

PROGETTO DI BASI DI DATI

Prenotazioni online

Alessandro Catalano X81000464

A.A. 2018/2019

INDICE

- 3 – Descrizione e specifiche
- 5 – Glossario dei termini
- 8 – Progettazione concettuale
- 11 – Dizionario dati
- 14 – Specifiche sulle operazioni
- 15 – Tavola dei volumi
- 17 – Analisi delle ridondanze
- 18 – Schemi delle operazioni
- 39 – Eliminazione delle gerarchie
- 41 – Partizionamento/Accorpamento Entità
- 42 – Schema ristrutturato
- 43 – Modello logico
- 44 – Schema Fisico
- 45 – Creazione del database
- 49 – Implementazione operazioni
- 54 – Vincoli d'integrità

Descrizione e specifiche

Si vuole progettare il database di un'applicazione per la gestione di prenotazioni online ad attività commerciali convenzionate.

Gli utenti (clienti) sono identificati da un id univoco e inoltre si memorizzano nome, cognome, data di nascita e sesso oltre ad un recapito di telefono verificato obbligatorio per essere eventualmente contattato in caso di cambiamenti.

Le attività, per essere registrate nel database ed essere disponibili agli utenti, devono fornire la partita iva, l'indirizzo di fatturazione, la denominazione, una breve descrizione e il settore di appartenenza. Ogni attività ha degli amministratori che si occupano di controllare ed eventualmente accettare le prenotazioni.

Le prenotazioni vengono messe a disposizione degli utenti dall'amministratore dell'attività.

Le prenotazioni sono caratterizzate da una data/ora, da un'attività e da un utente.

Inoltre l'applicazione prevede un sistema di feedback per evitare eventuali truffe; i clienti possono fornire un feedback solo sulle attività e viceversa le attività possono fornire un feedback sul cliente. Per ogni feedback si memorizza anche un commento (opzionale).

Glossario dei termini

Termine	Descrizione	Sinonimi	Termini collegati
Attività	Attività commerciale convenzionata che mette a disposizione gli orari	Azienda	Amministratore, Prenotazione
Utente	Persona fisica registrata che prenota appuntamenti nei suoi locali preferiti	Cliente	Prenotazione
Amministratore	Colui che gestisce le prenotazioni di un'attività	Admin	Attività
Prenotazione	La prenotazione con data e ora di un utente ad una qualche attività commerciale		Utente, Amministratore, Attività
Feedback	Il punteggio di un utente o di una attività commerciale	Recensione, Punteggio	Utente, Attività
Commento	Il testo di una recensione o una nota per una prenotazione		Feedback, Prenotazione

Raggruppamento dei dati

Dati di carattere generale

Si vuole progettare il database di un'applicazione in cui gli utenti possono prenotare appuntamenti per delle attività convenzionate.

Dati sulle attività

Per le attività (circa 2.000) rappresentiamo la partita iva, la denominazione, l'indirizzo di fatturazione, la descrizione, l'area di appartenenza ed il feedback. Una attività può appartenere ad una sola categoria.

Dati sugli utenti

Per gli utenti (circa 50.000) rappresentiamo l'id univoco, il nome, il cognome, l'email, la password, la data di nascita, il numero di telefono, il sesso ed il feedback.

Dati sugli amministratori

Per gli amministratori (circa 2.500) rappresentiamo il codice fiscale, il nome, il cognome, l'email, la password, la data di nascita, il numero di telefono, il sesso e l'attività che gestisce.

Dati sulle prenotazioni

Per le prenotazioni (circa 7.000 al giorno) rappresentiamo l'attività a cui fa riferimento insieme alla data e l'ora prestabilita e l'eventuale utente che ha confermato la prenotazione insieme al commento se presente.

Dati sui feedback

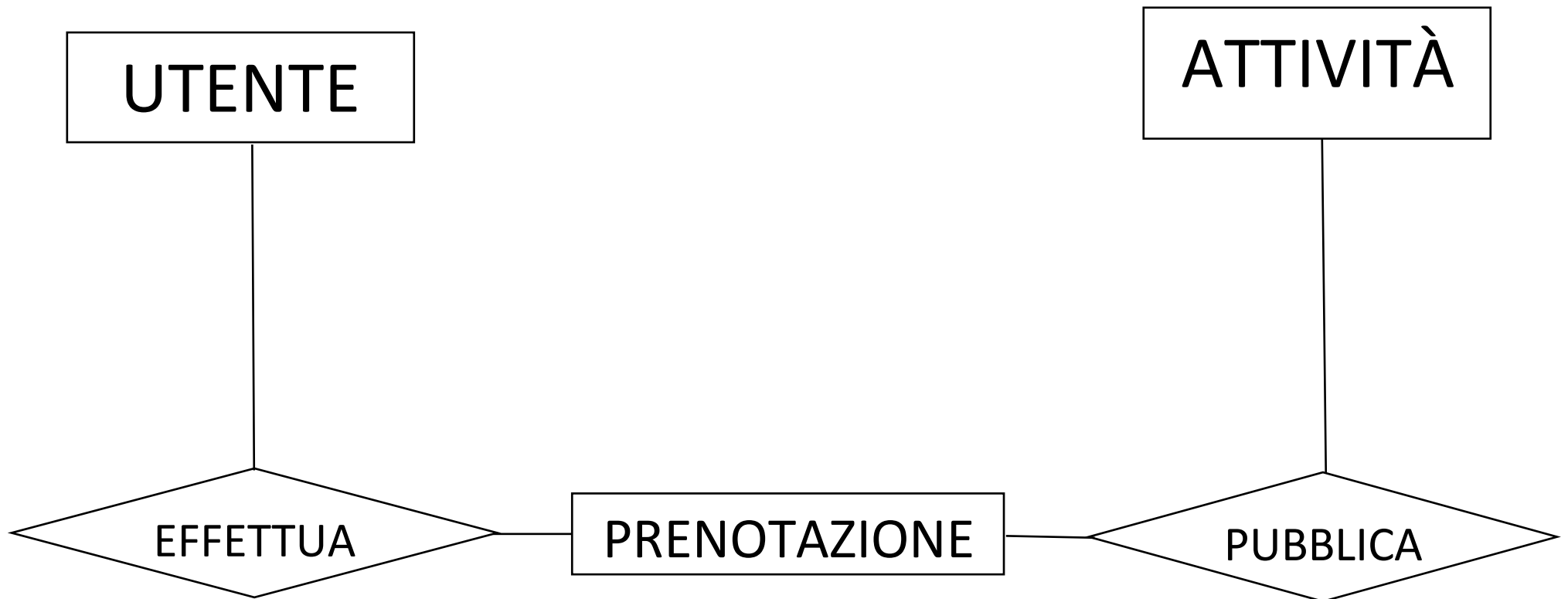
Per i feedback rappresentiamo l'utente o l'attività a cui fa riferimento, il punteggio (che va da 1 – pessimo a 5 – eccellente) e l'eventuale commento.

Dati sui commenti

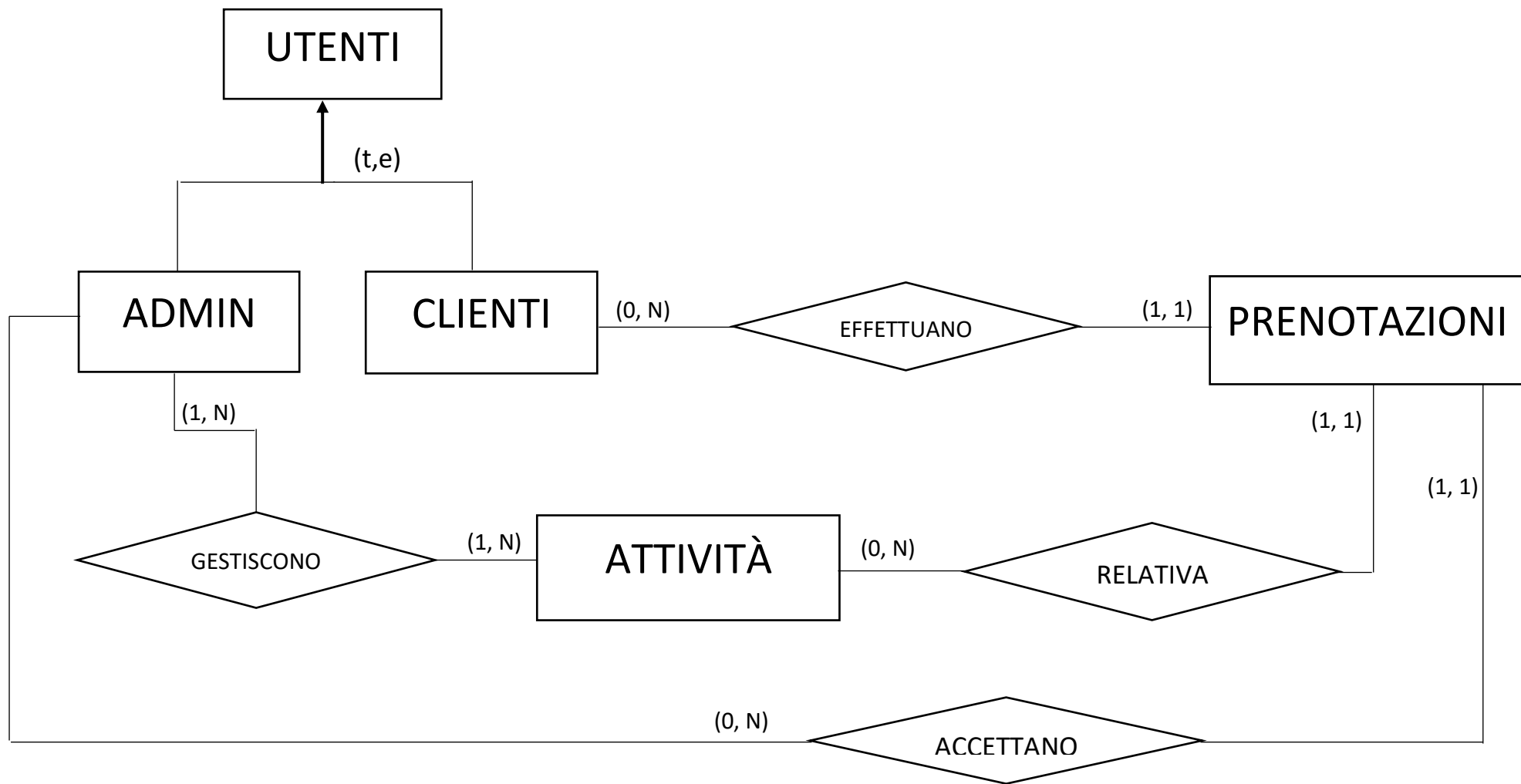
Per i commenti rappresentiamo un testo e l'utente che ha scritto il commento.

