Basi di Dati

JDBC - Esercitazione n. 4

"Quaderno" n. 4

La finalità di questa esercitazione è scrivere le parti mancanti di una semplice applicazione Java che utilizza JDBC per accedere ad una base di dati.

Attenzione. Il codice realizzato durante l'esercitazione, e eventualmente terminato a casa, deve essere archiviato in un file zip e caricato sul portale della didattica in quanto corrisponde al quarto "quaderno". La data entro la quale consegnare il materiale è riportata sul sito del corso.

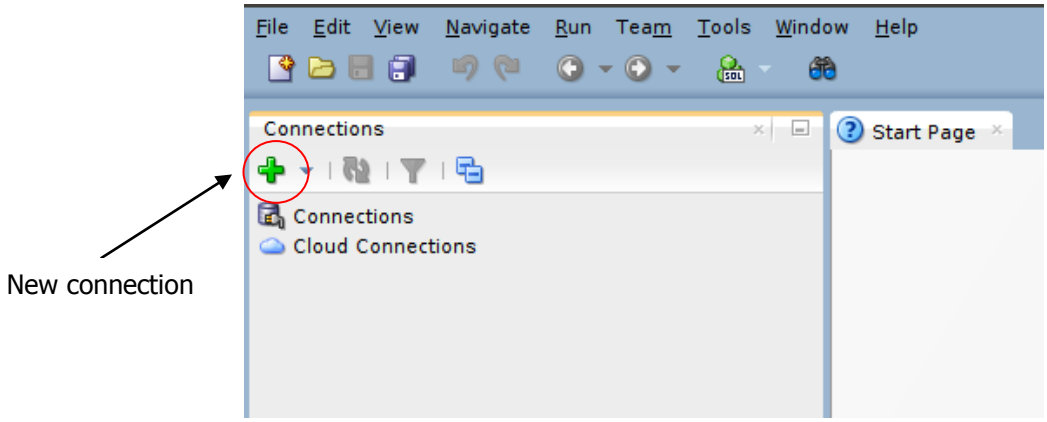
Passi preliminari per lo svolgimento delle esercitazioni

Connessione al server oracle

L'esecuzione degli script e dei comandi SQL sono eseguite tramite il software SQL Developer che permette di connettersi alla base di dati Oracle e eseguire script contenenti comandi SQL o singoli comandi SQL.

