

Database per Camperisti

Docente: Alfredo Pulvirenti

Studente: Christian Garozzo



Descrizione del progetto

- Si vuole progettare un Database di una piattaforma per la scelta di **Luoghi** di interesse per **Viaggiatori** muniti di camper o simili. I viaggiatori, a seguito di una registrazione, hanno accesso ad informazioni su **Spot** per sostare con i propri **Mezzi** e luoghi di interesse visitabili. Inoltre ad ogni spot vengono associate **Recensioni** che sono scritte solo da utenti registrati.
- Ogni **Viaggiatore** (caratterizzato da ID_Viaggiatore, Nome, Cognome potrebbe essere iscritto o meno, solo a seguito dell'iscrizione il viaggiatore può scrivere recensioni. Ogni **Recensione** è valutata da altri utenti con un sistema di votazione che va da 1 a 5 ed è caratterizzata da ID_Recensione, Valutazione, Spot, Testo. Ogni **Spot** è caratterizzato da ID_Spot, Luogo, Regione, TipologiaSpot. Ogni **Mezzo** è costituito da ID_Mezzo, Targa, TipologiaMezzo.

Glossario dei termini

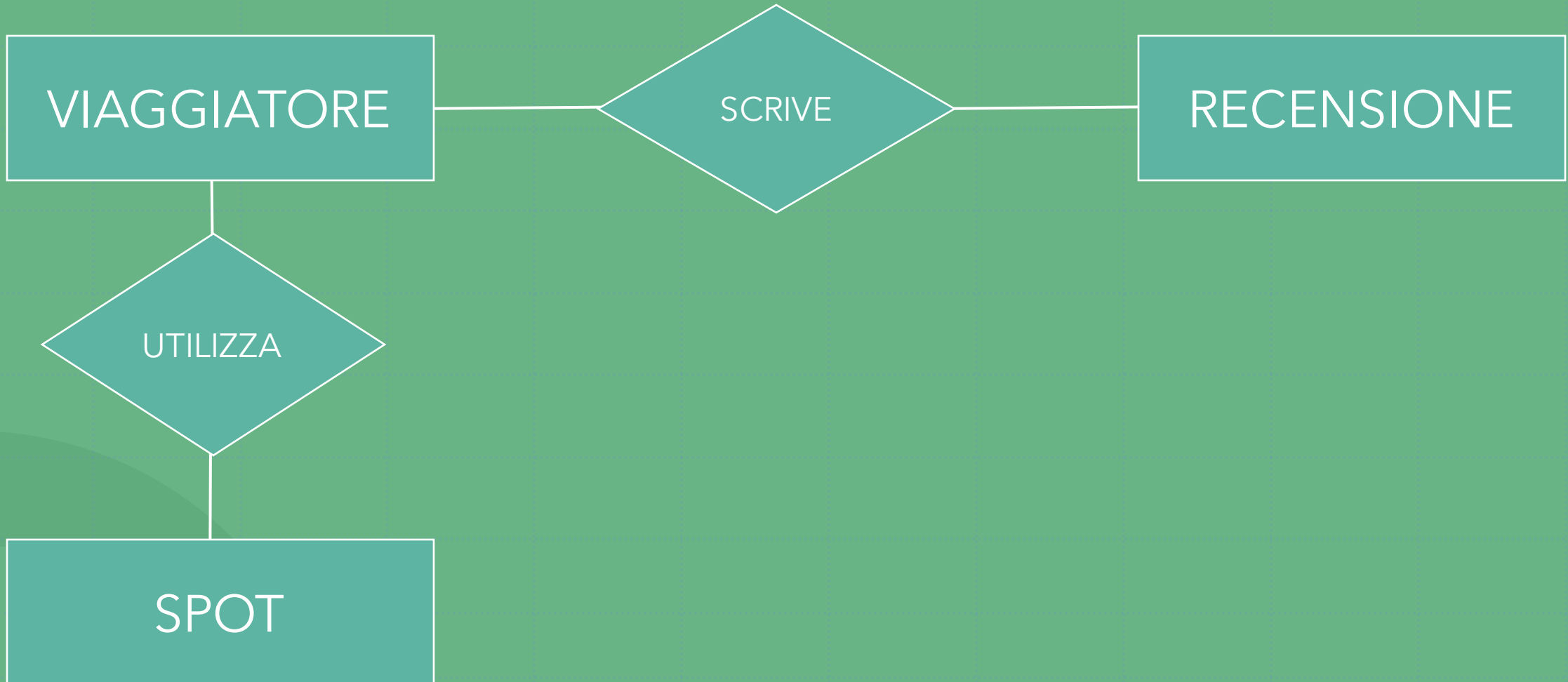
Termini	Descrizione	Sinonimi	Legami
Viaggiatore	Utente che viaggia con il proprio mezzo utilizzando piazzole di sosta.	Utente	Mezzo, Recensione
Mezzo	Vettura utilizzata per il viaggio, essa si divide in diverse tipologie.	Automobile, Vettura, Veicolo	Viaggiatore
Spot	Piazzola di sosta per i veicoli dei viaggiatori.	Piazzola di sosta	Viaggiatore, Recensione
Recensione	Commento che il viaggiatore può scrivere a seguito dell'utilizzo di uno spot.	Commento	Viaggiatore, Spot

