# Allevamento ittico

## Jacopo Mattia Marconi

### Indice

1. ANALISI DEL PROBLEMA:

Il contesto:

* Studiamo il contesto di una società che gestisce vasche a temperatura controllata per l’allevamento di pesci tropicali. L’allevamento consiste in una serie di vasche totalmente autosufficienti per fornire l’illuminazione e l’alimentazione necessarie alle diverse specie presenti. Ogni vasca è dotata di una serie di sensori per impostare, tramite degli attuatori, le condizioni ottimali per la specie di pesce contenuta (es. quantità di nutrienti, ciclo di illuminazione, temperatura dell’acqua, ecc…). Le vasche, quando sono in sede, sono collegate a un sistema centrale attraverso la rete aziendale a cui trasmettono tutti i parametri rilevati e da cui ricevono le impostazioni di funzionamento. Nel database del sistema centrale sono memorizzate le informazioni specifiche di configurazione standard dell’azienda e le necessità delle diverse specie di pesci da utilizzare per l’impostazione delle vasche. La società rende disponibile un servizio di noleggio per utenti esterni (ad esempio fiere che inseriscono le vasche come elemento di arredamento temporaneo); in tal caso le vasche per poter funzionare devono potersi collegare al sistema di gestione centrale. Gli utenti possono collegarsi ad un’applicazione per acquistare il servizio e per verificare lo stato della propria vasca. Settimanalmente la società deve generare la bolletta di pagamento per i propri clienti, a tal proposito è necessario conteggiare il consumo effettivo di ogni vasca noleggiata (costo fisso orario, nutrienti ed energia consumati).

Informazioni aggiuntive:

* Ipotizziamo che la società sia composta da una singola sede centrale comprensiva di vertice strategico, linea intermedia e nucleo operativo con ampia zona di esposizione per le vasche. Quest’ultime possono variare in peso, lunghezza, larghezza e profondità, conseguentemente a ciò, ogni vasca possiede un costo fisso qualora venga noleggiata. L’azienda possiede diversi modelli di vasche e per ogni modello ci possono essere una o più vasche prodotte. Le vasche dispongono di un’interfaccia di rete ethernet, con connettore standard RJ-45, e di un’interfaccia di rete wireless, con standard Wi-Fi 4 (IEEE 802.11n) con velocità 72-600 Mbit/s e frequenze di 2.4 o 5 GHz. Ogni singola vasca è dotata di uno schermo touch resistivo, per facilitarne l’uso del personale tramite guanti protettivi (necessari sul luogo di lavoro). Tramite lo schermo touch è possibile visualizzare informazioni primarie come temperatura, livello di ossigeno, livello dell’acqua e tipologia di pesci contenuti. Dal medesimo schermo è possibile accendere e spegnere le luci interne ed esterne della vasca (funzionalità pensata a fini dimostrativi). Le vasche situate all’interno dell’azienda possono essere collegate indipendentemente tramite rete cablata o wireless al sistema informativo centrale composto da un database server MySQL interno (non raggiungibile dalla rete internet) raggiungibile, invece, da un server web (posto in DMZ). I clienti possono accedere all’area riservata in seguito ad una autenticazione con mail e password personali attraverso il sito web aziendale. Le vasche noleggiate dai clienti, e ovviamente poste all’esterno dell’azienda, possono essere a loro volta collegate tramite rete cablata o wireless, a seconda delle esigenze, e comunicano al sistema centrale dell’azienda tramite VPN Site-to-site con protocollo di tunneling Ipsec. I clienti che usufruiscono del servizio di noleggio (unico servizio proposto dall’azienda, per garantire supporto e qualità) devono necessariamente possedere una partita iva, senza la quale non è possibile noleggiare le vasche. L’azienda dispone di un servizio di supporto telefonico di pronto intervento a tempo pieno.

Il sito web aziendale dispone di:

* una sezione composta da informazioni aziendali (storia, inizi, numeri, personale, politica d’impresa, ecc…);
* una sezione di vendita per le aziende del servizio di noleggio delle vasche con un catalogo delle vasche disponibili al noleggio e le relative informazioni (misure, costo, materiali utilizzati, ecc…);
* una sezione riservata, con accesso tramite mail e password personali, con differenti autorizzazioni in base al ruolo di ciascuno per l’azienda (personale interno o clienti);

Quando un nuovo utente si registra tra i campi richiesti ci saranno mail e password, quest’utlime saranno le credenziali di accesso all’area riservata. Il personale si dovrà registrare con la mail aziendale.

L’utente “cliente”:

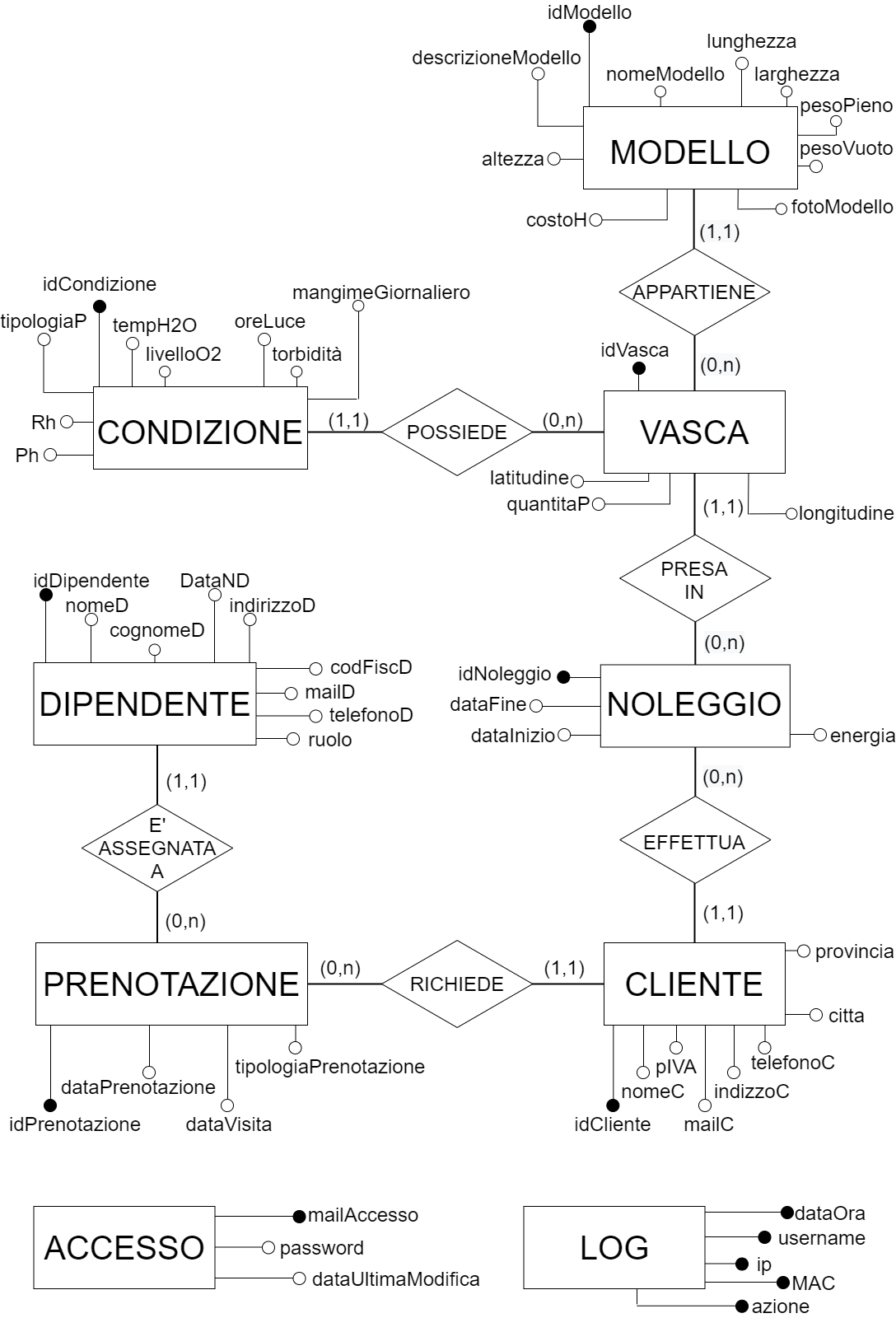
* Dopo aver eseguito l’accesso tramite autenticazione con mail e password personali, il singolo cliente potrà visualizzare lo stato completo della sua vasca e il costo fisso orario, nonché il costo totale dall’inizio del noleggio; è possibile visualizzare, inoltre, lo storico relativo ai noleggi delle vasche dell’utente stesso. In ogni momento, l’utente “cliente” è in grado di visualizzare lo storico dei dati relativi ad ogni singola vasca posseduta da lui stesso.

L’utente “personale”:

* Il personale interno dell’azienda, una volta effettuato l’accesso all’area riservata, è in grado di controllare lo stato di ciascuna vasca, interna all’azienda o noleggiata da un cliente, per monitorare lo stato di ciascuna e intervenire prontamente in caso di necessità.

Il sito web è scritto prevalentemente in PHP, per la facile integrazione con i servizi offerti da MySQL.

Oltre al sito web e all’area riservata al suo interno accessibile da ogni utente sarà necessario implementare un servizio di gestione interno all’azienda. Questo ulteriore servizio sarà sempre implementato tramite una web application in locale e sarà accessibile solamente ai dipendenti autorizzati. La web application in questione avrà il compito di monitorare e permettere all’utente di aggiornare i dati presenti nel database aziendale e di assegnare il personale alle singole prenotazioni online effettuate dai clienti bisognosi di assistenza tecnica o di un sopraluogo da parte di un dipendente (tecnico) dell’azienda stessa.



NOTE:

* Condizione: ogni condizione descrive i valori ideali dell’habitat riferiti alla tipologia di pesce.

L’attributo *mangimeGiornaliero* rappresenta i grammi di cibo che giornalmente ogni singolo pesce deve assumere.

* Modello: il costo orario di ogni singola modello di vasca dipende dalle misure del modello stesso.

Gli attributi *nomeModello* e *fotoModello* hanno uno scopo puramente espositivo, utilizzati nel sito web.

* Vasca: ogni vasca è dotata di GPS per motivi di sicurezza.

L’attributo *quantitaP*, riferito alla quanità dei pesci presenti, può essere null se la vasca non è in utilizzo o non è “abitata”.

* Noleggio: al momento del contratto di noleggio viene indicata la durata del noleggio stesso, per questo motivo l’attributo *dataFIne* non è opzionale.

L’attributo *energia* contiene il dato dell’energia consumata dall’inizio del noleggio, aggiornata giornalmente.

* Cliente: l’azienda permette il noleggio delle vasche solamente ai clienti dotati di partita IVA.
* Prenotazione: un cliente può richiedere una prenotazione online descrivendo solamente la tipologia di intervento richiesto. L’attributo *dataPrenotazione* si riferisce alla data nella quale il cliente richiede la prenotazione. L’attributo dataVisita è opzionale, viene aggiunto dal personale autorizzato interno dopo aver visionato le disponibilità dell’azienda.
* Dipendente: in base al ruolo ogni dipendente potrà, o meno, accedere al servizio di gestione interno dell’azienda o di essere disponibile ad essere mandato in assistenza esterna presso un cliente.
* Accesso: l’entità *Accesso* si riferisce agli accessi all’area riservata del sito web. Ogni accesso è diversificato grazie alla mail, che definisce univocamente l’entità stessa.

L’attributo *dataUltimaModifica* è necessario per permettere all’utente di cambiare password di accesso ogni 3 mesi.

* Log: l’entità Log viene utilizzata per tenere traccia di ogni azione svolta all’interno dell’area riservata. Ogni record memorizzato viene cancellato dopo 6 mesi.

L’attributo *MAC* è opzionale poichè è possibile tenerne traccia solo se l’ip è locale.

Documentazione delle ENTITA’:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NOME | DESCRIZIONE | ATTRIBUTI | OPZ. ATTR. | TIPO | DOMINIO |
| Condizione | L’entità *Condizione* contiene le informazioni sulle diverse condizioni che ogni vasca possiede a seconda della tipologia di pesce. | **idCondizione PK**  tipologiaP  tempH2O  livelloO2  oreLuce  torbidità  mangimeGiornaliero  Rh  Ph | No  No  No  No  No  No  No  No  No | Int  Varchar 20  Int  Float  Int  Int  Int  Int  Int  Int | >0  >=0  >=0  >=0  >0  >200  >0 |
| Modello | L’entità *Modello* contiene le informazioni proprie di un modello. | **idModello PK**  nomeModello  fotoModello  costoH  lunghezza  larghezza  altezza  pesoVuoto  pesoPieno  descrizioneModello | No  No  Si  No  No  No  No  No  No  No | Int  Varchar 20  Blob  Float  Float  Float  Float  Int  Int  Varchar 300 | >0  >0 && <600 >0 && <200  >0 && <250  >0  >pesoVuoto |
| Vasca | L’entità *Vasca* contiene le informazioni proprie di una vasca. | **idVasca PK**  quantitaPesci  latitudine  longitudine | No  Si  No  No | Int  Int  Double  Double | >=0 |
| Noleggio | L’entità *Noleggio* contiene le informazioni relative ai noleggi delle vasche effettuati dai clienti | **idNoleggio PK**  dataInizio  dataFine  energia | No  No  No  No | Int  Date  Date  Float | >dataInizio  >=0 |
| Cliente | L’entità *Cliente* contiene le informazioni relative ad ogni singolo cliente | **idCliente PK**  nomeC  pIVA  mailC  indirizzoC  telefonoC  citta  provincia | No  No  No  No  No  No  No  No | Int  Varchar 20  Char (11)  Varchar 20  Varchar 20  Char (10) |  |
| Prenotazione | L’entità *Prenotazione* contiene le informazioni relative ad ogni prenotazione di visita di un tecnico da parte di un cliente. | **idPrenotazione PK**  dataPrenotazione  dataVisita  tipologiaPrenotazione | No  No  Si  No | Int  Date  Date  Varchar 30 | {sopraluogo, assistenza tecnica, collaborazione, assistenza all’acquisto} |
| Dipendente | L’entità *Dipendente* si riferisce alle informazioni utili di ogni dipendente dell’azienda | **idDipendente PK**  nomeD  cognomeD  dataND  indirizzoD  codfiscD  mailD  telefonoD  ruolo | No  No  No  No  No  No  No  No  No | Int  Varchar 20  Varchar 20  Date  Varchar 20  Char 16  Varchar 20  Char 10  Varchar 20 | {tecnico, amministrativo, personale} |
| Accesso | L’entità *Accesso* contiene i dati relativi all’accesso all’area riservata del sito web aziendale | **mailAccesso PK**  dataUltimaModifica  password | No  No  No | Varchar 20  date  Varchar 20 | >8 caratteri (>1 maiuscola, >1 carattere speciale, >1 numero, no sequenze di alcun tipo) |
| Log | L’entità *Log* contiene i dati di ogni singola azione svolta da un utente all’interno dell’area riservata del sito web aziendale | **dataOra PK**  **username PK**  **ip PK**  **MAC PK**  **Azione PK** | No  No  No  Si  No | DateTime  Varchar 20  Varchar 15  Varchar 17  Varchar 20 |  |

Documentazione delle ASSOCIAZIONI:

|  |  |  |  |
| --- | --- | --- | --- |
| NOME | DESCRIZIONE | ENTITA’ COINVOLTE | ATTRIBUTI |
| possiede | L’associazione *possiede* descrive il legame tra le vasche e le relative condizioni | Condizione  Vasca |  |
| appartiene | L’associazione *appartiene* descrive il legame tra i modelli e le vasche prodotte | Modello  Vasca |  |
| presa in | L’associazione *presa in* descrive il legame tra le vasche e i relativi noleggi | Vasca  Noleggio |  |
| effettua | L’associazione *effettua* descrive il legame tra i noleggi e i clienti | Noleggio  Cliente |  |
| richiede | L’associazione *richiede* descrive il legame tra i clienti e le prenotazioni effettuate | Cliente  Prenotazione |  |
| è assegnata a | L’associazione *è assegnata a* descrive il legame tra le prenotazioni e i dipendenti a cui sono affidate le prenotazioni | Prenotazione  Dipendente |  |

CREATE DATABASE vascheittiche;

CREATE TABLE accessi (

mailAccesso varchar(40) NOT NULL,

dataUltimaModifica date NOT NULL,

password varchar(20) NOT NULL,

PRIMARY KEY (mailAccesso));

CREATE TABLE clienti (

idCliente int(5) NOT NULL,

nomeCliente varchar(30) NOT NULL,

pIVA char(11) NOT NULL,

mailCliente varchar(40) NOT NULL,

indirizzoCliente varchar(30) NOT NULL,

telefonoCliente char(10) NOT NULL,

citta varchar(30) NOT NULL,

provincia varchar(30) NOT NULL,

PRIMARY KEY (idCliente));

CREATE TABLE condizioni (

idCondizione int(5) NOT NULL,

tipologiaP varchar(30) NOT NULL,

tempH2O int(11) NOT NULL,

livelloO2 float NOT NULL,

oreLuce int(11) NOT NULL,

torbidita int(11) NOT NULL,

mangimeGiornaliero int(11) NOT NULL,

Rh int(11) NOT NULL,

Ph int(11) NOT NULL,

PRIMARY KEY (idCondizione));

CREATE TABLE dipendenti (

idDipendente int(5) NOT NULL,

nomeDipendente varchar(30) NOT NULL,

cognomeDipendente varchar(30) NOT NULL,

dataNDipendente date NOT NULL,

indirizzoDipendente varchar(30) NOT NULL,

codFiscDipendente char(16) NOT NULL,

mailDipendente varchar(40) NOT NULL,

telefonoDipendente char(10) NOT NULL,

ruoloDipendente varchar(20) NOT NULL,

PRIMARY KEY (idDipendente));

CREATE TABLE log (

dataOra date NOT NULL,

username varchar(30) NOT NULL,

ip varchar(15) NOT NULL,

MAC varchar(17) DEFAULT NULL,

azione varchar(30) NOT NULL,

PRIMARY KEY (dataOra, username, ip, MAC, azione));

CREATE TABLE noleggi (

idNoleggio int(5) NOT NULL,

dataInizio date NOT NULL,

dataFine date NOT NULL,

energia float NOT NULL,

idCliente int(5) NOT NULL,

idVasca int(5) NOT NULL,

PRIMARY KEY (idNoleggio),

FOREIGN KEY (idCliente) REFERENCES clienti (idCliente),

FOREIGN KEY (idVasca) REFERENCES vasche (idVasca));

CREATE TABLE prenotazioni (

idPrenotazione int(5) NOT NULL,

dataPrenotazione date NOT NULL,

dataVisita date NOT NULL,

tipologiaPrenotazione varchar(30) NOT NULL,

idCliente int(5) NOT NULL,

idDipendente int(5) DEFAULT NULL,

PRIMARY KEY (idPrenotazione),

FOREIGN KEY (idCliente) REFERENCES clienti (idCliente),

FOREIGN KEY (idDipendente) REFERENCES dipendenti (idDipendente));

CREATE TABLE modelli (

idModello int(5) NOT NULL,

nomeModello varchar(30) NOT NULL,

fotoModello longblob DEFAULT NULL,

costoH float NOT NULL,

lunghezza float NOT NULL,

larghezza float NOT NULL,

altezza float NOT NULL,

pesoVuoto int(11) NOT NULL,

pesoPieno int(11) NOT NULL,

descrizioneModello varchar(100) NOT NULL

PRIMARY KEY (idModello));

CREATE TABLE vasche (

idVasca int(5) NOT NULL,

quantitaPesci int(11) DEFAULT NULL,

latitudine varchar(10) NOT NULL,

longitudine varchar(10) NOT NULL,

idCondizione int(5) NOT NULL,

idModello int(5) NOT NULL,

PRIMARY KEY (idVasca),

FOREIGN KEY (idCondizione) REFERENCES condizioni (idCondizione),

FOREIGN KEY (idModello) REFERENCES modelli (idModello));

Condizioni(idCondizione, tempH2O, livelloO2, oreLuce, mangimeGiornaliero, torbidita, Rh, Ph, tipologiaP)

Modelli(idModello, costoH, lunghezza, larghezza, altezza, pesoVuoto, pesoPieno, fotoModello, descrizioneModello, nomeModello)

Vasche(idVasca, quantitaP, latitudine, longitudine, idCondizione, idModello)

Noleggi(idNoleggio, dataInizio, dataFine, energia, idVasca, idCliente)

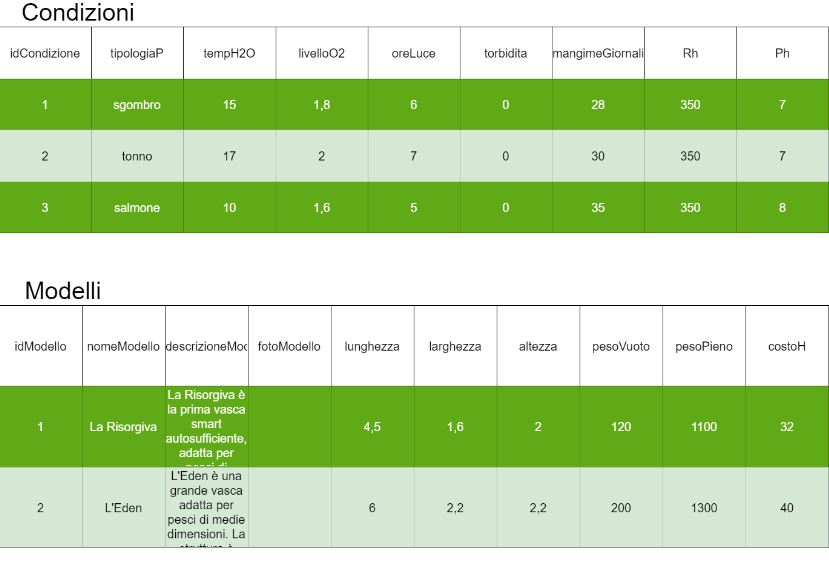
Clienti(idCliente, nomeCliente, pIVA, mailCliente, indirizzoCliente, telefonoCliente, citta, provincia)

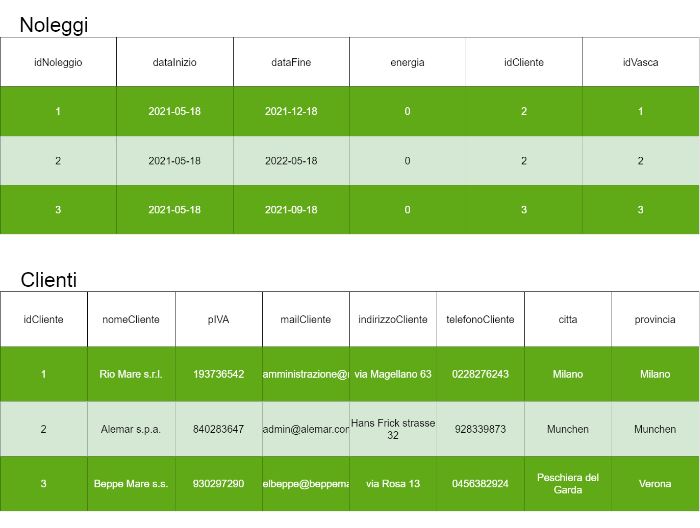
Prenotazioni(idPrenotazione, dataPrenotazione, dataVisita, tipologiaPrenotazione, idCliente, idDipendente)

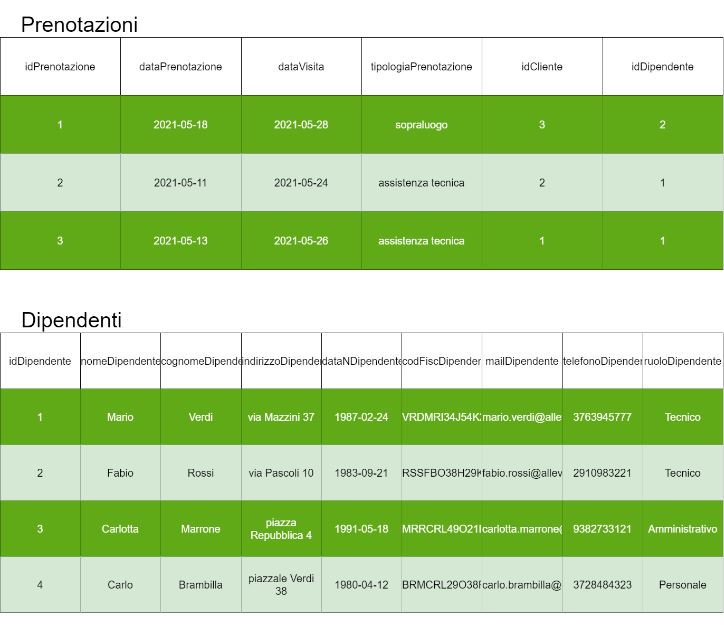
Dipendenti(idDipendente, nomeDipendente, cognomeDipendente, dataNDipendente, indirizzoDipendente, codFiscDipendente, mailDipendente, telefonoDipendente, ruolo)

Accessi(mailAccesso, password, dataUltimaModifica)

Log(dataOra, username, ip, MAC, azione)









Esempio query significative:

* Trovare i dipendenti (solo tecnici) con meno impegni programmati. Il risultato di questa query sarà utile al personale interno autorizzato per poter assegnare equamente gli impegni tra i tecnici presenti:

create view impegniDipendenti as SELECT d.idDipendente, COUNT(\*) AS impegniDipendente from prenotazioni p right JOIN dipendenti d ON p.idDipendente=d.idDipendente GROUP BY (d.idDipendente);

SELECT d.idDipendente, d.nomeDipendente, d.cognomeDipendente, i.impegniDipendente from impegniDipendenti i JOIN dipendenti d ON d.idDipendente=i.idDipendente where i.impegniDipendente=(select min(impegniDipendente) from impegniDipendenti) AND d.ruoloDipendente="Tecnico";

|  |  |  |  |
| --- | --- | --- | --- |
| idDipendente | nomeDipendente | cognomeDipendente | impegniDipendente |
| 2 | Fabio | Rossi | 1 |

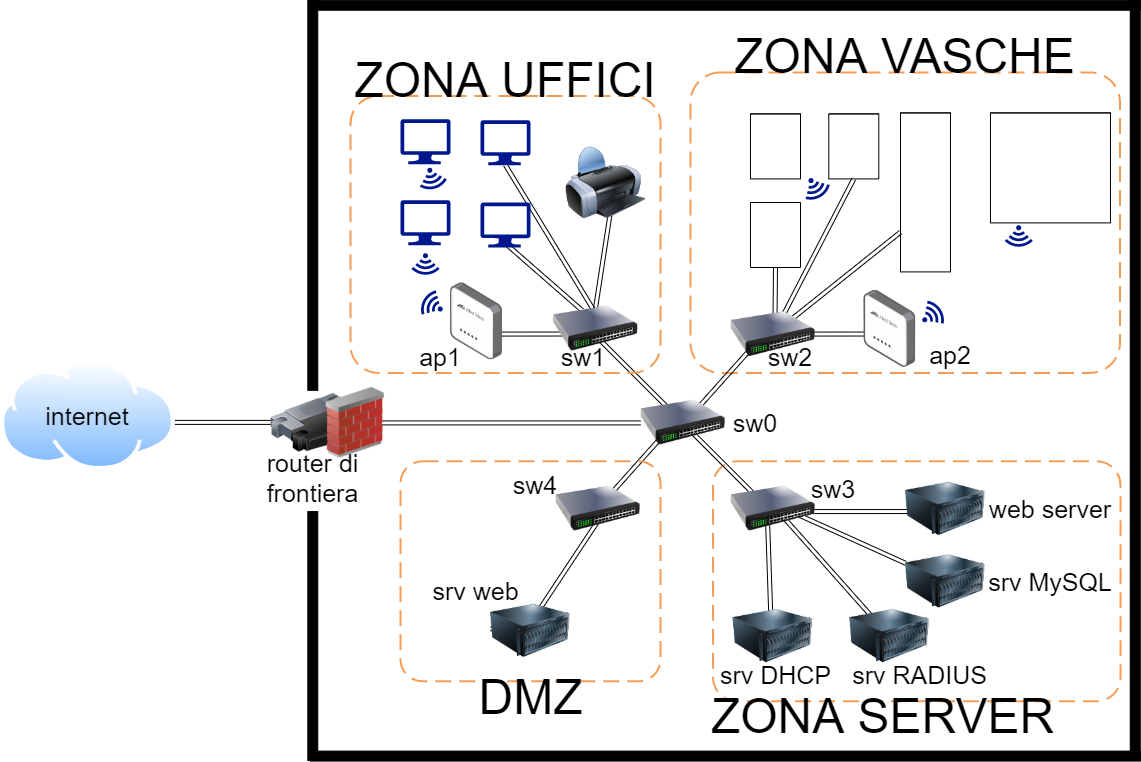
* Trovare il modello di vasca più noleggiato:

create view noleggiModelli as SELECT m.idModello, COUNT(\*) AS noleggiModello FROM noleggi n join vasche v ON v.idVasca=n.idVasca join modelli m ON m.idModello=v.idModello group BY (m.idModello);

SELECT nm.idModello, m.nomeModello, nm.noleggiModello from noleggiModelli nm JOIN modelli m ON m.idModello=nm.idModello where nm.noleggiModello=(select max(noleggiModello) from noleggiModelli);

|  |  |  |
| --- | --- | --- |
| nomeModello | idModello | noleggiModello |
| La Risorgiva | 1 | 2 |

Infrastruttura di rete interna all’azienda:



Esempio di possibile infrastruttura di rete esterna, posseduta da un cliente:

