

Progetto
Corso di Programmazione per il Web
A.A. 2022/2023
Docenti: Marco Mesiti, Sara Bonfitto

Titolo esercitazione:
City Delivery

Gruppo composto da:

954587 Ivan F. Muñoz G. *ivanfrancisco.munozguiario@studenti.unimi.it* [referente]

950788 Alessio Marrazzo *alessio.marrazzo@studenti.unimi.it*

Data consegna: 25/10/2023

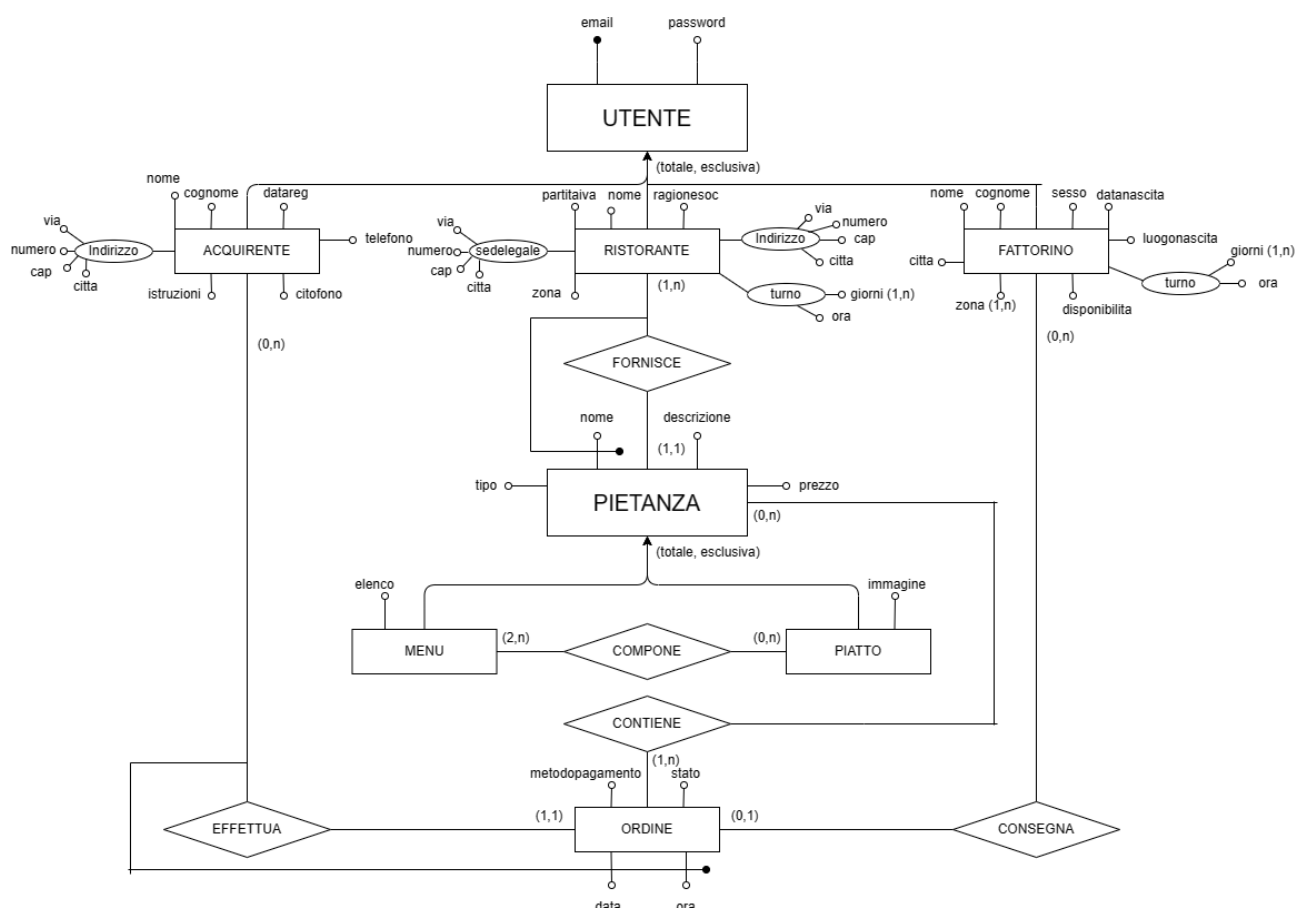
Il progetto è già stato consegnato in precedenza? No

