

*Ministero dell'Istruzione, dell'Università e della Ricerca***M070 – ESAME DI STATO DI ISTITUTO TECNICO INDUSTRIALE****CORSO DI ORDINAMENTO****Indirizzo: INFORMATICA****Tema di: INFORMATICA GENERALE, APPLICAZIONI TECNICO SCIENTIFICHE
INFORMATICA****(Testo valevole per i corsi di ordinamento e per i corsi sperimentali del Progetto “Sirio”)**

Si vuole creare un sistema per automatizzare la gestione dei percorsi di alternanza Scuola-Lavoro realizzati, a partire dall'anno scolastico 2011/2012, da un istituto tecnico.

I percorsi sono progettati ed attuati in collaborazione con aziende selezionate nel territorio. Per consentire l'inizio delle attività, ciascun Consiglio di Classe dell'istituto nomina un docente referente e seleziona gli studenti che frequenteranno le attività formative in alternanza presso le aziende. Le aziende nominano il tutor aziendale che seguirà gli studenti nel loro percorso di apprendimento. Gli studenti possono essere inviati a più riprese presso le aziende effettuando, in tal modo, più di un periodo di formazione. Al termine del periodo di formazione, il tutor aziendale fornisce al docente referente la valutazione sulle attività svolte dagli studenti. Nella base di dati si vogliono memorizzare i dati degli studenti, con l'indicazione della classe che frequentano, e dei docenti referenti.

Compito delle aziende è indicare il numero di studenti, l'inizio e la durata del periodo di formazione, e rilasciare un attestato di partecipazione che riporta le generalità dello studente, la classe, le attività svolte, la valutazione, i nominativi del tutor aziendale e del docente referente.

Il sistema per automatizzare la gestione dei percorsi in alternanza deve prevedere:

1. la memorizzazione delle informazioni delle aziende e dei tutor designati da ciascuna azienda e quelle riguardanti i periodi di formazione in alternanza (descrizione, argomento, modalità di svolgimento, ecc);
2. la gestione dello svolgimento dei periodi di formazione in alternanza (data inizio, data fine, valutazione, ecc.).

Il candidato, fatte le opportune ipotesi aggiuntive, realizzi:

1. un'analisi della realtà di riferimento individuando le possibili soluzioni e scelga quella che a suo motivato giudizio è la più idonea a rispondere alle specifiche indicate;
2. uno schema concettuale della base di dati;
3. uno schema logico della base di dati;
4. la definizione delle relazioni della base di dati in linguaggio SQL;



Ministero dell'Istruzione, dell'Università e della Ricerca

M070 – ESAME DI STATO DI ISTITUTO TECNICO INDUSTRIALE

CORSO DI ORDINAMENTO

Indirizzo: INFORMATICA

**Tema di: INFORMATICA GENERALE, APPLICAZIONI TECNICO SCIENTIFICHE
INFORMATICA**

(Testo valevole per i corsi di ordinamento e per i corsi sperimentali del Progetto “Sirio”)

5. le seguenti interrogazioni espresse in linguaggio SQL:
 - a. elencare le aziende ed i relativi tutor;
 - b. elencare gli studenti delle classi quinte che partecipano all'attività di alternanza, suddivisi per specializzazione;
 - c. determinare il numero di studenti che ciascuna azienda ha accolto nell'anno scolastico 2013/2014;
 - d. stabilire la classe con il maggior numero di studenti che, nell'anno scolastico 2012/2013, hanno frequentato i percorsi di alternanza;
 - e. stampare gli attestati relativi ai periodi di formazione a cui ha partecipato un singolo studente;
6. l'interfaccia utente che il candidato intende proporre per interagire con la base di dati e codificare, in un linguaggio di programmazione a scelta, un segmento significativo del progetto realizzato;
7. l'architettura del sito web dell'istituto che presenti alle famiglie ed alle aziende l'attività di alternanza scuola-lavoro.

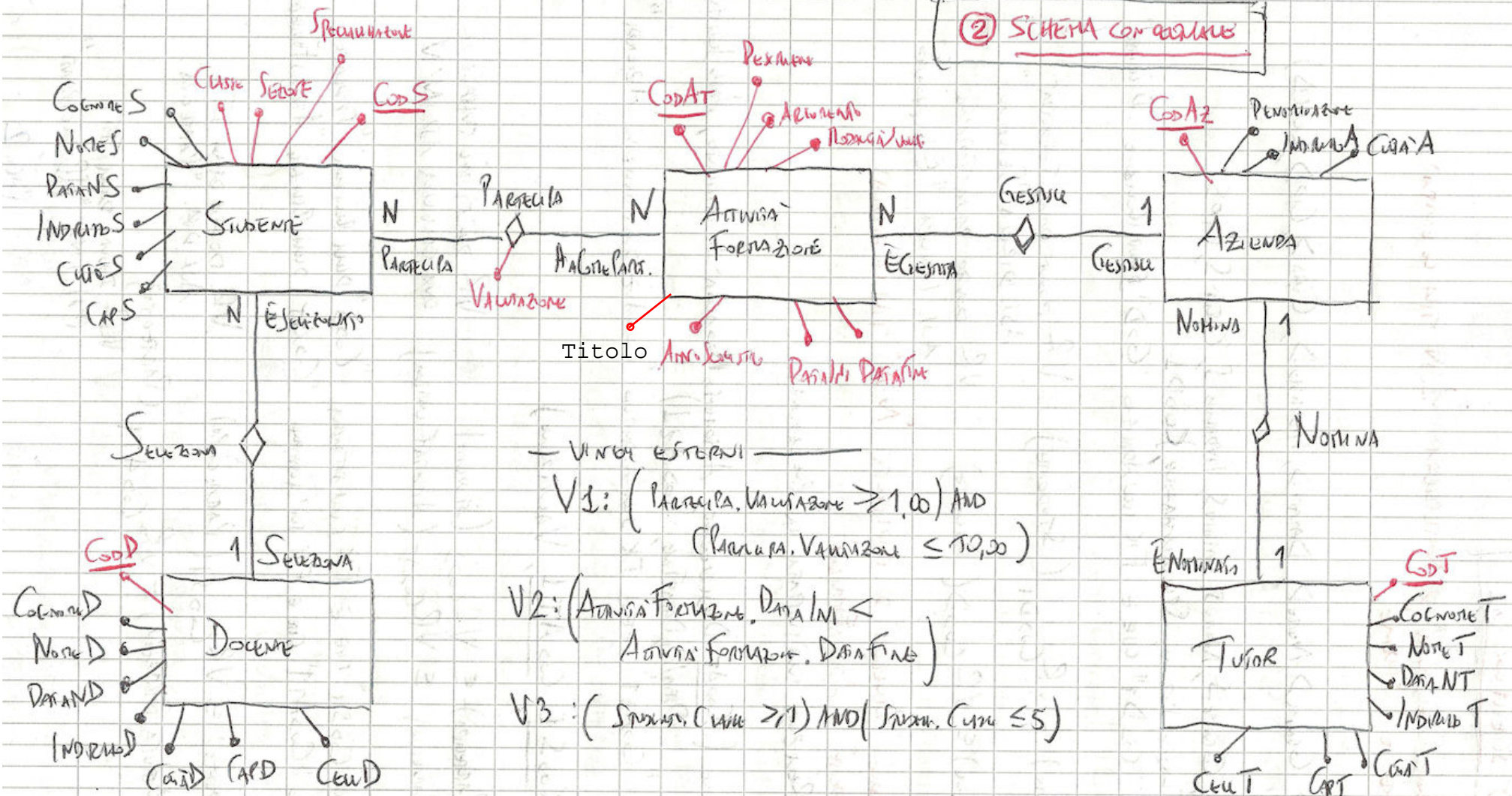
Durata massima della prova: 6 ore.

È consentito soltanto l'uso di manuali tecnici e di calcolatrici non programmabili.

È consentito l'uso del dizionario bilingue (italiano-lingua del paese di provenienza) per i candidati di madrelingua non italiana.

Non è consentito lasciare l'Istituto prima che siano trascorse 3 ore dalla dettatura del tema.

② Schema con regole



③ Schema Logico

(a) MAPPING Associazione "Sezione" di Montepulciano "1:N" TRA LE ENTITA' "Docente" e "Studente"

① Docente (CodD, GenereD, NomeD, DataND, Indirizzo, CognD, CIPD, CellD)

② Studente (CodS, GenereS, NomeS, DataNS, IndirizzoS, CognS, CIPS, Classe, Sezione, Specializzazione, CodDI)
 - con l'attributo "CodDI" che è FK sul attributo "CodD" della Relazione "Docente"

$VR_{CodD}(Docente) \in VR_{CodDI}(Studente)$ VR dovuto dalla relazione
 ASS. DIRETTA "Sezione"

$VR_{CodDI}(Studente) \in VR_{CodD}(Docente)$ VR dovuto dalla relazione
 ASS. INVERSA "Esecuzione"

(b) MAPPING Associazione "Partecipa" di Montepulciano "N:N" TRA LE ENTITA' "Studente" e "Attivita'Formazione"

Studente : Relazione già definita in precedenza

③ Attivita'Formazione (CodAT, Descrizione, Argomenti, Modalita' Svolgimento, Anno Scouting, DataIn, DataFine, CodAZ1)
 - con l'attributo "CodAZ1" che è FK sul attributo "CodAZ" della Relazione "Azienda"

④ Partecipa (CodS1, CodAT1, Valutazione)

$\begin{cases} VR_{CodS1}(Partecipa) \in VR_{CodS}(Studente) \\ VR_{CodAT1}(Partecipa) \in VR_{CodAT}(Attivita'Formazione) \end{cases}$ VR dovuto dal mapping di
 una relazione ASS. N:N

- con l'attributo "CodS1" che è FK sul attributo "CodS" della Relazione "Studente"

- con l'attributo "CodAT1" che è FK sul attributo "CodAT" della Relazione "Attivita'Formazione"

$VR_{CodS}(Studente) \in VR_{CodS1}(Partecipa)$ VR dovuto dalla Tot
 ASS. DIR. "Partecipa"

$VR_{CodAT}(Attivita'Formazione) \in VR_{CodAT1}(Partecipa)$ VR dovuto dalla Tot
 ASS. INV. "Partecipa"

(*) $V3(\dots) \Rightarrow V3(\text{COSTO}) : ((\text{Costo} \geq 1) \text{ AND } (\text{Costo} \leq 5))$
 Vincolo interpretativo su singoli N-PH sul dominio di 1 Agente

(c) MAPPING ASSUNZIONE ^{"Gestore"} di partecipanti $1:N$ tra le entità "Azienda" e "Attività Formative"
NB Azienda lato N

(3) Azienda (CodA2 Denominazione, Indirizzo, CittàA, CodT1)
 - GN (dominio "CodT1" che è FK sul dominio "CodT" con chiave "Turco"
 Attività Formative: Risorsa cui partecipa in precedenza

$VR_{\text{CodA2}}(\text{Azienda}) \subseteq VR_{\text{CodA2}}(\text{Attività Formative})$ VR dato Tot. At. Dir. "Gestore"
 $VR_{\text{CodT1}}(\text{Attività Formative}) \subseteq VR_{\text{CodT1}}(\text{Azienda})$ VR dato Tot. At. Inv. "È Gestore"

(d) MAPPING ASSUNZIONE ^{"Norma"} di partecipanti $1:1$ tra le entità "Azienda" e "Turco"
NB "Azienda" lato N

Azienda: Risorsa cui partecipa in precedenza

(2) Turco (CodT, GenereT, NomeT, DataNT, IndirizzoT, CittàT, CapT, CelT)

$VR_{\text{CodT1}}(\text{Azienda}) \subseteq VR_{\text{CodT1}}(\text{Turco})$ VR dato data Tot. At. Dir. "Norma"
 $VR_{\text{CodT}}(\text{Turco}) \subseteq VR_{\text{CodT1}}(\text{Azienda})$ VR dato data Tot. At. Inv. "È Normale"

(e) MAPPING NEZANZA DEI VINCHI

V. implicati di PK \Rightarrow Vinchi interpretativi su più N-PH

V. implicazioni multi. \Rightarrow Vinchi interpretativi referenziali
 Totalità dell'ASI.

Vinchi interpretativi referenziali creati dal
 mapping di un'assunzione $N:N$

$V1(\dots) \Rightarrow V1(\text{partecipanti}) : ((\text{Valutazione} > 1.00) \text{ AND } (\text{Valutazione} \leq 10.00))$

Vinchi interpretativi su singoli N-PH sul
 dominio di 1 Agente

$V2(\dots) \Rightarrow V2(\text{Attività Formative}) : (\text{DataIn} < \text{DataFin})$

$V3(\dots)$ (*) ^{VEDI SOPRA} Vinchi interpretativi su singoli N-PH sul
 dominio di più Agenti PAG. 3

④ DEFINING DATABASE IN SQL

Prerequisites: CREATE DATABASE CREATE CON | PRIMARY KEY AND AUTO INCREMENT
or DDL & DML SQL DEFINITIONS.

```
CREATE DATABASE AUTOMATICS;  
USE AUTOMATICS;
```

CREATE TABLE **Doctors**

①

```
(  
  Coid CHAR(10) NOT NULL,  
  Cfullname CHAR(30) NOT NULL,  
  Nost CHAR(30) NOT NULL,  
  Dnand DATE NOT NULL,  
  Indmild CHAR(25),  
  Cuid CHAR(30),  
  Gid CHAR(5),  
  Cell CHAR(30),  
  PRIMARY KEY (Coid)  
);
```

CREATE TABLE **Trice**

②

```
(  
  Coid CHAR(10) NOT NULL,  
  Cfullname CHAR(30) NOT NULL,  
  Nost CHAR(30) NOT NULL,  
  Dnand DATE NOT NULL,  
  Indmild CHAR(25),  
  Cuid CHAR(30),  
  Gid CHAR(5),  
  Cell CHAR(30),  
  PRIMARY KEY (Coid)  
);
```


CREATE TABLE **AZULENDA**

(2)

```
(
  CodAz CHAR(10) NOT NULL,
  Denominacion CHAR(50) NOT NULL,
  Indica A CHAR(255),
  Cua A CHAR(30),
  CodT1 CHAR(10) NOT NULL,
  PRIMARY KEY (CodAz),
  FOREIGN KEY (CodT1) REFERENCES Turise(CodT)
  ON DELETE CASCADE
  ON UPDATE CASCADE
);
```

CREATE TABLE **STUDENTE**

(4)

```
(
  CodS CHAR(10) NOT NULL,
  Cognome S CHAR(30) NOT NULL,
  Nome S CHAR(30) NOT NULL,
  Data N S DATE NOT NULL,
  Indico S CHAR(255),
  Cua S CHAR(30),
  Cap S CHAR(5),
  Cune INT NOT NULL,
  Sesso CHAR(1) NOT NULL,
  Specializazione CHAR(50) NOT NULL,
  CodD1 CHAR(10) NOT NULL,
  PRIMARY KEY (CodS),
  FOREIGN KEY (CodD1) REFERENCES Docenti(CodD)
  ON DELETE CASCADE
  ON UPDATE CASCADE,
  CHECK (Cune BETWEEN 1 AND 5)
);
```

VINGO V3

CREATE TABLE

AziendaFormulare

5

(

Cod AT CHAR (10) NOT NULL,
Descrizione CHAR (255) NOT NULL,
Accessorio CHAR (255) NOT NULL,
ModuloSubic CHAR (255) NOT NULL,
AnnoScritto CHAR (9) NOT NULL,
DataIn DATE NOT NULL,
DataFine DATE NOT NULL,
CodAZI CHAR (10) NOT NULL,

PRIMARY KEY (Cod AT)

FOREIGN KEY (CodAZI) REFERENCES Azienda (CodAZI)

ON DELETE CASCADE

ON UPDATE CASCADE

CHECK (DataIn < DataFine) // VINGLO V2

);

CREATE TABLE

Partecipanti

6

(

CodS1 CHAR (10) NOT NULL,
CodAT1 CHAR (10) NOT NULL,
Valutazione NUMERIC (2,2) NOT NULL,

PRIMARY KEY (CodS1, CodAT1)

FOREIGN KEY (CodS1) REFERENCES Studente (CodS)

ON DELETE CASCADE

ON UPDATE CASCADE

FOREIGN KEY (CodAT1) REFERENCES AziendaFormulare (CodAT)

ON DELETE CASCADE

ON UPDATE CASCADE

CHECK (Valutazione BETWEEN 1.00 AND 10.00) // VINGLO V1

);

PAR. 6

⑤ Query

① ELEMENTI CHE AZIENDA HA COMPRATO TUTOR

$Q_1 = A_{azienda} \bowtie T_{tutor}$
 $Cost1 = Cost$

SELECT *

FROM Azienda, Tutor

WHERE (Cost1 = Cost);

② ELEMENTI GLI STUDENTI DELLE CLASSE PRIME CHE PARTECIPANO ALL'ANNO DI AMMINISTRAZIONE, SUDDIVISI PER SPECIALEZZA

DISTINCT

SELECT Specialization, Course, Note

FROM Student, Partecipa

WHERE (Course = Course1) AND (Student.Course = 5);

⚠️ NON NECESSARIO JOIN SU

Amministratore

③ DETERMINARE IL NUMERO DI STUDENTI CHE CUI, CUI AZIENDA HA ACCORDO ANNO SCOLASTICO 2013/2014

SELECT COUNT(*) AS Totale

FROM Student, Partecipa, Amministratore

WHERE (Course = Course1) AND (CourseAT1 = CourseAT) AND (Amministratore Anno/Course = "2013/2014")

GROUP BY CourseAT1;

SELECT Denominazione, COUNT(*) AS Totale

FROM Student, Partecipa, Amministratore, Azienda

WHERE (Course = Course1) AND (CourseAT1 = CourseAT) AND (CourseAZ1 = CourseAZ2)

AND (Anno.Planisro = "2013/2014")

GROUP BY Denominazione;

- (d) STABILIRE LA CLASSE CON IL MAGGIOR NUMERO DI STUDENTI CHE, NELL'ANNO SCOLASTICO 2012/2013, HANNO FREQUENTATO I PERCORSI DI ADESIONE

```
CREATE VIEW t1 AS (SELECT Classe, Sezione, COUNT(*) AS Totale
FROM STUDENTE, PARTECIPA, ATTIVITA_FORNITA
WHERE (CodS=CodS1) AND (CodAT1=CodAT)
AND (AnnoScolare="2012/2013")
→ GROUP BY Classe, Sezione);
```

```
SELECT Classe, Sezione
FROM t1
WHERE Totale = (SELECT MAX(Totale)
FROM t1);
```

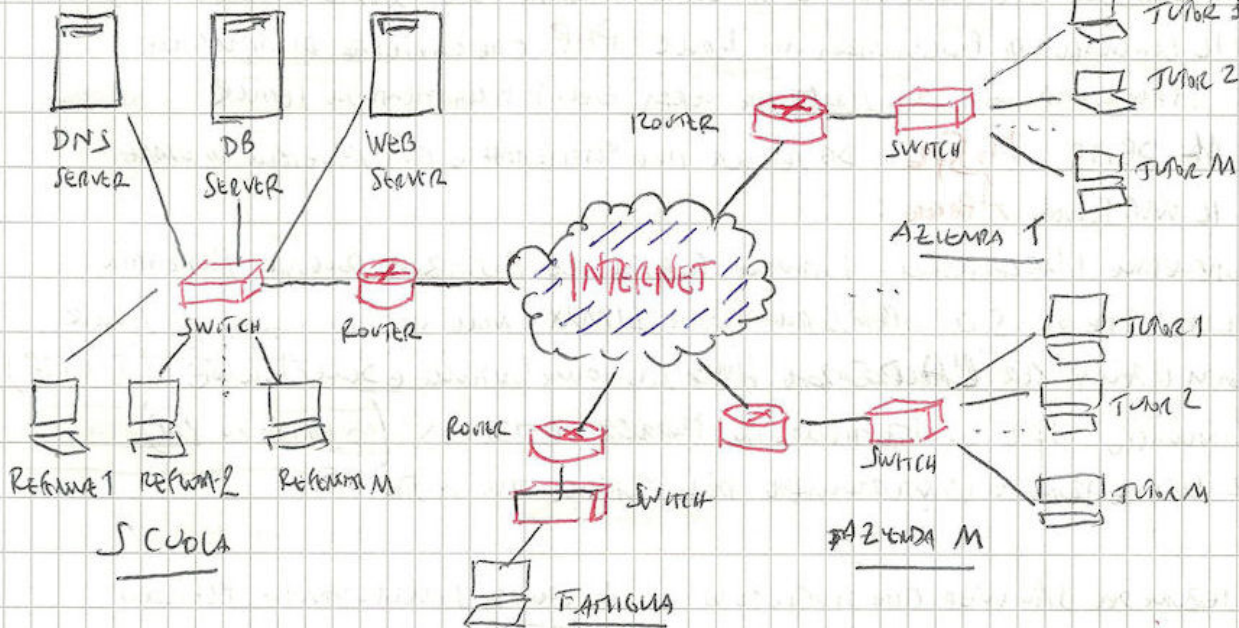
- (e) STABILIRE GLI ADESIONI RELATIVI AI PERCORSI DI FORMAZIONE A CUI HA PARTECIPATO UN SINGOLO STUDENTE

```
SELECT CognomeS, NomeS, Classe, Sezione, ATTIVITA_FORNITA.*, Valutazione,
CognomeT, NomeT, CognomeD, NomeD
FROM DOCENTE, STUDENTE, PARTECIPA, ATTIVITA_FORNITA, AZIENDA_TUTOR
WHERE ((CodD=CodD1) AND (CodS=CodS1) AND (CodAT1=CodAT)
AND (CodAZ1=CodAZ) AND (CodT1=CodT)
AND (CognomeS=[CognomeX])
AND (NomeS=[NomeX]))
ORDER BY AnnoScolare;
```


⑥ INTERFACCIA UTENTE PROGETTA PER INTERAGIRE CON LA BASE DATI E CODIFICATA IN UN LINGUAGGIO DI FORMAZIONE SICURA, UN SISTEMA SIGNIFICATIVO DEL PRODOTTO ACQUISITO

⑦ ARCHITETTURA DEL SITO WEB DELL'ISTITUTO CHE PRESENTI ANCHE FAMIGLIE ED ALTE AZIENDE L'ATTIVITÀ DI ADESIONE SCUOLA-LAVORO

Hip: L'ISTITUTO DISPOSTO DB SERVER e WEB SERVER SU MODERATA (ANCHE PER STESSE RAGIONI)



OGGI AL GIORNO D'OGGI OGNI AZIENDA HA UNA PROPRIA LAN CHE SFRUTTA ANCHE PER INTERAGIRE CON IL WEBSERVER DISPOSTO NELL'ISTITUTO SCOLASTICO (SOLUZIONE ANCHE PER WEB SERVER + DB SERVER FORNITI DA UN'ISP).

ANCHE LE SCUOLE AL GIORNO D'OGGI SONO DOTATE DI LAN E PER RIENTRARE I COSTI ENTRO UN CERTO TERMINE COMUNICANO TUTTI I DATI CON LA SCUOLA-AZIENDA FAMIGLIA TRAMITE RETE INTERNET (EVENTUALMENTE PER UNA MAGGIORE SICUREZZA POTREBBE ESSERE CREATA UNA VPN - VIRTUAL PRIVATE NETWORK).

I REFERENTI SCOLASTICI POSSONO AGGIORNARE I DATI TRAMITE RETE LOCALE O LOCALE PRODOTTO DI AUTENTICAZIONE DI TIPO LOGIN PASSWORD ED I SENSO.

I TUTOR AZIENDALI POSSONO ACCEDERE ALE INFORMAZIONI RELATIVE AI PROGETTI ED ALE ATTIVITÀ DI FORMAZIONE RELATIVE ALL'ATTIVITÀ SCOLA-LAVORO SENZA POTER ADESIONE MODIFICARE.

I REFERENTI SCOLASTICI POSSONO ACCEDERE ALE INFORMAZIONI RELATIVE ALE ADESIONI EVENTUALMENTE DEI LORO ALUNNI LORO AFFIDATI CON UN'OSSERVAZIONE COMPLETA DEI PRIVILEGI UTENTI POSSIBILI.

EVENTUALMENTE POTREBBE ESSERE AGGIUNTO UN SISTEMA DI IDENTIFICAZIONE PARTICOLARMENTE SICURO (TRAMITE WEB USANDO DATI LAN ED AUTENTICANDO IL RISULTATO) MEDIANTE CERTIFICATO SSL PER RENDERE POSSIBILE IL LAVORO DA CASA DA PARTE DEI TUTOR.

ESISTONO MOLTE SOLUZIONI PER LO SVILUPPO DELL'INTERFACCIA UTENTE VISO SERVER DEL SITO WEB RICHIESO

UNA SOLUZIONE CHE PROPORGA UN OTTIMO SOFTWARE OPEN SOURCE DI GRANDE DIFFUSIONE

- IL LINGUAGGIO DI PROGRAMMAZIONE VISO SERVER **PHP** CHE CONSENTE DI ARRIVARE PAGINE WEB CON CODE SCRIPT CHE VERRANNO ESECUITI DIRETTAMENTE IN SERVER

- UN DBMS **MySQL**: DB SERVER MULTIPURPOSE E PIU' SEMPLICE DA USARE

- IL WEB SERVER **Apache**:

A COMPLETARE L'ARCHITETTURA SOFTWARE FABBISOGNA LE POSIZIONI DI UNO SOFTWARE

EFFICIENTE DI S.O. OPEN SOURCE COME LINUX NONO VERGHI COME IL SERVER

MAIUSCOLA PER L'ACQUISIZIONE HW IN DUE FAMIGLIE E DUE AZIENDE (**LAMP**)

IL LINGUAGGIO SPL LO UTILIZZANO IN MODALITA' ESECUZIONE (OSPITO IN PHP)

PER LA CREAZIONE E L'INTEROPERAZIONE DELLE TABELLE DEL NO SQL DB.

LA SEZIONE DEL SITO WEB CHE SI OCCUPA DI ALIMENTARE SECONDO UNO PREVEDENDO UNA SERIE DI PAGINE IN GRADO DI POTER GESTIRE TUTTE LE INFORMAZIONI COME ANCHE ATTIVITA' FORMATIVE.

QUANTITATIVE DI UNO ELENCO IN UNO DBA RESPONSABILE DELLA SICUREZZA E DELLA RIPULIZIONE DI BACKUP E RECOVERY DELLE INFORMAZIONI RICHIESTE

VOLENDO ELIMINARE LA UN SECONDO DEL SITO POSSO KONTROLLARE LIMITARE LA

NUOVA ATTENZIONE ALLA PROCEDURA DI INSERIMENTO DEI DATI RICHIESTI AD UN DOCCENTE REFERENTE

Form per l'acquisizione dei dati
relativi ad un docente ricercato

INSERISCI DOCCENTE

Cognome	<input type="text"/>
Nome	<input type="text"/>
Indirizzo	<input type="text"/>
Provincia	<input type="text"/>
Città	<input type="text"/>
Cap	<input type="text"/>
Cell	<input type="text"/>

BUTTON

```

<form name="form" action="insert-doc.php" method="post">
  <fieldset>
    <legend> Inserisci </legend>
    <label> Cognome: <input type="text" name="Cognome"
      size="10" maxlength="10" tabindex="1"> </label>
    <br>
    .....
  </fieldset>
  <fieldset>
    <legend> Button </legend>
    <input type="Submit" value="OK" tabindex="2">
    <input type="Reset" value="Annulla" tabindex="3">
  </fieldset>
</form>

```

file INSERT-DOC.HTM

file insert-doc.php

schema massima.

- ```

<?php
(1) $conn = mysql_connect(IPAddress, "utente", "passwd xxx") OR
 die ("Errore connessione" . mysql_error() . mysql_errno());
(2) mysql_select_db ("database") OR die ("Errore selett db");
(3) $query = "INSERT INTO Docente
 VALUES ($_POST["Cognome"], $_POST["Cognome"], ...)";
(4) $ris = mysql_query($query, $conn) OR
 die ("Errore query" . mysql_error() . mysql_errno());
(5) se $ris = TRUE allora connessione esito positivo
 altrimenti connessione esito negativo
(6) mysql_close($conn) se die ("Errore chiusura connessione" . mysql_error() . mysql_errno());
?>

```