Ho inizializzato delle variabili che rappresentano l’URL del mio database, connection degli oggetti, connection e statement che ci serviranno per connetterci ed eseguire le query e poi un opggetto scanner che serve semplicemente per prendere l’input.

La prima funzione è la CONNESSIONE. La connessione riceve un username e la password e avvia la connessione attraverso il driver di JDBC con la funzione getconnecction .

Con la funzione getconnection gli passiamo l’url che abbiamo memorizzato all’inizio, l’username e la password che ci sono state date e se la connesione è avvenuta con successo ci dà un output altrimenti l’errore viene catturato.  
Quindi la connessione è abbastanza semplice, getconnection richiede la connessione a questo URL e poi la andiamo a memorizzare nella variabile connection (in alto).

**SELECT**

Abbiamo principalmente due cose da utilizzare. In primis dobbiamo formare la query, se scriviamo ALL andiamo a selezionare tutto dalla tabella Dipendente, se invece non scriviamo ALL ma qualcos’altro andiamo nell’else e prendiamo la stringa come condizione e quindi la scriviamo dopo il WHERE (sqlQuery=”SELECT \* FROM Dipendente WHERE “+condizione; ), ad esempio Where Nome = Paolo e va a metterla lì.

Così formiamo la query, dopo aver formato la query andiamo a creare lo **statement**, lo statement è il pezzo fondamentale che serve ad eseguire le query sul database. Quindi statement.executeQuery e gli passiamo la query che abbiamo scritto sopra (ndr in sqlQuery), questo che cosa produce? Un **ResultSet** prende tutte le colonne che sono state date come un risultato e si può scorrere premendo .next, fondamentalmente come se fosse una specie di array di tuple. Prende in input il risultato della query e poi te lo dà come se fosse un array, quindi fai .next e poi ti vai a prendere tutte le stringhe che si trovano all’interno (CF, Nome, Cognome, DdN, Via, Civico, Cap)con le funzioni getString e getInt (per il cap).  
Una volta preso il risultato te lo vai a mettere in una stringa dove appunto sintetizzi tutte le stringhe che hai memorizzato, una volta che ti esce questa stringa risultante fai risultato.close(); perché il resultset si deve chiudere e poi alla fine ritorniamo il risultato che la stringa che ci è uscita

Per le altre il concetto è lo stesso, solo che, poiché sono tutti update, anziché fare executeQuery facciamo executeUpdate, perché l’inserimento è un aggiornamento, l’aggiornamento e anche il delete sono un aggiornamento. La differenza principale è che queste query non ritornano un risultato, **sono solo o positive o negative, nel senso o è riuscito o non è riuscito**. Mente invece con la select hai proprio dei dati come risultato.

INSERT  
Appunto facciamo executeUpdate proprio perché questo è un update, la query viene formata tramite i parametri che ci vengono passati (in questo caso ci vengono passati dall’interfaccia grafica, il CF, nome, cognome, ddN, via, civico, cap). Formiamo la query e poi la eseguiamo attraverso executeUpdate

UPDATE  
Per l’update chiaramente la storia è simile, prendiamo la chiave primaria, la colonna da modificare e il valore che vogliamo inserire.  
Poiché c’è una distinzione tra mettere l’apice e non metterlo, allora andiamo a vedere se c’è un int. Se è un intero allora non mettiamo gli apici, se invece è una stringa allora andiamo a mettere gli apici fra il valore.  
Quindi formiamo la query e poi c’è sempre l’executeUpdate. Una volta che lo eseguiamo sta nel blocco trycatch, se è riuscito esce “aggiornamento riuscito” sennò stampa l’errore.

DELETE

Formiamo la query, facciamo l’update (quindi executeUpdate), se è riuscito esce “Operazione riuscita” altrimenti ti dà l’errore.

I passi base sono sempre 3:  
- Il primo è formare la query, quando noi andiamo a fare sqlQuery = qualcosa…  
- Il secondo è eseguirla, quindi fare executeQuery oppure executeUpdate per le altre

* Il terzo è dare un risultato (result+=… Oppure negli altri casi dare l’esito esempio “inserimento riuscito”)