PROGETTAZIONE CONCETTUALE

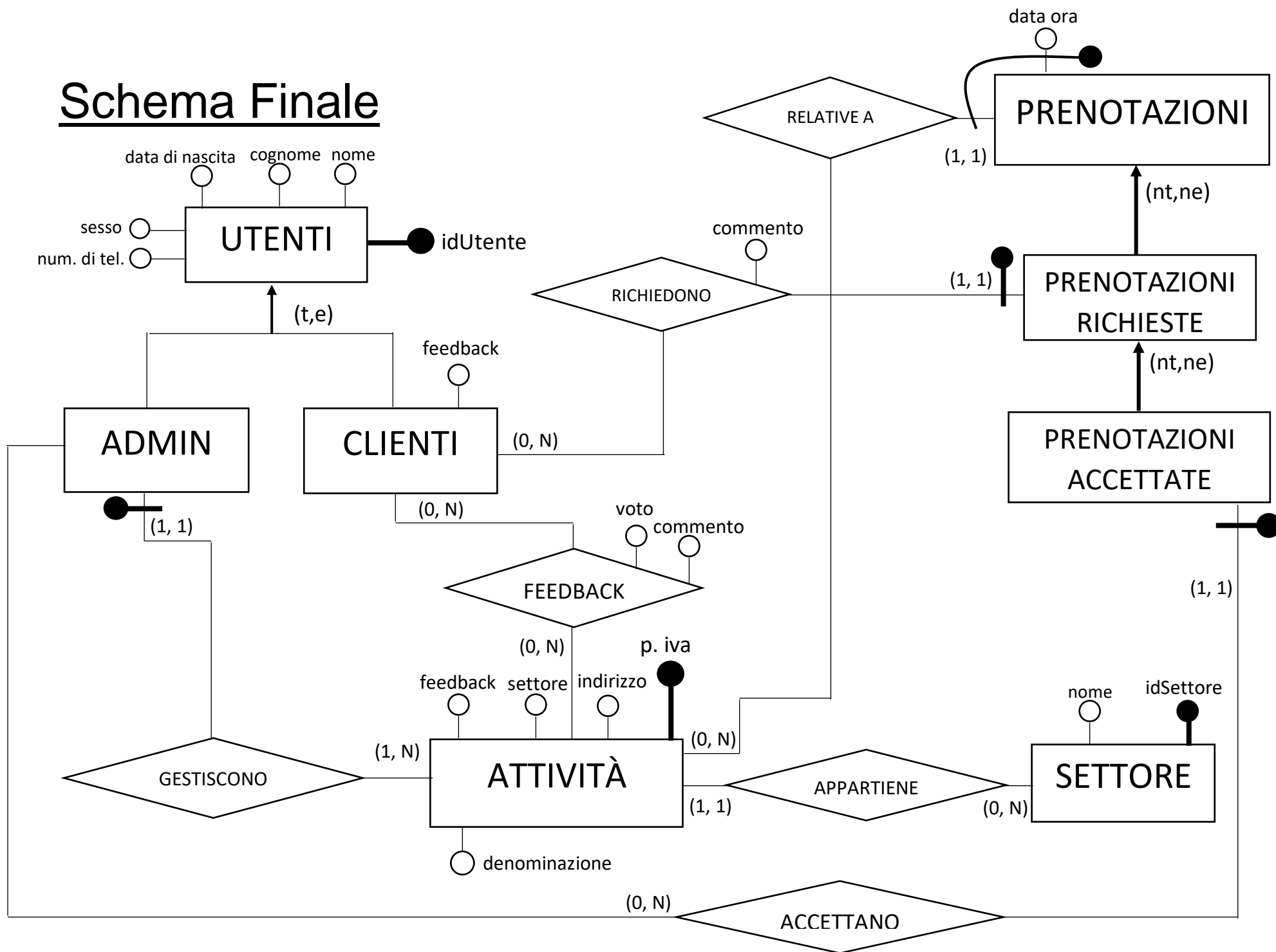
Schema scheletro



Schema Intermedio



Schema Finale



Dizionario dati – Entità

Entità	Descrizione	Attributi	Identificatore
Utenti	Utente registrato che effettua o gestisce prenotazioni	idUtente, nome, cognome, data di nascita, sesso, telefono, email, password	idUtente
Admin	Utente amministratore di una attività che può accettare le prenotazioni della stessa	<i>vedi</i> Utenti	<i>vedi</i> Utenti
Clienti	Cliente registrato che effettua prenotazioni	<i>come</i> Utenti + feedback	<i>vedi</i> Utenti
Attività	Attività convenzionata che mette a disposizione gli orari da prenotare online	p. iva, denominazione, indirizzo, settore, feedback	p. iva
Prenotazioni	Prenotazioni messe a disposizione da un'attività	data ora, commento	data ora, Attività (tramite relazione Relativa A)
Settore	Il settore di una o più attività	idSettore, nome	idSettore
Prenotazioni Richieste	Prenotazione che un cliente ha richiesto	<i>vedi</i> Prenotazioni	come Prenotazioni + Clienti (tramite relazione Richiedono)
Prenotazioni Accettate	Prenotazione che un admin ha confermato	<i>vedi</i> Prenotazioni	come Prenotazioni Richieste + Admin (tramite relazione Accettano)

Dizionario Dati – Relazioni

Relazione	Entità Partecipanti	Descrizione	Attributi
Richiedono	Clienti, Prenotazioni Richieste	Il cliente richiede una prenotazione	Commento
Gestiscono	Admin, Attività	Ogni attività è gestita da uno o più amministratori	
Feedback	Clienti, Attività	I clienti lasciano feedback alle attività e viceversa	Voto, Commento
Appartiene	Attività, Settore	Associa ogni attività con il suo settore d'appartenenza	
Accettano	Admin, Prenotazioni Accettate	Ogni prenotazione deve essere accettata da un amministratore	
Relativa a	Attività, Prenotazioni	Associa ogni prenotazione all'attività che la mette a disposizione	

Vincoli non esprimibili dallo schema E-R

Admin in Prenotazioni Accettate: solo gli admin dell'attività relativa alla prenotazione possono accettare la prenotazione stessa.

Cliente in Prenotazioni Richieste: un Cliente può richiedere solo una prenotazione per un'attività nella stessa data o una data passata.

Feedback: il voto nella relazione Feedback deve essere un intero compreso tra 1 e 5.

Data Ora in Prenotazioni: non si può inserire una data passata come data ora

Specifiche sulle operazioni

Le operazioni che si intendono effettuare sono:

1. Registrazione di un nuovo cliente (100 al giorno)
2. Inserimento di nuove prenotazioni (700 al giorno)
3. Richiesta di prenotazioni (800 al giorno)
4. Accettazione di prenotazioni (450 al giorno)
5. Cancellazione di prenotazione (20 al giorno)
6. Lasciare un feedback (50 al giorno)
7. Modifica dati utente (15 al giorno)
8. Registrazione di una nuova attività (5 al giorno)
9. Registrazione di nuovi admin (5 al giorno)
10. Cancellazione di un utente (5 al giorno)
11. Statistica migliori attività (2 al mese)
12. Eliminazione prenotazioni non accettate e scadute (1 al giorno)
13. Richiesta dati utente (compreso feedback) (500 al giorno)
14. Richiesta dati attività (compreso feedback) (1000 al giorno)
15. Elenco feedback di un cliente/attività (200 al giorno)
16. Trovare tutte le attività di un settore (5 al giorno)

TAVOLA DEI VOLUMI

Concetto	Tipo	Volume
Utenti	E	52.500
Admin	E	2.500
Clienti	E	50.000
Attività	E	2.000
Prenotazioni	E	8.000
Settore	E	50
Prenotazioni Richieste	E	6.000
Prenotazioni Accettate	E	4.000
Richiedono	R	50.000
Gestiscono	R	2.500
Feedback	R	500
Appartiene	R	2.000
Accettano	R	6.000
Relativa a	R	8.000

