

Sistemi distribuiti (nuovo ordinamento) Basi di dati, secondo modulo (vecchio ordinamento) 17 aprile 2002

Compito A

Tempo a disposizione: due ore e trenta minuti. Libri chiusi.

Domanda 1 (25%)

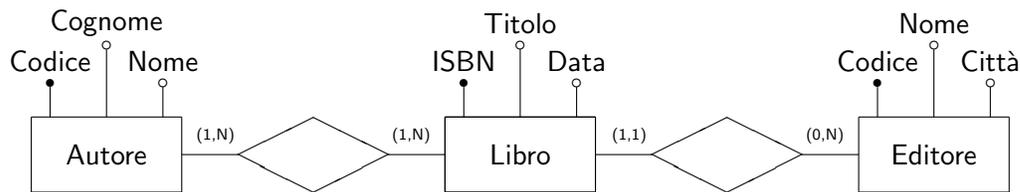
In al massimo due pagine (circa 300-500 parole), senza utilizzare schemi, illustrare i principali difetti dell'attuale sito del corso di studio e gli interventi che si intende proporre per superarli.

La risposta a questa domanda deve essere una sintesi (“executive summary”) del progetto; può includere riferimenti al progetto stesso, ma deve essere in prima battuta comprensibile senza consultare il progetto. Essa deve essere tecnicamente corretta, ma deve essere scritta in primo luogo per un “committente” non esperto.

Poiché sarà valutata anche la chiarezza espositiva, si consiglia fortemente di procedere preliminarmente attraverso una brutta copia.

Domanda 2 (15%)

Considerare i dati rappresentati dal seguente schema E-R



Indicare (con una breve motivazione) in quale (o quali) dei seguenti contesti

1. in una base di dati orientata agli oggetti
2. in una base di dati relazionale a oggetti
3. in un sito Web

può avere senso prevedere una struttura complessa (classe nel caso 1, relazione nidificata nel caso 2, schema di pagina nel caso 3) **Libro**, ogni istanza della quale contenga informazioni aggregate, e precisamente i dati specifici di un libro, la lista degli autori (con tutti i dati) e l'editore (con tutti i dati).

Domanda 3 (20%)

Con riferimento ad una base di dati relazionale contenente le relazioni Autore, Autore-Libro, Libro, Editore, corrispondente allo schema E-R mostrato nella domanda precedente, scrivere un metodo Java che, accedendo alla base di dati con JDBC, produca un elenco degli autori (cognome e nome) con, per ciascuno, i titoli dei libri che ha scritto.

Domanda 4 (40%)

Scrivere il codice per

- una pagina JSP `inputStudente.jsp`
- una classe servlet `ControllaInput`

che interagiscono come segue.

La pagina `inputStudente.jsp` contiene una form HTML per l'immissione di quattro parametri che rappresentano i seguenti dati relativi ad uno studente iscritto all'Università:

- nome (una stringa di caratteri)
- cognome (una stringa di caratteri)
- data di nascita (una stringa di caratteri)
- matricola (una stringa di caratteri)

Il processamento dei dati è affidato alla servlet `ControllaInput`, che deve fare le seguenti verifiche:

1. tutti e quattro i parametri devono essere presenti
2. il parametro relativo alla data deve rappresentare una data valida (vedi sotto il metodo `isCorrect`)
3. il parametro relativo alla matricola deve essere consistente con gli altri dati (vedi sotto il metodo `isValid`).

Se tutte e tre le verifiche danno esito positivo, la servlet `ControllaInput` deve reindirizzare il controllo ad una ulteriore servlet, `ProcessaStudente`, che non va implementata e, per funzionare correttamente, ha bisogno della matricola.

Se una delle verifiche dà esito negativo, la servlet `ControllaInput` deve ritornare il controllo alla pagina iniziale `inputStudente.jsp`, permettendogli di visualizzare un opportuno messaggio di errore.

Per la verifica 3 (quella sulla matricola) è richiesto un controllo più sofisticato: quando, nell'ambito della stessa sessione di lavoro, la verifica 3 fallisce per più di tre volte, la servlet `ControllaInput` deve reindirizzare la risposta ad una pagina HTML, che si chiama `utenteDistratto.html`.

