

# Progetto Basi di Dati - SecureNet

Bulychov Francesco Constantino - Matteo Mazzotti

## 1) Abstract

La SecureNet è un'azienda leader nel settore della sicurezza informatica, con una vasta presenza in Europa, negli Stati Uniti e nel Regno Unito. Oltre a fornire servizi di sicurezza informatica a livello globale, l'azienda offre una gamma completa di antivirus, progettati sia per privati che per organizzazioni aziendali. Questi antivirus sono disponibili in diversi pacchetti che soddisfano le diverse esigenze dei clienti.

La struttura aziendale si articola in varie sedi, ciascuna con un significativo numero di dipendenti. Il nucleo di ciascuna sede è costituito da un team di professionisti altamente qualificati, responsabili dell'esecuzione dei vari lavori richiesti dalle aziende clienti. Tali lavori possono essere suddivisi in due categorie principali: i "Servizi", che sono incarichi gestiti da specialisti individuali, e i "Progetti", che rappresentano compiti più complessi e vengono svolti da gruppi di dipendenti sotto la guida di un membro esperto del team.

Un altro elemento cruciale dell'organizzazione è il team dedicato all'assistenza clienti. Questo team è composto da operatori esperti che sono pronti ad affrontare le sfide tecniche che i clienti possono incontrare nell'utilizzo dell'antivirus o nei servizi forniti. L'assistenza può riguardare problemi con l'installazione, configurazione o utilizzo dell'antivirus, oltre a fornire soluzioni per questioni minori relative ai servizi offerti.

Per gestire in modo efficiente tutte queste attività, è necessario creare un database che dovrà tracciare le sottoscrizioni degli antivirus da parte dei clienti, la distribuzione delle licenze, i dettagli delle sedi aziendali con i rispettivi dipendenti, la pianificazione, l'esecuzione e il completamento dei lavori e dei progetti, nonché la gestione delle richieste e dell'assistenza clienti.

## 2) Analisi dei requisiti

### 2.1) Descrizione testuale

Si vuole realizzare una base di dati per la gestione delle attività svolte da una azienda multinazionale che si occupa di cybersecurity. Il sistema permetterà di migliorare l'efficienza complessiva dell'azienda tenendo traccia dei lavori in svolgimento, delle licenze degli antivirus venduti, delle recensioni, delle attività di assistenza e del personale.

La base di dati in questione presenta alcune entità principali dotate di attributi specifici, ovvero:

L'entità Cliente contiene le informazioni riguardanti i clienti dell'azienda come email, cognome, nome, indirizzo e telefono. Tra loro possono esserci sia privati sia organizzazioni, e se per entrambi esistono pacchetti specifici dell'antivirus, solamente le organizzazioni possono richiedere lavori all'azienda come servizi o progetti. Per ogni organizzazione viene anche memorizzato il rispettivo nome dell'azienda e la partita IVA.

L'entità Antivirus contiene informazioni sugli antivirus che l'azienda propone ai clienti. In particolare, ogni antivirus è identificato dal nome della versione e di cui viene memorizzato il prezzo mensile e una breve descrizione che riassume le sue caratteristiche. Tra i vari pacchetti disponibili, sono presenti opzioni che possono soddisfare sia clienti privati sia organizzazioni di qualsiasi dimensione.

L'entità Sottoscrizione permette di tenere traccia di tutti gli acquisti dei clienti riguardanti l'antivirus. Per ogni sottoscrizione viene memorizzata la rispettiva licenza, ovvero un codice seriale identificativo che svolge il ruolo di chiave di attivazione, la data di inizio in cui è stato effettuato l'acquisto e infine la data di scadenza.

L'entità Recensione contiene informazioni sulle recensioni che i clienti effettuano sulla copia dell'antivirus acquistata. Ogni recensione possiede un identificativo, una data in cui è stata scritta, un voto che varia da 0 a 5 e un commento opzionale che il cliente può scegliere di scrivere.