TAVOLA DELLE FREQUENZE

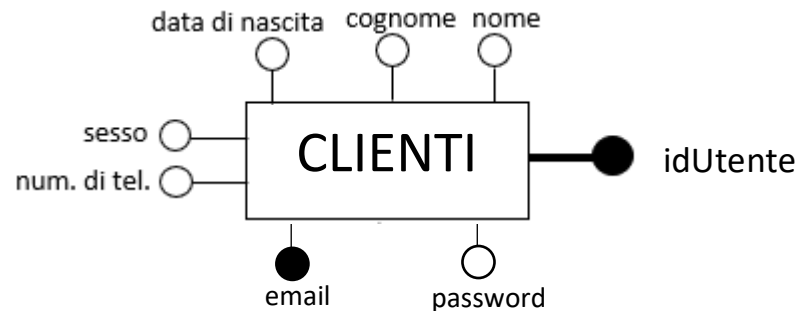
Operazione	Descrizione	Frequenza	Tipo
O1	Registrazione di un nuovo cliente	100 al giorno	I
O2	Inserimento di nuove prenotazioni	700 al giorno	I
O3	Richiesta di prenotazioni	800 al giorno	I
O4	Accettazione di prenotazioni	450 al giorno	I
O5	Cancellazione di prenotazione	20 al giorno	I
O6	Lasciare un feedback	50 al giorno	I
O7	Modifica dati utente	15 al giorno	I
O8	Registrazione di una nuova attività	5 al giorno	I
O9	Registrazione di nuovi admin	5 al giorno	I
O10	Cancellazione di un utente	5 al giorno	I
O11	Statistica migliori attività	2 al mese	B
O12	Eliminazione prenotazioni non accettate e scadute	1 al giorno	B
O13	Richiesta dati utente (compreso feedback)	500 al giorno	I
O14	Richiesta dati attività (compreso feedback)	1000 al giorno	I
O15	Elenco dei feedback di un cliente/attività	200 al giorno	I
O16	Elenco di tutte le attività di un dato settore	5 al giorno	I

Traduzione verso il modello relazionale: Analisi delle ridondanze

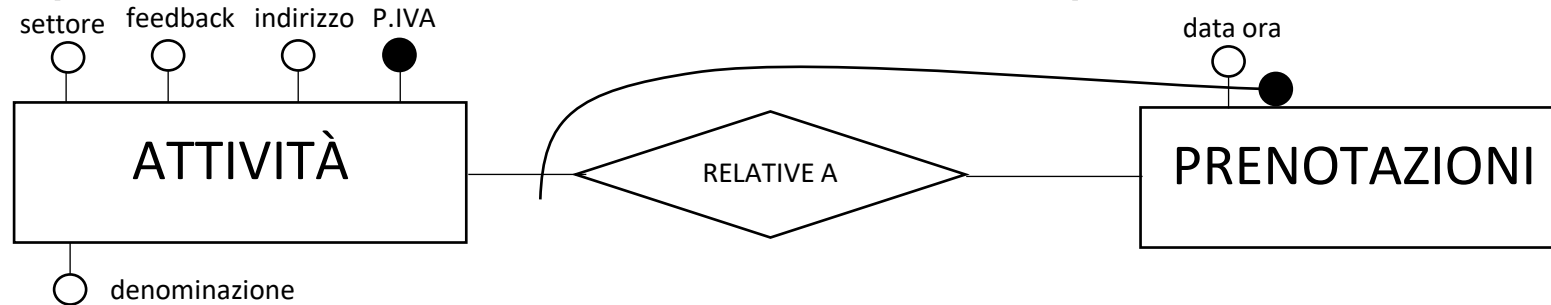
- L'attributo feedback in Attività e Cliente è ridondante in quanto è ricavabile dalla media voto della relazione Feedback
- L'attributo settore in Attività è ridondante in quanto si può ricavare dalla relazione Appartiene

Schemi delle operazioni

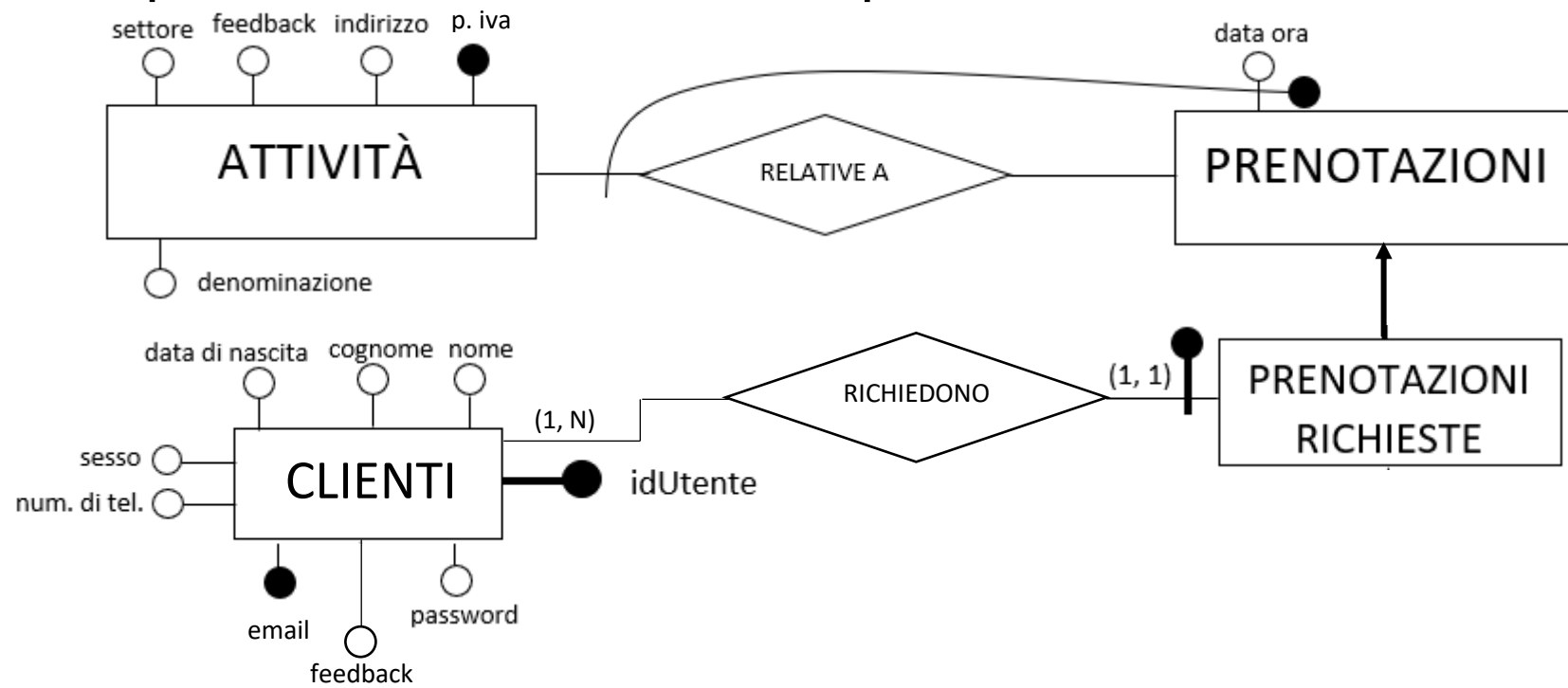
- Operazione 1 – Registrazione di un nuovo cliente



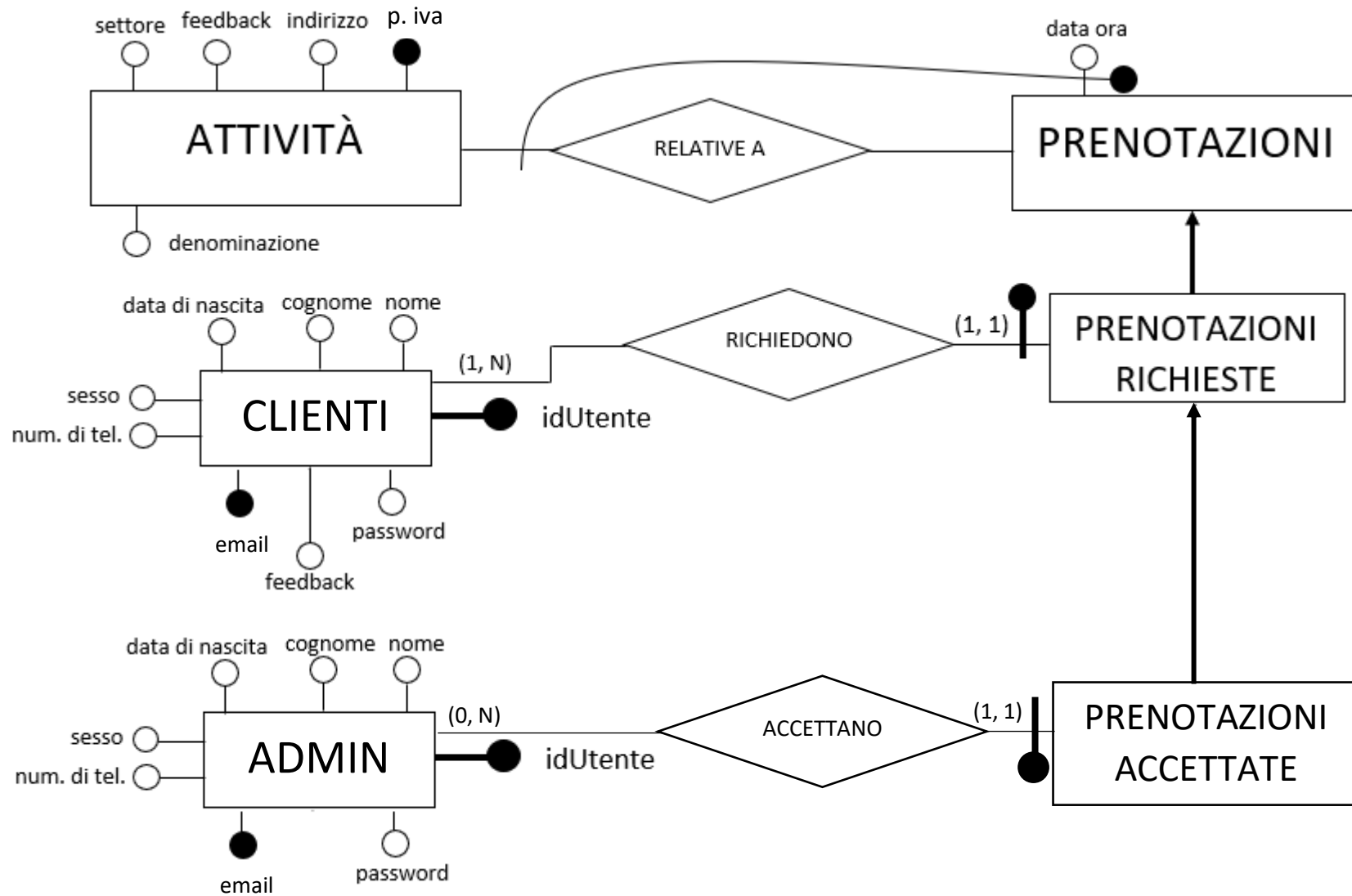
- Operazione 2 – Inserimento di nuove prenotazioni



