

Documentazione database
Sistema di gestione di personale e progetti all'interno di
un'azienda

Roberto Ingenito

Simone Ingenito

Lorenzo Sequino

5 gennaio 2023



UNIVERSITÀ DEGLI STUDI
DI NAPOLI FEDERICO II

Indice

1	Progettazione concettuale	3
1.1	Analisi dei requisiti	3
1.2	Schema concettuale	4
1.3	Dizionario delle entità	5
1.4	Dizionario delle associazioni	7
2	Ristrutturazione del modello concettuale	8
2.1	Analisi delle ridondanze	8
2.2	Eliminazione degli attributi multivalore	8
2.3	Eliminazione degli attributi composti	8
2.4	Analisi delle generalizzazioni	8
2.5	Identificazioni chiavi primarie	8
2.6	Schema ristrutturato UML	9
2.7	Schema ristrutturato ER	9
3	Traduzione al modello logico	10
3.1	Mapping associazioni	10
3.1.1	Associazioni 1-N	10
3.1.2	Associazioni N-N	10
3.2	Modello logico	11
4	Progettazione fisica	12
4.1	Creazione del database	12
4.2	Creazione dello schema	12
4.3	Attivazione estensioni	12
4.4	Gestione ruoli e permessi	12
4.5	Creazione domini	12
4.6	Creazione tabelle	13
4.7	Vincoli	13
4.7.1	Dizionario itra-relazionali	13
4.7.2	Dizionario inter-relazionai	13
4.8	Trigger e trigger function	13
4.9	Procedure e funzioni	13

1 Progettazione concettuale

1.1 Analisi dei requisiti

La base dati si deve occupare della gestione del personale di un'azienda:
la figura centrale di tutto lo schema è l'impiegato.

La prima classificazione di un generico impiegato viene fatta tra:

- impiegato assunto regolarmente
- impiegato assunto esclusivamente per lavorare ad un progetto

Gli impiegati assunti regolarmente, vengono assunti con un determinato ruolo e stipendio, inoltre hanno diverse specializzazioni in base sia al merito sia agli anni trascorsi all'interno dell'azienda.

Dal momento in cui viene assunto, l'impiegato diventa automaticamente un junior. Trascorsi 3 anni diventa middle, trascorsi altri 4 diventa senior. Inoltre, qualunque siano gli anni trascorsi all'interno dell'azienda, può essere promosso e diventare manager.

Si tiene traccia di tutti gli scatti di carriera all'interno dell'entità *career_log* dove sono memorizzati:

- ruolo precedente
- ruolo successivo
- data dello scatto

L'azienda è divisa in varie sedi dove sono presenti diversi laboratori.

Ogni impiegato afferisce ad un unico laboratorio.

Un impiegato può aver anche lavorato in più laboratori ma in periodi diversi.

Al momento dell'assunzione l'impiegato non ha ancora una collocazione ma viene stabilita successivamente.

Ad un impiegato senior può essere assegnata la gestione di laboratori e/o di progetti.

Ad ogni laboratorio è assegnato un manager scientifico che è un impiegato senior.

L'amministratore, a cui è affidata la gestione del database, si occupa di:

- inserire gli impiegati assunti
- monitorare gli scatti di carriera
- promuovere gli impiegati meritevoli a manager
- creare progetti e affidare la supervisione ad un manager e la gestione ad un referente scientifico

Per ogni progetto viene stabilita una data di inizio, una *deadline* per la conclusione del progetto e dei fondi.

Inoltre, si tiene traccia dell'effettiva data in cui il progetto è terminato *end_date*; può accadere che il progetto si protragga oltre la deadline

Ad un progetto possono prendere parte massimo 3 laboratori contemporaneamente.

Ogni impiegato che afferisce ad un laboratorio prende parte automaticamente a tutti i progetti a cui quel laboratorio sta lavorando.

Il manager scientifico di un laboratorio decide a quali progetti partecipare e abbandonare, e richiede attrezzatura.

Le richieste vengono memorizzate nell'entità *equipment_request*. Queste vengono valutate dal manager e dal referente scientifico del progetto a cui sono state fatte, i quali decidono, in base ai fondi del progetto, se soddisfare le richieste, con il vincolo che il costo totale delle attrezzature non può superare il 50% dei fondi del progetto.

Quando viene acquistata dell'attrezzatura, si salva l'acquisto nell'entità *purchase* che tiene traccia della data e del prezzo d'acquisto.

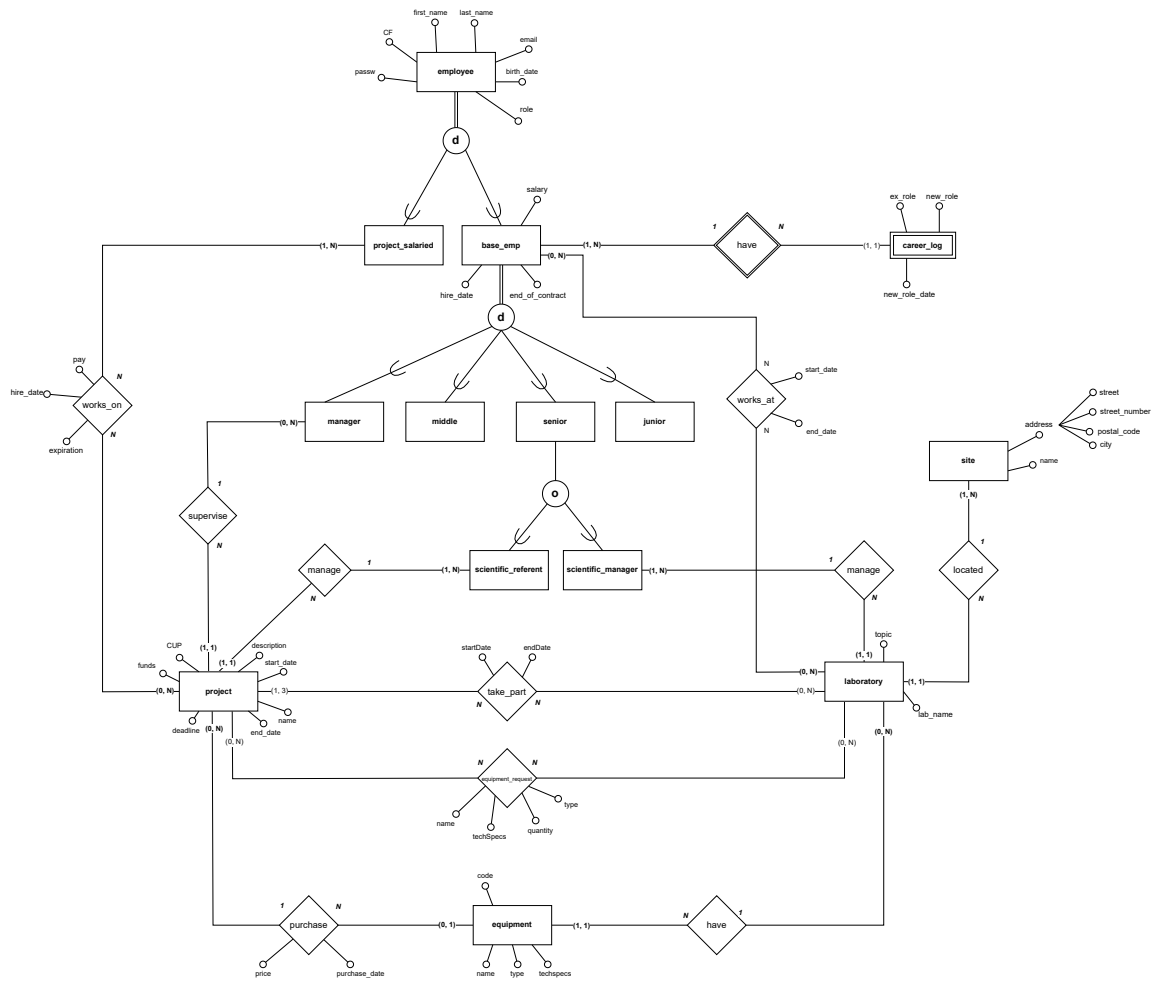
Ogni attrezzatura di ogni laboratorio viene memorizzata nell'entità *equipment*.

Un laboratorio può possedere già dell'attrezzatura (acquistata precedentemente da altri progetti) e riceverne altra tramite richieste a progetti. Di conseguenza l'attrezzatura acquistata in un laboratorio, rimane anche dopo la fine del progetto.

Con il restante 50% dei fondi, il manager e il referente scientifico di un progetto possono assumere del personale che viene pagato per lavorare esclusivamente per quel progetto, e non viene considerato come impiegato regolare.