L'entità Sede contiene le informazioni riguardanti le sedi dell'azienda, ognuna delle quali ha un nome univoco, una email, un numero di telefono e un indirizzo. Inoltre le sedi si occupano dell'assunzione dei dipendenti dell'azienda.

L'entità Dipendente contiene le informazioni sui dipendenti che lavorano all'interno di SecureNet. Ognuno di loro è identificato dal rispettivo codice fiscale, nome, cognome, email, numero di telefono e stipendio. Ogni dipendente può essere un professionista che si occupa dello svolgimento e/o coordinamento dei lavori, un assistente che si occupa del servizio di assistenza ai clienti oppure un direttore di una delle sedi dell'azienda.

L'entità Lavoro contiene le informazioni riguardanti i lavori che vengono richiesti dalle organizzazioni all'azienda. Di ogni lavoro è noto l'identificativo, il nome, il costo e una breve descrizione che racchiude i punti principali di ciò che verrà svolto. Inoltre viene tenuta traccia dell'inizio, dello stato in cui si trova attualmente la richiesta e la fine. Tra loro possono essere distinti in due categorie principali: servizi e progetti. I servizi riguardano la risoluzione di specifici problemi legati alla sicurezza delle aziende e vengono svolti da un singolo professionista, i progetti sono invece incarichi più lunghi e complessi e in quanto tali vengono svolti da un team coordinato da un professionista con più esperienza.

L'entità Assistenza contiene informazioni sulle richieste di assistenza da parte dei clienti. Per ognuna di esse si tiene traccia del codice identificativo, una breve descrizione del problema e della relativa soluzione, una data di inizio, lo stato corrente della richiesta e una data di fine. Le possibili richieste possono riguardare problemi relativi all'antivirus o ai servizi offerti.

## 2.2) Glossario dei termini

Termine	Descrizione
Cliente	Persona o organizzazione che ha registrato i suoi dati sul sito dell'azienda e che potrebbe potenzialmente effettuare un acquisto.
Antivirus	Software progettato, sviluppato e venduto dall'azienda per la protezione da possibili attacchi informatici e la sicurezza dei dati dei clienti.
Pacchetto	Insieme di funzionalità e strumenti offerti dal software antivirus.
Licenza	Codice seriale identificativo che svolge il ruolo di chiave di attivazione per l'antivirus.

Sede	Edificio di proprietà di una azienda che si occupa di fornire servizi e gestire i clienti di un determinato territorio.
Dipendente	Persona regolarmente assunta dall'azienda che svolge un determinato compito.
Professionista	Dipendente dell'azienda specializzato in un determinato settore che si occupa personalmente dello svoglimento/coordinamento dei lavori commissionati dai clienti.
Assistente	Dipendente dell'azienda che si occupa del servizio di assistenza ai clienti.
Lavoro	Incarico che un'organizzazione richiede all'azienda per affrontare questioni legate alla protezione dei propri sistemi e dati.
Servizio	Lavori relativamente brevi che richiedono l'impiego di un solo professionista.
Progetto	Lavori lunghi e complessi che richiedono un team coordinato da un professionista con più esperienza.

## 2.3) Operazioni tipiche

Operazione	Tipo	Frequenza
Inserimento di nuove richieste di lavoro	S	100 al giorno
Inserimento di nuove richieste di assistenza	S	300 al mese
Gestione delle sottoscrizioni attive dell'antivirus	L	500 al mese
Gestione del personale	S	500 al mese
Registrazione delle recensioni effettuate	S	100 al mese

## 3) Progettazione Concettuale

### 3.1) Lista Entit 

**Cliente** - Gerarchia Totale. Si divide in **Privato** e **Organizzazione**

- Email (Primary Key): VARCHAR(100) - Email del cliente
- Cognome: VARCHAR(100) - Cognome del cliente o del contatto con l'organizzazione
- Nome: VARCHAR(100) - Nome del cliente o del contatto con l'organizzazione
- Telefono: VARCHAR(20) - Numero di telefono del cliente o del contatto con l'organizzazione
- Indirizzo: attributo composto
  - Paese: VARCHAR(20)
  - Citt : VARCHAR(50)
  - CAP: VARCHAR(10)
  - Via: VARCHAR(50)
- **Organizzazione**
  - Nome\_Azienda: VARCHAR(100) - Nome dell'organizzazione
  - Partita\_IVA: VARCHAR(20) - Partita IVA dell'organizzazione

