

## APPELLO PASQUALE GESTIONE DELL'INFORMAZIONE AZIENDALE

### Domanda 1

Si spieghi, massimo in otto righe di testo, cosa s'intende per architettura a tre livelli di un sistema di data warehousing.

### Risposta

Si tratta di un sistema OLAP che comprende: (i) Staging Area, (ii) Data Warehouse, (iii) Data Marts

È il sistema più completo, che permette il prelievo di dati da fonti differenti con update asincrono e di tipo a batch (i dati stazionano nella staging area e, una volta puliti, arricchiti e verificati tramite le procedure ETL) vengono riversati nel data warehouse.

Tale architettura, oltre alle classiche funzionalità di "dashbording", "reporting", "query multidimensionali" è generalmente provvisto anche di strumenti di analisi più avanzati (es. statistica multivariata, data mining ecc.)

### Domanda 2

Si spieghi, massimo in cinque righe di testo, come funziona il prodotto cartesiano, operazione di algebra relazionale.

### Risposta

Il prodotto cartesiano opera su due tabelle A e B; non ci sono vincoli sulla struttura di suddette tabelle. Indichiamo con  $R_A$  e  $R_B$  in numero di record di A e di B, e con  $C_A$  e  $C_B$  il numero di campi di A e di B.

Allora il prodotto cartesiano restituisce una tabella con  $R_A \times R_B$  record, ciascuno con  $C_A + C_B$  record. Ciascuno è ottenuto combinando un record di A con un record di B.

### Domanda 3

Si consideri la seguente funzione, basata su DFunction ed operate su un'anagrafica studenti. Nel caso in cui la funzione venisse invocata usando la tua matricola ed un valore di N pari al giorno (in numero) della tua data di nascita, che valore verrebbe restituito in output? Si spieghi brevemente il funzionamento, massimo in cinque righe di testo.

```
Public Function CN (Mat As String, N As Integer) As String
```

```
Dim i As Integer
```

```
Dim s As String
```

```
    s = Nz(DLookUp("Cognome", "STUDENTI", "Matricola = '" & Mat & "'"), "XXX")
```

```
    For i = 1 To N Step 5
```

```
        CN = CN & i & '161116...'
```

```
    Next i
```

```
    CN = CN & "/" & s & '161116...Cognome'
```

```
End Function
```

### Risposta

La DLookUp restituisce il cognome dello studente la cui matricola è stata passata in input alla funzione.

Qualora la matricola non dovesse essere trovata, viene restituita la stringa XXX.

Per cui passando la "tua" matricola, la DLookUp restituirà il "tuo" cognome. Supponiamo che tale cognome sia Manzini. Supponiamo, inoltre, che "tu" sia nato il 25 dicembre. Per cui  $N = 25$ .

Allora il ciclo for viene eseguito 5 volte con i che vale {1, 6, 11, 16, 21}. Alla fine della quinta iterazione, infatti, i diventa 26, e il ciclo si interrompe.

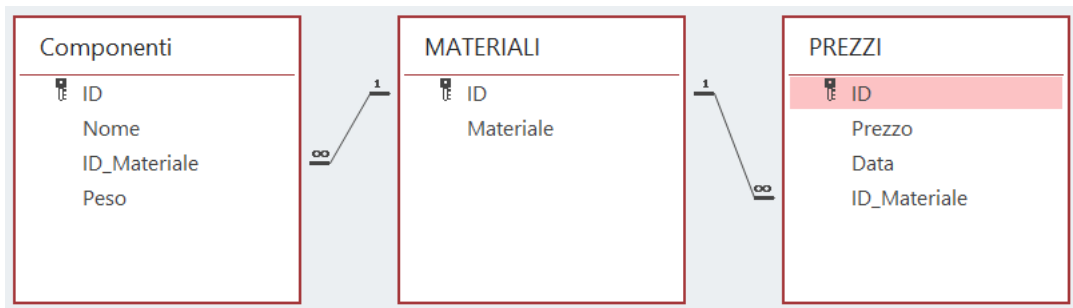
Dato che all'interno del ciclo, il valore di i viene via via "concatenato" (all'interno della variabile stringa CN), CN sarà pari a "16111621".

Prima di uscire, a CN viene aggiunto \Manzini

Per cui il risultato complessivo sarà → "16111621\Manzini".

### Esercizio 1

Si considerino le tre seguenti tabelle relative ad un ipotetico configuratore di prodotto. I componenti sono realizzati con un solo materiale ed hanno un certo peso in kg. I prezzi dei materiali variano nel tempo; tali variazioni sono elencate nella tabella PREZZI, in cui il campo Data indica proprio il momento in cui è stata osservata/registrata la variazione di prezzo. Il prezzo corrente è, ovviamente quello con la data più recente nella tabella prezzi.



