|  |  |  |
| --- | --- | --- |
|  |  | Progetto  Musicando |
|  |  | La musica dell’estate |
| Studente  Pizzi Andrea |  |  |
|  |  |  |
| Classe  V° C ITT  Scuola/Anno  IISS. Sandro Pertini  Anno Scolastico  2021/2021 |  | Dati del Cliente  **Nome Cliente**: Associazione Culturale *“Musicando”*  **Richieste del cliente:** Creazione di una piattaforma informatica per l’evento “Estate in Musica” 2021 costituita da un sito web e un sistema informativo automatizzato tramite l’utilizzo di un database deputato a l’organizzazione di tutti i dati che necessariamente devono essere memorizzati. La piattaforma sarà articolata in un ***sito Web*** che sarà in grado di interagire direttamente con la base dati divenendone di fatto la principale interfaccia tra utente e sistema.  Il sistema principalmente dovrà permettere all’utente di acquistare dei biglietti per gli eventi musicali organizzati dall’associazione culturale che verranno consegnati in formato digitale tramite messaggio email al recapito di posta specificato al momento dell’acquisto.  Un apposito script scritto server side si occuperà della funzionalità di invio dell’email non rendendo per tanto la necessità di installazione di un apposito server email.  Il committente oltre alla creazione della piattaforma informatica richiede la creazione di una adeguata ***infrastruttura di rete*** atta a supportare la piattaforma informatica progettata e permetta di gestire gli accessi ai vari capannoni in possesso dell’associazione in cui verranno organizzati gli eventi. L’infrastruttura dovrà inoltre prevedere un sistema di monitoraggio della concentrazione del ***diossido di carbonio*** nelle varie sale al fine di evitare la saturazione dell’aria mettendo in funzione il sistema d’areazione degli impianti.  L’infrastruttura di rete sarà progettata tenendo conto di alti standard di sicurezza che garantiscano l’affidabilità della rete, la continuità del servizio e l’integrità, sicurezza e consistenza dei dati. Per garantire tutto ciò si farà ampio affidamento alle soluzioni CISCO® azienda leader nel settore delle telecomunicazioni sinonimo per tutti di qualità e affidabilità.  Sommario   1. Realizzazione Piattaforma Informatica    1. Sito Web    2. Script mail       1. La funzione PHP mail    3. Analisi del Database       1. Analisi della realtà di interesse       2. Fase di progettazione concettuale       3. Fase di progettazione Logica       4. Fase di progettazione Fisica 2. Realizzazione Infrastruttura di rete    1. Scheda apparati    2. Schema Fisico della rete    3. Schema Logico di rete    4. Piano di Indirizzamento    5. Policy di sicurezza dei dati    6. Piano di Disaster Recovery    7. Script mail   1.2.1 Script mail  Lo script PHP *“mailPHP.php*” è costituito da un codice monolitico in grado di gestire in modo autonomo l’invio di email preimpostate grazie sfruttando la configurazione e le risorse del web server che lo ospita  Complementarmente invisibile al client permette di automatizzare il processo di invio di email. Il contenuto delle email viene generato automaticamente utilizzando i dati reperiti da una base di dati MySQL tramite il modulo del linguaggio mysqli.  Lo script reperisce dal database l’indirizzo di destinazione dell’email gli eventuali valori dell’header ed il contenuto del corpo del messaggio e tenta di inoltrare il messaggio restituendo un errore nel caso di esito negativo.  L’utente finale riceverà l’email con mittente il server web su cui è ospitato il sito.   * + 1. La funzione PHP mail   Il componente centrale dello script mailPHP è la funzione predefinita del linguaggio PHP mail() che permette di inviare email formattate in testo semplice o HTML con mittente il server web su cui è configurato. La funzione accetta come parametri:   * L’indirizzo email del destinatario * L’oggetto dell’email * Il contenuto del messaggio * Eventuali dati dell’header per la formattazione del testo ed il campo CC e CCN.   Se l’inoltro del messaggio avviene con successo la funzione restituirà il valore booleano true altrimenti restituirà false.  Documentazione ufficiale di riferimento al link:  <https://www.php.net/manual/en/function.mail.php>   |  |  | | --- | --- | | Link al codice |  | | mailphp.php | <https://github.com/aluisio90/Estate-in-Musica/blob/main/home_page/negozio/mailphp.php> |  * 1. Analisi del Database   L’associazione culturale Musicando intende promuovere la campagna promozionale **“Estate in Musica”** mediante una nuova **piattaforma informatica**; gli utenti potranno utilizzare il servizio di acquisto online dei biglietti, previa registrazione al sito web, e potranno ricevere il biglietto  elettronico direttamente per email.  A tale fine, sarà necessario le seguenti informazioni: i concerti proposti nel programma sono identificati da un **codice numerico** e sono caratterizzati da **titolo** e **descrizione**. Ogni concerto è composto da un numero di pezzi musicali. Ogni pezzo è caratterizzato da un **titolo** e da uno o più autori. Inoltre, per ogni concerto, si conosce **l’ordine** in cui in cui sono rappresentati i pezzi che fanno parte del programma.  