Si supponga di avere a disposizione una classe Java che si chiama `MyUtility`, con i seguenti metodi:

```
public static boolean isValid (String data)
```

che restituisce `true` se il parametro rappresenta una data valida, `false` altrimenti.

```
public static boolean isCorrect
```

```
(String matricola, String nome, String cognome, String dataNascita)
```

che restituisce `true` se esiste uno studente a cui corrispondono i dati forniti come parametri, `false` altrimenti

Sistemi distribuiti (nuovo ordinamento) Basi di dati, secondo modulo (vecchio ordinamento) 17 aprile 2002

Compito B

Tempo a disposizione: due ore e trenta minuti. Libri chiusi.

Domanda 1 (25%)

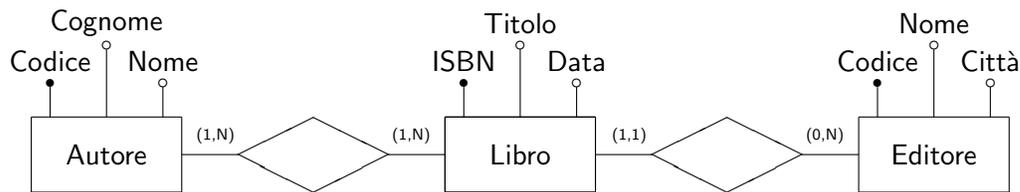
In al massimo due pagine (circa 300-500 parole), senza utilizzare schemi, illustrare i principali difetti dell'attuale sito del corso di studio e gli interventi che si intende proporre per superarli.

La risposta a questa domanda deve essere una sintesi (“executive summary”) del progetto; può includere riferimenti al progetto stesso, ma deve essere in prima battuta comprensibile senza consultare il progetto. Essa deve essere tecnicamente corretta, ma deve essere scritta in primo luogo per un “committente” non esperto.

Poiché sarà valutata anche la chiarezza espositiva, si consiglia fortemente di procedere preliminarmente attraverso una brutta copia.

Domanda 2 (15%)

Considerare i dati rappresentati dal seguente schema E-R



Indicare (con una breve motivazione) in quale (o quali) dei seguenti contesti

1. in una base di dati orientata agli oggetti
2. in una base di dati relazionale a oggetti
3. in un sito Web

può avere senso prevedere una struttura complessa (classe nel caso 1, relazione nidificata nel caso 2, schema di pagina nel caso 3) Autore, ogni istanza della quale contenga informazioni aggregate, e precisamente i dati personali di un autore e la lista dei libri che ha scritto, con tutti i dati, incluso il relativo editore (con tutti i dati).

Domanda 3 (20%)

Con riferimento ad una base di dati relazionale contenente le relazioni Autore, Autore-Libro, Libro, Editore, corrispondente allo schema E-R mostrato nella domanda precedente, scrivere un metodo Java che, accedendo alla base di dati con JDBC, produca un elenco dei titoli dei libri, ciascuno con i relativi autori.

Domanda 4 (40%)

Scrivere il codice per

- una pagina JSP `inputDatiStud.jsp`
- una classe servlet `EsaminaInput`

che interagiscono come segue.

La pagina `inputDatiStud.jsp` contiene una form HTML per l'immissione di quattro parametri che rappresentano i seguenti dati relativi ad uno studente iscritto all'Università:

- nome (una stringa di caratteri)
- cognome (una stringa di caratteri)
- data di nascita (una stringa di caratteri)
- matricola (una stringa di caratteri)

Il processamento dei dati è affidato alla servlet `EsaminaInput`, che deve fare le seguenti verifiche:

1. tutti e quattro i parametri devono essere presenti
2. il parametro relativo alla data deve rappresentare una data valida (vedi sotto il metodo `isCorrect`)
3. il parametro relativo alla matricola deve essere consistente con gli altri dati (vedi sotto il metodo `isValid`).

Se tutte e tre le verifiche danno esito positivo, la servlet `EsaminaInput` deve reindirizzare il controllo ad una ulteriore servlet, `ProcessaStudente`, che non va implementata e, per funzionare correttamente, ha bisogno della matricola.

Se una delle verifiche dà esito negativo, la servlet `EsaminaInput` deve ritornare il controllo alla pagina iniziale `inputDatiStud.jsp`, permettendogli di visualizzare un opportuno messaggio di errore.

Per la verifica 3 (quella sulla matricola) è richiesto un controllo più sofisticato: quando, nell'ambito della stessa sessione di lavoro, la verifica 3 fallisce per più di tre volte, la servlet `EsaminaInput` deve reindirizzare la risposta ad una pagina HTML, che si chiama `utenteDistratto.html`.

Si supponga di avere a disposizione una classe Java che si chiama `Servizio`, con i seguenti metodi:

```
public static boolean isValid (String data)
    che restituisce true se il parametro rappresenta una data valida, false altrimenti.
public static boolean isCorrect
    (String matricola, String nome, String cognome, String dataNascita)
    che restituisce true se esiste uno studente a cui corrispondono i dati forniti come
    parametri, false altrimenti
```

Sistemi distribuiti (nuovo ordinamento) Basi di dati, secondo modulo (vecchio ordinamento) 17 aprile 2002

Compito C

Tempo a disposizione: due ore e trenta minuti. Libri chiusi.

Domanda 1 (25%)

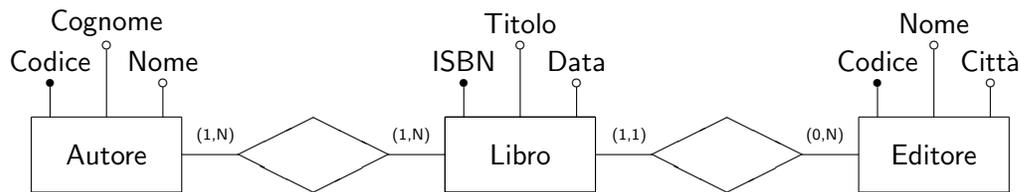
In al massimo due pagine (circa 300-500 parole), senza utilizzare schemi, illustrare i principali difetti dell'attuale sito del corso di studio e gli interventi che si intende proporre per superarli.

La risposta a questa domanda deve essere una sintesi (“executive summary”) del progetto; può includere riferimenti al progetto stesso, ma deve essere in prima battuta comprensibile senza consultare il progetto. Essa deve essere tecnicamente corretta, ma deve essere scritta in primo luogo per un “committente” non esperto.

Poiché sarà valutata anche la chiarezza espositiva, si consiglia fortemente di procedere preliminarmente attraverso una brutta copia.

Domanda 2 (15%)

Considerare i dati rappresentati dal seguente schema E-R



Indicare (con una breve motivazione) in quale (o quali) dei seguenti contesti

1. in una base di dati orientata agli oggetti
2. in una base di dati relazionale a oggetti
3. in un sito Web

può avere senso prevedere una struttura complessa (classe nel caso 1, relazione nidificata nel caso 2, schema di pagina nel caso 3) **Libro**, ogni istanza della quale contenga informazioni aggregate, e precisamente i dati specifici di un libro, la lista degli autori (con tutti i dati) e l'editore (con tutti i dati).

Domanda 3 (20%)

Con riferimento ad una base di dati relazionale contenente le relazioni Autore, Autore-Libro, Libro, Editore, corrispondente allo schema E-R mostrato nella domanda precedente, scrivere un metodo Java che, accedendo alla base di dati con JDBC, produca un elenco degli autori (cognome e nome) con, per ciascuno, i titoli dei libri che ha scritto.

Domanda 4 (40%)

Scrivere il codice per

- una pagina JSP `inputDatiStud.jsp`
- una classe servlet `ControllaInput`

che interagiscono come segue.

La pagina `inputDatiStud.jsp` contiene una form HTML per l'immissione di quattro parametri che rappresentano i seguenti dati relativi ad uno studente iscritto all'Università:

- nome (una stringa di caratteri)
- cognome (una stringa di caratteri)
- data di nascita (una stringa di caratteri)
- matricola (una stringa di caratteri)

Il processamento dei dati è affidato alla servlet `ControllaInput`, che deve fare le seguenti verifiche:

1. tutti e quattro i parametri devono essere presenti
2. il parametro relativo alla data deve rappresentare una data valida (vedi sotto il metodo `isCorrect`)
3. il parametro relativo alla matricola deve essere consistente con gli altri dati (vedi sotto il metodo `isValid`).

Se tutte e tre le verifiche danno esito positivo, la servlet `ControllaInput` deve reindirizzare il controllo ad una ulteriore servlet, `ProcessaStudente`, che non va implementata e, per funzionare correttamente, ha bisogno della matricola.

Se una delle verifiche dà esito negativo, la servlet `ControllaInput` deve ritornare il controllo alla pagina iniziale `inputDatiStud.jsp`, permettendogli di visualizzare un opportuno messaggio di errore.

Per la verifica 3 (quella sulla matricola) è richiesto un controllo più sofisticato: quando, nell'ambito della stessa sessione di lavoro, la verifica 3 fallisce per più di tre volte, la servlet `ControllaInput` deve reindirizzare la risposta ad una pagina HTML, che si chiama `utenteDistratto.html`.

Si supponga di avere a disposizione una classe Java che si chiama `Servizio`, con i seguenti metodi:

```
public static boolean isValid (String data)
    che restituisce true se il parametro rappresenta una data valida, false altrimenti.
public static boolean isCorrect
    (String matricola, String nome, String cognome, String dataNascita)
    che restituisce true se esiste uno studente a cui corrispondono i dati forniti come
    parametri, false altrimenti
```

Sistemi distribuiti (nuovo ordinamento) Basi di dati, secondo modulo (vecchio ordinamento) 17 aprile 2002

Compito D

Tempo a disposizione: due ore e trenta minuti. Libri chiusi.

Domanda 1 (25%)

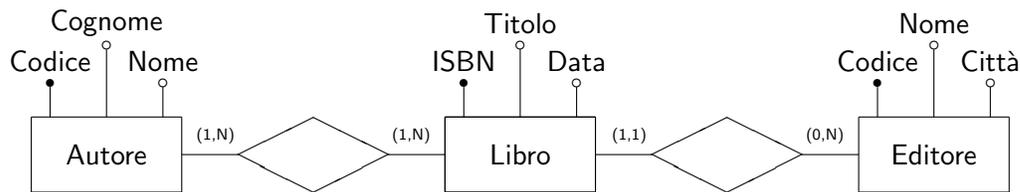
In al massimo due pagine (circa 300-500 parole), senza utilizzare schemi, illustrare i principali difetti dell'attuale sito del corso di studio e gli interventi che si intende proporre per superarli.

La risposta a questa domanda deve essere una sintesi (“executive summary”) del progetto; può includere riferimenti al progetto stesso, ma deve essere in prima battuta comprensibile senza consultare il progetto. Essa deve essere tecnicamente corretta, ma deve essere scritta in primo luogo per un “committente” non esperto.

Poiché sarà valutata anche la chiarezza espositiva, si consiglia fortemente di procedere preliminarmente attraverso una brutta copia.

Domanda 2 (15%)

Considerare i dati rappresentati dal seguente schema E-R



Indicare (con una breve motivazione) in quale (o quali) dei seguenti contesti

1. in una base di dati orientata agli oggetti
2. in una base di dati relazionale a oggetti
3. in un sito Web

può avere senso prevedere una struttura complessa (classe nel caso 1, relazione nidificata nel caso 2, schema di pagina nel caso 3) Autore, ogni istanza della quale contenga informazioni aggregate, e precisamente i dati personali di un autore e la lista dei libri che ha scritto, con tutti i dati, incluso il relativo editore (con tutti i dati).

Domanda 3 (20%)

Con riferimento ad una base di dati relazionale contenente le relazioni Autore, Autore-Libro, Libro, Editore, corrispondente allo schema E-R mostrato nella domanda precedente, scrivere un metodo Java che, accedendo alla base di dati con JDBC, produca un elenco dei titoli dei libri, ciascuno con i relativi autori.

Domanda 4 (40%)

Scrivere il codice per

- una pagina JSP `inputStudente.jsp`
- una classe servlet `EsaminaInput`

che interagiscono come segue.

