

Basi di Dati: Laboratorio



Query

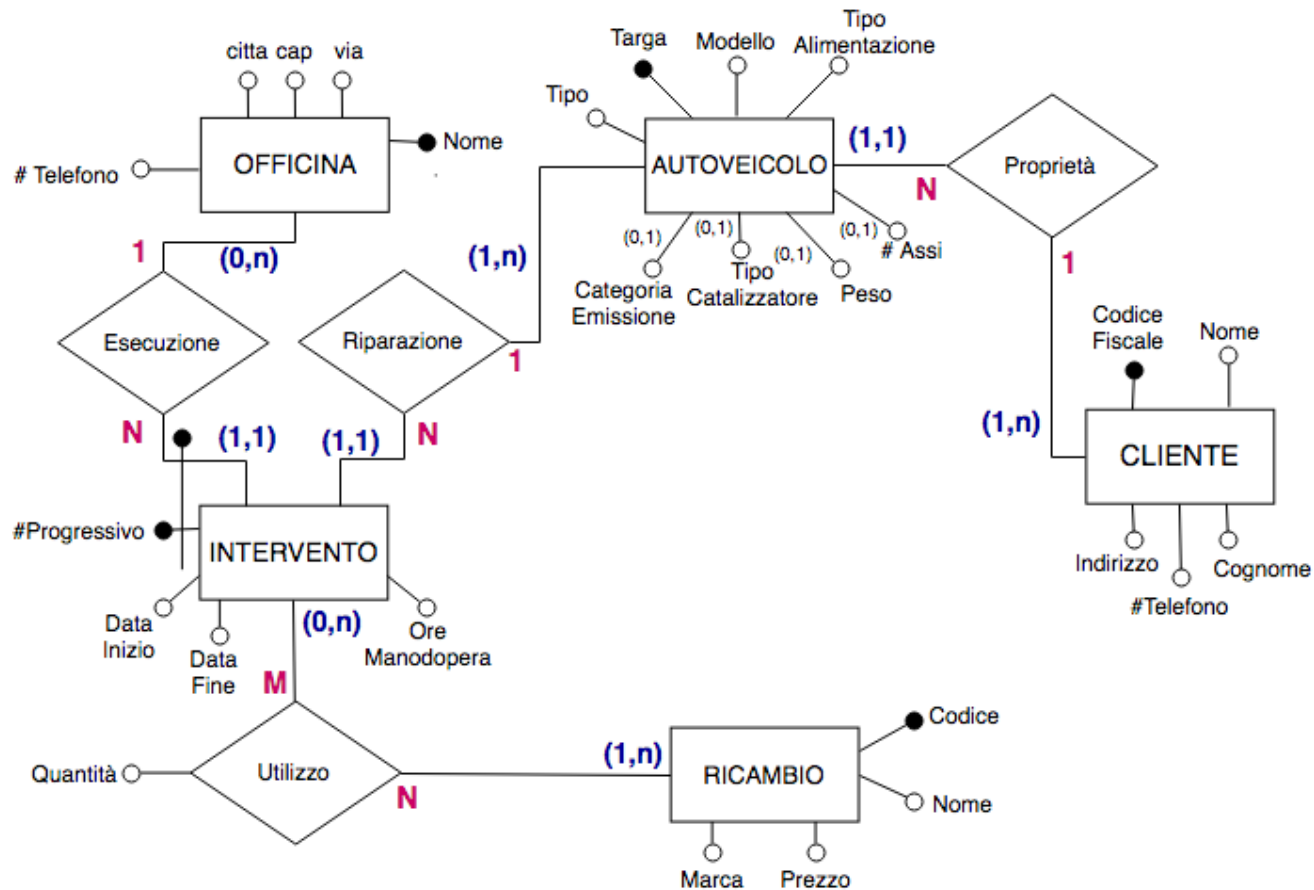
Prof. Giuseppe Polese

Dott. Stefano Cirillo

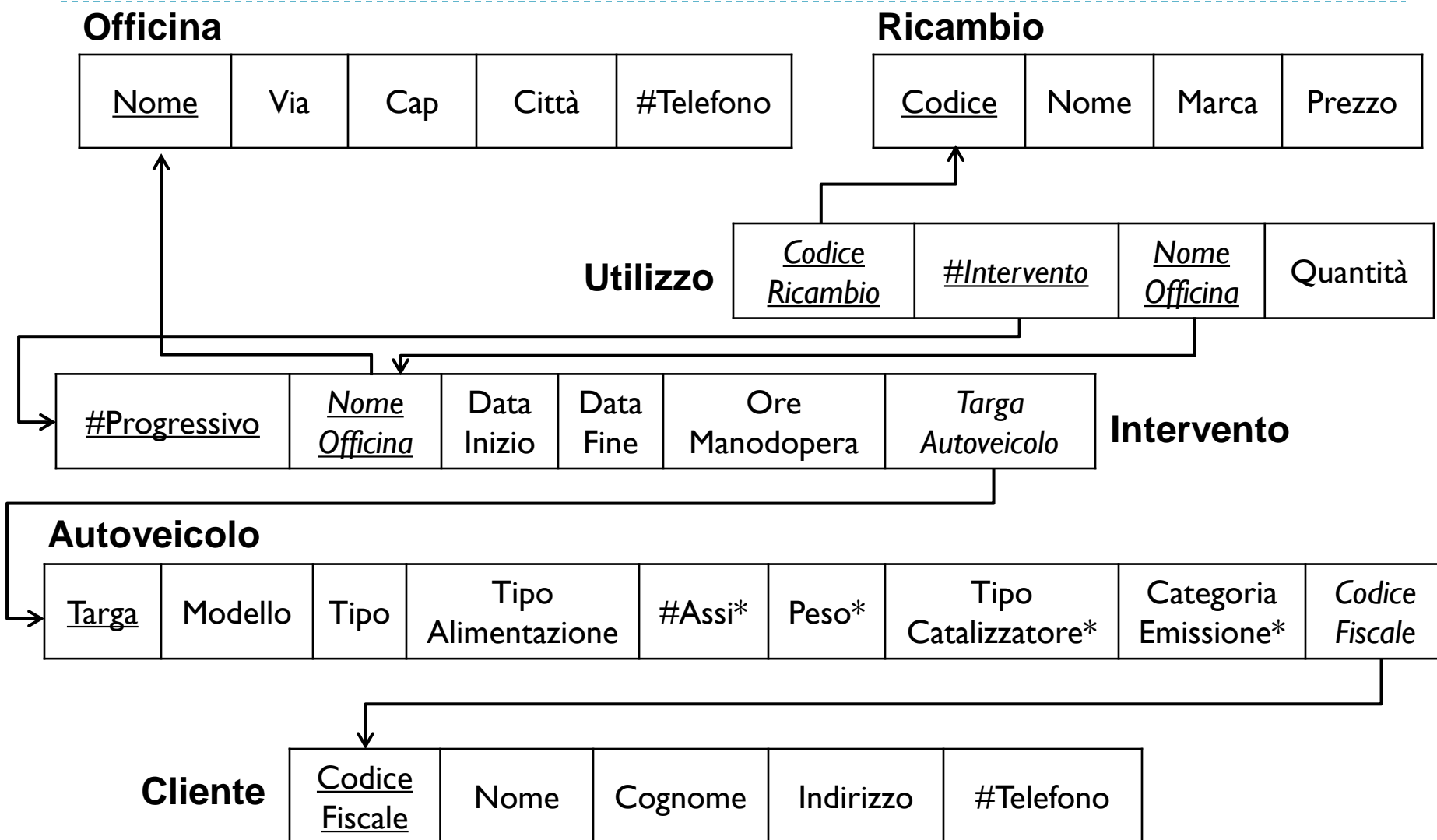
Outline

- ▶ Un caso di studio: officina
 - ▶ Schema Logico Relazionale
 - ▶ Richiami di Query in Algebra Relazionale e SQL
 - ▶ Conteggio delle tuple in Algebra Relazionale

Schema Ristrutturato



Schema Logico Relazionale



Inserimento dei dati

- Supponiamo di voler inserire i seguenti dati

officina

nome	via	cap	città	#telefono
Officina F.lli Ferraro	Lungomare C. Colombo, 307	84129	Salerno	NULL
Officina Ferraro Giuseppe	Via Madonna di Fatima, 130	84129	Salerno	NULL
Meccanica Ferraro A. & Figli	Via Venezia, 23	84098	Pontecagnano (SA)	089221154

ricambio

codice	nome	marca	prezzo
598382	Kit pastiglie freno, Freno a disco	Valeo	90,65
CT704	Cinghia dentata	CONTITECH	56,67
0451103351	Filtro olio	BOSCH	17,93

Inserimento dei dati

- Supponiamo di voler inserire i seguenti dati

Autoveicolo

targa	modello	tipo alimen.	tipo	#assi	peso	tipo catal.	Cat. Emis.	Codice Fiscale
FF111	Opel Corsa	GPL	2	NULL	NULL	A	Euro 6	LNEBCH05B41G230M
AB123	Fiat Punto	Benzina	2	NULL	NULL	B	Euro 4	MRARSS60D63A717X
CA171	Eurocarco 190E30	Diesel	1	4X2	9T	NULL	NULL	LNEBCH05B41G230M

cliente

codice fiscale	nome	cognome	città	telefono
LNEBCH05B41G230M	Elena	Bianchi	Via Napoli ,15 Pagani	NULL
MRARSS60D63A717X	Maria	Rossi	Via Ripa, 128 Battipaglia	0828444444
STFVRD99R07H703G	Stefano	Verdi	P.zza XXIV Maggio 3, Salerno	089151515

Caricamento dei dati

- Supponiamo di voler i seguenti dati

intervento

#prog	nome officina	data inizio	data fine	ore manodopera	targa autoveicolo
1	Meccanica Ferraro A. & Figli	2018-09-13	2018-09-14	5	CA171
1	Officina Ferraro Giuseppe	2018-10-22	2018-10-24	10	FF111
2	Meccanica Ferraro A. & Figli	2018-12-03	2018-12-03	2	CA171

utilizzo

codice ricambio	#intervento	nome officina	quantità
598382	1	Meccanica Ferraro A. & Figli	2
CT704	1	Officina Ferraro Giuseppe	1
0451103351	2	Meccanica Ferraro A. & Figli	1
CT704	1	Meccanica Ferraro A. & Figli	1

Inserimento di dati

- ▶ Vediamo come si possono inserire i dati all'interno delle tabelle
 - ▶ Il comando per inserire i dati in una tabella è il seguente:

```
INSERT INTO officina(nome, via, cap, città)
VALUES("Officina F.lli Ferraro", "Lungomare C. Colombo,
      307", "84129", "Salerno");
```

```
mysql> INSERT INTO officina(nome,via,cap,città)
-> VALUE("Officina Ferraro Giuseppe", "via Madonna di Fatima 130", "84129", "Salerno");
```


Inserimento di dati

- ▶ Vediamo come si possono inserire i dati all'interno delle tabelle
 - ▶ Il comando per inserire i dati in una tabella è il seguente:

```
INSERT INTO officina(nome, via, cap, città)
VALUES("Officina F.lli Ferraro", "Lungomare C. Colombo,
      307", "84129", "Salerno");
```

```
mysql> INSERT INTO officina(nome,via,cap,città)
-> VALUE("Officina Ferraro Giuseppe", "via Madonna di Fatima 130", "84129", "Salerno");
```

- ▶ Link Database Officina: <https://we.tl/t-ELyIhhfAZo>

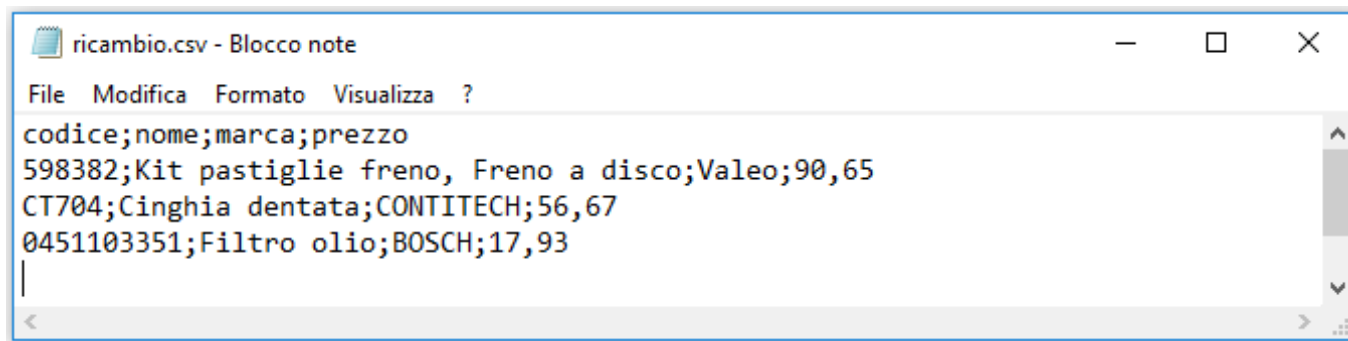
Caricamento dei dati (SQL)

- ▶ Un modo alternativo per caricare i dati si basa sull'utilizzo dei file .CSV
 - ▶ Il file .csv deve contenere una riga di valori per ogni tupla da inserire
 - ▶ Un file così costruito potrà essere automaticamente caricato mediante il seguente comando:

```
LOAD DATA LOCAL INFILE '<pathFile>' INTO TABLE  
  <tableName> FIELDS TERMINATED BY ';' LINES  
  TERMINATED BY '\n' (<attributeList>);
```

Caricamento dei dati (SQL)

- ▶ Un modo alternativo per caricare i dati si basa sull'utilizzo dei file .csv
 - ▶ Il file .csv deve contenere una riga di valori per ogni tupla da inserire



