

GESTIONE DELL'INFORMAZIONE AZIENDALE

Laurea magistrale in ingegneria gestionale a.a. 2021-2022

Primo appello estivo 08/06/2022

Nome _____

Cognome _____

SEZIONE 1 – DOMANDE

Domanda 1 (2 punti)

Si considerino le tabelle STUDENTI, ESAMI_SOSTENUTI ed ESAMI, riportate in figura 1.

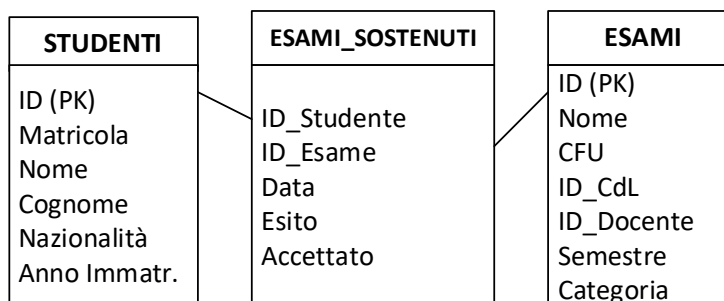


Figura 1. Porzione del DB del sistema informativo di un'università

Si chiede di:

- Spiegare perché il campo matricola non può essere usato come chiave primaria.
- Spiegare quale tipo di dato è presumibilmente utilizzato per il campo Esito ed Accettato.
- Spiegare che tipo di relazione lega le tre tabelle.
- Indicare uno o più campi con cui formare la chiave primaria della tabella ESAMI_SOSTENUTI.
- Spiegare il significato dei campi ID_CdL (CdL Sta per Corso di Laurea) e ID_Docente della tabella ESAMI.

Domanda 2 (3 punti)

Si scriva una query parametrica che, operando su una o più delle tabelle di figura 1, restituisce la media degli esami sostenuti da uno specifico studente (lo studente è il parametro della query).

Si supponga che nel data base: (i) siano registrati anche voti inferiori a 18 (bocciature) e (ii) si tenga traccia anche dei voti rifiutati da uno studente. Per cui uno studente potrebbe essere associato a più voti per uno stesso esame.

SOLUZIONE

```
SELECT Avg(Esito) As Media_Voti
FROM STUDENTI INNER JOIN ESAMI_SOSTENUTI ON STUDENTI.ID = ESAMI_SOSTENUTI.ID_Studente
WHERE Esito >= 18 AND Accettato = True AND Studente.Matricola = ["Inserire la matricola"]
```

Si noti che si potrebbe usare anche l'ID studente, in quel caso non servirebbe la Join, l'uso della matricola + però più corretto. Si tratta, infatti, di un altro campo univoco, che sicuramente deve essere presente per tutti

gli studenti che hanno effettuato almeno un esame. La matricola, inoltre, è più “informativa” rispetto all’ID che è semplicemente un progressivo generato automaticamente dal sistema.

Osservazione (non richiesta nell’esercizio).

Per avere l’indicazione della media di ciascuno studente, la query diventa la seguente:

```
SELECT Matricola, Avg(Esito) As Media_Voti
FROM STUDENTI INNER JOIN ESAMI_SOSTENUTI ON STUDENTI.ID = ESAMI_SOSTENUTI.ID_Studente
WHERE Esito >= 18 AND Accettato = True AND Studente.Matricola = ["Inserire la matricola"]
Group By Matricola
```

Domanda 2 (4 punti)

Si spieghi, in maniera sintetica e possibilmente utilizzando un esempio grafico, la differenza tra un gateway ed un evento intermedio di un diagramma BPMN.

Domanda 3 (4 punti)

Si consideri una tabella contenente dati meteo in cui, ad ogni giorno registrato nel database è associata la temperatura massima, quella minima (in gradi centigradi), il vento (in metri al secondo) e i millimetri di pioggia caduta. Un esempio è mostrato in tabella 1.

Tabella 1
Tabella METEO

ID	Data	Tmax_c	Tmin_c	Pioggia_mm	Vento_ms
1	01/01/2020	10	-5	0	10
2	02/01/2020	8	-3	0	0
...					
N	05/06/2022	25	16	15	10

Si sfrutti tale tabella per scrivere una query SQL, basata su operatori di gruppo, per spiegare la differenza tra gli operatori HAVING e WHERE.

SOLUZIONE

Supponiamo di voler trovare la temperatura minima media di ogni mese registratasi solo nei giorni piovosi. La query sarebbe la seguente:

```
SELECT Month(Data) AS Mese, AVG(T_min_c) AS Tmin
FROM METEO
WHERE Pioggia_mm > 0
GROUP BY Month(Data)
```

In questo caso, quindi, usiamo WHERE per eliminare dalla media tutti i giorni non piovosi.

