

Elaborato corso di Basi di Dati

A.A. 2021-22

GESTIONE DEL TRASPORTO PUBBLICO SU GOMMA



**azienda
napoletana
mobilità s.p.a.**

Realizzato da: Francesco Pio Manna

N46005124

1. Introduzione

Si vuole realizzare una base di dati per la gestione del trasporto pubblico su gomma dell'azienda ANM. Le specifiche utilizzate per la progettazione sono state ottenute mediante colloqui con responsabili dell'azienda. Per descrivere al meglio la realtà di interesse è stato consultato il sito ufficiale dell'azienda, <http://www.anm.it/>. Scopo di questo elaborato è dunque mostrare i processi di creazione, sviluppo e manipolazione dei dati del database relazionale attraverso una serie di funzioni.

2. Specifiche

La fase della specifica dei dati è una fase piuttosto delicata della progettazione, in quanto è il punto di partenza e quindi, in caso di errori nella specifica oppure di mal comprensione della stessa, potrebbe portare a errori di progettazione difficilmente risolvibili nelle fasi successive. Alcune regole generali per ottenere una specifica precisa e priva di ambiguità:

- Evitare termini troppo generici o troppo precisi. Mantenere un livello di astrazione costante.
- Evitare l'uso di sinonimi (termini diversi con il medesimo significato) e omonimi (termini uguali con differenti significati). Riferirsi allo stesso concetto sempre allo stesso modo.
- Usare frasi brevi e semplici possibilmente uniformandone la struttura.
- Dividere il testo in paragrafi. Dedicare ogni paragrafo alla descrizione di una specifica entità della realtà modellata. Evidenziare l'entità descritta in ogni paragrafo.

2.1 Specifiche sui dati

Si vuole sviluppare il sistema informativo per la gestione del trasporto pubblico su gomma. Le informazioni da memorizzare appartengono a diversi tipi di persone, in particolare autisti e controllori.

Ogni autista è caratterizzato dal proprio codice fiscale, da un nome e un cognome, dalla data di nascita e dalla data di conseguimento della patente.

I controllori sono caratterizzati anch'essi da un codice fiscale, un nome, un cognome, dalla data di nascita e dalla categoria. La categoria rappresenta il tipo di assicurazione per eventuali danni fisici stipulata con l'agenzia interinale.

L'azienda prevede contratti a tempo indeterminato e determinato, caratterizzati da una data di fine e inizio contratto e un compenso per numero di corse effettuate.

I contratti non vengono stipulati direttamente dall'azienda ma da un'agenzia interinale che mette a disposizione dell'azienda dei responsabili, identificati da una matricola, un nome e un cognome.

Ogni corsa di un autobus deve comprendere un codice numerico che la identifica e un singolo autista e controllore associati a essa.

Le varie linee di autobus sono descritte da un nome, dalla frequenza delle partenze, dalla località di partenza, dalla località di arrivo e da una tipologia (ad esempio, urbane, extraurbane ecc... dove per extraurbana si intende una linea che ha come luogo di partenza o arrivo un comune diverso da quello di Napoli).

Ogni linea è associata a un deposito, che è caratterizzato da un nome e da un affitto mensile pagato dall'azienda.

2.2. Specifiche sulle operazioni

La base di dati dovrà fornire le seguenti operazioni:

01: Visualizzare le linee che stazionano nei vari depositi;

02: Numero di turni effettuati da ogni autista;

03: Autisti che hanno effettuato almeno un turno;

04: Autista che ha effettuato il maggior numero di turni;

05: Nome di tutte le linee che partono da una certa località;

06: Guadagno di un certo autista;

07: Responsabile che ha stipulato almeno un contratto con i controllori.

2.3 Specifiche sul volume di dati

La base di dati verrà progettata tenendo conto del seguente volume di dati:

- Circa 700 autisti;
- Circa 700 controllori;
- 80 linee;
- 15 depositi;
- 2000 corse giornaliere.

3. Analisi dei requisiti

In questo paragrafo ci occupiamo della fase di analisi e ristrutturazione dei requisiti raccolti, producendo un insieme omogeneo non ambiguo e non ridondante di specifiche da utilizzare poi nelle fasi successive.

