Immagine che contiene testo, dispositivo, metro, pannello di controllo

Descrizione generata automaticamente

Abstract

La Formula 1 (F1), è la massima categoria di vetture monoposto a ruote scoperte da corsa su circuito definita dalla Federazione Internazionale dell'Automobile (FIA).

Si è sviluppato una base di dati contenente uno storico di tutti i piloti, auto, circuiti e i risultati delle varie stagioni della competizione.

Questi dati servono per la consultazione all'interno del sito ufficiale nel quale i tifosi potranno vedere i risultati della stagione in corso e di quelle passate.

Ogni stagione è composta da un numero deciso dalla FIA(Federazione Internazionale Automobilismo) di gare tra un insieme di piste sparse in tutto il mondo, ogni nazione può contenere più di una pista o nessuna, un weekend di gara è composto da qualifiche e gara.

I piloti vengono assunti dai vari team a inizio stagione ogni scuderia ha 2 piloti “guidanti” e un numero variabile di piloti di riserva (non presenti però nel database).

Ogni weekend di gara si suddivide in più sessioni: 3 sessioni di qualifiche nei quali i piloti cercano di segnare il giro più veloce in modo da garantirsi una posizione di partenza favorevole, infine vi è la gara, alla fine della gara si stila una graduatoria di arrivo e vengono assegnati dei punti di conseguenza, inoltre si registrano il giro più veloce e il pilota che secondo i tifosi è stato il più meritevole

Ogni anno il pilota che conclude le gare del calendario con più punti si aggiudica il titolo di campione del mondo piloti

Ogni anno il team che conclude le gare del calendario con più punti si aggiudica il titolo di campione del campionato mondiale dei costruttori

Ogni auto è di proprietà di una squadra, monta un motore costruito da una azienda e ha una cilindrata e tipologia decisa da regolamento, inoltre ogni auto monta pneumatici di una compagnia produttrice, nei vari anni si è passato dalla libera partecipazione dei fornitori al monopolio Pirelli.

Nella formula 1 non vi sono vincoli concernenti il sesso del pilota anche se la stragrande maggioranza risulta essere di sesso maschile.

Il numero di squadre è variabile negli anni e l’iscrizione è libera.

Analisi dei requisiti

Il progetto per ovvi motivi logistici non prenderà in considerazione i risultati di tutti i gran premi dal 1950, bensì dal 2020

Le gare si disputano su circuiti, nel database per cui sono presenti dati riguardo:

* Nome
* Alias
* Nazione
* Lunghezza

Le gare a loro volta sono caratterizzate da:

* Anno
* Gara num
* Nome gara
* Pilota del giorno
* Pilota veloce
* Hot lap

Le gare vengono corse da piloti dei quali si conoscono questi dati:

* Codice fiscale
* Nome
* Cognome
* Sesso
* Nazionalità
* Anno nascita
* Anno morte
* Sigla in gara

Ogni Pilota ogni anno guida una vettura e gli viene assegnato un numero a partire dalla stagione del 2014, per cui dei partecipanti alle stagioni sappiamo:

* Codice fiscale
* vettura
* Numero in gara

Un partecipante in un weekend di gara realizza una prestazione che consta di:

* Tempo q1,q2 e q3
* Posizione arrivo
* ritiro

Ogni autovettura è caratterizzata da:

* Nome
* Squadra
* motore
* pneumatico
* Anno

Di ogni squadra si sa:

* Nome
* Nazione
* Nome precedente

Degli Pneumatici si conosce:

* Nome
* Nazione

Infine, del motore sono presenti i dati di:

* Nome
* Nazione
* Cilindrata
* Tipologia di motore

Glossario

|  |  |  |
| --- | --- | --- |
| Termine | Descrizione | Collegamenti (attributo di…) |
| Alias | nome alternativo con cui è conosciuto il circuito | circuito |
| Lunghezza | Lunghezza in Km del tracciato | circuito |
| Gara num | Numero della gara nel calendario della stagione corrente | gara |
| Pilota del giorno | premio assegnato dal pubblico al pilota che si è distinto maggiormente nella gara | gara |
| Pilota veloce | premio assegnato dalla FIA al pilota che ha segnato il giro più veloce in gara | gara |
| Hot lap | tempo del giro veloce | gara |
| Anno morte | se sono morti | pilota |
| Sigla in gara | 3 lettere che vengono assegnate ad ogni pilota per motivo di grafica televisiva | pilota |
| Numero in gara | se presente è un numero che si affianca alla sigla in gara | partecipante |
| Tempo q1,q2 e q3 | tempi per le qualificazioni | prestazione |
| ritiro | Flag che segnala se il pilota abbia concluso la gara o se si è ritirato | prestazione |
| anno | Un’auto partecipa solo ad una stagione della competizione | autovettura |
| Nome precedente | molte squadre vengono acquistate da nuovi team e cambiate di nome | squadra |

Operazioni

|  |  |  |
| --- | --- | --- |
| OPERAZIONE | TIPO | FREQUENZA |
| Aggiunta risultato weekend di gara | Scrittura | 1 ogni 1-2 settimane |
| Aggiunta di un Pilota | Scrittura | 1-2 all’anno circa |
| Aggiunta vetture con componenti | Scrittura | 20 all’anno |
| Lettura risultati di gara | Lettura | 250000+ a settimana |
| Lettura classifica | Lettura | 250000+ a settimana |

Progettazione Concettuale

Lista entità

* Circuito
  + Nome varchar(60) PRIMARY KEY
  + Alias varchar(60) NOT NULL
  + Nazione varchar(2) NOT NULL
  + Lunghezza float NOT NULL
* Squadra
  + Nome varchar(40) PRIMARY KEY
  + Nazione varchar(2) NOT NULL
  + Nome\_precedente varchar(40)
* Pilota
  + Codice\_fiscale varchar(20) PRIMARY KEY
  + Nome varchar(20) NOT NULL
  + Cognome varchar(20) NOT NULL
  + Nazionalità varchar(2) NOT NULL
  + Anno\_nascita int NOT NULL
  + Anno\_morte int
  + Sigla\_in\_gara varchar(3) NOT NULL

L’entità Pilota si specializza in due sottocategorie con generalizzazione totale in base al sesso:

* + - Uomo
    - Donna
* Motore
  + Nome varchar(40) PRIMARY KEY
  + Nazione varchar(2) NOT NULL
  + Tipologia Vvarchar(20) NOT NULL

L’entità Motore si specializza in tre sottocategorie con generalizzazione totale in base alla cilindrata:

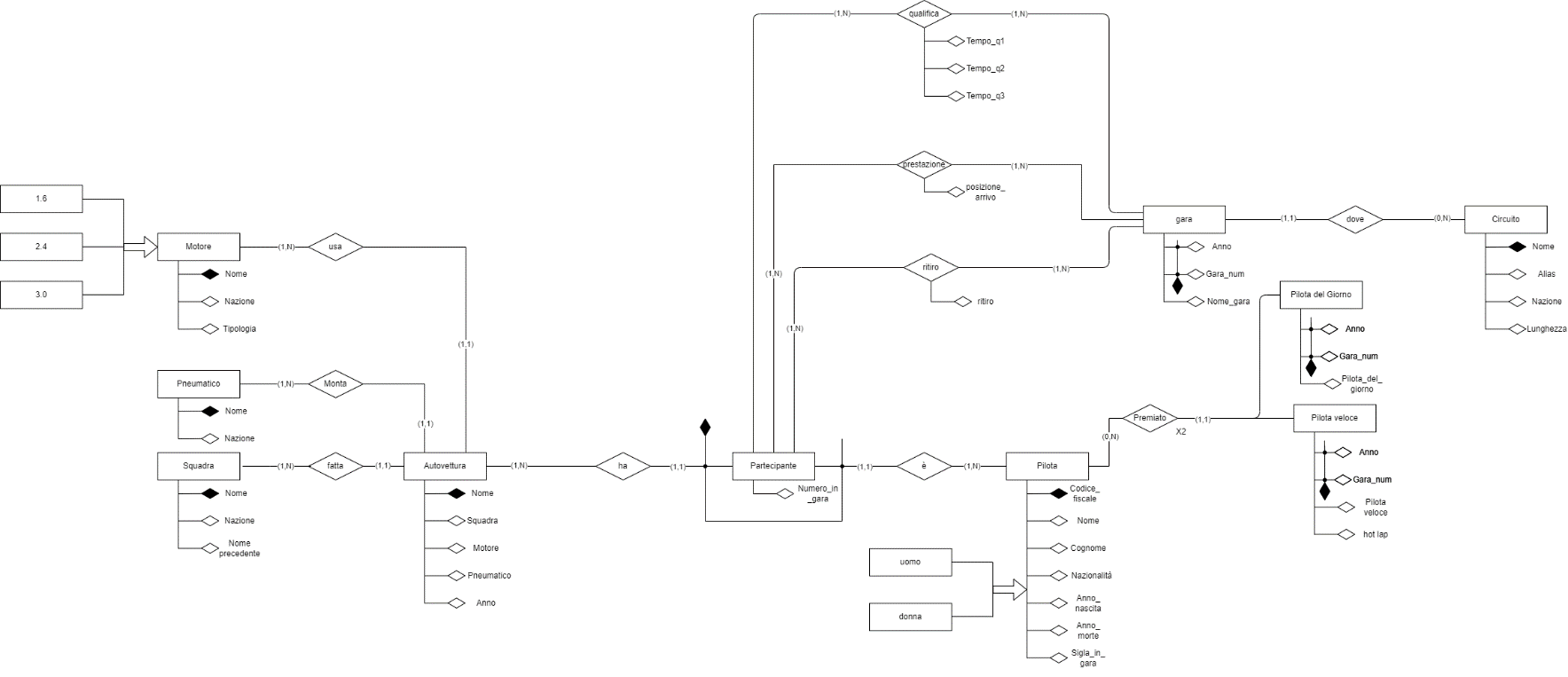
* + - 1.6
    - 2.4
    - 3.0
* Pneumatico
  + Nome varchar(20) PRIMARY KEY
  + Nazione varchar(2) NOT NULL
* Autovettura
  + Nome varchar(60) PRIMARY KEY
  + Squadra varchar(40) NOT NULL
  + Anno int NOT NULL
  + Motore varchar(40) NOT NULL
  + