**您好：**

首先再次感謝 **您**對本司開放職缺的支持，為能節省 **您**寶貴的求職時間，並讓 **您**在未來前來本司合作後，原有工作夥伴對 **您**的專業能力有一定的認識，我們準備了後頁的問卷。

麻煩 **您**針對我們所提出的問題及要求，以 **您**的專業知識，在3工作日內以 email 提供所有回應 (請寄到 [hr@ebti.com.tw](mailto:hr@ebti.com.tw)) 。萬一，我們所撰寫的問卷題目正好與 您的專精項目不同，部分問題不寫也是可以的，盡 您所長即可。如果 **您**願意花些時間在網路上搜尋答案，我們也欣然接受，畢竟網路大神可以解決的，未來在工作上也就可以解決，請 **您**不必介意解決這些問題的管道。

我們將在收到  **您**的回應後，立即與 **您**約定面談的時間及方式。原則上，面談方式將視疫情狀況，採取線上會議 (Google Meet) 或邀請 **您**前來本公司當面協談等方式。

對於我司冒昧請求 **您**協助填寫這份問卷，若 **您**覺得被冒犯，請千萬見諒。 我們並不是對 **您**的專業能理有所質疑，只是希望能拉近 **您**及我們之間對於工作內容的認知。

以上說明。

敬祝 **您**

**求職順利、前程遠大**

**智網通科技股份有限公司**

**( PS:若 您有任何疑慮或疑問，都歡迎 您以** [**hr@ebti.com.tw**](mailto:hr@ebti.com.tw) **統一聯絡窗口來告訴我們)**

1. **網頁設計**

**Q1 請用HTML 5與 CSS 3 製作如下圖頁面**



**可以使用 jQuery、Bootstrap等套件或框架，其資源使用CDN即可**

**分類的下拉選項包含：全部、最新資訊、中獎公告、優惠活動**

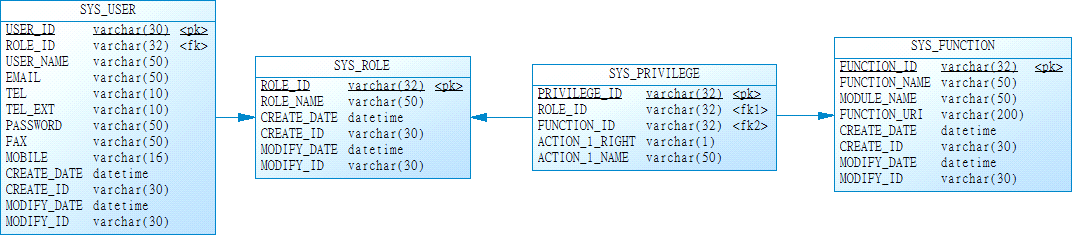
**狀態的下拉選項包含：全部、發布、未發布**

請提供製作的Q1.html檔，若有樣式有獨立製作，請提供Q1.css檔

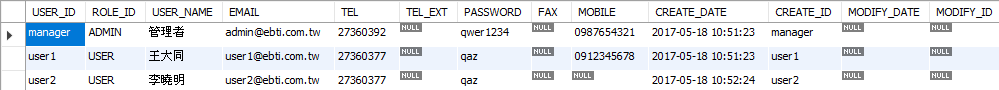
1. **資料庫**

**整體說明：**

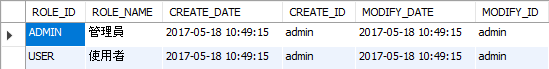
圖片中有四個資料表，分別為：使用者(SYS\_USER)、角色(SYS\_ROLE)、權限(SYS\_PRIVLEGE)、功能(SYS\_FUNCTION)，它們之間的關聯如下圖所示：



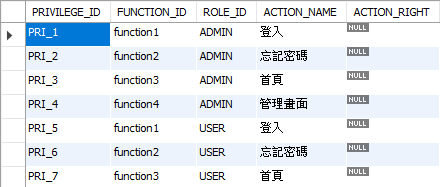
使用者(SYS\_USER)資料如下：



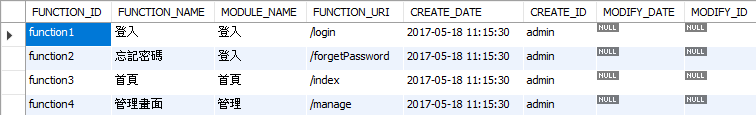
角色(SYS\_ROLE)資料如下：



權限(SYS\_PRIVLEGE)資料如下：



功能(SYS\_FUNCTION)資料如下：



**Q2. 請以標準SQL語法撰寫出以下描述：列出使用者(SYS\_USER)所擁有的功能(SYS\_FUNCTION)，且該使用者(SYS\_USER)的USER\_ID為"manager"。**

|  |
| --- |
| 請將SQL語法寫在此處，並實際執行該SQL提供執行結果附件圖檔Q2.jpg。  SELECT sys\_user.USER\_ID,sys\_function.MODULE\_NAME,sys\_function.FUNCTION\_NAME  FROM sys\_user  LEFT JOIN sys\_privilege ON sys\_user.ROLE\_ID = sys\_privilege.ROLE\_ID  LEFT JOIN sys\_function ON sys\_privilege.FUNCTION\_ID = sys\_function.FUNCTION\_ID  WHERE USER\_ID ="manager";  下方表格Q2\_init\_sql.txt內的SQL提供　您初始化上述資料表。此SQL為MySQL的語法，有可能與　您所使用的SQL引擎有相容性問題，請依據　您實際使用的SQL引擎來調整SQL。 |

| **Q2\_init\_sql.txt** |
| --- |
| /\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;  /\*!40101 SET NAMES utf8 \*/;  /\*!50503 SET NAMES utf8mb4 \*/;  /\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;  /\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;  CREATE TABLE IF NOT EXISTS `sys\_function` (  `FUNCTION\_ID` varchar(32) CHARACTER SET utf8 NOT NULL,  `FUNCTION\_NAME` varchar(50) COLLATE utf8\_unicode\_ci DEFAULT NULL,  `MODULE\_NAME` varchar(50) COLLATE utf8\_unicode\_ci DEFAULT NULL,  `FUNCTION\_URI` varchar(200) COLLATE utf8\_unicode\_ci DEFAULT NULL,  `CREATE\_DATE` datetime DEFAULT NULL,  `CREATE\_ID` varchar(30) COLLATE utf8\_unicode\_ci DEFAULT NULL,  `MODIFY\_DATE` datetime DEFAULT NULL,  `MODIFY\_ID` varchar(30) COLLATE utf8\_unicode\_ci DEFAULT NULL,  PRIMARY KEY (`FUNCTION\_ID`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;  /\*!40000 ALTER TABLE `sys\_function` DISABLE KEYS \*/;  INSERT INTO `sys\_function` (`FUNCTION\_ID`, `FUNCTION\_NAME`, `MODULE\_NAME`, `FUNCTION\_URI`, `CREATE\_DATE`, `CREATE\_ID`, `MODIFY\_DATE`, `MODIFY\_ID`) VALUES  ('function1', '登入', '登入', '/login', '2017-05-18 11:15:30', 'admin', NULL, NULL),  ('function2', '忘記密碼', '登入', '/forgetPassword', '2017-05-18 11:15:30', 'admin', NULL, NULL),  ('function3', '首頁', '首頁', '/index', '2017-05-18 11:15:30', 'admin', NULL, NULL),  ('function4', '管理畫面', '管理', '/manage', '2017-05-18 11:15:30', 'admin', NULL, NULL);  /\*!40000 ALTER TABLE `sys\_function` ENABLE KEYS \*/;  CREATE TABLE IF NOT EXISTS `sys\_privilege` (  `PRIVILEGE\_ID` varchar(32) CHARACTER SET utf8 NOT NULL,  `FUNCTION\_ID` varchar(32) CHARACTER SET utf8 DEFAULT NULL,  `ROLE\_ID` varchar(32) CHARACTER SET utf8 DEFAULT NULL,  `ACTION\_NAME` varchar(45) COLLATE utf8\_unicode\_ci DEFAULT NULL,  `ACTION\_RIGHT` varchar(45) COLLATE utf8\_unicode\_ci DEFAULT NULL,  PRIMARY KEY (`PRIVILEGE\_ID`),  KEY `SYS\_ROLE\_FK\_idx` (`ROLE\_ID`),  KEY `SYS\_FUNCTION\_FK\_idx` (`FUNCTION\_ID`),  CONSTRAINT `SYS\_PRIVILEGE\_FK1` FOREIGN KEY (`ROLE\_ID`) REFERENCES `sys\_role` (`ROLE\_ID`) ON DELETE NO ACTION ON UPDATE NO ACTION,  CONSTRAINT `SYS\_PRIVILEGE\_FK2` FOREIGN KEY (`FUNCTION\_ID`) REFERENCES `sys\_function` (`FUNCTION\_ID`) ON DELETE NO ACTION ON UPDATE NO ACTION  ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;  /\*!40000 ALTER TABLE `sys\_privilege` DISABLE KEYS \*/;  INSERT INTO `sys\_privilege` (`PRIVILEGE\_ID`, `FUNCTION\_ID`, `ROLE\_ID`, `ACTION\_NAME`, `ACTION\_RIGHT`) VALUES  ('PRI\_1', 'function1', 'ADMIN', '登入', NULL),  ('PRI\_2', 'function2', 'ADMIN', '忘記密碼', NULL),  ('PRI\_3', 'function3', 'ADMIN', '首頁', NULL),  ('PRI\_4', 'function4', 'ADMIN', '管理畫面', NULL),  ('PRI\_5', 'function1', 'USER', '登入', NULL),  ('PRI\_6', 'function2', 'USER', '忘記密碼', NULL),  ('PRI\_7', 'function3', 'USER', '首頁', NULL);  /\*!40000 ALTER TABLE `sys\_privilege` ENABLE KEYS \*/;  CREATE TABLE IF NOT EXISTS `sys\_role` (  `ROLE\_ID` varchar(32) CHARACTER SET utf8 NOT NULL,  `ROLE\_NAME` varchar(50) COLLATE utf8\_unicode\_ci DEFAULT NULL,  `CREATE\_DATE` datetime DEFAULT NULL,  `CREATE\_ID` varchar(30) COLLATE utf8\_unicode\_ci DEFAULT NULL,  `MODIFY\_DATE` datetime DEFAULT NULL,  `MODIFY\_ID` varchar(30) COLLATE utf8\_unicode\_ci DEFAULT NULL,  PRIMARY KEY (`ROLE\_ID`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;  /\*!40000 ALTER TABLE `sys\_role` DISABLE KEYS \*/;  INSERT INTO `sys\_role` (`ROLE\_ID`, `ROLE\_NAME`, `CREATE\_DATE`, `CREATE\_ID`, `MODIFY\_DATE`, `MODIFY\_ID`) VALUES  ('ADMIN', '管理員', '2017-05-18 10:49:15', 'admin', '2017-05-18 10:49:15', 'admin'),  ('USER', '使用者', '2017-05-18 10:49:15', 'admin', '2017-05-18 10:49:15', 'admin');  /\*!40000 ALTER TABLE `sys\_role` ENABLE KEYS \*/;  CREATE TABLE IF NOT EXISTS `sys\_user` (  `USER\_ID` varchar(32) CHARACTER SET utf8 NOT NULL,  `ROLE\_ID` varchar(32) CHARACTER SET utf8 DEFAULT NULL,  `USER\_NAME` varchar(50) COLLATE utf8\_unicode\_ci DEFAULT NULL,  `EMAIL` varchar(50) COLLATE utf8\_unicode\_ci DEFAULT NULL,  `TEL` varchar(10) COLLATE utf8\_unicode\_ci DEFAULT NULL,  `TEL\_EXT` varchar(10) COLLATE utf8\_unicode\_ci DEFAULT NULL,  `PASSWORD` varchar(50) COLLATE utf8\_unicode\_ci DEFAULT NULL,  `FAX` varchar(50) COLLATE utf8\_unicode\_ci DEFAULT NULL,  `MOBILE` varchar(16) COLLATE utf8\_unicode\_ci DEFAULT NULL,  `CREATE\_DATE` datetime DEFAULT NULL,  `CREATE\_ID` varchar(30) COLLATE utf8\_unicode\_ci DEFAULT NULL,  `MODIFY\_DATE` datetime DEFAULT NULL,  `MODIFY\_ID` varchar(30) COLLATE utf8\_unicode\_ci DEFAULT NULL,  PRIMARY KEY (`USER\_ID`),  KEY `SYS\_ROLE\_ID\_idx` (`ROLE\_ID`),  CONSTRAINT `SYS\_ROLE\_FK` FOREIGN KEY (`ROLE\_ID`) REFERENCES `sys\_role` (`ROLE\_ID`) ON DELETE NO ACTION ON UPDATE NO ACTION  ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;  /\*!40000 ALTER TABLE `sys\_user` DISABLE KEYS \*/;  INSERT INTO `sys\_user` (`USER\_ID`, `ROLE\_ID`, `USER\_NAME`, `EMAIL`, `TEL`, `TEL\_EXT`, `PASSWORD`, `FAX`, `MOBILE`, `CREATE\_DATE`, `CREATE\_ID`, `MODIFY\_DATE`, `MODIFY\_ID`) VALUES  ('manager', 'ADMIN', '管理者', 'admin@ebti.com.tw', '27360392', NULL, 'qwer1234', NULL, '0987654321', '2017-05-18 10:51:23', 'manager', NULL, NULL),  ('user1', 'USER', '王大同', 'user1@ebti.com.tw', '27360377', NULL, 'qaz', NULL, '0912345678', '2017-05-18 10:51:23', 'user1', NULL, NULL),  ('user2', 'USER', '李曉明', 'user2@ebti.com.tw', '27360377', NULL, 'qaz', NULL, NULL, '2017-05-18 10:52:24', 'user2', NULL, NULL);  /\*!40000 ALTER TABLE `sys\_user` ENABLE KEYS \*/;  /\*!40101 SET SQL\_MODE=IFNULL(@OLD\_SQL\_MODE, '') \*/;  /\*!40014 SET FOREIGN\_KEY\_CHECKS=IF(@OLD\_FOREIGN\_KEY\_CHECKS IS NULL, 1, @OLD\_FOREIGN\_KEY\_CHECKS) \*/;  /\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/; |