La raccolta dei requisiti consiste nella individuazione delle caratteristiche statiche (dei dati) e dinamiche (delle operazioni) dell'applicazione da realizzare. I requisiti vengono raccolti in specifiche espresse in linguaggio naturale e per questo motivo spesso ambigue e disorganizzate. L'analisi dei requisiti consiste nel chiarimento e nell'organizzazione delle specifiche raccolte.

In questa fase quindi lo scopo principale è analizzare attentamente le specifiche forniteci e creare un glossario dei termini, una sorta di contenitore di tutti i concetti principali individuati nella specifica. Di fondamentale importanza è anche l'individuazione di eventuali sinonimi e/o omonimi, per evitare qualunque incomprensione e ambiguità, e fornire una breve descrizione di tutti i concetti individuati. L'individuazione dei termini va fatta step-by-step, analizzando attentamente ogni paragrafo che, come precedentemente descritto, se ben congeniato, conterrà esattamente un singolo concetto e/o concetti ad esso correlati e i suoi attributi specifici/caratteristici.

DATI DI CARATTERE GENERALE

*Si vuole sviluppare il sistema informativo per la **gestione del trasporto pubblico su gomma**. Le informazioni da memorizzare appartengono a diversi tipi di persone, in particolare autisti e controllori.*

GLOSSARIO DEI TERMINI

<i>Termine</i>	<i>Descrizione</i>	<i>Sinonimi</i>	<i>Termini collegati</i>
Responsabile	Individuo che stipula i contratti degli autisti e dei controllori	Agenzia interinale	Autista, Controllore
Autista	Persona che guida gli autobus	Conducente	Corsa, Linea
Controllore	Persona preposta al controllo dei titoli di viaggio	Bigliettaio	Corsa, Linea
Corsa	Turno lavorativo dell'autista/controllore	Turno	
Linea	Autobus che collega due località	Autobus	Deposito
Deposito	Luogo in cui parcheggiano gli autobus una volta finita la giornata	Stazionamento	Linea

DATI SUGLI AUTISTI

*Ogni **autista** è caratterizzato dal proprio codice fiscale, da un nome, un cognome, dalla data di nascita e dalla data di conseguimento della patente. Circa 700 autisti presenti nell'azienda.*

DATI SUI CONTROLLORI

*I **controllori** sono caratterizzati anch'essi da un codice fiscale, un nome, un cognome, dalla data di nascita e dalla categoria. La categoria rappresenta il tipo di assicurazioni per eventuali danni fisici stipulata con l'agenzia interinale.*

DATI SUI CONTRATTI DI LAVORO

*L'azienda prevede **contratti** a tempo indeterminato e determinato, caratterizzati da una data di fine e inizio contratto e un compenso per numero di corse effettuate.*

*I contratti non vengono stipulati direttamente dall'azienda ma da un'agenzia interinale che mette a disposizione dell'azienda dei **responsabili**, identificati da una matricola, da un nome e un cognome.*

DATI SULLE CORSE

*Ogni **corsa** di un autobus deve comprendere un codice numerico che la identifica e un singolo autista e controllore associati a essa.*