Specifiche dei dati

- **Dati di carattere generale:** Si vuole progettare un DataBase di una piattaforma per la scelta di Luoghi d'interesse per Viaggiatori muniti di camper o simili.
- **Dati sui Viaggiatori:** Ogni Viaggiatore (caratterizzato da ID_Viaggiatore, Nome, Cognome) potrebbe essere iscritto o meno. I Viaggiatori a seguito di una registrazione hanno accesso ad informazioni.
- **Dati sulle Recensioni:** Ogni Recensione è valutata da altri utenti con un sistema di votazione che va da 1 a 5 ed è caratterizzata da ID_Recensione, Valutazione, Testo, Registrazione, Spot.
- **Dati sugli Spot:** Ogni Spot è caratterizzato da ID_Spot, Luogo.
- **Dati sul Mezzo:** Ogni Mezzo è costituito da ID_Mezzo, Targa, Tipologia.

Progettazione concettuale

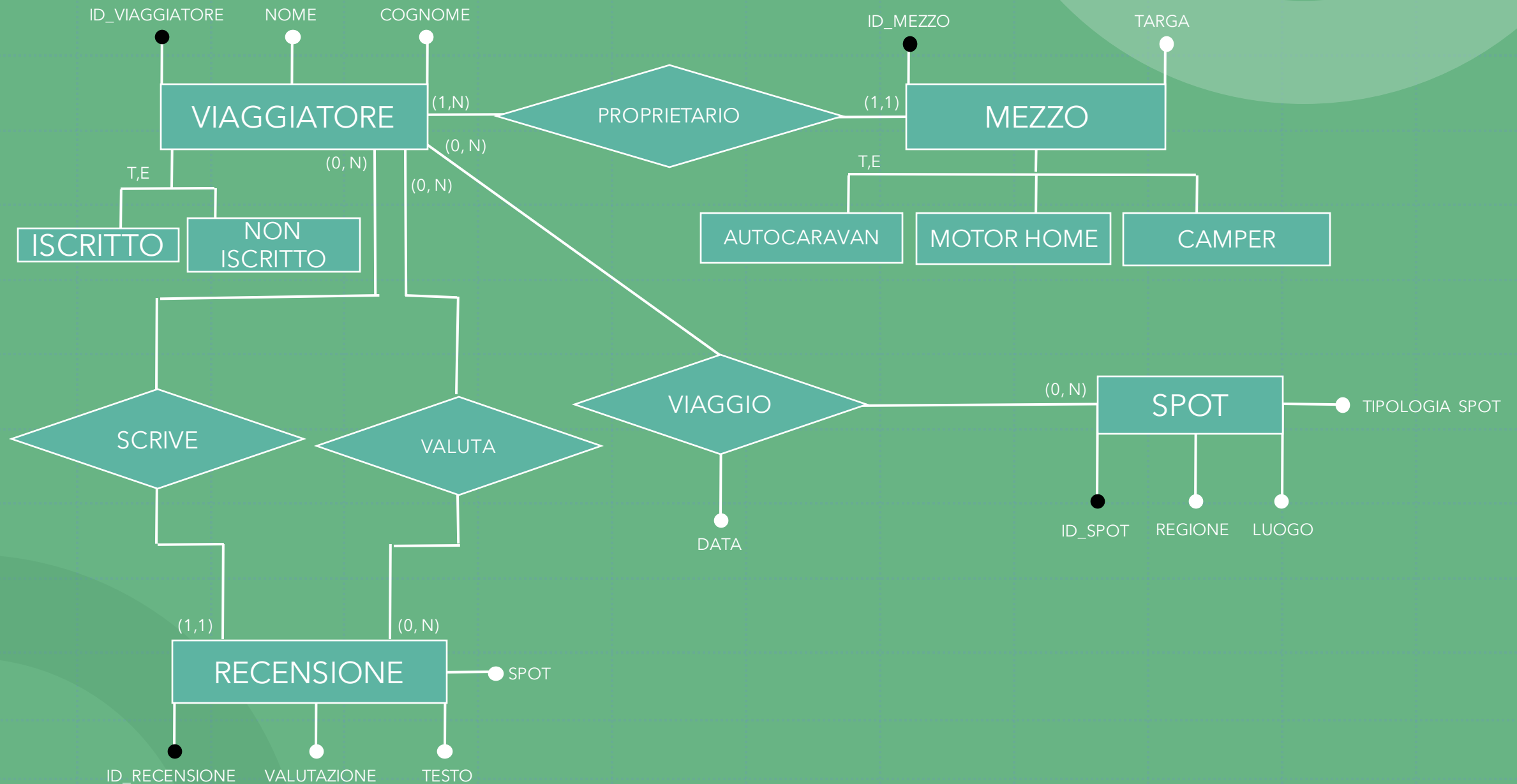
- Approccio top-down
- Schema scheletro



- Decido in seguito di eliminare le associazioni "DEPOSITATO" e "FATTE" poichè ridondanti e quindi superflue ai fini del corretto utilizzo del Data Base. Inoltre inserisco una gerarchia sull'entità "MEZZO", questo mi agevolerà nel collocarlo nel corretto spot e quindi dare informazioni al viaggiatore sulla piazzola di sosta più consona alla tipologia del proprio mezzo di trasporto.

- Continuo a snellire lo schema: trasformo viaggio in un'associazione, ristrutturò l'entità Registrazione e le associazioni connesse ad essa ed infine elimino l'entità luogo e la sua associazione.

