



# Laboratorio n°7 del 28-11-2017

Ing. Dario Cogliati



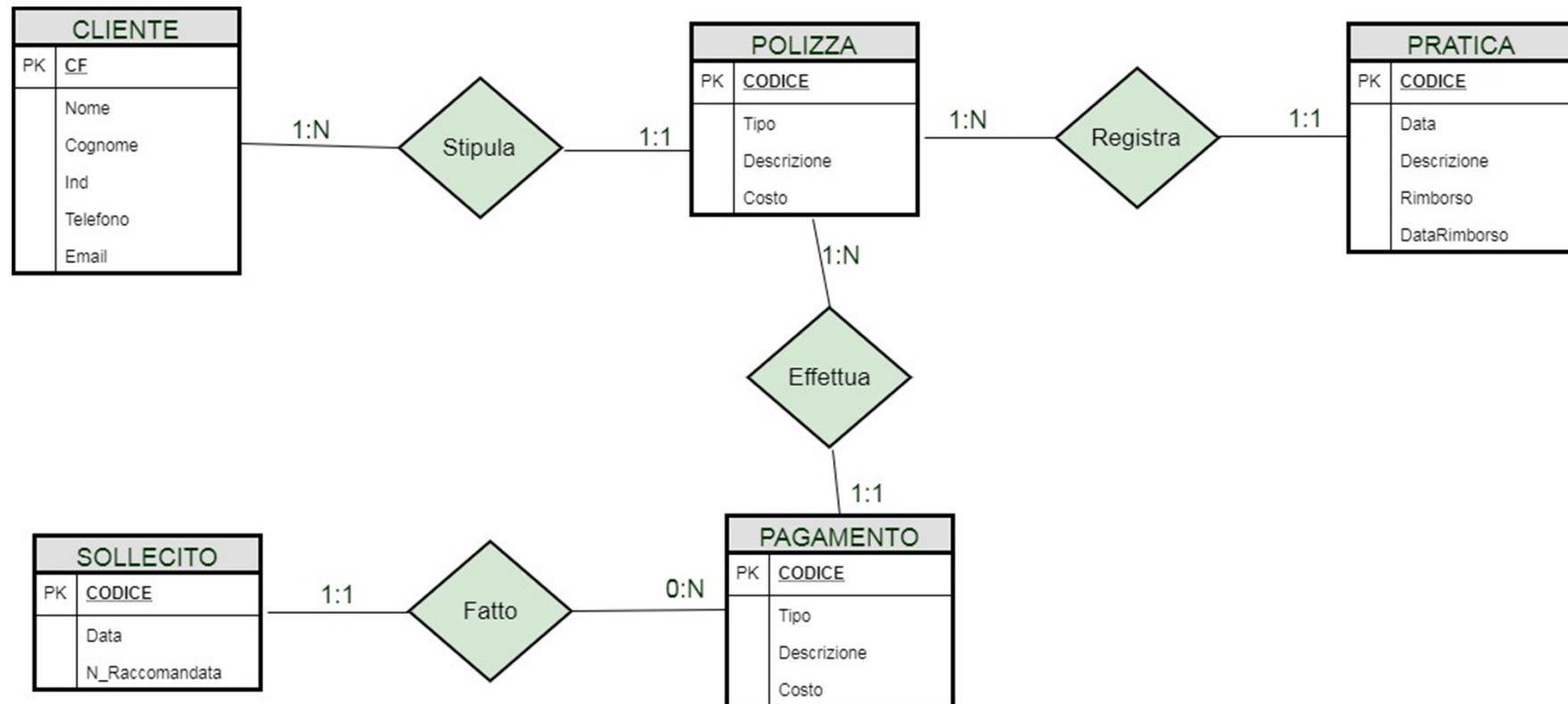
## Es1 – DB Assicurazioni

Progettare il sistema informativo di una assicurazione. L'assicurazione ha clienti con codice fiscale, cognome, nome, indirizzo, un numeri di telefono e un indirizzo e-mail. Ogni cliente ha una o più polizze. Ogni polizza ha un codice, un tipo, una descrizione, un costo annuale. Ad ogni polizza è associato un elenco dei pagamenti effettuati (con importo e data) e degli eventuali solleciti per ciascun pagamento. Per ogni polizza ci possono essere degli incidenti, con una pratica per incidente con un codice e le seguenti informazioni: data dell'incidente, descrizione. Ad ogni incidente ci possono essere dei rimborsi effettuati (con importo e data).

