Progetto Basi di Dati
Gestione di una Palestra durante il Covid-19

Bit Nicola (878249), Campanelli Alessio (878170), Gottardo Mario (879088)

AA. 2020/2021

Indice

1 Introduzione

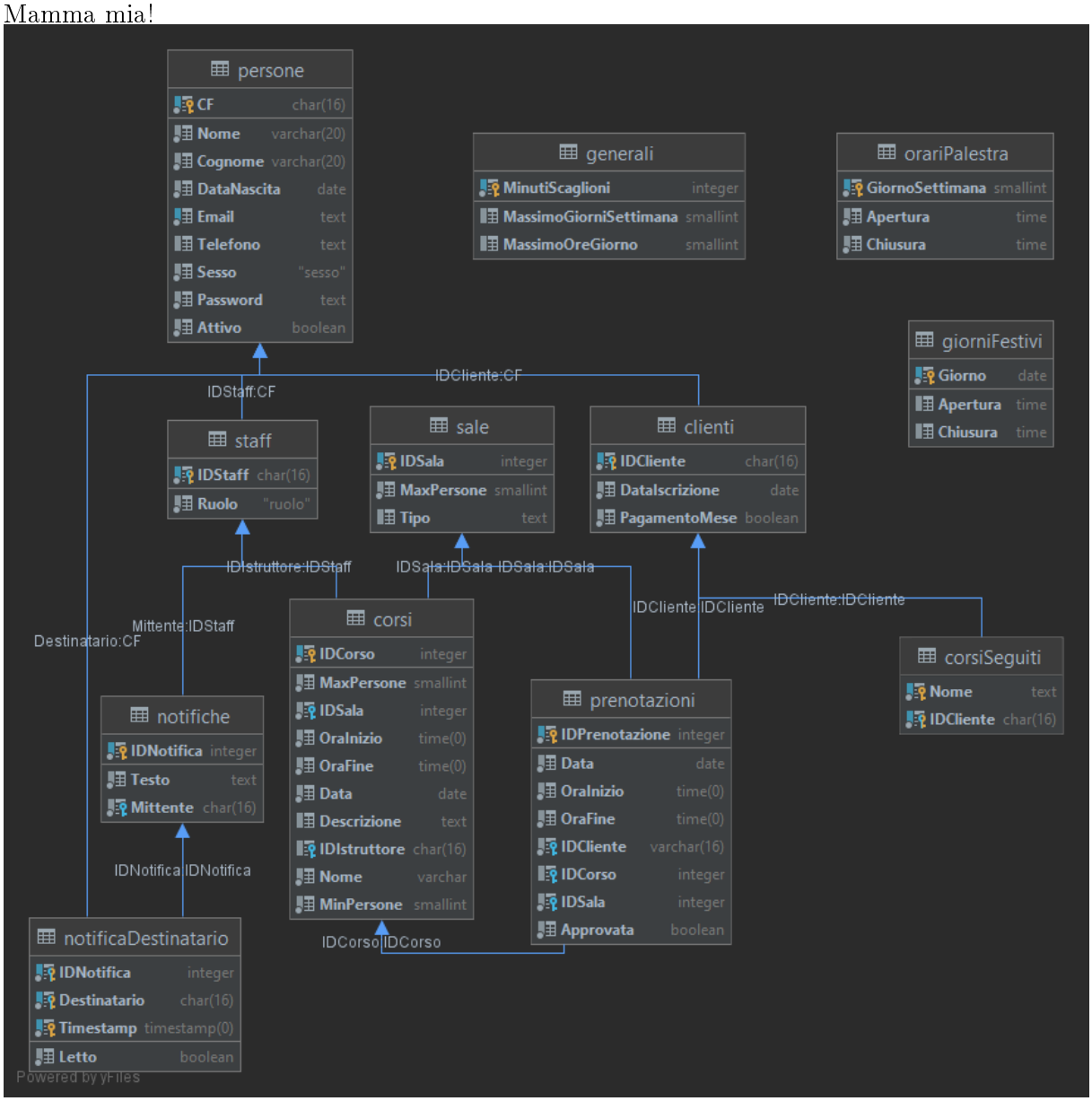
Introduzione da scrivere

1.1 Credenziali di accesso alla rete Hamachi

```
-Nome rete: progetto-DB
-Password: w^JT4fg
```

2 Database

Mamma mia!



Mamma mia!

3 Query principali

3.1 Contact Tracing

L'algoritmo di contact tracing prende in ingresso un istanza di Persona che sappiamo essere positiva (da qui in poi "positivo") e un numero di giorni da analizzare (al più 7 per evitare calcoli troppo pesanti), al termine dell'esecuzione, l'algoritmo ritorna una lista di potenziali positivi. L'algoritmo cerca l'ultima data in cui il positivo è entrato nella struttura, determina allora il range usando il numero di giorni in input.
Trova tutte le volte in cui il positivo è entrato, per ogni ingresso trova tutte le prenotazioni che intersecano quella in esame e salva il cliente e l'eventuale istruttore nel caso la prenotazione sia riferita ad un corso.

3.1.1 pseudocodice

```
def contact_tracing(zero, days):
    potential_infected = [zero.CF]
    days = int(days)
    if days > 7:
        days = 7

    last_zero_appearance_date = ultimo ingresso di positivo nella struttura

    if last_zero_appearance_date is not None:
        last_zero_appearance_date = last_zero_appearance_date.Data
    else:
        return [] # Se non e' mai entrato non ci saranno contatti da tracciare

    lower_limit_date = last_zero_appearance_date - timedelta(days=days)
    # definiamo il range usando il numero di giorni in input

    last_zero_appearances = tutti gli ingressi di positivo nell'intervallo

    for appearance in last_zero_appearances:
        prenotazioni = tutte le prenotazioni che intersecano quella in esame

        if appearance.IDCorso is not None:
            istruttore = istruttore del corso
            potential_infected.append(istruttore)

        for p in prenotazioni:
            potential_infected.append(p.IDCliente)
    return [get_persona_by_cf(cf) for cf in (list(set(potential_infected)))]
    # Ritorniamo l'istanza delle persone il cui codice fiscale risulta tra
    # i potenziali positivi, rimuovendo prima i duplicati
```

3.1.2 Query di contact tracing

Query per trovare l'ultimo ingresso del paziente zero

SELECT *

```
FROM prenotazioni p
WHERE p.IDCliente = zero.CF AND p.Data <= CURRENT_DATE AND
  p.Approvata = TRUE
ORDER BY p.DATA DESC
LIMIT 1
```

Query per trovare gli ingressi da analizzare del paziente zero

```
SELECT *
FROM prenotazioni p
WHERE p.IDCliente = zero.CF AND p.Approvata = TRUE AND
          p.Data BETWEEN lower_limit_date AND CURRENT_DATE
ORDER BY p.DATA DESC
```

Query per trovare gli ingressi da analizzare del paziente zero

```
SELECT *
FROM prenotazioni p
WHERE p.IDCliente = zero.CF AND p.Approvata = TRUE AND
          p.Data BETWEEN lower_limit_date AND CURRENT_DATE
ORDER BY p.DATA DESC
```

Query per trovare gli ingressi che si sovrappongono a quelli del paziente zero

```
SELECT *
FROM prenotazioni p
WHERE p.Data  = appearance.Data AND p.IDSala = appearance.IDSala AND
  (p.OraFine >= appearance.OraInizio OR p.OraFine <= appearance.OraFine)
 AND p.Approvata = TRUE
```

4 Scelte progettuali