#### **Antivirus**

- Versione (Primary Key): VARCHAR(100) - Nome della versione dell'antivirus
- Descrizione: VARCHAR(255) - Breve descrizione delle principali caratteristiche dell'antivirus
- Prezzo\_Mensile: NUMERIC(5,2) - Prezzo mensile dell'antivirus

#### **Sottoscrizione**

- Licenza (Primary Key): VARCHAR(20) - Codice seriale di una sottoscrizione ad un antivirus
- Inizio: DATE - Data di inizio della sottoscrizione
- Scadenza: DATE - Data di scadenza della sottoscrizione

#### **Recensione**

- ID (Primary Key): INTEGER - Codice identificativo della recensione
- Data\_Pubblicazione: DATE - Data in cui la recensione   stata scritta
- Voto: INT - Voto da 0 a 5 associato alla recensione
- Commento: VARCHAR - Commento opzionale associato alla recensione

## **Sede**

- Indirizzo (Primary Key): attributo composto
  - Paese: VARCHAR(20)
  - Città: VARCHAR(50)
  - CAP: VARCHAR(10)
  - Via: VARCHAR(50)
- Nome : VARCHAR(100) - Nome della sede
- Email: VARCHAR(100) - Email della sede
- Telefono: VARCHAR(20) - Numero di telefono della sede

## **Dipendente** - Gerarchia Totale. Si divide in **Professionista**, **Assistente** e **Direttore**

- CF (Primary Key): VARCHAR(16) - Codice fiscale del dipendente
- Cognome: VARCHAR(100) - Cognome del cliente o del contatto con l'organizzazione
- Nome: VARCHAR(100) - Nome del cliente o del contatto con l'organizzazione
- Email: VARCHAR(100) - Nome del cliente o del contatto con l'organizzazione
- Telefono: VARCHAR(20) - Numero di telefono del cliente o del contatto con l'organizzazione
- Stipendio: INT - Stipendio annuale del dipendente
- **Professionista**
  - Ruolo: VARCHAR(100) - Il ruolo che il professionista ricopre all'interno dell'azienda

## **Lavoro** - Gerarchia Totale. Si divide in **Servizio** e **Progetto**

- ID (Primary Key): INTEGER - Codice identificativo della richiesta del lavoro
- Nome: VARCHAR(100) - Nome del lavoro richiesto
- Descrizione: VARCHAR(255) - Breve descrizione del lavoro
- Costo: INT - Costo totale del lavoro
- Inizio: DATE - Data di presa in carico del lavoro
- Stato: VARCHAR(15) - Stato del lavoro
- Fine: DATE - Data di fine del lavoro

## **Assistenza**

- ID (Primary Key): INTEGER - Codice identificativo della richiesta di assistenza
- Problema: VARCHAR(255) - Breve descrizione del problema per cui é stata richiesta l'assistenza
- Soluzione: VARCHAR(255) - Breve descrizione della soluzione al problema
- Inizio: DATE - Data di presa in carico della richiesta di assistenza
- Stato: VARCHAR(15) - Stato della richiesta di assistenza
- Fine: DATE - Data di fine dell'assistenza

