

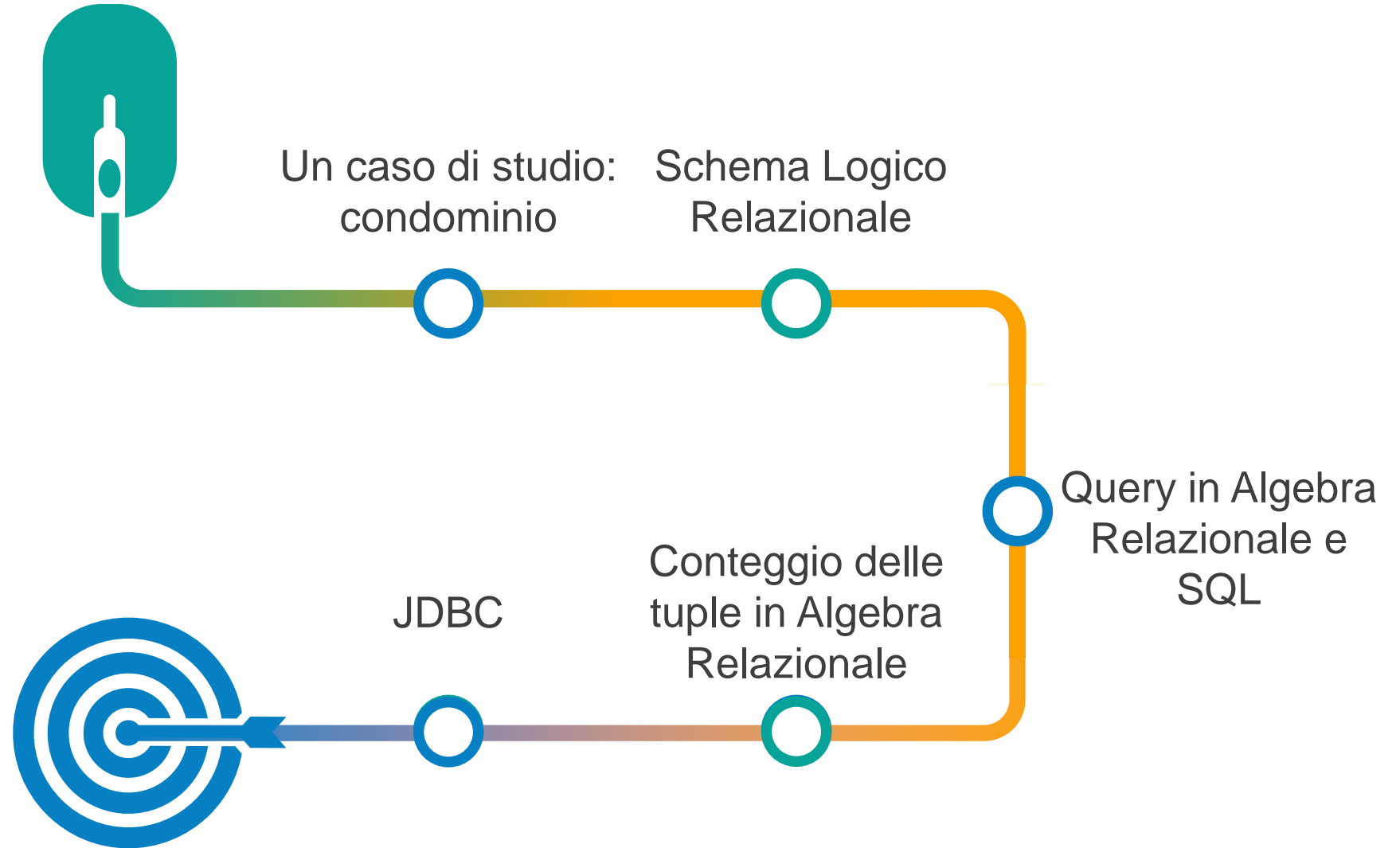


BASI DI DATI QUERY

