# Test di fine settimana – Week 5

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome | Eleonora |
|  |  | Cognome | Lombardo |
|  |  | Data | 18-06-2021 |

Leggete attentamente ogni domanda e argomentare quanto più possibile **fornendo anche degli esempi**.  
ATTENZIONE: Le domande a risposta multipla possono contenere più risposte corrette.

1. *Spiegare come funziona una connection pool.*

*Una connection pool consiste in un insieme di connessioni “conservate” e pronte all’uso che l’utente può chiamare nel momento in cui ci si debba connettere al database, non c’è quindi bisogno di crearne una nuova fin quando ve ne sono disponibili nel pool. Quando termina la connessione utente-db la connessione ritorna nel pool.*

1. *Quale/i delle seguenti affermazioni è esatta riguardo la connection string?*

La connection string è un oggetto che gestisce la connessione a un database

La connection string è una stringa che fornisce i parametri di inizializzazione per creare una connessione

La connection string è composta da valori in formato chiave-valore

Se un utente accede al db da diverse postazioni, la connection string cambia

1. *Spiegare le differenze tra le modalità di connessione al database*

*VI sono due modalità di connessione al database: disconnected mode e connected mode.*

*La disconnected mode apre la connessione solo per la ritrieve dei dati per poi chiuderla immediatamente. Salva i dati ricavati in memoria. La connection mode invece tiene attiva la connessione fin quando non è “esplicitamente chiusa” (con le parentesi)*

1. *Spiegare cosa è un DataAdapter e specificare in quale modalità si usa*

*Il DataAdapter si usa nella modalità di connessione al db disconnected ed è semplicemente un insieme di comandi e connessioni utilizzati per fare la fill del DataSet con i dati del db in memoria e poi per aggiornare il server SQL.*

1. *Data un’entità Persona con Nome, Cognome, Eta scrivere una query Linq con query syntax e lambda expression che prelevi le persone maggiorenni ordinate per cognome e per nome*

*QUERY EXPRESSION*

*IEnumerable <Persona> person=*

*from p in person*

*where p.Eta>=18*

*orderby p.Cognome, p.Nome*

*select p;*

*IEnumerable<Persona> person =*

*Persona.Where(person=> person.Eta>=18).OrderBy(person=> person.Cognome).ThenBy (person=>person.Nome);*

1. *Spiegare la differenza tra concorrenza ottimistica e pessimistica nell’accesso ad un database.*

*La concorrenza pessimistica usa una specie di variabile di lock, cioè ogni qualvolta deve modificare un dato controlla se il dato è “bloccato” da qualche utente, se non lo è l’utente blocca l’accesso a quei dati fino al termine delle operazioni e quindi rilasciato.*

*La concorenza ottimistica invece non blocca alcun dato o set di dati, semplicemente controlla se l’ultima modifica risale a dopo che l’utente ha letto i dati. Nel caso un qualche utente abbia modificato i dati nell’intervallo tra quando l’utente ha letto i dati e quando ha deciso di caricarli sul db allora si ha una violazione della concorrenza ottimistica, in questo caso si deve decidere se sovrascrivere i dati o annullare l’operazione.*

*Esercitazione Pratica*

*Creare una Console App che gestisca la registrazione degli esami di uno Studente.*

*Lo studente è definito con:*

* *Nome*
* *Cognome*
* *AnnoNascita*

*L’esame è definito da:*

* *Nome*
* *CFU*
* *Data*
* *Votazione*
* *Passato*

*Vedere tutti gli studenti del database*

*Registrare un esame per uno studente specifico*

*Mostrare gli esami di uno studente ordinati per votazione e per data”*

*Aggiungere un nuovo studente utilizzando la modalità disconnessa di ADO.NET*

*Requisiti Tecnici:*

* *Usare la tecnologia ADO.NET*
* *Utilizzare opportunamente il Repository Pattern*
* *Realizzare le interrogazioni attraverso System.Linq in modalità Lambda Expression*

*Opzionale: Gestire l’interazione con l’utente con un menù.*

*Caricare la prova pratica e teorica su Github.*