### 3.2) Tabella delle relazioni

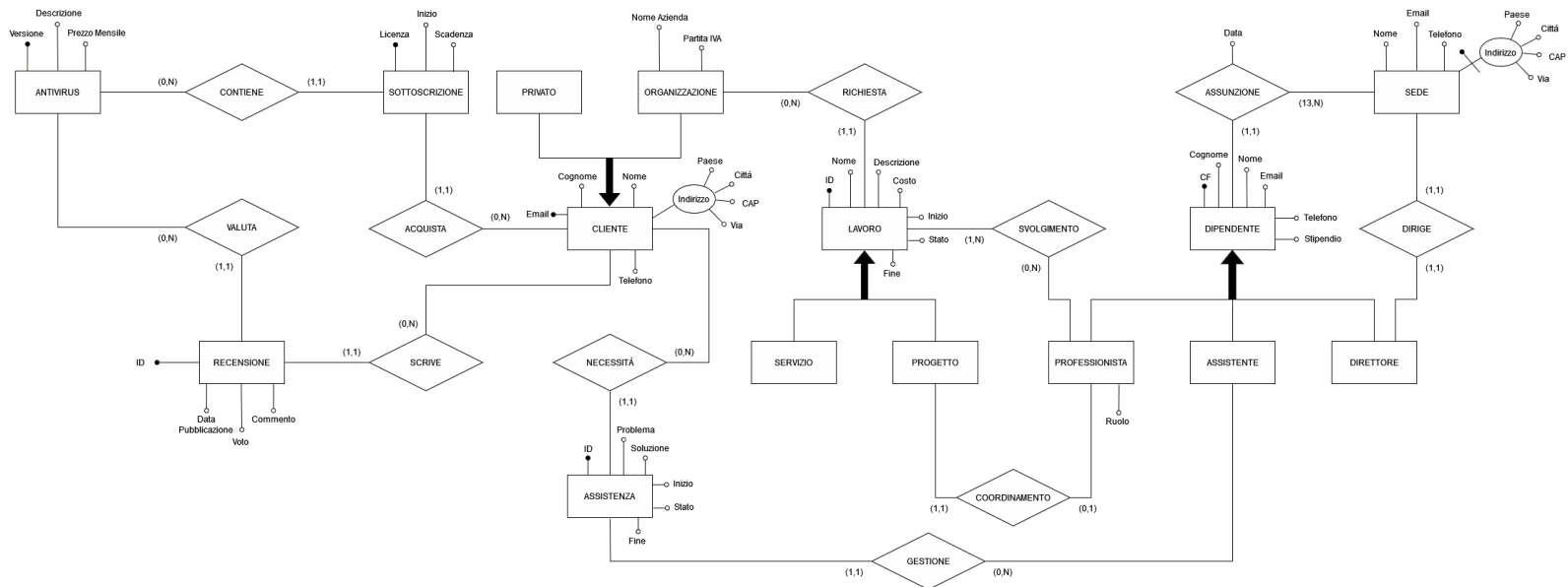
Relazione	Entità Coinvolte	Descrizione	Attributi
Acquisita	Cliente (0,N) Sottoscrizione (1,1)	Un cliente può acquistare più sottoscrizioni all'antivirus o non acquistarne nessuna. Una sottoscrizione si riferisce ad un solo cliente.	
Contiene	Sottoscrizione (1,1) Antivirus (0,N)	Una sottoscrizione si riferisce ad uno specifico antivirus. Un antivirus può essere sottoscritto più volte o non esserlo affatto.	
Scrive	Cliente (0,N) Recensione (1,1)	Un cliente può decidere di scrivere recensioni riguardanti l'antivirus. Una specifica recensione è scritta solamente da un cliente.	
Valuta	Recensione (1,1) Antivirus (0,N)	Una recensione valuta un antivirus nello specifico. Un antivirus può essere valutato da più recensioni o non esserlo affatto.	
Assunzione	Sede (13,N) Dipendente (1,1)	Una sede deve avere minimo 13 dipendenti e può assumerne quanti ne preferisce. Ogni dipendente è assunto da una singola sede.	Data: DATE - Data dell'assunzione
Dirige	Direttore (1,1) Sede(1,1)	Il direttore dirige una specifica sede. La sede è diretta da un solo direttore.	
Richiesta	Organizzazione (0,N) Lavoro (1,1)	Un'organizzazione può richiedere più lavori all'azienda. Uno specifico lavoro è richiesto da una sola organizzazione.	
Svolgimento	Professionista (0,N) Lavoro (1,N)	Un professionista può svolgere più lavori o non svolgerne nessuno. Un lavoro ha almeno un professionista che lo svolge o più di uno nel caso di un progetto.	
Coordinamento	Professionista (0,1) Progetto (1,1)	Un professionista può essere scelto per coordinare un progetto. Un progetto deve essere coordinato da un professionista.	
Necessità	Cliente (0,N) Assistenza (1,1)	Un cliente può avere necessità di assistenza. Una specifica richiesta di assistenza è associata ad un singolo cliente.	