1. Progettare lo schema Entità-relazione per il sistema
2. Progettare lo schema relazionale



# Es1 - SOLUZIONE



CLIENTE(CF,Nome,Cognome,Ind,Telefono,Email)

POLIZZA(CODICE,Tipo,Descrizione,Costo,CF)

PAGAMENTO(CODICE,Tipo,Descrizione,Costo, ,codice\_polizza)

SOLLECITO(CODICE,Data,N\_raccomandata, ,codice\_pagamento)

PRATICA(CODICE,Data,Descrizione,Rimborso,DataRimborso,codice\_polizza)

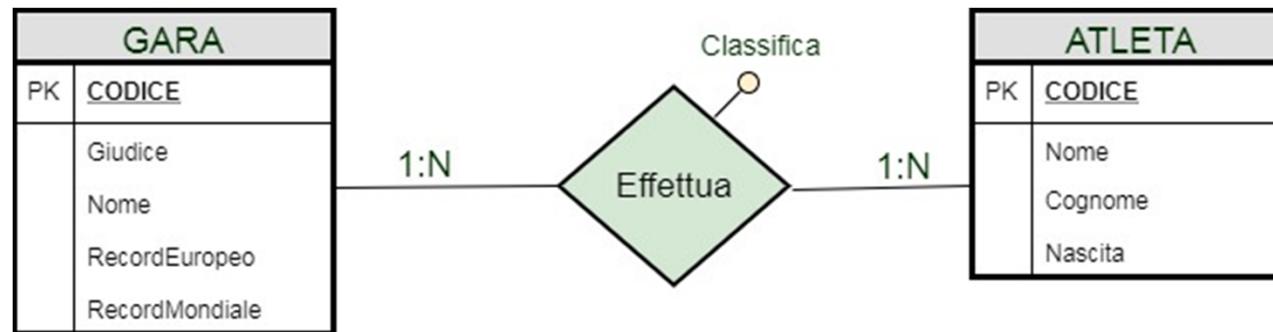


## Es2 – DB Atletica

Si vuole progettare un sistema informativo per le iscrizioni ad un meeting di atletica leggera. Il programma del meeting prevede un certo numero di gare. Per ogni gara il sistema deve memorizzare un codice identificativo, un giudice di gara, un nome (es. 100 metri, 200 metri), il record europeo ed il record mondiale. Gli atleti (nome, cognome e data di nascita) possono iscriversi ad una o più gare del meeting. Ogni atleta che si iscrive ad una gara deve specificare se vuole partecipare in «classifica» o «fuori classifica». Progettare la base di dati a livello concettuale e a livello logico evidenziando le chiavi primarie.



## Es2 - SOLUZIONE



GARA(**CODICE**,Giudice,Nome,RecordEuropeo,RecordMondiale)

EFFETTUA( **COD GARA, COD ATLETA**, Classifica)

ATLETA(**CODICE**,Nome,Cognome,Nascita)