L’esecutore di ogni concerto è identificato da **matricola** ed è caratterizzato da **nome** e **stato di provenienza**. L’esecutore può essere un’orchestra oppure un solista. Nel caso delle orchestre si memorizzano il **nome del direttore** e il **numero di elementi** dell’orchestra, mentre nel caso dei solisti si considera lo **strumento suonato**. Per le orchestre, inoltre, si memorizzano i componenti (un orchestrale può suonare in una sola orchestra), identificati da **matricola univoca** all’interno dell’orchestra di appartenenza, e caratterizzati da **nome**, **data di nascita** e **strumenti suonati**.  I concerti sono tenuti in sale da concerto, identificate da **codice univoco** e caratterizzate da **nome**, **indirizzo**, **capienza massima**, **numero di posti** a sedere ed eventualmente da uno o più numeri di telefono.  Ogni concerto può essere *programmato* in una o più *date*, nella stessa sala oppure in sale diverse. Per ogni programmazione di un concerto, è noto il prezzo del biglietto, che può essere diverso per programmazioni diverse dello stesso concerto.   * + 1. Analisi della realtà di interesse   Dall’analisi della documentazione sulla realtà di interesse emerge come i concerti organizzati abbiano una certa centralità rispetto ai restanti elementi della realtà che si possono evidenziare. Tutte le informazioni organizzate si possono infatti definire strettamente correlati con i concerti.  Di ogni concerto è necessario memorizzare i dati relative al titolo accompagnato da una descrizione e un codice identificativi numero che per natura dovrà essere univoco.  Associati ai dati dei concerti si trovano associati i dati relativi ai brani che verranno eseguiti durante il concerto, riguardanti Il titolo del brano, una breve descrizione e un codice identificativo. Per ogni brano bisogna inoltra specificare l’autore o gli autori che lo hanno scritto rappresentati nella realtà da nome, cognome e codice identificativo.  Dall’analisi risulta un’altra associazione tra i concerti ed i biglietti venduti specificati da una matricola, una descrizione ed un prezzo variabile da concerto in concerto.  A loro volta i dati dei biglietti sono correlati oltre con i rispettivi concerti anche con lo spettatore che li ha comperati, specificato da un nome, un cognome, un indirizzo email, un numero di telefono ed infine un codice ID.  Per quanto riguarda le sale concerto il fatto che possano avere più numeri di telefono implica la necessita di associarli a quest’ultime tranandoli come un’entità a sé stante.  Trattati i concerti il secondo concetto più importante della base dati sono i brani che sono associati non solo con i dati degli autori che li hanno scritti ma anche con gli esecutori specificati da una matricola univoca, un nome e la nazione di provenienza.  Ogni esecutore si può a sua volta specializzare in un solista oppure in un’orchestra.  Per ogni solista bisognerà specificare inoltre lo strumento suonato, mentre invece per ogni orchestra bisognerà specificare il nome del direttore e anche il numero di elementi di cui è composto. In merito agli elementi dell‘orchestra meglio noti come orchestrali bisognerà tenere traccia di ognuno il nome, la data di nascita ed un codice identificativo.  La necessità di integrare direttamente i dati dei singoli spettatori è data dal bisogno di tener traccia dei biglietti che sono stati venduti per ogni concerto, per tanto esiste una stretta correlazione tra i biglietti e gli acquirenti e tra i biglietti e gli spettacoli   * + 1. Fase di progettazione concettuale   Entità  Dall’analisi della realtà di interesse sono risultate importanti le seguenti entità per il modellamento della realtà in oggetto.  CONCERTO: L’entità CONCERTO rappresenta nella realtà i concerti che vengono organizzati all’interno dell’evento “Un Estate in Musica”  BRANO: L’entità BRANO rappresenta all’interno della realtà di interesse un brano che viene eseguito nell’ambito di un concerto.  AUTORE: L’entità AUTORE rappresenta all’interno della realtà un autore che ha scritto uno o più brani che vengono eseguiti nei concerti.  ESECUTORE: L’entità ESECUTORE rappresenta all’interno della realtà un artista che esegue durante i concerti programmati un brano musicale.  ORCHESTRA: L’entità ORCHESTRA rappresenta all’interno della realtà di interesse una specializzazione dell’entità ESECUTORE particolare.  SOLISTA: L’entità SOLISTA rappresenta all’interno della realtà di interesse una specializzazione dell’entità ESECUTORE particolare.  ORCHESTRALE: L’entità ORCHESTRALE rappresenta all’interno della realtà di interesse il singolo componente di una orchestra.  SALA CONCERTO: L’entità SALA CONCERTO rappresenta all’interno della realtà di interesse la sala in cui vengono ospitati i concerti che vengono programmati.  NUMERO DI TELEFONO: L’entità NUMERO DI TELEFONO rappresenta nella realtà di interesse un numero telefonico di una sala concerto.  SPETTATORE: L’entità SPETTATORE rappresenta nella realtà di interesse il singolo utente/spettatore che acquista biglietti per vedere degli spettacoli.  