Si chiede di scrivere una query che restituisce il valore/prezzo di ogni componente, al netto del valore di trasformazione, ottenuto come prodotto tra peso e prezzo del materiale. A tal fine si proceda a:

- Scrivere una query che restituisce per ogni materiale l'ultima data in cui il suo prezzo è stato aggiornato (si avranno in pratica due campi per record, il primo riporta l'ID del materiale, il secondo la data di aggiornamento del prezzo).
- Scrivere una query che, sfruttando la precedente, riporti il prezzo attuale di ogni materiale.
- Si usi quest'ultima query per ottenere il risultato, finale, voluto (valore di ogni componente).

### Soluzione

#### Date ultimo aggiornamento - UA

```
SELECT PREZZI.ID_Materiale, Max(PREZZI.Data) AS DM
FROM PREZZI
GROUP BY PREZZI.ID_Materiale
```

#### Prezzi correnti dei materiali - PCM

```
SELECT Prezzi.ID_Materiale, Prezzo
FROM PREZZI, UA
WHERE PREZZI.Data = UA.DM AND PREZZI.ID_Materiale = DM.ID_Materiale
```

#### Prezzi Componenti

```
SELECT Nome, Materiale, Prezzo*Peso AS Totale
FROM (MATERIALI INNER JOIN Componenti ON MATERIALI.ID = Componenti.ID_Materiale) INNER JOIN
PCM ON MATERIALI.ID = PCM.ID_Materiale
```

## Esercizio 2

Si supponga di disporre del seguente modulo di classe chiamato PERSON.

```
Option Compare Database
```

```
Public Name As String
```

```
Public Surname As String
```

```
Public Date_Of_Birth As Date
```

```
Public Id As String
```

```
Private Bs As PERSON
```

```
Property Get Boss() As Person
```

```
    Set Boss = Bs
```

```
End Property
```

```
Property Set Boss(B As Person)
```

```
    Set Bs = B
```

```
End Property
```

```
Public Function Gen_Id(Optional Start As Variant) As String
```

```
    If IsMissing(Start) Then Start = Left(Me.Name, 3) & "_" & Left(Me.Surname, 3) & "_"
```

```
    Gen_Id = Start & Int(Rnd() * 10000)
```

```
End Function
```

```
Public Function Age() As Integer
```

```
    Age = Year(Date) - Year(Date_Of_Birth)
```

```
    If Subtract(Date_Of_Birth) Then Age = Age - 1
```

```
End Function
```

```
Private Function Reduce(DOF) As Boolean
```

```
Dim M1 As Integer, M2 As Integer
```

```
Dim D1 As Integer, D2 As Integer
```

```
    M1 = Month(DOF)
```

```
    M2 = Month(Date)
```

```
    D1 = Day(DOF)
```

```
    D2 = Day(Date)
```

```
    Subtract = False
```

```
    If M1 > M2 Or (M1 = M2 And D1 > D2) Then
```

```
        Subtract = True
```

```
    End If
```

```
End Function
```

Si chiede di:

- Indicare cosa rappresenti tale classe e di spiegarne il funzionamento, in massimo dieci righe di testo.
- Completare la funzione `Public Function New_One(Name As String, Surname As String, Date_Of Birth As Date) As Person`, che sfrutta la classe PERSON per creare uno specifico "oggetto" persona. Tale funzione aggiunge anche l'ID, ma non il boss.
- Si usi (due volte) la funzione prima creata per realizzare due oggetti P1 e P2, di tipo PERSON. P1 rappresenta te stesso, mentre P2 è John Doe nato il primo gennaio del 50. Sapendo che John Doe è il tuo capo (boss), come dev'essere aggiornare l'oggetto P1 per contenere questa informazione?
- Supponiamo che nel DataBase sia presente una tabella T\_PEOPLE idonea a contenere tutte le informazioni di un oggetto di tipo PERSON. Si disegni tale tabella, definendo il tipo di ogni campo e indicando chiaramente il campo che funge da chiave primaria. Si faccia particolare attenzione al campo che deve contenere il Boss.
- Si crei una query operante su T\_PEOPLE che restituisce ID e cognome di ogni persona e ID e cognome del suo capo (boss) di primo livello
- Supponiamo ora che PEOPLE sia una collezione un vettore (a scelta), contenente oggetti di tipo PERSON. Si scriva una procedura che legge ciascun oggetto PERSON di PEOPLE e inserisce tali informazioni nella tabella T\_PEOPLE. Si assuma la collection come variabile globale già valorizzata.  
**Sugg.** Nel caso in cui usiate una collezione si consiglia l'utilizzo di un ciclo `For Each In`, nel caso di un vettore si consiglia l'utilizzo di un classico ciclo `For Next`. Per l'aggiornamento della tabella è praticamente indifferente utilizzare una query di `Insert Into`, oppure un `Recordset`.
- **Facoltativo.** Si aggiunga un metodo alla classe PERSON che restituisce una stringa contenente tutta la lista, in ordine gerarchico di tutti i capi di una persona.  
**Sugg.** Per creare il metodo, è consigliabile l'uso di una funzione ricorsiva, oppure di una funzione basata su ciclo While.