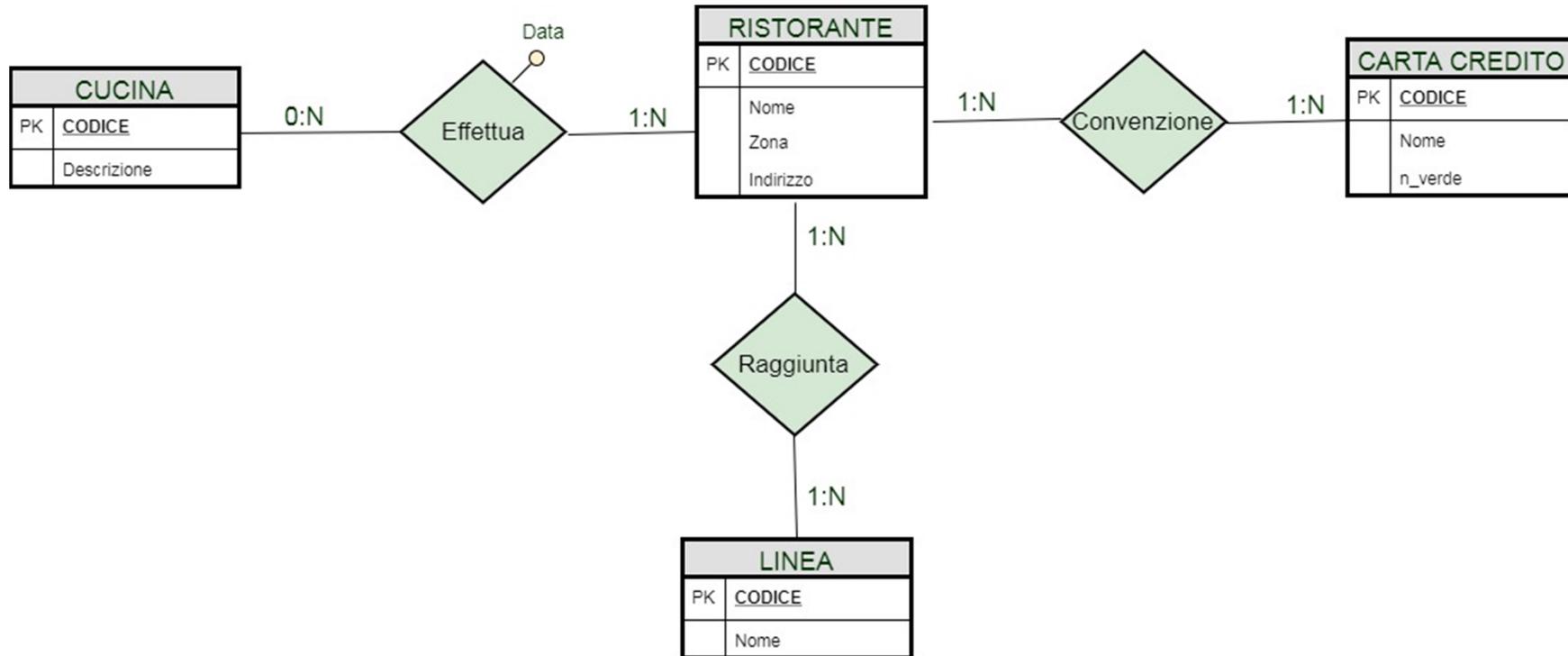
## Es3 – DB Ristorante

***Produrre uno schema Entità-Relazione per la seguente realtà di interesse (dichiara eventuali ipotesi aggiuntive).***

Diversi ristoranti adottano diversi tipi di cucina (cinese, thailandese, italiana, ...). Ad ogni tipo di cucina si vuole associare un codice ed una breve descrizione ad uso dei turisti. I ristoranti sono siti in zone diverse della città, ognuna delle quali è raggiunta da almeno una linea urbana di autobus. I ristoranti, di cui si fornisce anche nome e indirizzo, accettano esclusivamente le carte di credito con cui sono convenzionati. Per ogni tipo di carta di credito è disponibile un numero verde in caso sia necessaria una consulenza telefonica.



## ES3 - SOLUZIONE



CUCINA(CODICE,Descrizione)  
EFFETTUA( COD CUCINA, COD RISTORANTE, Data)  
RISTORANTE(CODICE,Nome,zona,indirizzo)  
CARTA\_CREDITO(CODICE,Nome,n\_verde)  
CONVENZIONE(CODICE RIST,CODICE CARTA)  
LINEA(CODICE,Nome)  
RAGGIUNTA(CODICE\_RIST,CODICE LINEA)



