



La Fabbrica di Marchingegni

“Sistema Wonka”

PROGETTO DI BASI DI DATI - Corso A

| | |
|----------------------------|---------------|
| Gaetano Antonicchio | 616685 |
| Mario Proia | 616679 |
| Giada Traina | 616682 |

Data di consegna: 26/06/2021

Anno Accademico 2020/2021

1. Descrizione del dominio

Il Software Wonka si occupa di gestire i dipendenti, la costruzione e la vendita di marchingegni all'interno di una fabbrica.

Ogni **Dipendente** ha un unico codice identificativo (codice dipendente) assegnato al momento dell'assunzione. Per ogni dipendente si raccolgono dati relativi al nome, cognome, anno di nascita, codice fiscale (indicato con CF nello schema), indirizzo, telefono, nazionalità e sesso. Un dipendente viene assunto dall'azienda a mezzo di contratti mensili. La retribuzione di ogni dipendente dipende dalla quantità di ore lavorate. Ogni **Contratto** prevede un ammontare massimo di ore mensili, per le quali viene specificata una tariffa oraria. Le **Presenze** vengono utilizzate per tenere traccia delle ore effettive in cui il dipendente ha prestato servizio nella fabbrica e vengono registrate manualmente dal dipendente (questo permette di calcolare l'importo da versare al dipendente ogni mese).

Ogni dipendente lavora a più **Progetti di Costruzione** e ad ogni progetto prendono parte più dipendenti. Tra questi, un dipendente viene scelto come **Capogruppo**, il quale assume responsabilità su tutte le fasi del progetto al quale prende parte.

Ogni progetto di costruzione ha una data di inizio e una data di fine. Durante il periodo prefissato è possibile realizzare diversi marchingegni. Quando un marchingegno viene costruito, viene indicata la data di costruzione e viene assegnato un numero seriale (che identifica il singolo marchingegno costruito).

I **Marchingegni Costruiti** sono delle istanze dei **Modelli di Marchingegno** (i quali sono un'entità astratta). Quest'ultimi presentano tutti un codice modello, un nome, una descrizione e un prezzo di listino. Un marchingegno costruito può essere acquistato da un solo cliente, tuttavia, è possibile che per un singolo marchingegno costruito ci sia più di una prenotazione, poiché una prenotazione può essere annullata e un altro cliente potrebbe prenotare nuovamente quel marchingegno.

Una **prenotazione**, per essere valida, deve essere approvata da un addetto vendite (che è un dipendente della fabbrica). Si è scelto di non tenere traccia delle prenotazioni che non vengono approvate. Ogni prenotazione ha un codice di prenotazione, una data di prenotazione, un importo anticipato (che corrisponde al 40% dell'importo totale) e una data di consegna (specificata dal cliente).

Dato che una prenotazione può essere annullata, sono state distinte due sottoclassi: **Prenotazioni Annullate** e **Prenotazioni Finalizzate**. La prima è una prenotazione per la quale il cliente ha deciso di non acquistare più il marchingegno/i e viene specificata, in tal caso, la data di annullamento. L'annullamento può essere **gratuito** (se il recesso viene comunicato entro le 48 ore dalla data di presentazione) o **oneroso** (in questo caso è prevista una tassa di annullamento).

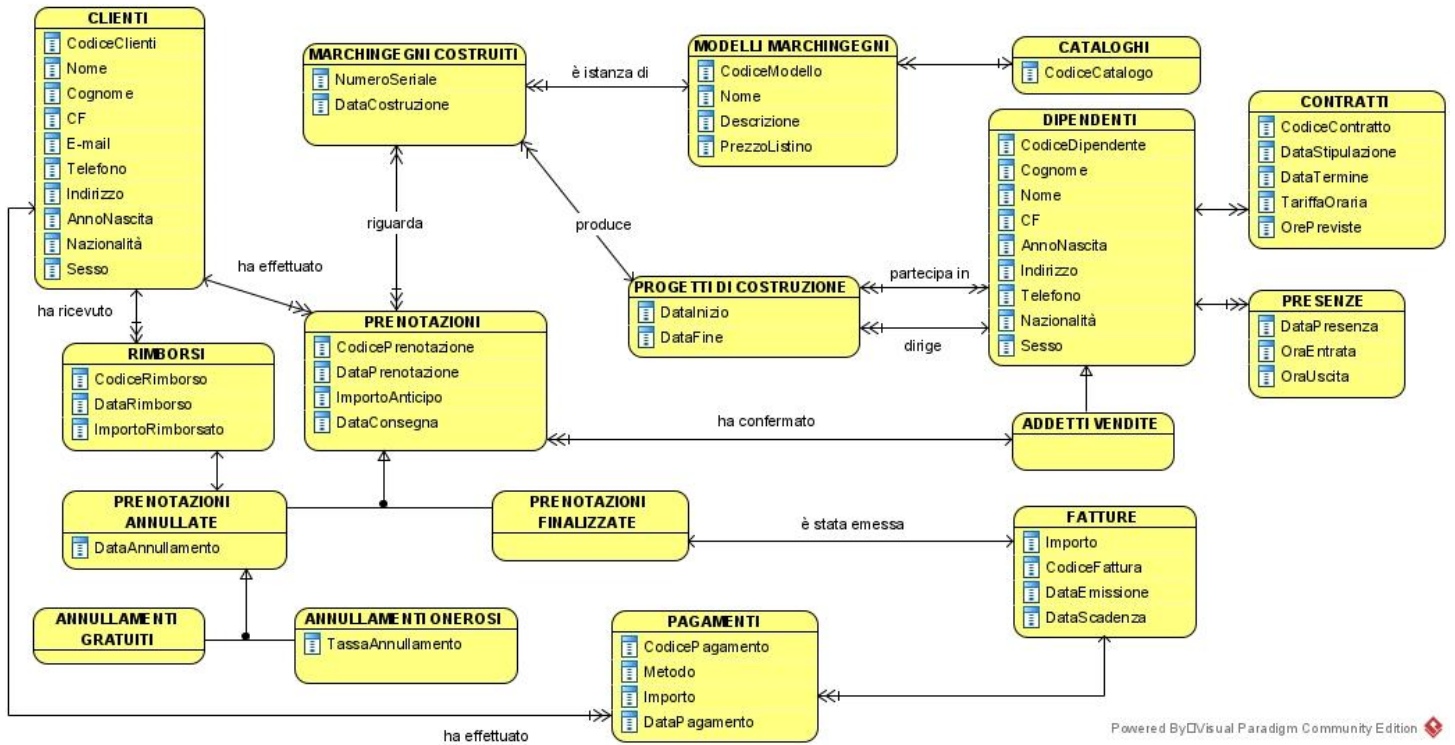
Poiché al momento della prenotazione il cliente aveva anticipato l'importo, è previsto il **Rimborso** della somma versata. Ogni cliente può ricevere nessuno o tanti rimborsi ed ogni rimborso riguarda un singolo cliente. Per ogni rimborso si tiene traccia del codice rimborso, della data e dell'importo rimborsato.

Per ogni **Prenotazione finalizzata** viene emessa una fattura, la quale riguarda una singola prenotazione. Ogni fattura presenta un importo, un codice fattura (che la identifica), una data di emissione e una data di scadenza. La prenotazione è finalizzata quando viene consegnato fisicamente il marchingegno all'atto della vendita. In quel momento viene consegnata la fattura. È stato ritenuto opportuno considerare la prenotazione annullabile in qualsiasi momento, ma comunque prima della presa in consegna del marchingegno.

Il cliente può effettuare più **Pagamenti** sulla relativa fattura, fino a quando l'importo pagato non eguaglia quello della fattura.

Ogni **Cliente**, per poter acquistare un marchingegno, deve essere necessariamente registrato sul sito web della fabbrica. L'iscrizione avviene tramite un indirizzo e-mail, il quale non può assumere valori nulli. Inoltre, per ogni cliente, viene registrato il codice cliente (che lo identifica), il nome, il cognome, il codice fiscale, il telefono, l'anno di nascita, il sesso, la nazionalità e l'indirizzo.

2. Schema Concettuale



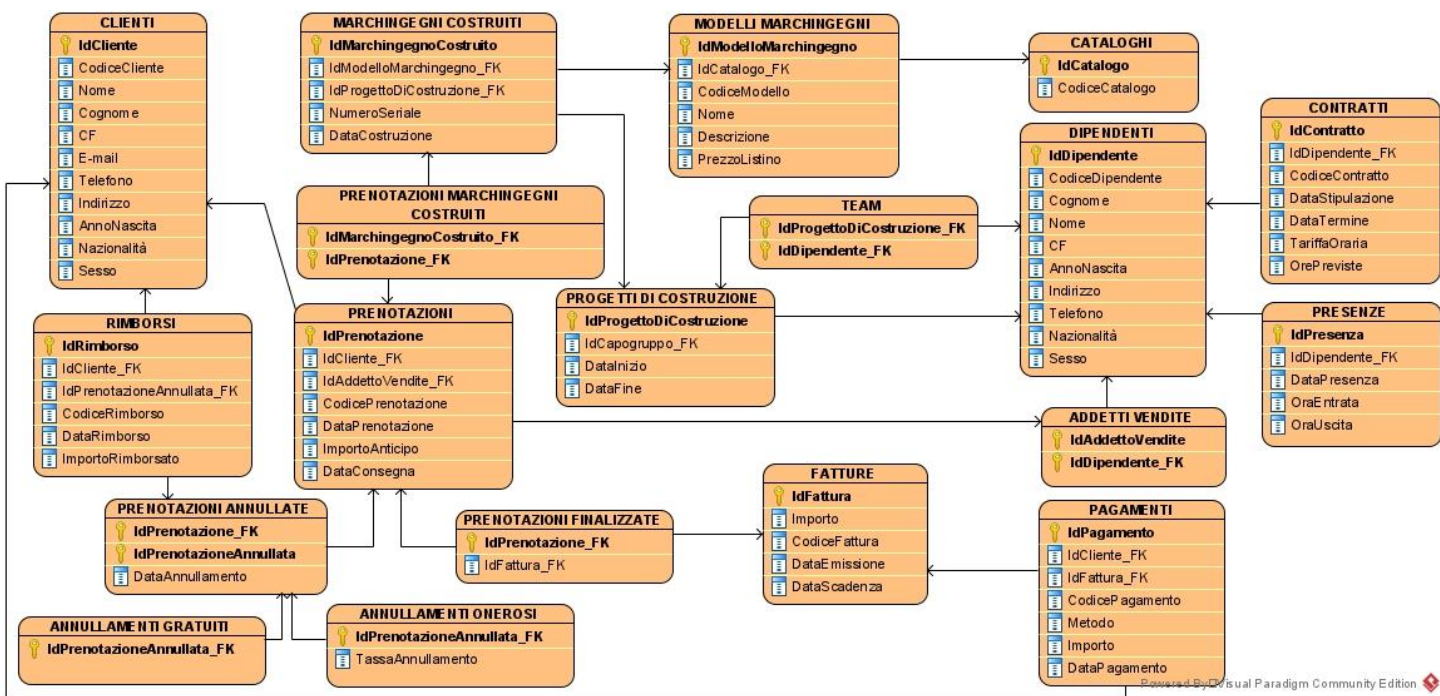
Vincoli intra-relazionali:

- Non possono esistere due Clienti con lo stesso codice fiscale;
- Non possono esistere due Dipendenti con lo stesso codice fiscale;
- Non possono esistere due Fatture con lo stesso codice fattura;
- Non possono esistere due Rimborsi / Pagamenti con lo stesso codice rimborso/pagamento.
- In Pagamenti, il Metodo può avere solo uno dei seguenti valori: ‘contanti’, ‘mastercard’, ‘paypal’;
- In Fatture, Rimborsi, Pagamenti e Prenotazioni, l’importo è un intero con valori > 0 ;
- Il codice fiscale (CF) di Dipendenti e Clienti, deve essere di 16 caratteri;
- Il numero di telefono dei Clienti e dei Dipendenti non può avere meno di 9 cifre;
- Tutte le date (DataCostruzione, DataCosegna, DataEmissione, DataScadenza, DataAnnullamento, DataInizio, DataFine, DataPresenza, DataPagamento) devono essere in formato Date;
- In Presenze, OraEntrata e OraUscita sono in formato Time;
- In Presenze l’OraUscita non può essere antecedente all’OraEntrata per la stessa Data;
- In ProgettiDiCostruzione, DataInizio deve essere \leq alla DataFine;
- In Fatture, la DataEmissione deve essere \leq alla DataScadenza;
- In Prenotazioni, la DataPrenotazione deve essere \leq alla DataConsegna;
- Gli attributi di Clienti e Dipendenti non possono avere valori NULL;
- In MarchingegniCostruiti il NumeroSeriale è UNIQUE;
- Il Sesso di Clienti e Dipendenti può essere solo: ‘M’, ‘F’;

Vincoli inter-relazionali:

- Il totale dei pagamenti relativi ad una singola prenotazione, deve avere un importo pari a quello specificato nella fattura;
- In Prenotazioni, l'ImportoAnticipo è pari al 40% dell'importo fatturato per quella prenotazione;
- La TassaAnnullamento corrisponde al 5% dell'ImportoAnticipo;
- In Rimborsi, l'ImportoRimborsato è pari all'ImportoAnticipo se l'annullamento della prenotazione è gratuito, altrimenti corrisponde all'ImportoAnticipo - TassaAnnullamento;
- La DataAnnullamento deve essere \geq alla DataPrenotazione;
- La DataRimborso deve essere \geq alla DataAnnullamento;
- La DataPrenotazione deve essere \geq alla DataCostruzione;

3. Schema logico relazionale



Schema logico relazionale in formato testuale

Cataloghi (IdCatalogo, CodiceCatalogo)

ModelliMarchingegni (IdModelloMarchingegno, IdCatalogo*, CodiceModello, Nome, Descrizione, PrezzoListino)

MarchingegniCostruiti (IdMarchingegnoCostruito, IdModelloMarchingegno*, IdProgettoDiCostruzione*, NumeroSeriale, DataCostruzione)

ProgettiDiCostruzione (IdProgettoDiCostruzione, IdCapogruppo*, DataInizio, DataFine)

Team (IdProgettoDiCostruzione*, IdDipendente*)

Dipendenti (IdDipendente, CodiceDipendente, Nome, Cognome, CF, Sesso, AnnoNascita, Indirizzo, Telefono, Nazionalità)

AddettiVendite (IdAddettoVendite, IdDipendente*)

Contratti (IdContratto, IdDipendente*, CodiceContratto, DataStipulazione, DataTermine, OrePreviste, TariffaOraria)

Presenze (IdPresenza, IdDipendente*, DataPresenza, OraEntrata, OraUscita)

Prenotazioni (IdPrenotazione, IdCliente*, IdAddettoVendita*, CodicePrenotazione, DataPrenotazione, ImportoAnticipo, DataConsegna)

Prenotazioni-MarchingegniCostruiti (IdPrenotazione*, IdMarchingegnoCostruito*)

PrenotazioniAnnullate (IdPrenotazione*, IdPrenotazioneAnnullata, DataAnnullamento)

AnnullamentiGratuiti (IdPrenotazioneAnnullata*)

AnnullamentiOnerosi (IdPrenotazioneAnnullata*, TassaAnnullamento)

PrenotazioniFinalizzate (IdPrenotazione*, IdFattura*)

Fatture (IdFattura, CodiceFattura, Importo, DataEmissione, DataScadenza)

Pagamenti (IdPagamento, IdCliente*, IdFattura*, CodicePagamento, Metodo, Importo, DataPagamento)

Rimborsi (IdRimborso, IdCliente*, IdPrenotazioneAnnullata*, CodiceRimborso, DataRimborso, ImportoRimborsato)

Clienti (IdCliente, CodiceCliente, Nome, Cognome, Sesso, AnnoNascita, Indirizzo, Email, Telefono, CF, Nazionalità)

Dipendenze Funzionali

Cataloghi:

IdCatalogo → CodiceCatalogo
CodiceCatalogo → IdCatalogo

Clienti:

IdCliente → CodiceCliente, Nome, Cognome, Sesso, AnnoNascita, Indirizzo, Email, Telefono, CF, Nazionalità
Telefono → Nome, Cognome, Sesso, AnnoNascita, Indirizzo, CodiceCliente, Email, IdCliente, CF, Nazionalità
Email → Nome, Cognome, Sesso, AnnoNascita, Indirizzo, CodiceCliente, Telefono, IdCliente, CF, Nazionalità
CF → CodiceCliente, Nome, Cognome, Sesso, AnnoNascita, Indirizzo, Email, Telefono, IdCliente, Nazionalità
CodiceCliente → Nome, Cognome, Sesso, AnnoNascita, Indirizzo, IdCliente, Email, Telefono, CF, Nazionalità

Dipendenti:

IdDipendente → CodiceDipendente, Nome, Cognome, AnnoNascita, Indirizzo, Telefono, CF, Nazionalità, Sesso
Telefono → Nome, Cognome, AnnoNascita, Indirizzo, CodiceDipendente, IdDipendente, CF, Nazionalità, Sesso
CF → CodiceDipendente, Nome, Cognome, AnnoNascita, Indirizzo, Telefono, IdDipendente, Nazionalità, Sesso
CodiceDipendente → Nome, Cognome, AnnoNascita, Indirizzo, IdDipendente, Telefono, CF, Nazionalità, Sesso

Contratti:

IdContratto → CodiceContratto, DataStipulazione, DataTermine, TariffaOraria, OrePreviste, IdDipendente
CodiceContratto → IdContratto, DataStipulazione, DataTermine, TariffaOraria, OrePreviste, IdDipendente

MarchingegniCostruiti:

IdMarchegnoCostruito → NumeroSeriale, DataCostruzione, IdProgettoDiCostruzione, IdModelloMarchegno
NumeroSeriale → IdMarchegnoCostruito, IdModelloMarchegno, DataCostruzione, IdProgettoDiCostruzione

ModelliMarchingegni:

IdModelloMarchegno → Nome, IdCatalogo, Descrizione, PrezzoListino, CodiceModello
CodiceModello → Nome, IdCatalogo, Descrizione, PrezzoListino, IdModelloMarchegno

Fatture:

IdFattura → CodiceFattura, Importo, DataEmissione, DataScadenza
CodiceFattura → IdFattura, Importo, DataEmissione, DataScadenza

Pagamenti:

IdPagamento → IdCliente, IdFattura, CodicePagamento, Metodo, Importo, DataPagamento
CodicePagamento → IdCliente, IdFattura, IdPagamento, Metodo, Importo, DataPagamento

Rimborsi:

IdRimborso → IdCliente, IdPrenotazioneAnnullata, CodiceRimborso, DataRimborso, ImportoRimborsato
CodiceRimborso → IdCliente, IdPrenotazioneAnnullata, IdRimborso, DataRimborso, ImportoRimborsato

ProgettiDiCostruzione:

IdProgettoDiCostruzione → IdCapogruppo, DataInizio, DataFine
IdProgettoDiCostruzione, DataInizio, DataFine → IdCapogruppo

Prenotazioni:

IdPrenotazione → CodicePrenotazione, IdCliente, IdAddettoVendita, CodicePrenotazione, DataPrenotazione,
ImportoAnticipo, DataConsegna
CodicePrenotazione → IdPrenotazione, IdCliente, IdAddettoVendita, CodicePrenotazione, DataPrenotazione,
ImportoAnticipo, DataConsegna

Presenze:

IdPresenza → IdDipendente, DataPresenza, OraEntrata, OraUscita
IdPresenza, DataPresenza, OraEntrata, OraUscita → IdDipendente

Tutti gli schemi **sono in BCNF** poiché, ogni dipendenza funzionale ha come determinante, o la chiave primaria o la chiave naturale, che per motivi pratici non è stata scelta come primaria. Come si può notare dalle dipendenze funzionali elencate, le chiavi naturali come ad esempio 'CF' oppure 'NumeroSeriale', sono tutte superchiavi.

4. Interrogazioni in SQL

a. Uso di Proiezione, Join e Restrizione;

Per ogni dipendente che ha lavorato in qualche progetto di costruzione iniziato dopo il 2018, restituire IdDipendente, Nome e Cognome.

```
SELECT DISTINCT d.IdDipendente, d.Nome, d.Cognome
FROM Dipendenti d
JOIN Team t ON (t.IdDipendente = d.IdDipendente)
JOIN ProgettiDiCostruzione pdc ON (pdc.IdProgettoDiCostruzione =
t.IdProgettoDiCostruzione)
WHERE pdc.DataInizio >= '01/01/2019'
```

b. Uso di Group By con Having, Where e Sort;

Per ogni Nazionalità, riportare il numero di clienti nati dopo il 1990, se il numero totale di clienti è almeno 100. Il risultato deve essere ordinato in ordine crescente.

```
SELECT c.Nazionalità, COUNT(*) AS Num_Clienti
FROM Clienti c
WHERE (c.AnnoNascita > '1990')
GROUP BY c.Nazionalità
HAVING (Num_Clienti ≥ 100)
ORDER BY Num_Clienti
```

c. Uso di Join, Group By con Having e Where;

Per ogni nazione riportare il numero totale delle prenotazioni di marchinegni effettuate prima del 2010 da clienti di sesso maschile. Si è interessati soltanto alle nazioni che hanno almeno 10 prenotazioni ciascuna.

```
SELECT c.Nazionalità, COUNT(*) AS NumPrenotazioni
FROM Clienti c
JOIN Prenotazioni p ON (p.IdCliente = c.IdCliente)
WHERE (p.DataPrenotazione < '01/01/2010') AND (c.Sesso = 'M')
GROUP BY c.Nazionalità
HAVING (NumPrenotazioni ≥ 10)
```

d. Uso di Select annidata con Quantificazione Esistenziale;

Per ogni cliente che ha effettuato almeno un pagamento prima del 2017 riportare Nome, Cognome e IdCliente.

```
SELECT c.Nome, c.Cognome, c.IdCliente
FROM Clienti c
WHERE EXISTS (SELECT *
               FROM Pagamenti p
               WHERE (c.IdCliente = p.IdCliente)
               AND (p.Data < '01/01/2017'))
```

e. Uso di Select annidata con Quantificazione Universale;

Per ogni cliente che ha pagato solo in contanti, riportare l'IdCliente, Nome, Cognome e numero totale di prenotazioni effettuate a suo nome.

```
SELECT c.IdCliente, c.Nome, c.Cognome, COUNT(*) AS NumPrenotazioni
FROM Clienti c JOIN Prenotazioni p ON (p.IdCliente = c.IdCliente)
WHERE NOT EXISTS (SELECT *
                  FROM Pagamenti p
                  WHERE (c.IdCliente = p.IdCliente)
                  AND NOT (p.Metodo = 'Contanti'))
GROUP BY c.IdCliente, c.Nome, c.Cognome
```

f. Uso di subquery di confronto quantificato usando una subquery.

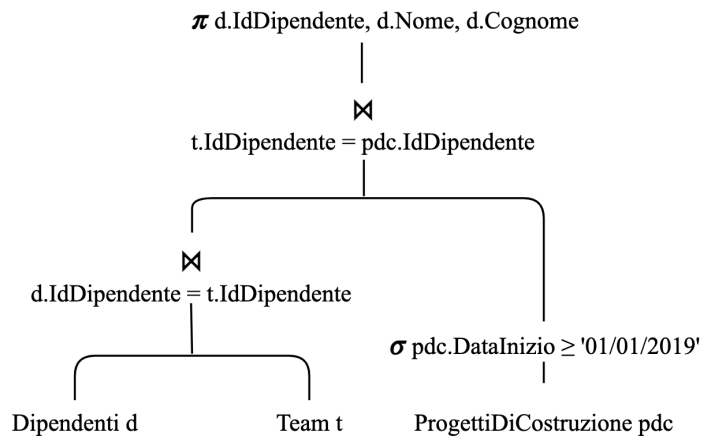
Riportare il CodicePagamento e DataPagamento, dei pagamenti effettuati in contanti, il cui importo è maggiore di ogni pagamento effettuato con metodo mastercard.

```
SELECT p1.CodicePagamento, p1.DataPagamento
FROM Pagamenti p1
WHERE (p1.Metodo = 'Contanti')
AND p1.Importo > ALL (SELECT p2.Importo
                     FROM Pagamenti p2
                     WHERE (p2.Metodo = 'Mastercard'))
```


5. Piani di accesso logici

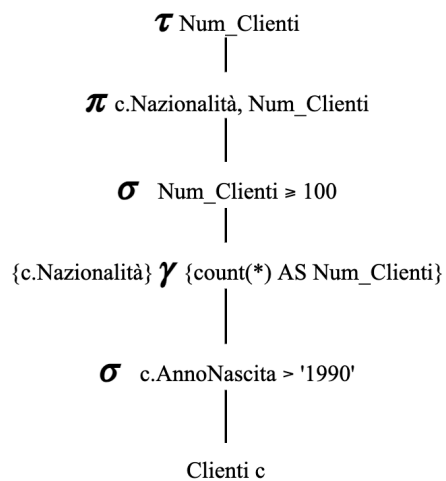
Query a)

Per ogni dipendente che ha lavorato a qualche progetto di costruzione dopo il 2018, riportare IdDipendente, Nome e Cognome.



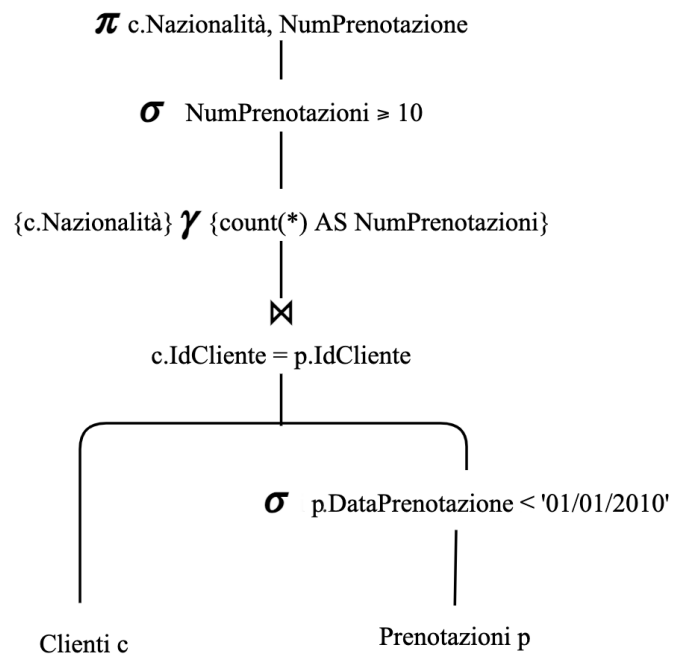
Query b)

Per ogni Nazionalità, riportare il numero di clienti nati dopo il 1990, se il numero totale di clienti è almeno 100. Il risultato deve essere ordinato in ordine crescente.



Query c)

Per ogni Nazione (del Cliente) riportare il numero totale di prenotazioni effettuate prima del 2010, se il numero totale è almeno 10.