1) Connessione alla base di dati

- Aprire il programma Oracle SQL Developer
- Cliccare su New connection

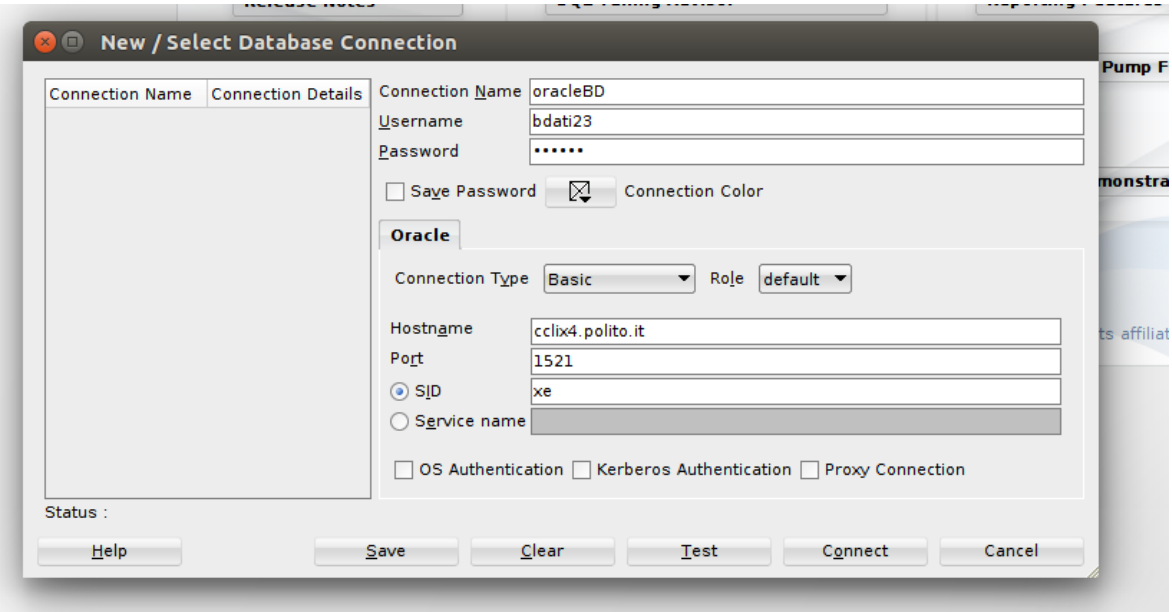


2) Login

Autenticarsi inserendo i seguenti dati:

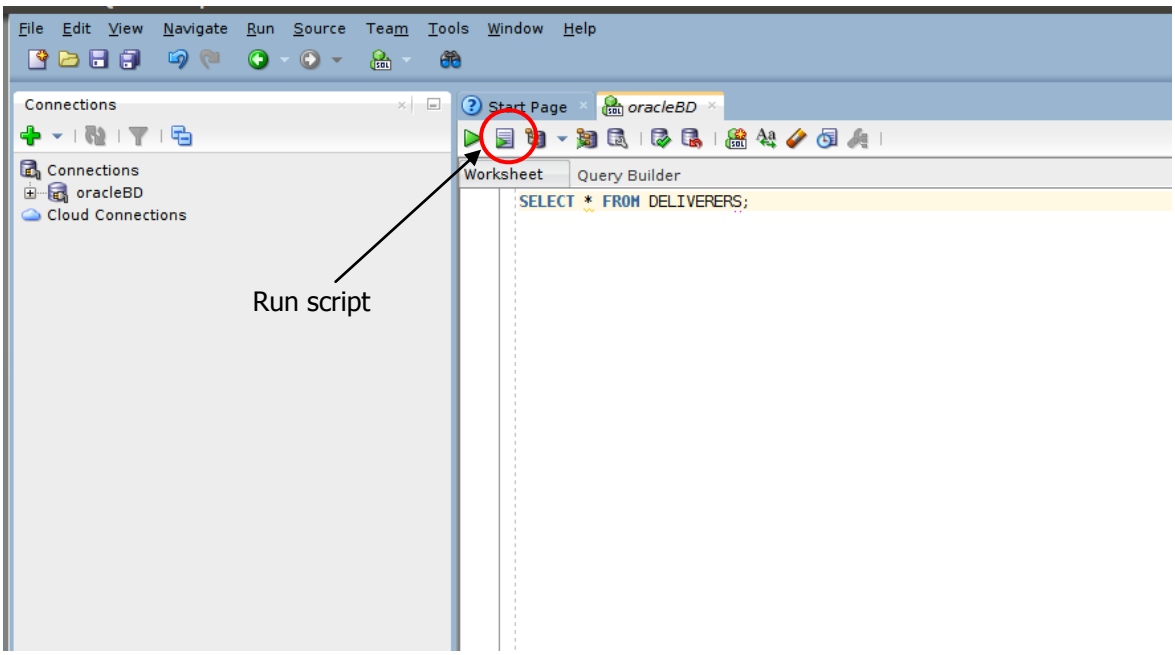
- Connection name: oracleBD
- Username: bdati*XY*
  - *XY* indica le ultime due cifre del numero del pc utilizzato
- Password: orac*XY*
  - *XY* indica le ultime due cifre del numero del pc utilizzato
- Hostname: cclix4.polito.it
- Port: 1521
- SID: xe

Ad esempio, collegandosi dalla macchina numero 23 del laboratorio, usare come username bdati23 e come password orac23.



