

UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA

Corso di Laurea in Ingegneria dell'Informazione per Videogame e Realtà Virtuale

PROGETTAZIONE E IMPLEMENTAZIONE DI UN SISTEMA INFORMATIVO PER IL POLIMARCHE RACING TEAM: RACCOLTA E ANALISI DI INFORMAZIONI SULLE SESSIONI DI PROVA IN PISTA

DESIGN AND IMPLEMENTATION OF AN INFORMATION SYSTEM FOR THE POLIMARCHE RACING TEAM: STORAGE AND ANALYSIS OF INFORMATION ABOUT TRACK TEST SESSIONS

Relatore:
Prof.ssa Claudia Diamantini

Tesi di Laurea di:
Tommaso Galante

A.A. 2025/2026

Indice

0 Introduzione 1

1 Analisi dei requisiti 3

1.1 Raccolta documentazione (modulistica) 3

1.2 Requisiti espressi in linguaggio naturale 6

1.3 Strutturazione dei requisiti 8

1.3.1 Scopo del sistema 8

1.3.2 Frasi relative ai componenti 8

1.3.3 Frasi relative alle sessioni 8

1.3.4 Frasi relative alle misurazioni 9

1.4 Specifica delle operazioni 9

2 Progettazione concettuale 11

2.1 Spiegazione della modalità con cui si intende procedere 11

2.2 Identificazione delle entità e relazioni fondamentali 11

2.3 Scheletro dello schema 12

2.4 Sviluppo delle componenti dello scheletro 13

2.5 Verbalizzazione del modello 16

2.6 Breve analisi di qualità dello schema 16

2.7 Dizionario dei dati 17

2.7.1 Tabella delle entità 17

2.7.2 Tabella delle relazioni 19

3 Progettazione logica 21

3.1 Tavola dei volumi 21

3.2 Tavola delle operazioni 22

3.3 Ristrutturazione dello schema concettuale 24

3.4 Partizionamento/accorpamento di concetti 27

3.4.1 Partizionamento o accorpamento di concetti 27

3.4.2 Eliminazione degli attributi multivalore 27

3.5 Elenco degli identificatori principali 28

3.6 Rappresentazione grafica del modello concettuale ristrutturato 30

3.7 Traduzione verso il modello relazionale 30

3.8 Normalizzazione 34

4 Codifica SQL e Testing 35

4.1 Definizione dello schema e immagini successive all'inserimento dei dati 35

4.2 Codifica delle operazioni e codice che ne verificano l'esecuzione 40
5 Conclusioni e sviluppi futuri 51
6 Bibliografia e sitografia 52

Indice delle figure

Figura 1 - Scheda tecnica relativa al sensore IMU 4
Figura 2 - Scheda tecnica relativa al sensore rotativo senza contatto 4
Figura 3 - Scheda tecnica relativa al trasduttore piezoresistente di pressione 5
Figura 4 - Scheda tecnica della struttura del messaggio CAN relativo al sensore IMU Aviorace 5
Figura 5 - Struttura della tabella Excel dei segnali generici inviati tramite protocollo CAN 6
Figura 6 - Struttura della tabella Excel dei segnali specifici dei dispositivi inverter inviati tramite protocollo CAN 6
Figura 7 - Rappresentazione tramite modello E-R delle entità principali del sistema 12
Figura 8 - Rappresentazione tramite modello E-R delle entità e relazioni fondamentali 13
Figura 9 - Rappresentazione tramite modello E-R delle entità e relazioni associate all'entità "Componente" 14
Figura 10 - Rappresentazione tramite modello E-R delle entità e relazioni associate all'entità "Messaggio CAN" 14
Figura 11 - Rappresentazione tramite modello E-R delle entità e relazioni associate all'entità "Sessione" 15
Figura 12 - Rappresentazione tramite modello E-R dello schema concettuale completo 15
Figura 13 - Rappresentazione tramite modello E-R delle entità e relazioni appartenenti alla gerarchia "Componente" 24
Figura 14 - Rappresentazione tramite modello E-R ristrutturata delle entità e relazioni appartenenti alla gerarchia "Componente" 25
Figura 15 - Rappresentazione tramite modello E-R delle entità e relazioni appartenenti alla gerarchia "Componente Elettronico" 26
Figura 16 - Rappresentazione ristrutturata delle entità e relazioni appartenenti alla gerarchia "Componente Elettronico" nel modello E-R 26
Figura 17 - Rappresentazione dell'attributo multivalore "Telefono" appartenente all'entità "Pilota" 27
Figura 18 - Rappresentazione dell'eliminazione dell'attributo e della trasformazione in entità "Telefono" 28
Figura 19 - Rappresentazione tramite modello E-R dello schema concettuale ristrutturato 30
Figura 20 - Schema di definizione delle tabelle rappresentanti la base di dati 35
Figura 21 - Rappresentazione grafica dell'analisi di velocità di un pilota in diverse sessioni in un circuito 46
Figura 22 - Rappresentazione grafica della velocità media dei piloti in un circuito specifico 47
Figura 23 - Rappresentazione grafica della temperatura delle ruote in una sessione di corsa 48

0 Introduzione

Attraverso questa tesi progettuale si intende illustrare per gradi la progettazione e lo sviluppo di una base di dati relazionale a supporto della gestione e dell'analisi dei dati relativi al lavoro del gruppo universitario Polimarche Racing Team. L'idea di progetto nasce dalla necessità di organizzare in modo strutturato e coerente le numerose informazioni generate durante l'attività della squadra, tra cui i dati relativi ai kart, ai piloti, ai componenti installati, alle comunicazioni dei sensori e alle misurazioni effettuate nelle sessioni di prova in pista.

Sono quindi emerse delle questioni sul come realizzare un sistema informativo che permettesse non solo la corretta organizzazione di archiviazione di queste informazioni, ma anche che sia di supporto all'estrazione dei dati e all'analisi utile alla comprensione delle prestazioni dei kart e dei loro piloti. L'obiettivo delle tesi consiste quindi nella progettazione di una base di dati relazionale completa che, partendo dall'analisi dei requisiti, percorre diverse fasi e termina con la sua implementazione tramite codice e il successivo testing del prodotto finale.

L'obiettivo è stato così scomposto in:

- Individuare e organizzare le principali informazioni necessarie alla gestione dei componenti, delle misurazioni e delle sessioni in pista;
- Definire le entità, gli attributi e le relazioni fondamentali nella fase di progettazione concettuale;
- Trasformare il modello concettuale in uno schema logico, verificandone la qualità e l'efficienza tramite normalizzazione;
- Implementare lo schema della base di dati attraverso codice SQL;
- Inserire le istanze e accertarsi del corretto funzionamento delle relazioni e dei vincoli definiti;
- Realizzare interrogazioni SQL in grado di estrarre informazioni significative all'analisi delle sessioni e delle prestazioni.

Nella fase iniziale si sono analizzati i requisiti del sistema e la documentazione fornita relativa alle informazioni da gestire. Successivamente ci si è dedicati all'individuazione delle entità e delle relazioni e alla costruzione di un dizionario di dati in modo da costruire uno schema concettuale riassuntivo, tramite la tecnica del modello E-R. In seguito, nella fase di progettazione logica, il modello concettuale ha subito una importante trasformazione, durante

la quale lo schema è stato ristrutturato e tradotto nel modello relazionale, a struttura tabellare, e poi verificata la sua correttezza mediante la normalizzazione.

Infine, il sistema ottenuto è stato implementato in SQL e sottoposto a una fase di testing attraverso l'inserimento dei dati e l'esecuzione di interrogazioni rappresentative delle operazioni più frequenti.

Si è dedicata particolare attenzione alla memorizzazione delle informazioni sui componenti del kart e alla gestione delle sessioni in pista e delle misurazioni provenienti dai sensori. Tramite l'utilizzo delle interrogazioni SQL è infatti possibile mettere in relazione piloti, kart e sessioni, o, in generale, i dati acquisiti, consentendo di effettuare confronti sulle prestazioni. I risultati di alcune interrogazioni sono rappresentate graficamente, permettendo una visualizzazione più immediata dei parametri, come la velocità media e la durata di una sessione di corsa.

La tesi è strutturata in cinque capitoli principali:

- Nel Capitolo 1 viene introdotta l'attività di analisi dei requisiti, attraverso la raccolta della documentazione necessaria allo sviluppo, definendo inoltre lo scopo del sistema informativo e le principali operazioni supportate.
- Nel Capitolo 2 ci si è concentrati sulla progettazione concettuale, quindi lo sviluppo e la verbalizzazione dello schema concettuale in modello E-R delle entità e relazioni identificate.
- Nel Capitolo 3 viene descritto il processo di progettazione logica, che comprende l'analisi dei volumi e delle operazioni, la ristrutturazione dello schema concettuale, la determinazione degli identificatori primari, la traduzione nel modello relazionale e la normalizzazione.
- Nel Capitolo 4 è illustrata la fase di codifica e testing. Viene, infatti, presentato il codice utilizzato per la creazione dello schema e per l'inserimento dei dati. Inoltre, sono descritte le principali operazioni e i risultati ottenuti dalla loro esecuzione.
- Nel Capitolo 5 vengono riportate le conclusioni del progetto, evidenziando i risultati raggiunti e alcuni possibili sviluppi futuri.

Nel complesso, l'elaborato propone la realizzazione di una raccolta strutturata di dati operativi pensata per organizzare e valorizzare le informazioni prodotte nell'ambito del kart racing, ed è progettata in modo tale da poter offrire la possibilità di essere ampliata nel tempo, per aggiungere ulteriori campi di procedure attinenti.

1 Analisi dei requisiti

1.1 Raccolta documentazione (modulistica)

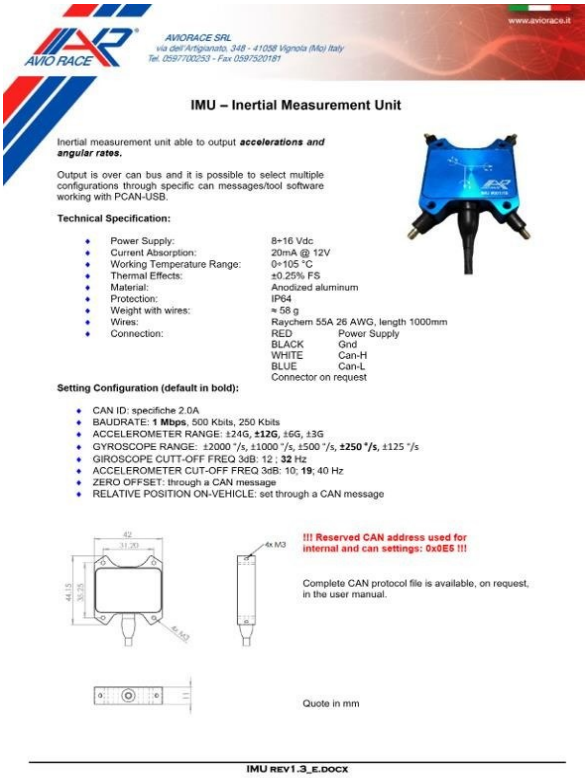
Durante la fase preliminare del progetto, il maggior carico di lavoro è stato dedicato all'acquisizione e all'analisi dei requisiti necessari alla realizzazione della base di dati. Pertanto, si sono approfondite, insieme al team studentesco, le esigenze primarie sulla gestione, cercando di comprendere che tipo di dati venissero acquisiti, come fossero attualmente organizzati e quali fossero le principali necessità di consultazione delle informazioni.

A seguito del confronto, il gruppo ha fornito, tramite e-mail, molteplici documenti interni contenenti dettagli tecnici e descrittivi relativi agli oggetti presi in analisi. La documentazione ricevuta rappresenta la modulistica effettivamente utilizzata per la gestione quotidiana dei dati e costituisce quindi il punto di partenza per l'analisi dei requisiti.

I documenti risultavano piuttosto eterogenei e non organizzati all'interno di un'unica infrastruttura dedicata. Prima della realizzazione del sistema informativo, infatti, i dati venivano consultati principalmente attraverso le singole schede tecniche dei componenti, dette anche *datasheet*, e mediante una tabella Excel in cui erano presenti le informazioni relative ai segnali acquisibili.

L'analisi di questa documentazione ha permesso di individuare i concetti principali da cui estrarre le informazioni di gestione e le relazioni esistenti tra di esse, ponendo le fondamenta per la successiva progettazione e strutturazione della base di dati.

Di seguito vengono riportati i documenti di partenza utilizzati per lo sviluppo del sistema, con l'obiettivo di trasformare le informazioni raccolte in una struttura organizzata, coerente e facilmente consultabile dai membri del team.



AVIORACE SRL
via dell'Artigianato, 348 - 41058 Vignola (Mo) Italy
Tel. 0597700253 - Fax 0597520181
www.aviorace.it

IMU – Inertial Measurement Unit

Inertial measurement unit able to output ***accelerations and angular rates.***

Output is over can bus and it is possible to select multiple configurations through specific can messages/tool software working with PCAN-USB.

Technical Specification:

Power Supply:	8÷16 Vdc
Current Absorption:	20mA @ 12V
Working Temperature Range:	0÷105 °C
Thermal Effects:	±0.25% FS
Material:	Anodized aluminum
Protection:	IP64
Weight with wires:	≈ 58 g
Wires:	Raychem 55A 26 AWG, length 1000mm
Connection:	RED Power Supply BLACK Gnd WHITE Can-H BLUE Can-L Connector on request

Setting Configuration (default in bold):

- CAN ID: specifiche 2.0A
- BAUDRATE: **1 Mbps**, 500 Kbits, 250 Kbits
- ACCELEROMETER RANGE: ±24G, **±12G**, ±6G, ±3G
- GYROSCOPE RANGE: ±2000 °/s, ±1000 °/s, ±500 °/s, **±250 °/s**, ±125 °/s
- GIROSCOPE CUTT-OFF FREQ 3dB: 12 ; **32** Hz
- ACCELEROMETER CUT-OFF FREQ 3dB: 10; **19**; 40 Hz
- ZERO OFFSET: through a CAN message
- RELATIVE POSITION ON-VEHICLE: set through a CAN message

!!! Reserved CAN address used for internal and can settings: 0x0E5 !!!

Complete CAN protocol file is available, on request, in the user manual.

Quote in mm

Figura 1 - Scheda tecnica relativa al sensore IMU



AVIORACE SRL
via dell'Artigianato, 348 - 41058 Vignola (Mo) Italy
Tel. 0597700253 - Fax 0597520181
www.aviorace.it

NON-CONTACT ROTARY SENSORS *SRH SERIES*

Series of "non-contact" rotary potentiometers designed to withstand the harsh environmental conditions of motorsports. Using proven "Hall Effect" technology, the SRH series sensors offer high performance and reliability at operational temperatures up to +150°C. These small and lightweight sensors are available with a large range of options. Used for detection of angular positions of mechanical organs such as throttle, pedals, etc.

SPECIFICATIONS:

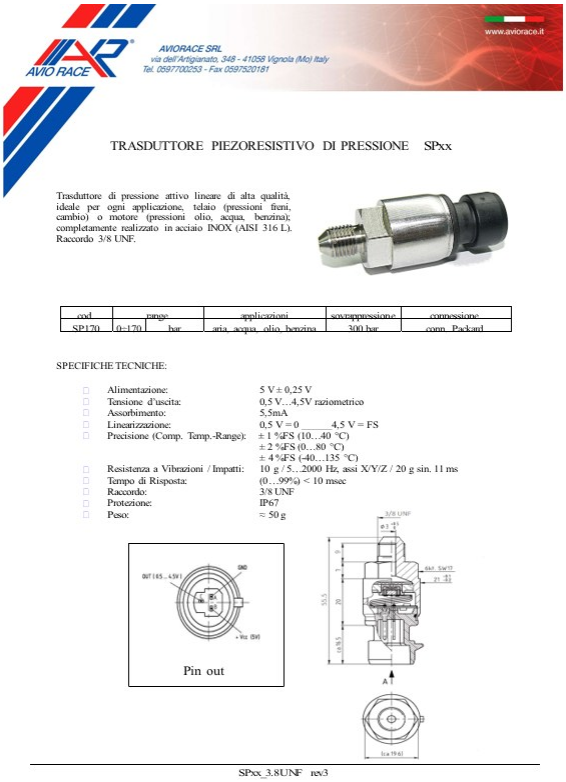
Electrical

Technology:	Hall Effect
Electrical angle:	30° - 100° - 360° STD
Channel Option:	Single & Dual channel (redundant)
Supply Voltage:	5V DC (ratiometric output (10/90% Vin)) 6-30V DC (absolute output 0.5-4.5V)
Output signal:	Analogue voltage
Resolution:	12bit@5KHz
Independent Linearity:	≤0.3% of measurement range
Temperature Coefficient:	50ppm/K° of output signal
Reverse Polarity Protection:	Yes
Over Voltage Protection:	40V DC
MAX Output Current:	8mA (per channel)

Mechanical

Mechanical Angle:	360° continuous rotation
MAX Shaft Velocity:	600 RPM
Torque:	<3Ncm
Housing material:	Aluminum
Shaft Material:	Stainless Steel
Weight:	15g (without cable)
Sealing:	IP67
Operating Temperature:	-40°C ÷ +150°C
Vibrations:	5-2000Hz random
Cable type and length:	~1000mm AWG26 Spec 55A
Wiring:	**Vcc** **Gnd** **Segnale**
Single output (3 wires):	RED BLACK YELLOW
Dual output (6 wires):	BROWN BLUE WHITE
Allignment:	Sensor output is at electrical center, when the shaft flat (D) or shaft ident " (F,B) is aligned with the cable exit.

