

Ministero dell'Istruzione, dell'Università e della Ricerca

I044 – ESAME DI STATO DI ISTRUZIONE SECONDARIA SUPERIORE

Indirizzo: ITIA - INFORMATICA E TELECOMUNICAZIONI ARTICOLAZIONE INFORMATICA

Tema di: INFORMATICA - Tipologia C

Il candidato (che potrà eventualmente avvalersi delle conoscenze e competenze maturate attraverso esperienze di alternanza scuola-lavoro, stage o formazione in azienda) svolga la prima parte della prova e due tra i quesiti proposti nella seconda parte.

PRIMA PARTE

Un'azienda start-up vuole costruire una piattaforma Web che consenta il car pooling tra viaggiatori sul territorio nazionale, con l'obiettivo di diffondere l'uso di una mobilità flessibile e personalizzata in termini di percorsi e costi.

Gli utenti della piattaforma possono essere di due tipi: utenti-autisti (coloro che offrono un passaggio con la propria macchina) e utenti-passeggeri (coloro che usufruiscono del passaggio).

Gli autisti devono registrarsi sul sito ed inserire i propri dati: generalità, numero e scadenza patente di guida, dati dell'automobile utilizzata, recapito telefonico, email, fotografia.

Per ogni viaggio che intendono condividere, gli autisti devono indicare città di partenza, città di destinazione, data ed ora di partenza, contributo economico richiesto ad ogni passeggero, tempi di percorrenza stimati. È responsabilità dell'autista, mano a mano che accetterà passeggeri per un certo viaggio, dichiarare chiuse le prenotazioni per quel viaggio, utilizzando un'apposita funzione sul portale.

L'utente-passeggero si deve registrare sulla piattaforma, indicando cognome e nome, documento di identità, recapito telefonico ed email. La piattaforma fornisce ai passeggeri la possibilità di indicare città di partenza e di destinazione e data desiderata; presenta quindi un elenco di viaggi (per cui non siano ancora chiuse le prenotazioni), ciascuno con le caratteristiche dell'autista e le modalità del viaggio stesso inserite dall'autista (orario, eventuali soste previste alle stazioni di servizio, possibilità di caricare bagaglio o animali, ...).

Il passeggero sceglie quindi il viaggio desiderato con il corrispondente autista, anche esaminando il voto medio e i giudizi dei feedback assegnati tramite la piattaforma dai precedenti passeggeri all'autista stesso, e si prenota. Le informazioni sul passeggero vengono inviate per email dalla piattaforma all'autista scelto, il quale può consultare sul portale il voto medio e i giudizi dei feedback ricevuti dal passeggero da parte di precedenti autisti e decidere se accettarlo o meno. Il passeggero di conseguenza riceverà una email di accettazione o di rifiuto della prenotazione effettuata, contenente, in caso di accettazione, un promemoria con città di partenza e destinazione, data e orario del viaggio, dati dell'autista e della sua automobile.

A viaggio effettuato, il passeggero può inserire un feedback sull'autista, espresso sia in forma di voto numerico che di giudizio discorsivo. A sua volta, l'autista può inserire un feedback sul passeggero, espresso sia in forma di voto numerico che di giudizio discorsivo. Sia i voti medi che i singoli giudizi dei feedback ricevuti da ciascun autista sono disponibili ai passeggeri; analogamente, sia i voti medi che i singoli giudizi dei feedback ricevuti da ciascun passeggero sono disponibili agli autisti.

Il candidato, fatte le opportune ipotesi aggiuntive, sviluppi:

1. un'analisi della realtà di riferimento, giungendo alla definizione di uno schema concettuale della base di dati che, a suo motivato giudizio, sia idoneo a gestire la realtà presentata;
2. il relativo schema logico;
3. le seguenti interrogazioni espresse in linguaggio SQL:
 - a) data una città di partenza, una di arrivo e una data, elencare gli autisti che propongono un viaggio corrispondente con prenotazioni non ancora chiuse, in ordine crescente di orario, riportando i dati dell'auto e il contributo economico richiesto;
 - b) dato il codice di una prenotazione accettata, estrarre i dati necessari per predisporre l'email di promemoria da inviare all'utente passeggero;
 - c) dato un certo viaggio, consentire all'autista di valutare le caratteristiche dei passeggeri visualizzando l'elenco di coloro che lo hanno prenotato, con il voto medio dei feedback ricevuti da

ciascun passeggero, presentando solo i passeggeri che hanno voto medio superiore ad un valore indicato dall'autista;

4. il progetto di massima della struttura funzionale dell'applicazione Web, realizzando, con appropriati linguaggi a scelta sia lato client che lato server, un segmento significativo dell'applicazione che consente l'interazione con la base di dati.

SECONDA PARTE

I. In relazione al tema proposto nella prima parte, il candidato integri il modello già realizzato al fine di gestire in automatico il numero di posti disponibili nei vari viaggi, evitando che sia responsabilità dell'autista dichiarare chiuse le prenotazioni sul portale. Nel momento in cui inserisce un viaggio, l'autista dichiara il numero massimo di posti disponibili. Mano a mano che gli autisti accettano le prenotazioni, il sistema visualizzerà solo i viaggi con posti ancora disponibili: a tal fine, una prenotazione non ancora accettata dall'autista non comporta alcun impegno del posto, che resta così ancora disponibile per prenotazioni di altri passeggeri. Per ciascun viaggio, la piattaforma mostrerà il numero dei posti disponibili e il numero delle prenotazioni non ancora accettate. Il candidato sviluppi poi la pagina web, sia lato client che lato server, per fornire ai passeggeri tali informazioni.

II. In relazione al tema proposto nella prima parte, il candidato immagini di volere documentare al committente l'operatività della piattaforma proposta. A tal fine, imposti una relazione tecnica che presenti le principali caratteristiche dell'applicazione Web in termini di organizzazione e funzionalità. In particolare, imposti la struttura di tale relazione, motivando le scelte e scrivendo un esempio significativo dei relativi contenuti.

III. Dato il seguente schema relazionale: film (id, titolo, durata, anno di produzione, genere); attore (id, nome, cognome, data_nascita, fotografia); recita (id_film, id_attore, ruolo); il candidato:

- determini la modalità di gestione del campo 'fotografia' che prevede la memorizzazione di una immagine dell'attore in un formato grafico (es. JPG);
- formalizzi in linguaggio SQL lo schema fisico corrispondente allo schema relazionale, sapendo che:
 - a. il campo 'genere' ammette solo i seguenti valori: fantasy, giallo, commedia, horror, drammatico, fantascienza, azione;
 - b. per la relazione 'recita', i campi 'id_film' e 'id_attore' referenziano rispettivamente la chiave primaria delle relazioni 'film' e 'attore';
- discuta l'uso degli indici nel modello fisico di una base di dati e suggerisca con motivato giudizio indici appropriati per questo schema relazionale, definendoli in linguaggio SQL.

IV. Un'azienda desidera sviluppare un'applicazione Web per la prenotazione on-line di eventi culturali, fruibile sia da computer desktop che da dispositivi mobili come tablet e smartphone. Il candidato esponga i punti critici da affrontare relativamente alle differenti proprietà di visualizzazione delle varie tipologie di dispositivi e alla rispettiva fruizione dei contenuti. Illustri possibili misure risolutive, con esempi relativi all'applicazione in questione.

Commento al tema proposto

La traccia fa riferimento ad un'applicazione di car pooling (condivisione di passaggi in auto) e affronta una problematica molto attuale e che già conta applicazioni importanti e diffuse (blablacar, uber, ecc.).

I punti da sviluppare nella prima parte sono laboriosi ma non sono necessarie particolari intuizioni o conoscenze e competenze che non siano coerenti con i programmi di studio del quinto anno.

I quattro quesiti della seconda parte (dei quali due da svolgere a scelta del candidato) richiedono molto tempo per essere svolti e un certo sforzo di impostazione e organizzazione del lavoro.

Soluzione prima parte

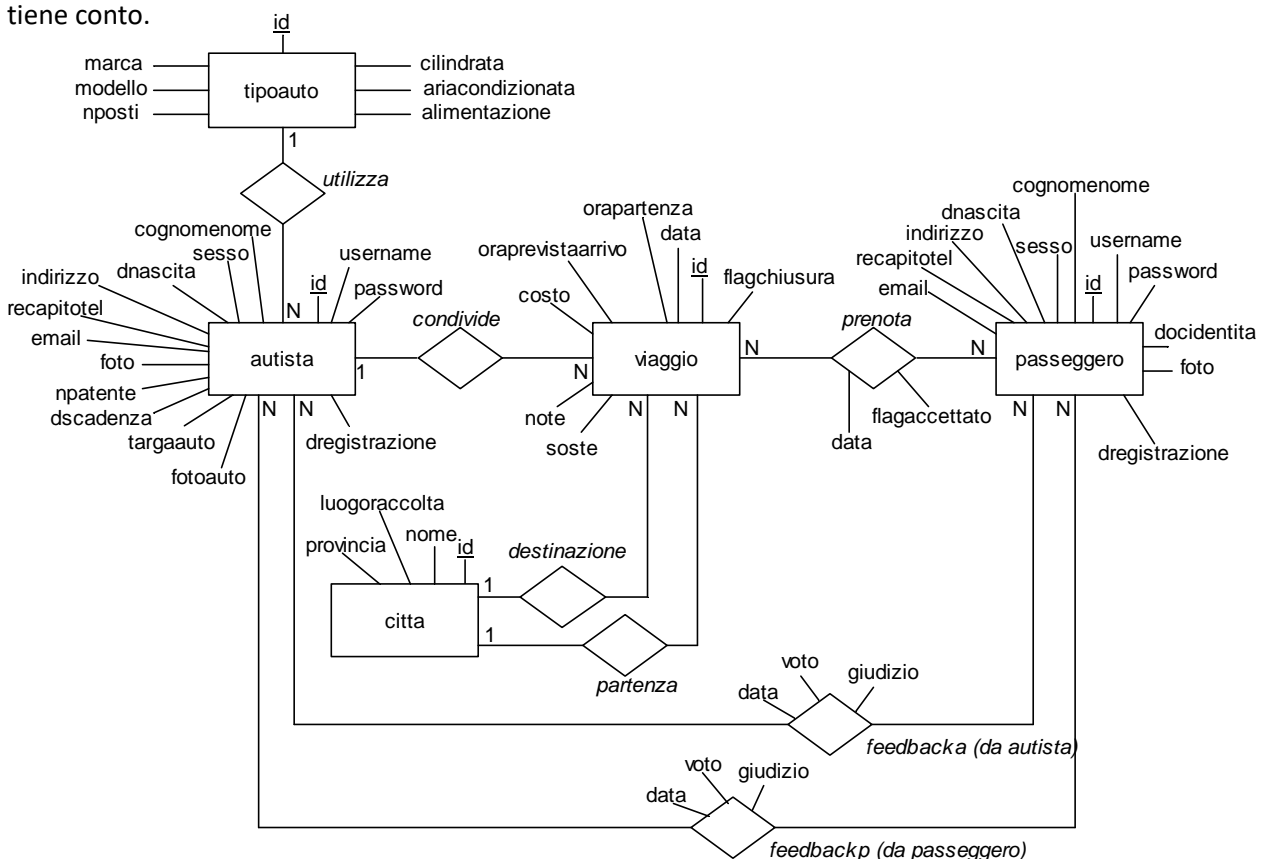
1 un'analisi della realtà di riferimento giungendo alla definizione di uno schema concettuale della base di dati che, a suo motivato giudizio, sia idoneo a gestire la realtà presentata

Si tratta di realizzare un portale che deve offrire un insieme di servizi che consentano la condivisione di autovetture private tra gruppi di viaggiatori. Attraverso il portale si realizza una web community, una sorta di social network, di viaggiatori che interagiscono tra di loro, condividono esperienze, si scambiano voti e giudizi. La mole di dati che bisogna gestire è enorme e i servizi vengono erogati attraverso una web application incentrata su una base di dati potente, veloce e scalabile. Si utilizzano i seguenti strumenti software: MySQL come Database, Apache come Web Server, Php come linguaggio lato Server, HTML, CSS, Java Script per implementare le pagine Web.