Se sì, indica tutte le date degli appelli in cui il progetto è stato consegnato:

Se sì, indica brevemente (due righe al massimo) le modifiche apportate rispetto all'ultimo progetto consegnato:

1. Progettazione concettuale

1.1 Schema ER



1.2 Vincoli di dominio e assunzioni fatte

In questa sezione vanno riportati e numerati tutti i vincoli che devono essere garantiti nella base di dati e che non si possono direttamente desumere dallo schema ER. **Si noti che non è necessario riportare i vincoli di chiave e/o di cardinalità.** Questi si desumono direttamente dallo schema ER. La numerazione dei vincoli permette una più facile identificazione nelle fasi successive della progettazione ed implementazione.

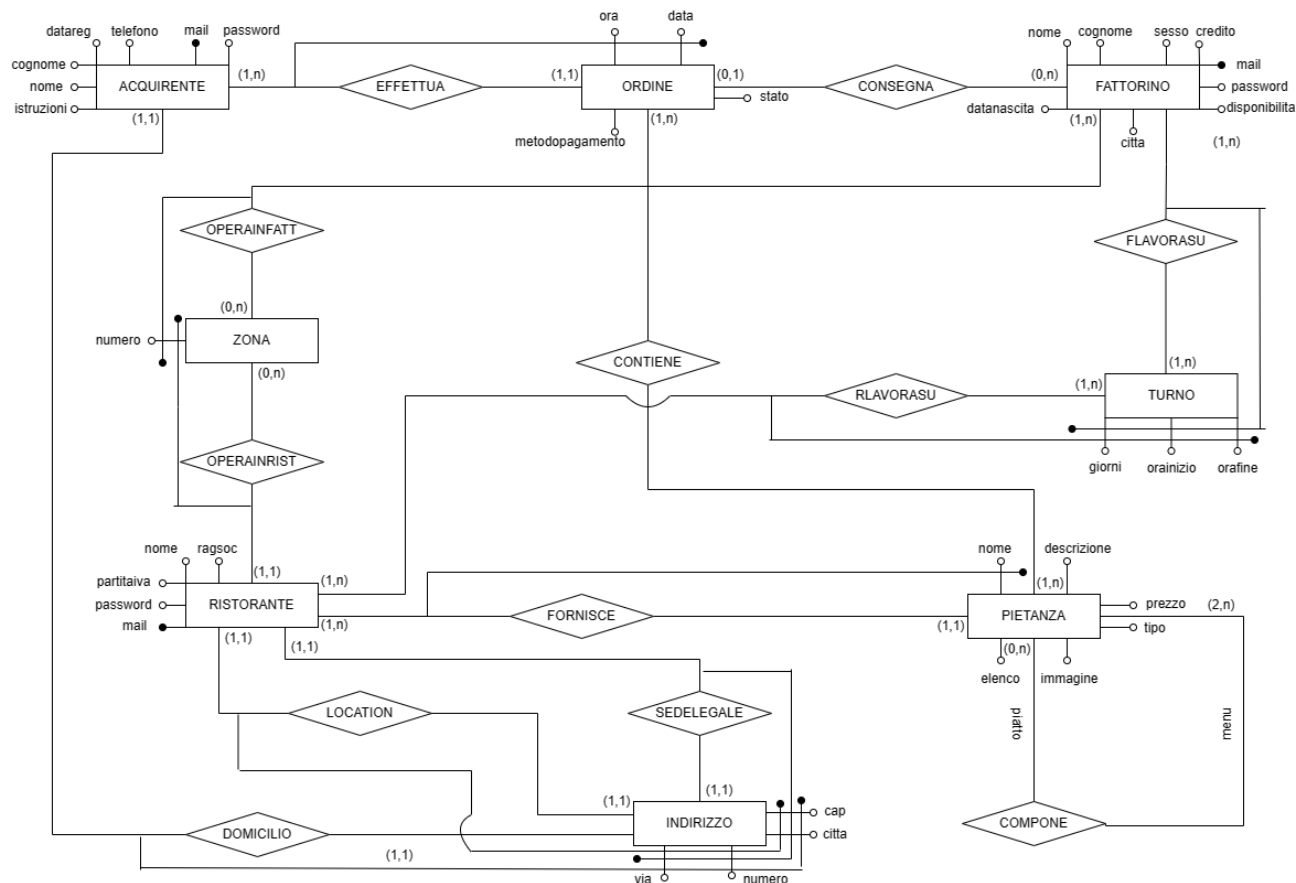
Esempio:

| Numero vincolo | Entità/Relazione coinvolta/e | Descrizione |
|----------------|-----------------------------------|---|
| V1 | Fattorino, Ristorante, Acquirente | L'attributo "città" di queste entità può assumere solo i seguenti valori: "Milano", "Torino", "Trento", "Roma", "Palermo" e "Cagliari". |
| V2 | Ristorante, Fattorino | L'attributo "zona" assume solo i valori 1, 2, 3, 4 e 5 indipendentemente dalla città |
| V3 | Fattorino | "sesso" ammette anche "NB" (Not Binary) oltre a "M" e "F" |
| V5 | Ordine | "stato" ammette "in preparazione", "in consegna" |

| | | |
|----|----------|---|
| | | e “consegnato”. |
| V6 | Ordine | “metodopagamento” può essere solo “contanti” o “carta”. |
| V7 | Pietanza | “tipo” assume valori “menu” o “piatto” |

| Numero assunzione | Entità/Relazione coinvolta/e | Descrizione |
|-------------------|------------------------------|---|
| A1 | Fattorino | Opera in più zone di una sola città |
| A2 | Fattorino | Ogni giorno può fare uno e un solo turno secondo gli orari specificati tra i vincoli |
| A3 | Fattorino | Il valore di credito si aggiorna automaticamente con il 10% del valore dell'ordine consegnato |

1.3 Schema ER ristrutturato



| Numero vincolo | Entità/Relazione coinvolta/e | Descrizione |
|----------------|-----------------------------------|--|
| V8 | Acquirente, Ristorante, Fattorino | Registrandosi all'applicazione, si può operare al più in veste di una sola di queste 3 entità. |
| V9 | Pietanza | Quando l'attributo “tipo” vale |

| | | |
|-----|-----------------------|---|
| | | "menu", "immagine" è NULL. Quando "tipo" vale "piatto", "elenco" vale NULL. |
| V10 | Fattorino, Ristorante | Queste due entità hanno turni che non coincidono |
| V11 | Pietanza, Compone | Una pietanza di tipo "menu" è composta da almeno 2 pietanze di tipo "piatto". |

| Numero assunzione | Entità/Relazione coinvolta/e | Descrizione |
|-------------------|------------------------------|--|
| A4 | Ristorante | Non sono ammesse catene di ristoranti |

2. Progettazione logica e comandi SQL

Acquirente(mail, password, nome, cognome, datereg, telefono, istruzioni)

Fattorino(mail, password, nome, cognome, sesso, datanascita, luogonascita, citta, disponibilita, credito)

Indirizzo(via, numero, cap, citta)

Domicilio(mail^{Acquirente}, via^{Indirizzo}, numero^{Indirizzo}, cap^{Indirizzo}, citta^{Indirizzo})

Consegna(mail^{Fattorino}, ora^{Ordine}, data^{Ordine})

Ordine(data, ora, stato, metodopagamento, mail^{Acquirente})

Zona(numero)

Operainfatt(mail^{Fattorino}, numero^{Zona})

Turno(giorni, orainizio, orafine)

Flavorasu(mail^{Fattorino}, giorni^{Turno}, orario^{Turno})

Ristorante(mail, password, partitaiva_U, nome, ragsoc, numero^{Zona})

Rlavorasu(mail^{Ristorante}, giorni^{Turno}, orainizio^{Turno}, orafine^{Turno})

Location(mail^{Ristorante}, via^{Indirizzo}, numero^{Indirizzo}, cap^{Indirizzo}, citta^{Indirizzo})

Sedelegale(mail^{Ristorante}, via^{Indirizzo}, numero^{Indirizzo}, cap^{Indirizzo}, citta^{Indirizzo})

Pietanza(nome, mail^{Ristorante}, prezzo, descrizione, tipo, elenco, immagine)

Compone(piatto^{Pietanza}, menu^{Pietanza})

Contiene(nome^{Pietanza}, mail^{Pietanza}, ora^{Ordine}, data^{Ordine})

1) Trova i nomi e cognomi degli acquirenti il cui ordine è stato consegnato dopo le ore 20, riportando anche l'orario in cui il momento della consegna è avvenuto.

```
SELECT acquirente.nome, acquirente.cognome, ordine.ora
FROM acquirente INNER JOIN ordine ON acquirente.mail=ordine.mail
WHERE ordine.ora > "20:00:00" AND ordine.stato="consegnato";
```

Con l'INNER JOIN vengono affiancate tutte le tuple delle tabelle Acquirente e Ordine che hanno una corrispondenza con la mail dell'Acquirente. Infine, tali tuple vengono filtrate in base all'ora e allo stato della consegna.

2) Contare il numero di fattorini disponibili per zona a Milano.

```

SELECT zona, COUNT(DISTINCT mailfatt) AS numero_fattorini
FROM operainfatt
WHERE mailfatt IN (
    SELECT mail
    FROM fattorino
    WHERE citta="Milano");

GROUP BY zona

```

Con la nested query si cercano i fattorini che operano a Milano.
 La main query seleziona le zone in cui operano i fattorini. COUNT conta quanti fattorini lavorano nelle 5 zone di Milano.

3) Trova i nomi dei ristoranti con sede legale a Roma che offrono menù composti da un piatto di pasta.

```

SELECT ristorante.nome
FROM ristorante INNER JOIN sedelegale ON sedelegale.ristorante=ristorante.mail
WHERE sedelegale.citta="Roma" AND ristorante.mail IN (
    SELECT mail
    FROM pietanza
    WHERE tipo="menu" AND elenco LIKE "%pasta%");

```

Per questa interrogazione ci si avvale di una sub-query:

- a) La nested query trova le mail dei ristoranti che offrono un piatto di pasta nel menu
- b) La main query si occupa di trovare i nomi dei ristoranti con sede legale a Roma. A tal fine, viene eseguito un INNER JOIN tra Ristorante e Sedelegale per identificare le tuple delle due tabelle che abbiano in comune la mail del ristorante. Si procede al filtraggio delle tuple, tali che abbiano sede legale a Roma e le cui mail appaiano nel risultato della sub-query.

4) Riporta i nomi dei fattorini che hanno effettuato una consegna

```

SELECT fattorino.nome, fattorino.cognome
FROM fattorino INNER JOIN ordine ON fattorino.mail=ordine.mailfatt
WHERE ordine.stato="consegnato"

```

5) Calcola la media dei prezzi delle pietanze.

```

SELECT AVG(pietanza.prezzo) AS media
FROM pietanza;

```

L'AVG() calcola la media di tutti i cibi presenti nel database.