Supponiamo ora di voler vedere solo i mesi con temperatura media minima inferiore o uguale a 5°C. Per aggiungere questo ulteriore filtro, dato che lavoriamo su un campo di gruppo calcolato, dobbiamo usare HAVING, come mostrato di seguito:

```

SELECT Month(Data) AS Mese, AVG(T_min_c) AS Tmin
FROM METEO
WHERE Pioggia_mm > 0
GROUP BY Month(Data)
HAVING AVG(T_min_c) <= 5

```

SEZIONE 2 - ESERCIZI

Esercizio 1 (6 punti)

Si supponga di disporre della vista di Tabella 2, contenente tutti i record necessari ad alimentare un fatto di vendita.

Si chiede di:

- Indicare chiaramente quali siano le dimensioni e quali le metriche.
- Disegnare il diagramma DFM che rappresenta il corrispondente fatto di vendita.
- Supponendo che l'implementazione preveda un'architettura MOLAP, si disegnino e si compilino tutte le matrici multidimensionali ottenibili a partire dalla vista di partenza.

Tabella 2
Vista di partenza

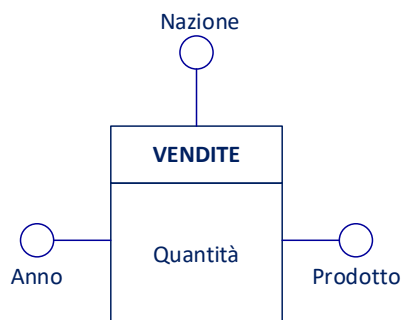
Prodotto	Nazione	Anno	Quantità
P1	ITA	2019	10
P2	ITA	2020	20
P1	USA	2020	10
P1	ITA	2020	10
P3	USA	2019	20
P2	USA	2019	10
P3	ITA	2020	20

SOLUZIONE

Dimensioni: Prodotto, Nazione ed Anno

Metrica: Quantità venduta

DFM



Supponiamo che gli assi siano i seguenti:

- x: Prodotto (riga della matrice)
- y: Nazione (colonna della matrice)
- z: Anno (dimensione "trasversale")

Le coordinate sono le seguenti:

- Prodotto {P1, P2, P3}
- Nazione {ITA, USA}
- Anno {2019, 2002}

A fronte di ciò avremo 2 matrici (asse z degli anni) ciascuna di dimensioni 3x2. Le matrici sono le seguenti:

ANNO 2019	ITA	USA
P1	10	None
P2	None	10
P3	None	20

ANNO 2020	ITA	USA
P1	10	10
P2	20	None
P3	20	None

Esercizio 2 (11 + 3 punti)

Si scrivano le tre seguenti funzioni VBA, operanti sulla tabella Meteo (tabella 2).

- Public Function Good_Day(D As Date, Optional Filter As Variant) As Boolean. Tale funzione riceve in input una data ed un criterio di filtraggio operante sulle condizioni meteo (es. sereno senza vento, vento inferiore a 5 m/s e temperatura minima maggiore di 20°C, ecc.), e restituisce True se le condizioni meteo registrate nella data di input soddisfano le condizioni di filtraggi. La funzione restituisce False se le condizioni non sono soddisfatte o se la data cercata non è presente. Il filtro è opzionale, se non viene passato in input, la funzione restituisce True se nella data cercata non ha piovuto (giorno sereno). Si sviluppi tale funzione utilizzando una DlookUp. **(3 punti)**
- Public Function Raining_Days(Optional Filter As Variant) As Double. Tale funzione riceve un criterio di filtraggio operante sulle date (es. solo mese di gennaio) e restituisce la percentuale di giorni di pioggia registrati nei giorni che soddisfano la condizione di filtraggio. Se non viene passato alcun filtro, la percentuale viene calcolata su tutti i record della tabella. Suggerimento: è preferibile scrivere tale funzione utilizzando un recordset, ma l'uso di una o più DFunction è comunque accettabile. **(4 punti)**

- Public Function Tmin() As Variant. Tale funzione restituisce una matrice con tante righe quante sono le giornate in cui è stata registrata la temperatura minima assoluta. Si noti, infatti, che la stessa temperatura minima assoluta potrebbe essere stata registrata in più giornate differenti. Inoltre, ogni riga della matrice contiene 4 valori (4 colonne) che rappresentano: il giorno, il mese e l'anno, della data in cui è stata registrata la temperatura minima, e la temperatura minima stessa. Quest'ultima sarà ovviamente uguale in ogni riga della matrice. **(4 punti)**

Soluzione

```
Public Function Good_Day(D As Date, Optional Filter As Variant) As Boolean
```

```
Dim Wher_Cond As String
```

```
Dim F_Date As String
```

```
Dim Day As Integer
```

```
F_Date = Format(D, "\#mm\dd\yyyy\#") 'Per sicurezza mettiamo formato americano
```

```
Where_cond = "Data = #" & D & "#" 'La data è sempre presente nella condizione di filtraggio
```

```
If IsMissing(Filter) Then 'Se non c'è nessuna condizione mettiamo il default: giornata non piovosa
```

```
Where_cond = Where_cond & "And Pioggia_mm = 0"
```

```
Else
```

```
Where_cond = Where_cond + " And " + Filter
```

```
End If
```

```
Day = Nz(DLookup("ID", "Meteo", Where_cond)) 'Il giorno (se presente) che soddisfa la condizione
```

```
If Day <> 0 Then
```

```
Good_Day = True
```

```
Else
```

```
Good_Day = False
```

```
End If
```

```
End Function
```

```
Public Function Raining_Days(Optional Filter As Variant) As Double
```

```
Dim Rcs As Recordset2
```

```
Dim MySQL As String
```

```
Dim N_Rain As Integer, N_Sun As Integer
```

```
If IsMissing(Filter) Then 'Se non c'è nessun filtro apriamo il recordset sulla tabella di partenza
```

```
Set Rcs = CurrentDb.OpenRecordset("Meteo")
```

```
Else
```

```
MySQL = "SELECT * FROM METEO WHERE " & Filter 'Altrimenti aggiungiamo la condizione di filtraggio
```

```
Set Rcs = CurrentDb.OpenRecordset(MySQL)
```

```
End If
```

```
Do While Not Rcs.EOF 'Conteggiamo i giorni di pioggia e i giorni soleggiati
```

```
If Rcs.Fields!Pioggia_mm > 0 Then
```

```
N_Rain = N_Rain + 1
```

```
Else
```

```
N_Sun = N_Sun + 1
```

```
End If
```

```

    Rcs.MoveNext
Loop

Raining_Days = N_Rain / (N_Rain + N_Sun)
Rcs.Close
Set Rcs = Nothing
End Function

Public Function Tmin() As Variant
Dim Y As Integer, M As Integer, D As Integer
Dim V() As Variant, C As Variant
Dim min As Integer
Dim MySQL As String
Dim N As Integer
Dim Rcs As Recordset2
    min = DMin("T_min_c", "Meteo")
    MySQL = "SELECT Data FROM METEO WHERE T_min_c = " & min 'Giorni con temperature minima assoluta
    Set Rcs = CurrentDb.OpenRecordset(MySQL)
    Rcs.MoveLast
    N = Rcs.RecordCount 'Il numero di record corrisponde al numero di righe della matrice
    ReDim V(1 To N, 1 To 4) 'Dimensioniamo la matrice
    Rcs.MoveFirst 'Torniamo sul primo record e iniziamo a compilare la matrice, ciclando su tutti i record
    For i = 1 To N
        V(i, 1) = Day(Rcs.Fields!Data)
        V(i, 2) = Month(Rcs.Fields!Data)
        V(i, 3) = Year(Rcs.Fields!Data)
        V(1, 4) = min
        Rcs.MoveNext
    Next i

Tmin = V 'Assegniamo la matrice alla variabile d'uscita
Rcs.Close
Set Rcs = Nothing

End Function

```

Facoltativo

- Si esegua la funzione Good_Day(), usando una condizione di filtro che accetta solo le giornate soleggiate, con vento inferiore a 5m/s e temperatura minima superiore a 15°C. **(1 punto)**
- Si scriva una procedura che riceve in input la matrice restituita da Tmin() e, per ogni riga della matrice, stampa a video la scritta "Temperatura minima di X°C registrata il gg-mm-aa", dove X è la temperatura minima, gg il giorno, mm il mese e aa l'anno. **(2 punti)**

Soluzione

Per eseguire Good_Day direttamente dalla finestra immediata possiamo scrivere:

```
? Good_Day(#01/01/2020#, "Pioggia_mm 0 AND Vento_ms < 5 AND T_min_C > 15")
```

Oppure, possiamo eseguirla tramite una sub routine di prova come la seguente:

```
Public Sub Try_Good_Day()  
Dim D As Date  
    D = DateSerial(2022, 2, 23) 'creiamo una data  
    X = Good_Day(D, "Pioggia_mm 0 AND Vento_ms < 5 AND T_min_C > 15")  
    Debug.Print (X)  
End Sub
```

Per quanto riguarda la seconda domanda, possiamo scrivere una sub routine come la seguente, in cui creiamo la matrice e poi la leggiamo.

```
Public Sub Read_Data()  
Dim S As String  
Dim M As Variant  
    M = Tmin()  
    X = UBound(M)  
    For i = 1 To UBound(M)  
        S = "Temperatura minima di " & M(i, 4) & "°C registrata in data " & M(i, 1) & "-" & M(i, 2) & "-" & M(i, 3) & "  
        Debug.Print (S)  
    Next i  
End Sub
```