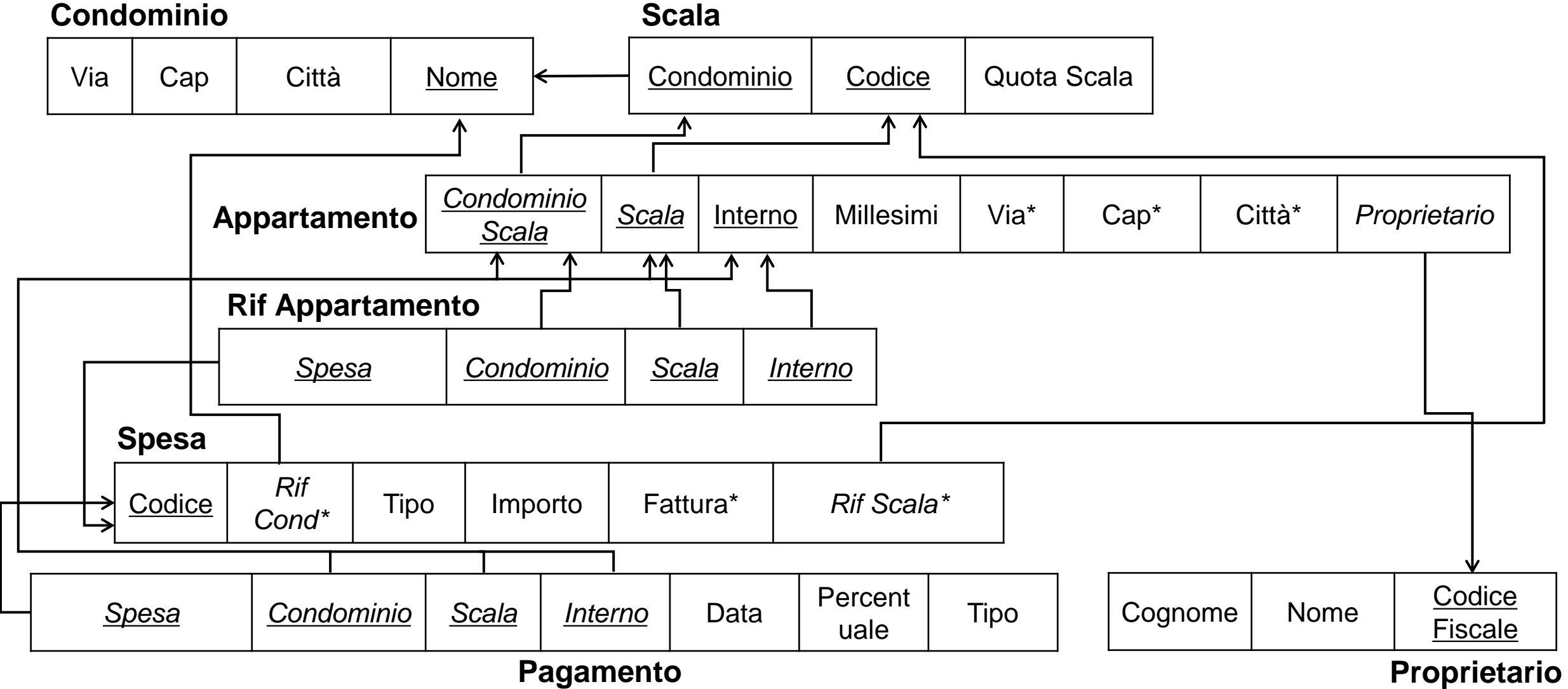
Polese G. Caruccio L. Breve B.