Poiché è ragionevole pensare che il servizio di car pooling venga utilizzato prevalentemente da giovani o comunque da persone che fanno largo uso di dispositivi mobili, si prevede di accedere alla piattaforma anche attraverso un'apposita app per tablet e smartphone.

Diagramma E/R

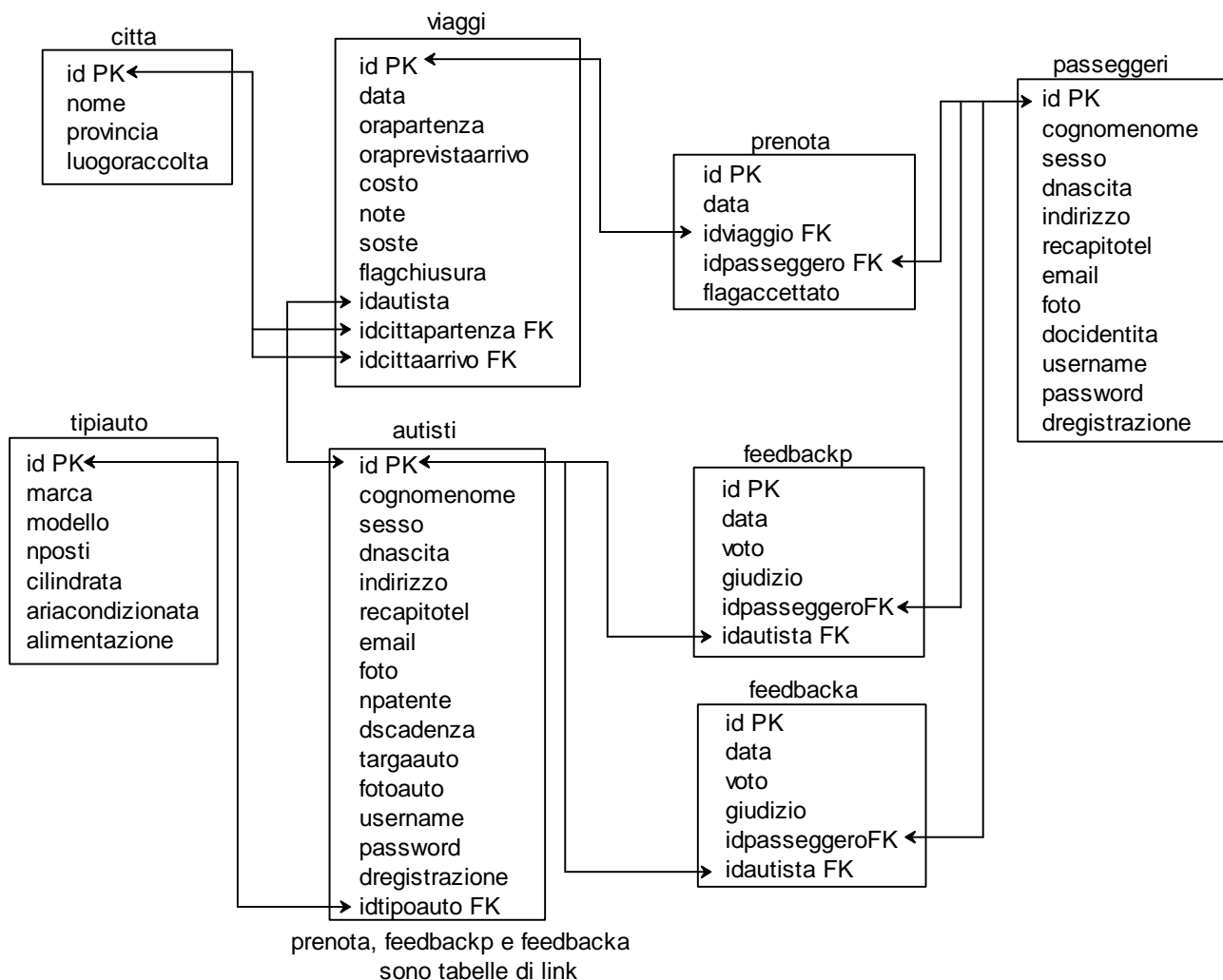
Facendo un'analisi accurata emergono diverse problematiche di cui non si fa cenno nella traccia (il percorso, le tappe intermedie, il calcolo delle distanze, la copertura assicurativa, ...) e di cui nella soluzione non si tiene conto.