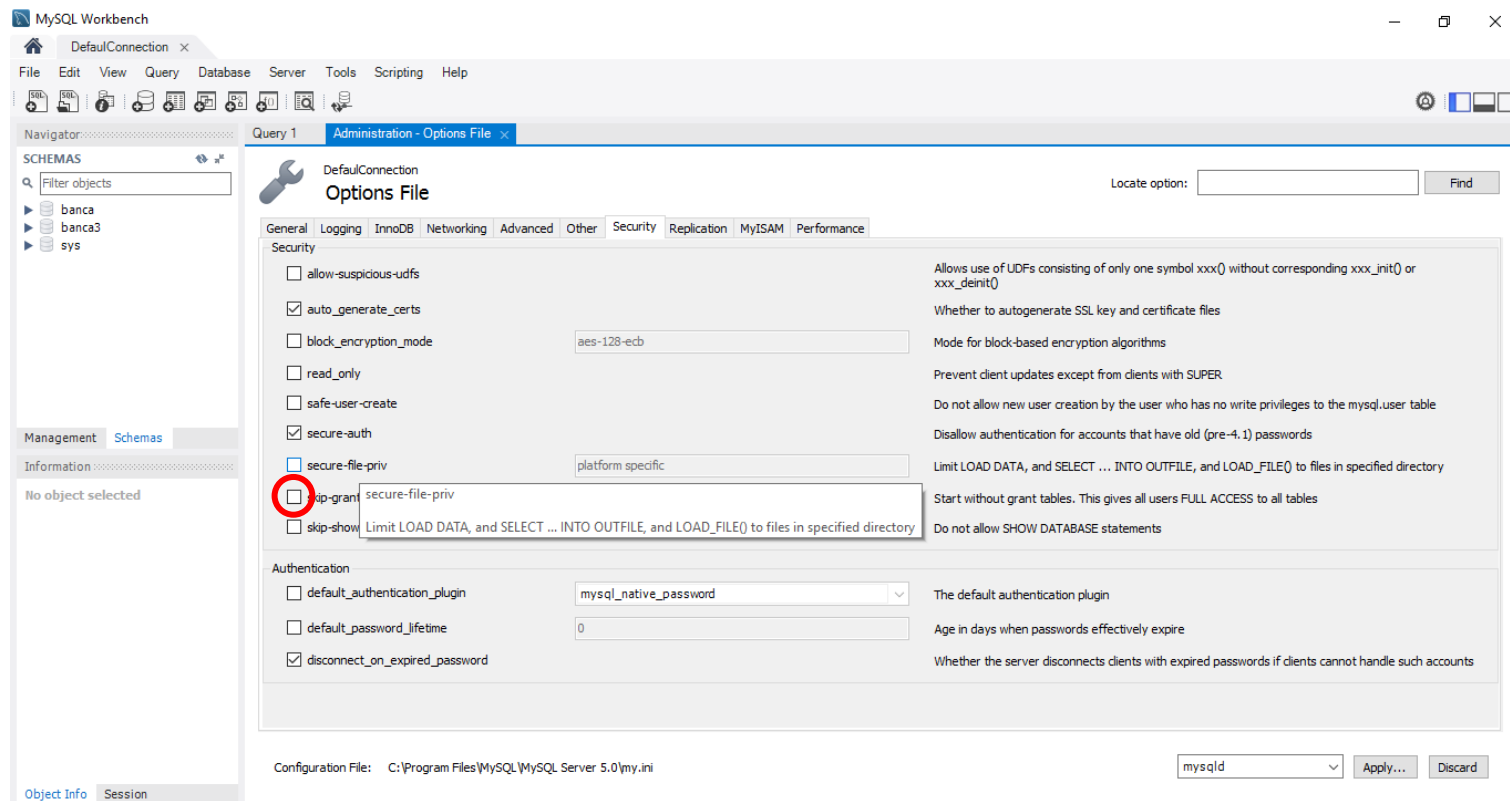
```
LOAD DATA INFILE 'C:\\Utenti\\Loredana\\Desktop\\ricambio.csv'
INTO TABLE ricambio
FIELDS TERMINATED BY ';'
LINES TERMINATED BY '\\n'
(codice,nome,marca,prezzo);
```

Problema Workbench e LOAD DATA

- ▶ In MySQL Workbench potreste avere problemi ad eseguire il comando **LOAD DATA LOCAL INFILE** ...
- ▶ È necessario deselezionare la policy **secure-file-priv**
 - ▶ **Server > Options File >**
 - ▶ Attivare il tab **Security**
 - ▶ Deselezionare la policy **secure-file-priv**
 - ▶ Cliccare su **Apply**

Problema Workbench e LOAD DATA

- In MySQL Workbench potreste avere problemi ad eseguire il comando **LOAD DATA LOCAL INFILE ...**



Caricamento dei dati (Workbench)

1. Creare il Database e le Tabelle
2. Creare/Scaricare il/i file CSV con i dati

autoveicolo.csv

cliente.csv

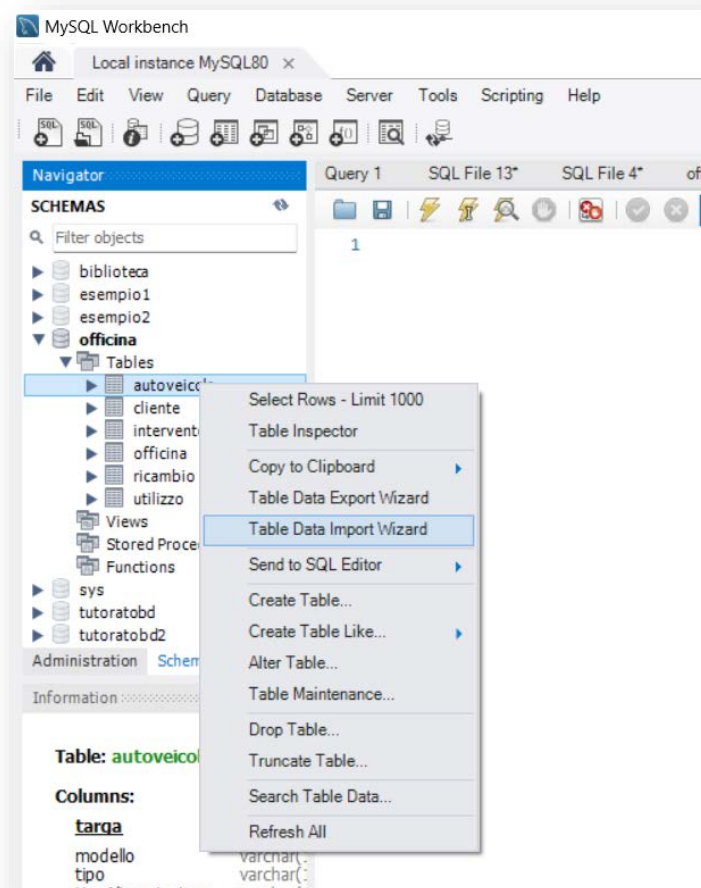
intervento.csv

officina.csv

ricambio.csv

utilizzo.csv

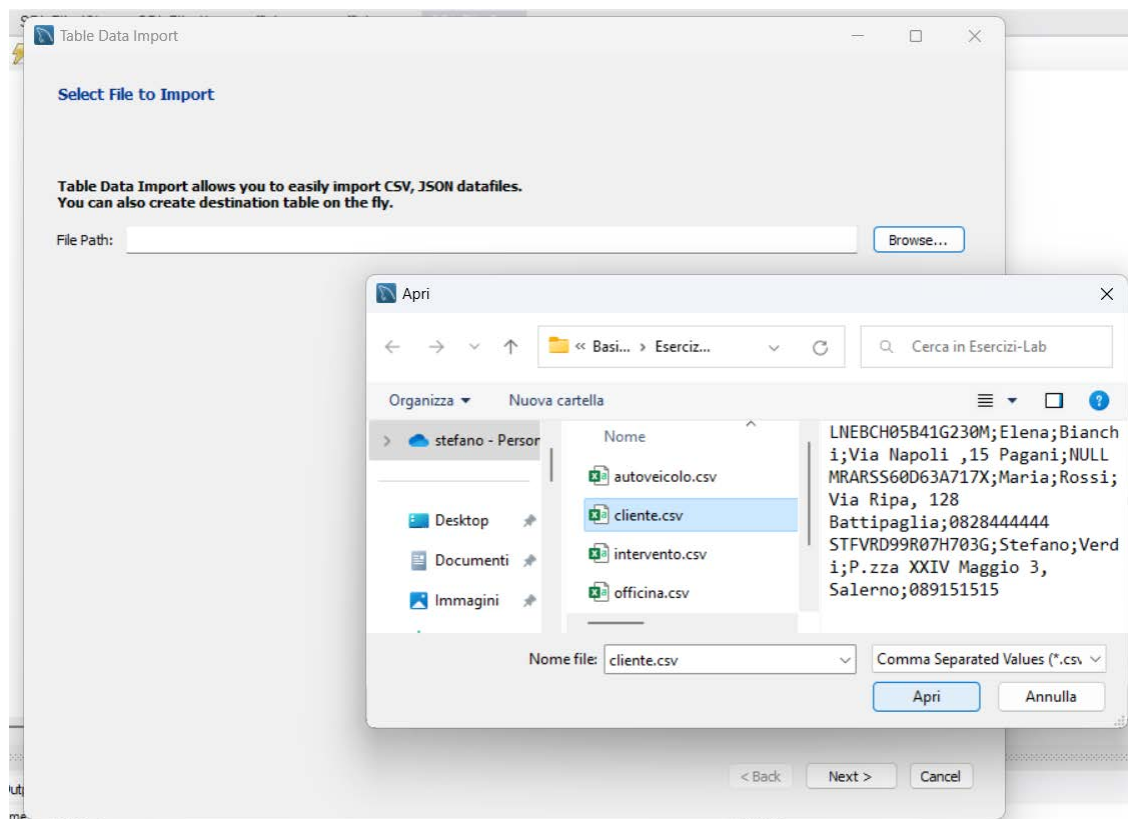
3. Tasto DX sulla tabella e selezionare:
'Table Data Import Wizard'
4. Seguire le istruzioni