La pagina `inputStudente.jsp` contiene una form HTML per l'immissione di quattro parametri che rappresentano i seguenti dati relativi ad uno studente iscritto all'Università:

- nome (una stringa di caratteri)
- cognome (una stringa di caratteri)
- data di nascita (una stringa di caratteri)
- matricola (una stringa di caratteri)

Il processamento dei dati è affidato alla servlet `EsaminaInput`, che deve fare le seguenti verifiche:

1. tutti e quattro i parametri devono essere presenti
2. il parametro relativo alla data deve rappresentare una data valida (vedi sotto il metodo `isCorrect`)
3. il parametro relativo alla matricola deve essere consistente con gli altri dati (vedi sotto il metodo `isValid`).

Se tutte e tre le verifiche danno esito positivo, la servlet `EsaminaInput` deve reindirizzare il controllo ad una ulteriore servlet, `ProcessaStudente`, che non va implementata e, per funzionare correttamente, ha bisogno della matricola.

Se una delle verifiche dà esito negativo, la servlet `EsaminaInput` deve ritornare il controllo alla pagina iniziale `inputStudente.jsp`, permettendogli di visualizzare un opportuno messaggio di errore.

Per la verifica 3 (quella sulla matricola) è richiesto un controllo più sofisticato: quando, nell'ambito della stessa sessione di lavoro, la verifica 3 fallisce per più di tre volte, la servlet `EsaminaInput` deve reindirizzare la risposta ad una pagina HTML, che si chiama `utenteDistratto.html`.

Si supponga di avere a disposizione una classe Java che si chiama `MyUtility`, con i seguenti metodi:

```
public static boolean isValid (String data)
```

che restituisce `true` se il parametro rappresenta una data valida, `false` altrimenti.

```
public static boolean isCorrect
```

```
(String matricola, String nome, String cognome, String dataNascita)
```

che restituisce `true` se esiste uno studente a cui corrispondono i dati forniti come parametri, `false` altrimenti

Sistemi distribuiti (nuovo ordinamento)
Basi di dati, secondo modulo (vecchio ordinamento)
17 aprile 2002

Istruzioni e informazioni importanti

1. Ciascuno studente riceve cinque fogli protocollo; non può riceverne altri e deve riconsegnarli tutti (insieme al testo); non può usare altri fogli.
2. Scrivere su ciascun foglio (in alto a destra sulla prima facciata, con calligrafia leggibile) cognome (sottolineato), nome, numero di matricola (se noto), sigla del compito (A, B, C, ...) e corso di studio di appartenenza (laurea ingegneria informatica n.o., laurea ingegneria informatica v.o., diploma ingegneria informatica, laurea ingegneria elettronica v.o., etc.)
3. Destinare due fogli alla “bella copia”, utilizzandone uno per le risposte alle prime tre domande e l’altro per la risposta alla quarta;
4. Si consiglia di rispondere alle domande nell’ordine, ricopiando subito nei fogli di bella copia.
5. È possibile consultare materiale di ogni tipo; è anche possibile scambiare materiale, ma solo previa autorizzazione.
6. Al termine rimanere seduti al proprio posto, senza scrivere dopo il segnale di conclusione (pena l’annullamento del compito) aspettando che il docente ritiri l’elaborato.
7. Durante la prova non è consentito uscire e rientrare. Nell’ultima mezz’ora non è consentito uscire.

**Sistemi distribuiti (nuovo ordinamento)
Basi di dati, secondo modulo (vecchio ordinamento)
17 aprile 2002**

**Istruzioni e informazioni importanti
per gli assistenti di aula**

1. può essere utile mettere il foglio istruzioni sul banco prima che gli studenti entrino, per disporli con più ordine
2. gli studenti possono consultare materiale di ogni tipo e scambiarselo, ma solo previa autorizzazione
3. fare l'appello all'entrata ma controllare i documenti dopo
4. far sedere gli studenti due posti sì e uno no a file alterne; se fossero di più, metterli due file sì e una no, mantenendo però l'incolonnamento
5. cinque fogli per uno subito e poi niente
6. i compiti debbono seguire lo schema A B A B nelle file dispari e C D C D nelle file pari
7. non far alzare nessuno per nessun motivo e non fornire spiegazioni
8. illustrare le istruzioni sul foglio distribuito a tutti gli studenti