DATI SULLE LINEE

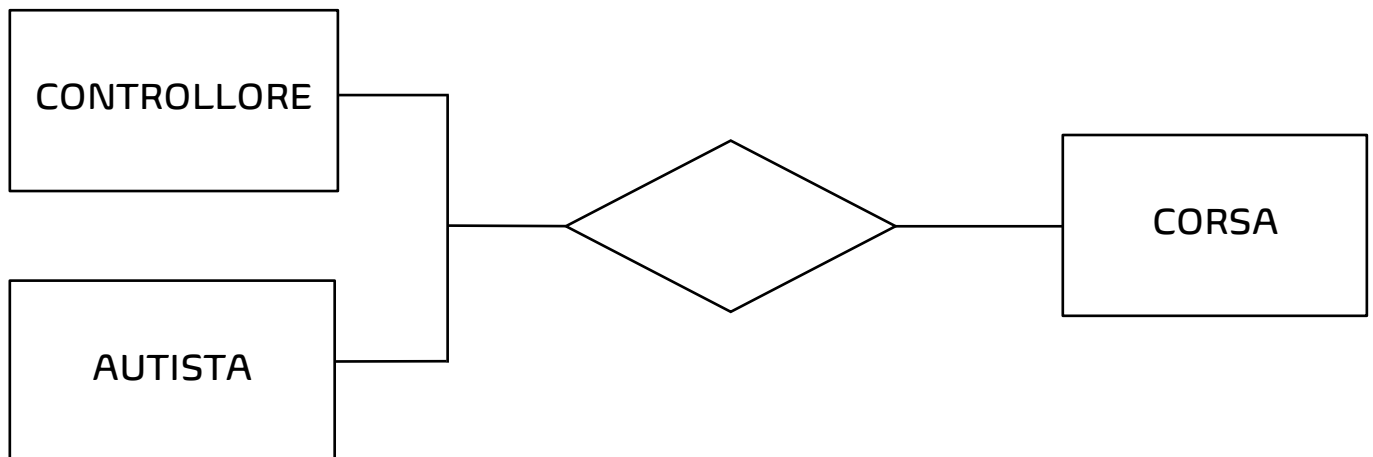
*Le varie **linee** di autobus sono descritte da un nome, dalla frequenza delle partenze, dalla località di partenza, dalla località di arrivo e possono essere urbane oppure extraurbane (per extraurbana si intende una linea che ha come luogo di partenza o arrivo un comune diverso da quello di Napoli).*

DATI SUI DEPOSITI

*Ogni linea è associata a un **deposito**, che è caratterizzato da un nome e da un affitto mensile pagato dall'azienda.*

4. Progettazione concettuale

Sulla base delle specifiche si è arrivati a formulare lo schema scheletro del progetto. Lo schema scheletro è una prima rappresentazione concettuale della realtà di interesse, individuando i concetti principali senza formalizzare associazioni e cardinalità.

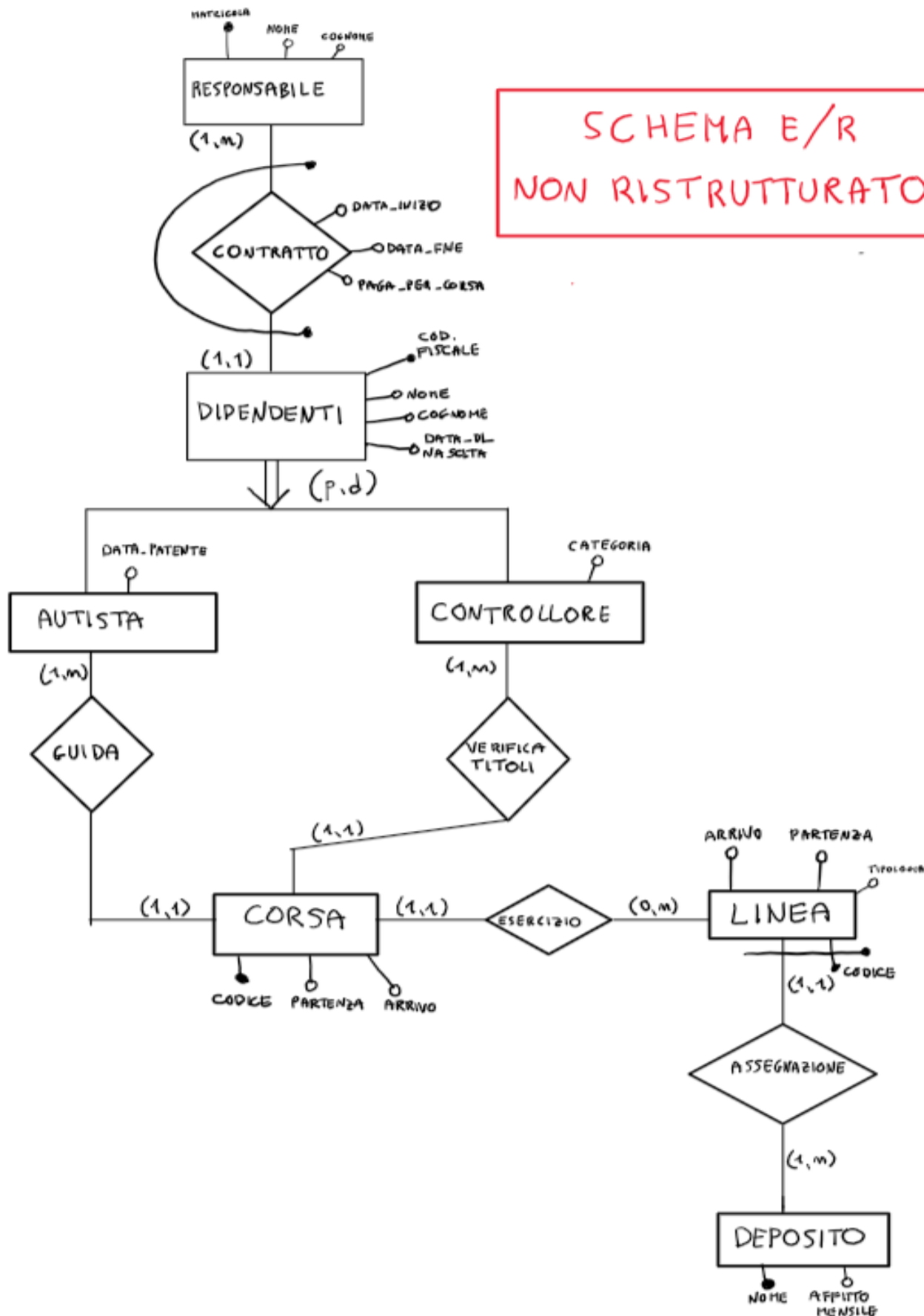


Dopo lo schema scheletro, si procede con la realizzazione del modello concettuale del progetto, senza ristrutturarlo, operazione che verrà poi effettuata nella successiva fase.

Si è deciso di inserire una entità responsabile, che rappresenta la figura professionale dell'agenzia interinale che si interfacerà con i futuri dipendenti. Questa, tramite un contratto, è legata ai dipendenti dell'azienda stessa. Un dipendente, con il responsabile può avere uno e un solo contratto (la cardinalità minima è uno perché altrimenti non sarebbe un dipendente di quell'azienda).

L'entità dipendente si specializza in due entità figlie, autisti e controllori. La generalizzazione risulta essere parziale e disgiunta, in quanto ci sono anche altri tipi di dipendenti nell'azienda non rappresentati da questo modello e perché un'autista non può essere anche un controllore e viceversa. Entrambe sono collegate alle linee su cui poi effettivamente dovranno lavorare tramite le corse (turni). La singola corsa può essere presieduta da un singolo autista e da un singolo controllore, mentre una linea potrebbe non avere dipendenti che hanno lavorato su di essa (si pensi a una linea appena progettata, che prima ha bisogno di un collaudo). La singola linea può stazionare su un singolo deposito, il quale invece può accogliere più

linee. Quindi, il risultato della progettazione concettuale è il seguente:



5. Progettazione logica

La progettazione logica permette la traduzione dello schema concettuale in un modello di rappresentazione dei dati. Il prodotto in questa fase è detto schema logico della base di dati e fa riferimento ad un modello logico dei dati. Tale modello descrive i dati secondo una rappresentazione concreta e indipendente dai dettagli fisici.