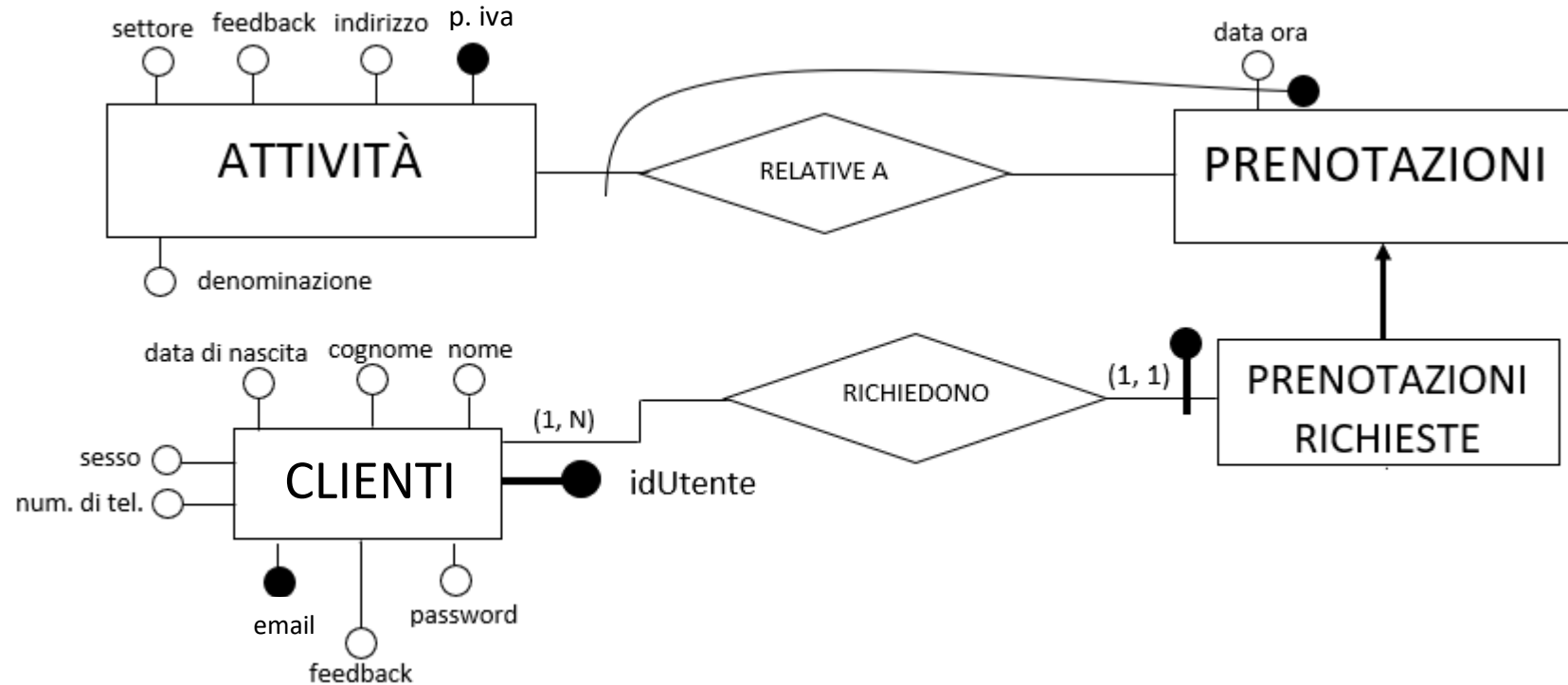
- Operazione 3 – Richiesta di prenotazioni



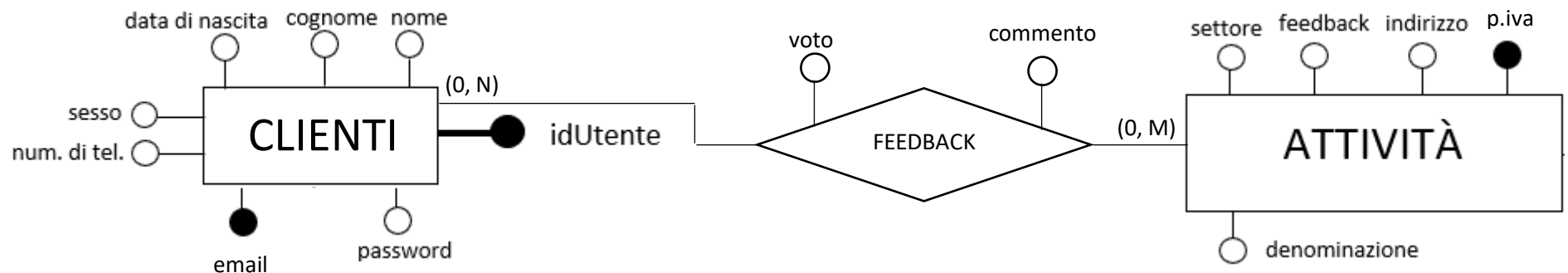
- Operazione 4 – Accettazione di prenotazioni



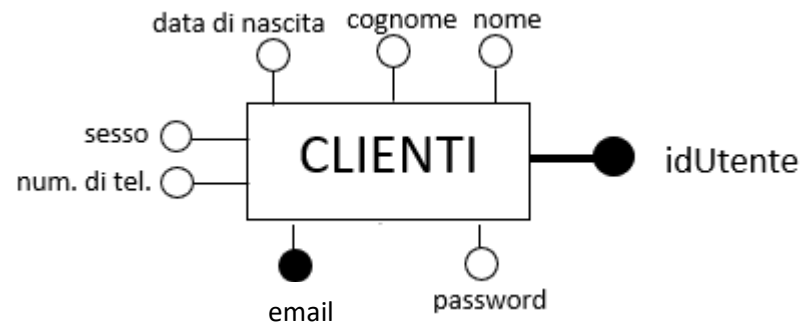
- Operazione 5 – Cancellazione di prenotazione



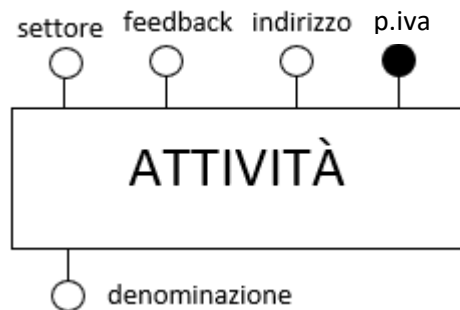
- Operazione 6 – Lasciare un feedback



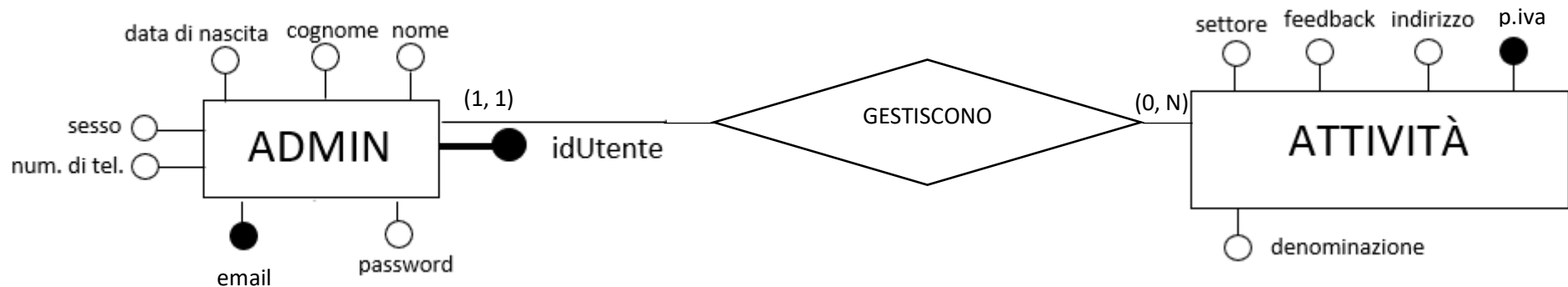
- Operazione 7 – Modifica dei dati utente



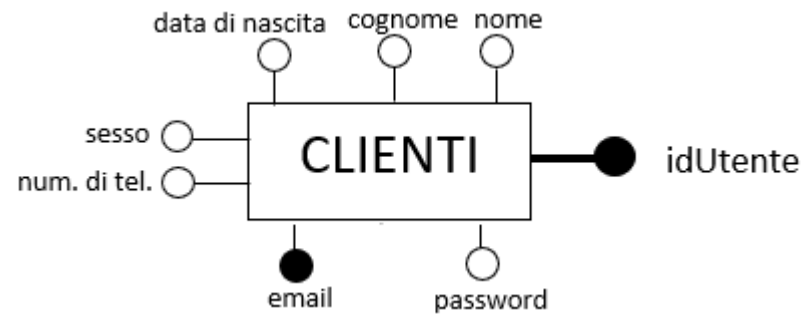
- Operazione 8 – Registrazione nuova attività