Scrittura ed esecuzione di script SQL

Scrivere lo script SQL da eseguire nell'area di lavoro (Worksheet) e eseguirlo premendo il tasto "Run script". Usando i comandi presenti sotto la voce File è possibile salvare uno script o caricarne uno già esistente.



Creazione della base di dati

[DA FARE SOLO LA PRIMA VOLTA]

- Scaricare lo script **esercitazionejdbcOracle.sql** dalla pagina web del corso: http://dbdmg.polito.it/wordpress/teaching/04afqoa-basi-di-dati-ing-informatica-iii-anno/
- Aprire e successivamente eseguire lo script usando SQL Developer.

1. Descrizione della base di dati Corsi di informatica

La base di dati *Corsi di informatica* raccoglie informazioni relative ai corsi di una piccola società che gestisce dei corsi di informatica. La base di dati memorizza l'informazione sui corsi disponibili, sui clienti e sulle iscrizioni dei clienti ai corsi.
Il modello E-R presente in Figura 1 rappresenta il modello concettuale della base di dati *Corsi di informatica*.

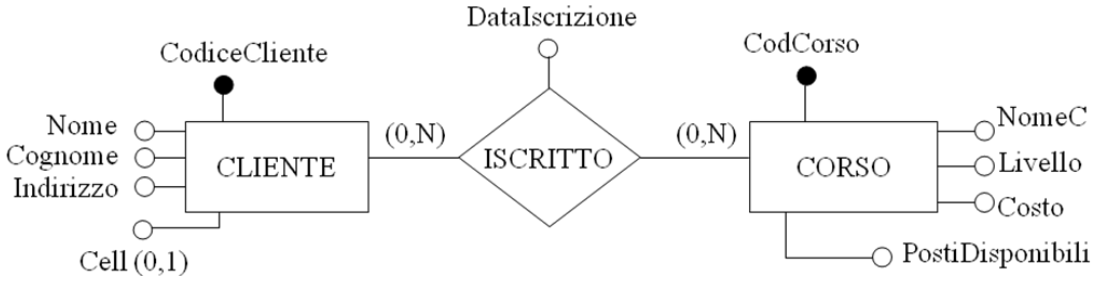


Figura 1

La base di dati è caratterizzata dal seguente schema logico (le chiavi primarie sono sottolineate):

CLIENTE (CodiceCliente, Nome, Cognome, Indirizzo, Cell*)

CORSO (CodCorso, NomeC, Livello, PostiDisponibili)

ISCRITTO_CORSO (CodiceCliente, CodiceCorso, DataIscrizione)

Ogni cliente è univocamente identificato da un codice ed è caratterizzato da nome, cognome, indirizzo e numero di cellulare (se disponibile).

I corsi sono univocamente identificati da un codice e sono caratterizzati da un nome, dal livello (un numero intero compreso tra 1 e 4) e dal numero di posti disponibili. **Nota.** L'attributo PostiDisponibili contiene l'informazione sul numero di posti ancora disponibili. Il suo valore viene aggiornato in seguito ad ogni nuova iscrizione.

Per ogni cliente si memorizzano nella base di dati i corsi cui è iscritto e la date di iscrizione ai vari corsi. Ogni cliente può iscriversi a più corsi. **Attenzione.** Un cliente può iscriversi ad un corso solo se il corso di suo interesse ha ancora almeno un posto disponibile (attributo PostiDisponibili maggiore o uguale a 1)

Lo script **esercitazionejdbcOracle.sql** crea le tabelle appena descritte e carica dei dati iniziali al loro interno (il contenuto di ogni tabella è visualizzabile sotto SQL Developer cliccando sul nome della tabella e poi selezionando il tab/sottoform *Data*).