Definite le varie entità che andranno a modellare la realtà di interesse si può adesso definire il seguente sistema informativo:  Sistema Informativa (Formato Tabellare)   |  |  | | --- | --- | | Entità | Attributi | | SPETTATORE |  | |  | Nome | |  | Cognome | |  | Email | |  | Telefono | |  | ID | | CONCERTO |  | |  | Descrizione | |  | Titolo | |  | Codice\_Concerto | |  | Prezzo | |  |  | | SALA\_CONCERTO |  | |  | Capienza\_Massima | |  | Numero\_Posti | |  | Codice\_Sala | |  | Nome | |  | Indirizzo | | TELEFONO |  | |  | Numero | |  | Tipologia | | BRANO |  | |  | Codice\_Brano | |  | Titolo | |  | Descrizione | | AUTORI |  | |  | Codice\_Autore | |  | Nome | |  | Cognome | |  | Codice\_Brano | | ESECUTORE |  | |  | Stato\_Di\_Provenienza | |  | Matricola | |  | Nome | | SOLISTA |  | |  | Strumento\_Suonato | |  | Matricola | | ORCHESTRA |  | |  | Nome\_Direttore | |  | Numero\_Di\_Elementi | |  | Matricola | | ORCHESTRALE |  | |  | Codice\_Orchestrale | |  | Nome | |  | Data\_Nascita | |  | Strumento\_Suonato |   Da una più approfondita analisi delle entità che sono state individuate emerge l’esistenza di una gerarchia ISA tra le entità ESECUTORE, SOLISTA e ORCHESTRA in quanto le entità SOLISTA e ORCHESTRA presentano di avere degli attributi comuni per mezzo dell’entità ESECUOTORE. Per tanto si stabilisce un rapporto gerarchico tra l’entità ESECUTORE detta entità Padre le entità SOLISTA ed ESECUTORE dette entità Figlie.  Le gerarchie possono essere classificate secondo due proprietà fondamentali che sono: la COPERTURA e la DISGIUNZIONE.  **La COPERTURA** indica se le istanze delle entità figlie possano o meno essere istanze anche dell’entità padre. Ogni gerarchia può presentare una:   * COPERTURA Parziale quanto alcune ma non tutte le istanze delle entità figlie appartengono all’entità padre. * COPERTURA Totale quando tutte le istanze dell’entità figlie appartengono anche all’entità padre   **La DISGIUNZIONE** indica invece se esistono delle istanze comuni tra le entità figlie.   * Si definiscono entità esclusive quando ogni istanza di una qualsiasi entità figlia appartiene unicamente all’entità da cui proviene * Si definiscono invece entità sovrapposte da quest’ultime condividono delle istanze comuni.   Nel seguente caso si può classificare la gerarchia formata da ESECUTORE (padre) e ORCHESTRA ed SOLISTA (figlie) come una gerarchia Parziale-Esclusiva in quanto le Entità SOLISTA e ORCHESTRA non possono rappresentare tutte le istanze di ESECUTORE ed un ESECUTORE può essere o un SOLISTA o un’ORCHESTRA ma non entrambe (un solista indica un musicista che non appartiene a nessuna organizzazione o gruppo musicale).  Possiamo rappresentare con uno schema ibrido (E/R ed diagramma delle Classi UML) la gerarchia nello schema concettuale. |
|  |  |  |
|  |  |  |
|  |  | Per ottenere lo schema finale del database in progettazione è necessario affrontare e trattare la gerarchia ISA in modo da ricondurre la gerarchia in associazione tra entità. Nel caso di una gerarchia Parziale-Esclusiva si aprono due possibili soluzioni per trattarla ed eliminarla.  La prima soluzione consiste nell’accorpare l’entità figlie all’entità padre, gli attributi delle figlie diventano perciò attributi dell’entità padre. Tale soluzione implica però due aspetto:  Il primo, è necessario distinguere le varie istanze dell’entità figlie ormai accorpate al padre prevedendo un apposito attributo tipo oltre a settare a NULL tutti gli attributo non necessari per quella data istanza.  Il secondo aspetto invece è detterminato proprio dal fatto che settando a NULL i valori non necessari si potrebbero creare situazioni in cui si hanno relazione con troppi valori NULL (spreco di memoria).  La secondo soluzione consiste invece nel trasformare il rapporto gerarchico tra le entità in associazioni 1:1 particolari tra le entità figlie e l’entità padre mantenendo nello schema quindi sia le entità figlie che l’entità padre. Associazioni particolare poiché nelle fase di derivazione dello schema logico saranno trattate come associazioni 1:N. Detterminante nella definizione di una tale situazione è il rapporto che intercorre tra l’entità figlie e l’entità padre e all’esclusività delle loro istanze.  Tenendo conto del numero di attributi NULL che si potrebbero ottenere trattando l’associazione con il primo caso si risolutivo si è scelto di applicare la seconda metodologia risolutiva mantenendo distinte le entità figlie e l’entità padre.  Con la seconda soluzione così si presenta lo schema concettuale finale del database in questione. |
|  |  |  |
|  |  | Relazioni  Una volta definito in modo chiaro tutte le associazioni tra le varie entità passiamo a definirle formalmente con delle “Regole di lettura”.  Concettualmente una associazione è un legame di tipo logico tra una o più entità caratterizzato dalle proprietà che sono: la partecipazione(numero minimo di istanze di y associate ad una istanza di x) e la molteplicità (numero massimo di istanze di y associate alle istanze di x e viceversa) che assieme definiscono la cardinalità dell’associazione (esprime contemperamento la molteplicità della associazione diretta e di quella inversa).  Tra le entità ASPETTATORE e CONCERTO si può stabilire una associazione Diretta-Totale inversa Totale con cardinalità complessa (N:N) tale che:  : Ogni SPETTATORE DEVE *acquistare* UNO O PIU’ BIGLIETTI;  : Ogni BIGLIETTO DEVE *essere acquistato* da UNO O PIU’ SPETTATORI;  Tra le entità CONCERTO e SALA\_CONCERTO si può stabilire un’associazione Diretta-Totale ed Inversa-Parziale Totale con cardinalità complessa (N:N) tale che:  : Per Ogni CONCERTO SI DEVE *prenotare* UNA O PIU’ SALE\_CONCERTO;  : Ogni SALA\_CONCERTO DEVE *essere prenotata* da UNO O PIU’ CONCERTI  Tra le entità SALA\_CONCERTO e TELEFONO si può stabilire una associazione Diretta-Totale e Inversa-Parziale con molteplicità semplice (1:N) tale che:  : Ogni SALA\_CONCERTO DEVE possedere uno o più TELEFONI  : Ogni TELEFONO DEVE essere posseduto da UNA O PIU’ SALE CONCERTI.  