Caricamento dei dati (Workbench)

► Seguire le istruzioni:

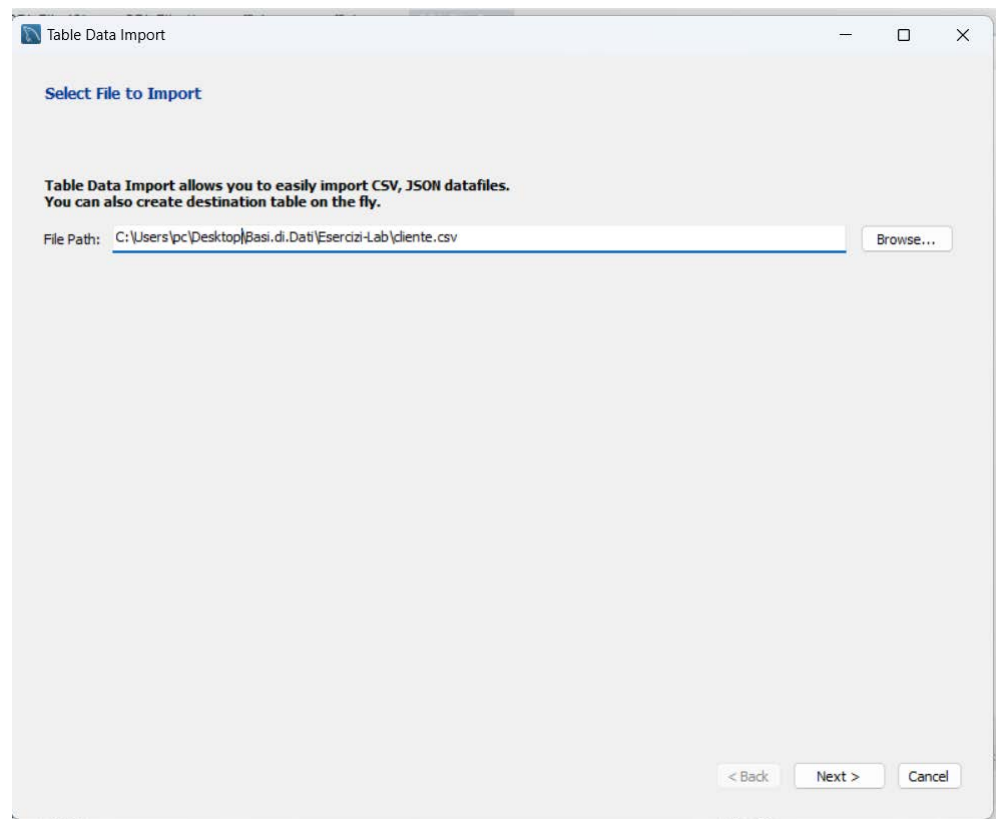
1. Scegliere il Path dove si trova il file da importare



Caricamento dei dati (Workbench)

► Seguire le istruzioni:

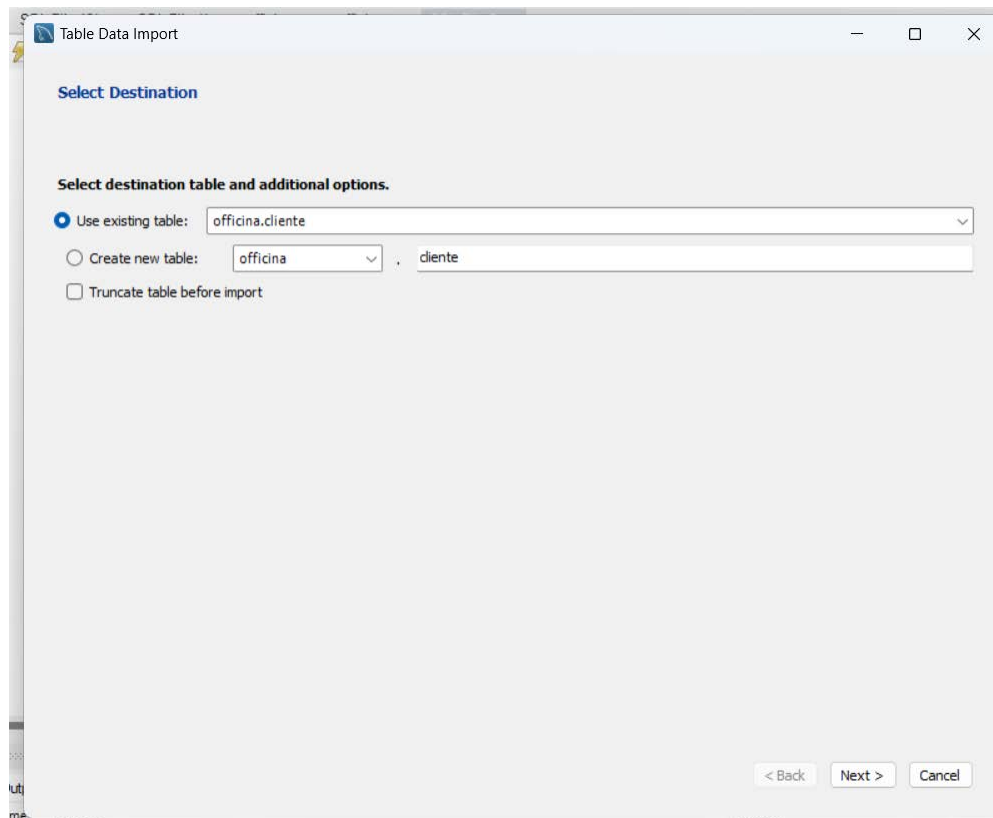
1. Scegliere il Path dove si trova il file da importare
2. Selezionare 'Next'



Caricamento dei dati (Workbench)

► Seguire le istruzioni:

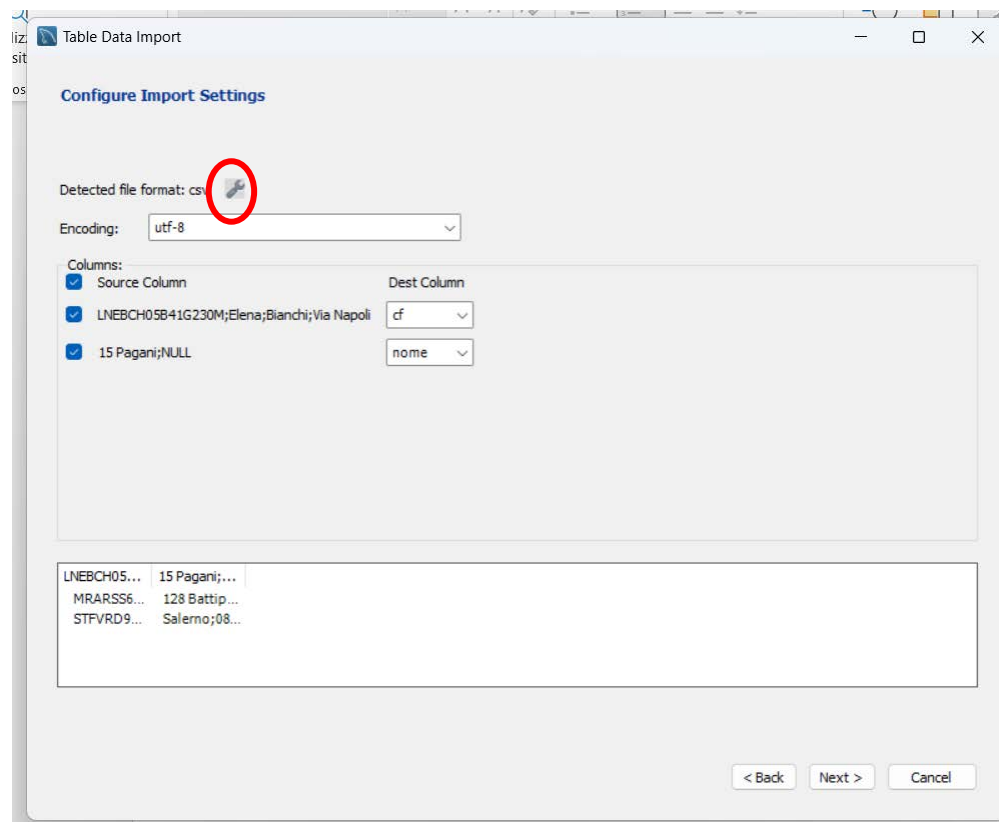
1. Scegliere il Path dove si trova il file da importare
2. Selezionare 'Next'
3. Controllare che sia selezionato **'Use existing table'** e che ci siano i dati inseriti
 1. In caso il campo sia vuoto, inserire <database_name>.<table>