**Q3. 承上題(Q2)資料，請再增加使用者分數(USER\_SCORE)資料表，並在其中增加資料如下圖。請以標準SQL語法撰寫出以下描述：列出使用者(SYS\_USER)王大同及李曉明的各科(CLASS)平均與加總分數(SCORE)。**



|  |
| --- |
| 請將SQL語法寫在此處，並實際執行該SQL提供執行結果附件圖檔Q3.jpg。  下方表格Q3\_init\_sql.txt內的SQL提供　您初始化上述資料表。此SQL為MySQL的語法，有可能與　您所使用的SQL引擎有相容性問題，請依據　您實際使用的SQL引擎來調整SQL。  SELECT CREATE\_ID,sum(SCORE) as 加總,avg(SCORE) 平均  FROM pre\_test.user\_score  GROUP BY CREATE\_ID; |

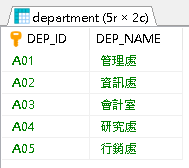
| **Q3\_init\_sql.txt** |
| --- |
| /\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;  /\*!40101 SET NAMES utf8 \*/;  /\*!50503 SET NAMES utf8mb4 \*/;  /\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;  /\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;  CREATE TABLE IF NOT EXISTS `user\_score` (  `SCORE\_ID` int(11) NOT NULL AUTO\_INCREMENT,  `CREATE\_DATE` datetime DEFAULT NULL,  `CREATE\_ID` varchar(50) COLLATE utf8\_unicode\_ci DEFAULT NULL,  `CLASS` varchar(50) COLLATE utf8\_unicode\_ci DEFAULT NULL,  `SCORE` int(11) DEFAULT NULL,  PRIMARY KEY (`SCORE\_ID`)  ) ENGINE=InnoDB AUTO\_INCREMENT=19 DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;  /\*!40000 ALTER TABLE `user\_score` DISABLE KEYS \*/;  INSERT INTO `user\_score` (`SCORE\_ID`, `CREATE\_DATE`, `CREATE\_ID`, `CLASS`, `SCORE`) VALUES  (1, '2021-01-10 11:30:00', 'user1', '國文', 76),  (2, '2021-01-10 11:30:00', 'user1', '英文', 58),  (3, '2021-01-10 11:30:00', 'user1', '數學', 81),  (4, '2021-02-10 11:30:00', 'user1', '國文', 66),  (5, '2021-02-10 11:30:00', 'user1', '英文', 60),  (6, '2021-02-10 11:30:00', 'user1', '數學', 74),  (7, '2021-03-10 11:30:00', 'user1', '國文', 59),  (8, '2021-03-10 11:30:00', 'user1', '英文', 57),  (9, '2021-03-10 11:30:00', 'user1', '數學', 69),  (10, '2021-01-10 11:30:00', 'user2', '國文', 80),  (11, '2021-01-10 11:30:00', 'user2', '英文', 61),  (12, '2021-01-10 11:30:00', 'user2', '數學', 58),  (13, '2021-02-10 11:30:00', 'user2', '國文', 81),  (14, '2021-02-10 11:30:00', 'user2', '英文', 56),  (15, '2021-02-10 11:30:00', 'user2', '數學', 59),  (16, '2021-03-10 11:30:00', 'user2', '國文', 90),  (17, '2021-03-10 11:30:00', 'user2', '英文', 55),  (18, '2021-03-10 11:30:00', 'user2', '數學', 54);  /\*!40000 ALTER TABLE `user\_score` ENABLE KEYS \*/;  /\*!40101 SET SQL\_MODE=IFNULL(@OLD\_SQL\_MODE, '') \*/;  /\*!40014 SET FOREIGN\_KEY\_CHECKS=IF(@OLD\_FOREIGN\_KEY\_CHECKS IS NULL, 1, @OLD\_FOREIGN\_KEY\_CHECKS) \*/;  /\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/; |

**Q4. 請以標準SQL語法撰寫出以下描述：列出使用者分數(USER\_SCORE)時，增加一個欄位來判斷分數(SCORE)是否未滿60分，若未滿60分則顯示"不及格"，若60分以上則顯示"及格"。**

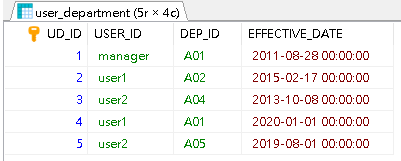
|  |
| --- |
| 請將SQL語法寫在此處，並實際執行該SQL提供執行結果附件圖檔Q4.jpg。  **SELECT \*,if(SCORE>=60,'及格','不及格') '及格' FROM pre\_test.user\_score ;** |

**Q5. 承Q2資料，請再增加部門(department)、使用者部門(user\_department)資料表，並在其中增加資料如下圖。請以標準SQL語法撰寫出以下描述：請列出在2020/10/01查詢時，所有使用者(SYS\_USER)帳號與姓名，以及當下所在的部門名稱。**