Figura 2 - Scheda tecnica relativa al sensore rotativo senza contatto



AVIORACE SRL
via dell'Artigianato, 348 - 41058 Vignola (Mo) Italy
Tel. 0597700253 - Fax 0597520181
www.aviorace.it

TRASDUTTORE PIEZORESISTIVO DI PRESSIONE SPxx

Trasduttore di pressione attivo lineare di alta qualità, ideale per ogni applicazione, telaio (pressioni freni, cambio) o motore (pressioni olio, acqua, benzina); completamente realizzato in acciaio INOX (AISI 316 L). Raccordo 3/8 UNF.

cod.	range	applicazioni	sovrappressione	connessione
SP170	0÷170 bar	aria, acqua, olio, benzina	300 bar	conn. Packard

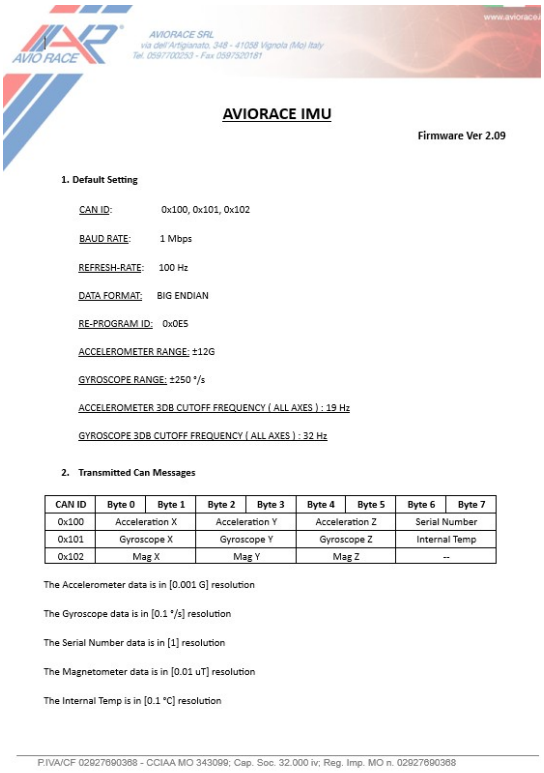
SPECIFICHE TECNICHE:

Alimentazione:	5 V ± 0,25 V
Tensione d'uscita:	0,5 V...4,5V raziometrico
Assorbimento:	5,5mA
Linearizzazione:	0,5 V = 0 _____ 4,5 V = FS
Precisione (Comp. Temp.-Range):	± 1 %FS (10…40 °C) ± 2 %FS (0…80 °C) ± 4 %FS (-40…135 °C)
Resistenza a Vibrazioni / Impatti:	10 g / 5...2000 Hz, assi X/Y/Z / 20 g sin. 11 ms
Tempo di Risposta:	(0…99%) < 10 msec
Raccordo:	3/8 UNF
Protezione:	IP67
Peso:	≈ 50 g

Pin out

SPxx_3.8UNF rev3

Figura 3 - Scheda tecnica relativa al trasduttore piezoresistente di pressione



AVIORACE SRL
via dell'Artigianato, 348 - 41058 Vignola (Mo) Italy
Tel. 059/700253 - Fax 059/520181

AVIORACE IMU

Firmware Ver 2.09

1. Default Setting

CAN ID: 0x100, 0x101, 0x102

BAUD RATE: 1 Mbps

REFRESH-RATE: 100 Hz

DATA FORMAT: BIG ENDIAN

RE-PROGRAM ID: 0x0E5

ACCELEROMETER RANGE: ±12G

GYROSCOPE RANGE: ±250 °/s

ACCELEROMETER 3DB CUTOFF FREQUENCY (ALL AXES) : 19 Hz

GYROSCOPE 3DB CUTOFF FREQUENCY (ALL AXES) : 32 Hz

2. Transmitted Can Messages

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x100	Acceleration X		Acceleration Y		Acceleration Z		Serial Number	
0x101	Gyroscope X		Gyroscope Y		Gyroscope Z		Internal Temp	
0x102	Mag X		Mag Y		Mag Z		--	

The Accelerometer data is in [0.001 G] resolution

The Gyroscope data is in [0.1 °/s] resolution

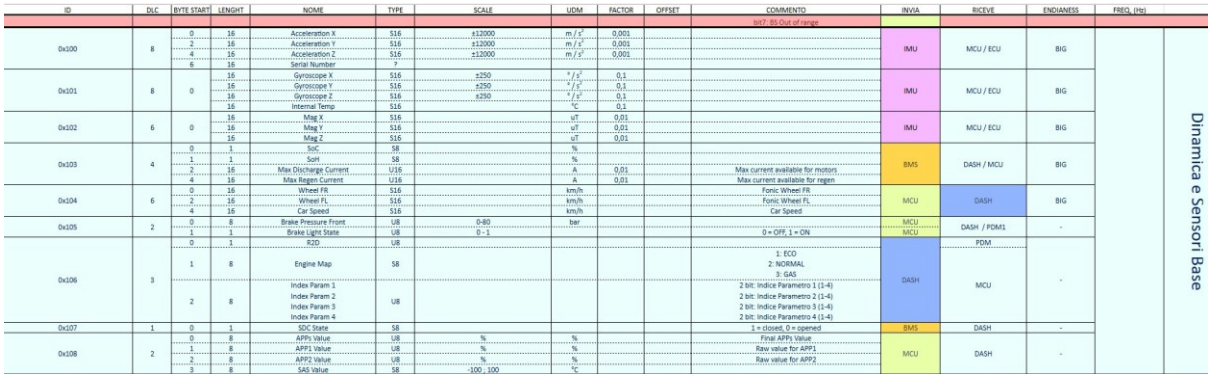
The Serial Number data is in [1] resolution

The Magnetometer data is in [0.01 uT] resolution

The Internal Temp is in [0.1 °C] resolution

P.IVA/CF 02927690368 - CCIAA MO 343099; Cap. Soc. 32.000 iv; Reg. Imp. MO n. 02927690368

Figura 4 - Scheda tecnica della struttura del messaggio CAN relativo al sensore IMU Aviorace



ID	DLC	BYTE START	LENGHT	NOME	TYPE	SCALE	UDM	FACTOR	OFFSET	COMMENTO	INVIA	RICEVE	ENDIANESS	FREQ. (Hz)	
										bit7: BS Out of range					
0x100	8	0	16	Acceleration X	S16	±12000	m / s²	0,001			IMU	MCU / ECU	BIG		Dinamica e Sensori Base
		2	16	Acceleration Y	S16	±12000	m / s²	0,001							
		4	16	Acceleration Z	S16	±12000	m / s²	0,001							
		6	16	Serial Number	?										
0x101	8	0	16	Gyroscope X	S16	±250	°/s²	0,1			IMU	MCU / ECU	BIG		
			16	Gyroscope Y	S16	±250	°/s²	0,1							
			16	Gyroscope Z	S16	±250	°/s²	0,1							
			16	Internal Temp	S16		°C	0,1							
0x102	6	0	16	Mag X	S16		uT	0,01			IMU	MCU / ECU	BIG		
			16	Mag Y	S16		uT	0,01							
			16	Mag Z	S16		uT	0,01							
0x103	4	0	1	Soc	S8		%				BMS	DASH / MCU	BIG		
		1	1	Soh	S8		%								
		2	16	Max Discharge Current	U16		A	0,01		Max current available for motors					
		4	16	Max Regen Current	U16		A	0,01		Max current available for regen					
0x104	6	0	16	Wheel FR	S16		km/h			Fonic Wheel FR	MCU	DASH	BIG		
		2	16	Wheel FL	S16		km/h			Fonic Wheel FL					
		4	16	Car Speed	S16		km/h			Car Speed					
0x105	2	0	8	Brake Pressure Front	U8	0-80	bar				MCU	DASH / PDM1	-		
		1	1	Brake Light State	U8	0 - 1				0 = OFF, 1 = ON	MCU	DASH	-		
0x106	3	0	1	R2D	U8						DASH	PDM	-		
		1	8	Engine Map	S8					1: ECO 2: NORMAL 3: GAS		MCU			
		2	8	Index Param 1 Index Param 2 Index Param 3 Index Param 4	U8					2 bit: Indice Parametro 1 (1-4) 2 bit: Indice Parametro 2 (1-4) 2 bit: Indice Parametro 3 (1-4) 2 bit: Indice Parametro 4 (1-4)					
0x107	1	0	1	SDC State	S8					1 = closed, 0 = opened	BMS	DASH	-		
0x108	2	0	8	APPs Value	U8	%	%			Final APPs Value	MCU	DASH	-		
		1	8	APP1 Value	U8	%	%			Raw value for APP1					
		2	8	APP2 Value	U8	%	%			Raw value for APP2					
		3	8	SAS Value	S8	-100 ; 100	°C								

Figura 5 - Struttura della tabella Excel dei segnali generici inviati tramite protocollo CAN

ID	DLC	BYTE START	LENGHT	NOME	TIPO	SCALA	UDM	FACTOR	OFFSET	COMMENTO	INVIA	RICEVE
0x184	8	0	16	AMK_CTR_WORD	U16					AMK_Control_Word	MCU	INV1
		2	16	INV1_TORQUE_VAL	S16	-210 ; 210	%Nm	0,1		INV1 Setpoint % Coppia Nominale		
		4	16	INV1_TORQUE_POS_LIM	S16	0 ; 210	%Nm	0,1		INV1 Limite Coppia Positiva		
		6	16	INV1_TORQUE_NEG_LIM	S16	-210 ; 0	%Nm	0,1		INV1 Limite Coppia Negativa		
0x185	8	0	16	AMK_CTR_WORD	U16					AMK_Control_Word	MCU	INV2
		2	16	INV2_TORQUE_VAL	S16	-210 ; 210	%Nm	1		INV2 Setpoint % Coppia Nominale		
		4	16	INV2_TORQUE_POS_LIM	S16	0 ; 210	%Nm	1		INV2 Limite Coppia Positiva		
		6	16	INV2_TORQUE_NEG_LIM	S16	-210 ; 0	%Nm	1		INV2 Limite Coppia Negativa		
0x283	8	0	16	INV1_STAT_WORD INV1_ACT_SPEED	U16					INV1 AMK_Status_Word INV1 Velocità attuale	INV1	MCU
		2	16	INV1_ACT_TORQUE	S16	-22000 ; 22000	rpm	1				
		4	16	INV1_ACT_MAGN_CURR	S16		Nm	1		INV1 Coppia erogata (feedback) INV1		
		6	16		S16	0 ; 35	A	1		Corrente negativa x Defl.		
0x284	8	0	16	INV2_STAT_WORD INV2_ACT_SPEED	U16					INV2 AMK_Status_Word INV2 Velocità attuale	INV2	MCU
		2	16	INV2_ACT_TORQUE	S16	-22000 ; 22000	rpm	1				
		4	16	INV2_ACT_MAGN_CURR	S16		Nm	1		INV2 Coppia erogata (feedback) INV2		
		6	16		S16	0 ; 35	A	1		Corrente negativa x Defl.		
0x285	8	0	16	INV1_MOTOR_TEMP	S16		°C	0,1		INV1 Motor Temp	INV1	MCU
		2	16	INV1_TEMP	S16		°C	0,1		INV1 Temp		
		4	16	INV1_ERROR_NUM	U16					INV1 Numero Errore		
		6	16	INV1_IGBT_TEMP	S16		°C	0,1		INV1 IGBT Temp		
0x286	8	0	16	INV2_MOTOR_TEMP	S16		°C	0,1		INV2 Motor Temp	INV2	MCU
		2	16	INV2_TEMP	S16		°C	0,1		INV2 Temp		
		4	16	INV2_ERROR_NUM	U16					INV2 Numero Errore		
		6	16	INV2_IGBT_TEMP	S16		°C	0,1		INV2 IGBT Temp		
0x287	8	0	16	INV1_DC_BUS	U16	0 ; 540	V	1		INV1 Tensione DC BUS	INV1	MCU
		2	16	INV1_OVERLOAD_MOTOR	U16					INV1 Warning Motor		
		4	32	INV1_CURR_PHASE_U	S32		A	1		INV1 Corrente Fase U		
0x288	8	0	16	INV2_DC_BUS	U16	0 ; 540	V	1		INV2 Tensione DC BUS	INV2	MCU
		2	16	INV2_OVERLOAD_MOTOR	U16					INV2 Warning Motor		
		4	32	INV2_CURR_PHASE_U	S32		A	1		INV2 Corrente Fase U		
0x289	8	0	32	INV1_CURR_PHASE_V	S32		A	1		INV1 Corrente Fase V INV1	INV1	MCU
		4	32	INV1_CURR_PHASE_W	S32		A	1		Corrente Fase W		
0x290	8	0	32	INV2_CURR_PHASE_V	S32		A	1		INV2 Corrente Fase V INV2	INV2	MCU
		4	32	INV2_CURR_PHASE_W	S32		A	1		Corrente Fase W		
0x291	8	0	32	INV1_ERROR_INFO1 INV1_ACT_POWER	S32					INV1 Codice ERRORE Info1 INV1	INV1	MCU
		4	32		S32		W	1		Potenza attuale		
0x292	8	0	32	INV2_ERROR_INFO1 INV2_ACT_POWER	S32					INV2 Codice ERRORE Info1 INV2	INV2	MCU
		4	32		S32		W	1		Potenza attuale		
					U16							
					S16							
					S16							
				INV1_STAT_WORD	S16							
				INV1_ACT_SPEED	S16							
				INV1_MOTOR_TEMP	U16							

Figura 6 - Struttura della tabella Excel dei segnali specifici dei dispositivi inverter inviati tramite protocollo CAN

1.2 Requisiti espressi in linguaggio naturale

L'obiettivo del progetto mira alla creazione di un sistema informativo in grado di soddisfare le richieste del gruppo studentesco Polimarche Racing Team, permettendo di memorizzare e gestire l'elevata quantità di informazioni acquisite dall'attività dei piloti e kart di corsa durante le sessioni di prova in pista. Esaminando le necessità espresse dal team e la documentazione descritta nella fase di analisi dei requisiti, sono stati individuati i presupposti chiave su cui basare la progettazione della base di dati.

Per quanto riguarda i kart, il sistema deve essere in grado di immagazzinare le informazioni tecniche e descrittive di ciascun mezzo, come il modello, la categoria di gara, lo stato di attivazione e l'anno di omologazione. Inoltre, deve essere possibile associare ai kart, i componenti installati su di essi, con attributi di identificazione, il produttore, il modello, il peso e la data di installazione.

Per la gestione dei componenti, fondamentale è la suddivisione in dispositivi elettronici e non, riportando concetti quali il tipo di componente, le grandezze che sono in grado di misurare e le specifiche sull'alimentazione, oltre alle caratteristiche sui componenti sopra citate. I sensori hanno un'importanza maggiore, poiché sono i dispositivi che acquisiscono e trasmettono i dati in tempo reale, per questo è indispensabile assegnare loro proprietà aggiuntive, come la posizione di montaggio e il tipo di grandezza registrabile.

Tra le necessità del team è stata dichiarata quella sul trattamento delle informazioni relative alle comunicazioni e ai segnali acquisiti dai componenti elettronici, in particolare i sensori. Pertanto devono essere inseriti i messaggi di comunicazione dello standard CAN utilizzati, i segnali associati e i propri attributi di costruzione, come l'unità di misura, i valori range, massimo e minimo e il fattore di conversione. Questa organizzazione permette così una corrispondenza tra i componenti montati sui kart e i dati che possono essere recepiti dai ricevitori.

Nel progetto vengono inseriti anche i dati anagrafici dei piloti, permettendo di associare i kart e le sessioni di corsa al personale di riferimento. È possibile anche consultare più recapiti telefonici e le informazioni necessarie alla loro amministrazione.

Un elemento centrale del sistema è la gestione delle sessioni di corsa. Per ogni sessione risulta segnato il kart utilizzato e il pilota, la data e l'orario di svolgimento, la durata, il circuito ed eventuali note descrittive; in questo modo sarà possibile mantenere uno storico delle attività svolte e mettere in relazione ogni sessione con i dati misurati in ogni secondo.

Per ogni misurazioni, per questo motivo, vengono proposti il segnale di riferimento del protocollo CAN, il valore rilevato e il relativo istante temporale, e, come sopra accennato, la sessione nella quale è stata acquisita. Queste informazioni permettono successive interrogazioni strutturate, in modo da evidenziare importanti analisi di prestazione dei parametri principali rilevati nelle prove, come la velocità e l'accelerazione. Ulteriori estrazioni riguardano i confronti tra i tempi di esecuzione di piloti differenti o il paragone di sessioni di corsa di uno stesso pilota in un determinato circuito.

In sintesi, la base di dati deve fornire uno strumento ordinato di supporto alla gestione delle attività del team e all'analisi dei dati raccolti, e luogo in cui i dati vengono organizzati in modo

coerente e facilmente consultabile, garantendo l'integrità delle informazioni e riducendo la possibilità di duplicazioni o errori. Per questo, i risultati delle interrogazioni di analisi vengono rappresentati anche graficamente, rendendo la loro lettura più immediata.

1.3 Strutturazione dei requisiti

1.3.1 Scopo del sistema

Lo scopo del sistema riguarda la progettazione e l'implementazione di un sistema informativo in grado di gestire in maniera strutturata ed organizzata le informazioni riguardanti i valori e i componenti di un kart da corsa, utile alla storicizzazione e all'analisi avanzata dei dati.

Il sistema deve consentire la registrazione, la consultazione e l'aggiornamento dei dati sui componenti elettronici e meccanici, sessioni di corsa, piloti e le informazioni tecniche dei sensori e componenti elettronici che utilizzano protocollo CAN per la trasmissione.

Si memorizzano molteplici tipologie di dato, al fine di garantire un controllo efficiente e globale delle attività quotidiane, riducendo sprechi e difficoltà organizzative. Il sistema assicura, inoltre, una progettazione tale da poter essere modificato e ampliato nel tempo.

1.3.2 Frasi relative ai componenti

Relativamente ai componenti, sia elettronici che non, il sistema memorizza tutte le informazioni principali e necessarie alla loro gestione: codice seriale, nome, modello, produttore, peso e data di installazione.

Per ciascun componente elettronico sono registrati i gradi di protezione, la corrente assorbita, il tipo di comunicazione, la temperatura operativa e la tensione di alimentazione; in aggiunta, per quanto riguarda i sensori, i componenti elettronici più rilevanti, sono presenti la posizione di montaggio e la grandezza standardizzata di misurazione.

1.3.3 Frasi relative alle sessioni

La sessione di corsa è la categoria che permette più confronti e operazioni di analisi avanzata dei dati. Difatti, tutti i concetti della base di dati sono associati alla sessione, in modo da ridurre la difficoltà di estrazione di informazioni. Il sistema permette di memorizzare la data, l'orario, la durata delle sessioni, il circuito e i dati su kart e pilota. Inoltre, risulta indispensabile associare la sessione alle misurazioni acquisite.

1.3.4 Frasi relative alle misurazioni

Le misurazioni, nella base di dati, rappresentano un concetto a sé stante e sono caratterizzate dal timestamp e i valori collegati al rilevamento dei segnali, che comprendono dati entro un range, byte iniziale, lunghezza del segnale trasmesso, tipo, fattore di misurazione e il nome del segnale associato a una struttura specifica del messaggio CAN inviato, che possiede il data length code, cioè la lunghezza del campo dati di un messaggio, e l'endianess, cioè l'ordine in cui i byte di un valore composto da più byte vengono trasmessi.

1.4 Specifica delle operazioni

In questo stadio di progettazione si riflette anche sulle operazioni e richieste indispensabili che i clienti necessitano di effettuare sul sistema. Queste operazioni vengono esaminate una a una, per poi essere testate tramite programmazione SQL e rese praticabili dagli utenti finali.

Di seguito viene riportato l'elenco definitivo delle operazioni più importanti considerate e, tra parentesi, la presupposta frequenza di esecuzione nella base di dati:

1. Inserimento di una nuova misurazione (in media 1000 volte al mese)
2. Inserimento di una nuova sessione di prova (2 volte al mese)
3. Inserimento di un nuovo pilota (1 volta ogni 2 anni)
4. Inserimento di un nuovo kart (1 volta ogni 3 anni)
5. Inserimento di un nuovo componente elettronico (15 volte all'anno)
6. Inserimento di un nuovo sensore (10 volte all'anno)
7. Inserimento di una nuova comunicazione tra un componente elettronico e un messaggio CAN (15 volte all'anno)
8. Consultazione dei segnali contenuti in un messaggio CAN (10 volte al mese)
9. Consultazione delle misurazioni in una sessione (in media 10 volte al mese)
10. Consultazione delle informazioni di una sessione con relativi pilota e kart (5 volte al mese)
11. Consultazione di una misurazione rispetto ai valori range del segnale (2 volte al mese)
12. Consultazione del numero di sessioni per pilota (1 volta ogni 2 mesi)
13. Consultazione dell'elenco dei sensori e le loro informazioni (1 volta al mese)
14. Consultazione dei componenti non elettronici (10 volte all'anno)
15. Consultazione dei messaggi CAN e i componenti associati (3 volte al mese)
16. Consultazione del valore massimo e medio di una misurazione per ogni sessione

(in media 10 volte al mese)

17. Consultazione del tempo totale di utilizzo di ogni kart (2 volte all'anno)
18. Consultazione dei componenti installati dopo una certa data (10 volte all'anno)
19. Consultazione delle prestazioni di velocità di un pilota in sessioni di prova diverse (2 volte al mese)
20. Consultazione delle analisi di velocità di un pilota in sessioni di prova diverse (2 volte al mese)
21. Consultazione del confronto delle velocità medie tra piloti diversi in un circuito specifico (2 volte al mese)
22. Consultazione della temperatura delle gomme nel tempo in una sessione di corsa (2 volte al mese)
23. Consultazione del confronto della temperatura media, massima e minima di ciascuna ruota in una sessione di corsa (2 volte al mese)
24. Modifica delle informazioni di un componente (1 volta al mese)
25. Modifica delle informazioni di un kart (1 volta ogni 6 mesi)
26. Modifica delle informazioni di una sessione (1 volta al mese)
27. Cancellazione delle misure errate di una sessione (2 volte al mese)
28. Cancellazione di un componente (5 volta all'anno)
29. Cancellazione di un pilota (1 volta ogni 5 anni)

Le operazioni sono riportate suddividendole in gruppi. Quelle che generalmente risultano più frequenti sono le operazioni di consultazione, mentre alcune, ad esempio di inserimento o cancellazione, sono eseguite sporadicamente, per merito della "qualità" della progettazione del sistema e della stabilità evolutiva dell'azienda a cui appartiene il sistema informativo.

Le operazioni 19, 20, 21, 22 e 23 rappresentano una piccola porzione di strumenti di analisi avanzata per comprendere l'andamento di un'impresa, in questo caso, il corretto funzionamento dei kart e la "bravura" dei piloti assunti.

2 Progettazione concettuale

2.1 Spiegazione della modalità con cui si intende procedere

Terminata l'analisi dei requisiti, si procede con la progettazione concettuale, in cui i così detti concetti astratti vengono resi visualizzabili graficamente, tramite l'utilizzo del modello E-R, modello entità-relazione, al fine di ottenere una visione chiara e strutturata delle risorse da trasformare poi in entità o informazioni di gestione, collegate tra loro.

Questa fase consente di individuare le entità fondamentali e le relazione che intercorrono tra di esse, fornendo una rappresentazione grafica e completa del funzionamento del sistema.

L'approccio a questa progettazione è basato su una strategia ibrida/mista. Si è pertanto deciso di procedere nel seguente modo.

Una volta analizzati i requisiti che hanno permesso di raccogliere una grande quantità di informazioni e di ottenere una visione d'insieme delle attività che si svolgono all'interno del team studentesco, è stato possibile individuare le entità fondamentali coinvolte, come i componenti, le sessioni e i messaggi CAN. Si è quindi partiti da una visione globale per poi scendere nel dettaglio dei singoli componenti, utilizzando una strategia di progetto top-down per l'identificazione di attributi ed eventuali gerarchie, come componente, che si divide, in componente elettronico e meccanico.

Poi si è proceduto col collegare le entità tramite relazioni (strategia di progetto bottom-up), fino ad arrivare alla creazione di nuove categorie astratte e pertinenti (strategia di progetto inside-out).

Infine, si è raggiunta la realizzazione dello schema concettuale definitivo, che rappresenta in modo sintetico ma completo, la struttura del sistema informativo degli aspetti considerati.

2.2 Identificazione delle entità e relazioni fondamentali

Grazie ad una articolata analisi dei requisiti, è stato possibile individuare alcune entità fondamentali, che rappresentano gli elementi principali del sistema informativo del gruppo di lavoro.

Le massime entità individuate sono: "Componente", "Messaggio_CAN" e "Sessione".

Con "Componente" si intendono tutti i dispositivi presenti nel kart, per i quali è necessario memorizzare informazioni tecniche e descritti, e comprende componenti elettronici e non. Degli

esempi possono essere: batteria, sospensioni, motore e tanti altri, compresi i componenti più piccoli, come dischi e sensori.

Con “Messaggio_CAN” si indicano i possibili messaggi trasmissibili con l’uso del protocollo CAN, in cui sono stabiliti gli standard di comunicazione seriale tra reti digitali. Questa entità dovrà essere poi sviluppata in una specializzazione, in modo che presenti degli attributi caratterizzanti per ciascun tipo di segnale.

Con “Sessione” si fa riferimento all’insieme di tutte le prove in pista effettuate con un kart da uno specifico pilota, in uno specifico circuito. Collegata ad essa è necessario creare un’entità in cui memorizzare i dati in tempo reale provenienti dai recettori dei sensori.

L’individuazione di queste entità ha permesso di definire una prima struttura concettuale del sistema.

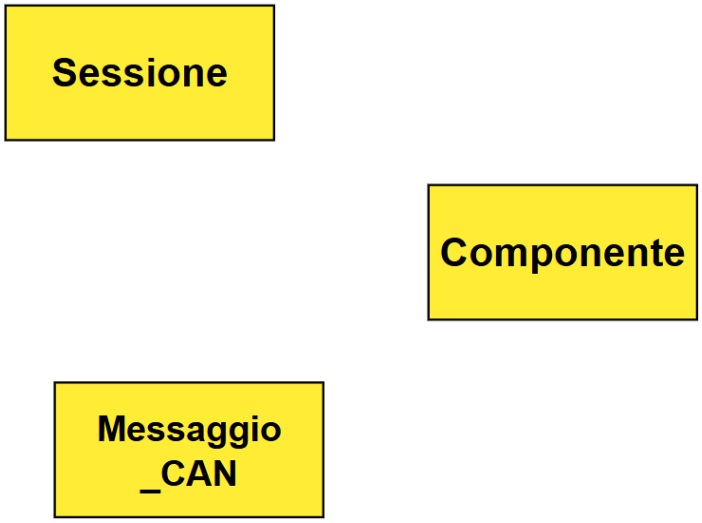


Figura 7 - Rappresentazione tramite modello E-R delle entità principali del sistema

2.3 Scheletro dello schema

Nello scheletro dello schema devono essere aggiunte alle entità principali individuate, anche le relazioni che intercorrono tra di esse, senza entrare nel dettaglio degli attributi.

Lo schema concettuale ruota attorno all’entità “Sessione”, che rappresenta il fulcro della memorizzazione dei dati e gestisce le varie prove di misurazione in pista.

Una delle entità principali collegate alla sessione tramite la relationship “Rilevazione”, è “Messaggio_CAN”, tutti gli standard per il trasferimento dei dati dai sensori. Un altro ruolo

fondamentale è svolto da “Componente”, l’insieme dei dispositivi nei kart, che viene associato a “Sessione” per mezzo di “Utilizzo”.

Qua sotto viene riportata una rappresentazione grafica delle relazioni, quindi i collegamenti, rappresentati da rombi, aggiunte alle entità fondamentali.

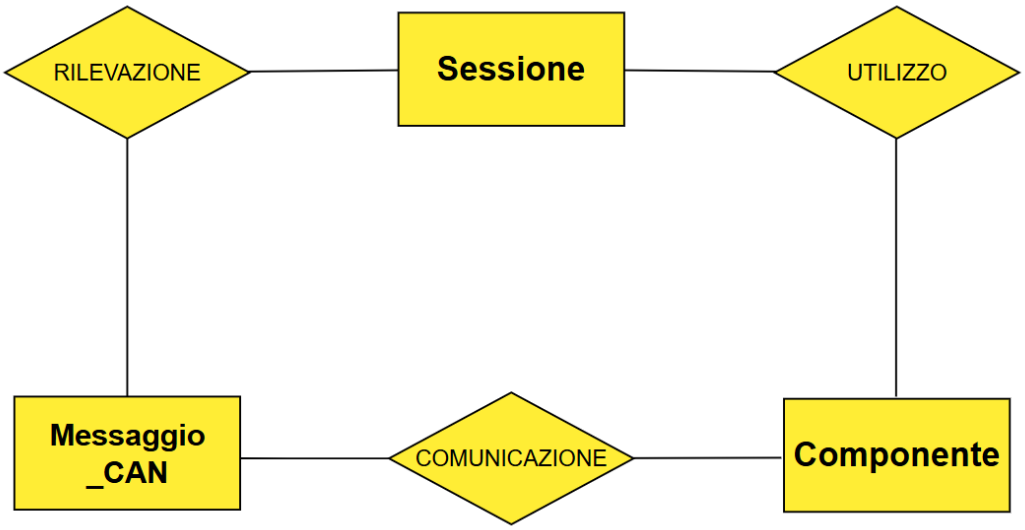


Figura 8 - Rappresentazione tramite modello E-R delle entità e relazioni fondamentali

2.4 Sviluppo delle componenti dello scheletro

In questo paragrafo inizia lo sviluppo dello schema dettagliato per la base di dati che permette di approfondire le diverse entità con le loro relazioni e i loro attributi. Le entità principali vengono trattate singolarmente, così da comprendere al meglio ogni decisione messa in atto.

Le relazioni permettono di associare uno o più componenti al messaggio CAN che loro stessi inviano, uno o più messaggi che vengono trasmessi e uno o più componenti che sono presenti nel kart a una determinata sessione. Perciò, si è utilizzata una strategia top-down per la specializzazione delle entità sopra citate e la definizione di attributi ed entità astratte, in modo tale che fossero coerenti con le richieste dei clienti della base di dati.

Ogni componente è associato al “Kart” su cui è montato, è identificato univocamente ed è descritto da informazioni tecniche e specifiche. Il “Componente” viene scomposto e reso una gerarchia che possiede le entità “Componente Elettronico” e “Componente Meccanico”, a sua volta “Componente Elettronico” si divide in più elementi, tra cui l’entità “Sensore”; quest’ultima possiede attributi propri, e verrà poi sviluppata come entità singola, per poter

essere ampliata ulteriormente. Il componente elettrico è il responsabile della ricezione e trasmissione dei segnali, tramite l'utilizzo del "Messaggio_CAN".

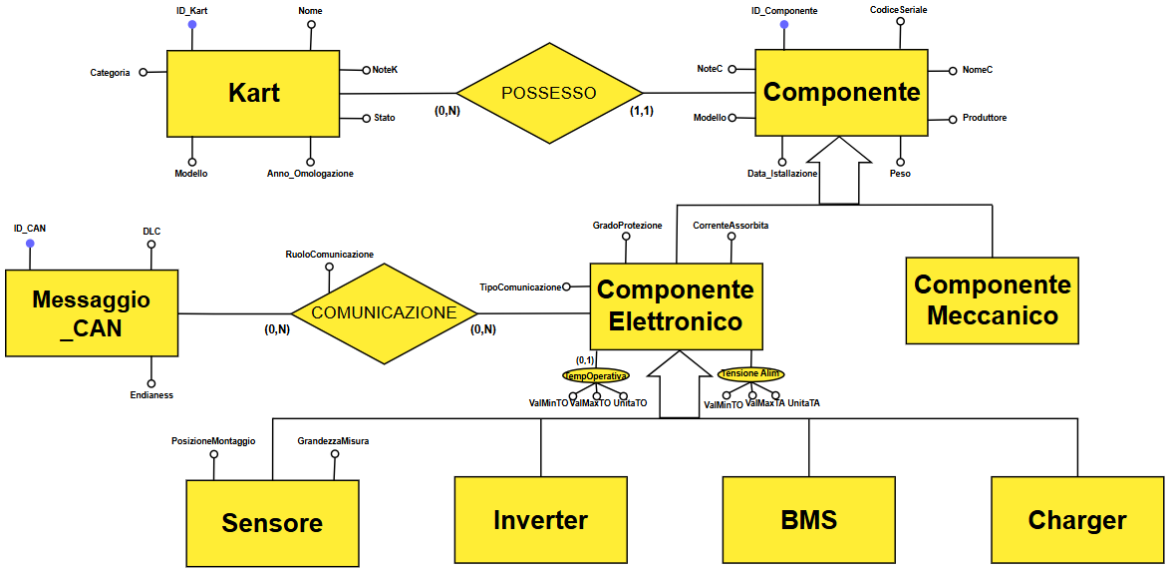


Figura 9 - Rappresentazione tramite modello E-R delle entità e relazioni associate all'entità "Componente"

L'entità "Messaggio_CAN" descrive le caratteristiche specifiche e comuni di trasmissione dei dati ricevuti dai sensori durante le misurazioni, e viene così associata all'entità "Segnale" in cui si specializzano le informazioni su alcuni componenti che inviano o ricevono messaggi CAN.