a.a. 2023/2024

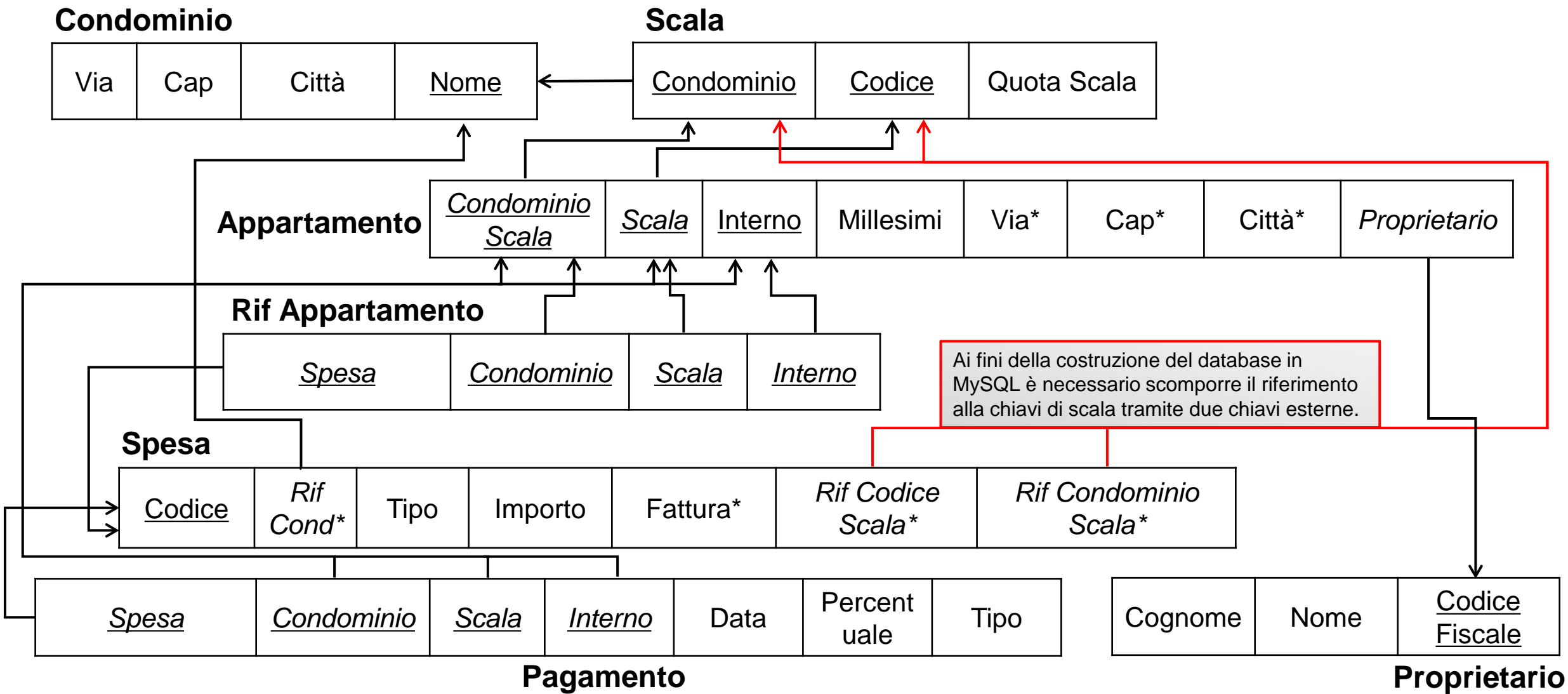
Outline



MAPPING NELLO SCHEMA LOGICO RELAZIONALE



MAPPING NELLO SCHEMA LOGICO RELAZIONALE



QUERY

- (algebra - SQL) Elencare le vie di tutti i condomini
- (SQL) Elencare tutte le spese di condominio ordinate per condominio e per importo (rifCond, importo, fattura)
- (algebra - SQL) Elencare le spese in cui è coinvolto l'interno "2" scala "B" del condominio "Alighieri" (codice, importo, tipo)
- (SQL) Elencare i proprietari il cui cognome inizia con "M" (cognome, nome)
- (algebra - SQL) Elencare le spese corredate di fattura (codice, importo, fattura)

QUERY 1

- (algebra - SQL) Elencare le vie di tutti i condomini

$\text{PROJ}_{\text{via}}(\text{condominio})$

```
SELECT DISTINCT via  
FROM condominio;
```

QUERY 2

- (SQL) Elencare tutte le spese di condominio ordinate per condominio e per importo (rifCond, importo, fattura)

```
SELECT rifCond, importo, fattura
      FROM spesa
     WHERE tipo = 'condominio'
    ORDER BY rifCond, importo;
```


QUERY 3

- (algebra) Elencare le spese in cui è coinvolto l'interno "2" scala "B" del condominio "Alighieri" (codice, importo, tipo)

PROJ codice, tipo, importo (**SEL** (rifCond='Alighieri' \wedge tipo='condominio') \vee (rifCodiceScala='B' \wedge rifCondScala='Alighieri' \wedge tipo='scala')) (spesa))
U

PROJ codice, tipo, importo (spesa **JOIN** codice=spesa (**SEL** condominio='Alighieri' \wedge scala='B' \wedge interno='2' (rif_appartamento)))

QUERY 3

- (SQL) Elencare le spese in cui è coinvolto l'interno "2" scala "B" del condominio "Alighieri" (codice, importo, tipo)

```
SELECT codice, tipo, importo
```

```
FROM spesa
```

```
WHERE (rifCond='Alighieri' AND tipo='condominio') OR (rifCodiceScala='B' AND  
rifCondScala='Alighieri' AND tipo='scala')
```

```
UNION ALL
```

```
SELECT codice, tipo, importo
```

```
FROM spesa JOIN rif_appartamento ON spesa.codice=rif_appartamento.spesa
```

```
WHERE condominio='Alighieri' AND scala='B' AND interno=2
```