Gestione	Assistente (0,N) Assistenza (1,1)	Un assistente può gestire più richieste di assistenza o non gestirne nessuna. Una specifica richiesta è gestita da un singolo assistente.	
----------	--------------------------------------	--	--

#### Vincoli non rappresentabili tramite schema E-R:

- Una **Recensione** può avere un voto compreso solamente tra 0 e 5
- Lo stato di un **Lavoro** o di un'**Assistenza** è rappresentato solamente da: Pianificato, In Corso, Completato, Annullato

### 3.3) Schema concettuale



## 4) Progettazione Logica

### 4.1) Analisi delle ridondanze

Una possibile ridondanza può essere trovata nella relazione tra Sede e Direttore, in quanto quest'ultimo dirige per forza la sede presso la quale è assunto, rivelando così la relazione "Dirige" inutilmente ridondante e di conseguenza rimossa.

## 4.2) Eliminazione delle generalizzazioni

**Cliente:** Le entità figlie Privato e Organizzazione vengono accorpate nell'entità padre Cliente siccome rappresentano lo stesso concetto. Vengono aggiunti gli attributi "Nome Azienda" e "Partita IVA", i quali precedentemente appartenevano ad Organizzazione, mentre ora sono opzionali e permettono di identificare facilmente la tipologia di cliente senza aggiungere ulteriori attributi proprio poiché non verranno mai fornite questo tipo di informazioni da un cliente privato. La relazione "Richiesta" ora coinvolge le entità Cliente (0,N) a Lavoro (1,1).

**Lavoro:** Le entità figlie Servizio e Progetto, poiché rappresentano un concetto simile, vengono accorpate nell'entità padre Lavoro. L'unica relazione che non è in comune, ovvero Coordinamento, viene risolta aggiungendo l'attributo "Coordinatore" nell'entità Lavoro. Siccome ogni progetto deve avere obbligatoriamente un coordinatore, questo nuovo attributo svolgerà anche la funzione di identificare se uno specifico lavoro è un servizio o un progetto.

**Dipendente:** L'entità padre Dipendente viene accorpata nelle entità figlie Professionista, Assistente e Direttore in quanto rappresentano concetti diversi, ognuno con le proprie relazioni distinte tra loro. L'unica relazione in comune, ovvero Assunzione, viene ricreata per ognuna delle nuove entità, coinvolgendo quindi:

Professionista (1,1) a Sede (10,N)