Lo stesso impiegato può essere assunto più volte (anche per lo stesso progetto ma in date differenti), per questo motivo ognuno di questi impiegati non viene rimosso una volta terminato il contratto. La relazione *works_on* infatti, tiene traccia di tutte le volte in cui un impiegato è stato assunto per lavorare ai progetti.

1.2 Schema concettuale



1.3 Dizionario delle entità

Entità	Descrizione	Attributi
employee	Generico impiegato dell'azienda	cf : Codice fiscale, informazione utile all'azienda first_name last_name birth_date role : Professione (sviluppatore, designer, ...) email : Email necessaria per l'accesso al gestionale passw : Password necessaria per l'accesso al gestionale
base_emp	Specializzazione di <i>employee</i> . Assunto a tempo indeterminato.	salary : Stipendio hire_date : Data assunzione
career_log	Entità che memorizza gli scatti di carriera degli impiegati <i>base_emp</i>	ex_role : Ruolo precedente (stringa vuota quando è assunto) new_role : Ruolo successivo (stringa vuota quando è licenziato) new_role_date : Data scatto di carriera
project_salaried	Specializzazione di <i>employee</i> . Assunto per lavorare esclusivamente ad un progetto.	
junior	Specializzazione di <i>base_emp</i> . Impiegato che lavora da meno di 3 anni.	
middle	Specializzazione di <i>base_emp</i> . Impiegato che lavora dai 4 ai 7 anni.	
senior	Specializzazione di <i>base_emp</i> . Impiegato che lavora da più di 7 anni.	
manager	Specializzazione di <i>base_emp</i> . Impiegato che ha ricevuto una promozione in base al merito.	
scientific_referent	Specializzazione di <i>senior</i> . Impiegato a cui è stata affidata la gestione di un progetto.	
scientific_manager	Specializzazione di <i>senior</i> . Impiegato a cui è stata affidata la gestione di un laboratorio.	

project	Progetto a cui prendono parte i laboratori e i relativi impiegati.	CUP: Codice Univoco Progett name: Nome del progetto description: Descrizione del progetto start_date: Data di inizio del progetto end_date: Data di effettiva fine del progetto deadline: Data prevista per la fine del progetto funds: Somma di denaro disponibile per l'acquisto di attrezzature e personale
laboratory	Luogo all'interno dell'azienda nel quale gli impiegati lavorano ai progetti.	topic: Campo di studi del laboratorio lab_name: Nome del laboratorio
site	Sede dell'azienda in cui possono essere presente i laboratori.	name: Nome della sede address: Indirizzo <ul style="list-style-type: none"> • street: Via • street_number: Numero civico • postal_code: CAP • city: Città
equipment	Attrezzatura presente all'interno di un laboratorio	code: Codice del prodotto name: Nome del prodotto type: Tipo di prodotto (computer, microscopio, ...) techspecs: Specifiche tecniche del prodotto

1.4 Dizionario delle associazioni

Associazioni	Descrizione	Attributi
works_on	Associazione tra <i>project_salaried</i> e <i>project</i> multi-a-molti	pay : Paga stabilita per lavorare a quel progetto hire_date : Data di assunzione expiration : Data di fine contratto
works_at	Associazione tra <i>base_emp</i> e <i>laboratory</i> multi-a-molti .	start_date : Data di assegnazione di un impiegato ad un laboratorio end_date : Data di fine lavoro
supervise	Associazione tra <i>manager</i> e <i>project</i> uno-a-molti .	
manage	Associazione tra <i>scientific_referent</i> e <i>project</i> uno-a-molti .	
manage	Associazione tra <i>scientific_manager</i> e <i>laboratory</i> uno-a-molti .	
take_part	Associazione tra <i>project</i> e <i>laboratory</i> multi-a-molti .	start_date : Data in cui un laboratorio inizia a lavorare ad un progetto end_date : Data di fine lavoro
purchase	Associazione tra <i>project</i> e <i>equipment</i> uno-a-molti .	purchase_date : Data di acquisto price : Prezzo dell'attrezzatura
equipment_request	Associazione tra <i>project</i> e <i>laboratory</i> multi-a-molti .	
have	Associazione tra <i>equipment</i> e <i>laboratory</i> uno-a-molti .	
have	Associazione tra <i>base_emp</i> e <i>career_log</i> uno-a-molti .	
located	Associazione tra <i>site</i> e <i>laboratory</i> uno-a-molti .	