➤ Schema finale



Dizionario delle entità

Entità	Descrizione	Attributi	Chiave
Viaggiatore	Utente che viaggia con il proprio mezzo.	ID_Viaggiatore, Nome, Cognome	ID_Viaggiatore
Mezzo	Veicolo utilizzato dal viaggiatore per compiere il suo itinerario.	ID_Mezzo, Targa	ID_Mezzo
Recensione	Scrittura o valutazione dell'utente su uno spot utilizzato durante il viaggio	ID_Recensione, Valutazione, Testo, Spot	ID_Recensione
Spot	Piazzola di sosta per i mezzi dei viaggiatori	ID_Spot, Regione, Luogo, TipologiaSpot	ID_Spot

Dizionario delle relazioni

Relazioni	Entità coinvolte	Descrizione	Attributi
Scrive	Viaggiatore, Recensione	Associa il viaggiatore alla sua Recensione	\
Valuta	Viaggiatore, Recensione	Associa il viaggiatore alla sua eventuale valutazione della recensione.	\
Proprietario	Viaggiatore, Mezzo	Associa il viaggiatore al mezzo che potrebbe usare per il viaggio.	\
Viaggio	Viaggiatore, Spot	Associa il viaggiatore al giusto spot che verrà utilizzato.	Data

Vincoli non esprimibili dallo schema E-R

- Solo utenti registrati possono scrivere una recensione.
- Gli utenti possono recensire solo gli spot utilizzati per il Viaggio.
- Si possono collocare Mezzi solo negli Spot che hanno possibilità di accogliere la tipologia del veicolo.
- Il sistema di valutazione deve contenere dati consistenti.

Progettazione Logica

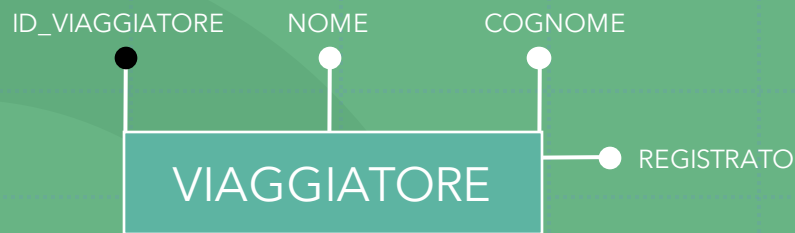
➤ Volume dei dati

Concetto	Tipo	Volume
Viaggiatore	E	200
Recensione	E	100
Mezzo	E	250
Spot	E	50
Scrive	R	100
Valuta	R	100
Proprietario	R	250
Viaggio	R	500

➤ Volume delle frequenze			
Operazione	Descrizione	Frequenza	Tipo
OP.1	Aggiungi Viaggiatori.	1/Giorno	I
OP.2	Associa mezzo ad un Viaggiatore.	2/Giorno	I
OP.3	Aggiungi Mezzo.	4/Mese	B
OP.4	Aggiungi Spot.	2/Mese	B
OP.5	Associa Viaggiatore a Spot.	5/Giorno	I
OP.6	Aggiungi Recensione(Compresa di Scrittura e Valutazione).	5/Giorno	I
OP.7	Mostra tutti i Viaggiatori.	1/Giorno	I
OP.8	Mostra tutti i Mezzi.	1/Giorno	I
OP.9	Mostra viaggi fatti in data 24/07/21.	1/Anno	B
OP.10	Mostra Recensione dei Viaggiatori.	1/Giorno	I
OP.11	Mostra tutti i viaggiatori che non hanno ancora fatto un viaggio.	1/Mese	B

Ristrutturazione dello Schema E-R

- **Analisi delle ridondanze:** Non ci sono attributi ridondanti né associazioni o entità ridondanti.
- **Eliminazione delle generalità:** L'entità Viaggiatore presenta una generalizzazione totale ed esclusiva in «Non registrati» e «Registrati» che può essere eliminata tramite un collasso verso l'alto e l'inserimento di un attributo selettore nell'entità padre, cioè "Registrato", che indica se il Viaggiatore è abilitato o meno a scrivere una recensione. Inoltre l'entità Mezzo presenta anch'essa una generalizzazione totale ed esclusiva in «AutoCaravan», «MotorHome» e «Camper» che può essere eliminata anche in questo caso tramite un collasso verso l'alto e l'inserimento di un attributo selettore nell'entità padre, cioè "TipologiaMezzo", che indica se il Mezzo è idoneo o meno a sostare in uno specifico Spot.



Traduzione verso il modello Logico

➤ Si considerino come chiavi primarie quelle in grassetto, come chiavi esterne quelle sottolineate.

Viaggiatore(**ID_Viaggiatore**, Nome, Cognome, Registrato)

Proprietario(ID_Viaggiatore, ID_Mezzo)

Mezzo(**ID_Mezzo**, Targa, TipologiaMezzo)

Valuta(Viaggiatore, Recensione)