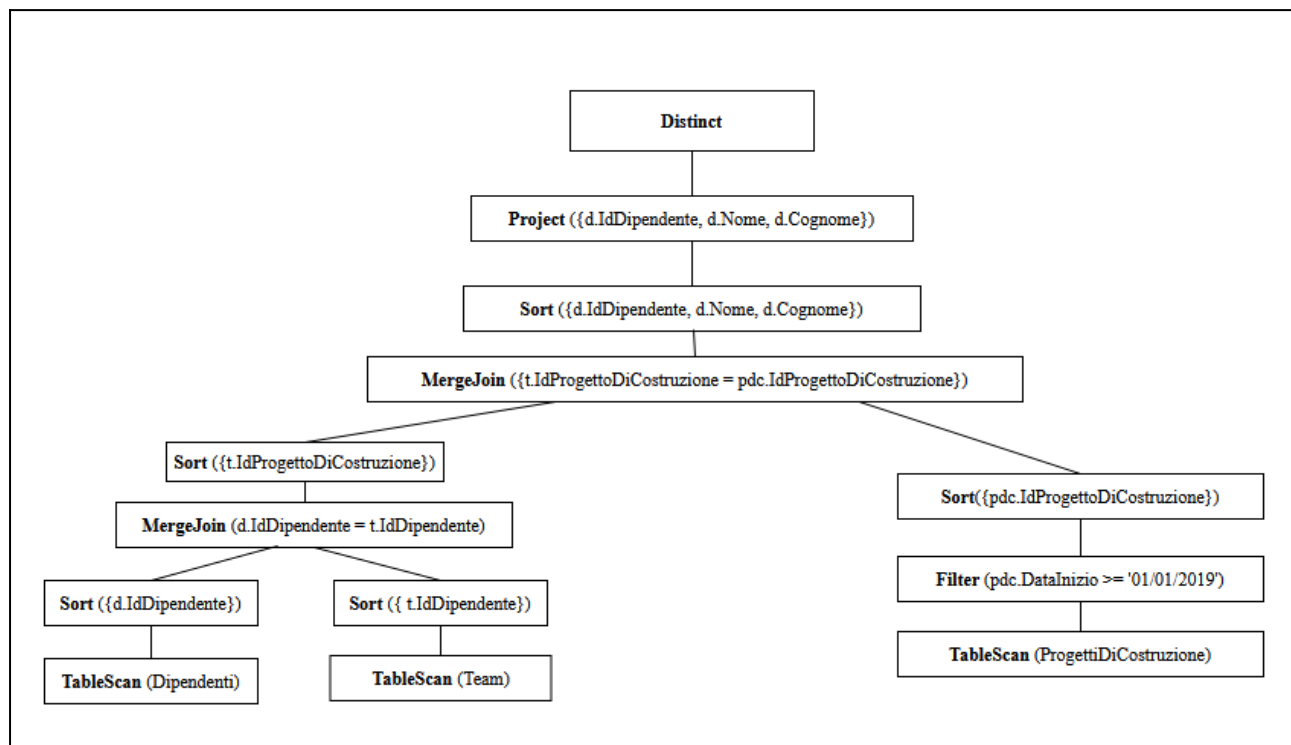


6. Piani di accesso fisici

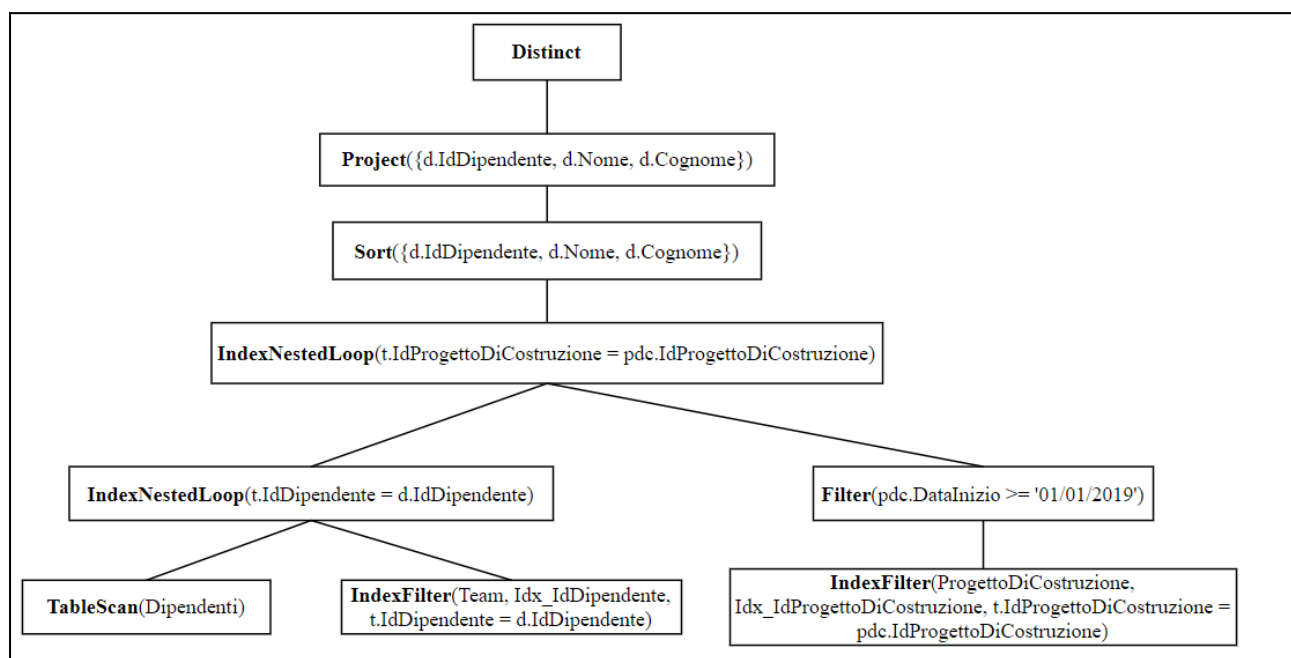
Query a)

Per ogni dipendente che ha lavorato a qualche progetto di costruzione dopo il 2018, riportare IdDipendente, Nome e Cognome.

Senza indici



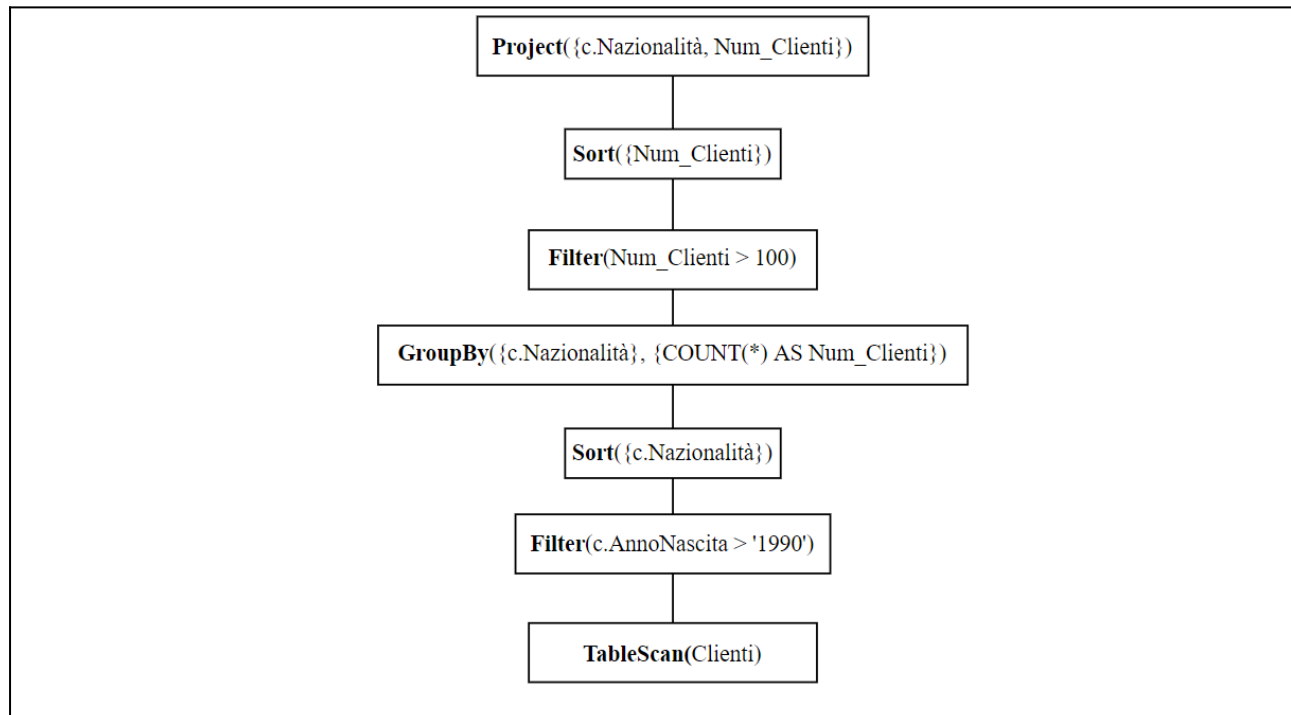
Con indici su Team.IdDipendente e ProgettiDiCostruzione.IdProgettoDiCostruzione