部門(department)資料如下：



使用者部門(user\_department)資料如下：



|  |
| --- |
| 請將SQL語法寫在此處，並實際執行該SQL提供執行結果附件圖檔Q5.jpg。  下方表格Q5\_init\_sql.txt內的SQL提供　您初始化上述資料表。此SQL為MySQL的語法，有可能與　您所使用的SQL引擎有相容性問題，請依據　您實際使用的SQL引擎來調整SQL。  SELECT s.USER\_NAME,s.EMAIL,d.DEP\_NAME,MAX(u.EFFECTIVE\_DATE) EFFECTIVE\_DATE FROM pre\_test.user\_department u LEFT JOIN sys\_user s ON u.USER\_ID = s.USER\_ID  LEFT JOIN department d ON d.DEP\_ID = u.DEP\_ID GROUP BY u.USER\_ID |

| **Q5\_init\_sql.txt** |
| --- |
| /\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;  /\*!40101 SET NAMES utf8 \*/;  /\*!50503 SET NAMES utf8mb4 \*/;  /\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;  /\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;  CREATE TABLE IF NOT EXISTS `department` (  `DEP\_ID` varchar(10) COLLATE utf8\_unicode\_ci NOT NULL,  `DEP\_NAME` varchar(50) COLLATE utf8\_unicode\_ci NOT NULL,  PRIMARY KEY (`DEP\_ID`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;  /\*!40000 ALTER TABLE `department` DISABLE KEYS \*/;  INSERT INTO `department` (`DEP\_ID`, `DEP\_NAME`) VALUES  ('A01', '管理處'),  ('A02', '資訊處'),  ('A03', '會計室'),  ('A04', '研究處'),  ('A05', '行銷處');  /\*!40000 ALTER TABLE `department` ENABLE KEYS \*/;  CREATE TABLE IF NOT EXISTS `user\_department` (  `UD\_ID` int(11) NOT NULL AUTO\_INCREMENT,  `USER\_ID` varchar(32) COLLATE utf8\_unicode\_ci NOT NULL,  `DEP\_ID` varchar(10) COLLATE utf8\_unicode\_ci NOT NULL,  `EFFECTIVE\_DATE` datetime NOT NULL,  PRIMARY KEY (`UD\_ID`)  ) ENGINE=InnoDB AUTO\_INCREMENT=6 DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;  /\*!40000 ALTER TABLE `user\_department` DISABLE KEYS \*/;  INSERT INTO `user\_department` (`UD\_ID`, `USER\_ID`, `DEP\_ID`, `EFFECTIVE\_DATE`) VALUES  (1, 'manager', 'A01', '2011-08-28 00:00:00'),  (2, 'user1', 'A02', '2015-02-17 00:00:00'),  (3, 'user2', 'A04', '2013-10-08 00:00:00'),  (4, 'user1', 'A01', '2020-01-01 00:00:00'),  (5, 'user2', 'A05', '2019-08-01 00:00:00');  /\*!40000 ALTER TABLE `user\_department` ENABLE KEYS \*/;  /\*!40101 SET SQL\_MODE=IFNULL(@OLD\_SQL\_MODE, '') \*/;  /\*!40014 SET FOREIGN\_KEY\_CHECKS=IF(@OLD\_FOREIGN\_KEY\_CHECKS IS NULL, 1, @OLD\_FOREIGN\_KEY\_CHECKS) \*/;  /\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/; |

1. **程式設計**

**整體說明：**以下問卷請　您以JAVA或C#其中之一程式語言來撰寫並執行。

**Q6: 請從十二生肖中隨機挑一個生肖後，再從十二星座中隨機挑一個星座形成一對結果(生肖, 星座)，重複以上步驟1000次產生出1000對結果，計算出這1000對結果的重複次數，次數由高到低列出。例：(猴, 魔羯座): 98, (雞, 牡羊座): 77, (牛, 獅子座): 61……**