2 Ristrutturazione del modello concettuale

2.1 Analisi delle ridondanze

È presente una ridondanza nel campo *hire_date* di *base_emp* in quanto è possibile calcolarla dall'entità associata *career_log*

2.2 Eliminazione degli attributi multivalore

Non sono presenti attributi multivalore

2.3 Eliminazione degli attributi composti

L'attributo *address* di *site* è un attributo composto.

Siccome l'indirizzo è unico per ogni sede, si è deciso di accorpare gli attributi di *address* in *site*.

L'attributo *tech_specs* di *equipment* potrebbe essere scomposto in più attributi ma in questo caso non è necessario ai fini della traccia, quindi è più comodo memorizzarlo come un unico campo stringa.

2.4 Analisi delle generalizzazioni

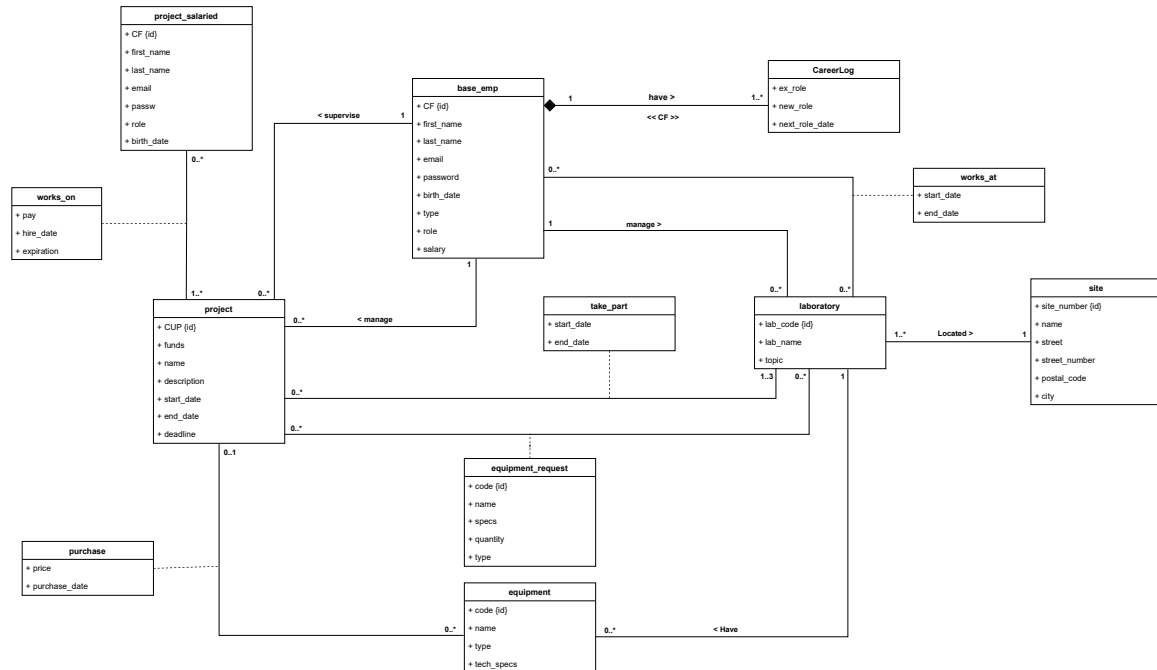
Procediamo all'eliminazione delle generalizzazioni partendo dal basso:

1. Accorpamento di *scientific_referent* e *scientific_manager* in *senior* in quanto non avendo attributi non è possibile inserire valori nulli che occupano spazio inutile.
Fatta tale scelta bisogna cambiare la cardinalità da $(1, n)$ a $(0, n)$ in entrambe le associazioni, poiché un *senior* può anche non essere nessuno dei due o uno soltanto (overlapping parziale)
2. Accorpamento di *junior*, *middle*, *senior*, *manager* all'interno di *base_emp* aggiungendo un attributo che ne specifica il tipo.
Notare che questa scelta aggiunge una **ridondanza** poiché il tipo è reperibile effettuando un'interrogazione in *career_log*. Nonostante ciò è più efficiente e immediato eseguire determinate operazioni avendo quest'attributo disponibile direttamente in *base_emp*
3. Accorpamento di *employee* all'interno di *project_salaried* e *base_emp* in quanto la generalizzazione è totale.

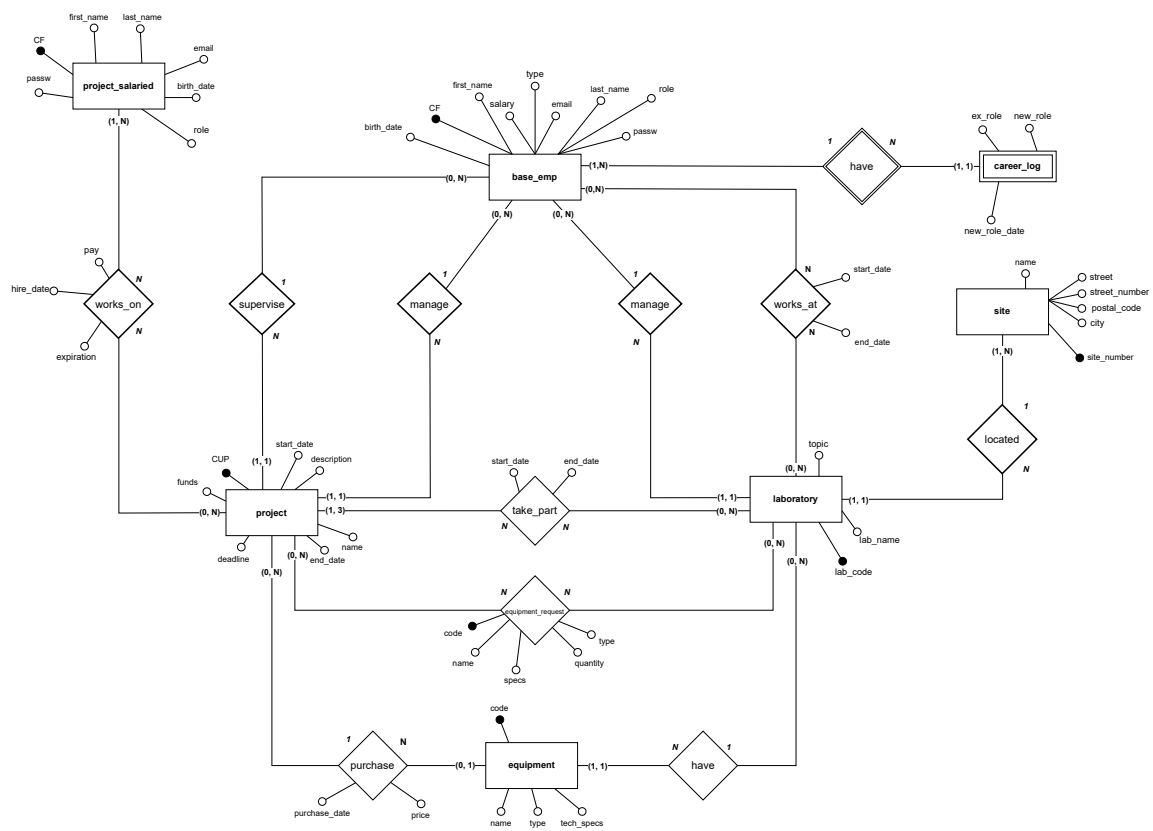
2.5 Identificazioni chiavi primarie

- In *base_emp* abbiamo l'attributo *CF* (Codice Fiscale), una stringa di 16 caratteri
- In *project_salaried* abbiamo l'attributo *CF* (Codice Fiscale), una stringa di 16 caratteri
- *project* è identificato dal *CUP* (Codice Univoco Progetto)
- *equipment* è identificato da *code* che è un codice generato randomicamente
- *equipment_request* è identificato da *code* che è un codice generato randomicamente
- *site* è identificato da un nuovo attributo *site_number* che si autoincrementa
- *laboratory* è identificato da un nuovo attributo *lab_code* che si autoincrementa

2.6 Schema ristrutturato UML



2.7 Schema ristrutturato ER



3 Traduzione al modello logico

3.1 Mapping associazioni

3.1.1 Associazioni 1-N

- *Site - Located - Laboratory*: inserimento chiave di *Site* in *Laboratory* come chiave esterna
- *base_emp - Supervise - Project*: inserimento chiave di *base_emp* all'interno di *Project* come chiave esterna
- *base_emp - Manage - Project*: inserimento chiave di *base_emp* all'interno di *Project* come chiave esterna
- *base_emp - Manage - Laboratory*: inserimento di chiave di *base_emp* all'interno di *Laboratory* come chiave esterna
- *Laboratory - Have - Equipment*: inserimento di chiave di *Laboratory* in *Equipment* come chiave esterna
- *Project - Purchase - Equipment*: *Equipment* ha una partecipazione parziale, si procede come N-N
- *base_emp - Have - career_log*: relazione identificante, inserimento di chiave primaria di *base_emp* in *career_log* come chiave esterna

3.1.2 Associazioni N-N

Per ognuna di queste relazioni, si inseriscono le chiavi delle due associazioni come chiavi esterne.