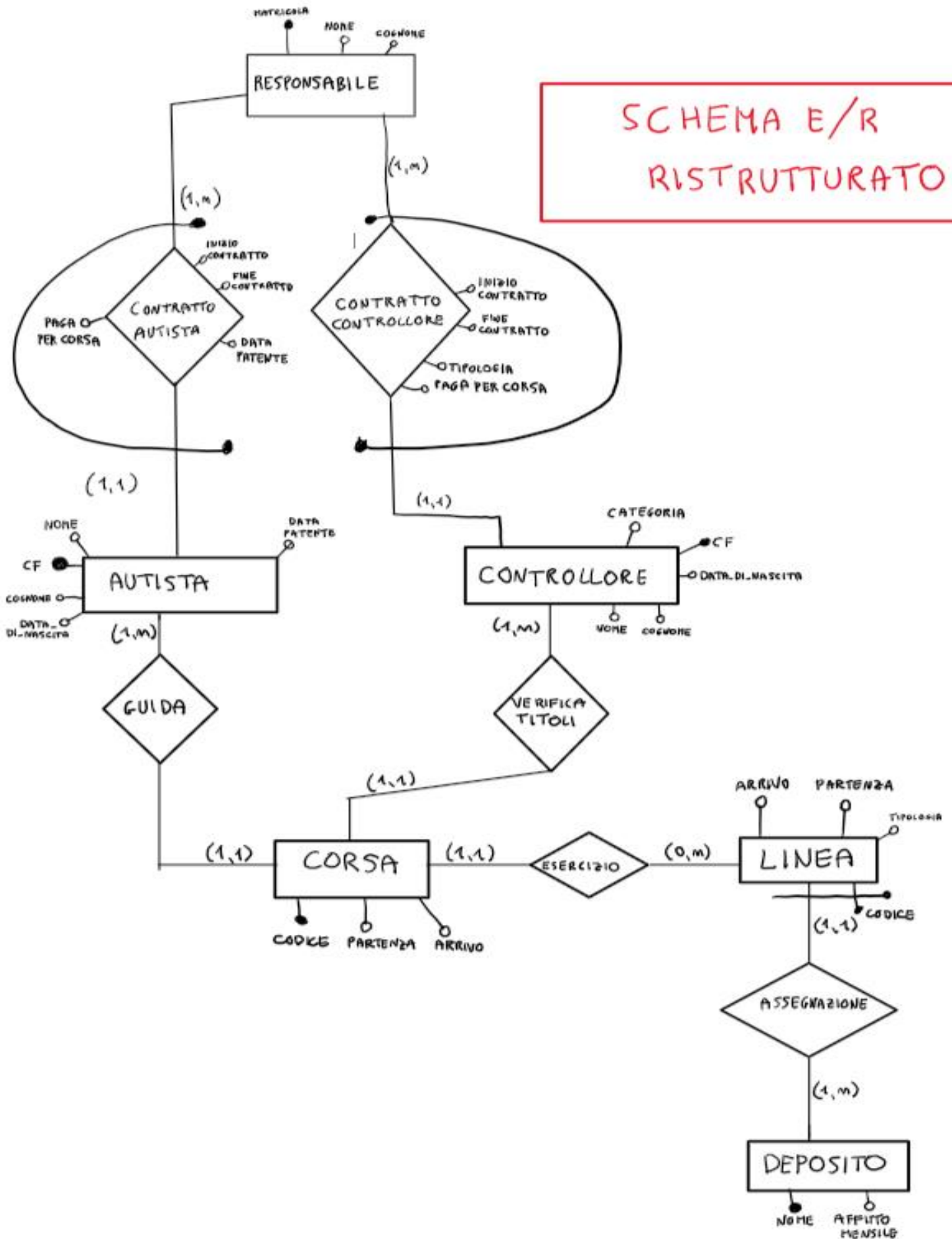
5.1 Ristrutturazione modello E/R

La prima fase della progettazione logica è la cosiddetta ristrutturazione del modello concettuale, in quanto quest'ultimo non può essere tradotto direttamente nel modello logico. La ristrutturazione consiste nell'eliminazione di attributi multivalore e composti e la scelta/aggiunta degli identificatori primari. Ultima fase è l'eliminazione di generalizzazioni/specializzazioni.

Nello schema è presente una specializzazione. Quest'ultima, che specializza i dipendenti, viene risolta accorpando nelle entità figlie l'entità padre, sdoppiando anche il contratto per tipo di dipendente (autista o controllore).

