieSelection"&gt;</code>
36	
-	(e) 部分
42	
43	<code>&lt;/select&gt;</code>
44	<code>&lt;div&gt;</code>
45	<code>&lt;p&gt;電影票的總數目：</code> <code>&lt;span id="totalTickets"&gt;0&lt;/span&gt;</code> <code>&lt;/p&gt;</code>
46	<code>&lt;p&gt;總價格： HK\$</code> <code>&lt;span id="totalPrice"&gt;0&lt;/span&gt;</code> <code>&lt;/p&gt;</code>
47	<code>&lt;/div&gt;</code>
48	
49	<code>&lt;hr /&gt;</code>
50	
51	<code>&lt;div&gt;</code>
52	<code>&lt;input type="submit" value="Buy Tickets" /&gt;</code>
53	<code>&lt;/div&gt;</code>
54	<code>&lt;/form&gt;</code>
55	
56	<code>&lt;script src="/main.js"&gt;&lt;/script&gt;</code>
57	<code>&lt;/body&gt;</code>
58	<code>&lt;/html&gt;</code>

(d) 修改第 27 行和第 30 行，使該欄位只能輸入數字。

(1 分)

(e) 在第 36-43 行，使用以下 php 手稿來填充下拉式選單。

行號	代碼
⋮	
36	<?php
37	include_once 'MOVIES.php';
38	
39	for (\$i = 0; \$i < count(\$MOVIES); \$i++) {
40	echo '<option value="' (e)(ii)(1) '">' (e)(ii)(2) '</option>';
41	}
42	?>
⋮	

檔案 MOVIES.php 包含以下 2 行代碼-

行號	代碼
1	<?php
2	\$MOVIES = ["咖啡的歷史", "城市中的商業", "混合的色彩"];

(i) 寫出使用 MOVIES.php 檔案來儲存電影名稱的一個優點。 (1 分)

(ii) 完成 index.php 的第 40 行，使第一部電影的值為 1 而不是 0。 (2 分)

(f) 檔案 main.js 包含以下數行代碼：

行號	代碼
1	function updateTotalTickets() {
2	const totalAdultTickets = Number(
3	document.getElementById("numberOfAdult").value
4	);
5	const totalConcessionTickets = Number(
6	document.getElementById("numberOfConcession").value
7	);
8	const totalTickets =
9	totalAdultTickets + totalConcessionTickets;
10	
11	document.getElementById("totalTickets").innerHTML =
	totalTickets;
12	document.getElementById("totalPrice").innerHTML = calPrice(
13	totalAdultTickets,
14	totalConcessionTickets
15	);
16	return;
17	}
18	
19	function calPrice(numberOfAdult, numberOfConcession) {
20	
21	
22	
23	
24	(f)(i) 部分
25	
26	return totalPrice;
27	}
28	
29	document.addEventListener("DOMContentLoaded", function () {
30	document
31	.getElementById("numberOfAdult")
32	.addEventListener(" (f)(ii) ", updateTotalTickets);
33	document
34	.getElementById("numberOfConcession")
35	.addEventListener(" (f)(ii) " , updateTotalTickets);
36	});

- (i) 子程式 calPrice() 用於計算需支付的票價總額。每張成人票價為 HK\$ 100，每張優惠票價為 \$50。參數 numberOfAdult 和 numberOfConcession 分別代表成人票和優惠票的數量。完成 calPrice()。 (2 分)
- (ii) 每當票數的輸入欄位更改時，都應呼叫函數 updateTotalTickets()。寫下 main.js 第 32 行和第 35 行的一個關鍵詞。 (1 分)
- (iii) 寫出一個使用客戶端手稿程式而非伺服器端手稿程式來實現 (ii) 中目的的原因。 (1 分)

檔案 process.php 用於處理從 index.php 接收到的數據。檔案 process.php 包含以下數行代碼：

行號	代碼
1	<?php
.	.
.	.
.	.
7	\$conn = mysqli_connect(\$host, \$username, \$password, \$db);
8	
9	if (!\$conn) {
10	die("Connection Error" . mysqli_error());
11	}
12	
13	\$customerName = \$_POST["customer-name"];
14	\$numAdu = \$_POST["number-adult"];
15	\$numCon = \$_POST["number-concession"];
16	\$movie = \$_POST["movie"];
17	
18	\$fields = [\$customerName, \$movie, \$numAdu, \$numCon];
19	
20	\$values = "CURRENT_TIMESTAMP, ";
21	
22	foreach ( <input type="text" value="(g)(i)(1)"/> ) {
23	if (\$field == \$fields[0]) {
24	\$values = \$values . "'" . \$field . "'" . ", ";
25	} else {
26	\$values = <input type="text" value="(g)(i)(2)"/> );
27	}
28	}
29	
30	\$values = substr(\$values, 0, -2);
31	
32	\$sql = "INSERT INTO PURCHASES (purchaseDT, name, movie, numAdult, numConcession) VALUES (" . \$values . ")";
33	
34	mysqli_query(\$conn, \$sql);
35	
36	mysqli_close(\$conn);
37	
38	header("Location: /success.php?name=" . \$customerName);
39	?>