## Soluzione

Si tratta di una classe che rappresenta impiegati. Per ciascuno di essi si ha nome, cognome, data di nascita, codice identificativo, e capo di primo livello (il capo è, a sua volta, una variabile di tipo PERSON. Sono presenti le proprietà Get e Let per assegnare il capo, e due metodi per creare un id casuale e per calcolare l'età.

### **Funzione crea persona**

```
Public Function New_One(N As String, C As String, D As Date) As Person
```

```
Dim P As Person
```

```
Set P = New Person
```

```
P.Name = N
```

```
P.Surname = C
```

```
P.Date_Of_Birth = D
```

```
P.Id = P.Gen_Id()
```

```
Set New_One = P
```

```
End Sub
```

### **Creazione di due persone**

```
Dim P1 AS Persons, P2 AS Person
```

```
Set P1 = New_One("Giovanni", "Manzini", #23/12/1948#)
```

```
Set P2 = New_One ("John", "Doe", #1/1/1960#)
```

Set P1.Boss = P2

### **Definizione tabella**

La tabella avrà i seguenti campi, il campo ID è stringa, perché verrà letto direttamente dalla collection P:

- ID (PK) Stringa
- Nome – Stringa
- Cognome – Stringa
- Data\_di\_Nascita – Data
- Boss (FK) – FK interna, tipo dato stringa

### **Query per gerarchia**

```
SELECT T_PEOPLE.Surname, T_PEOPLE_1.Surname As Boss
FROM T_PEOPLE LEFT JOIN T_PEOPLE AS T_PEOPLE_1 ON T_PEOPLE.Boss = T_PEOPLE_1.ID;
```

### **Riempimento Tabella (con SQL)**

```
Public Sub Make_Table()
Dim MySQL As String
MySQL = "DELETE * FROM PEOPLE"
DoCmd.RunSQL (MySQL)
For Each P In People
MySQL = "INSERT INTO PEOPLE (ID, Name,Surname,Date_Of_Birth)"
If Not P.Boss Is Nothing Then MySQL = Left(MySQL, Len(MySQL) - 1) & ",Boss)"
MySQL = MySQL & " VALUES (" & P.Id & ",",
MySQL = MySQL & P.Name & ",",
MySQL = MySQL & P.Surname & ",#"
MySQL = MySQL & P.Date_Of_Birth & "#,"
If P.Boss Is Nothing Then
MySQL = MySQL & ")"
Else
MySQL = MySQL & ", " & P.Boss.Id & ")"
End If
DoCmd.RunSQL (MySQL)
Next P
End Sub
```

### **Riempimento Tabella (con recordset)**

```
Public Sub Make_Table2()
Dim MySQL As String
Dim Rcs As Recordset2
MySQL = "DELETE * FROM PEOPLE"
DoCmd.RunSQL (MySQL)
Set Rcs = CurrentDb.OpenRecordset("PEOPLE")
Rcs.Edit
For Each P In People
'A new empty record
Rcs.AddNew
Rcs.Fields("Id") = P.Id
```

```
Rcs.Fields("Name") = P.Name
Rcs.Fields("Surname") = P.Surname
Rcs.Fields("Date_Of_Birth") = P.Date_Of_Birth
If Not P.Boss Is Nothing Then Rcs.Fields("Boss") = P.Boss.Id
Rcs.Update
Next P
End Sub
```

### ***Facoltativo***

Il modo più semplice è quello di usare una funzione ricorsiva.

```
Public Function My_Bosses() As String
    If Me.Boss Is Nothing Then
        My_Bosses = "No One"
        Exit Function
    Else
        Set B = Me.Boss
        My_Bosses = B.Id & " " & B.My_Bosses()
    End If
End Function
```