### ***Produrre uno schema Entità-Relazione e schema relazionale per la seguente realtà di interesse***

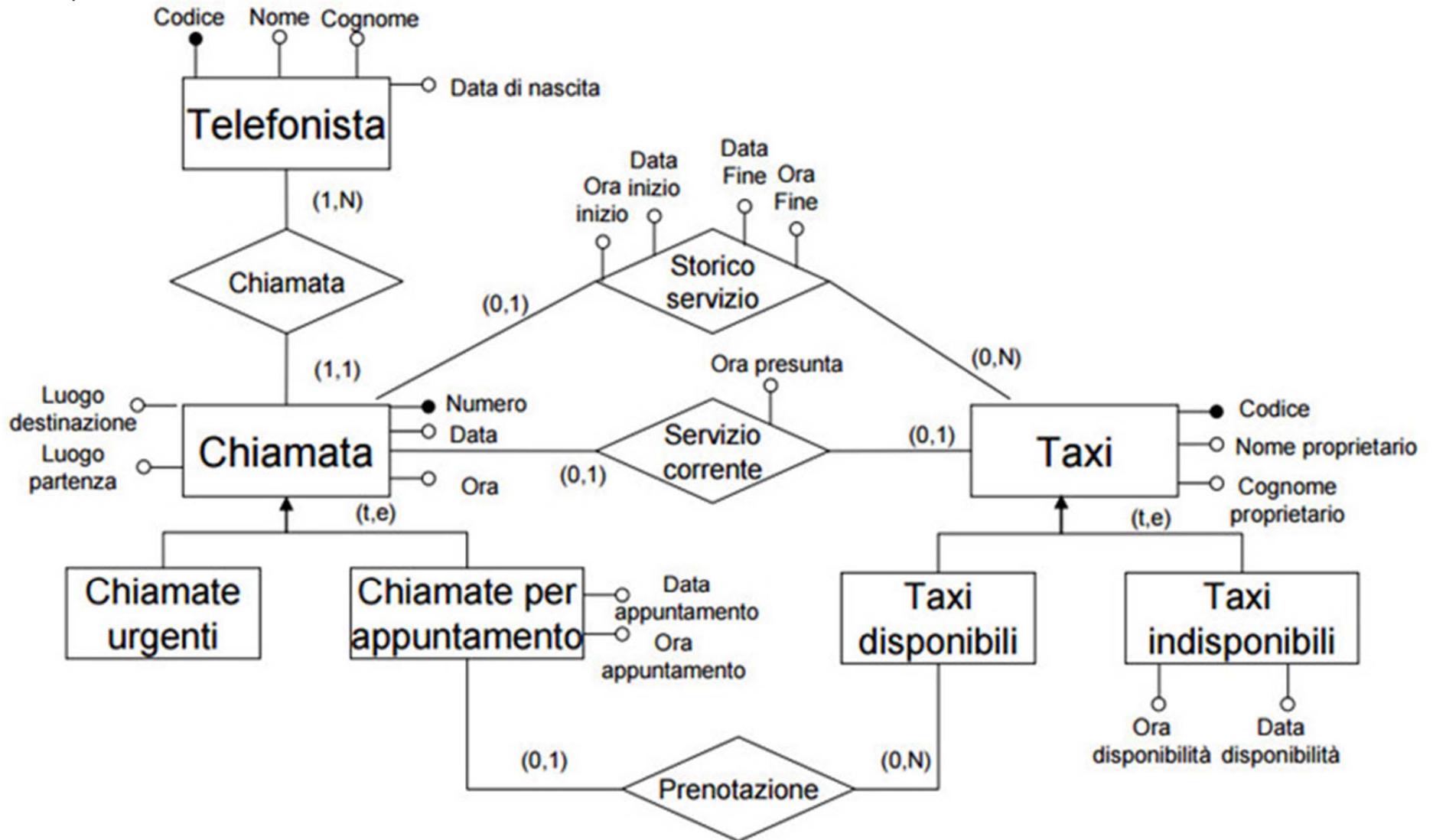
Il centralino del radio-taxi riceve le chiamate di richiesta di un taxi; al centralino lavorano le telefoniste, caratterizzate da un codice, cognome, nome, data di nascita. Le chiamate vengono tutte registrate e sono caratterizzate da un numero di chiamata, la data e ora della chiamata, il luogo della partenza e il luogo di destinazione (che le telefoniste richiedono sempre, non accettando chiamate se questo non viene fornito); per ogni chiamata, si vuole anche sapere quale è stata la telefonista che la ha ricevuta. Le chiamate sono poi suddivise in chiamate urgenti e chiamate per appuntamento; per queste ultime si vuole conoscere anche la data e ora dell'appuntamento richiesto dal cliente.

I taxi sono caratterizzati da un codice, il nome e cognome del proprietario (che fa anche da conducente); i taxi si dividono in disponibili (che possono quindi servire le chiamate) e indisponibili, e di questi ultimi si vuole sapere fino a che data e ora sono indisponibili. •

Le chiamate vengono servite dai taxi; si vuole sapere in un certo momento chi sta servendo una certa chiamata (una chiamata può essere servita al più da un taxi e un taxi può servire al più una chiamata), con l'indicazione dell'ora presunta alla quale il taxi terminerà di servire la chiamata; in più, si vuole anche tenere storia di tutte le volte che un taxi ha servito una chiamata, con l'indicazione della data e ora di inizio e data e ora di fine servizio; da ultimo, i taxi disponibili vengono riservati per servire le chiamate ad appuntamento (per ogni chiamata c'è al più un taxi riservato).



# ES4 - SOLUZIONE





## ES4 - SOLUZIONE

- Telefonista (**cod telefonista**, nome, cognome, data\_nascita)
- Chiamata (**numero**, data, ora, luogo\_partenza, luogo\_destinazione, cod\_telefonista, tipo, data\_appuntamento\*, ora\_appuntamento\*, cod\_taxi\_prenotato\*)
- Taxi (**cod taxi**, nome\_proprietario, cognome\_proprietario, tipo, data\_disponibilità, ora\_disponibilità)
- Servizio corrente (**numero chiamata**, cod\_taxi, ora\_presunta)
- Storico servizio (**numero chiamata**, cod\_taxi, ora\_inizio, data\_inizio, ora\_fine, data\_fine)



## Es5 – DB Concerto

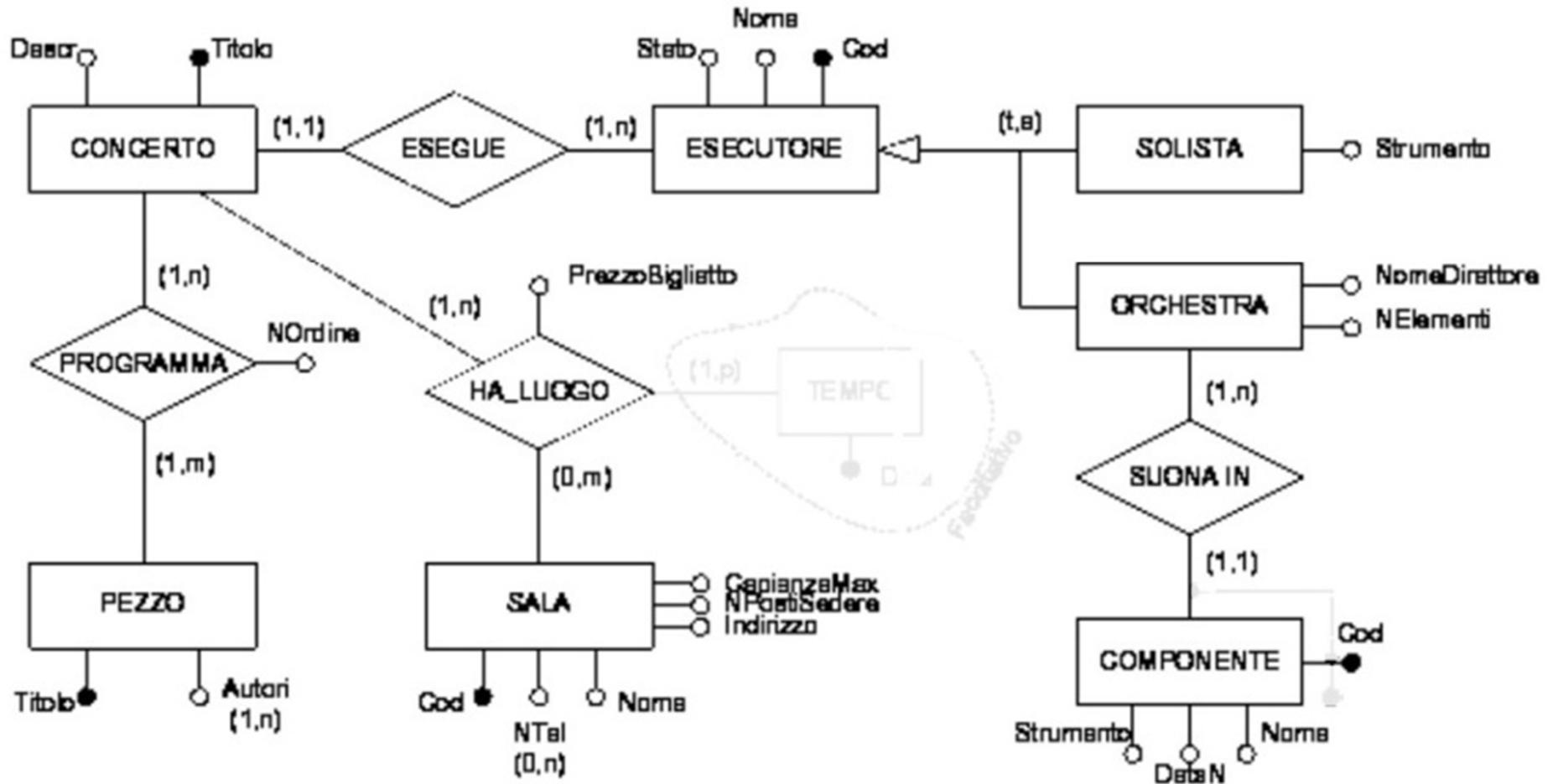
- ***Produrre uno schema Entità-Relazione per la seguente realtà di interesse***