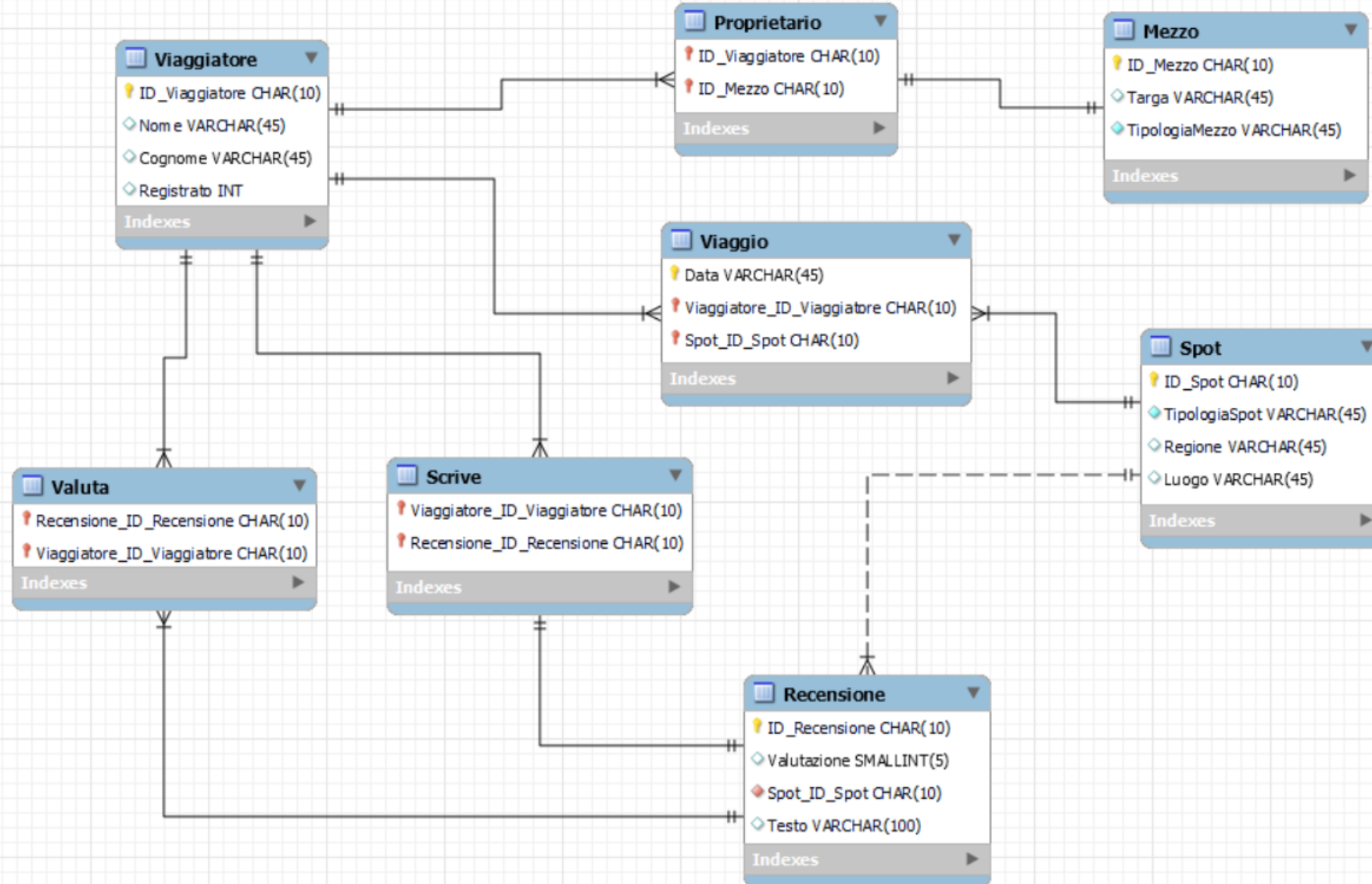
Scriva(Viaggiatore, Recensione)

Recensione(**ID_Recensione**, Spot, Valutazione, Testo)

Viaggio(Viaggiatore, Spot, Data)

Spot(**ID_Spot**, Luogo, Regione, TipologiaSpot)

Progettazione fisica



Codifica delle tabelle SQL

➤ VIAGGIATORE

```
CREATE TABLE IF NOT EXISTS `mydb`.`Viaggiatore` (  
  `ID_Viaggiatore` CHAR(10) NOT NULL,  
  `Nome` VARCHAR(45) NULL,  
  `Cognome` VARCHAR(45) NULL,  
  `Registrato` INT NULL,  
  PRIMARY KEY (`ID_Viaggiatore`))  
ENGINE = InnoDB;
```

➤ PROPRIETARIO

```
CREATE TABLE IF NOT EXISTS `mydb`.`Proprietario` (  
  `ID_Viaggiatore` CHAR(10) NOT NULL,  
  `ID_Mezzo` CHAR(10) NOT NULL,  
  PRIMARY KEY (`ID_Viaggiatore`, `ID_Mezzo`),  
  INDEX `fk_Proprietario_Mezzo1_idx` (`ID_Mezzo` ASC)  
  VISIBLE)  
ENGINE = InnoDB;
```

➤ MEZZO

```
CREATE TABLE IF NOT EXISTS `mydb`.`Mezzo` (  
  `ID_Mezzo` CHAR(10) NOT NULL,  
  `Targa` VARCHAR(45) NULL,  
  `TipologiaMezzo` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`ID_Mezzo`))  
ENGINE = InnoDB;
```

➤ REGISTRAZIONE

```
CREATE TABLE IF NOT EXISTS `mydb`.`Registrazione` (  
  `ID_Registrazione` CHAR(10) NOT NULL,  
  `Data` DATE NULL,  
  PRIMARY KEY (`ID_Registrazione`))  
ENGINE = InnoDB;
```

➤ SPOT

```
CREATE TABLE IF NOT EXISTS `mydb`.`Spot` (  
  `ID_Spot` CHAR(10) NOT NULL,  
  `TipologiaSpot` VARCHAR(45) NOT NULL,  
  `Regione` VARCHAR(45) NULL,  
  `Luogo` VARCHAR(45) NULL,  
  PRIMARY KEY (`ID_Spot`))  
ENGINE = InnoDB;
```

➤ SCRIVE

```
CREATE TABLE IF NOT EXISTS `mydb`.`Scrivi` (  
  `Viaggiatore_ID_Viaggiatore` CHAR(10) NOT NULL,  
  `Recensione_ID_Recensione` CHAR(10) NOT NULL,  
  PRIMARY KEY (`Viaggiatore_ID_Viaggiatore`,  
    `Recensione_ID_Recensione`),  
  INDEX `fk_Commento_Viaggiatore1_idx` (`Viaggiatore_ID_Viaggiatore`  
    ASC) VISIBLE,  
  INDEX `fk_Scrivi_Recensione1_idx` (`Recensione_ID_Recensione` ASC)  
    VISIBLE)  
ENGINE = InnoDB;
```

➤ RECENSIONE

```
CREATE TABLE IF NOT EXISTS `mydb`.`Recensione` (  
  `ID_Recensione` CHAR(10) NOT NULL,  
  `Valutazione` SMALLINT(5) NULL,  
  `Spot_ID_Spot` CHAR(10) NOT NULL,  
  `Testo` VARCHAR(100) NULL,  
  PRIMARY KEY (`ID_Recensione`),  
  INDEX `fk_Recensione_Spot1_idx` (`Spot_ID_Spot` ASC) VISIBLE)  
ENGINE = InnoDB;
```

➤ VALUTA

```
CREATE TABLE IF NOT EXISTS `mydb`.`Valuta` (  
  `Recensione_ID_Recensione` CHAR(10) NOT NULL,  
  `Viaggiatore_ID_Viaggiatore` CHAR(10) NOT NULL,  
  PRIMARY KEY (`Recensione_ID_Recensione`,  
    `Viaggiatore_ID_Viaggiatore`),  
  INDEX `fk_Valuta_Recensione1_idx` (`Recensione_ID_Recensione` ASC)  
    VISIBLE,  
  INDEX `fk_Valuta_Viaggiatore1_idx` (`Viaggiatore_ID_Viaggiatore` ASC)  
    VISIBLE)  
ENGINE = InnoDB;
```

➤ VIAGGIO

- CREATE TABLE IF NOT EXISTS `mydb`.`Viaggio` (
 - `Data` VARCHAR(45) NOT NULL,
 - `Viaggiatore_ID_Viaggiatore` CHAR(10) NOT NULL,
 - `Spot_ID_Spot` CHAR(10) NOT NULL,
 - PRIMARY KEY (`Viaggiatore_ID_Viaggiatore`, `Spot_ID_Spot`, `Data`),
 - INDEX `fk_Viaggio_Spot1_idx` (`Spot_ID_Spot` ASC) VISIBLE)
- ENGINE = InnoDB;

Operazioni in SQL

▪ OP.1 - AGGIUNGI VIAGGIATORI

- INSERT INTO `mydb`.`viaggiatore` (`ID_Viaggiatore`, `Nome`, `Cognome`, `Registrato`) VALUES ('X8100', 'Christian', 'Garozzo', '1');
- INSERT INTO `mydb`.`viaggiatore` (`ID_Viaggiatore`, `Nome`, `Cognome`, `Registrato`) VALUES ('X8101', 'Giuseppe', 'Cavallaro', '0');
- INSERT INTO `mydb`.`viaggiatore` (`ID_Viaggiatore`, `Nome`, `Cognome`, `Registrato`) VALUES ('X8102', 'Calogero', 'Lo Leggio', '1');
- INSERT INTO `mydb`.`viaggiatore` (`ID_Viaggiatore`, `Nome`, `Cognome`, `Registrato`) VALUES ('X8103', 'Enzo', 'Barba', '1');
- INSERT INTO `mydb`.`viaggiatore` (`ID_Viaggiatore`, `Nome`, `Cognome`, `Registrato`) VALUES ('X8104', 'Carlo', 'Baudò', '0');
- INSERT INTO `mydb`.`viaggiatore` (`ID_Viaggiatore`, `Nome`, `Cognome`, `Registrato`) VALUES ('X8105', 'Andrea', 'Barreca', '1');

▪ **OP.2 - ASSOCIA MEZZO AD UN VIAGGIATORE**

- INSERT INTO `mydb`.`proprietario` (`ID_Viaggiatore`, `ID_Mezzo`) VALUES ('X8105', 'M01');

▪ **OP.3 - AGGIUNGI MEZZO**

- INSERT INTO `mydb`.`mezzo` (`ID_Mezzo`, `Targa`, `TipologiaMezzo`) VALUES ('M01', 'XJ99A', 'Camper');
- INSERT INTO `mydb`.`mezzo` (`ID_Mezzo`, `Targa`, `TipologiaMezzo`) VALUES ('M02', 'XK88A', 'AutoCaravan');
- INSERT INTO `mydb`.`mezzo` (`ID_Mezzo`, `Targa`, `TipologiaMezzo`) VALUES ('M03', 'HK77B', 'MotorHome');

▪ **OP.4 - AGGIUNGI SPOT**

- INSERT INTO `mydb`.`spot` (`ID_Spot`, `TipologiaSpot`, `Regione`, `Luogo`) VALUES ('S01', 'Camper', 'Sicilia', 'Taormina');

▪ **OP.5 - ASSOCIA VIAGGIATORE A SPOT**

- INSERT INTO `mydb`.`viaggio` (`Data`, `Viaggiatore_ID_Viaggiatore`, `Spot_ID_Spot`) VALUES ('24/07/21', 'X8105', 'S01');
- INSERT INTO `mydb`.`viaggio` (`Data`, `Viaggiatore_ID_Viaggiatore`, `Spot_ID_Spot`) VALUES ('02/09/94', 'X8100', 'S01');

▪ **OP.6 - AGGIUNGI RECENSIONE (Compreso di scrittura e valutazione)**

- INSERT INTO `mydb`.`recensione` (`ID_Recensione`, `Valutazione`, `Spot_ID_Spot`, `Testo`) VALUES ('R01', '5', 'S01', 'Fantastico!');
- INSERT INTO `mydb`.`scrive` (`Viaggiatore_ID_Viaggiatore`, `Recensione_ID_Recensione`) VALUES ('X8105', 'R01');
- INSERT INTO `mydb`.`valuta` (`Recensione_ID_Recensione`, `Viaggiatore_ID_Viaggiatore`) VALUES ('R01', 'X8105');

- **OP.7 - Mostra tutti i Viaggiatori**

- SELECT *
- FROM VIAGGIATORE

	ID_Viaggiatore	Nome	Cognome	Registrato
▶	X8100	Christian	Garozzo	1
	X8101	Giuseppe	Cavallaro	0
	X8102	Calogero	Lo Leggio	1
	X8103	Enzo	Barba	1
	X8104	Carlo	Baudo	0
	X8105	Andrea	Barreca	1

- **OP.8 - Mostra tutti i Mezzi**

- SELECT *
- FROM MEZZO

	ID_Mezzo	Targa	TipologiaMezzo
▶	M01	XJ99A	Camper
	M02	XK88A	AutoCaravan
	M03	HK77B	MotorHome

- **OP.9 - Mostra Viaggi fatti in data 24/07/21**

- SELECT *
- FROM VIAGGIO
- WHERE DATA = '24/07/21'

	Data	Viaggiatore_ID_Viaggiatore	Spot_ID_Spot
▶	24/07/21	X8105	S01

- **OP.10 - Mostra Recensione dei Viaggiatori**

- SELECT ID_Viaggiatore, ID_Recensione, Nome, Cognome, Testo
- FROM scrive, viaggiatore, recensione
- WHERE ID_Recensione = Recensione_ID_Recensione AND
ID_Viaggiatore = Viaggiatore_ID_Viaggiatore

	ID_Viaggiatore	ID_Recensione	Nome	Cognome	Testo
▶	X8105	R01	Andrea	Barreca	Fantastico!

▪ **OP-11 Mostra tutti i Viaggiatori che non hanno ancora fatto un viaggio.**

```
SELECT *  
FROM viaggiatore  
WHERE ID_Viaggiatore not in (SELECT ID_Viaggiatore  
                             FROM viaggio, viaggiatore  
                             WHERE ID_Viaggiatore =      Viaggiatore_ID_Viaggiatore)
```

	ID_Viaggiatore	Nome	Cognome	Registrato
►	X8101	Giuseppe	Cavallaro	0
	X8102	Calogero	Lo Leggio	1
	X8103	Enzo	Barba	1
	X8104	Carlo	Baudo	0

Implemento i vincoli

1. Solo utenti registrati possono scrivere o valutare una recensione.
2. Si possono collocare Mezzi solo negli Spot che hanno possibilità di accogliere la tipologia del veicolo.
3. Il sistema di valutazioni deve contenere dati consistenti.
4. Gli utenti possono recensire solo gli spot utilizzati per il Viaggio.

Trigger in SQL

```
1) CREATE TRIGGER T1
• AFTER INSERT ON VALUTA
• FOR EACH ROW
• BEGIN
• IF (
•   ( SELECT REGISTRATO
•     FROM VIAGGIATORE
•     WHERE ID_Viaggiatore = Viaggiatore
•   ) = '0'
• ) THEN
• DELETE FROM VALUTA
• WHERE Viaggiatore = new.Viaggiatore;
• END IF;
• END
```

```
1) CREATE TRIGGER T1
• AFTER INSERT ON SCRIVE
• FOR EACH ROW
• BEGIN
• IF (
•   ( SELECT REGISTRATO
•     FROM VIAGGIATORE
•     WHERE ID_Viaggiatore = Viaggiatore
•   ) = '0'
• ) THEN
• DELETE FROM SCRIVE
• WHERE Viaggiatore = new.Viaggiatore;
• END IF;
• END
```

Asserzioni in SQL

- **2)**
- Create Assertion A1
- Check Not Exists (
• Select *
• From Proprietario, Mezzo, Viaggio, Spot
• Where Proprietario.ID_Viaggio = Viaggio.Viaggiatore and
• Proprietario.ID_Mezzo = Mezzo.ID_Mezzo and
• Viaggio.Spot = Spot.ID_Spot and
• TipologiaMezzo <> TipologiaSpot
•)

Asserzioni in SQL

- **3)**
- Create Assertion A2
- Check Not Exists (Select *
- From Recensione
- Where valutazione > 5 and valutazione < 1
-)

- **4)**
- Create Assertion A3
- Check Not Exists (Select*
- From Viaggio,Recensione,Scrive
- Where Scrive.ID_Viaggiatore =
Viaggio.ID_Viaggiatore and
- Scrive.ID_Recensione =
Recensione.ID_Recensione and
- Viaggio.Spot <> Recensione.Spot)