Letture del diagramma E/R

- Ogni autista può condividere N viaggi, ogni viaggio è condiviso da 1 autista
- 1. Ogni autista usa 1 tipo di automobile, ogni tipo di automobile può essere usato da N autisti (Si ipotizza che ogni autista utilizzi un'unica automobile)
- Ogni viaggio ha 1 città di partenza, da ogni città possono partire N viaggi
- Ogni viaggio ha 1 città di destinazione, a ogni città possono concludersi N viaggi
- Ogni passeggero può prenotare N viaggi, ogni viaggio può essere prenotato da N passeggeri
- Ogni autista può esprimere N feedback su ogni passeggero, ogni passeggero può ricevere N feedback da ogni autista
- Ogni passeggero può esprimere N feedback su ogni autista, ogni autista può ricevere N feedback da ogni passeggero

2 il relativo schema logico



Definizione della struttura delle tabelle

Si fa riferimento a MySQL. Nome del database: carpooling

tabella	campi	key	tipo	lung.	Descrizione/note
tipiauto	id	PK	Int	4	Contatore auto_increment
	marca		VarChar	20	Si potrebbe utilizzare una tabella di marche
	modello		VarChar	20	
	nposti		Int	2	
	cilindrata		Int	5	
	ariacondizionata		Char	1	Ammette i valori S e N
	alimentazione		VarChar	20	Ammette i valori: Benzina, Gasolio, Metano, Elettrica
citta	id	PK	Int	5	Contatore auto_increment
	nome		VarChar	30	Es. Teramo
	provincia		Char	2	Es. TE (si utilizza ovviamente una tabella di province)
	Luogo raccolta		Text		Es. Piazza Dante
autisti	id	PK	Int	8	Contatore auto_increment
	cognomenome		VarChar	30	
	sexso		Char	1	Ammette i valori F e M (tipo Enum)
	dnascita		Date		
	indirizzo		VarChar	40	
	recapitel		VarChar	40	Recapiti telefonici
	email		VarChar	30	
	foto		Text		Link all'immagine
	npatente		VarChar	20	
	dscadenza		Date		Data scadenza patente
	targaauto		VarChar	20	
	fotoauto		Text		Link all'immagine
	dregistrazione		Date		Data di registrazione al portale
	username		VarChar	15	
	password		VarChar	32	Si utilizza la codifica MD5
idtipoauto	FK	Int	4	Fa riferimento a tipiauto.id	
passeggeri	id	PK	Int	8	Contatore auto_increment
	cognomenome		VarChar	30	
	sexso		Char	1	Ammette i valori F e M (tipo Enum)
	dnascita		Date		
	indirizzo		VarChar	40	
	recapitel		VarChar	40	Recapiti telefonici
	email		VarChar	30	
	foto		Text		Link all'immagine
	docidentita		VarChar	20	Documento di identità
	dregistrazione		Date		Data di registrazione al portale
	username		VarChar	15	
	password		VarChar	32	Si utilizza la codifica MD5
viaggi	id	PK	Int	8	Contatore auto_increment
	data		Date		
	orapartenza		Time		
	oraprevistaarrivo		Time		
	costo		Decimal	8,2	Contributo economico per passeggero
	note		Text		Possibilità di portare animali ecc.