2. Informazioni utili per la compilazione e l'esecuzione del codice Java da realizzare

La classe che implementa il driver JDBC di Oracle da utilizzare per connettersi alla base di dati è **oracle.jdbc.driver.OracleDriver**. Tale classe è inclusa nella libreria Java **classes12.jar** presente sul sito del corso. Tale libreria deve essere inclusa tra le libreria nel proprio progetto Eclipse per poter accedere alla base di dati Oracle.

L'invocazione del metodo *getConnection(String url, String username, String password)* per la creazione della connessione verso la base di dati deve essere eseguita impostando i seguenti valori per i tre parametri:

- url: "jdbc:oracle:thin:@cclix4.polito.it:1521:XE"
- username: "bdati*XY*"
- password: "orac *XY*"

dove *XY* indica le ultime due cifre del numero del pc utilizzato.

3. Applicazione da realizzare

L'applicazione da realizzare consiste in una semplice applicazione grafica che permette di gestire le seguenti operazioni:

- Visualizzazione dei dati di un cliente e elenco dei corsi cui il cliente è iscritto
- Iscrizione dei clienti ai corsi presenti nella base di dati

L'obiettivo di questa esercitazione consiste nel capire il funzionamento dei metodi Java che permettono di accedere ad una base di dati. Per questo motivo vi forniamo come base di partenza un progetto Eclipse che contiene già tutte le classi necessarie per la realizzazione dell'applicazione da realizzare, e in particolare quelle per la gestione delle finestre. Il progetto Eclipse da cui partire è presente nel file **esercitazioneJDBC.zip**, scaricabile dal sito del corso.

Quello che dovete fare è implementare alcuni metodi, attualmente vuoti o contenenti del codice "statico", che servono per accedere alla base di dati. I metodi da implementare sono quelli della classe it.polito.db.DB (file sorgente DB.java). Tutte le altre classi del progetto non necessitano di essere modificate.

Prima di provare a modificare il codice, importare il progetto Eclipse dentro l'ambiente di lavoro Eclipse (selezionare la versione Eclipse 4.4 (Luna) Standard SDK tra le varie versioni disponibili sui PC del LABINF), compilare il progetto e provare ad eseguirlo.

Per l'importazione del progetto fonitovi dentro Eclipse:

- Lanciare Eclipse
- Seleziona la voce di menu File → Import
- Nella finestra di dialogo che si apre selezionare "General → Existing Projects into Workspace" e premere il tasto "Next".
- Nella finestra successiva selezionare come "Select root directory" la cartella nella quale avete decompresso il contenuto del file esercitazioneJDBC.zip e premete "Finish".

Alla fine delle operazioni di importazione dentro Eclipse avrete a disposizione un progetto dal nome **esercitazioneJDBCOracle**. Questo progetto è il punto di partenza che dovete modificare per realizzare l'applicazione richiesta.

Il metodo main si trova nella classe it.polito.Main. Provate ad eseguire il progetto (come una Java Application) per verificare che l'importazione sia avvenuta con successo. Il codice attuale crea una prima finestra contenente due pulsanti ("Info Cliente" e "Iscrizione Corsi"), associati alle due operazioni citate all'inizio della

sezione. Prendo il primo pulsante ("Info Cliente") si apre una seconda finestra che permette di specificare il codice di un cliente (textbox Codice Cliente) e visualizza i dati del cliente, più relativi corsi, premendo il tasto "Info". La versione attuale del codice visualizza sempre le stesse informazioni indipendentemente dal codice fornito in ingresso in quanto non accede alla base di dati.

Premendo il tasto "Iscrizione Corsi" della finestra principale si apre una finestra che permette di iscrivere i clienti ai corsi. Tramite un menu a tendina si seleziona il corso di interesse, poi si inserisce il codice del cliente che si vuole iscrivere al corso selezionato nella textbox apposita e infine, premendo il tasto "Iscrivi" si effettua l'iscrizione del cliente al corso selezionato. Anche questa parte al momento è "statica", ossia indipendentemente dal corso e dal codice cliente inserito il programma dirà sempre che l'iscrizione è andata a buon fine anche se in realtà non è successo nulla perché l'applicazione attuale non accedere alla base di dati.