Tra le entità CONCERTO e BRANO si può stabilire un’associazione Diretta-Totale e Inversa-Totale con molteplicità complessa (N:N) tale che:  : Per Ogni CONCERTO DEVE programmare UNO O PIU’ BRANI;  : Ogni BRANO DEVE essere programmato DA UNO O PIU’ CONCERTI  Tra le entità BRANO e AUTORE si può stabilire un’associazione Diretta-Totale e Inversa-Parziale con cardinalità complessa (N:N) tale che:  : Ogni AUTORE DEVE scrivere UNO O PIU’ BRANI;  : Ogni BRANO DEVE essere SCRITTO da UNO O PIU’ AUTORI;  Tra le entità BRANO e ESECUTORE si può stabilire un’associazione Diretta-Totale e Inversa Parziale con molteplicità complessa (N:N) Tale che  : Ogni ESECUTORE DEVE suonare UNO O PIU’ BRANI;  : Ogni BRANO DEVE essere eseguito da UNO O PIU’ ESECUTORI;  Tra le entità ESECUTORE e SOLISTA si può stabilire una associazione Diretta-Parziale e Inversa Totale con molteplicità bi-univoca tale che:  : Ogni ESECUTORE PUO’ essere Un SOLISTA;  : Ogni SOLISTA DEVE essere un ESECUTORE;  Tra le entità ESECUTORE e ORCHESTRA si può stabilire un’associazione Diretta-Parziale e Inversa Totale con molteplicità bi-univoca tale che:  : Ogni ESECUTORE PUO’ essere Un ORCHESTRA;  : Ogni ORCHESTRA DEVE essere un ESECUTORE;  Tra le entità ORCHESTRA e ORCHESTRALE si può stabilire un’associazione Diretta-Totale e Inversa-Totale con molteplicità semplice (1:N) tale che:  : Ad Ogni ORCHESTRA DEVE appartenere UNO O PIU’ ORCHESTRALI;  : Ogni ORCHESTRALE DEVE appartenere ad un’ORCHESTRA;    Vincoli di Integrità  Trattiamo adesso i vincoli di integrità sulle istanze delle entità che costituiscono lo schema concettuale definito.  I vincoli di integrità sono delle asserzioni che possono risultare **VERE** oppure **FALSE** rispetto ad una istanza di una entità.  Se un’asserzione risulta VERA allo significa che l’istanza soddisfa i vincoli imposti se invece l’asserzione risulta FALSA allora significa che l’istanza viola uno o più vincoli.  Nel momento in cui si creano le istanze dell’entità e si valorizzano noi accetteremo solamente quelle che non violano nessun vincolo di integrità.  I vincoli di integrità si suddividono in:  **Impliciti**  **Espliciti**.  I Vincoli **Impliciti** sono imposti dalla struttura dei dati e si suddividono alloro volta in:  Vincoli di chiave primaria (PK)  Vincoli di Relazione  I vincoli di chiave primaria sono indicati sul diagramma E/R da gli attributi sottolineati  I vincoli di relazione invece impongono che: “Date due Entità X e Y è un’associazione A tra esse, si impone che non si possa inserire un elemento dell’entità X che non sia associato ad un elemento dell’entità Y. Tale Vincoli si rappresenta sul diagramma E/R con linee continue e discontinue (la partecipazione).  I vincoli Espliciti invece definiscono il modo in cui i valori degli attributi variano; tali vincoli si esprimono con la seguente sintassi:  Dall’analisi è risultato necessario definire i seguenti vincoli espliciti.  1.1 Sito Web  L’associazione Culturale Musicando dispone attualmente di server fisico ospitato all’interno di una piccola sala server locata all’interno dell’edificio amministrativo del parco di proprietà dell’associazione Musicando in via XXX.  Il server disponeva al momento dell’avvio di un sistema operativo installato appartenente alle distribuzioni della famiglia GNU – Linux, Ubuntu Server (ver.21.04).  E’ stato necessario configurare sul medesimo server un web server per offrire il servizio web (WWW); è stato scelto come web server Apache HTTP Server. Dato i numerosi moduli di cui dispone è in grado di interfacciarsi con molteplici sistemi in questo caso particolare con il sistema informatico dell’associazione Musicando costituito fisicamente da un altro server dedicato su cui è installata un’istanza del DBMS mariaDB deputato alla gestione della base dati di riferimento di tutta l’organizzazione. La comunicazione tra server web e server MySQL avviene mediante il linguaggio PHP (utilizzabile configurando l’apposito modulo di Apache) ed il modulo mysqli offerto sempre dal linguaggio PHP.  Il server web ospita per quest’anno il sito web di un progetto offerto dall’associazione Culturale musicando, “Estate in Musica 2021“. Il sito web fungerà oltre che da principale mezzo informativo e promozionale dell’evento anche da portare per l’acquisto dei biglietti dei vari eventi organizzati dall’associazione nei capannoni che ha disposizione al Parco Musicando.  Gli utenti del sito potranno quindi sia visionare le informazioni sugli eventi organizzati (compreso il programma della serata) che acquistare i biglietti collegandosi al sito  <https://estateinmusica.altervista.org/> (link del sito)  Il portale web è organizzato in quattro sezioni distinte:   * Home (<https://estateinmusica.altervista.org/>) * Negozio (<https://estateinmusica.altervista.org/negozio> ) * Contatti (<https://estateinmusica.altervista.org/contatti> ) * Login (<https://estateinmusica.altervista.org/login> )   Nella sezione Home gli utenti troveranno le principali informazioni sull’evento stesso e sui vari spettacoli che si organizzeranno. I membri dello staff Musicando potranno comunicare con gli utenti del sito tramite appositi post visibili dalla home page.  Alla sezione negozio invece gli utenti potranno prendere visione dei vari eventi organizzati con la possibilità di vedere il programma di ogni concerto premendo sull’bottone “Scopri il Programma” oppure acquistare direttamente il biglietto per l’evento premendo sul bottone “Acquista Adesso”. |
