|  |  |
| --- | --- |
| Rollo – ride your way  Gruppo 1: Duluta, Manfo, Marcolin, Fedorenco | Documentazione tecnica del sistema informativo per la gestione di un servizio di bike sharing basato su una piattaforma web. Viene descritto il processo di progettazione della base di dati, dall'analisi del problema alla normalizzazione dello schema, fino all’implementazione in SQL e alla definizione delle query per l’interazione con l’applicazione. |

Rollo – ride your way

Documentazione tecnica per la parte informatica

# **1.** **Analisi del problema**

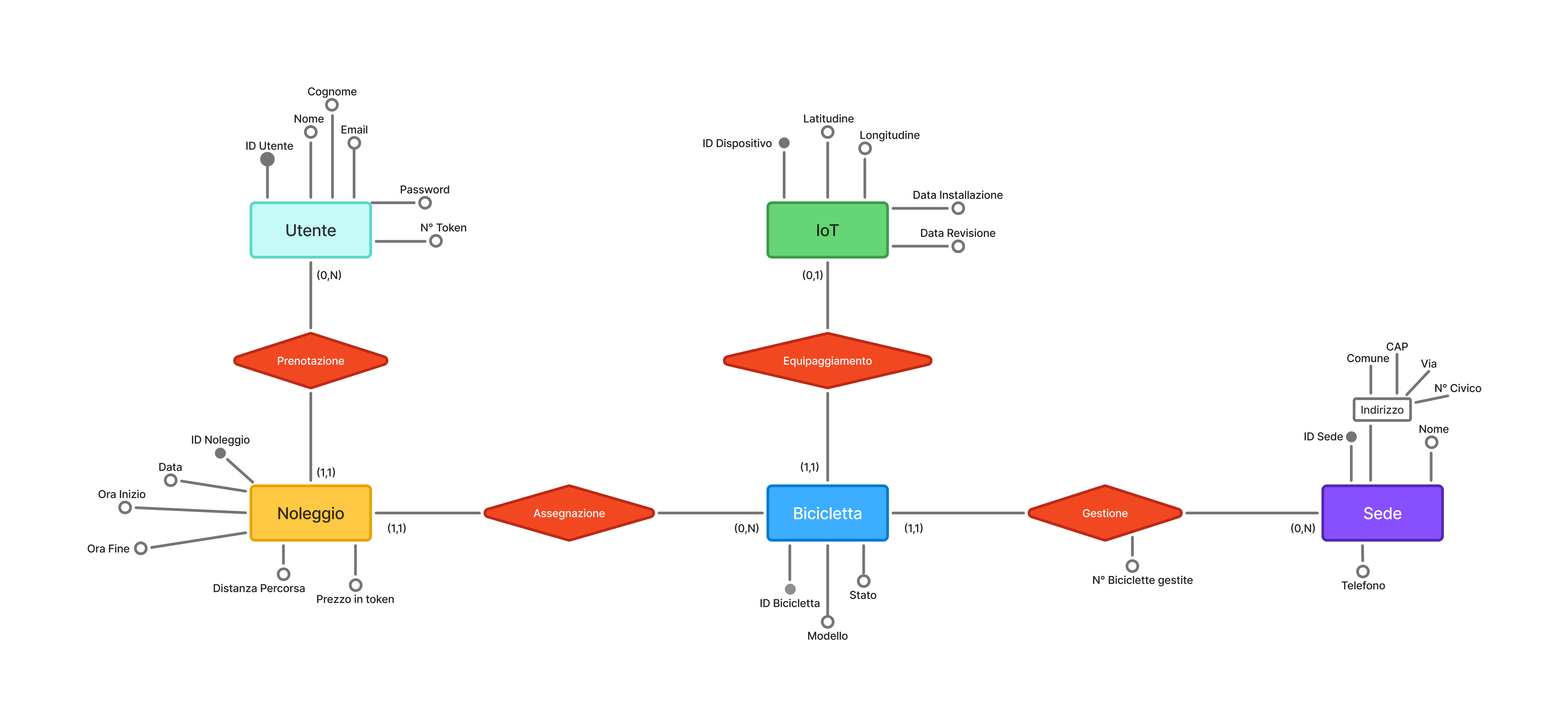
Il progetto "Rollo - Ride your way" nasce con l’obiettivo di realizzare una web app per il noleggio e la gestione di biciclette.   
La base di dati che si appoggia alla web app deve permettere la registrazione e gestione degli utenti, la localizzazione delle sedi (che gestiscono un certo numero di biciclette e si occupano della loro revisione), il controllo dei dispositivi IoT installati su ogni bicicletta e la tracciabilità dei noleggi effettuati.

Ogni noleggio sarà registrato con informazioni relative a:

* data e orario di inizio/fine
* distanza percorsa
* costo (in token)
* bicicletta utilizzata
* utente che ha effettuato il noleggio

Si è scelto di utilizzare dei token per i pagamenti in quanto la connessione ad un servizio per i pagamenti bancari come “Nexi” porterebbe con sé delle complicanze realizzative rilevanti, come ad esempio l’obbligo di registrazione dell’attività all’Agenzia delle Entrate. I token saranno guadagnabili attraverso il “Bonus di Benvenuto”, il bonus “Droppa una story con Rollo” (un contenuto sui social che promuova il brand) e le eventuali pubblicità facoltative, se si riuscirà ad implementarle.

# **2. Proposta iniziale di schema E-R**



[Documentazione nel file allegato: Documentazione E-R.pdf](Documentazione%20E-R.pdf)

# **3. Motivazione delle scelte iniziali**

Lo schema iniziale è stato progettato tenendo conto dei seguenti criteri:

* Separazione chiara tra entitàdistinte: Utenti, Biciclette, Noleggi, Sedi e IoT.
* Ogni entità ha un suo identificatore singolo interno, sufficiente ad indivituare in modo univoco ogni sua istanza.

1. **Utente**Id\_Utente, nome e cognome, email, password e numero di token collegati all’account.  
   Un utente può prenotare uno o più noleggi.
2. **Noleggio**Id\_Noleggio, data, ora inizio e ora fine, distanza percorsa e prezzo in token.  
   Un noleggio è prenotato da un solo utente ed ha soltanto una bicicletta assegnata.
3. **Bicicletta**Id\_Bicicletta, modello e stato (se utilizzabile o occupata/in manutenzione).  
   Una bicicletta può essere assegnata ad uno o più noleggi. Equipaggia una sola scheda IoT ed è gestita da una sola Sede.
4. **Sede**

Id\_Sede, nome (può avere un nome in codice con cui è riconosciuta più facilmente tra i dipendenti), Indirizzo (**attributo composto** da comune, CAP, via e numero civico) e contatto telefonico.  
***Nota: l’associazione tra Bicicletta e Sede presenta un attributo che costituisce il numero di biciclette gestite dalla singola sede.***

1. **IoT**

Id\_Dispositivo, latitudine e logitudine, data dell’installazione e data della prossima revisione programmata.

Questa struttura riflette la realtà operativa di un sistema di bike sharing e consente una gestione flessibile ed espandibile del servizio.

# **4. Schema E-R ristrutturato**

Nella ristrutturazione dello schema E-R vengono eliminate le generalizzazioni (in questo caso non sono state previste) e gli attributi delle associazioni. Inoltre, gli attributi composti vengono separati (in attributi singoli) nel caso in cui non è efficace l’archiviazione di tutti i dati che ne fanno parte in una sola tupla.   
Vengono applicate le seguenti modifiche:

* L’attributo “N° Biciclette gestite” dell’associazione “Gestione” diventa attributo di “Sede”.
* L’attributo composto “Indirizzo” di “Sede” viene scomposto negli attributi “comune”, “cap”, “via” e “n° civico”.

# **5. Traduzione a schema logico e normalizzazione**

Per controllare se lo schema prodotto rispetta le regole della normalizzazione, ogni entità è stata convertita in una tabella con attributi atomici e chiavi primarie ben definite. I nomi delle entità diventano al plurale.

Utenti(id\_Utente, nome, cognome, email, password)

Noleggi (id\_Noleggio, data, ora\_Inizio, ora\_Fine, distanza\_Percorsa, prezzo, *utente*, *bicicletta*)

Biciclette (id\_Bicicletta, modello, *scheda\_IoT*, *sede\_appartenenza, stato*)

IoT (id\_Dispositivo, latitudine, longitudine, data\_Installazione, data\_Revisione)

Sedi (id\_Sede, nome, comune, cap, via, civico, num\_Biciclette\_Gestite, telefono)

Viene di seguito descritto il confronto con le principali forme normali studiate.

* **1NF** ✅**:** Tutti gli attributi sono atomici. Non ci sono valori multipli o campi ripetuti.
* **2NF** ✅**:** Ogni attributo non chiave è completamente dipendente dalla chiave primaria. Nessuna tabella ha chiavi composite.
* **3NF** ✅**:** Tutte le dipendenze funzionali sono dirette: non ci sono attributi che dipendono da altri attributi non-chiave.
* **Forma Normale di Boyce-Codd (FNBC)** ✅**:**   
  La FNBC è una forma normale più rigorosa della 3NF. Una tabella è in FNBC se per ogni dipendenza funzionale X→Y, X è una superchiave.

Nello schema utilizzato:

Tutte le dipendenze funzionali hanno come determinante una chiave o superchiave.

~ Lo schema è dunque **normalizzato in FNBC**.

[Documentazione nel file allegato: Documentazione - Schema Logico.pdf](Documentazione%20-%20Schema%20Logico.pdf)

# **6. Vincoli di integrità**

* **Chiavi primarie**
  + Utenti(id\_Utente)
  + Sedi(id\_Sede)
  + Biciclette(id\_Bicicletta)
  + IoT(id\_Dispositivo)
  + Noleggi(id\_Noleggio)
* **Chiavi esterne**
  + Noleggi.utente → Utenti.id\_Utente
  + Noleggi.bicicletta → Biciclette.id\_Bicicletta
  + Biciclette.scheda\_IoT → IoT.id\_Dispositivo
  + Biciclette.sede\_appartenenza → Sedi.id\_Sede
* **Vincoli sui dati**
  + prezzo ≥ 0
  + distanza\_Percorsa ≥ 0
  + ora\_Fine ≥ ora\_Inizio (se ora\_Fine è presente)
  + data\_Revisione ≥ data\_Installazione

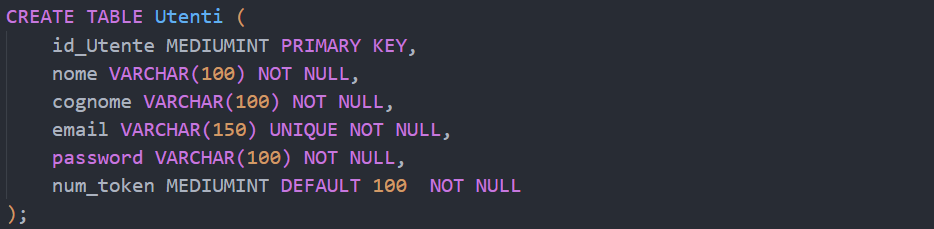
# **7. Implementazione SQL**

✅ CREATE DATABASE

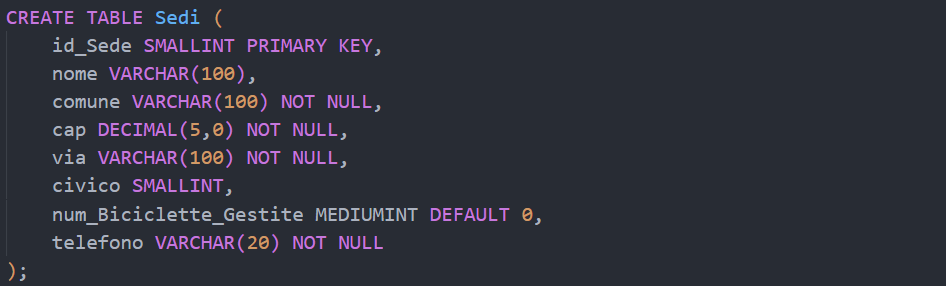


✅ CREATE TABLE

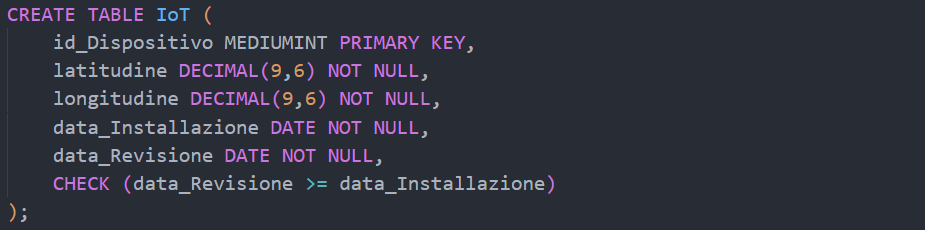
* Utenti



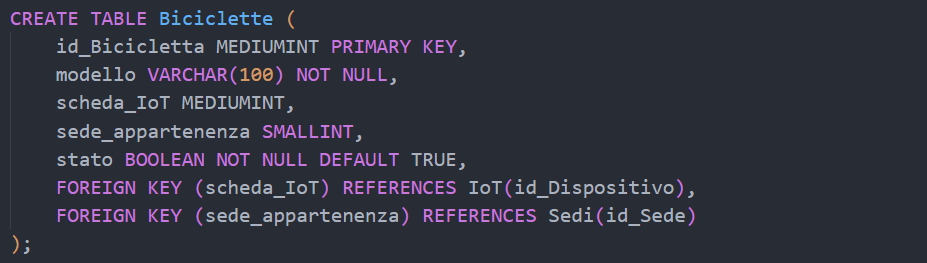
* Sedi



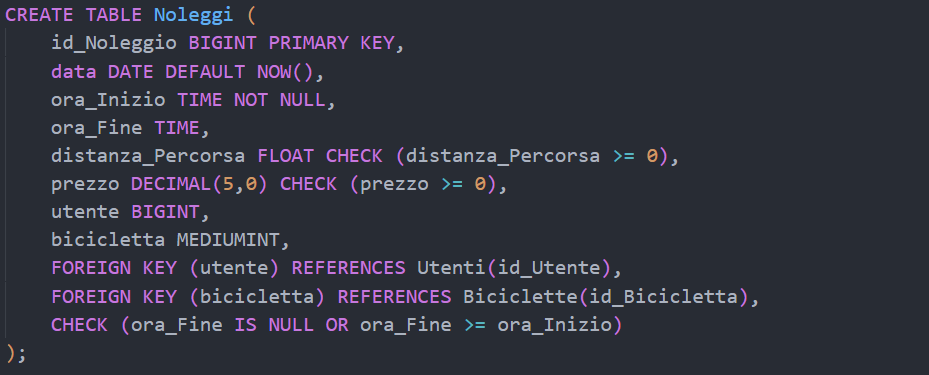
* IoT



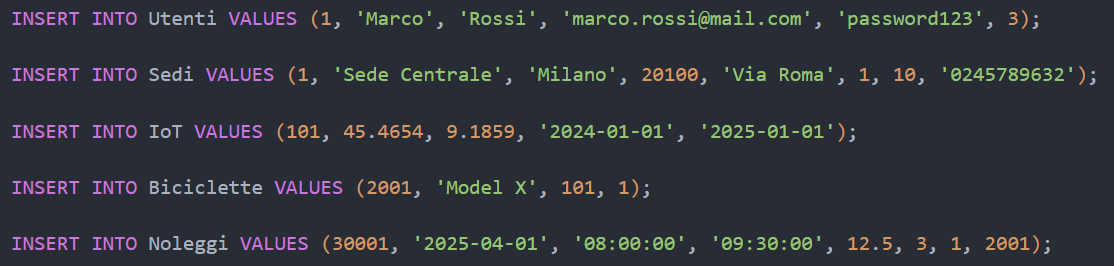
* Biciclette



* Noleggi



✅ INSERT (esempi con dati fittizi per il controllo del corretto funzionamento)



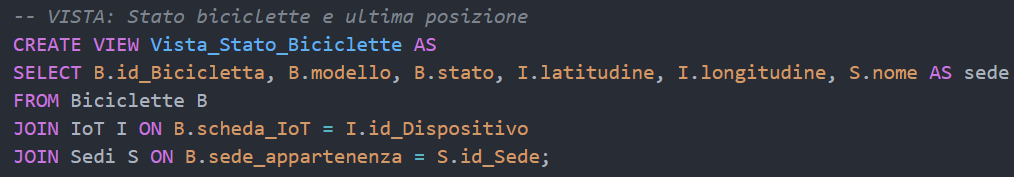
# **8. Viste utili per la web app**

* Noleggi con dati utente e bicicletta

A screen shot of a computer code

Description automatically generated

* Stato biciclette e ultima posizione



# **9. Query SELECT previste per la web app**

A computer screen shot of a code

Description automatically generated

# **10. Conclusioni**

Il progetto **Rollo - Ride your way** ha permesso di progettare e documentare in modo completo la base di dati per un sistema di bike sharing.  
Attraverso l’analisi del problema, la progettazione concettuale e logica, la normalizzazione fino alla forma normale di Boyce-Codd (FNBC), e l’implementazione in SQL, è stato costruito uno schema solido e coerente, pronto per l'integrazione in una web app funzionale.

In futuro, il sistema potrà essere esteso per includere:

* statistiche sull'utilizzo delle biciclette,
* sistema di prenotazione anticipata,
* gestione di account premium e abbonamenti,
* interfacciamento in tempo reale con i dispositivi IoT per il tracciamento live.

Il lavoro rappresenta una base solida sia a livello teorico che pratico per un'applicazione reale e scalabile.