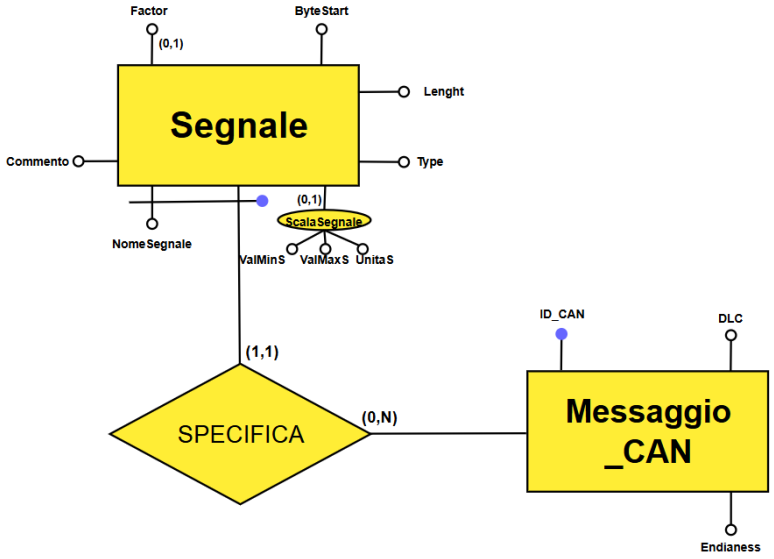


Figura 10 - Rappresentazione tramite modello E-R delle entità e relazioni associate all'entità "Messaggio CAN"

L’entità “Sessione” rappresenta la fase in cui vengono effettuate misurazioni dei dati di una prova di corsa del kart. Ogni sessione è associata a delle “Misurazioni”, con attributi timestamp e valore, al “Kart” utilizzato in pista e al suo “Pilota”, con le proprie informazioni personali.

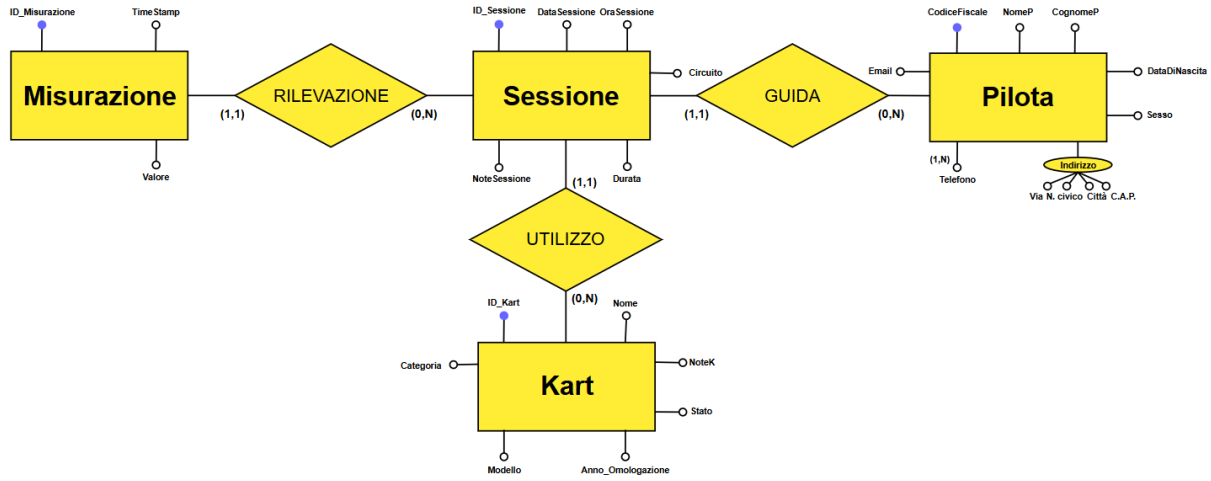


Figura 11 - Rappresentazione tramite modello E-R delle entità e relazioni associate all’entità “Sessione”

Integrando tutte le parti, si ottiene lo schema concettuale completo rappresentato di seguito.

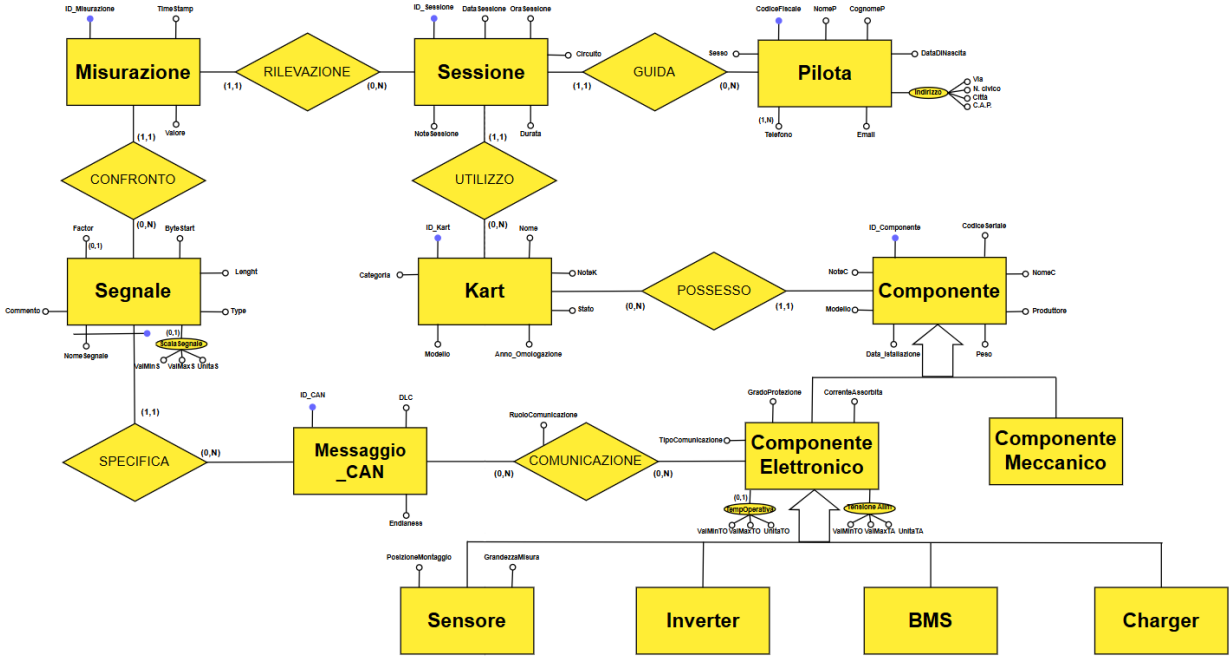


Figura 12 - Rappresentazione tramite modello E-R dello schema concettuale completo

2.5 Verbalizzazione del modello

Riassumendo ciò che è stato spiegato fino ad ora, lo schema concettuale progettato descrive l'organizzazione e la gestione dei dati di un kart ed è stato sviluppato seguendo un approccio di tipo misto. In particolare, la progettazione è partita dall'analisi delle singole entità principali del contesto, che sono state successivamente collegate tra loro per ottenere una rappresentazione complessiva coerente.

Lo schema rappresenta le principali aree fisiche e informative, includendo la gestione delle misurazioni e delle sessioni di allenamento/prova, l'organizzazione dei componenti ed i suoi sottogruppi.

Per ciascuna entità sono stati definiti gli attributi più rilevanti, mentre le relazioni descrivono in modo chiaro i legami e le interazioni tra i diversi componenti. Particolare attenzione è stata dedicata alla specializzazione dei componenti, distinguendo tra componenti elettronici e meccanici, e dividendo i componenti elettronici nei suoi sottogruppi.

Nel complesso, lo schema appare chiaro, ben strutturato e compatto, e consente di descrivere in modo efficace l'organizzazione, costituendo una base solida per eventuali sviluppi successivi del progetto.

2.6 Breve analisi di qualità dello schema

Il modello concettuale realizzato presenta un buon livello di qualità complessiva, risultando coerente e completo rispetto ai requisiti del progetto. Le entità individuate rappresentano in modo adeguato le principali aree di gestione, senza introdurre particolari superflui.

Le relazioni tra le entità sono state definite in maniera chiara e permettono di descrivere efficacemente i legami tra componenti, kart, piloti, misurazioni e sessioni. La scelta di porre l'entità sessione come elemento centrale rispecchia correttamente la realtà operativa del team universitario.

Il modello risulta piuttosto flessibile, in quanto consente future estensioni, come l'aggiunta di nuove tipologie di personale e componenti, alcune informazioni in più sui circuiti, sulle caratteristiche dei sensori e su ulteriori metodi di comunicazione.

Nel complesso, il modello soddisfa i requisiti individuati e costituisce una solida base per le successive fasi di progettazione.

2.7 Dizionario dei dati

2.7.1 Tabella delle entità

Per poter visualizzare in modo chiaro e schematico il sistema preso in analisi nella progettazione concettuale, vengono riportate tutte le entità coinvolte con una breve descrizione, i propri attributi e il loro identificatore chiave, cioè l'attributo che rende univoco ogni elemento della tabella.

Nome Entità	Descrizione	Attributi	Identificatore
Componente	Rappresenta il singolo componente presente nel kart	ID_Componente (numerico), CodiceSeriale (numerico), NomeC (stringa), Produttore (stringa), Peso (stringa), Data_Installazione (datetime), Modello (stringa), NoteC (testo)	ID_Componente
Componente Elettronico	Rappresenta il singolo componente elettronico presente nel kart, che invia e riceve dati per l'analisi	ID_Componente (numerico), GradoProtezione (stringa), CorrenteAssorbita (stringa), TipoComunicazione (stringa), TensioneAlimentazione [ValMin (numerico), ValMax (numerico), Unità (stringa)], TemperaturaOperativa [ValMin (numerico), ValMax (numerico), Unità (stringa)]	ID_Componente
Sensore	Rappresentazione l'insieme dei sensori del sistema, sottoinsieme del componente elettronico	ID_Componente (numerico), PosizioneMontaggio (stringa), GrandezzaMisura (stringa)	ID_Componente

Pilota	Rappresenta l'insieme dei piloti per la guida dei kart	CodiceFiscale (numerico), Nome (stringa), Cognome (stringa), DataDiNascita (data), Sesso (carattere), Indirizzo [(Via (stringa), NumeroCivico (numerico), Città (carattere), C.A.P. (stringa)], Email (stringa), Telefono (stringa)	CodiceFiscale
Messaggio CAN	Rappresenta l'entità astratta che descrive gli standard di comunicazione per il trasferimento dei dati	ID_CAN (numerico), DataLenghtCode (numerico), Endianess (stringa)	ID_CAN
Segnale	Rappresenta la specializzazione dei dati, per ogni componente, trasferiti con il messaggio CAN	NomeSegnale (stringa), ID_CAN (numerico), ScalaSegnale [ValMin (numerico), ValMax (numerico), Unità (stringa)], Type (numerico), Lenght (numerico), ByteStart (numerico), Factor (numerico), Commento (testo)	NomeSegnale, ID_CAN
Misurazione	Rappresenta le misure in tempo reale, ricevute dai sensori per l'analisi dei dati ed il confronto con gli standard CAN	ID_Misurazione (numerico), TimeStamp (timestamp), Valore (numerico)	ID_Misurazione

Sessione	Rappresenta il momento in cui vengono effettuate le prove di corsa per l'allenamento e la misurazione dei dati	ID_Sessione (numerico), DataSessione (data), OraSessione (orario), Durata (orario), Circuito (stringa), NoteSessione (testo)	ID_Sessione
Kart	Indica la vettura utilizzata per svolgere le gare di corsa	ID_Kart (numerico), Nome (stringa), Stato (stringa), Anno_Omologazione (anno), Modello (stringa), Categoria (stringa), NoteK (testo)	ID_Kart

2.7.2 Tabella delle relazioni

Come per le entità, vengono schematizzate anche le relationship tra entità e le loro caratteristiche.

Nome Relazione	Descrizione	Entità coinvolte	Attributi
Possesso	Associa Kart e Componente	Kart (0, N) – Componente (1, 1)	
Comunicazione	Associa Componente Elettronico e Messaggio_CAN	Componente Elettronico (0, N) – Messaggio_CAN (0, N)	RuoloComunicazione (stringa)
Specifica	Associa Segnale e Messaggio_CAN	Segnale (1, 1) – Messaggio_CAN (0, N)	
Confronto	Associa Segnale e Misurazione	Segnale (0, N) – Misurazione (1, 1)	
Utilizzo	Associa Kart e Sessione	Kart (0, N) – Sessione (1, 1)	

Rilevazione	Associa Misurazione e Sessione	Misurazione (1, 1) – Sessione (0, N)	
Guida	Associa Sessione e Pilota	Sessione (1 ,1) – Pilota (0, N)	

3 Progettazione logica

3.1 Tavola dei volumi

La progettazione logica è la fase in cui i componenti del sistema informativo, prima rappresentati graficamente e in modo concettuale, subiscono una trasformazione che li porta a prendere la tipica forma delle basi di dati relazionali: una struttura tabellare.

Viene qui sotto riportato il numero di elementi, o *n-uple*, che caratterizza ogni tabella rappresentativa di ciascun concetto, ampiamente illustrato in precedenza.

Concetto	Tipo	Volume
Componente	Entità	250
Componente Elettronico	Entità	170
Sensore	Entità	100
Messaggio CAN	Entità	50
Segnale	Entità	150
Misurazione	Entità	2000
Sessione	Entità	200
Pilota	Entità	4
Kart	Entità	3
Possesso	Relazione	250
Comunicazione	Relazione	5000
Specifica	Relazione	150
Confronto	Relazione	2000

Utilizzo	Relazione	200
Rilevazione	Relazione	2000
Guida	Relazione	200

3.2 Tavola delle operazioni

Nella fase di ristrutturazione logica è necessario conoscere, in maniera immediata, la frequenza delle operazioni fondamentali che poi verranno utilizzate dai clienti, tramite linguaggio SQL, per estrarre informazioni dalla base di dati.

Operazione	Frequenza
Operazione 1	1000 volte al mese
Operazione 2	2 volte al mese
Operazione 3	1 volta ogni 2 anni
Operazione 4	1 volta ogni 3 anni
Operazione 5	15 volte all'anno
Operazione 6	10 volte all'anno
Operazione 7	15 volte all'anno
Operazione 8	10 volte al mese
Operazione 9	10 volte al mese
Operazione 10	5 volte al mese
Operazione 11	2 volte al mese

Operazione 12	1 volta ogni 2 mesi
Operazione 13	1 volta al mese
Operazione 14	10 volte all’anno
Operazione 15	1 volta al mese
Operazione 16	10 volte al mese
Operazione 17	2 volte all’anno
Operazione 18	10 volte all’anno
Operazione 19	2 volte al mese
Operazione 20	2 volte al mese
Operazione 21	2 volte al mese
Operazione 22	2 volte al mese
Operazione 23	2 volte al mese
Operazione 24	1 volta al mese
Operazione 25	1 volta ogni 6 mesi
Operazione 26	1 volta al mese
Operazione 27	2 volta al mese
Operazione 28	5 volte all’anno
Operazione 29	1 volta ogni 5 anni

3.3 Ristrutturazione dello schema concettuale

La fase più importante della progettazione logica riguarda la ristrutturazione del modello E-R, in cui ci si occupa dell'eliminazione delle gerarchie presenti, poiché il modello relazionale non consente di rappresentare direttamente generalizzazioni (entità padre) e specializzazioni (entità figlie).

Tali gerarchie sono quindi eliminabili a patto di essere sostituite con opportune entità e relazioni. Sono possibili vari metodi per la sostituzione, che possono essere scelti in base alle varie esigenze:

1. Accorpamento delle entità figlie nell'entità genitore
2. Accorpamento dell'entità genitore nelle entità figlie
3. Sostituzione delle gerarchie con delle relazioni
4. Soluzioni ibride delle prime

Nel modello considerato sono presenti due gerarchie fondamentali, di seguito analizzate singolarmente:

- Generalizzazione parziale "Componente", che si divide in alcuni dei suoi concetti: "Componente Elettronico" e "Componente Meccanico".

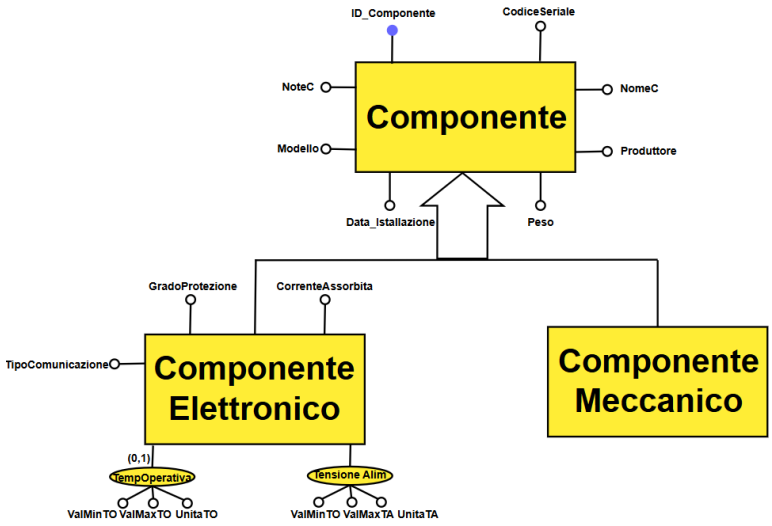


Figura 13 - Rappresentazione tramite modello E-R delle entità e relazioni appartenenti alla gerarchia "Componente"

Per l'eliminazione della generalizzazione "Componente" si è deciso di adottare una soluzione ibrida, tra il primo ed il terzo metodo, poiché si tratta di una gerarchia a più livelli con relazioni e attributi associati alle entità figlie.

L'entità "Componente Meccanico" non presenta attributi specifici né relazioni, di conseguenza, gli accessi all'entità padre e alle entità figlie risultano contestuali. Diversamente, l'entità "Componente Elettronico" possiede alcuni attributi propri che comporterebbe l'inserimento di diversi valori NULL nel caso di accorpamento di tutte le entità figlie nel genitore, con conseguente spreco di memoria. Inoltre, "Componente Elettronico" è coinvolto in due relazioni indipendenti alle altre entità e le operazioni sui suoi dati sono significativamente più frequenti rispetto a quelle delle altre. Pertanto, mantenere tutti gli elementi in un'unica tabella potrebbe determinare un costo computazionale elevato durante le interrogazioni.

Infine, il secondo metodo, cioè l'accorpamento dell'entità padre nelle figlie, risulterebbe poco efficace: le figlie erediterebbero tutti gli attributi dell'entità padre, portando alla generazione di numerose ridondanze.

Il modello viene quindi modificato in questo modo:

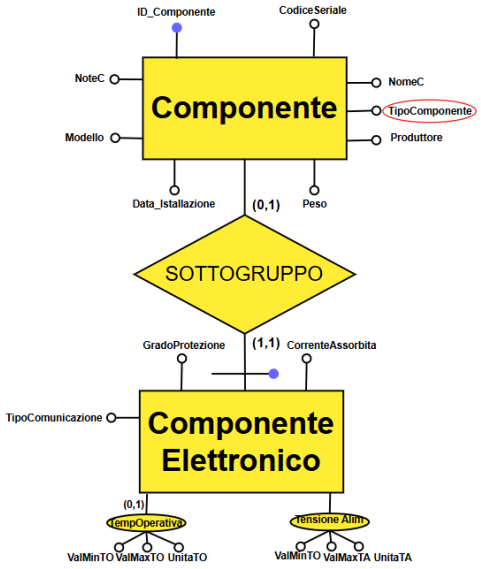


Figura 14 - Rappresentazione tramite modello E-R ristrutturata delle entità e relazioni appartenenti alla gerarchia "Componente"

Nel processo di ristrutturazione si è inserito l'attributo "TipoComponente" nell'entità "Componente", dove si memorizza il tipo di componente, ed è stata creata una relationship tra "Componente" e "Componente Elettronico", denominata "Sottogruppo", con cardinalità (0,1) e (1,1).

- Generalizzazione parziale “Componente Elettronico”, creando così una gerarchia a più livelli, che si divide in alcuni dei suoi componenti: “Sensore”, “Inverter”, “BMS” e “Charger”.

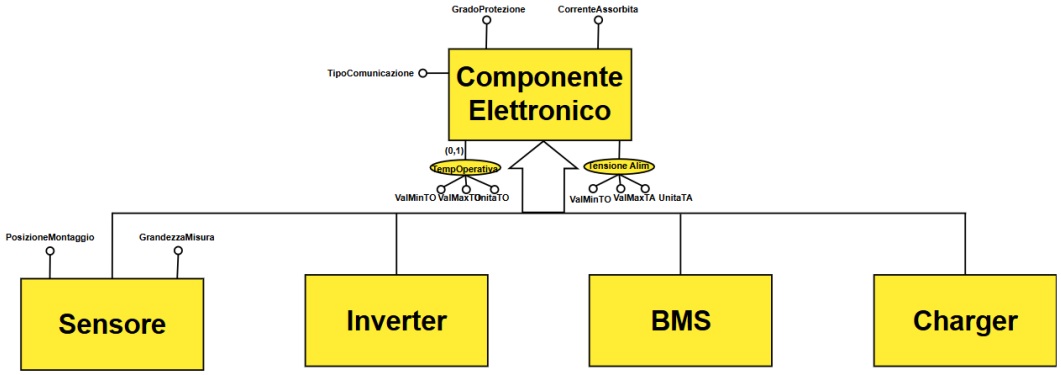


Figura 15 - Rappresentazione tramite modello E-R delle entità e relazioni appartenenti alla gerarchia “Componente Elettronico”

Per l’eliminazione della generalizzazione “Componente Elettronico” si è adottare nuovamente il metodo misto, ovvero l’accorpamento delle entità figlie nell’entità padre e la creazione di una relazione con l’entità “Sensore”. Non è necessario quindi ristrutturare il modello E-R tramite il secondo metodo, che porterebbero nuovamente ad una ulteriore aggiunta di componenti ridondanti.

Il modello viene quindi modificato in questo modo:

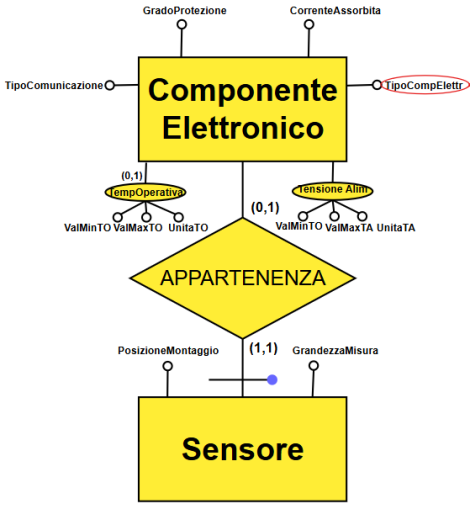


Figura 16 - Rappresentazione ristrutturata delle entità e relazioni appartenenti alla gerarchia “Componente Elettronico” nel modello E-R

E’ stato inserito l’attributo “TipoCompElettr”, che indica il nome del componente elettronico.

Per semplificare il modello E-R sono presenti due generalizzazioni parziali; sono state inserite unicamente le entità figlie che risultano necessarie per modellare le relazioni del sistema e per la visualizzazione dello schema, evitando di introdurre elementi non utili alla visione.

3.4 Partizionamento/accorpamento di concetti

3.4.1 Partizionamento o accorpamento di concetti

Per quanto riguarda il partizionamento o accorpamento, non sono state individuate entità per le quali risulti conveniente effettuare operazioni di partizionamento o di accorpamento.

L’unica ipotesi di accorpamento presa in considerazione riguarda la relazione “Rilevazione” tra le entità “Misurazione” e “Sessione”; tale soluzione è stata tuttavia scartata in quanto, oltre alla grande quantità di attributi di entrambe le entità, nella maggior parte dei casi, gli accessi alle due entità avvengono in modo indipendente.

3.4.2 Eliminazione degli attributi multivalore

Gli attributi multivalore portano a relazioni scorrette, per questo motivo, tali attributi devono essere sostituiti con relazioni ed entità dedicate.

Nel modello è presente un unico attributo appartenente a questa selezione, denominato “Telefono”, posseduto dall’entità “Pilota”, finalizzato a registrare più recapiti telefonici di un pilota per eventuali necessità.

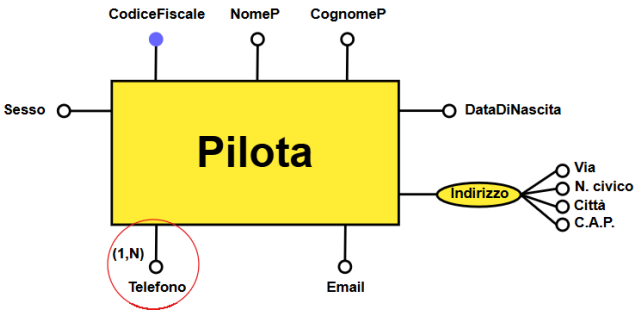


Figura 17 - Rappresentazione dell'attributo multivalore "Telefono" appartenente all'entità "Pilota"

Come già spiegato, è quindi necessario eliminare gli attributi multivalore, di conseguenza, si è provveduto a rimuovere l'attributo "Telefono" dalla suddetta entità trasformando anch'esso in entità con una relazione "Contatto_Pilota" con cardinalità (1, N):

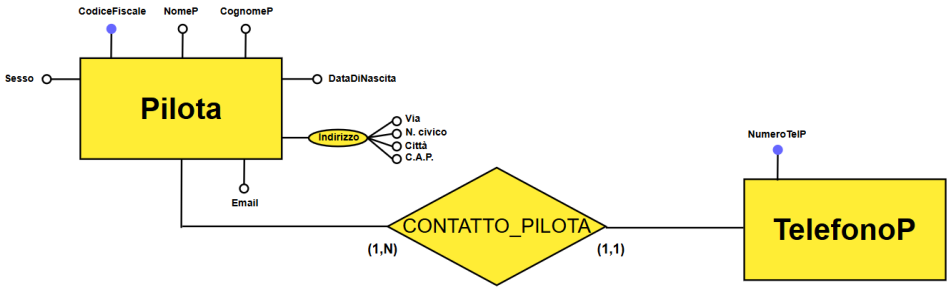


Figura 18 - Rappresentazione dell'eliminazione dell'attributo e della trasformazione in entità "Telefono"

Infine, come accade con le gerarchie, il modello relazionale non è in grado di rappresentare entità dotate di attributi composti, pertanto, durante la fase di ristrutturazione, questo genere di attributi sono stati scomposti in elementi più semplici.

3.5 Elenco degli identificatori principali

Di seguito sono esposti gli identificatori delle entità presenti nel modello E-R, cioè gli attributi, e quindi le colonne, che caratterizzano o identificano ciascun elemento delle tabelle.

Entità	Identificatore
Componente	ID_Componente
Componente Elettronico	ID_Componente
Sensore	ID_Componente
Kart	ID_Kart
Messaggio_CAN	ID_CAN
Misurazione	ID_Misurazione

Segnale	ID_CAN, NomeSegnale
Sessione	ID_Sessione
Pilota	CodiceFiscale
TelefonoP	NumeroTelP

Alcune spiegazioni sugli identificatori:

- Entità Componente: è stato scelto di utilizzare un identificatore ID, e non il codice seriale, in quanto i componenti possono essere lo stesso prodotto, ma in posizioni diverse. L'uso di un ID consente pertanto di identificare in modo univoco ciascun componente, evitando ambiguità.
- Entità Segnale: è stato scelto di utilizzare un identificatore ID anche per i segnali, con l'aggiunta del nome del segnale, poiché "Segnale" può essere considerato un sottogruppo di "Messaggio_CAN" con l'aggiunta di alcune caratteristich specifichee. L'utilizzo di un ID garantisce quindi una corretta e univoca identificazione dei segnali e messaggi CAN.

3.6 Rappresentazione grafica del modello concettuale ristrutturato

Viene qui proposto la rappresentazione grafica finale della ristrutturazione del modello concettuale della base di dati, da poter, in fase conclusiva, implementare in un software, tramite linguaggio SQL.

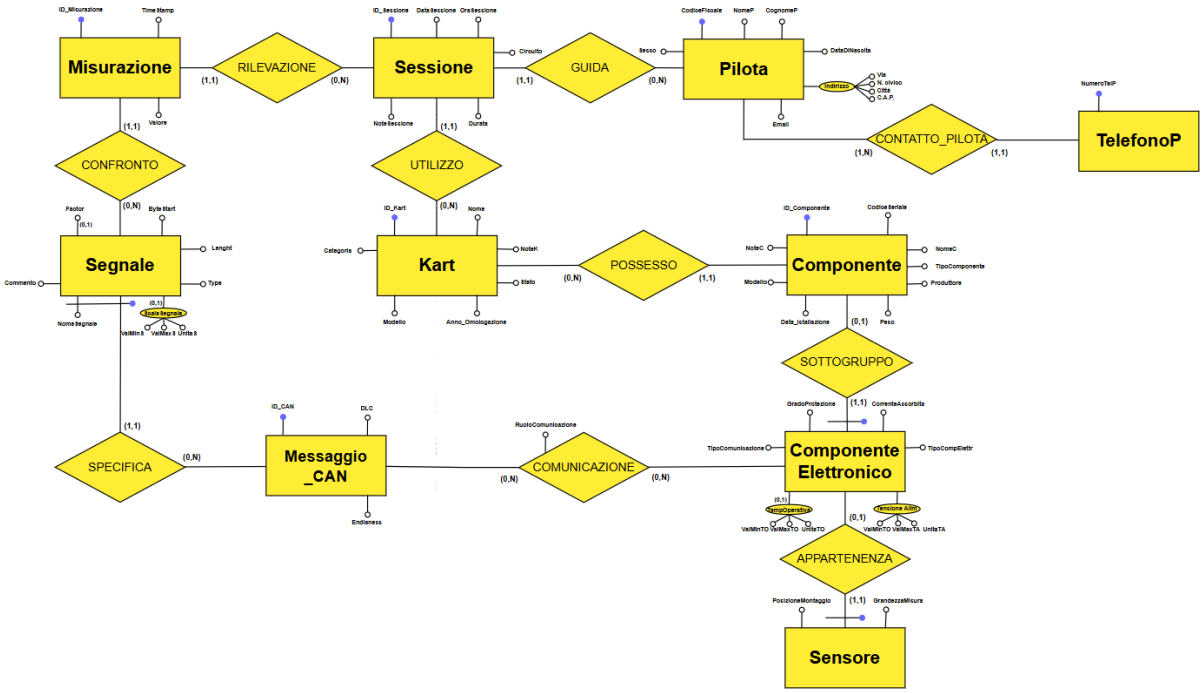


Figura 19 - Rappresentazione tramite modello E-R dello schema concettuale ristrutturato

3.7 Traduzione verso il modello relazionale

Per semplificare al massimo lo schema e tradurre le entità e le relationship in tabelle, è necessario procedere con alcune accortezze, seguendo diverse tecniche di traduzione che si basano sulla cardinalità delle relationship (MOLTI A MOLTI, UNO A MOLTI e UNO A UNO).

Entità - Relazione	Traduzione	Spiegazione
Componente	Componente (ID_Componente, Idx_Kart, CodiceSeriale, NomeC, TipoComponente, Produttore, Peso, Data_Installazione, Modello, NoteC)	Relationship UNO A UNO con Componente Elettronico (con identificatore esterno) e UNO A MOLTI con Kart

Componente Elettronico	ComponenteElettronico (ID_componente, TipoCompElett, ValMinTensAlim, ValMaxTensAlim, UnitaTensAlim, ValMinTempOperativa, ValMaxTempOperativa, UnitaTempOperativa, GradoProtezione, CorrenteAssorbita, TipoComunicazione)	Relationship UNO A UNO con Sensore (con identificatore esterno) e relationship MOLTI A MOLTI con Messaggio_CAN
Sensore	Sensore (Idx_Componente, PosizioneMontaggio, GrandezzaMisurata)	Relationship UNO A UNO con ComponenteElettronico (con identificatore esterno)
Comunicazione	Comunicazione (Idx_componente, ID_can, RuoloComunicazione)	Relationship MOLTI A MOLTI tra Messaggio_CAN e Componente Elettronico
Messaggio_CAN	Messaggio_CAN (ID_CAN, DataLenghtCode, Endianess)	Relationship UNO A MOLTI con Segnale (con identificatore esterno) e MOLTI A MOLTI con ComponenteElettronico
Segnale	Segnale (NomeSegnale, ID_can, ByteStart, Lenght, Type, ValMinScalaSegnale, ValMaxScaleSegnale, UnitaScalaSegnale, Factor, Commento)	Relationship UNO A MOLTI con Misurazione e Messaggio_CAN
Misurazione	Misurazione (ID_Misurazione, nomeSegnale, Idx_CAN, ID_sessione, TimeStamp, Valore)	Relationship UNO A MOLTI con Sessione e Segnale

Sessione	Sessione (ID_Sessione, ID_kart, codiceFiscale, DataSessione, OraSessione, Durata, Circuito, NoteSessione)	Relationship UNO A MOLTI con Kart e Pilota
Pilota	Pilota (CodiceFiscale, NomeP, CognomeP, Sesso, DataDiNascita, Via, NCivico, Citta, CAP, Email)	Relationship UNO A MOLTI con Sessione e TelefonoP
TelefonoP	TelefonoP (NumeroTelP, codFiscale)	Relationship UNO A MOLTI con Pilota
Kart	Kart (ID_Kart, Nome, Modello, Categoria, Stato, Anno_Omologazione, NoteK)	Relationship UNO A MOLTI con Sessione e Componente

Si illustra la definitiva traduzione delle entità, una volta elencate singolarmente, con le relative chiavi esterne che permettono di non violare i vincoli di riferimento tra tabelle.

Traduzione	**Vincolo di riferimento**
Componente (ID_Componente, Idx_Kart, CodiceSeriale, NomeC, TipoComponente, Produttore, Peso, Data_Installazione, Modello, NoteC)	Idx_Kart -> Kart.ID_Kart
ComponenteElettronico (ID_componente, TipoCompElett, ValMinTensAlim, ValMaxTensAlim, UnitaTensAlim, ValMinTempOperativa, ValMaxTempOperativa, UnitaTempOperativa, GradoProtezione, CorrenteAssorbita, TipoComunicazione)	ID_componente -> Componente.ID_Componente
Sensore (Idx_Componente, PosizioneMontaggio, GrandezzaMisurata)	Idx_Componente -> Componente.ID_Componente

Comunicazione (Idx_componente, ID_can, RuoloComunicazione)	Idx_componente -> Componente.ID_Componente ID_can -> Messaggio_CAN.ID_CAN
Messaggio_CAN (ID_CAN, DataLenghtCode, Endianess)	
Segnale (NomeSegnale, ID_can, ByteStart, Lenght, Type, ValMinScalaSegnale, ValMaxScaleSegnale, UnitaScalaSegnale, Factor, Commento)	ID_can -> Messaggio_CAN.ID_CAN
Misurazione (ID_Misurazione, nomeSegnale, Idx_CAN, ID_sessione, TimeStamp, Valore)	nomeSegnale -> Segnale.NomeSegnale Idx_CAN -> Messaggio_CAN.ID_CAN ID_sessione -> Sessione.ID_Sessione
Sessione (ID_Sessione, ID_kart, codiceFiscale, DataSessione, OraSessione, Durata, Circuito, NoteSessione)	ID_kart -> Kart.ID_Kart codiceFiscale -> Pilota.CodiceFiscale
Pilota (CodiceFiscale, NomeP, CognomeP, Sesso, DataDiNascita, Via, NCivico, Citta, CAP, Email)	
TelefonoP (NumeroTelP, codFiscale)	codFiscale -> Pilota.CodiceFiscale
Kart (ID_Kart, Nome, Modello, Categoria, Stato, Anno_Omologazione, NoteK)	

3.8 Normalizzazione

In questa fase di verifica, analizzando lo schema tradotto nel modello relazionale, si è giunti alla conclusione che tutte le associazioni sono in forma normale di Boyce-Codd, di conseguenza, non dovrebbero verificarsi ridondanze o anomalie che possano compromettere la "qualità" della base di dati.

Entità	Commento
Componente	Non esistono dipendenze non banali fra gli attributi
ComponenteElettronico	Non esistono dipendenze non banali fra gli attributi
Sensore	Non esistono dipendenze non banali fra gli attributi
Comunicazione	Non esistono dipendenze non banali fra gli attributi
Messaggio_CAN	Non esistono dipendenze non banali fra gli attributi
Segnale	Non esistono dipendenze non banali fra gli attributi
Misurazione	Non esistono dipendenze non banali fra gli attributi
Sessione	Non esistono dipendenze non banali fra gli attributi
Pilota	Non esistono dipendenze non banali fra gli attributi
TelefonoP	Non esistono dipendenze non banali fra gli attributi
Kart	Non esistono dipendenze non banali fra gli attributi

4 Codifica SQL e Testing

4.1 Definizione dello schema e immagini successive all'inserimento dei dati

Questa fase rappresenta la parte conclusiva della progettazione della base di dati, nella quale lo schema precedentemente definito viene implementato concretamente attraverso l'utilizzo di codice SQL nell'applicazione web phpMyAdmin. In particolare, vengono create le tabelle e concretizzati i propri attributi, chiavi e vincoli di integrità, per poi procedere con l'inserimento delle istanze necessarie al popolamento della base di dati.

Si riporta di seguito lo schema di definizione delle tabelle e i codici utilizzati, con le relative immagini delle istanze inserite.

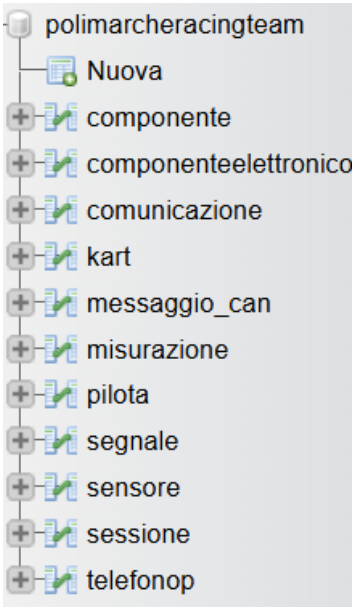


Figura 20 - Schema di definizione delle tabelle rappresentanti la base di dati

COMPONENTE

```
CREATE TABLE Componente (
	ID_Componente INT AUTO_INCREMENT PRIMARY KEY,
	Idx_Kart INT,
	CodiceSeriale VARCHAR(50),
	NomeC VARCHAR(100),
	TipoComponente VARCHAR(100),
	Produttore VARCHAR(100),
	Peso DECIMAL(10,2),
	Data_Installazione DATE,
	Modello VARCHAR(100),
	NoteC TEXT
	FOREIGN KEY(Idx_Kart)
		REFERENCES Kart(ID_Kart)
		ON UPDATE CASCADE
		ON DELETE NO ACTION
);
```

ID_Componente	Idx_Kart	CodiceSeriale	NomeC	TipoComponente	Produttore	Peso	Data_Installazione	Modello	NoteC
101	1	K1-IMU-001	IMU	IMU	AvioRace SRL	0.06	2025-03-10	IMU rev1.3	Firmware 2.09, CAN 0x100-0x102, range +-12G / +-25...
102	1	K1-INV-001	Inverter 1	Inverter	AMK	3.20	2025-03-10	AMK-INV	Inverter motore sinistro
103	1	K1-INV-002	Inverter 2	Inverter	AMK	3.20	2025-03-10	AMK-INV	Inverter motore destro

COMPONENTE ELETTRONICO

```
CREATE TABLE ComponenteElettronico (
	ID_componente INT PRIMARY KEY,
	TipoCompElett VARCHAR(100) NOT NULL,
	ValMinTensAlim DECIMAL(10,2) NOT NULL,
	ValMaxTensAlim DECIMAL(10,2) NOT NULL,
	CHECK(ValMinTensAlim <= ValMaxTensAlim),
	UnitaTensAlim ENUM('V', 'VDC', 'VDA') NOT NULL,
	CHECK(UnitaTensAlim == 'V' OR UnitaTensAlim == 'VDC' OR UnitaTensAlim ==
	'VDA'),
	ValMinTempOperativa DECIMAL(10,2),
	ValMaxTempOperativa DECIMAL(10,2),
	CHECK(ValMinTempOperativa <= ValMaxTempOperativa),
	UnitaTempOperativa VARCHAR(20),
	GradoProtezione VARCHAR(20) NOT NULL,
	CorrenteAssorbita DECIMAL(10,2) NOT NULL,
	TipoComunicazione VARCHAR(100 NOT NULL),
	FOREIGN KEY(Id_componente)
		REFERENCES Componente(ID_Componente)
		ON UPDATE CASCADE
		ON DELETE CASCADE
);
```

ID_componente	TipoCompElett	ValMinTensAlim	ValMaxTensAlim	UnitaTensAlim	ValMinTempOperativa	ValMaxTempOperativa	UnitaTempOperativa
101	IMU	8.00	16.00	V	0.00	105.00	C
102	Inverter	18.00	32.00	VDC	40.00	105.00	F
103	Inverter	18.00	32.00	V	-20.00	70.00	C

SENSORE

```
CREATE TABLE Sensore (
	Idx_Componente INT PRIMARY KEY NOT NULL,
	PosizioneMontaggio VARCHAR(100) NOT NULL,
	GrandezzaMisurata VARCHAR(100) NOT NULL,
	FOREIGN KEY (Idx_Componente)
		REFERENCES Componente(ID_Componente)
		ON UPDATE CASCADE
		ON DELETE NO ACTION
);
```

Idx_Componente	PosizioneMontaggio	GrandezzaMisurata
101	Baricentro, sotto il sedile	Accelerazioni, velocita angolari, campo magnetico
107	Pedale acceleratore	Posizione angolare pedale (APPS1)
108	Pedale acceleratore	Posizione angolare pedale (APPS2)

COMUNICAZIONE

```
CREATE TABLE Comunicazione (
	Idx_componente INT NOT NULL,
	ID_can INT NOT NULL,
	RuoloComunicazione ENUM('Trasmette', 'Riceve') NOT NULL,
	PRIMARY KEY (Idx_componente, ID_can),
	FOREIGN KEY (Idx_componente)
		REFERENCES Componente(ID_Componente)
		ON UPDATE CASCADE
		ON DELETE NO ACTION,
	FOREIGN KEY (ID_can)
		REFERENCES Messaggio_CAN(ID_CAN)
		ON UPDATE CASCADE
		ON DELETE NO ACTION
);
```

Idx_componente	ID_can	RuoloComunicazione
101	256	Trasmette
101	257	Trasmette
101	258	Trasmette

MESSAGGIO_CAN

```
CREATE TABLE Messaggio_CAN (
	ID_CAN INT AUTO_INCREMENT PRIMARY KEY,
	DataLenghtCode INT NOT NULL,
	Endianess ENUM('BIG', 'LITTLE') NOT NULL
);
```

ID_CAN	DataLenghtCode	Endianess
256	8	BIG
257	8	BIG
258	6	LITTLE

SEGNALE

```
CREATE TABLE Segnale (
	NomeSegnale VARCHAR(100) NOT NULL,
	ID_can INT NOT NULL,
	ByteStart INT NOT NULL,
	Lenght INT NOT NULL,
	Type VARCHAR(50) NOT NULL,
	ValMinScalaSegnale DECIMAL(10,2),
	ValMaxScaleSegnale DECIMAL(10,2),
	CHECK(ValMinScalaSegnale <= ValMaxScaleSegnale),
	UnitaScalaSegnale VARCHAR(50),
	Factor DECIMAL(10,4),
	Commento TEXT,
	PRIMARY KEY (NomeProdotto, ID_can),
	FOREIGN KEY (ID_can)
		REFERENCES Messaggio_CAN(ID_CAN)
		ON UPDATE CASCADE
		ON DELETE NO ACTION
);
```

NomeSegnale	ID_can	ByteStart	Lenght	Type	ValMinScalaSegnale	ValMaxScaleSegnale	UnitaScalaSegnale	Factor	Commento
Acceleration_X	256	0	16	S16	-12.00	12.00	G	0.0010	Accelerazione asse X
Acceleration_Y	256	2	16	S16	-12.00	12.00	G	0.0010	Accelerazione asse Y
Acceleration_Z	256	4	16	S16	-12.00	12.00	G	0.0010	Accelerazione asse Z

MISURAZIONE

```
CREATE TABLE Misurazione (
	ID_Misurazione INT AUTO_INCREMENT PRIMARY KEY,
	nomeSegnale VARCHAR(100) NOT NULL,
	Idx_CAN INT NOT NULL,
	ID_sessione INT NOT NULL,
	TimeStamp DATETIME NOT NULL,
	Valore DECIMAL(10,2),
	FOREIGN KEY (Idx_CAN)
		REFERENCES Messaggio_CAN(ID_CAN)
		ON UPDATE CASCADE
		ON DELETE NO ACTION,
	FOREIGN KEY (ID_sessione)
		REFERENCES Sessione(ID_sessione)
		ON UPDATE CASCADE
		ON DELETE NO ACTION
);
```

ID_Misurazione	nomeSegnale	Idx_CAN	ID_sessione	TimeStamp	Valore
1	Acceleration_X	256	1	2026-09-05 09:31:00	0.85
2	Car_Speed	260	1	2026-09-05 09:31:01	42.50
3	INV1_ACT_SPEED	643	1	2026-09-05 09:31:02	7800.00

SESSIONE

```
CREATE TABLE Sessione (
	ID_Sessione INT AUTO_INCREMENT PRIMARY KEY,
	ID_kart INT NOT NULL,
	codiceFiscale VARCHAR(16) NOT NULL,
	DataSessione DATE NOT NULL,
	OraSessione TIME NOT NULL,
	Durata TIME NOT NULL,
	Circuito VARCHAR(100) NOT NULL,
	NoteSessione TEXT,
	FOREIGN KEY (ID_kart)
		REFERENCES Kart(ID_Kart)
		ON UPDATE CASCADE
		ON DELETE NO ACTION,
	FOREIGN KEY (codiceFiscale)
		REFERENCES Pilota(CodiceFiscale)
		ON UPDATE CASCADE
		ON DELETE NO ACTION
);
```

ID_Sessione	ID_kart	codiceFiscale	DataSessione	OraSessione	Durata	Circuito	NoteSessione
1	1	RSSMRC02A15C351X	2026-09-05	09:30:00	00:20:00	Kartodromo di Franciacorta	Test accelerazione e frenata
2	2	VRDLCU03C22G273K	2026-09-05	11:00:00	00:15:00	Kartodromo di Franciacorta	Prova engine map ECO
3	3	BNCGLI01H50C351P	2026-09-12	10:00:00	00:10:00	Circuito di Pomposa	Taratura sensori pedale

PILOTA

```
CREATE TABLE Pilota (
	CodiceFiscale VARCHAR(16) PRIMARY KEY,
	NomeP VARCHAR(40) NOT NULL,
	CognomeP VARCHAR(40) NOT NULL,
	Sesso ENUM('F', 'M') NOT NULL,
	DataDiNascita DATE NOT NULL,
	Via VARCHAR(50) NOT NULL,
	NumCivico VARCHAR(10) NOT NULL,
	Citta CHAR(2) NOT NULL,
	CAP CHAR(5) NOT NULL,
	CHECK (LENGTH(CAP) = 5),
	Email VARCHAR(50) NOT NULL UNIQUE
);
```

CodiceFiscale	NomeP	CognomeP	Sesso	DataDiNascita	Via	NCivico	Citta	CAP	Email
BNCGLI01H50C351P	Giulia	Bianchi	F	2001-06-10	Corso Italia	78	CT	95127	giulia.bianchi@teamkart.it
FRRDVD04M10F205T	Davide	Ferrari	M	2004-08-10	Via Giuseppe Verdi	9	MI	20121	davide.ferrari@teamkart.it
RSSMRC02A15C351X	Marco	Rossi	M	2002-01-15	Via Etnea	120	AN	60130	marco.rossi@teamkart.it

TELEFONOP

```
CREATE TABLE TelefonoP (
	NumeroTelP VARCHAR(15) PRIMARY KEY,
	codFiscale CHAR(16) NOT NULL,
	FOREIGN KEY(CodFiscale)
		REFERENCES Pilota(CodiceFiscale)
		ON UPDATE NO ACTION
		ON DELETE CASCADE
);
```

NumeroTelP	codFiscale
0952233445	BNCGLI01H50C351P
3473456789	BNCGLI01H50C351P
3204567890	FRRDVD04M10F205T

KART

```
CREATE TABLE Kart (
	ID_Kart INT AUTO_INCREMENT PRIMARY KEY,
	Nome VARCHAR(100) NOT NULL,
	Modello VARCHAR(100),
	Categoria VARCHAR(100),
	Stato ENUM('Operativo', 'Dismesso', 'In manutenzione') NOT NULL,
	Anno_Omologazione YEAR NOT NULL,
	NoteK TEXT
);
```

ID_Kart	Nome	Modello	Categoria	Stato	Anno_Omologazione	NoteK
1	Falcon	CRG Road Rebel	KZ2	Operativo	2025	Kart principale, doppio inverter AMK, IMU AvioRace
2	Thunder	Birel ART	KZ	Dismesso	2024	Kart di riserva, stessa configurazione del Falcon
3	Thunder	Birel ART	KZ	In manutenzione	2026	Nuovo telaio, in fase di taratura sensori pedale

4.2 Codifica delle operazioni e codice che ne verificano l'esecuzione

Procedendo con la fase conclusiva, vengono implementate le principali operazioni effettuabili sulla base di dati attraverso interrogazioni scritte in linguaggio SQL. Le operazioni individuate rappresentano alcune delle richieste più frequenti che possono essere attuate dagli utenti e permettono la gestione e l'analisi quotidiana delle relazioni tra le diverse tabelle della base di dati.

Vengono riportate di seguito le interrogazioni SQL relative alle operazioni considerate, accompagnate dai risultati della loro esecuzione, al fine di verificarne il corretto funzionamento.

1) Inserimento di una nuova misurazione (1000 volte al mese)

```
/*prima si inseriscono il sensore e sessione se non presenti nella tabella e poi la misurazione*/
INSERT INTO Misurazione (ID_Misurazione, NomeSegnale, Idx_CAN, ID_sessione, TimeStamp, Valore)
VALUES(...);
```

2) Inserimento di una nuova sessione di prova (2 volte al mese)

```
INSERT INTO Sessione (ID_Sessione, ID_kart, CodiceFiscale, DataSessione, OraSessione, Durata, Circuito, NoteSessione)
VALUES(...);
```

3) Inserimento di un nuovo pilota (1 volta ogni 2 anni)

```
INSERT INTO Pilota (CodiceFiscale, NomeP, CognomeP, Sesso, DataDiNascita, Via, NumCivico, Citta, CAP, Email)
VALUES(...);
/*poi, se presente, si inserisce il numero di telefono */
INSERT INTO TelefonoP (NumeroTelP, CodFiscale)
VALUES (...);
```

4) Inserimento di un nuovo kart (1 volta ogni 3 anni)

```
INSERT INTO Kart (ID_Kart, Nome, Modello, Categoria, Stato, Anno_Omologazione, NoteK)
VALUES (...);
```

5) Inserimento di un nuovo componente elettronico (15 volte all'anno)

```
/*prima si inserisce il componente*/
INSERT INTO Componente (ID_Componente, Idx_Kart, CodiceSeriale, NomeC, TipoComponente, Produttore, Peso, Data_Installazione, Modello, NoteC)
VALUES (...);
/*poi si aggiunge il componente elettronico*/
INSERT INTO ComponenteElettronico (ID_componente, TipoCompElett, ValMinTensAlim, ValMaxTensAlim, UnitaTensAlim, ValMinTempOperativa, ValMaxTempOperativa, UnitaTempOperativa, GradoProtezione, CorrenteAssorbita, TipoComunicazione)
VALUES (...);
```

6) Inserimento di un nuovo sensore (10 volte all'anno)

```
/*prima di inserire il sensore si utilizza il procedimento di inserimento
dell'operazione precedente (operazione 5)*/
INSERT INTO Sensore (Idx_Componente, PosizioneMontaggio, GrandezzaMisurata)
VALUES (...);
```

7) Inserimento di una nuova comunicazione tra un componente elettronico e un messaggio CAN (15 volte all'anno)

```
/*si considerano già inseriti i valori di componente elettronico e messaggio CAN*/
INSERT INTO Comunicazione (Idx_componente, ID_can, RuoloComunicazione)
VALUES (...);
```

8) Consultazione dei segnali contenuti in un messaggio CAN (10 volte al mese)

```
SELECT NomeSegnale, ByteStart, Lenght, Type, ValMinScalaSegnale,
ValMaxScaleSegnale, UnitaScalaSegnale, Factor, Commento
FROM Segnale
WHERE ID_can = 272;
```

NomeSegnale	ByteStart	Lenght	Type	ValMinScalaSegnale	ValMaxScaleSegnale	UnitaScalaSegnale	Factor	Commento
VelocitaAnteriore	0	16	Unsigned	0.00	250.00	km/h	0.0100	Velocità rilevata dal sensore sulla ruota anterior...

9) Consultazione delle misurazioni in una sessione (in media 10 volte al mese)

```
SELECT ID_Misurazione, nomeSegnale, TimeStamp, Valore
FROM Misurazione
WHERE ID_sessione = 3;
```

ID_Misurazione	nomeSegnale	TimeStamp	Valore
13	Acceleration_X	2026-09-12 10:01:00	0.30
14	Car_Speed	2026-09-12 10:01:01	18.40
15	INV1_ACT_SPEED	2026-09-12 10:01:02	3100.00

10) Consultazione delle informazioni di una sessione con relativi pilota e kart (5 volte al mese)

```
SELECT Sessione.ID_Sessione, Sessione.DataSessione, Sessione.OraSessione,
Sessione.Durata, Sessione.Circuito, Sessione.NoteSessione, Pilota.NomeP,
Pilota.CognomeP, Kart.ID_Kart, Kart.Nome
FROM Sessione JOIN Pilota ON Sessione.CodiceFiscale = Pilota.CodiceFiscale
JOIN Kart ON Sessione.ID_kart = Kart.ID_Kart
WHERE Sessione.ID_Sessione = 3;
```

ID_Sessione	DataSessione	OraSessione	Durata	Circuito	NoteSessione	NomeP	CognomeP	ID_Kart	Nome
3	2026-09-12	10:00:00	00:10:00	Circuito di Pomposa	Taratura sensori pedale	Giulia	Bianchi	3	Thunder

11) Consultazione di una misurazione rispetto ai valori range del segnale (2 volte al mese)

```
SELECT Misurazione.ID_Misurazione, Misurazione.NomeSegnale,
Misurazione.Valore, Segnale.ValMinScalaSegnale, Segnale.ValMaxScaleSegnale
FROM Misurazione JOIN Segnale ON Misurazione.nomeSegnale =
Segnale.NomeSegnale AND Misurazione.Idx_CAN = Segnale.ID_can
WHERE Misurazione.ID_Misurazione = 120;
```

ID_Misurazione	NomeSegnale	Valore	ValMinScalaSegnale	ValMaxScaleSegnale
120	VelocitaPosteriore	120.70	0.00	250.00

12) Consultazione del numero di sessioni per pilota (1 volta ogni 2 mesi)

```
SELECT Pilota.CodiceFiscale, Pilota.NomeP, Pilota.CognomeP,
COUNT(Sessione.ID_Sessione) AS NumeroSessioni
FROM Pilota LEFT JOIN Sessione ON Pilota.CodiceFiscale =
Sessione.codiceFiscale
GROUP BY Pilota.CodiceFiscale, Pilota.NomeP, Pilota.CognomeP;
```

CodiceFiscale	NomeP	CognomeP	NumeroSessioni
BNCGLI01H50C351P	Giulia	Bianchi	2
FRRDVD04M10F205T	Davide	Ferrari	3
RSSMRC02A15C351X	Marco	Rossi	2
VRDLCU03C22G273K	Lucia	Verdi	2

13) Consultazione dell'elenco dei sensori e le loro informazioni (1 volta al mese)

```
SELECT Sensore.Idx_Componente, Sensore.PosizioneMontaggio,
Sensore.GrandezzaMisurata, Componente.NomeC, Componente.CodiceSeriale,
Componente.Produttore, Componente.Modello
FROM Sensore JOIN Componente ON Sensore.Idx_Componente =
Componente.ID_Componente;
```

Idx_Componente	PosizioneMontaggio	GrandezzaMisurata	NomeC	CodiceSeriale	Produttore	Modello
101	Baricentro, sotto il sedile	Accelerazioni, velocita angolari, campo magnetico	IMU	K1-IMU-001	AvioRace SRL	IMU rev1.3
107	Pedale acceleratore	Posizione angolare pedale (APPS1)	APPS1	K1-APP-001	AvioRace SRL	SRH series
108	Pedale acceleratore	Posizione angolare pedale (APPS2)	APPS2	K1-APP-002	AvioRace SRL	SRH series

14) Consultazione dei componenti non elettronici (10 volte all'anno)

```
SELECT Componente.ID_Componente, Componente.CodiceSeriale,
Componente.NomeC, Componente.TipoComponente, Componente.Produttore,
Componente.Peso, Componente.Data_Installazione, Componente.Modello,
Componente.NoteC
```

```
FROM Componente LEFT JOIN ComponenteElettronico ON Componente.ID_Componente
= ComponenteElettronico.ID_componente
WHERE ComponenteElettronico.ID_componente IS NULL;
```

ID_Componente	CodiceSeriale	NomeC	TipoComponente	Produttore	Peso	Data_Installazione	Modello	NoteC
113	K1-BAT-001	Battery pack	Batteria	Autoprodotto	12.50	2025-03-20	HV Li-ion 540V	Pacco batteria alta tensione
114	K1-MOT-001	Motore 1	Motore elettrico	AMK	3.60	2025-03-20	PMSM	Motore sinistro
115	K1-MOT-002	Motore 2	Motore elettrico	AMK	3.60	2025-03-20	PMSM	Motore destro

15) Consultazione dei messaggi CAN e i componenti associati (3 volte al mese)

```
SELECT Messaggio_CAN.ID_CAN, Messaggio_CAN.DataLenghtCode,
Messaggio_CAN.Endianess, Componente.ID_Componente, Componente.NomeC,
Componente.CodiceSeriale, Comunicazione.RuoloComunicazione
FROM Messaggio_CAN JOIN Comunicazione ON Messaggio_CAN.ID_CAN =
Comunicazione.ID_can JOIN Componente ON Comunicazione.Idx_componente =
Componente.ID_Componente;
```

ID_CAN	DataLenghtCode	Endianess	ID_Componente	NomeC	CodiceSeriale	RuoloComunicazione
256	8	BIG	101	IMU	K1-IMU-001	Trasmette
256	8	BIG	201	IMU	K2-IMU-001	Trasmette
256	8	BIG	301	IMU	K3-IMU-001	Trasmette

16) Consultazione del valore massimo e medio di una misurazione per ogni sessione (in media 10 volte al mese)

```
SELECT ID_sessione, nomeSegnale, MAX(Valore) AS ValoreMassimo, AVG(Valore)
AS ValoreMedio
FROM Misurazione
WHERE nomeSegnale = 'Acceleration_X'
GROUP BY ID_sessione, nomeSegnale;
```

ID_sessione	NomeSegnale	ValoreMassimo	ValoreMedio
1	Acceleration_X	1.55	1.112500
2	Acceleration_X	2.10	1.275000
3	Acceleration_X	2.10	1.225000

17) Consultazione del tempo totale di utilizzo di ogni kart (2 volte all'anno)

```
SELECT Kart.ID_Kart, Kart.Nome, SUM(Sessione.Durata) AS TempoTotaleSecondi
FROM Kart JOIN Sessione ON Kart.ID_Kart = Sessione.ID_kart
GROUP BY Kart.ID_Kart, Kart.Nome;
```

ID_Kart	Nome	TempoTotaleSecondi
1	Falcon	9500
2	Thunder	5300
3	Thunder	2200

18) Consultazione dei componenti installati dopo una certa data (10 volte all'anno)

```
SELECT ID_Componente, CodiceSeriale, NomeC, TipoComponente, Produttore,
Peso, Data_Installazione, Modello, NoteC
FROM Componente
WHERE Data_Installazione > '2025-01-01';
```

ID_Componente	CodiceSeriale	NomeC	TipoComponente	Produttore	Peso	Data_Installazione	Modello	NoteC
101	K1-IMU-001	IMU	IMU	AvioRace SRL	0.06	2025-03-10	IMU rev1.3	Firmware 2.09, CAN 0x100-0x102, range +-12G / +-25...
102	K1-INV-001	Inverter 1	Inverter	AMK	3.20	2025-03-10	AMK-INV	Inverter motore sinistro
103	K1-INV-002	Inverter 2	Inverter	AMK	3.20	2025-03-10	AMK-INV	Inverter motore destro

19) Consultazione delle prestazioni di velocità di un pilota in sessioni di prova diverse (2 volte al mese)

```
SELECT m.ID_sessione, s.ID_kart, k.Nome, s.DataSessione, s.OraSessione,
m.TimeStamp, m.Valore AS Velocita
FROM Misurazione m JOIN Sessione s ON m.ID_sessione = s.ID_Sessione JOIN
Kart k ON s.ID_kart = k.ID_Kart
WHERE s.codiceFiscale = 'FRRDVD04M10F205T' AND m.NomeSegnale =
'VelocitaAnteriore'
ORDER BY m.ID_sessione, m.TimeStamp;
```

ID_sessione ▲ 1	ID_kart	Nome	DataSessione	OraSessione	TimeStamp ▲ 2	Velocita
4	1	Falcon	2026-09-12	14:30:00	2026-09-12 14:30:01	42.50
4	1	Falcon	2026-09-12	14:30:00	2026-09-12 14:30:03	67.80
4	1	Falcon	2026-09-12	14:30:00	2026-09-12 14:30:05	91.40
8	2	Thunder	2026-09-26	16:00:00	2026-09-26 16:00:01	45.30
8	2	Thunder	2026-09-26	16:00:00	2026-09-26 16:00:03	72.60
8	2	Thunder	2026-09-26	16:00:00	2026-09-26 16:00:05	97.80
9	1	Falcon	2026-10-03	10:00:00	2026-10-03 10:00:01	40.20
9	1	Falcon	2026-10-03	10:00:00	2026-10-03 10:00:03	65.40
9	1	Falcon	2026-10-03	10:00:00	2026-10-03 10:00:05	89.70

20) Consultazione delle analisi di velocità di un pilota in sessioni di prova diverse (2 volte al mese)

```
SELECT m.ID_sessione, s.ID_kart, AVG(m.Valore) AS VelocitaMedia,
MIN(m.Valore) AS VelocitaMinima, MAX(m.Valore) AS VelocitaMassima
FROM Misurazione m JOIN Sessione s ON m.ID_sessione = s.ID_Sessione
WHERE s.codiceFiscale = 'FRRDVD04M10F205T' AND m.NomeSegnale =
'VelocitaAnteriore' AND s.Circuito = 'Circuito di Pomposa'
GROUP BY m.ID_sessione, s.ID_kart
ORDER BY m.ID_sessione;
```

ID_sessione ▲ 1	ID_kart	VelocitaMedia	VelocitaMinima	VelocitaMassima
4	1	90.76	42.50	121.40
8	2	97.72	45.30	132.50
9	1	88.50	40.20	118.40

Di seguito vengono rappresentate graficamente alcuni sessioni di corsa di un pilota su un circuito, riportando l'andamento della velocità media, minima e massima. Il grafico permette di confrontare visivamente le prestazioni di un pilota durante le diverse sessioni.