QUERY 4

- (SQL) Elencare i proprietari il cui cognome inizia con "M" (cognome, nome)

```
SELECT cognome, nome  
FROM proprietario  
WHERE cognome LIKE 'M%';
```

QUERY 5

- (algebra - SQL) Elencare le spese corredate di fattura (codice, importo, fattura)

PROJ codice, importo, fattura (**SEL** fattura **IS NOT NULL** (spesa))

```
SELECT codice, importo, fattura  
      FROM spesa  
      WHERE fattura IS NOT NULL;
```

SQL

- (algebra - SQL) Elencare i proprietari degli appartamenti che hanno effettuato almeno 2 spese di appartamento (cognome, nome)
- (algebra - SQL) Elencare gli appartamenti che hanno pagato spese di Appartamento ma non di Scala (condominio, scala, interno)
- (SQL) Calcolare il numero di pagamenti effettuati dall'appartamento interno "2" scala "B" del condominio "Alighieri" per le spese di tipo "Condominio" (numPagamenti)
- (SQL) Elencare i proprietari a cui sono associate spese di appartamento per un importo superiore a 500 euro (cognome, nome)

QUERY 6

- (algebra) Elencare i proprietari degli appartamenti che hanno effettuato almeno 2 spese di appartamento (cognome, nome)

```
PROJcognome, nome(proprietario
  JOINcodiceFiscale=proprietario
  (PROJproprietario(appartamento
    JOIN
    (SELspesa<>spesa1(rif_appartamento
      JOIN
      (RENspesa1<-spesa(rif_appartamento)))))))
```

QUERY 6

- (SQL) Elencare i proprietari degli appartamenti che hanno effettuato almeno 2 spese di appartamento (cognome, nome)

```
SELECT cognome, nome
```

```
FROM proprietario
```

```
WHERE codiceFiscale IN (SELECT proprietario
```

```
FROM appartamento NATURAL JOIN rif_appartamento GROUP BY  
condominio, scala, interno, proprietario
```

```
HAVING COUNT(*) > 1);
```

QUERY 7

- (algebra) Elencare gli appartamenti che hanno pagato spese di Appartamento ma non di Scala (condominio, scala, interno)

PROJcondominio,scala,interno(pagamento **JOIN**_{spesa=codice}
(**SEL**_{tipo='Appartamento'}(spesa)))

-

PROJcondominio,scala,interno(pagamento **JOIN**_{spesa=codice}
(**SEL**_{tipo='Scala'}(spesa)))

QUERY 7

- (SQL) Elencare gli appartamenti che hanno pagato spese di Appartamento ma non di Scala (condominio, scala, interno)

```
SELECT condominio, scala, interno
FROM pagamento INNER JOIN spesa ON pagamento.spesa = spesa.codice
WHERE spesa.tipo = 'Appartamento' AND NOT EXISTS (
SELECT condominio, scala, interno
FROM pagamento AS p INNER JOIN spesa AS s ON p.spesa = s.codice
WHERE s.tipo = 'Scala' AND pagamento.condominio = p.condominio AND
pagamento.scala = p.scala AND pagamento.interno = p.interno);
```

QUERY 8

- (SQL) Calcolare il numero di pagamenti effettuati dall'appartamento interno "2" scala "B" del condominio "Alighieri" per le spese di tipo "Condominio" (numPagamenti)

```
SELECT COUNT(*) AS numPagamenti
FROM pagamento
WHERE interno = '2' AND scala = 'B' AND condominio =
    'Alighieri'
AND spesa IN (SELECT codice
    FROM spesa
    WHERE tipo = 'Condominio');
```

QUERY 9

- (SQL) Elencare i proprietari a cui sono associate spese di appartamento per un importo superiore a 500 euro (cognome, nome)

```
SELECT cognome, nome
```