請提供內含main與相關實作方法的Q6.java或Q6.cs檔，並實際執行該程式提供執行結果附件圖檔Q6.jpg。

**Q7. 請用Web開發框架來寫出簡單的留言版功能，如以下情境描述：**

**上方為留言輸入表單，下方為留言顯示區域。**

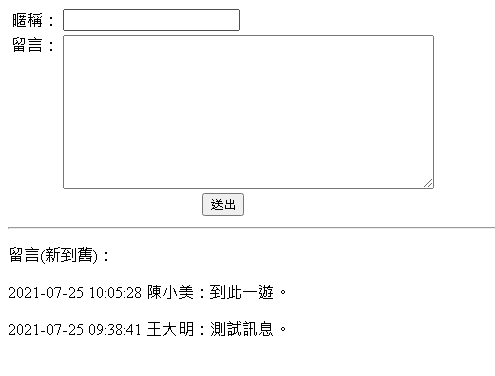
**表單可輸入暱稱及留言，輸入完畢後，按下送出，將於下方留言顯示區域顯示留言紀錄(時間、暱稱及留言)。**

**留言紀錄排序由新到舊，最新的呈現在最上方。**

**※註：留言資料可用List儲存不需要存入資料庫。**

**※註：Web開發框架，如：寫Java可用Servlet、Spring MVC；寫C#可用ASP.NET。**

**示意圖：**



請提供開發的專案檔(含原始碼)，並實際執行該程式提供執行結果附件圖檔Q7.jpg。

GitHub:https://github.com/day20180721/pre\_test\_comment/tree/master/comment/src/test/java/com/littlejenny/comment

1. **作業系統**

**Q8: 請以Linux shell script撰寫出以下模擬情境：**

**切換目錄至的/tmp目錄，壓縮/tmp目錄內所有副檔名為log(.log)的檔案，並於畫面顯示syslog.log檔中所有有包含"root"字串的紀錄。**

|  |
| --- |
| 請將shell script寫在此處，並實際執行該shell script提供執行結果附件圖檔Q8.jpg。  #!/bin/bash  cd /tmp  logs=$(ls | grep .log)  tar cvf "logs.tar" $logs  cat /tmp/syslog.log | grep "root"  ls -f /tmp | grep ".tar"  下方表格Q8\_syslog.txt內的文字為syslog.log的檔案內容，請使用這些文字建立一個檔案並將其上傳至　您的Linux環境下的/tmp，並改名成syslog.log。 |