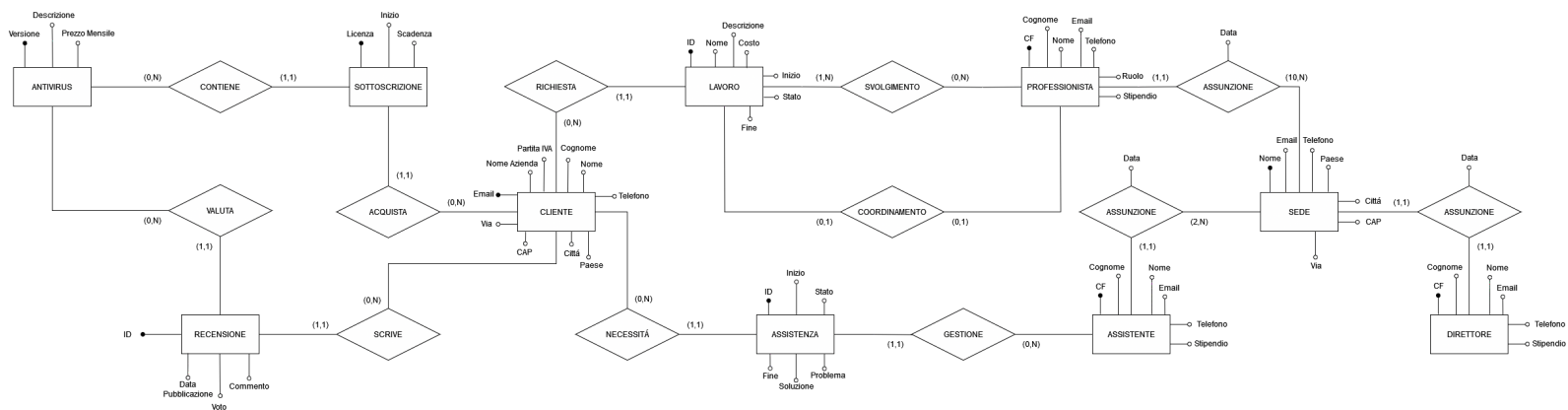
Assistente (1,1) a Sede (2,N)

Direttore (1,1) a Sede (1,1).

## 4.3) Scelta degli identificatori primari

Per l'entità Sede era stato scelto di utilizzare l'attributo composto "Indirizzo" (Paese, Città, CAP, Via) come identificatore primario, ma questo avrebbe portato all'aggiunta di 4 attributi sulle entità Professionista, Assistente e Direttore. Si è visto invece che anche l'attributo "Nome" di Sede è identificativo, poiché non possono esistere due sedi con lo stesso nome, di conseguenza è stato deciso di utilizzare quest'ultimo.