Di seguito si riporta il modello e/r dopo la ristrutturazione:

SCHEMA E/R RISTRUTTURATO



5.2. Traduzione delle entità e delle associazioni

Per le entità *autisti* e *controllori* si è scelto come id il codice fiscale, mentre per le entità *contratto_autista* e *contratto_controllore*, in funzione delle cardinalità, si è scelto di identificarli univocamente con gli identificatori esterni di autista/controllore e del responsabile. Le associazioni *Guida* e *Verifica Titolo* non sono state tradotte in una relazione in quanto hanno la stessa chiave di *Corsa* e lo stesso numero di occorrenze quindi si può procedere alla loro fusione. Essendo le cardinalità 1 a 1 non ci saranno valori nulli. Stesso discorso per l'associazione *Esercizio*. L'associazione *Assegnazione* non è stata tradotta in quanto l'identificatore primario di *Deposito* è un identificatore esterno per *Linea*.

Lo schema logico finale è il seguente:

RESPONSABILI (matricola, nome, cognome)

CONTRATTI_AUTISTI (autista:AUTISTI, responsabile:RESPONSABILI, data_inizio, data_fine, paga_per_corsa)

CONTRATTI_CONTROLLORI (controllore:CONTROLLORI, responsabile:RESPONSABILI, data_inizio, data_fine, paga_per_corsa)

AUTISTI (codice_fiscale, nome, cognome, data_di_nascita, patente)

CONTROLLORI (codice_fiscale, nome, cognome, data_di_nascita, categoria)

CORSE (codice, autista:AUTISTI, controllore:CONTROLLORI, partenza, arrivo, linea:LINEE)

LINEE (nome, partenza, arrivo, frequenza, tipologia, deposito:DEPOSITI)

DEPOSITI (nome, affitto_mensile)

6. Implementazione delle operazioni

Di seguito si riportano i comandi per le creazioni/modifica delle tabelle e la creazione dei vincoli e varie interrogazioni da effettuare sulla base di dati che possono essere utili all'addetto alla gestione del trasporto pubblico.