	soste		Text		Luoghi e tempi di eventuali soste
viaggi	Flagchiusura		Char	1	Ammette i valori S e N Vale S quando l'autista chiude le prenotazioni
	idautista	FK	Int	8	Fa riferimento a autisti.id
	idcittapartenza	FK	Int	5	Fa riferimento a citta.id
	Idcittaarrivo	FK	Int	5	Fa riferimento a citta.id
prenota (tabella di link)	id	PK	Int	8	Contatore auto_increment
	data		Date		
	idviaggio		Int	8	Fa riferimento a viaggi.id
	idpasseggero		Int	8	Fa riferimento a passeggeri.id
	flagaccettato		Char	1	Ammette i valori S e N Vale S se l'autista accetta la prenotazione
Feedbackp (tabella di link)	id	PK	Int	8	Contatore auto_increment
	data		Date		
	voto		Int	1	Ammette i valori: 1,2,3,4 e 5
	giudizio		Text		Giudizio sul passeggero
	idpasseggero	FK	Int	8	Fa riferimento a passeggeri.id
	idautista	FK	Int	8	Fa riferimento a autisti.id
Feedbacka (tabella di link)	id	PK	Int	8	Contatore auto_increment
	data		Date		
	voto		Int	1	Ammette i valori: 1,2,3,4 e 5
	giudizio		Text		Giudizio sull'autista
	idpasseggero	FK	Int	8	Fa riferimento a passeggeri.id
	idautista	FK	Int	8	Fa riferimento a autisti.id

3. le seguenti interrogazioni espresse in linguaggio SQL:

a) data una città di partenza, una di arrivo e una data, elencare gli autisti che propongono un viaggio corrispondente con prenotazioni non ancora chiuse, in ordine crescente di orario, riportando i dati dell'auto e il contributo economico richiesto;

```
Select autisti.cognomenome,autisti.targaauto,tipiauto.*,viaggi.costo
from autisti,tipiauto,viaggi
where autisti.idtipoauto=tipiauto.id
and viaggi.idautista=autisti.id
and viaggi.idcittapartenza=(select id from citta where nome=cittapartenzaX)
and viaggi.idcittaarrivo=(select id from citta where nome=cittaarrivoX)
and viaggi.data=datax
and viaggi.flagchiusura="N"
order by viaggi.orapartenza
```

b) dato il codice di una prenotazione accettata, estrarre i dati necessari per predisporre l'email di promemoria da inviare all'utente passeggero;

```
Select passeggeri.*,prenota.*,viaggi.*
from passeggeri,prenota,viaggi
where passeggeri.id=prenota.idpasseggero
and prenota.idviaggio=viaggi.id
and prenota.id=codiceX
```

Il nome della città di partenza e quello della città di destinazione si possono estrarre successivamente con le seguenti due query:

```
Select nome from citta where id=cittapartenzaX
Select nome from citta where id=cittaarrivoX
```

dove cittapartenzaX e cittaarrivoX sono rispettivamente i valori di viaggi.idcittapartenza e viaggi.idcittaarrivo estratti con la prima query

c) dato un certo viaggio, consentire all'autista di valutare le caratteristiche dei passeggeri visualizzando l'elenco di coloro che lo hanno prenotato, con il voto medio dei feedback ricevuti da ciascun passeggero, presentando solo i passeggeri che hanno voto medio superiore ad un valore indicato dall'autista;

```
Select passeggeri.cognomenome,avg(feedbackp.voto)
from passeggeri, prenota,feedbackp
where passeggeri.id =prenota.idpasseggero
and passeggeri.id=feedback.idpasseggero
and prenota.idviaggio=viaggioX
group by passeggeri.cognomenome
having avg(feedbackp.voto)>votoX
```

4. il progetto di massima della struttura funzionale dell'applicazione Web, realizzando, con appropriati linguaggi a scelta sia lato client che lato server, un segmento significativo dell'applicazione che consente l'interazione con la base di dati.

Logo Login Registrati

data menù

Slider immagini o comunicazioni/news

Città partenza Teramo

Città destinazione Roma

Data 23/06/2017

Cerca un viaggio Offri un viaggio

Inserisci un feedback

Passeggero Pinco Pallino

☆☆☆☆☆ Vota

info e contatti social networks

Si sceglie di codificare la fase di immissione tipiauto. Per semplicità il codice è molto essenziale e non viene utilizzato alcun css.

Lato client: html e javascript addtipi.html

```
<html>
<head> <title>Inserimento tipo auto</title></head>
<script language="JavaScript">
function valida()
{
ma=document.form1.marca.value;
mo=document.form1.modelo.value;
np= document.form1.nposti.value;
ci= document.form1.cilindrata.value;
if(ma==""){alert("Inserire la marca");return false;}
if(mo==""){alert("Inserire il modello");return false;}
if(np==""){alert("Inserire il numero di posti");return false;}
if(ci==""){alert("Inserire la cilindrata");return false;}
return true;
}
</script>
```

Inserimento Tipo Auto

Marca

Fiat

Modello

Panda

Nposti

5

Cilindrata

1200

Aria condizionata

Si No

Alimentazione

Benzina Gasolio Metano Elettrica

Azzera i campi

Inserisci

```
<body>
  <h1>Inserimento Tipo Auto </h1>
  <form name="form1" action="addtipi.php" method="POST" onsubmit="return valida()" >
    Marca<br/><input type="text" name="marca" size=20> <br/>
    Modello<br/><input type="text" name="modello" size=20> <br/>
    Nposti<br/><input type="text" name="nposti" size=5><br/>
    Cilindrata<br/><input type="text" name="cilindrata" size=5><br/>
    Aria condizionata<br/><input type="radio" name="ariacond" value="S" checked> Si
    <input type="radio" name="ariacond" value="N" > No<br/>
    Alimentazione<br/><input type="radio" name="alim" value="Benzina" checked>
    Benzina<input type="radio" name="alim" value="Gasolio"> Gasolio
    <input type="radio" name="alim" value="Metano"> Metano
    <input type="radio" name="alim" value="Elettrica"> Elettrica<br/>
    <br><input type="reset" value="Azzera i campi">
    <input type="submit" value="Inserisci">
  </form>
</body>
</html>
```

Lato server: PHP addtipi.php

```
<?php
$DB_Host='myserver';
$DB_User='myuser';
$DB_Password='mypassword';
$DB_Name='carpooling';
//Best practices vs mysql injection
$conn=mysqli_connect($DB_Host,$DB_User,$DB_Password,$DB_Name);
$marca=mysqli_real_escape_string($conn,$_POST["marca"]);
$modello=mysqli_real_escape_string($conn,$_POST["modello"]);
$nposti=(int)$_POST["nposti"];
$cilindrata=(int)$_POST["cilindrata"];
$ariacond=$_POST["ariacond"];
$alim=$_POST["alim"];
$s="Insert into tipiauto (marca,modello,nposti,cilindrata,ariacondizionata,alimentazione)
  Values('$marca','$modello','$nposti','$cilindrata','$ariacond','$alim')";
$q=mysqli_query($conn,$s);
mysqli_close($conn);
header("Location:addtipi.html");
?>
```

Teramo 23 giugno 2017

Prof. Mauro De Berardis ITT "Alessandrini" Teramo

Soluzione seconda parte

Si sceglie di rispondere ai quesiti III e IV

III. Dato il seguente schema relazionale: *film* (*id*, *titolo*, *durata*, *anno di produzione*, *genere*); *attore* (*id*, *nome*, *cognome*, *data_nascita*, *fotografia*); *recita* (*id_film*, *id_attore*, *ruolo*); il candidato:

• determini la modalità di gestione del campo 'fotografia' che prevede la memorizzazione di una immagine dell'attore in un formato grafico (es. JPG);

• formalizzi in linguaggio SQL lo schema fisico corrispondente allo schema relazionale, sapendo che:

a. il campo 'genere' ammette solo i seguenti valori: *fantasy*, *giallo*, *commedia*, *horror*, *drammatico*, *fantascienza*, *azione*;

b. per la relazione 'recita', i campi 'id_film' e 'id_attore' referenziano rispettivamente la chiave primaria delle relazioni 'film' e 'attore'

Anche se è possibile memorizzare le immagini all'interno di un database e gestirle utilizzando SQL, generalmente si ritiene che sia meglio archiviare le immagini in semplici cartelle della root del Web server e allocare nel database i link alle immagini stesse. Nel nostro caso il campo "fotografia" è un campo di tipo Text che contiene il percorso e il nome del file (ad esempio: "/images/attori/marlonbrando.jpg"). Questa scelta è la stessa utilizzata nei principali CMS (Content Management System) per la gestione dei siti web (WordPress, Joomla!, Drupal ecc.)

```
create table if not exists film(  
id int(8) primary key auto_increment,  
titolo text,  
durata float  
annoproduzione int(4),  
genere Enum(' giallo', ' commedia', 'horror', 'drammatico', 'fantascienza', ' azione ' ));
```

```
create table if not exists attori(  
id int(8) primary key auto_increment,  
nome varchar(20),  
cognome varchar(20),  
data_nascita date,  
fotografia text);
```

recita è un'entità debole, meglio comunque inserire una chiave primaria *id*

```
create table if not exists recita(  
id int(8) primary key auto_increment,  
id_film int(8),  
id_attore int(8),  
ruolo varchar(30),  
foreign key(id_film) references film(id),  
foreign key(id_attore) references attori(id) );
```

Un indice è una struttura di dati ausiliaria che permette di velocizzare un'interrogazione Sql per estrarre i dati da una o più tabelle.

Ad esempio se si vuole elencare i film prodotti nel 1985 si utilizza la seguente query: *Select * from film where annoproduzione=1985*. La query legge tutte le righe della tabella (a volte possono essere milioni) e stampa le righe in cui il campo annoproduzione vale 1985. Creando un indice sul campo annoproduzione attraverso la query: *CREATE INDEX anno_ind ON film (`annoproduzione`)* la ricerca precedente diventa molto più veloce

IV. Un'azienda desidera sviluppare un'applicazione Web per la prenotazione on-line di eventi culturali, fruibile sia da computer desktop che da dispositivi mobili come tablet e smartphone. Il candidato esponga i punti critici da affrontare relativamente alle differenti proprietà di visualizzazione delle varie tipologie di dispositivi e alla rispettiva fruizione dei contenuti. Illustri possibili misure risolutive, con esempi relativi all'applicazione in questione.

Una qualsiasi applicazione web deve essere responsive ovvero deve adattarsi automaticamente ai dispositivi dai quali viene utilizzata. Deve essere visualizzata correttamente sia su computer desktop che su dispositivi mobili come tablet e smartphone evitando all'utente la necessità di scorrere e ridimensionare le pagine visitate per poter accedere ai contenuti.

Per realizzare una pagina web responsive, occorre modificare dinamicamente il layout, ovvero modificare il layout a fronte di risoluzioni video differenti. Pertanto occorre fissare e definire i cosiddetti breakpoint facendo riferimento principalmente alle seguenti dimensioni espresse in px: 320,480,768,1024. Un breakpoint è un valore in px in cui si verifica una modifica del layout della pagina: tale modifica si può fare attraverso le media queries che sono moduli css3 che attraverso il supporto @media permettono di utilizzare diversi fogli di stile a seconda del dispositivo.

Esempio di media query:

```
@media only screen and (max-device-width: 767px) {  
  .container {  
    width: 420px;  
    margin-top:5px;  
    ...  
  }  
}
```

In tal caso solo se il device è screen e la dimensione massima è di 767px, alla classe container si applica lo stile width:420px e margin-top: 5px

Teramo 26 giugno 2017

Prof. Mauro De Berardis ITT "Alessandrini" Teramo