Caricamento dei dati (Workbench)

► Seguire le istruzioni:

1. Scegliere il Path dove si trova il file da importare
2. Selezionare 'Next'
3. Controllare che sia selezionato **'Use existing table'** e che ci siano i dati inseriti
 1. In caso il campo sia vuoto, inserire `<database_name>.<table>`
4. Controllare il delimitatore di colonna
 1. Se il delimitatore non è corretto, cliccare sul tasto con l'icona della chiave inglese



Caricamento dei dati (Workbench)

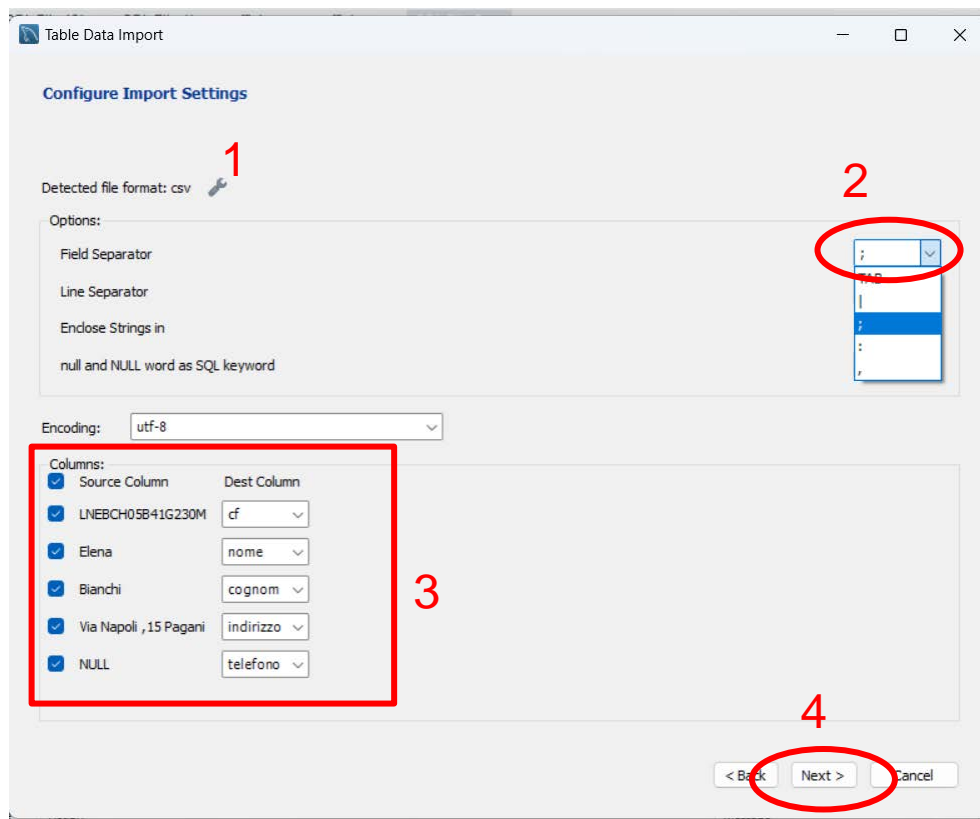
► Seguire le istruzioni:

1. Scegliere il Path dove si trova il file da importare
2. Selezionare 'Next'
3. Controllare che sia selezionato **'Use existing table'** e che ci siano i dati inseriti

1. In caso il campo sia vuoto, inserire `<database_name>.<table>`

4. Controllare il delimitatore di colonna

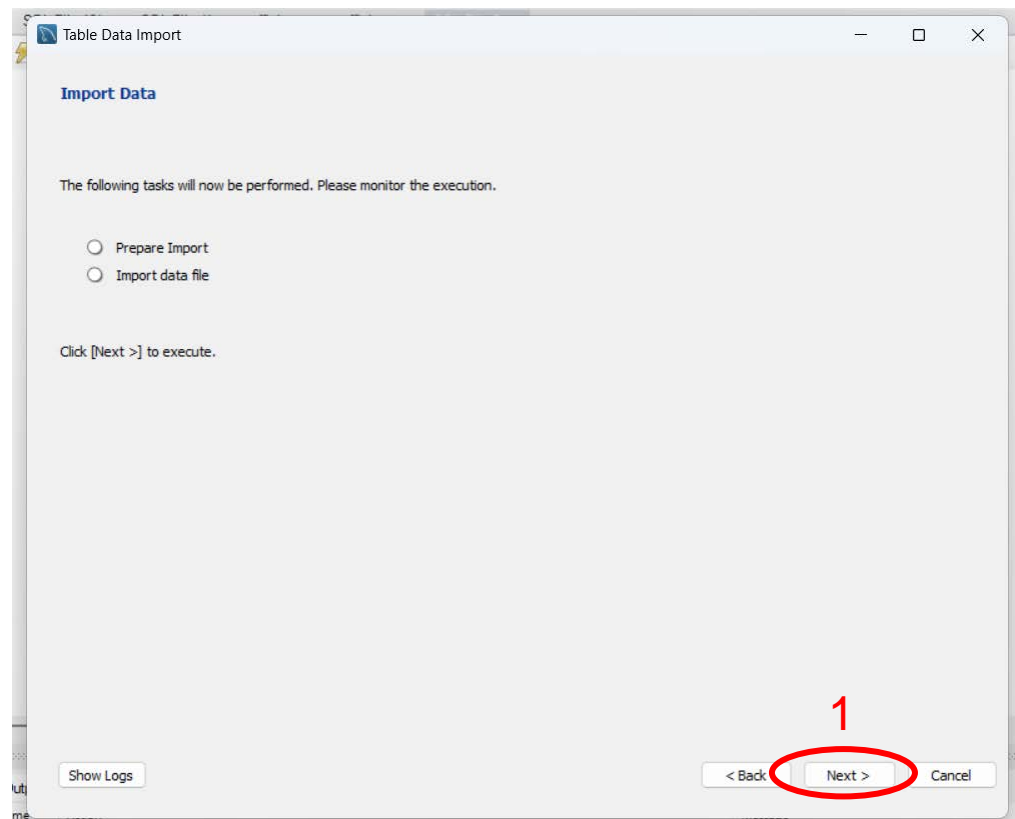
1. Se il delimitatore non è corretto, cliccare sul tasto con l'icona della chiave inglese
2. Selezionare il delimitatore corretto
3. Verificare che i valori corrispondano alle colonne della tabella
4. Cliccare il tasto **'Next'**



Caricamento dei dati (Workbench)

► Seguire le istruzioni:

1. Scegliere il Path dove si trova il file da importare
2. Selezionare 'Next'
3. Controllare che sia selezionato '**Use existing table**' e che ci siano i dati inseriti
 1. In caso il campo sia vuoto, inserire <database_name>.<table>
4. Controllare il delimitatore di colonna
 1. Se il delimitatore non è corretto, cliccare sul tasto con l'icona della chiave inglese
 2. Selezionare il delimitatore corretto
 3. Verificare che i valori corrispondano alle colonne della tabella
 4. Cliccare il tasto '**Next**'
5. Cliccare '**Next**'



Query di interrogazione: Algebra Relazionale

- ▶ La proiezione

PROJ lista_attributi(tabella)

- ▶ Esempio
 - ▶ Restituire il nome e il cognome dei clienti

Query di interrogazione: Algebra Relazionale

- ▶ La proiezione

PROJ lista_attributi(tabella)

- ▶ Esempio

- ▶ Restituire il nome e il cognome dei clienti

PROJ Nome, Cognome(Cliente)

Query di interrogazione: SQL

- ▶ Il comando select

```
SELECT lista_elementi_selezione  
FROM lista_riferimenti_tabella
```

- ▶ Restituire il nome e il cognome dei clienti

Query di interrogazione: SQL

- ▶ Il comando select

```
SELECT lista_elementi_selezione  
FROM lista_riferimenti_tabella
```

- ▶ Restituire il nome e il cognome dei clienti

```
SELECT nome, cognome  
FROM cliente
```


Query di interrogazione: SQL

- ▶ Il comando select

```
SELECT [ALL| DISTINCT] lista_elementi_selezione  
      FROM lista_riferimenti_tabella
```

- ▶ Restituire il cognome dei clienti

Query di interrogazione: SQL

- ▶ Il comando select

```
SELECT [ALL| DISTINCT] lista_elementi_selezione  
      FROM lista_riferimenti_tabella
```

- ▶ Restituire il cognome dei clienti

```
SELECT cognome  
FROM cliente
```

```
SELECT DISTINCT cognome  
FROM cliente
```

Query di interrogazione: SQL

- ▶ Il comando select

```
SELECT [ALL| DISTINCT] lista_elementi_selezione  
      FROM lista_riferimenti_tabella
```

- ▶ Restituire il cognome dei clienti

```
SELECT cognome  
FROM cliente
```

```
SELECT DISTINCT cognome  
FROM cliente
```

- ▶ NOTA: In Algebra Relazionale la proiezione elimina le tuple duplicate

Query di interrogazione: SQL

- ▶ Il comando select

```
SELECT [ALL| DISTINCT] lista_elementi_selezione  
      FROM lista_riferimenti_tabella
```

- ▶ Restituire tutti i dati delle officine

Query di interrogazione: SQL

- ▶ Il comando select

```
SELECT [ALL| DISTINCT] lista_elementi_selezione  
      FROM lista_riferimenti_tabella
```

- ▶ Restituire tutti i dati delle officine

```
SELECT *  
FROM officina
```

Query di interrogazione: Algebra Relazionale

- ▶ La selezione

`SEL lista_condizioni(tabella)`

- ▶ Esempio

- ▶ Restituire tutte le officine di Salerno

Query di interrogazione: Algebra Relazionale

► La selezione

SEL lista_condizioni(tabella)

► Esempio

- Restituire tutte le officine di Salerno

SEL città='Salerno' (Officina)

Query di interrogazione: SQL

► Il comando select

```
SELECT [ALL| DISTINCT] lista_elementi_selezione  
      FROM lista_riferimenti_tabella  
      [WHERE espressione_condizionale]
```

► Esempio

- Restituire tutte le officine di Salerno

Query di interrogazione: SQL

- ▶ Il comando select

```
SELECT [ALL| DISTINCT] lista_elementi_selezione  
      FROM lista_riferimenti_tabella  
      [WHERE espressione_condizionale]
```

- ▶ Esempio

- ▶ Restituire tutte le officine di Salerno

```
SELECT *  
FROM officina  
WHERE citta = 'Salerno'
```

Join Naturale: Algebra Relazionale

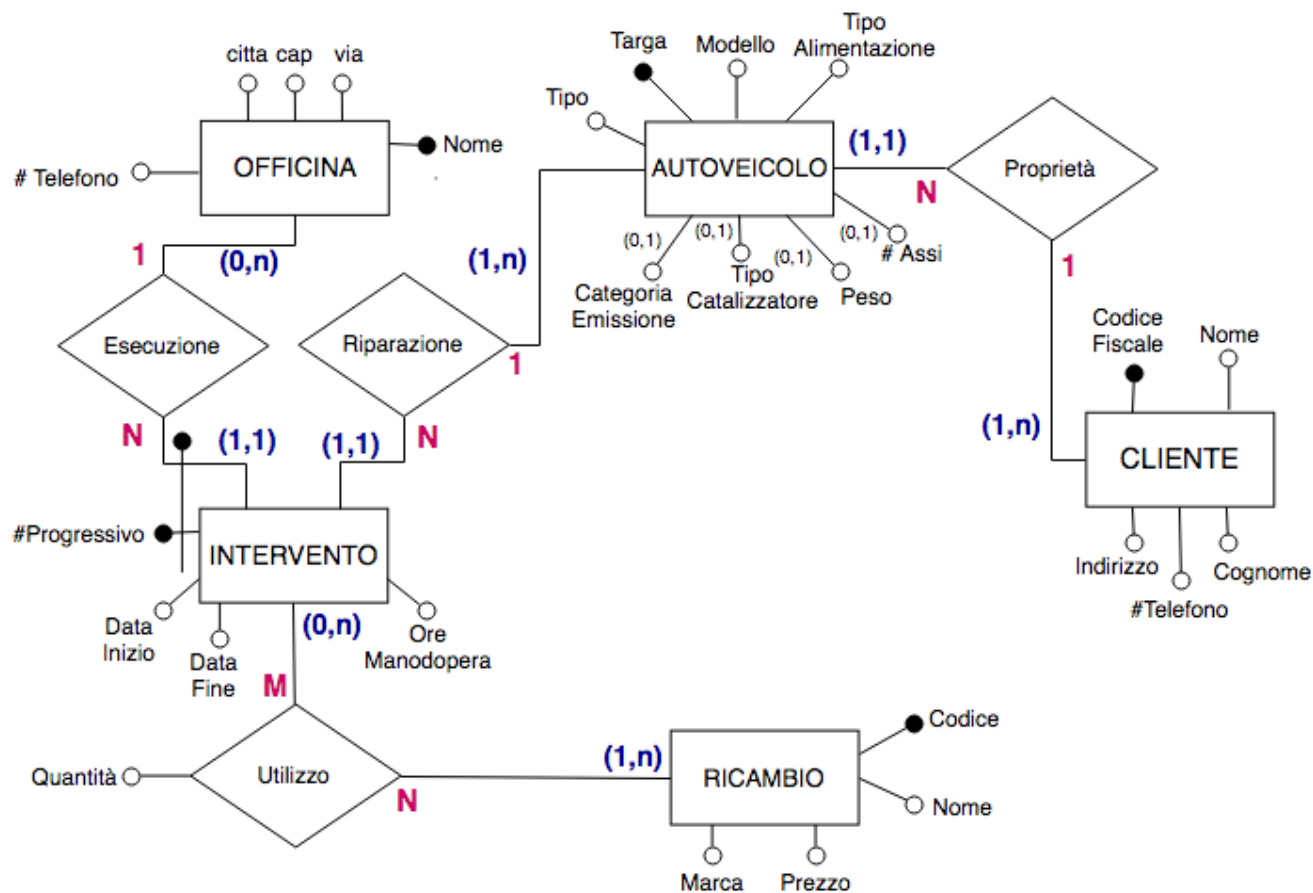
- ▶ Join naturale

`relazione_1 JOIN relazione_2`

- ▶ Esempio

- ▶ Restituire tutti i clienti (nome e cognome) proprietari di autoveicoli con categoria emissione “Euro 6”