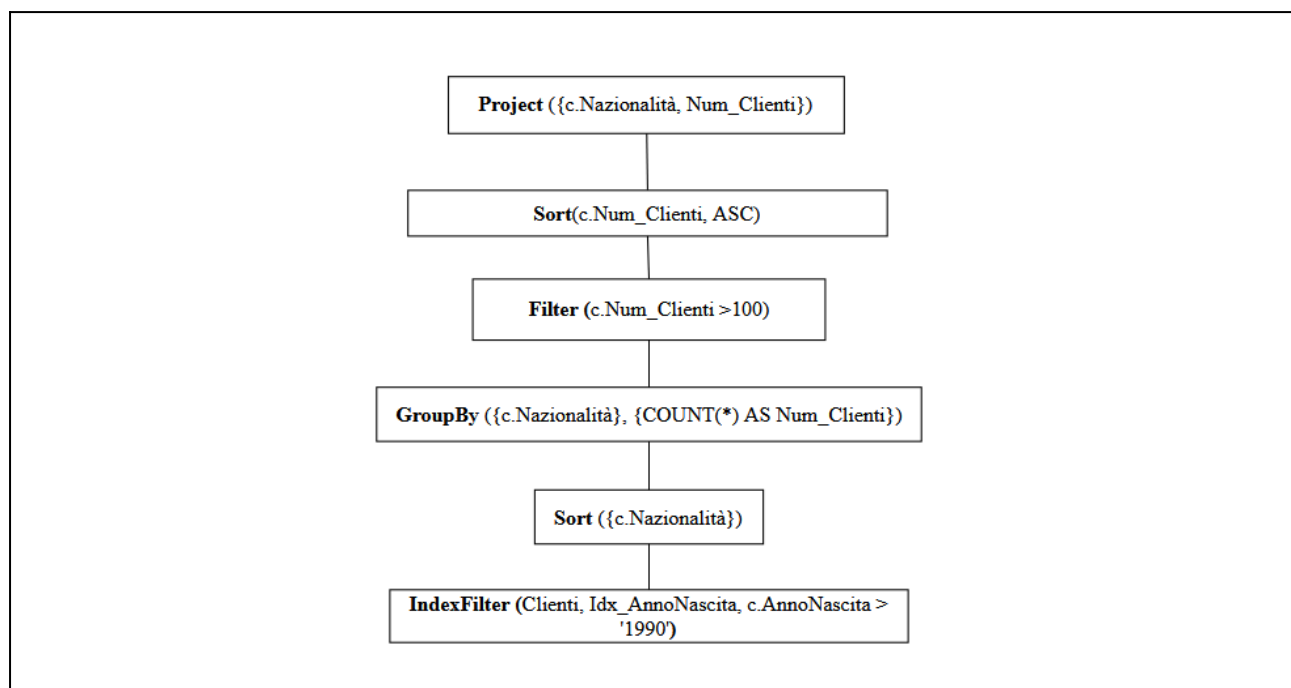
Query b)

Per ogni Nazionalità, riportare il numero di clienti nati dopo il 1990, se il numero totale di clienti è almeno 100. Il risultato deve essere ordinato in ordine crescente.