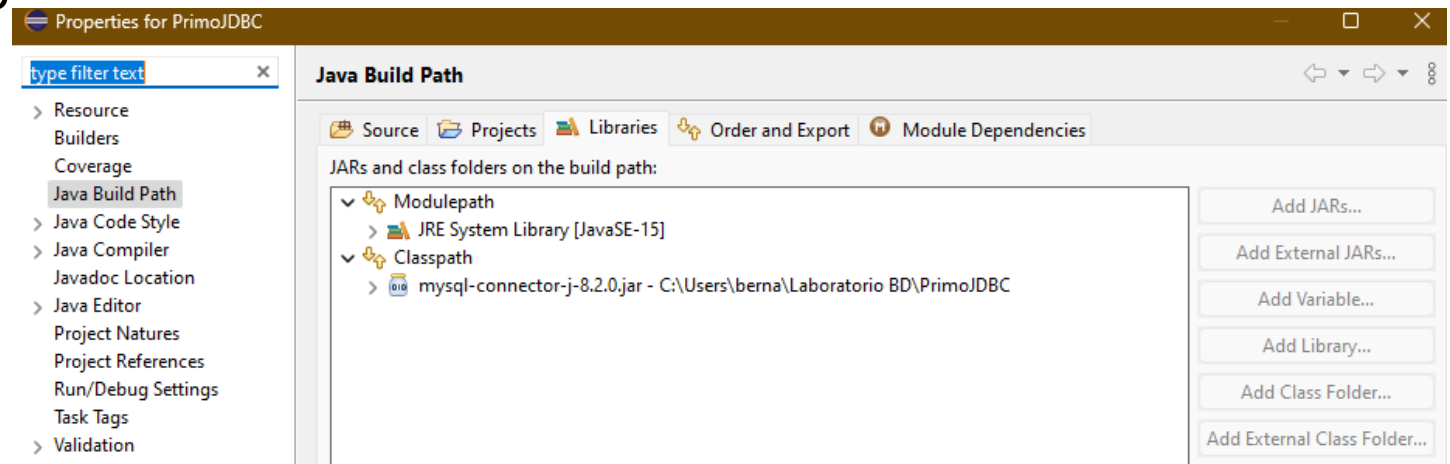
```
FROM proprietario JOIN appartamento ON  
    proprietario.codiceFiscale = appartamento.proprietario  
NATURAL JOIN rif_appartamento
```

```
WHERE rif_appartamento.spesa = ANY (SELECT codice  
    FROM spesa
```

```
WHERE tipo = 'Appartamento' AND importo > 500);
```

Applicazione Java con JDBC

- Scaricare il file mysql-connector-java-8.0.xx.jar
- Creare un nuovo progetto Java
 - File > New > Java Project
 - Definire il nome del progetto
- Importare il file per la connessione con il DBMS
 - Tasto destro sul nome del progetto
 - Build Path > Configure Build Path > Add External JARs
 - Selezionare il file .jar scaricato



Un programma con JDBC

- Usando JDBC scrivere un'applicazione che permetta l'esecuzione e la stampa dei risultati della seguente query: preso in input un intero X, estrarre i dati degli appartamenti a cui sono associate spese la cui somma è maggiore di X

```
import java.sql.*;
import java.util.*;
public class esercizioJDBC {
    public static void main(String[] arg){
        Scanner input = new Scanner(System.in);
        System.out.println("Inserisci la spesa totale attesa");
        int x = Integer.parseInt(input.nextLine());
        esercizioJDBC.executeEsercizio(x);
    }
    public static void esercizio(int x) {
        Connection con = null ;
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            String url = "jdbc:mysql://localhost:3306/condominio"
                + "?useUnicode=true&useJDBCCompliantTimezoneShift=true"
                + "&useLegacyDatetimeCode=false&serverTimezone=UTC";
            String username = "<username>"; String pwd = "<pwd>";
            con = DriverManager.getConnection(url,username,pwd);
        }
        catch(Exception e){System.out.println("Connessione fallita!!!");}
```

Un programma con JDBC

- Usando JDBC scrivere un'applicazione che permetta l'esecuzione e la stampa dei risultati della seguente query: preso in input un intero X, estrarre i dati degli appartamenti a cui sono associate spese la cui somma è maggiore di X

```
...
    try {
        String query = "SELECT condominio, scala, interno" +
            + " FROM rif_appartamento JOIN spesa ON "
            + " rif_appartamento.spesa = spesa.codice "
            + " GROUP BY condominio, scala, interno "
            + " HAVING SUM(importo) > "+x;

        Statement pquery = con.createStatement();
        ResultSet result = pquery.executeQuery(query);
        System.out.println("Ecco i condomini che hanno pagato più +
            + di "+x+" euro per spese");
        while (result.next()){
            String condominio = result.getString("condominio");
            String scala = result.getString("scala");
            int interno = result.getInt("interno");
            System.out.println(condominio+" \t "+scala+" \t "+interno);
        }
    }
    catch (Exception e){
        System.out.println("Errore nell'interrogazione");
    }
}
```