Associazione	Relazione	Associazione
project_salaried	works_on	project
project	take_part	laboratory
project	equipment_request	laboratory
base_emp	works_at	laboratory

3.2 Modello logico

Gli attributi sottolineati sono chiavi primarie,
gli attributi con un asterisco * alla fine sono chiavi esterne

<i>base_emp</i>	(<u>CF</u> , first_name, last_name, email, passw, birth_date, type, role, salary)
<i>career_log</i>	(lab_code, lab_name, topic, cf_scientific_manager*, site_number*) cf_scientific_manager \mapsto base_emp.CF site_number \mapsto site.site_number
<i>laboratory</i>	(<u>lab_code</u> , lab_name, topic, cf_scientific_manager*, site_number*) cf_scientific_manager \mapsto base_emp.CF site_number \mapsto site.site_number
<i>take_part</i>	(start_date, end_date, CUP*, lab_code*) CUP \mapsto project.CUP lab_code \mapsto laboratory.lab_code
<i>project_salaried</i>	(<u>CF</u> , first_name, last_name, email, passw, birth_date, role)
<i>works_on</i>	(pay, hire_date, expiration, CF*, CUP*) CF \mapsto project_salaried.CF CUP \mapsto project.CUP
<i>site</i>	(<u>site_number</u> , name, street, street_number, postal_code, city)
<i>equipment</i>	(<u>code</u> , name, type, tech_specs, lab_code*) lab_code \mapsto laboratory.lab_code
<i>purchase</i>	(purchase_date, price, CUP*, equipment_code*) CUP \mapsto project.CUP equipment_code \mapsto equipment.code
<i>works_at</i>	(start_date, end_date, cf_base_emp*, lab_code*) cf_base_emp \mapsto base_emp.CF lab_code \mapsto laboratory.lab_code
<i>equipment_request</i>	(<u>code</u> , type, name, specs, quantity, CUP*, lab_code*) CUP \mapsto project.CUP lab_code \mapsto laboratory.lab_code
<i>project</i>	(<u>CUP</u> , funds, name, description, start_date, end_date, deadline, cf_manager*, cf_scientific_referent*) cf_manager \mapsto base_emp.CF cf_scientific_referent \mapsto base_emp.CF

4 Progettazione fisica

4.1 Creazione del database

4.2 Creazione dello schema

4.3 Attivazione estensioni

Estensione utilizzata per la cifratura delle password, usata per l'autenticazione lato client.

```
CREATE EXTENSION IF NOT EXISTS "pgcrypto";
```

Estensione utilizzata per generare codici univoci causali.

```
CREATE EXTENSION IF NOT EXISTS "uuid-oss";
```

4.4 Gestione ruoli e permessi

4.5 Creazione domini

```
CREATE DOMAIN cf_type AS VARCHAR(16)
CONSTRAINT cf_type_length CHECK (length(VALUE) = 16);
```

```
CREATE DOMAIN cup_type AS VARCHAR(15)
CONSTRAINT check_cup_type CHECK ( length(VALUE) = 15);
```

```
CREATE DOMAIN emp_type AS VARCHAR(10)
CONSTRAINT emp_type_check CHECK
    (VALUE IN ('', 'junior', 'middle', 'senior', 'manager')
    AND VALUE IS NOT NULL);
```

```
CREATE DOMAIN equipment_name_type
AS VARCHAR(30);
```

```
CREATE DOMAIN name_type AS VARCHAR(30);
```

```
CREATE DOMAIN password_type AS VARCHAR(100);
```

```
CREATE DOMAIN salary_type AS DECIMAL(8,2);
```

4.6 Creazione tabelle

4.7 Vincoli

4.7.1 Dizionario itra-relazionali

4.7.2 Dizionario inter-relazionai

4.8 Trigger e trigger function

4.9 Procedure e funzioni