(g) 考慮 process.php 的內容：

- (i) 第 22-28 行的 foreach 迴圈將 \$fields 的每個值附加到字串 \$values 中以供插入。分別完成第 22 行和第 26 行。 (2 分)
- (ii) process.php 第 38 行的功能是什麼？ (1 分)

乙部完

**丙部 算法與程式編寫**

11. S1 是一個大小為 8 的堆疊。以下子程式用於操作該堆疊-

子程式	描述
len(S)	傳回堆疊 S 中的元素數量
push(S, el)	將元素 el 新增至堆疊 S
pop(S)	傳回並移除堆疊 S 中最頂端的元素
rev(S)	反轉堆疊 S 中的元素

(a) 以偽代碼編寫子程式 isFull(S)。如果堆疊已滿，傳回 True，相反則傳回 False。  
(1 分)

(b) 考慮以下 rev(S) 的設計，其中 A 是一個索引由 0 開始、長度為 8 的陣列。

```

rev(S)
    設 i 由 0 至 7 執行
        A[i] ← pop(S)
    設 i 由 0 至 7 執行
        push(S, A[i])
    
```

這個設計有一個錯誤。簡略描述該錯誤及並描述相應的解決方案。  
(2 分)

12. 以下算法的偽代碼表示對一個索引由 0 開始的整數陣列 A 執行的插入排序。陣列 A 中有 N 個元素。

```

i ← 0
重覆
    (a)(1)
    當 j > 0 AND (a)(2) 執行
        T ← A[j]
        A[j] ← A[j - 1]
        A[j - 1] ← T
        j ← j - 1
    i ← i + 1
直至 (a)(3)
    
```

(a) 完成上述算法以將 A 按升序排列。  
(3 分)

(b) 寫下一種當 N 很大時，執行效率比插入排序更高的排序算法名稱。  
(1 分)

13. 7 人參加了一個速配約會節目。節目結束後，每位參與者需要對其他參與者進行評分，分數範圍為 1 至 7 分。

二維陣列  $sc$  儲存節目後的評分。在以下例子中， $sc[i, j]$  指的是第  $i$  個人給予第  $j$  個人的評分。

$i \backslash j$	0	1	2	3	4	5	6
0	-1	2	1	1	3	6	7
1	3	-1	7	6	4	2	4
2	5	3	-1	3	7	7	6
3	6	7	1	-1	2	3	4
4	1	2	6	5	-1	3	1
5	1	4	2	6	5	-1	6
6	6	3	7	3	1	2	-1

- (a) 第 5 個人給第 0 個人的評分是多少？ (1 分)
- (b) 子程式 `star(per)` 顯示索引為 `per` 的人的平均分，以及他/她是否為明星表現者，即他/她收到 3 個或以上 5 分或以上的評分。

例如：

	顯示	描述
<code>star(0)</code>	0*: 3.67	0 旁邊的星號 (*) 表示有至少 3 人或以上給予他/她至少 5 分
<code>star(1)</code>	1: 3.50	

請以 Python/C++/Pascal 完成子程式 `star(per)`。請在答題紙上註明你選擇的程式語言。 (2 分)

(c) 子程式 `pair(i, j)` 的偽代碼表示如下—

```
pair(i, j)
    如果 i = j 則 傳回 -1
    傳回 SC[i, j] + SC[j, i]
```

速配約會節目的經理編寫了以下程式，以找出得分最高的一對參與者—

```
10    bi ← 0
20    bj ← 0
30    設 i 由 0 至 6 執行
40        設 j 由 0 至 6 執行
50            如果 pair(bi, bj) < pair(i, j) 則
60                bi ← i
70                bj ← j
80    輸出 bi, bj
```

(i) 執行上述程式後，`bi` 必須小於 `bj`。為什麼？ (1 分)

(ii) 修改第 40 行以減少程式的迭代次數。 (1 分)

14. 為了控制一個機械人，使用了用戶自訂的數據類型 INS 來提供方向和移動的單位數目。INS 中有兩個欄位：

欄位	描述
dir	代表移動方向的字符。可以是 'E' (東) 或 'N' (北)
dis	此整數代表移動的單位數目

陣列 InsArr 儲存若干個 INS 類型的指令。假設機械人總是從 (0, 0) 開始。以下是 InsArr 的內容示例，以及執行這些指令後的相應結果。

i	0	1	2	3
InsArr[i]	{dir: 'E', dis: 2}	{dir: 'N', dis: 1}	{dir: 'E', dis: -1}	{dir: 'N', dis: 2}