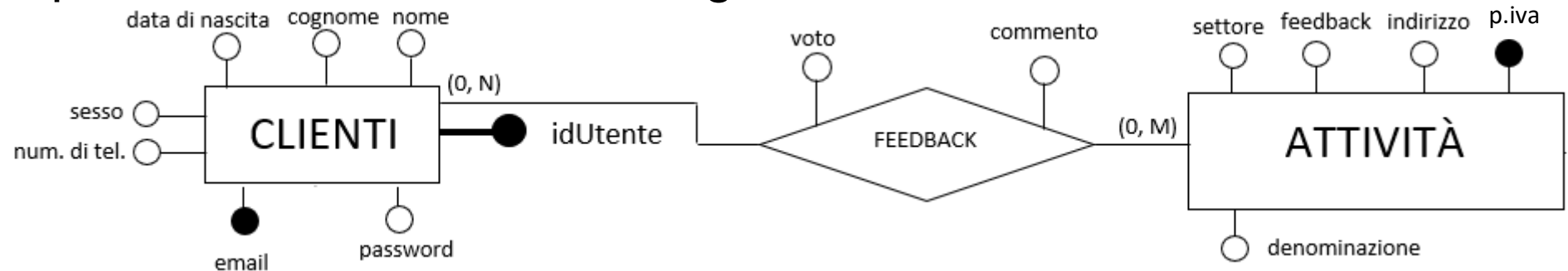
- Operazione 9 – Registrazione nuovi admin



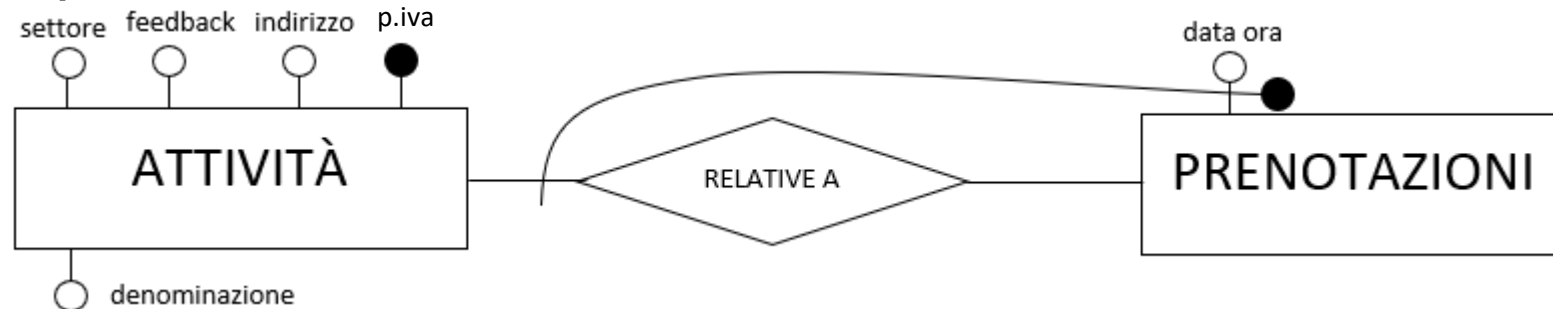
- Operazione 10 – Cancellazione di un utente



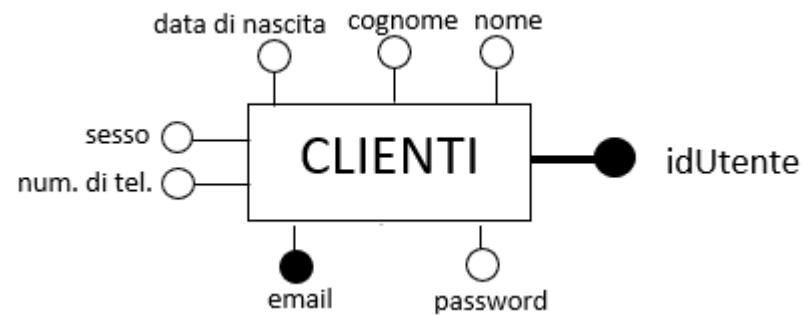
- Operazione 11 – Statistica migliori attività



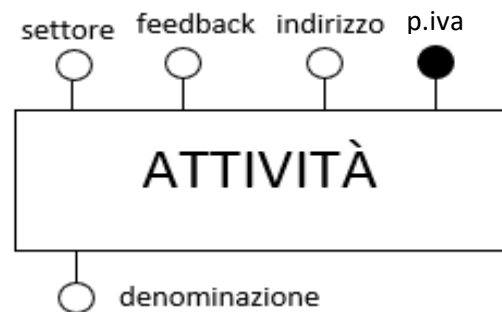
- Operazione 12 – Eliminazione prenotazioni non accettate e scadute



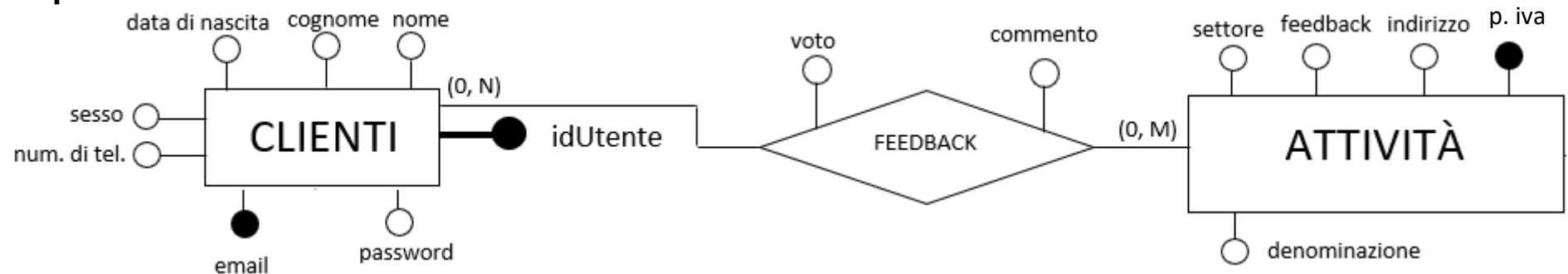
- Operazione 13 – Richiesta dati utente



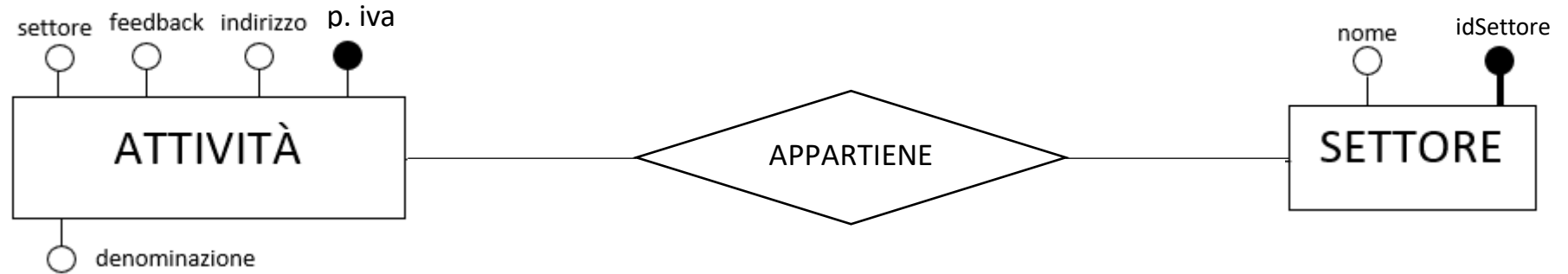
- Operazione 14 – Richiesta dati attività



- Operazione 15 – Elenco feedback di un cliente/attività



- Operazione 16 – Elenco di tutte le attività di un determinato settore



Valutazione del costo delle ridondanze

- Se vogliamo mantenere l'attributo feedback in Clienti questo comporta 4 byte di memoria in più. Avendo stimato in 50.000 i Clienti si ha $4 * 50.000 = 200.000$ byte di memoria in più.
Le operazioni coinvolte sono O6 e O13, O15
- Se vogliamo mantenere l'attributo feedback in Attività questo comporta 4 byte di memoria in più. Avendo stimato in 2.000 Le Attività si ha $4 * 2.000 = 8.000$ byte di memoria in più.
Le operazioni coinvolte sono O6, O11, O14 e O15
- Se vogliamo mantenere l'attributo settore in Attività questo comporta 16 byte di memoria in più. Avendo stimato in 2.000 Le Attività si ha $16 * 2.000 = 32.000$ byte di memoria in più.
Le operazioni coinvolte sono O8, O14 e O16

❖ Valutazione O6 con ridondanza feedback su Clienti

Descrizione	E/R	Accessi	Tipo
Feedback	R	1	S
Clienti	E	1	S

Considerando $1S = 2L$ abbiamo $(2*2) * 50 = 200$ accessi

❖ Valutazione O6 senza ridondanza feedback su Clienti

Descrizione	E/R	Accessi	Tipo
Feedback	R	1	S

Considerando $1S = 2L$ abbiamo $(1*2) * 50 = 100$ accessi

❖ Valutazione O13 con ridondanza feedback su Clienti

Descrizione	E/R	Accessi	Tipo
Clienti	E	1	L

Quindi abbiamo $1 * 500 = 500$ accessi

❖ Valutazione O13 senza ridondanza feedback su Clienti

Stimiamo che in media un cliente abbia 10 feedback.