Si vuole rappresentare una base di dati per la gestione del programma di una manifestazione di concerti di musica classica. I concerti proposti nel programma sono caratterizzati da titolo e descrizione. Ogni concerto è composto da un certo numero di pezzi musicali, ognuno caratterizzato da un titolo e da uno o più autori. Inoltre, per ogni concerto si conosce l'ordine in cui sono rappresentati i pezzi che fanno parte del programma. L'esecutore di ogni concerto è identificato da un codice, ed è caratterizzato da nome e stato di provenienza. L'esecutore può essere un'orchestra oppure un solista. Nel caso delle orchestre si memorizzano il nome del direttore e il numero di elementi nell'orchestra, mentre nel caso dei solisti si considera lo strumento suonato. Per le orchestre, inoltre, si memorizzano i componenti (un orchestrale può suonare in una sola orchestra), identificati da un codice univoco all'interno dell'orchestra, e caratterizzati da nome, data di nascita e lo strumento suonato. I concerti sono tenuti in sale da concerto, identificate da un codice univoco e caratterizzate da nome, indirizzo, capienza massima, numero di posti a sedere, ed eventualmente da uno o più numeri di telefono. Ogni concerto può essere programmato in una o più date (ma mai nella stessa sala). Per ogni programmazione di un concerto è noto il prezzo del biglietto, che può essere diverso per programmazioni diverse dello stesso concerto.

- ***Costruire, a partire dallo schema concettuale derivato al punto precedente, lo schema logico relazionale per la stessa base di dati.***



# ES6 - SOLUZIONE





## ES6 - SOLUZIONE

- CONCERTO(TitoloC, Descr)
- SOLISTA(Cod, Nome, Stato, Strumento)
- ORCHESTRA(Cod, Nome, Stato, NomeDirettore, NElementi)
- COMPONENTE(CodCom, CodOrchestra, Nome, DataN, Strumento)
- ORCHESTRA\_ESEGUE(CodConcerto, CodOrchestra)
- SOLISTA\_ESEGUE(CodConcerto, CodSolista)
- SALA(Cod, Nome, Indirizzo, NPostiSedere, CapienzaMax)
- NTEL\_SALA(Numero, CodSala)
- HA\_LUOGO(CodConcerto, CodSala, Data, PrezzoBiglietto)
- PROGRAMMA (TitoloC, TitoloP, Nordine)
- AUTORI(TitoloP, NomeAutore)