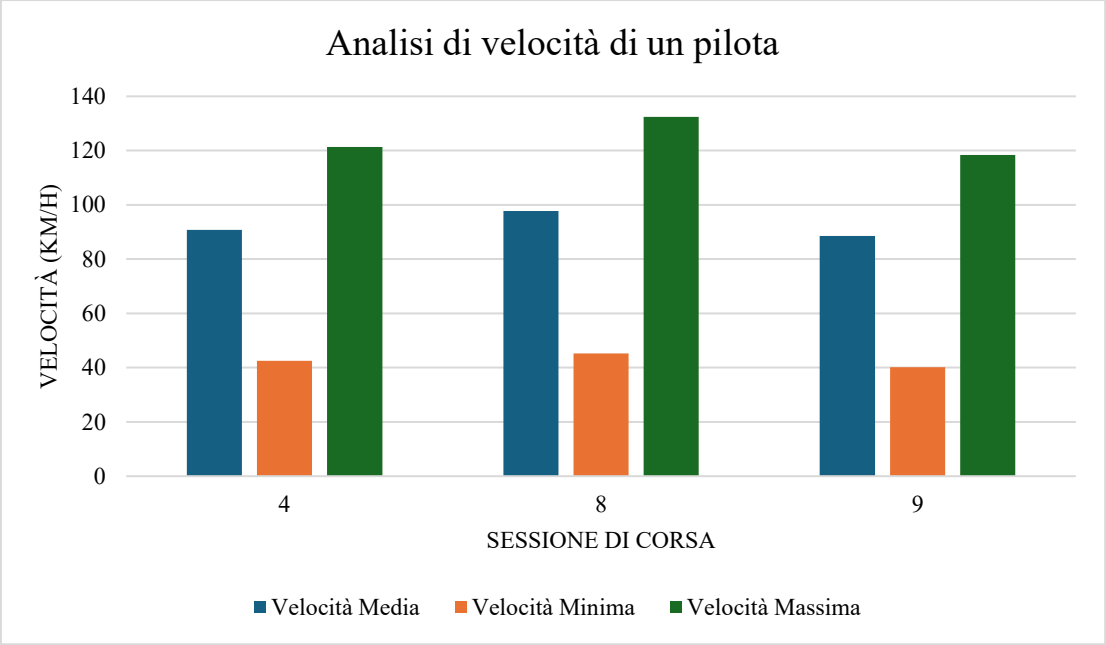


Figura 21 - Rappresentazione grafica dell'analisi di velocità di un pilota in diverse sessioni in un circuito

21) Consultazione del confronto delle velocità medie tra piloti diversi in un circuito specifico (2 volte al mese)

```
SELECT p.CodiceFiscale, p.NomeP, p.CognomeP, TIMESTAMPDIFF(SECOND,
CONCAT(s.DataSessione, ' ', s.OraSessione), m.TimeStamp) AS Secondo,
AVG(m.Valore) AS VelocitaMedia
FROM Misurazione m JOIN Sessione s ON m.ID_sessione = s.ID_Sessione JOIN
Pilota p ON s.codiceFiscale = p.CodiceFiscale
WHERE s.Circuito = 'Circuito di Pomposa' AND m.NomeSegnale =
'VelocitaAnteriore'
GROUP BY p.CodiceFiscale, p.NomeP, p.CognomeP, TIMESTAMPDIFF(SECOND,
CONCAT(s.DataSessione, ' ', s.OraSessione), m.TimeStamp)
ORDER BY Secondo, p.CognomeP;
```

CodiceFiscale	NomeP	CognomeP	Secondo ▲ 1	VelocitaMedia
BNCGLI01H50C351P	Giulia	Bianchi	1	35.40
FRRDVD04M10F205T	Davide	Ferrari	1	41.35
VRDLCU03C22G273K	Lucia	Verdi	1	39.60
BNCGLI01H50C351P	Giulia	Bianchi	3	58.70
FRRDVD04M10F205T	Davide	Ferrari	3	66.60
VRDLCU03C22G273K	Lucia	Verdi	3	63.20
BNCGLI01H50C351P	Giulia	Bianchi	5	79.80
FRRDVD04M10F205T	Davide	Ferrari	5	90.55
VRDLCU03C22G273K	Lucia	Verdi	5	86.50
BNCGLI01H50C351P	Giulia	Bianchi	7	94.60
FRRDVD04M10F205T	Davide	Ferrari	7	106.95
VRDLCU03C22G273K	Lucia	Verdi	7	101.70
BNCGLI01H50C351P	Giulia	Bianchi	9	102.30
FRRDVD04M10F205T	Davide	Ferrari	9	115.00
VRDLCU03C22G273K	Lucia	Verdi	9	110.90
BNCGLI01H50C351P	Giulia	Bianchi	11	96.80
FRRDVD04M10F205T	Davide	Ferrari	11	102.65
VRDLCU03C22G273K	Lucia	Verdi	11	104.30
BNCGLI01H50C351P	Giulia	Bianchi	13	80.50
FRRDVD04M10F205T	Davide	Ferrari	13	87.60
VRDLCU03C22G273K	Lucia	Verdi	13	89.10
BNCGLI01H50C351P	Giulia	Bianchi	15	67.20
FRRDVD04M10F205T	Davide	Ferrari	15	71.30
VRDLCU03C22G273K	Lucia	Verdi	15	73.80

Vengono rappresentati graficamente alcuni istanti misurati durate le sessioni di corsa dei piloti su un circuito, riportando l'andamento della velocità media nel tempo. Il grafico di seguito permette di confrontare visivamente le prestazioni dei diversi piloti durante le sessioni.

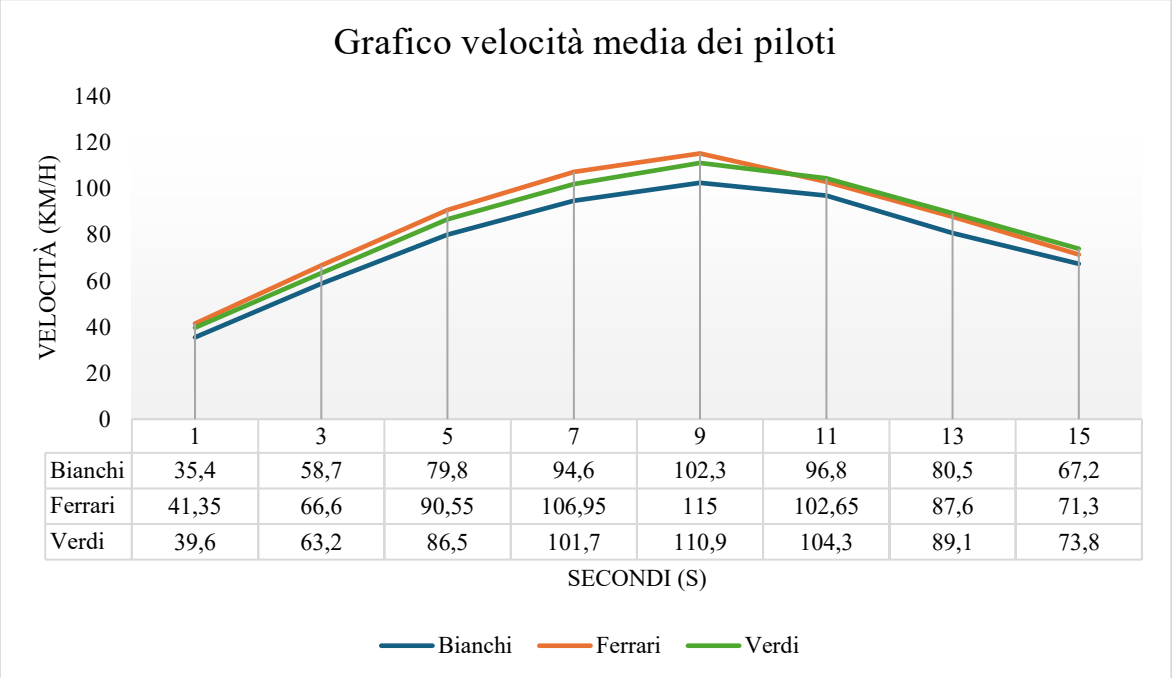


Figura 22 - Rappresentazione grafica della velocità media dei piloti in un circuito specifico

22) Consultazione della temperatura delle gomme nel tempo in una sessione di corsa (2 volte al mese)

```
SELECT TIMESTAMPDIFF(SECOND, CONCAT(s.DataSessione, ' ', s.OraSessione),
antdx.TimeStamp) AS Secondo, antdx.Valore AS ANTDX, antsx.Valore AS ANTSX,
postdx.Valore AS POSTDX, postsx.Valore AS POSTSX
FROM Sessione s JOIN Misurazione antdx ON antdx.ID_sessione = s.ID_Sessione
AND antdx.NomeSegnale = 'Tyre_Temp_FR' JOIN Misurazione antsx ON
antsx.ID_sessione = s.ID_Sessione AND antsx.TimeStamp = antdx.TimeStamp AND
antsx.NomeSegnale = 'Tyre_Temp_FL' JOIN Misurazione postdx ON
postdx.ID_sessione = s.ID_Sessione AND postdx.TimeStamp = antdx.TimeStamp
AND postdx.NomeSegnale = 'Tyre_Temp_RR' JOIN Misurazione postsx ON
postsx.ID_sessione = s.ID_Sessione AND postsx.TimeStamp = antdx.TimeStamp
AND postsx.NomeSegnale = 'Tyre_Temp_RL'
WHERE s.ID_Sessione = 4
ORDER BY Secondo;
```

Secondo ▲ 1	ANTDX	ANTSX	POSTDX	POSTSX
0	28.50	27.80	30.10	29.40
10	36.20	35.10	38.70	37.50
20	44.80	43.20	47.60	46.10
30	53.10	51.40	56.20	54.30
40	59.20	57.60	61.30	58.90
50	64.10	61.80	66.20	63.50
60	68.40	66.10	70.50	67.80

Vengono rappresentati graficamente alcuni istanti misurati durate una sessioni di corsa di un pilota, riportando l'andamento delle temperature delle gomme in gradi. Il grafico di seguito permette di confrontare visivamente le temperature misurate dai sensori nelle gomme.

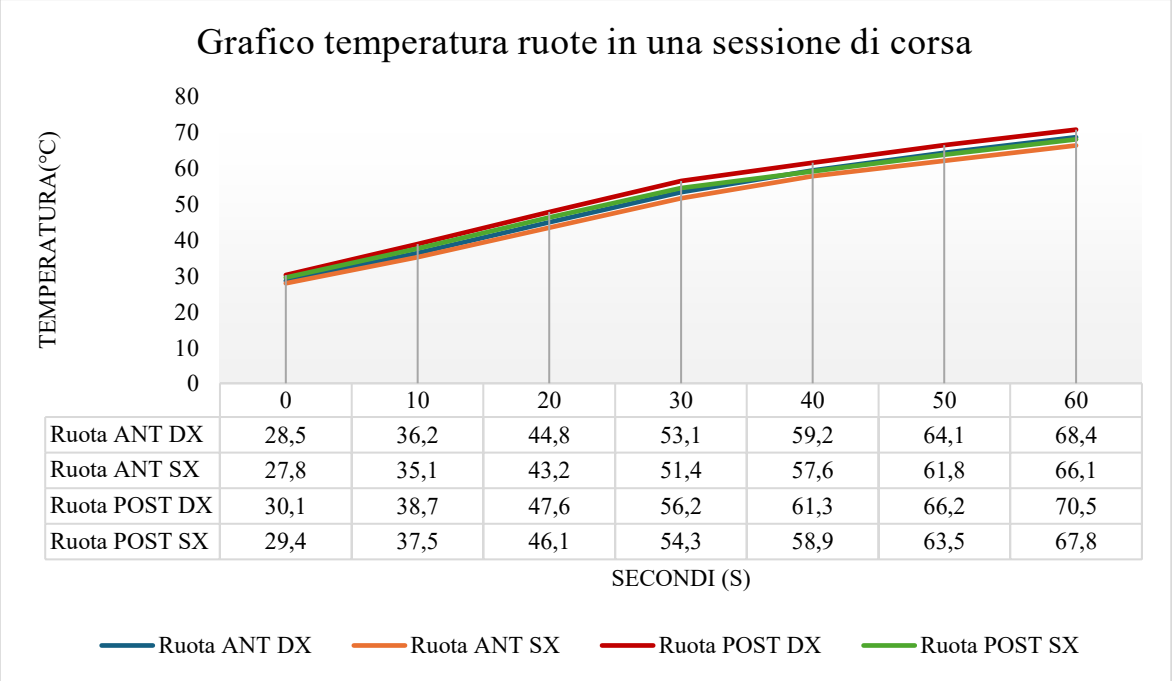


Figura 23 - Rappresentazione grafica della temperatura delle ruote in una sessione di corsa

23) Consultazione del confronto della temperatura media, massima e minima di ciascuna ruota in una sessione di corsa (2 volte al mese)

```
SELECT nomeSegnale AS Ruota, AVG(Valore) AS TemperaturaMedia, MAX(Valore)
AS TemperaturaMassima, MIN(Valore) AS TemperaturaMinima
FROM Misurazione
WHERE ID_sessione = 4 AND nomeSegnale IN ('Tyre_Temp_FR', 'Tyre_Temp_FL',
'Tyre_Temp_RR', 'Tyre_Temp_RL')
GROUP BY nomeSegnale;
```

Ruota	TemperaturaMedia	TemperaturaMassima	TemperaturaMinima
Tyre_Temp_FL	49.00	66.10	27.80
Tyre_Temp_FR	50.61	68.40	28.50
Tyre_Temp_RL	51.07	67.80	29.40
Tyre_Temp_RR	52.94	70.50	30.10

24) Modifica delle informazioni di un componente (1 volta al mese)

```
UPDATE Componente SET NomeC = '...', TipoComponente = '...', Produttore =
'...', Peso = ..., Data_Installazione = '...', Modello = '...', NoteC =
'...'
WHERE ID_Componente = ...;
```

25) Modifica delle informazioni di un kart (1 volta ogni 6 mesi)

```
UPDATE Kart SET Nome = '...', Modello = '...', Categoria = '...', Stato =
'...', Anno_Omologazione = ..., NoteK = '...'
WHERE ID_Kart = ...;
```

26) Modifica delle informazioni di una sessione (1 volta al mese)

```
UPDATE Sessione SET ID_kart = ..., CodiceFiscale = '...', DataSessione =
'...', OraSessione = '...', Durata = '...', Circuito = '...', NoteSessione
= '...'
WHERE ID_Sessione = ...;
```

27) Cancellazione delle misure errate di una sessione (2 volte al mese)

```
DELETE FROM Misurazione
WHERE ID_sessione = ... AND ID_Misurazione IN (SELECT m.ID_Misurazione
                                               FROM Misurazione m JOIN Segnale s
                                               ON m.NomeSegnale = s.NomeProdotto
                                               AND m.Idx_CAN = s.ID_can
                                               WHERE m.ID_sessione = ... AND
                                               (m.Valore < s.ValMinScalaSegnale
```

```
                                        OR m.Valore >
                                        s.ValMaxScaleSegnale));
```

28) Cancellazione di un componente (5 volte all'anno)

```
DELETE FROM Componente
WHERE ID_Componente = ...;
```

29) Cancellazione di un pilota (1 volta ogni 5 anni)

```
DELETE FROM Pilota
WHERE CodiceFiscale = '...';
```

5 Conclusioni e sviluppi futuri

Il progetto ha descritto i passaggi effettuati per la realizzazione di una base di dati relazionale, un sistema organizzato di tabelle che permette di memorizzare in maniera ordinata le principali informazioni quotidiane appartenenti al team universitario Polimarche Racing Team.

Il sistema permette, grazie alla scelte progettuali effettuate, di storicizzare ed analizzare i propri dati tramite interrogazioni in linguaggio SQL, così da essere di supporto alle decisioni future dei membri del team sulle attività di corsa.

Il lavoro vuole evidenziare come la progettazione di una corretta struttura di archiviazione, cioè la base di dati, comporta una maggiore facilità di gestione e reperibilità delle enormi quantità di informazioni che oggi giorno vengono generate costantemente.

La base di dati realizzata costituisce inoltre una possibile struttura di partenza per futuri ampliamenti, come l'integrazione di ulteriori parametri di telemetria, l'inserimento di altre tipologie di personale e componenti, l'incremento delle misurazioni raccolte durante le sessioni, l'espansione delle caratteristiche dei sensori e dei messaggi del protocollo CAN o, come ultimo, lo sviluppo di strumenti più avanzati di analisi per il confronto delle prestazioni dei piloti e dei kart.

6 Bibliografia e sitografia

La tesi in questione è di tipo prettamente progettuale, ciò significa che la realizzazione è a carico del progettista e non comporta così una vera e propria lista di libri e siti consultati. Nonostante ciò, le fasi di realizzazione sono fondate sulle tecniche standard di creazione di una base di dati, possono perciò essere caratterizzate da scelte che prendono spunto dalle fonti sotto citate.

Ullman J. D., *Basi di dati e basi di conoscenza*, Gruppo Editoriale Jackson, Milano, 1991.
Silberschatz A., Korth H. F., Sudarshan S., *Database System Concepts*, McGraw-Hill, New York, 2020.
Batini C., Pernici B., Santucci G. (a cura di), *Sistemi Informativi*, FrancoAngeli, Milano, 2005.

https://learn.univpm.it/

https://www.polimarcheracingteam.com/it/

Ringraziamenti

Giunto al termine di questo percorso accademico, desidero dedicare qualche riga a tutte le persone che mi sono state accanto e che, in modi diversi, hanno contribuito al raggiungimento di un importante obiettivo.

Un ringraziamento speciale va alla Prof.ssa Claudia Diamantini, per la disponibilità, i consigli e il supporto che mi ha dedicato durante la realizzazione di questo elaborato. Desidero ringraziarla non soltanto per avermi guidato in questo percorso di tesi, ma anche perché, attraverso le sue lezioni di Basi di Dati, è riuscita a trasmettermi una passione autentica per questa materia, facendomi scoprire un interesse che è andato ben oltre il semplice studio per un esame. Le sue lezioni hanno contribuito a orientare il mio percorso e ad alimentare la curiosità che mi ha accompagnato anche durante la realizzazione del progetto. Per questo, le sono sinceramente grato.

Ringrazio dal profondo del cuore i miei genitori e tutta la mia famiglia, per essermi sempre stata accanto, per aver creduto in me anche nei momenti più difficili, in cui forse non ci credevo nemmeno io, e per avermi sostenuto nelle scelte e nelle sfide che ho affrontato, tra ansie e momenti di sconforto. Il loro sostegno è stato fondamentale per arrivare fin qui e sono certo che continuerà a esserlo anche nelle sfide future.

Un grazie sincero va anche ai miei più cari amici e tutte le persone che hanno condiviso con me questo percorso, rendendolo più leggero, significativo e ricco di momenti che ricorderò per sempre.

Infine, ringrazio me stesso per la determinazione e la dedizione con cui ho affrontato le difficoltà incontrate lungo il cammino, dalle notti insonni alle mattine degli esami in cui tutto appariva più difficile e raggiungere questo traguardo sembrava quasi impossibile. Invece ci sono riuscito, e la fine di questo percorso di studi rappresenta anche il risultato di una mia crescita personale.