Schema Ristrutturato



Join Naturale: Algebra Relazionale

- ▶ Join naturale

`relazione_1 JOIN relazione_2`

- ▶ Esempio

- ▶ Restituire tutti i clienti (nome e cognome) proprietari di autoveicoli con categoria emissione “Euro 6”

PROJ_{Nome,Cognome}(**Cliente**
JOIN
(**SEL**_{Categoria Emissione='Euro 6'}(**Autoveicolo**)))

Join Naturale: SQL

► Join naturale

```
SELECT [ALL| DISTINCT] lista_elementi_selezione  
      FROM tabella_1 NATURAL JOIN tabella_2  
      [WHERE espressione_condizionale]
```

► Esempio

- Restituire tutti i clienti (nome e cognome) proprietari di autoveicoli con categoria emissione “Euro 6”

Join Naturale: SQL

► Join naturale

```
SELECT [ALL| DISTINCT] lista_elementi_selezione  
      FROM tabella_1 NATURAL JOIN tabella_2  
      [WHERE espressione_condizionale]
```

► Esempio

- Restituire tutti i clienti (nome e cognome) proprietari di autoveicoli con categoria emissione “Euro 6”

```
SELECT nome, cognome  
FROM cliente NATURAL JOIN autoveicolo  
WHERE categoriaEmissione = 'Euro 6'
```

Equi-Join: Algebra Relazionale

- ▶ Equi-Join

`relazione_1 JOIN condizione join relazione_2`

- ▶ Esempio

- ▶ Restituire tutte le auto (targa, modello) su cui sono stati effettuati interventi a partire dal 03/11/2018

Equi-Join: Algebra Relazionale

► Equi-Join

`relazione_1 JOIN condizione join relazione_2`

► Esempio

- Restituire tutte le auto (targa, modello) su cui sono stati effettuati interventi a partire dal 03/11/2018

```
PROJTarga,Modello(Autoveicolo  
  JOINTarga = Targa Autoveicolo  
(SELData Inizio >= '2018-11-03' (Intervento)))
```


Equi-Join: SQL

► Equi-Join

```
SELECT [ALL| DISTINCT] lista_elementi_selezione  
      FROM tabella_1 INNER JOIN tabella_2  
      ON condizione_join  
      [WHERE espressione_condizionale]
```

► Esempio

- Restituire tutte le auto (targa, modello) su cui sono stati effettuati interventi a partire dal 03/11/2018

Equi-Join: SQL

► Equi-Join

```
SELECT [ALL| DISTINCT] lista_elementi_selezione
      FROM tabella_1 INNER JOIN tabella_2
            ON condizione_join
      [WHERE espressione_condizionale]
```

► Esempio

- Restituire tutte le auto (targa, modello) su cui sono stati effettuati interventi a partire dal 03/11/2018

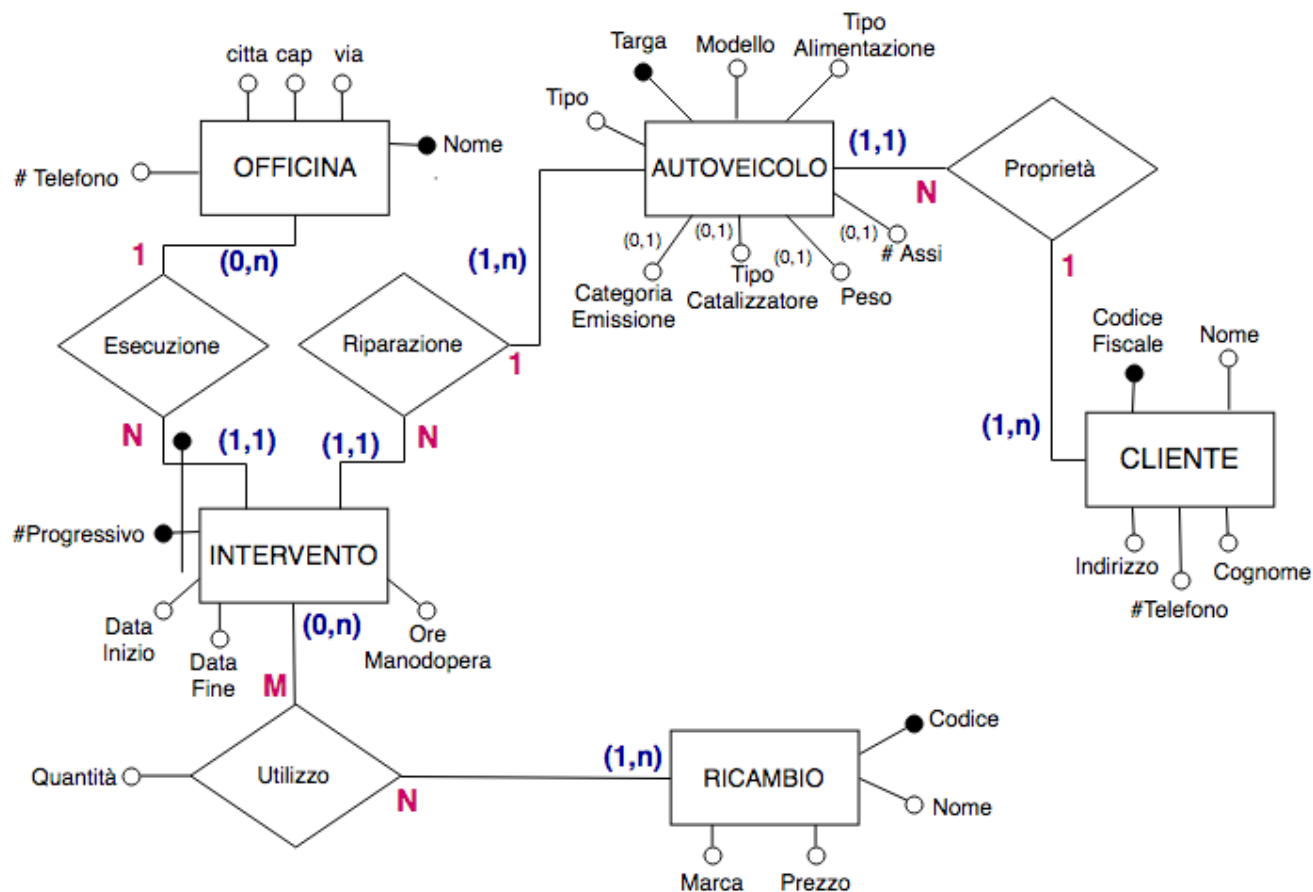
```
SELECT targa, modello
FROM autoveicolo INNER JOIN intervento ON
      autoveicolo.targa = intervento.targaAutoveicolo
WHERE dataInizio >= '2018-11-03'
```

Funzioni di Aggregazione: SQL

► Esempio

- Per ogni ricambio restituire il numero di interventi in cui sono stati impiegati (nome, numInterventi)

Schema Ristrutturato



Funzioni di Aggregazione: SQL

► Esempio

- Per ogni ricambio restituire il numero di interventi in cui sono stati impiegati (nome, numInterventi)

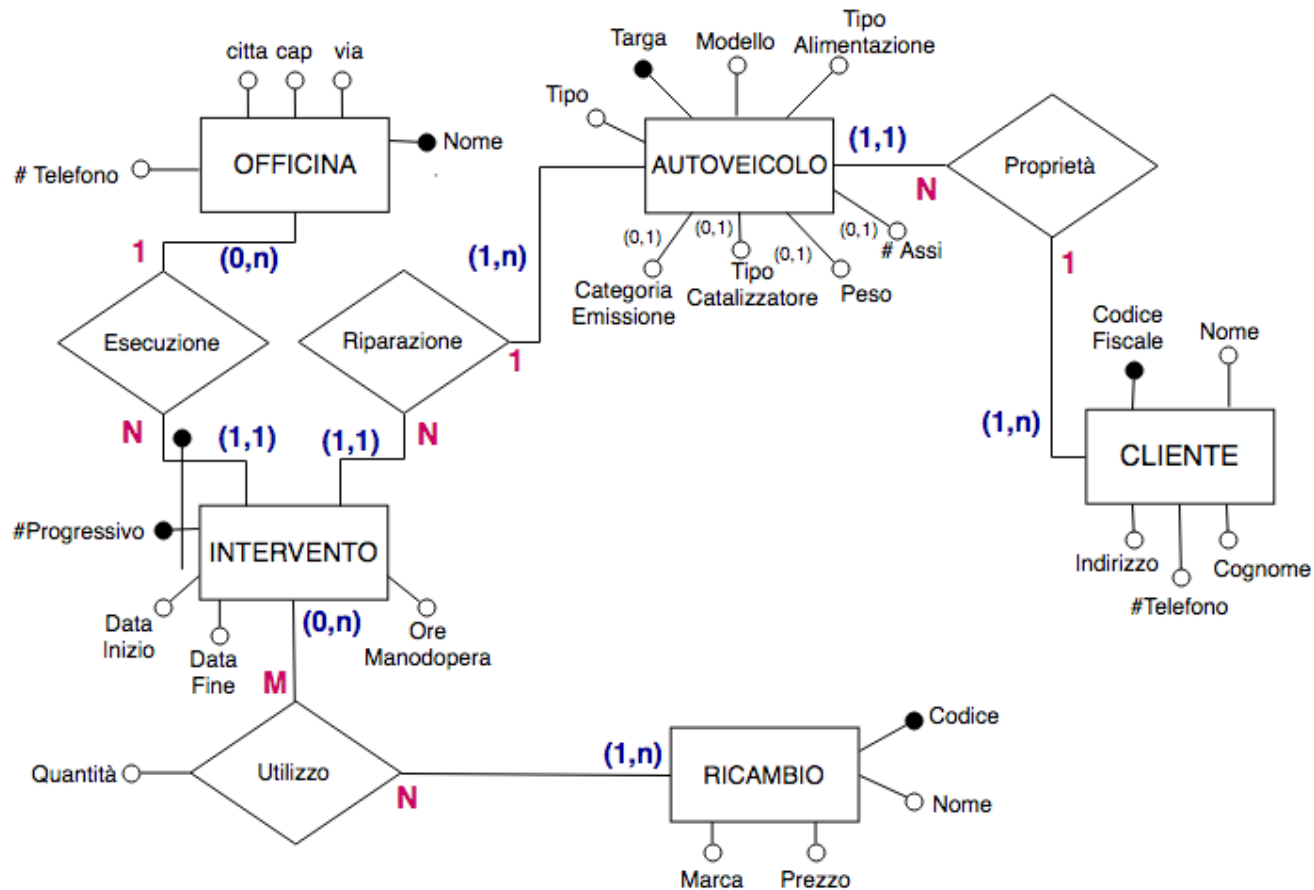
```
SELECT nome, COUNT(codiceRicambio) AS  
    numInterventi  
FROM ricambio INNER JOIN utilizzo ON  
    ricambio.codice = utilizzo.codiceRicambio  
GROUP BY codiceRicambio,nome
```

Funzioni di Aggregazione: SQL

► Esempio

- Per ogni ricambio restituire il numero di interventi in cui sono stati impiegati e la quantità totale richiesta (nome, numInterventi, quantitaTotale)

Schema Ristrutturato



Funzioni di Aggregazione: SQL

► Esempio

- Per ogni ricambio restituire il numero di interventi in cui sono stati impiegati e la quantità totale richiesta (nome, numInterventi, quantitaTotale)

```
SELECT nome, COUNT(codiceRicambio) AS  
    numInterventi, SUM(quantità) AS quantitaTotale  
FROM ricambio INNER JOIN utilizzo ON  
    ricambio.codice = utilizzo.codiceRicambio  
GROUP BY codiceRicambio,nome
```


Funzioni di Aggregazione: SQL

► Esempio

- Per ogni ricambio restituire il numero di interventi in cui sono stati impiegati, con una quantità maggiore di uno, e la quantità totale richiesta (nome, numInterventi, quantitaTotale)

Funzioni di Aggregazione: SQL

► Esempio

- Per ogni ricambio restituire il numero di interventi in cui sono stati impiegati, con una quantità maggiore di uno, e la quantità totale richiesta (nome, numInterventi, quantitaTotale)

```
SELECT nome, COUNT(codiceRicambio) AS  
    numInterventi, SUM(quantità) AS quantitaTotale  
FROM ricambio INNER JOIN utilizzo ON  
    ricambio.codice = utilizzo.codiceRicambio  
WHERE quantità > 1  
GROUP BY codiceRicambio,nome
```

Funzioni di Aggregazione: SQL

► Esempio

- Per ogni ricambio restituire la quantità totale richiesta se e solo se sono stati impiegati in più di un intervento (nome, quantitaTotale)

Funzioni di Aggregazione: SQL

► Esempio

- Per ogni ricambio restituire la quantità totale richiesta se e solo se sono stati impiegati in più di un intervento (nome, quantitaTotale)

```
SELECT nome, SUM(quantità) AS quantitaTotale
FROM ricambio INNER JOIN utilizzo ON
    ricambio.codice = utilizzo.codiceRicambio
GROUP BY codiceRicambio,nome
HAVING COUNT(codiceRicambio) > 1
```

Query Annidate: SQL

► Esempio

- Restituire i dati dei clienti (nome, cognome) che hanno richiesto almeno un intervento

Query Annidate: SQL

► Esempio

- Restituire i dati dei clienti (nome, cognome) che hanno richiesto almeno un intervento

```
SELECT DISTINCT nome, cognome  
FROM cliente NATURAL JOIN autoveicolo INNER JOIN  
intervento ON autoveicolo.targa =  
intervento.targaAutoveicolo
```

Query Annidate: SQL

► Esempio

- Restituire i dati dei clienti (nome, cognome) che hanno richiesto almeno un intervento

```
SELECT nome, cognome  
FROM cliente AS c  
WHERE c.codiceFiscale IN (SELECT a.codiceFiscale  
    FROM autoveicolo AS a INNER JOIN  
    intervento AS i  
    ON a.targa = i.targaAutoveicolo)
```

Query Annidate: SQL

► Esempio

- Restituire i dati dei clienti (nome, cognome) che hanno richiesto almeno un intervento

```
SELECT nome, cognome
FROM cliente AS c
WHERE EXISTS (SELECT *
                FROM autoveicolo AS a INNER JOIN
                    intervento AS i
                ON a.targa = i.targaAutoveicolo
                WHERE a.codiceFiscale=c.codiceFiscale)
```


Query Annidate: SQL

► Esempio

- Restituire i dati dell'officina (nome) con il massimo numero di ore di manodopera impiegate in un qualche intervento

```
SELECT nome
FROM officina AS o INNER JOIN intervento AS i
    ON o.nome = i.nomeOfficina
WHERE oreManodopera >= (
    SELECT MAX(oreManodopera)
    FROM intervento)
```

Conteggio tuple Algebra Relazionale

- ▶ Selezionare i clienti (nome, cognome) che abbiano autoveicoli tutti dello stesso tipo

Conteggio tuple Algebra Relazionale

- Selezionare i clienti (nome, cognome) che abbiano autoveicoli tutti dello stesso tipo

PROJ_{nome, cognome}(**Cliente**)

-

(**PROJ**_{nome, cognome} (**Cliente** **JOIN**

(**SEL**_{Tipo<>Tipo1} (**Autoveicolo**

JOIN

(**REN**_{tipo1←tipo}(**PROJ**_{codiceFiscale, tipo}(**Autoveicolo**))))))

Query in SQL

- ▶ Selezionare i clienti (nome, cognome) che abbiano autoveicoli tutti dello stesso tipo

```
SELECT nome, cognome  
FROM cliente NATURAL JOIN autoveicolo  
GROUP BY codiceFiscale, nome, cognome  
HAVING COUNT(DISTINCT tipo) = 1
```