|  |  |  |
|  |  | * + 1. Fase di progettazione Logica   Definiamo adesso lo schema logico relazionale del database applicando le regole di derivazione allo schema concettuale precedentemente ottenuto.  Schema Logico della base di dati  SPETTATORI (Nome, Cognome, Email, Telefono, ID);  CONCERTI (Descrizione, Titolo, Codice\_Concerto);  BIGLIETTI (Matricola, Descrizione, ID, Codice\_Concerto, Prezzo);  SALE\_CONCERTI (Capienza\_Massima, Numero\_Posti, Codice\_Sala, Nome, Indirizzo);  *PRENOTATE* (Codice\_Concerto, Data\_Prenotazione, Codice\_Sala);  TELEFONI (Numero, Tipologia, Codice\_Sala);  BRANI (Codice\_Brano, Titolo, Descrizione);  *PROGRAMMATI* (Ordine, Codice\_Concerto, Codice\_Brano);  AUTORI (Codice\_Autore, Nome, Cognome);  *SCRIVONO* (Codice\_Autore, Codice\_Brano);  ESECUTORI (Stato\_Di\_Provenienza, Matricola, Nome);  SOLISTI (Strumento\_Suonato, Matricola);  ORCHESTRE (Nome\_Direttore, Numero\_Di\_Elementi, Matricola);  ORCHESTRALI (Codice\_Orchestrale, Nome, Data\_Nascita, **Matricola,** Strumento\_Suonato);  *SUONANO* (Matricola, Codice\_Brano); |
|  |  | Schema logico della base dati formato tabellare |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Entità | | | Attributi | Chiavi | formato | Dimensione  (byte) | |
| SPETTATORI | | |  |  |  |  | |
|  | | | Nome |  | Stringa | 45 | |
|  | | | Cognome |  | Stringa | 45 | |
|  | | | Email |  | Stringa | 254 | |
|  | | | Telefono |  | Stringa | 15 | |
|  | | | ID | PK | Intera | 15 | |
| CONCERTI | | |  |  |  |  | |
|  | | | Descrizione |  | Stringa | 50 | |
|  | | | Titolo |  | Stringa | 25 | |
|  | | | Codice\_Concerto | PK | Intero | 8 | |
|  | | | Prezzo |  | Decimale(float) | 4 | |
| ACQUISTANO | | |  |  |  |  | |
|  | | | Matricola | PK | Intero | 15 | |
|  | | | Tipo |  | Stringa | / | |
|  | | | ID | FK | Intera | 15 | |
|  | | | Codice\_Concerto | FK | Intero | 8 | |
|  | | | Prezzo |  | Decimale(float) | 4 | |
| SALA\_CONCERTI | | |  |  |  |  | |
|  | | | Capienza\_Massima |  | Intero | 10 | |
|  | | | Numero\_Posti |  | Intero | 10 | |
|  | | | Codice\_Sala | PK | Intero | 2 | |
|  | | | Nome |  | Stringa | 45 | |
|  | | | Indirizzo |  | Stringa | 45 | |
| PRENOTATE | | |  |  |  |  | |
|  | | | Codice\_Concerto | FK | Intero | 8 | |
|  | | | Data\_Prenotazione | PK | Data/Ora |  | |
|  | | | Codice\_Sala | PK – FK | Intero | 2 | |
| TELEFONI | | |  |  |  |  | |
|  | | | Numero | PK | Stringa | 15 | |
|  | | | Tipologia |  | Stringa | 10 | |
|  | | | Codice\_Sala | FK | Intero | 2 | |
| BRANI | | |  |  |  |  | |
|  | | | Codice\_Brano | PK | Intero | 8 | |
|  | | | Titolo |  | Stringa | 45 | |
|  | | | Descrizione |  | Stringa | 50 | |
| PROGRAMMATI | | |  |  |  |  | |
|  | | | Ordine | PK | Intero | AUTO | |
|  | | | Codice\_Concerto | FK | Intero | 8 | |
|  | | | Codice\_Brano | FK - PK | Intero | 8 | |
| AUTORI | | |  |  |  |  | |
|  | | | Codice\_Autore | PK | Intero | 4 | |
|  | | | Nome |  | Stringa | 25 | |
|  | | | Cognome |  | Stringa | 25 | |
| SCRIVONO | | |  |  |  |  | |
|  | | | Codice\_Autore | FK – PK | Intero | 4 | |
|  | | | Codice\_Brano | FK - PK | Intero | 2 | |
| ESECUTORI | | |  |  |  |  | |
|  | | | Stato\_Di\_Provenienza |  | Stringa | 25 | |
|  | | | Matricola | PK | Intero | 4 | |
|  | | | Nome |  | Stringa | 25 | |
| SOLISTI | | |  |  |  |  | |
|  | | | Strumento\_Suonato |  | Stringa | 45 | |
|  | | | Matricola | FK - PK | Intero | 4 | |
| ORCHESTRE | | |  |  |  |  | |
|  | | | Nome\_Direttore |  | Stringa | 50 | |
|  | | | Numero\_Di\_Elementi |  | Intero | 7 | |
|  | | | Matricola | FK - PK | Intero | 4 | |
| ORCHESTRALI | | |  |  |  |  | |
|  | | | Codice\_Orchestrale | PK | Intero | 4 | |
|  | | | Nome |  | Stringa | 50 | |
|  | | | Data\_Nascita |  | Data |  | |
|  | | | Matricola | FK | Intero | 4 | |
|  | | | Strumento\_Suonato |  | Stringa | 25 | |
| SUONANO | | |  |  |  |  | |
|  | | | Matricola | FK - PK | Intero | 4 | |
|  | | | Codice\_Brano | FK - PK | Intero | 8 | |
|  |  |  | | | | |
|  |  | Verifica della qualità dello schema logico relazionale  Prima di procedere è necessario verificare la qualità delle relazioni, e quindi la qualità complessiva, dello schema logico per verificare se quest’ultimo soddisfi detterminati standard di qualità.  La normalizzazione ha come scopo quello di fornire un metodo che consenta di verificare e quindi progettare un DB senza anomalia ossia costituito da relazioni corrette.  Detto ciò è importante precisare che la normalizzazione rappresenta unicamente un metodo di verifica dei risultati della progettazione e non costituisce in nessun caso una metodologia di programmazione.  Le tabelle si definiscono NON corette quando sono afflitte delle seguenti anomalia:  • Anomalie di inserimento  • Anomalie di Cancellazione  • Anomalie di aggiornamento  Nel caso in cui una o più tabelle presentino tali anomalie allora si intraprenderà un processo di normalizzazione che partendo da una relazione non normalizzata la ottimizza fino a raggiungere un certo grado di normalizzazione.  Il processo consiste nel verificare se lo schema logico rispetti le forme normali delle tabelle relazionali, cioè una serie di proprietà che garantiscano l’assenza di anomalie. Per quanto riguarda questo pecifico caso andremmo a verificare se lo schema logico rispetti la 1FN, 2FN e 3FN.  **1FN:** Possiamo verificare che tutte le relazioni sono in forma normale in quanto gli attributi di ogni relazione sono atomici e ogni relazione possiede una chiave (PK)  **2FN:** Possiamo verificare che tutte le relazioni sono in 2FN in quanto sono in 1FN e tutti gli attributi non chiave di ogni relazione dipendono funzionalmente e completamente dalle rispettive chiavi primarie.  **3FN**: Possiamo verificare che tutte le relazioni sono in 3FN in quanto sono in 2FN è tutti gli attributi non chiave di ogni relazione dipendono direttamente dalle rispettive chiavi primarie.   * + 1. Fase di progettazione Fisica   Il sistema informatico dell’associazione Culturale Musicando è costituito da un server fisico dedicato su cui è configurata un’istanza del DBMS MySQL (Oracle) configurato con il motore MyISAM per la gestione fisica dei dati (operazioni di scrittura, lettura e aggiornamento).  Query  Utilizzando il linguaggio SQL (Standard Query Language) come DDL (Data Definition Language) definiamo la struttura delle tabelle (campi e tipo). | | | | |
|  |  | CREATE DATABASE my\_estateinfesta;  /\*...\*/  /\*Dichiarazione delle tabelle\*/  /\*CREAZIONE DELLA TABELLA SPETTATORI\*/  CREATE TABLE SPETTATORI (      Nome VARCHAR(45) NOT NULL,      Cognome VARCHAR(45) NOT NULL,      Email VARCHAR(100) UNIQUE,      Telefono VARCHAR(15) UNIQUE,      ID INT(15) PRIMARY KEY AUTO\_INCREMENT    );  /\*CREAZIONE DELLA TABELLA CONCERTI\*/  CREATE TABLE CONCERTI(      Descrizione VARCHAR(50) DEFAULT ' ',      Titolo VARCHAR(25) NOT NULL,      Codice\_Concerto INT(8)PRIMARY KEY      Prezzo FLOAT(4) NOT NULL,        CHECK Prezzo > 0  );  /\*CREAZIONE DELLA TABELLA BIGLIETTI\*/  CREATE TABLE ACQUISTANO(      Matricola INT(2) AUTO\_INCREMENT PRIMARY KEY,      Tipo SET('Ridotto', 'Intero') NOT NULL DEFAULT 'Intero',      ID INT(15) NOT NULL,      Codice\_Concerto INT(8) NOT NULL,        FOREIGN KEY (Codice\_Concerto) REFERENCES CONCERTI(Codice\_Concerto),      FOREIGN KEY (ID) REFERENCES SPETTATORI(Codice\_Concerto)  );  /\* CREAZIONE TABELLE CONCERTI\*/  CREATE TABLE SALE\_CONCERTI(      Capienza\_Massima INT(10) NOT NULL,      Numero\_Posti INT(10) NOT NULL,      Codice\_Sala INT(2) PRIMARY KEY,      Nome VARCHAR(45) NOT NULL,      Indirizzo VARCHAR(45) NOT NULL  );  /\*CREAZIONE TABELLA PRENOTAZIONI\*/  CREATE TABLE PRENOTATE(      Codice\_Concerto INT(8)NOT NULL,      Data\_Prenotazione DATETIME NOT NULL,      Codice\_Sala INT(2) NOT NULL,      PRIMARY KEY(Codice\_Concerto, Codice\_Sala),      FOREIGN KEY(Codice\_Concerto) REFERENCES CONCERTI(Codice\_Concerto) ON DELETE CASCADE,      FOREIGN KEY(Codice\_Sala) REFERENCES SALE\_CONCERTI(Codice\_Sala)ON DELETE CASCADE          );  /\*CREAZIONE TABELLLA TELEFONI\*/  CREATE TABLE TELEFONI(      Numero VARCHAR(15) PRIMARY KEY,      Tipologia VARCHAR(10) DEFAULT ' ',      Codice\_Sala INT(2) NOT NULL,        FOREIGN KEY (Codice\_Sala) REFERENCES SALE\_CONCERTI(Codice\_Sala)      ON UPDATE CASCADE  );  /\*CREAZIONE TABELLA BRANI\*/  CREATE TABLE BRANI(      Codice\_Brano INT(8) PRIMARY KEY,      Titolo VARCHAR(45) NOT NULL,      Descrizione VARCHAR(50) DEFAULT ' '    );  /\*CREAZIONE TABELLA PROGRAMMATI\*/  CREATE TABLE PROGRAMMATI(      Ordine INT(4) NOT NULL,      Codice\_Concerto INT(8) NOT NULL,      Codice\_Brano INT(8)NOT NULL,        PRIMARY KEY( Codice\_Concerto, Ordine),        FOREIGN KEY (Codice\_Concerto) REFERENCES CONCERTI(Codice\_Concerto) ON UPDATE CASCADE,      FOREIGN KEY (Codice\_Brano) REFERENCES BRANI(Codice\_Brano) ON DELETE CASCADE        );  /\*CREAZIONE TABELLA AUTORI\*/  CREATE TABLE AUTORI(      Codice\_Autore INT(4) PRIMARY KEY,      Nome VARCHAR(25) NOT NULL,      Cognome VARCHAR(25) NOT NULL  );  /\*CREAZIONE DELLA TABELLA SCRIVONO\*/  CREATE TABLE SCRIVONO(      Codice\_Autore INT(4),      Codice\_Brano INT (2),        PRIMARY KEY(Codice\_Brano, Codice\_Autore),    FOREIGN KEY(Codice\_Autore) REFERENCES AUTORI(Codice\_Autore) ON UPDATE CASCADE,      FOREIGN KEY(Codice\_Brano) REFERENCES BRANI(Codice\_Brano) ON UPDATE CASCADE          );  /\*CREAZIONE TABELLA ESECUTORI\*/  CREATE TABLE ESECUTORI(      Stato\_Di\_Provenienza VARCHAR(25) NOT NULL,      Matricola INT(4) PRIMARY KEY,      Nome VARCHAR(25) NOT NULL      );  /\*CREAZIONE TABELLA SOLISTA\*/  CREATE TABLE SOLISTI(      Strumento\_Suonato VARCHAR(45),      Matricola INT(4) PRIMARY KEY,        FOREIGN KEY (Matricola) REFERENCES ESECUTORI(Matricola) ON DELETE CASCADE ON UPDATE CASCADE      );  /\*CREAZIONE TABELLA ORCHESTRA\*/  CREATE TABLE ORCHESTRE(      Nome\_Direttore VARCHAR(50) NOT NULL,      Numero\_Di\_Elementi TINYINT DEFAULT 0,      Matricola INT(4) PRIMARY KEY,        FOREIGN KEY(Matricola) REFERENCES ESECUTORI(Matricola) ON DELETE CASCADE ON UPDATE CASCADE    );  /\*CREAZIONE TABELLA ORCHESTRA\*/  CREATE TABLE ORCHESTRALI(      Codice\_Orchestrale INT(4) PRIMARY KEY,      Nome VARCHAR(50),      Data\_Nascita DATE,      Matricola INT(4) NOT NULL,      Strumento\_Suonato VARCHAR(25) NOT NULL,      FOREIGN KEY(Matricola) REFERENCES ORCHESTRE(Matricola)    );  /\*CREAZIONE TABELLA SUONANO\*/  CREATE TABLE SUONANO(      Matricola INT(4),      Codice\_Brano INT(8),        PRIMARY KEY(Matricola, Codice\_Brano),      FOREIGN KEY(Matricola) REFERENCES ESECUTORI(Matricola) ON DELETE CASCADE,      FOREIGN KEY(Codice\_Brano) REFERENCES BRANI(Codice\_Brano) ON UPDATE CASCADE      ); | | | | |
|  |  | Utilizzando il linguaggio SQL (Standard Query Language) come DML (Data Manipulation Language) inseriamo un set dati di test (campi e tipo).  INSERT INTO `AUTORI` (`Codice\_Autore`, `Nome`, `Cognome`)  VALUES  (1, 'Vasco Rossi', 'Mario'),  (2, 'Rubla', 'Carrisi'),  (3, 'Tina', 'Mancini'),  (4, 'Jhoana', 'Russo'),  (5, 'Giovanni', 'Paganini'),  (6, 'Carlo', 'Paganini'),  (7, 'Umberto', 'Terzani'),  (8, 'Attilio', 'Fontana'),  (90, 'Gabriel', 'De Ferrera'); | | | | |
|  |  | INSERT INTO `BRANI` (`Codice\_Brano`, `Titolo`, `Descrizione`) VALUES  (1, 'Viva La Vita', 'Musica pop'),  (2, 'Il Silenzio del cuore', 'Musica Sentimentale'),  (3, 'Forse ti Amo', 'Musica Storica'),  (4, 'La Marcia Imperiale', 'Musica Classica'),  (6, 'Nona Sinfonia', ' Musica Classica'),  (7, 'La Primavera', ' Musica Classica'),  (8, 'L''autunno', ' Musica Classica'),  (9, 'Titanic', 'Colonna Sonora'),  (20, 'Come il domani', 'Musica folk'),  (21, 'Meglio di Te', 'Musica Techno'),  (22, 'Gioca con Me', ' Musica Regge '),  (23, 'Datemi un Martello', 'Musica disco'),  (24, 'She is', ' Musica Tradizionale'),  (25, 'Concerto Largo Adagio', ' Componimento Sinfonico Musicale'),  (26, 'E quest ed il mare', 'Musica Popolare'),  (27, 'Come non dirtelo', ' Musica Pop'),  (28, 'Boleno', 'Musica Latina'),  (29, 'Laura non c''e', 'Musica Pop'),  (30, 'Jhonny va via', ' Musica Etnica'),  (31, 'Fiore di Rosa', ' Musica Tradizionale'),  (32, 'La Ballata di Primavera', ' Musica Tradizionale'),  (33, 'Hola', ' Musica Latina');  INSERT INTO `CONCERTI`  (`Descrizione`, `Titolo`, `Codice\_Concerto`, `Prezzo`)  VALUES  (' Opera Lirica in Tre Atti', 'Le Nozze di Figaro', 1, 45),  ('Suonata Rock per appasionati', 'Rock in Fiesta', 2, 25),  ('Opera Lirica tipica della produzione Verdiana', 'Aida', 3, 50),  ('Minuetto nottorno', 'Al Chiaro Di Luna', 4, 25),  ('Esecuzione della banda di Castel Verde', 'Parata annuale di Castel ', 5, 47);  INSERT INTO `ESECUTORI` (`Stato\_Di\_Provenienza`, `Matricola`, `Nome`) VALUES  ('Italia', 2374, 'Orchestra Reale di Napoli'),  ('Italia', 2353, 'Orchestra Reale di Sardeg'),  ('Italia', 8739, 'Conservatori Santa Cecili'),  ('Francia', 7685, 'Orchestra privata di Pari'),  ('USA', 98347, 'Ledy Gaga'),  ('Belgio', 8765, 'Madame De Boy'),  ('Olanda', 654, 'Federich Ozlands'),  ('USA', 8757, 'Jo Di Maggio'),  ('Cina', 8756, 'Zoe Tzu'),  ('Cina', 7645, 'Dorotea Mao');  INSERT INTO `ORCHESTRALI` (`Codice\_Orchestrale`, `Nome`, `Data\_Nascita`, `Matricola`, `Strumento\_Suonato`) VALUES  (1, 'Anna', '2012-02-17', 100, 'Violino'),  (2, 'Raul', '2021-05-24', 80, 'Chitarra'),  (3, 'Tina', '2021-05-28', 120, 'Pianoforte'),  (4, 'Lucia', '2021-05-02', 40, 'Batteria'),  (5, 'Giovanni', '2021-05-04', 40, 'Tromba');  INSERT INTO `ORCHESTRE` (`Nome\_Direttore`, `Numero\_Di\_Elementi`, `Matricola`) VALUES  ('Anna Kerinova', 40, 2374),  ('Luigi Pierbasso', 100, 2353),  ('Karl Kolliman', 120, 8793),  ('Luise Bretèn', 80, 7685);  INSERT INTO `PRENOTATE` (`Codice\_Concerto`, `Data\_Prenotazione`, `Codice\_Sala`) VALUES  (2, '2021-05-21', 1),  (2, '2021-05-21', 2),  (2, '2021-05-21', 4),  (4, '2021-02-01', 3),  (5, '2021-05-09', 1),  (5, '2021-05-09', 2),  (6, '2021-05-07', 2),  (7, '2021-05-02', 3),  (7, '2021-07-12', 1),  (8, '2021-05-02', 3),  (9, '2021-01-04', 6),  (19, '2021-07-06', 34),  (20, '2021-04-09', 3);  INSERT INTO `PROGRAMMATI` (`Ordine`, `Codice\_Concerto`, `Codice\_Brano`) VALUES  (1, 5, 4),  (1, 7, 6),  (1, 1, 2);  INSERT INTO `SALE\_CONCERTI` (`Capienza\_Massima`, `Numero\_Posti`, `Codice\_Sala`, `Nome`, `Indirizzo`) VALUES  (1000, 1000, 3, 'Capannone\_2', 'Via XXX N.20'),  (750, 750, 34, 'Capannone\_1', 'Via XXX N.20'),  (500, 500, 1, 'Capannone\_3', 'Via XXX N.20'),  (900, 900, 6, 'Capannone\_4', 'Via XXX N.20');  INSERT INTO `SCRIVONO` (`Codice\_Autore`, `Codice\_Brano`)  VALUES  (1, 1),  (2, 1),  (1, 2),  (4, 3),  (4, 4),  (20, 4),  (20, 5),  (8, 7),  (9, 9),  (5, 20),  (6, 24),  (90, 25),  (2, 29),  (4, 32),  (90, 33);  INSERT INTO `SOLISTI` (`Strumento\_Suonato`, `Matricola`)  VALUES  ('Chitarra ', 98347),  ('Arpa', 8765),  ('Violino', 654),  ('Chitarra', 8756),  ('Pianoforte', 8757),  ('Batteria', 7645);  INSERT INTO `SUONANO` (`Matricola`, `Codice\_Brano`) VALUES  (3, 3),  (6, 21),  (654, 0),  (2353, 5),  (2374, 1),  (7645, 2),  (8739, 6),  (8756, 0),  (8757, 1),  (8757, 4),  (98347, 7),  (98347, 20);  INSERT INTO `TELEFONI` (`Numero`, `Tipologia`, `Codice\_Sala`) VALUES  ('337 888 9541', 'Segreteria', 34),  ('987 547 5226', 'Reception', 6),  ('179 456 125', 'Interno Sa', 34),  ('778 859 6653', 'Segreteria', 1),  ('126 98 5691', 'Principale', 1),  ('286 236 4527', 'Principale', 34),  ('777 888 6594', 'Segreteria', 34),  ('445 9886 124', 'Emergenza ', 3),  ('335 9998 4571', ' Reception', 1),  ('746 985 1257', ' Numero Un', 6);  Utilizzando il linguaggio SQL (Standard Query Language) come QL (Query Language) per definire le interrogazioni (campi e tipo).  /\* (1) \*/  /\*Elencare per data degli eventi i biglietti acquistati da un utente(AREA RISERVATA)\*/  SELECT \*  FROM  SPETTATORI,  ACQUISTANO,  CONCERTI,  PRENOTATE,  SALE\_CONCERTI  WHERE  ACQUISTANO.ID = SPETTATORI.ID AND ACQUISTANO.Codice\_Concerto = CONCERTI.Codice\_Concerto AND  PRENOTATE.Codice\_Concerto = CONCERTI.Codice\_Concerto AND PRENOTATE.Codice\_Sala = SALE\_CONCERTI.Codice\_Sala AND  SPETTATORI.ID = ['Inserire ID'];  /\* (2) \*/  /\* Elencare in una scheda informativa il programma di un dato concerto elecando quindi:     - La lista dei brani ordinata secondo l'ordine di esecuzione     - Gli esecutori dei brani  \*/  SELECT  BRANI.Titolo,  PROGRAMMATI.Ordine,  BRANI.Descrizione,  ESECUTORI.Nome AS nome\_esecutore,  AUTORI.Nome AS nome\_autore,  AUTORI.Cognome AS cognome\_autore  FROM  BRANI,  SCRIVONO,  AUTORI,  SUONANO,  ESECUTORI,  PROGRAMMATI,  CONCERTI  WHERE  BRANI.Codice\_Brano = SCRIVONO.Codice\_Brano AND AUTORI.Codice\_Autore = SCRIVONO.Codice\_Autore AND  BRANI.Codice\_Brano = SUONANO.Codice\_Brano AND ESECUTORI.Matricola = SUONANO.Matricola AND  PROGRAMMATI.Codice\_Brano = BRANI.Codice\_Brano AND PROGRAMMATI.Codice\_Concerto = CONCERTI.Codice\_Concerto AND  CONCERTI.Codice\_Concerto = ['Inserire codice\_concerti'];  /\*(3)\*/  /\*Disporre in una scheda informativa i dati di un dato musicista e/o orchestra\*/     SELECT \* FROM ESECUTORE, ORCHESTRA, SOLISTA       WHERE     ESECUTORE.Matricola = ORCHESTRA.Matricola AND     ESECUTORE.Matricola = SOLISTA.Matricola AND     ESECUTORE.Matricola = ['inserire matricola'];  /\*(4)\*/  /\*  Organizzare uno scheda informativa con i dati di tutti gli eventi programmati specificando:     Data e ora dell'evento     Luogo dell'evento e via     uno o piu' numeri di riferimento     una breve descrizione dell'evento (se disponibile)     Numero di posti ancora disponibile (calcolato sulla base dei biglietti gia' venduti)  \*/  SELECT            PRENOTATE.Data\_Prenotazione,            SALE\_CONCERTI.Numero\_Posti,            SALE\_CONCERTI.Nome,            SALE\_CONCERTI.Indirizzo,            CONCERTI.Titolo,            CONCERTI.Descrizione,            CONCERTI.Prezzo,            CONCERTI.Codice\_Concerto            FROM            PRENOTATE,            SALE\_CONCERTI,            CONCERTI            WHERE            PRENOTATE.Codice\_Sala = SALE\_CONCERTI.Codice\_Sala AND            PRENOTATE.Codice\_Concerto = CONCERTI.Codice\_Concerto; | | | | |

|  |  |  |
| --- | --- | --- |
|  |  | Giornale della scuola |
|  |  |  |
| Dott.ssa Tanaka  (718) 555 0100  sycamoremiddle.org  ytanaka@sycamoremiddle.edu |  | scuola elementare |
|  |  |  |
| Eventi in programma  Scuola di danza  09/09 | 17:00-19:00  Fiera del libro  01/10 - 01/11  Foto di classe  15/10 |  | Bentornati tra i banchi di scuola!  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam non varius ipsum. Fusce venenatis quam vel libero scelerisque, a mattis diam sollicitudin. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam non varius ipsum. Fusce venenatis quam vel libero sce lerisque, a mattis diam sollicitudin.  Maecenas massa velit, consectetur sit amet da pibus et, dapibus sit amet odio. Vestibulum ultric ies erat sit amet massa ullamcorper ferm entum. Quisque a suscipit nulla, non posuere tellus. Vivamus justo erat, suscipit sit amet ultr icie. Ma ecenas massa velit, consectetur sit amet dapibus et, dapibus sit amet odio. Vestibulum ultricies erat sit amet massa ullamcorper.  Lorem ipsum dolor sit amet, consectetur adipi scing elit. Aliquam non varius ipsum. Fusce ven enatis quam vel libero scelerisque, a mattis diam sollicitudin. Lorem ipsum dolor sit amet, cons ectetur adipiscing elit. Aliquam non varius ipsum. Fusce venenatis quam vel libero sce lerisque, a mattis diam sollicitudin. Quisque a suscipit nulla, non posuere tellus. |

|  |  |  |
| --- | --- | --- |
|  |  | Elementi di rilievo recenti  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam non varius ipsum. Fusce venenatis quam vel libero scelerisque, a mattis diam sollicitudin. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam non varius ipsum. Fusce venenatis quam vel libero scelerisque, a mattis diam sollicitudin.  Maecenas massa velit, consectetur sit amet dapibus et, dapibus sit amet odio. Vestibulum ultricies erat sit amet massa ullamco rper ferme ntum. Quisque a suscipit nulla, non posuere tellus. Viva mus justo erat, suscipit sit amet dui sed, ultricies porttitor mi.  Aliquam pharetra risus vitae odio imperdiet condimentum. Fusce in quam varius, dictum mi sit amet, aliquam mi. Suspe ndisse potenti. Pellentesque auctor id neque vel maximus. |
| Elementi di rilievo autunnali  Lorem ipsum dolor sit amet, consec tetur adi piscing elit. Aliquam non varius ipsum. Fusce venenatis quam vel libero scelerisque. Quam vel libero sce lerisque, a matm.  Lorem ipsum dolor sit amet, consec tetur adi piscing elit. Aliquam non varius ipsum. Fusce venenatis quam vel libero scelerisque. Quam vel libero sce lerisque, a mattis diam venenatis quam vel. |  | banchi in una classe |
|  |  |
|  | La nostra prossima area di focus  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ma ecenas massa velit, consectetur sit amet dapibus et, dapibus sit amet odio. Aliquam non varius ipsum. Fusce venenatis quam vel libero scelerisque, a mattis diam sollicitudin. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Non varius ipsum. Fusce venenatis quam vel libero sce lerisque, a mattis.  Maecenas massa velit, consectetur sit amet da pibus et, dapibus sit amet odio. Vestibulum ultric ies erat sit amet massa ullamcorper ferm entum. Quisque a suscipit nulla, non posuere tellus. Vivamus justo erat, suscipit sit amet ultr icie. Ma ecenas massa velit, consectetur sit amet dapibus et, dapibus sit amet odio. Vestibulum ultricies erat sit amet massa ullamcorper. nulla, non posuere tellus. Vivamus justo erat, suscipit sit amet dui sed, ultricie. Ma ecenas massa velit. |