| **Q8\_syslog.txt** |
| --- |
| Aug 2 07:35:02 004 anacron[38389]: Job `cron.daily' terminated  Aug 2 07:35:02 004 anacron[38389]: Normal exit (1 job run)  Aug 2 07:38:19 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.89.198:123 (ntp.ubuntu.com).  Aug 2 07:38:30 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.89.199:123 (ntp.ubuntu.com).  Aug 2 07:38:40 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.91.157:123 (ntp.ubuntu.com).  Aug 2 07:38:50 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.94.4:123 (ntp.ubuntu.com).  Aug 2 08:13:09 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.89.199:123 (ntp.ubuntu.com).  Aug 2 08:13:19 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.91.157:123 (ntp.ubuntu.com).  Aug 2 08:13:29 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.89.198:123 (ntp.ubuntu.com).  Aug 2 08:13:39 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.94.4:123 (ntp.ubuntu.com).  Aug 2 08:17:01 004 CRON[43690]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)  Aug 2 08:47:58 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.91.157:123 (ntp.ubuntu.com).  Aug 2 08:48:08 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.89.198:123 (ntp.ubuntu.com).  Aug 2 08:48:18 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.89.199:123 (ntp.ubuntu.com).  Aug 2 08:48:29 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.94.4:123 (ntp.ubuntu.com).  Aug 2 09:17:01 004 CRON[50184]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)  Aug 2 09:22:47 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.89.198:123 (ntp.ubuntu.com).  Aug 2 09:22:57 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.89.199:123 (ntp.ubuntu.com).  Aug 2 09:23:08 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.91.157:123 (ntp.ubuntu.com).  Aug 2 09:23:18 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.94.4:123 (ntp.ubuntu.com).  Aug 2 09:57:36 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.91.157:123 (ntp.ubuntu.com).  Aug 2 09:57:47 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.89.199:123 (ntp.ubuntu.com).  Aug 2 09:57:57 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.94.4:123 (ntp.ubuntu.com).  Aug 2 09:58:07 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.89.198:123 (ntp.ubuntu.com).  Aug 2 10:17:01 004 CRON[56671]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)  Aug 2 10:32:26 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.89.199:123 (ntp.ubuntu.com).  Aug 2 10:32:36 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.89.198:123 (ntp.ubuntu.com).  Aug 2 10:32:46 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.91.157:123 (ntp.ubuntu.com).  Aug 2 10:32:56 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.94.4:123 (ntp.ubuntu.com).  Aug 2 10:58:59 004 systemd[1]: Starting Daily apt download activities...  Aug 2 10:59:01 004 systemd[1]: Started Daily apt download activities.  Aug 2 11:07:15 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.89.198:123 (ntp.ubuntu.com).  Aug 2 11:07:25 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.94.4:123 (ntp.ubuntu.com).  Aug 2 11:07:35 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.89.199:123 (ntp.ubuntu.com).  Aug 2 11:07:46 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.91.157:123 (ntp.ubuntu.com).  Aug 2 11:17:01 004 CRON[63255]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)  Aug 2 11:42:04 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.89.199:123 (ntp.ubuntu.com).  Aug 2 11:42:14 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.89.198:123 (ntp.ubuntu.com).  Aug 2 11:42:25 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.94.4:123 (ntp.ubuntu.com).  Aug 2 11:42:35 004 systemd-timesyncd[730]: Timed out waiting for reply from 91.189.91.157:123 (ntp.ubuntu.com).  Aug 2 11:59:58 004 systemd[1]: Created slice User Slice of ebti.  Aug 2 11:59:58 004 systemd[1]: Starting User Manager for UID 1000...  Aug 2 11:59:58 004 systemd[1]: Started Session 1790 of user ebti.  Aug 2 11:59:58 004 systemd[67875]: Listening on REST API socket for snapd user session agent.  Aug 2 11:59:58 004 systemd[67875]: Reached target Paths.  Aug 2 11:59:58 004 systemd[67875]: Reached target Timers.  Aug 2 11:59:58 004 systemd[67875]: Reached target Sockets.  Aug 2 11:59:58 004 systemd[67875]: Reached target Basic System.  Aug 2 11:59:58 004 systemd[67875]: Reached target Default.  Aug 2 11:59:58 004 systemd[67875]: Startup finished in 33ms.  Aug 2 11:59:58 004 systemd[1]: Started User Manager for UID 1000.  Aug 2 12:04:15 004 systemd[1]: Started Session 1791 of user ebti. |

**Q9: 請用Windows Power Shell或Windows command撰寫出以下模擬情境：**

**切換目錄至C:\log下，將C:\log目錄裡面的system.log更改名稱加上昨天日期(例：2021/8/5執行時，就將檔名改成：system\_20210804.log)，壓縮後，將檔案移動到 C:\log\_history內。**

|  |
| --- |
| 請將**Windows Power Shell或Windows command**寫在此處，並實際執行該shell提供執行結果附件圖檔Q9.jpg  cd C:\logs  echo Hello > system.log  set today=%date:~0,4%%date:~5,2%%date:~8,2%  ren system.log system\_%today%.log  move system\_%today%.log C:\log\_history\system\_%today%.log  pause |