6.1 Creazione delle tabelle

```
CREATE TABLE `responsabili` (  
  `matricola` int(5) NOT NULL,  
  `cognome` varchar(50) DEFAULT NULL,  
  `nome` varchar(50) DEFAULT NULL,  
  PRIMARY KEY (`matricola`)  
)
```

```
CREATE TABLE `contratti_autisti` (  
  `autista` char(16) NOT NULL,  
  `responsabile` int(5) NOT NULL,  
  `data_inizio` date DEFAULT NULL,  
  `data_fine` date DEFAULT NULL,  
  `paga_per_corsa` float DEFAULT NULL,  
  PRIMARY KEY (`autista`, `responsabile`),  
  KEY `fk_resp` (`responsabile`),  
  CONSTRAINT `fk_autista_contratto` FOREIGN KEY (`autista`) REFERENCES `autisti` (`codice_fiscale`) ON DELETE NO ACTION ON UPDATE NO ACTION,  
  CONSTRAINT `fk_resp` FOREIGN KEY (`responsabile`) REFERENCES `responsabili` (`matricola`) ON DELETE NO ACTION ON UPDATE NO ACTION  
)
```

```
CREATE TABLE `contratti_controllori` (  
  `controllore` char(16) NOT NULL,  
  `responsabile` int(5) NOT NULL,  
  `data_inizio` date DEFAULT NULL,  
  `data_fine` date DEFAULT NULL,  
  `paga_per_corsa` float DEFAULT NULL,  
  PRIMARY KEY (`controllore`, `responsabile`),  
  KEY `fk_resp_contratto` (`responsabile`),  
  CONSTRAINT `fk_resp_contratto` FOREIGN KEY (`responsabile`) REFERENCES `responsabili` (`matricola`) ON DELETE NO ACTION ON UPDATE NO ACTION,  
  CONSTRAINT `fk_controllore_contratto` FOREIGN KEY (`controllore`) REFERENCES `controllori` (`codice_fiscale`) ON DELETE NO ACTION ON UPDATE NO ACTION  
)
```

```
CREATE TABLE `autisti` (  
  `codice_fiscale` char(16) NOT NULL,  
  `nome` varchar(255) DEFAULT NULL,  
  `cognome` varchar(255) DEFAULT NULL,  
  `data_di_nascita` date DEFAULT NULL,  
  `patente` date DEFAULT NULL,  
  PRIMARY KEY (`codice_fiscale`)  
)
```

```
CREATE TABLE `controllori` (  
  `codice_fiscale` char(16) NOT NULL,  
  `nome` varchar(255) DEFAULT NULL,  
  `cognome` varchar(255) DEFAULT NULL,  
  `data_di_nascita` date DEFAULT NULL,  
  `categoria` int(11) DEFAULT NULL,  
  PRIMARY KEY (`codice_fiscale`)  
)
```

```
CREATE TABLE `corse` (  
  `codice` int(11) NOT NULL,  
  `autista` char(16) NOT NULL,  
  `controllore` char(16) NOT NULL,  
  `partenza` timestamp NULL DEFAULT current_timestamp(),  
  `arrivo` timestamp NULL DEFAULT current_timestamp(),  
  `linea` varchar(3) NOT NULL,  
  PRIMARY KEY (`codice`),  
  KEY `fk_autisti` (`autista`),  
  KEY `fk_controllori` (`controllore`),  
  KEY `fk_linee` (`linea`),  
  CONSTRAINT `fk_autisti` FOREIGN KEY (`autista`) REFERENCES `autisti` (`codice_fiscale`) ON DELETE NO ACTION ON  
UPDATE  
  NO ACTION,  
  CONSTRAINT `fk_controllori` FOREIGN KEY (`controllore`) REFERENCES `controllori` (`codice_fiscale`) ON DELETE NO ACTION ON  
UPDATE  
  NO ACTION,  
  CONSTRAINT `fk_linee` FOREIGN KEY (`linea`) REFERENCES `linee` (`nome`) ON DELETE NO ACTION ON  
UPDATE  
  NO ACTION  
)
```

```
CREATE TABLE `linee` (  
  `nome` varchar(3) NOT NULL,  
  `partenza` varchar(255) DEFAULT NULL,  
  `arrivo` varchar(255) DEFAULT NULL,  
  `frequenza` varchar(255) DEFAULT NULL,  
  `tipologia` varchar(255) NOT NULL,  
  `deposito` varchar(50) DEFAULT NULL,  
  PRIMARY KEY (`nome`),  
  KEY `fk_depositi` (`deposito`),  
  CONSTRAINT `fk_depositi` FOREIGN KEY (`deposito`) REFERENCES `depositi` (`nome`) ON DELETE NO ACTION ON  
UPDATE  
  NO ACTION  
)
```

```
CREATE TABLE `depositi` (  
  `nome` varchar(50) NOT NULL,  
  `affitto_mensile` float DEFAULT NULL,  
  PRIMARY KEY (`nome`)  
)
```

6.2 Operazioni di insert/delete/update/select

```
INSERT INTO `depositi` (`nome`, `affitto_mensile`) VALUES ('Casalnuovo', '1300');
```

```
INSERT INTO `depositi` (`nome`, `affitto_mensile`) VALUES ('Arzano',1600);
```

```
INSERT INTO `depositi` (`nome`, `affitto_mensile`) VALUES ('Brin',1400);
```

```
INSERT INTO `depositi` (`nome`, `affitto_mensile`) VALUES ('Casalnuovo',1600);
```

```
INSERT INTO `depositi` (`nome`, `affitto_mensile`) VALUES ('Cavalleggeri',1300);
```

```
INSERT INTO `depositi` (`nome`, `affitto_mensile`) VALUES ('Posillipo',1500);
```

```
INSERT INTO `depositi` (`nome`, `affitto_mensile`) VALUES ('Pozzuoli',1600);
```

```
INSERT INTO `depositi` (`nome`, `affitto_mensile`) VALUES ('Puglie',1800);
```

```
UPDATE `depositi` SET `nome`='Vomero'WHERE nome = 'Vomerp'
```

```
DELETE FROM `linee` WHERE `nome` = 152
```

Il risultato delle operazioni:

nome	varchar	affitto_mensile	float
Arzano		1600	
Brin		1400	
Casalnuovo		1300	
Cavalleggeri		1200	
Posillipo		1500	
Pozzuoli		1600	
Puglie		1800	

Alcune interrogazioni, con i relativi risultati:

```
/* PER OGNI LINEA, IL DEPOSITO A CUI E' ASSEGNATO */
```

```
select linee.nome as linea, depositi.nome as deposito  
from linee LEFT JOIN depositi on linee.deposito=depositi.nome
```

linea	deposito
152	Brin
R2	Brin
R5	Brin
165	Cavalleggeri
612	Cavalleggeri
618	Cavalleggeri
633	Cavalleggeri
R6	Cavalleggeri
P8	Pozzuoli
201	Puglie

autista	numero_corse_effettuate
1234567890ASDFGH	0
MNNFNC00T02F839C	1
MNNFNC00T02F839E	0
MNNFNC00T02F839M	1
MNNFNC00T02F839O	0
MNNFNC00T02F839P	0
MNNFNC00T02F839Q	3
MNNFNC00T02F839R	0
MNNFNC00T02F839T	2
MNNFNC00T02F839W	3
MNNFNC00T02F839Y	0

```
/* NUMERO DI TURNI EFFETTUATI DA OGNI AUTISTA*/
```

```
SELECT autisti.codice_fiscale as autista, count(corse.codice) as numero_corse_effettuate
FROM autisti left join corse on autisti.codice_fiscale=corse.autista
GROUP BY autisti.codice_fiscale
```

```
/* AUTISTI CHE HANNO EFFETTUATO ALMENO UNA CORSA */
```

```
select autisti.nome, autisti.cognome
from autisti
where autisti.codice_fiscale in (select corse.autista
                                from corse
                                group by corse.autista
                                having count(*)>0 )
```

nome	cognome
Luigi	Scognamiglio
Pasquale	Pasqua
Francesco Pio	Manna
Nicola	Palmieri
Kiril	Litvinovic

```
/* AUTISTA CHE HA EFFETTUATO IL MAGGIOR NUMERO DI TURNI */
```

```
create or replace view turni_totali_autisti(autista, numero_turni) as
select autisti.codice_fiscale, count(*) as turni_totali
from autisti left join corse on autisti.codice_fiscale = corse.autista
group by autisti.codice_fiscale

select autisti.nome, autisti.cognome
from autisti
where autisti.codice_fiscale = (select turni_totali_autisti.autista
                                from turni_totali_autisti
                                where turni_totali_autisti.numero_turni = (select max(turni_totali_autisti.numero_turni)
                                                                                from turni_totali_autisti ))
```

nome	cognome
Francesco Pio	Manna

```

/* NOME DI TUTTE LE LINEE CHE PARTONO DA PIANURA O POZZUOLI */

select linee.nome
from linee
where linee.partenza = "Pianura" OR linee.partenza = "Pozzuoli"

/* OPPURE IN ALTERNATIVA */

select linee.nome
from linee
where linee.partenza = "Pianura"
UNION
select linee.nome
from linee
where linee.partenza = "Pozzuoli"

```

nome ▲
612
633
R6
P8

```

/* STIPENDIO DELL'AUTISTA FRANCESCO PIO MANNA */

select contratti_autisti.autista, sum(contratti_autisti.paga_per_corsa) * 4 as stipendio
from contratti_autisti
where contratti_autisti.autista = (select autisti.codice_fiscale
                                   from autisti
                                   where autisti.nome = "Francesco Pio" AND autisti.cognome = "Manna" )

```

autista ▲	stipendio ▲
MNNFNC00T02F839Q	60

/* dopo l'aggiornamento del codice fiscale dell'autista nella tabella AUTISTI, modifica il relativo campo AUTISTA in contratti_autisti */

```

create trigger aggiornamento
after update on autisti
for each row
begin
    update contratti_autisti set contratti_autisti.autista =: new.codice_fiscale
    where contratti_autisti.autista =: old.codice_fiscale
end

```

/* dopo la cancellazione di un record della tabella CONTROLLORI, viene settato al valore "0000000000000000" il campo controllore in contratti_controllori */

```

create trigger eliminazione
after delete on controllori
for each row
begin
    update contratti_controllori set contratti_controllori.controllore = "0000000000000000"
    where contratti_controllori.controllore = old.codice_fiscale
end

```