Senza indici



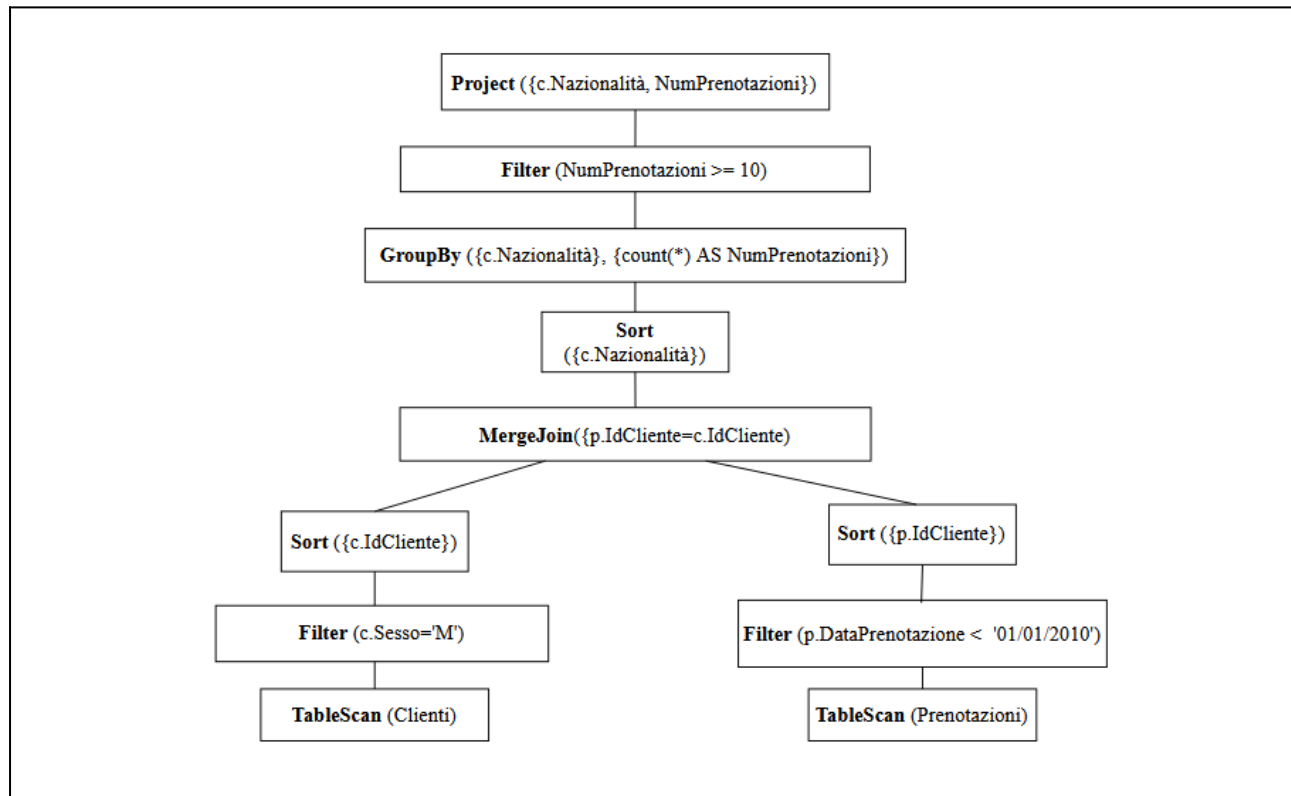
Con indice su Clienti.AnnoNascita



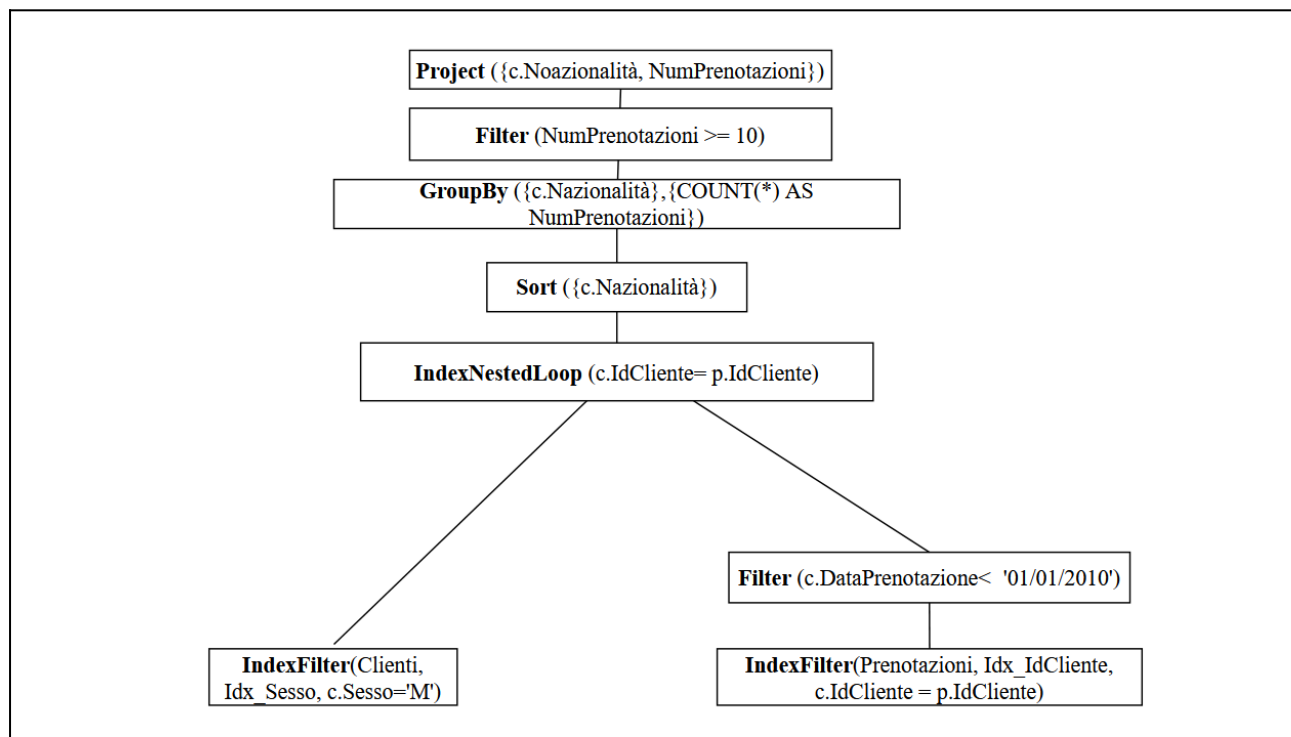
Query c)

Per ogni Nazione (del Cliente) riportare il numero totale di prenotazioni effettuate prima del 2010, se il numero totale è almeno 10

Senza indici



Con indici c su Prenotazioni.IdCliente, Clienti.Sesso)



Note sull'ottimizzazione dei piani d'accesso

In tutte le query per le quali sono stati costruiti piani d'accesso fisici (sia con che senza indice/i), in generale non è possibile ottimizzare ulteriormente i piani d'accesso rimuovendo la Sort prima della GroupBy.

Questo perché non siamo a conoscenza del tipo di organizzazione dei dati usata. Quindi non abbiamo garanzia che l'input della GroupBy sia ordinato sull'attributo di raggruppamento.

Tuttavia, nella **query b** è possibile rimuovere la Sort prima della GroupBy, soltanto se i dati relativi ai clienti sono memorizzati utilizzando un'organizzazione sequenziale sul campo Nazionalità. In questo caso, dato che la GroupBy raggruppa proprio sul campo Nazionalità, non è necessario ordinare i dati in quanto questi saranno già ordinati sull'attributo di raggruppamento.

Nella **query c**, anche se l'organizzazione dei dati è di tipo sequenziale sull'attributo Nazionalità, non è possibile rimuovere la Sort in quanto nel piano d'accesso fisico l'operatore MergeJoin/IndexNestedLoops ritorna in output dati ordinati in base all'IdCliente, il quale non corrisponde all'attributo di raggruppamento specificato nella GroupBy.