Descrizione	E/R	Accessi	Tipo
Feedback	R	10	L
Clienti	E	1	L

Quindi abbiamo $(10+1) * 500 = 5.500$ accessi

❖ Valutazione O15 con ridondanza feedback su Clienti

Descrizione	E/R	Accessi	Tipo
Feedback	R	10	L

Quindi abbiamo $10 * 200 = 2.000$ accessi

❖ Valutazione O15 senza ridondanza feedback su Clienti

Descrizione	E/R	Accessi	Tipo
Feedback	R	10	L

Quindi abbiamo $10 * 200 = 2.000$ accessi

Mantenendo la ridondanza abbiamo 2.700 accessi

Togliendo la ridondanza abbiamo 7.600 accessi

Con la ridondanza risparmiamo 4.900 accessi a fronte di soli 200KB di memoria in più.

**Conviene mantenere la ridondanza
dell'attributo feedback in Clienti**

❖ Valutazione O6 con ridondanza feedback su Attività

Descrizione	E/R	Accessi	Tipo
Feedback	R	1	S
Attività	E	1	S

Considerando $1S = 2L$ abbiamo $(2*2) * 50 = 200$ accessi

❖ Valutazione O6 senza ridondanza feedback su Attività

Descrizione	E/R	Accessi	Tipo
Feedback	R	1	S

Considerando $1S = 2L$ abbiamo $(1*2) * 50 = 100$ accessi

❖ Valutazione O14 con ridondanza feedback su Attività

Descrizione	E/R	Accessi	Tipo
Attività	E	1	L

Quindi abbiamo $1 * 1000 = 1000$ accessi

❖ Valutazione O14 senza ridondanza feedback su Attività

Stimiamo che in media un'attività abbia 100 feedback.

Descrizione	E/R	Accessi	Tipo
Feedback	R	100	L
Attività	E	1	L

Quindi abbiamo $(100+1) * 500 = 50.500$ accessi

❖ Valutazione O15 con ridondanza feedback su Attività

Descrizione	E/R	Accessi	Tipo
Feedback	R	100	L

Quindi abbiamo $10 * 200 = 20.000$ accessi

❖ Valutazione O15 senza ridondanza feedback su Attività

Descrizione	E/R	Accessi	Tipo
Feedback	R	10	L

Quindi abbiamo $100 * 200 = 20.000$ accessi

L'Operazione 11 non viene considerata in quanto di tipo Batch dunque il suo costo può essere trascurato.

Mantenendo la ridondanza abbiamo 21.200 accessi

Togliendo la ridondanza abbiamo 70.600 accessi

Con la ridondanza risparmiamo 49.400 accessi a fronte di soli 8KB di memoria in più.

**Conviene mantenere la ridondanza
dell'attributo feedback in Attività**

❖ Valutazione O8 con ridondanza settore su Attività

Descrizione	E/R	Accessi	Tipo
Attività	E	1	S

Quindi abbiamo $(1*2)*5 = 10$ accessi

❖ Valutazione O8 senza ridondanza settore su Attività

Descrizione	E/R	Accessi	Tipo
Attività	E	1	S
Appartiene	R	1	L
Settore	E	1	L

Quindi abbiamo $((1*2)+1+1)*5 = 20$ accessi

❖ Valutazione O14 con ridondanza settore su Attività

Descrizione	E/R	Accessi	Tipo
Attività	E	1	L

Quindi abbiamo $1 \cdot 1000 = 1.000$ accessi

❖ Valutazione O14 senza ridondanza settore su Attività

Descrizione	E/R	Accessi	Tipo
Attività	E	1	L
Appartiene	R	1	L
Settore	E	1	L

Quindi abbiamo $(1+1+1) \cdot 1000 = 3.000$ accessi

❖ Valutazione O16 con ridondanza settore su Attività

Descrizione	E/R	Accessi	Tipo
Attività	E	2000	L

Quindi abbiamo $2000 * 5 = 10.000$ accessi

❖ Valutazione O16 senza ridondanza settore su Attività

Descrizione	E/R	Accessi	Tipo
Attività	E	2000	L
Appartiene	R	2000	L
Settore	E	2000	L

Quindi abbiamo $(2000 + 2000 + 2000) * 5 = 30.000$ accessi

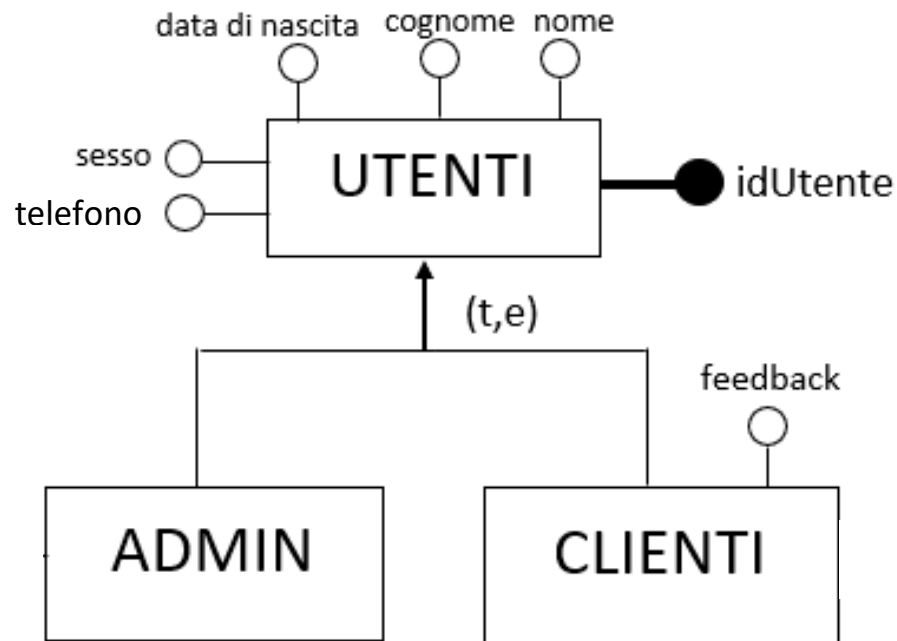
Mantenendo la ridondanza abbiamo 11.010 accessi

Togliendo la ridondanza abbiamo 33.020 accessi

Con la ridondanza risparmiamo 22.010 accessi a fronte di soli 32KB di memoria in più.

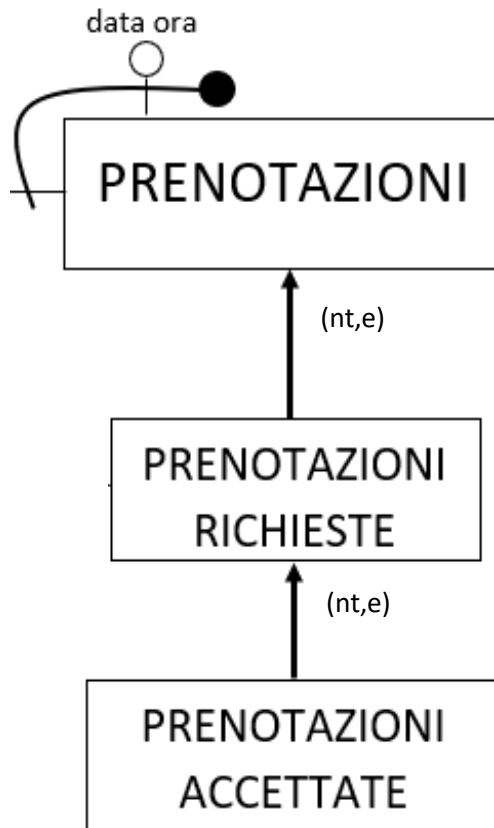
**Conviene mantenere la ridondanza
dell'attributo settore in Attività**

Traduzione verso il modello relazionale: ELIMINAZIONE DELLE GERARCHIE



Per questa gerarchia si effettuerà un collasso verso il basso poiché nel database si accederà separatamente alle entità figlie.

La risoluzione della gerarchia avverrà con la creazione di due entità separate Admin e Clienti.



Per questa gerarchia si effettuerà un collasso verso l'alto.

La risoluzione della gerarchia avverrà con la creazione un'unica entità Prenotazioni.

L'entità creata avrà 4 campi: la data e l'ora, l'attività che la mette a disposizione, l'eventuale cliente che ha richiesto la prenotazione e l'eventuale admin che ha accettato la prenotazione.

I valori nulli sono al massimo due e quindi trascurabili.

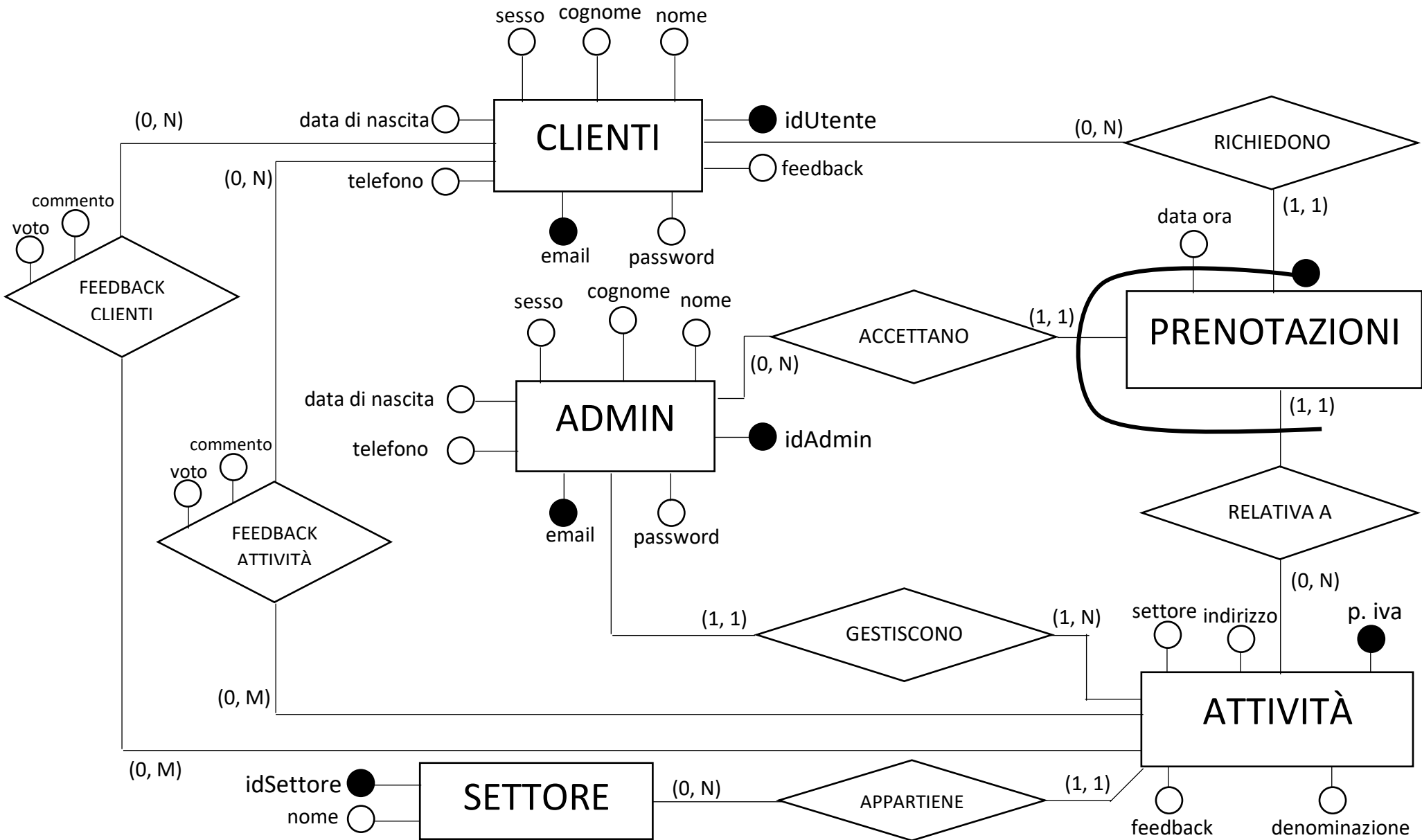
Traduzione verso il modello relazionale:

Partizionamento/Accorpamento di Entità e Associazioni

L'associazione Feedback si partiziona in due entità:

1. FeedbackCliente che indica tutti i feedback che le aziende lasciano ad i clienti
2. FeedbackAttività che indica tutti i feedback che i clienti lasciano alle attività

SCHEMA RISTRUTTURATO



Modello Logico

Utenti(idUtente, nome, cognome, sesso, data_di_nascita, telefono, email, password, feedback)

Admin(idAdmin, nome, cognome, sesso, data_di_nascita, telefono, email, password, idAttività)

Attività(P.IVA, denominazione, indirizzo, settore, feedback, idSettore)

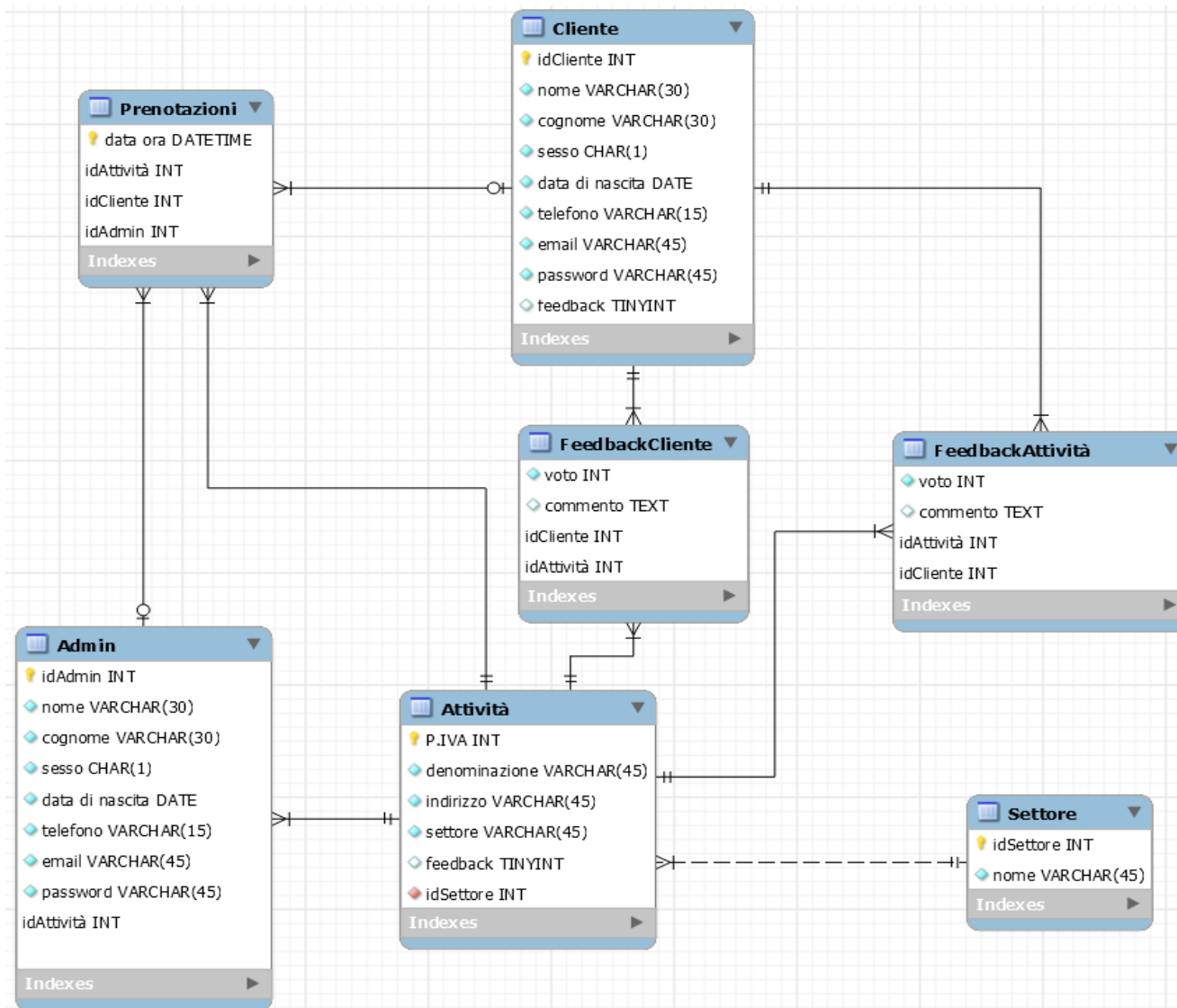
Settore(idSettore, nome)

Prenotazioni(data ora, attività, cliente, admin)

FeedbackCliente(idUtente, idAttività, voto, commento)

FeedbackAttività(idAttività, idUtente, voto, commento)

SCHEMA FISICO



CREAZIONE DEL DATABASE

```
CREATE DATABASE IF NOT EXISTS `GestionePrenotazioni`;
```

```
CREATE TABLE IF NOT EXISTS
```

```
`GestionePrenotazioni`.`Cliente` (  
  `idCliente` INT NOT NULL AUTO_INCREMENT,  
  `nome` VARCHAR(30) NOT NULL,  
  `cognome` VARCHAR(30) NOT NULL,  
  `sex` CHAR(1) NOT NULL,  
  `data di nascita` DATE NOT NULL,  
  `telefono` VARCHAR(15) NOT NULL,  
  `email` VARCHAR(45) NOT NULL UNIQUE,  
  `password` VARCHAR(45) NOT NULL,  
  `feedback` FLOAT NULL,  
  PRIMARY KEY (`idCliente`))
```

```
CREATE TABLE IF NOT EXISTS
```

```
`GestionePrenotazioni`.`Attività` (  
  `P.IVA` CHAR(11) NOT NULL,  
  `denominazione` VARCHAR(45) NOT NULL,  
  `indirizzo` VARCHAR(45) NOT NULL,  
  `settore` VARCHAR(45) NOT NULL,  
  `feedback` FLOAT NULL,  
  `idSettore` INT NOT NULL,  
  PRIMARY KEY (`P.IVA`),  
  CONSTRAINT `fk_Settore`  
    FOREIGN KEY (`idSettore`)  
    REFERENCES  
  `GestionePrenotazioni`.`Settore` (`idSettore`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)
```

CREATE TABLE IF NOT EXISTS

```
`GestionePrenotazioni`.`Admin` (  
  `idAdmin` INT NOT NULL AUTO_INCREMENT,  
  `nome` VARCHAR(30) NOT NULL,  
  `cognome` VARCHAR(30) NOT NULL,  
  `sex` CHAR(1) NOT NULL,  
  `data di nascita` DATE NOT NULL,  
  `telefono` VARCHAR(15) NOT NULL,  
  `email` VARCHAR(45) NOT NULL,  
  `password` VARCHAR(45) NOT NULL,  
  `idAttività` CHAR(11) NOT NULL,  
  PRIMARY KEY (`idAdmin`, `idAttività`),  
  CONSTRAINT `fk_Actività`  
    FOREIGN KEY (`idAttività`)  
    REFERENCES  
`GestionePrenotazioni`.`Attività` (`P.IVA`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)
```

CREATE TABLE IF NOT EXISTS

```
`GestionePrenotazioni`.`Prenotazioni` (  
  `data ora` DATETIME NOT NULL,  
  `idAttività` INT NOT NULL,  
  `idCliente` INT NULL,  
  `idAdmin` CHAR(11) NULL,  
  PRIMARY KEY (`data ora`, `idAttività`, `idCliente`,  
  `idAdmin`),  
  CONSTRAINT `fk_Cliente`  
    FOREIGN KEY (`idCliente`)  
    REFERENCES `GestionePrenotazioni`.`Cliente`  
  (`idCliente`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION,  
  CONSTRAINT `fk_Actività`  
    FOREIGN KEY (`idAttività`)  
    REFERENCES `GestionePrenotazioni`.`Attività` (`P.IVA`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION,  
  CONSTRAINT `fk_Admin`  
    FOREIGN KEY (`idAdmin`)  
    REFERENCES `GestionePrenotazioni`.`Admin`  
  (`idAdmin`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)
```

CREATE TABLE IF NOT EXISTS

```
`GestionePrenotazioni`.`FeedbackCliente` (  
  `voto` TINYINT NOT NULL,  
  `commento` TEXT NULL,  
  `idCliente` INT NOT NULL,  
  `idAttività` INT NOT NULL,  
  PRIMARY KEY (`idCliente`, `idAttività`),  
  CONSTRAINT `fk_Cliente`  
    FOREIGN KEY (`idCliente`)  
      REFERENCES  
`GestionePrenotazioni`.`Cliente` (`idCliente`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION,  
  CONSTRAINT `fk_Actività`  
    FOREIGN KEY (`idAttività`)  
      REFERENCES  
`GestionePrenotazioni`.`Attività` (`P.IVA`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)
```

CREATE TABLE IF NOT EXISTS

```
`GestionePrenotazioni`.`FeedbackAttività` (  
  `voto` TINYINT NOT NULL,  
  `commento` TEXT NULL,  
  `idAttività` INT NOT NULL,  
  `idCliente` INT NOT NULL,  
  PRIMARY KEY (`idCliente`, `idAttività`),  
  CONSTRAINT `fk_Actività`  
    FOREIGN KEY (`idAttività`)  
      REFERENCES  
`GestionePrenotazioni`.`Attività` (`P.IVA`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION,  
  CONSTRAINT `fk_Cliente`  
    FOREIGN KEY (`idCliente`)  
      REFERENCES  
`GestionePrenotazioni`.`Cliente` (`idCliente`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)
```

CREATE TABLE IF NOT EXISTS

`GestionePrenotazioni`.`**Settore**` (
 `idSettore` INT NOT NULL AUTO_INCREMENT,
 `nome` VARCHAR(45) NOT NULL,
 PRIMARY KEY (`idSettore`))

IMPLEMENTAZIONE OPERAZIONI

Operazione 1 – Registrazione di un nuovo utente

```
INSERT INTO Clienti(nome, cognome, sesso, `data di nascita`, telefono, email, password)  
VALUES('Mario', 'Rossi', 'M', '15-07-1963', '3401974622', 'mario.rossi@mail.it', 'prova1234')
```

Operazione 2 – Inserimento nuove prenotazioni

```
INSERT INTO Prenotazioni(`data ora`, attivita) VALUES('27-02-2019 10:35:00', '08785648206')
```

Operazione 3 – Richiesta di prenotazioni

```
INSERT INTO Prenotazioni(`data ora`, attività, cliente)  
VALUES('27-02-2019 10:35:00', '08785648206', 5)
```

Operazione 4 – Accettazione di prenotazioni

```
UPDATE Prenotazioni  
SET admin = 56  
WHERE `data ora` = '27-02-2019 10:35:00' AND attivita = '08785648206' AND cliente = 5
```

Operazione 5 – Cancellazione di prenotazioni

DELETE FROM Prenotazioni

WHERE `data ora` = '27-02-2019 10:35:00' AND attivita = '08785648206' AND cliente = 5

Operazione 6 – Lasciare un feedback

- Se un cliente lascia un feedback ad un'attività

```
INSERT INTO FeedbackAttivita(idUtente, idAttivita, voto, commento)
VALUES (22, '08785648206', 4, 'Il cibo era molto buono ma abbiamo atteso po'' troppo')
```

- Se un'attività lascia un feedback ad un cliente

```
INSERT INTO FeedbackCliente(idAttivita, idUtente, voto)
VALUES ('08785648206', 22, 5)
```

Operazione 7 – Modifica dati utente

UPDATE Clienti

SET telefono = '3914750994'

WHERE idUtente = 84

Operazione 8 – Registrazione di una nuova attività

INSERT INTO `Attività`(`P.IVA`, denominazione, settore, idSettore)

VALUES ('08714185603', 'Bella Napoli', 'Alimentare', 1)

Operazione 9 – Registrazione nuovi Admin

INSERT INTO Admin(nome, cognome, sesso, `data di nascita`, telefono, email, password, idAttivita) VALUES('Luigi', 'Bianchi', 'M', '04-03-1988', '3719573928', 'luigi.bianchi@mail.it', 'password88', '08785648206')

Operazione 10 – Cancellazione di un utente

DELETE FROM Utente

WHERE email = 'paolo.neri75@mail.com'

Operazione 11 – Statistica migliori attività

Per migliori attività si intendono tutte le attività che hanno una media feedback di almeno 4,5 e almeno 100 voti

SELECT denominazione, indirizzo

FROM Attività A

WHERE feedback >= 4.5 AND (SELECT COUNT(*)

FROM FeedbackAttività FA

WHERE FA.idAttività = A. `P.IVA`) >= 100

Operazione 12 – Eliminazione prenotazioni non accettate e scadute

DELETE FROM Prenotazioni

WHERE admin = NULL AND `data ora` < NOW()

Operazione 13 – Richiesta dati utente

SELECT * FROM Clienti WHERE idCliente = 14

Operazione 14 – Richiesta dati attività

SELECT * FROM Attività WHERE `P.IVA` = '08785648206'

Operazione 15 – Elenco feedback

- Di un dato Cliente

```
SELECT * FROM FeedbackClienti WHERE idCliente = 321
```

- Di una data Attività

```
SELECT * FROM FeedbackAttività WHERE idAttività = '08785648206'
```

Operazione 16 – Trovare tutte le attività di un settore

```
SELECT *
```

```
FROM Attività
```

```
WHERE idSettore = 11
```

VINCOLI D'INTEGRITÀ

1. Solo gli admin dell'attività relativa alla prenotazione possono accettare la prenotazione stessa.
2. Un Cliente può richiedere solo una prenotazione per un'attività nella stessa data.
3. Un Cliente non può effettuare una prenotazione per una data passata.
4. Un Admin non può accettare una prenotazione per una data passata.
5. Il voto in FeedbackClienti e FeedbackAttività deve essere un intero compreso tra 1 e 5.

Vincolo 1 – Solo Admin dell'Attività possono accettare le prenotazioni

```
CREATE TRIGGER prenotazioni_admin
BEFORE UPDATE OF Admin ON Prenotazioni
FOR EACH ROW
BEGIN
    IF(New.admin NOT IN (SELECT A.idAdmin
                        FROM Admin A
                        WHERE A.idAttività = New.idAttività)
    ) THEN SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Non sei autorizzato ad accettare questa prenotazione';
    ENDIF;
END;
```

Vincolo 2 – Prenotazioni dello stesso utente nella stessa data

```
CREATE TRIGGER cliente_prenotazioni_giorno
BEFORE INSERT ON Prenotazioni
FOR EACH ROW
BEGIN
    IF( EXISTS( SELECT *
                FROM Prenotazioni
                WHERE cliente = New.cliente AND `data ora` = New.`data ora`
                  AND attivita = New.attivita;
            )
    ) THEN SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'Esiste già una prenotazione per questa data';
    ENDIF;
END;
```


Vincolo 3 – Effettuare prenotazioni in data passata

```
CREATE TRIGGER cliente_prenotazioni_passate
BEFORE INSERT ON Prenotazioni
FOR EACH ROW
BEGIN
    IF( New.`data ora` < CURRENT_TIMESTAMP
        ) THEN SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'Non puoi effettuare prenotazioni per una data passata';
    ENDIF;
END;
```

Vincolo 4 – Accettare prenotazioni in data passata

```
CREATE TRIGGER admin_prenotazioni_passate
BEFORE UPDATE OF admin ON Prenotazioni
FOR EACH ROW
BEGIN
    IF( New.`data ora` < CURRENT_TIMESTAMP
        ) THEN SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'Non puoi accettare prenotazioni per una data passata';
    ENDIF;
END;
```

Vincolo 5 – voto feedback tra 1 e 5

```
ALTER TABLE FeedbackCliente  
ADD CONSTRAINT constr_voto_fed_cli  
CHECK(voto BETWEEN 1 AND 5)
```

```
ALTER TABLE FeedbackAttività  
ADD CONSTRAINT constr_voto_fed_att  
CHECK(voto BETWEEN 1 AND 5)
```

Vincolo 6.1 – aggiornare attributo feedback in Cliente

```
CREATE TRIGGER aggiorna_feedback_cliente
AFTER INSERT ON FeedbackCliente
FOR EACH ROW
BEGIN
    DECLARE X number;
    DECLARE somma number;

    SELECT COUNT(*) INTO X, SUM(voto) INTO somma
    FROM FeedbackClienti
    WHERE idCliente = New.idCliente

    UPDATE Cliente
    SET feedback = (somma+New.voto)/(X+1)
END;
```

Vincolo 6.2 – aggiornare attributo feedback in Attività

```
CREATE TRIGGER aggiorna_feedback_attività
AFTER INSERT ON FeedbackAttività
FOR EACH ROW
BEGIN
    DECLARE X number;
    DECLARE somma number;

    SELECT COUNT(*) INTO X, SUM(voto) INTO somma
    FROM FeedbackAttività
    WHERE idAttività = New.idAttività

    UPDATE Attività
    SET feedback = (somma+New.voto)/(X+1)
END;
```