## 4.4) Schema E-R ristrutturato





## 4.5) Creazione delle tabelle

*Legenda: (Primary Key, Attributo semplice, #Chiave esterna)*

**Cliente** (Email, Nome\_Azienda, Partita\_IVA, Cognome, Nome, Telefono, Paese, Città, CAP, Via)

**Antivirus** (Versione, Descrizione, Prezzo\_Mensile)

**Sottoscrizione** (#Cliente, #Antivirus, Licenza, Inizio, Scadenza)

Cliente -> Cliente.Email

Antivirus -> Antivirus.Versione

**Recensione** (ID, #Cliente, #Antivirus, Data\_Pubblicazione, Voto, Commento)

Cliente -> Cliente.Email

Antivirus -> Antivirus.Versione

**Sede** (Nome, Email, Telefono, Paese, Città, Cap, Via)

**Direttore** (CF, Cognome, Nome, Email, Telefono, Stipendio, #Sede, Data\_Assunzione)

Sede -> Sede.Nome

**Professionista** (CF, Cognome, Nome, Email, Telefono, Ruolo, Stipendio, #Sede, Data\_Assunzione)

Sede -> Sede.Nome

**Lavoro** (ID, #Cliente, Nome, Descrizione, Costo, Inizio, Stato, Fine, #Coordinatore)

Cliente -> Cliente.Email

Coordinatore -> Professionista.CF

**Svolgimento** (#Professionista, #Lavoro)

Professionista -> Professionista.CF

Lavoro -> Lavoro.ID

**Assistente** (CF, Cognome, Nome, Email, Telefono, Stipendio, #Sede, Data\_Assunzione)

Sede -> Sede.Nome

**Assistenza** (ID, #Cliente, Inizio, Stato, Fine, Problema, Soluzione, #Assistente)

Cliente -> Cliente.Email

Assistente -> Assistente.CF

## 5) Definizione delle Query e degli Indici associati

### 5.1) Queries

1. Elenco delle e-mail dei clienti che hanno speso più di  $n$  euro (in questo caso 50) in antivirus e che non hanno lasciato recensioni.

```
SELECT c.Email FROM Cliente c JOIN Sottoscrizione s ON c.Email = s.Cliente
JOIN Antivirus a ON s.Antivirus = a.Versione
LEFT JOIN Recensione r ON c.Email = r.Cliente AND a.Versione = r.Antivirus
GROUP BY c.email, c.nome, c.cognome, a.prezzo_mensile, r.id
HAVING a.Prezzo_Mensile * 12 > 50 AND r.ID IS NULL;
```

```
*-----*
| email |
*-----*
| claudia.mueller@swissbanking.com |
| emma.smith@healthcare.com |
| francesca.gagnon@outlook.com |
| jessica.nguyen@healthgroup.com |
| luca.muller@mail.com |
| maria.gonzalez@spahealth.com |
| maria.rossi@libero.it |
| roberto.rossi@biotechresearch.com |
*-----*
```

2. Elenco in ordine decrescente degli antivirus con il maggior numero di sottoscrizioni e la media dei voti ricevuti.

```
SELECT a.Versione AS Antivirus, COUNT(DISTINCT s.Cliente) AS NumeroSottoscrizioni,
ROUND(AVG(r.Voto),1) AS MediaVoti
FROM Antivirus a
LEFT JOIN Sottoscrizione s ON a.Versione = s.Antivirus LEFT JOIN Recensione r
ON a.Versione = r.Antivirus
GROUP BY a.Versione
ORDER BY NumeroSottoscrizioni DESC;
```

```
*-----*
| antivirus | numerosottoscrizioni | mediavoti |
*-----*
| Business | 5 | 3.3 |
| Enterprise | 3 | 5.0 |
| Premium | 3 | 1.0 |
| Standard | 2 | 4.0 |
| Ultimate | 2 | 4.0 |
| Plus | 1 | 3.0 |
*-----*
```

3. Elenco in ordine decrescente dei nomi e cognomi dei professionisti che hanno svolto il ruolo di coordinatore, ordinati in base al numero di progetti.

```
SELECT P.Cognome, P.Nome, COUNT(L.ID) AS NumeroProgetti
FROM Professionista P
JOIN Lavoro L ON P.CF = L.Coordinatore
GROUP BY P.Cognome, P.Nome
ORDER BY NumeroProgetti DESC;
```

cognome	nome	numeroprogetti
Williams	Emma	2
Esposito	Giulia	1

4. Lista di denaro speso ogni anno da ogni sede per retribuire tutti i rispettivi dipendenti.

```
SELECT S.Nome AS Sede, SUM(P.Stipendio) AS TotaleStipendi
FROM (SELECT CF, Sede, Stipendio FROM Professionista
UNION ALL
SELECT CF, Sede, Stipendio FROM Direttore
UNION ALL
SELECT CF, Sede, Stipendio FROM Assistente) P JOIN Sede S ON P.Sede = S.Nome
GROUP BY S.Nome
ORDER BY S.Nome;
```

sede	totalestipendi
Berlin Office	248000
London Office	262000
Madrid Office	320000
New York Office	406000
Rome Office	280000
Zurich Office	382000

5. Elenco degli assistenti con il rispettivo numero di assistenze gestite, separate in base al tipo di cliente (organizzazione o privato)

```
SELECT A.Nome AS NomeAssistente, A.cognome AS CognomeAssistente
SUM(CASE WHEN C.Partita_IVA IS NOT NULL THEN 1 ELSE 0 END) AS NumeroAssistenzeOrganizzazioni,
SUM(CASE WHEN C.Partita_IVA IS NULL THEN 1 ELSE 0 END) AS NumeroAssistenzePrivati
FROM Assistente A
JOIN Assistenza Asst ON A.CF = Asst.Assistente
JOIN Cliente C ON Asst.Cliente = C.Email
GROUP BY A.Nome, A.Cognome;
```

nomeassistente	cognomeassistente	numeroassistenzeorganizzazioni	numeroassistenzeprivati
Alessio	Giaretta	2	0
James	Stone	0	2
Elena	Magro	0	2
Giulia	Rossi	2	0

6. Soldi guadagnati nell'anno  $n$  (in questo caso 2023) dai contratti di lavoro e dalla vendita di antivirus.

```
SELECT
  (SELECT SUM(ALL costo) as GuadagnoProgetti
   FROM lavoro l
   WHERE EXTRACT(YEAR FROM(l.fine))='2023'),
  (SELECT SUM(prezzo_mensile * ((s.scadenza - s.inizio) / 30)) as GuadagnoAntivirus
   FROM sottoscrizione s
   JOIN antivirus a ON s.antivirus = a.versione
   WHERE EXTRACT(YEAR FROM(s.inizio)) = '2023');
```

guadagnoprogetti	guadagnoantivirus
20500	23417.57

7. Elenco in ordine decrescente dei nomi e cognomi dei direttori con le rispettive sedi che hanno guadagnato di più dalle richieste di lavoro.

```
SELECT d.nome, d.cognome, se.nome, SUM(l.costo) as guadagno
FROM lavoro l,svolgimento sv, sede se, direttore d, professionista p
WHERE l.id=sv.lavoro
AND sv.professionista=p.cf
AND p.sede=se.nome
AND d.sede=se.nome
GROUP BY d.nome, d.cognome, se.nome
ORDER BY guadagno DESC;
```

nome	cognome	nome	guadagno
Sarah	Johnson	New York Office	18500
Rodrigo	García	Madrid Office	15000
Elizabeth	Wallace	London Office	12500
Anna	Ferrari	Rome Office	11000
Manuel	Burger	Zurich Office	6000
William	Franklin	Berlin Office	5500

## 5.2) Indici

La tabella Cliente é di notevoli dimensioni, contiene numerose tuple e la frequenza di lettura é elevata: si pensi, per esempio, ai suoi utilizzi nelle recensioni, nei lavori, nelle assistenze e nelle sottoscrizioni. Di conseguenza si é scelto di creare l'indice sulla tabella Cliente:

```
CREATE INDEX idx_EmailCliente ON Cliente(Email);
```

## 6) Applicazione Software

L'applicazione software è stata sviluppata utilizzando il linguaggio C++ e fa uso della libreria libpq per interrogare il database utilizzando le query indicate sopra.

Per compilare il file main.cpp è necessario utilizzare librerie e header forniti da PostgreSQL. Questi file sono presenti all'interno della cartella "dependencies".

Il comando da eseguire per la compilazione del codice è il seguente:

```
g++ main.cpp -o codice -I /usr/include/postgresql -lpq
```

L'esecuzione di questo comando genererà il file codice.exe.

Il programma stampa a schermo un menù dal quale è possibile scegliere la query da eseguire per interrogare il database, per le query parametriche è possibile scegliere direttamente dal programma il parametro da inserire.

### 6.1) Funzioni utilizzate

```
PGconn *connect();
```

Consente al programma di interfacciarsi con il database utilizzando le credenziali definite all'inizio del codice.

```
void checkResults(PGresult *res, const PGconn *conn);
```

Controlla i risultati restituiti dalle query eseguite, in caso di errore il programma ferma la sua esecuzione e stampa a schermo un messaggio di errore.

```
void fill_query(string *QueryArray);
```

Riempie un array di stringhe inizialmente vuoto con le query per interrogare il database.

```
PGresult *ParamQuery(PGconn *conn, string QueryArray[], int QueryNumber);
```

Permette l'esecuzione di query parametriche facendo inserire all'utente il parametro.

```
void StampaRisultati(PGresult *res);
```

Stampa i risultati della query eseguita.

```
void StampaMenu();
```

Stampa a schermo il menù dal quale l'utente può scegliere quale query eseguire.

```
int main(int argc, char **argv)
```

Funzione main che gestisce l'interrogazione del database, in base alla query scelta dall'utente richiama le funzioni opportune.