5						
4						
3		★				
2						
1		☆	☆			
0	★		☆			
y/x	0	1	2	3	4	5

- ☆ 代表中間步驟  
★ 代表起點和終點位置

在上述例子中，起點位置和終點位置之間的距離是  $\sqrt{1^2 + 3^2} = \sqrt{10}$ 。

- (a) 假設 InsArr 中有 C 個指令。完成以下用以找出起點位置與終點位置之間距離算法的偽代碼。(2 分)

```

CalcDist()
  dx ← 0
  dy ← 0
  設 i 由 0 至 C - 1 執行
    如果 (a)(1) 則
      dx ← dx + InsArr[i].dis
    否則
      dy ← (a)(2)
  傳回 (dx ** 2 + dy ** 2) 的開方根

```

- (b) 描述一個可以安裝在機械人上的傳感器，以檢測路徑上的潛在障礙物。(1 分)

15. ABC 鎮正在進行投票，以決定區內最好的餐廳。每張選票包括餐廳名稱和一個 0.0 至 5.0 (包括頭尾) 之間的分數。陣列  $N$  和  $S$  用於儲存名稱和對應的分數。 $N$  和  $S$  的首 7 個元素如下：

$i$	0	1	2	3	4	5	6
$N[i]$	WCD	WCD	KFD	SAD	SAD	KFD	WCD
$S[i]$	3.2	4.0	4.5	1.2	3.7	4.1	3.7

- (a) 根據  $N$  和  $S$  的首 7 個元素，餐廳「KFD」的平均分是多少？ (1 分)

假設總共有  $C$  張選票。

- (b) 請以 Python、C++ 或 Pascal 編寫一個程式，以輸出擁有最高評分的選票數量，並在答題紙上註明你所採用的程式編寫語言。 (3 分)

- (c) 以下的子程式 `getUnique()` 傳回一個新陣列  $NX$ ，其中僅儲存  $N$  中的名稱各一次 (即不重複)。

```

getUnique()
10  size ← 1
20  NX[0] ← N[0]
30  設  $i$  由 0 至  $C - 1$  執行
40  如果 NOT exists(NX, N[i], size) 則
50      size ← size + 1
60      NX 的大小調整為 size
70      NX[i] ← N[i]
80  傳回 NX

```

- (i) 子程式 `exists(arr, el, size)` 在 `el` 存在於 `arr` 時傳回 `True`。否則，它傳回 `False`。以偽代碼編寫 `exists(arr, el, size)`。 (3 分)
- (ii) 在 `getUnique()` 的第 40 行之後有一個錯誤。簡略解釋該錯誤並提出相應的更正。 (2 分)
- (iii) 假設  $N$  已按升序排列。簡略描述對 `getUnique()` 的更改以提高效率。 (2 分)
- (d) 子程式 `getAverage(T)` 傳回名為  $T$  的餐廳的平均評分。請以 Python、C++ 或 Pascal 編寫 `getAverage(T)`。並在答題紙上註明你所採用的程式編寫語言。 (2 分)
- (e) 在找出每間餐廳的平均評分後，此結果以鏈表的形式儲存。舉出使用鏈表的一個優點和一個缺點。 (2 分)

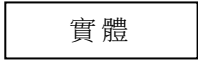






**丙部完**

**試卷完**

**數據庫 (SQL指令—建基於 SQL-92 標準)**

常數	FALSE, TRUE
運算符	+, -, *, /, >, <, =, >=, <=, <>, %, _ , ' , AND, NOT, OR
SQL	ABSOLUTE (ABS), AVG, INT, MAX, MIN, SUM, COUNT ASC, AT, CHAR (CHR), CHAR_LENGTH (LEN), LOWER, TRIM, SPACE, SUBSTRING (SUBSTR/MID), UPPER, VALUE (VAL) DATE, DAY, MONTH, YEAR ADD, ALL, ALTER, ANY, AS, ASC, BETWEEN, BY, CREATE, DELETE, DESC, DISTINCT, DROP, EXISTS, FROM, GROUP, HAVING, IN, INDEX, INNER JOIN, INSERT, INTEGER, INTERSECT, INTO, LEFT [OUTER] JOIN, LIKE, MINUS, NULL, RIGHT [OUTER] JOIN, FULL [OUTER] JOIN, ON, ORDER, SELECT, SET, TABLE, TO, UNION, UNIQUE, UPDATE, VALUES, VIEW, WHERE

**實體關係圖所採用的符號**

意思	符號	意思	符號
實體		一對一關係	
屬性		一對多關係	
主要屬性		多對多關係	
關係		參與限制： 在強制參與一面用   在選擇性參與一面用 ○	