Sono di seguito elencati i metodi della classe it.polito.db.DB (file sorgente DB.java) da implementare al fine di fare in modo che l'applicazione acceda alla base di dati. Per ogni metodo da implementare è fornita una breve descrizione relativa a cosa deve fare.

- *public DB()* - costruttore della classe it.polito.db.DB

Nel costruttore della classe DB si deve instanziare/registrare il Driver JDBC usato per accedere alla base di dati. Nel caso di Oracle la classe che implementa il Driver JDBC per oracle è **oracle.jdbc.driver.OracleDriver**

- *public boolean OpenConnection()*

Questo metodo crea la connessione verso la base di dati e memorizza tale connessione nella variabile di classe *conn*

- *public String ottieniDatiCliente(long codice_cliente)*

Questo metodo riceve come parametro di input il codice di un cliente e restituisce i dati di tale cliente sottoforma di stringa. Le informazioni sul cliente devono essere reperite interrogando in modo opportuno la base di dati. La stringa restituita dal metodo è generata concatenando i nomi degli attributi che caratterizzano i clienti e i valori di tali attributi per il cliente con codice identificativo uguale a *codice_cliente*. Se ad esempio il cliente selezionato ha nome "Paolo", cognome "Garza", indirizzo "Via Inesistente 24, Torino" e numero di cellulare "0110907022" allora il metodo deve restituire la stringa "Nome: Paolo\nCognome: Garza\nIndirizzo: Via Inesistente 24, Torino\nCell: 0110907022". Vedere l'esempio di codice attualmente presente nel progetto fornito per capire meglio come formattare la stringa da restituire.

Se il cliente con codice uguale a quello presente in *codice_cliente* non esiste nella base di dati il metodo restituisce la stringa "Cliente inesistente".

- *public List<String> ottieniCorsiCliente(long codice_cliente)*

Questo metodo riceve come parametro di input il codice di un cliente e restituisce i nomi dei corsi cui il cliente è iscritto usando una lista di stringhe. Ogni stringa presente nella lista restituita corrisponde al nome di uno dei corsi cui il cliente con codice pari a *codice_cliente* è iscritto.

- *public List<String> ottieniCodiciCorsi()*

Questo metodo restituisce l'elenco di corsi per cui c'è ancora almeno un posto libero (ossia i corsi presenti nella tabella *Corso* per i quali l'attributo *PostiDisponibili* è maggiore o uguale a 1). L'elenco è restituito dal metodo come lista di stringhe. Ogni stringa corrisponde ad uno dei corsi selezionati dall'interrogazione e contiene codice e nome del corso. Ogni stringa è formattata usando il formato "codice_corso - nome_corso". Vedere l'esempio di codice attualmente presente nel progetto fornito per capire meglio come formattare ogni singola stringa della lista di stringhe da restituire.

- *public boolean aggiungiIscrizione(long codCorso, long codCliente)*

Questo metodo riceve in ingresso (parametri di input) il codice di un corso e il codice di un cliente e iscrive il cliente al corso (ossia aggiunge una nuova iscrizione nella base di dati). L'iscrizione consiste nell'inserimento della tupla appropriata nella tabella *Iscritto_Corso*. Una volta aggiunta l'iscrizione nella tabella *Iscritto_Corso*, il metodo deve aggiornare (decrementandolo di uno) il valore dei posti disponibili per il corso indicato in *codCorso* (praticamente si deve eseguire un'istruzione SQL di aggiornamento che va a decrementare di uno il valore del campo *Corso.PostiDisponibili* per il corso *codCorso*). Il metodo ritorna *true* se l'iscrizione è avvenuta correttamente, *false* altrimenti.

- *public void CloseConnection()*

Questo metodo chiude la connessione verso la base di dati.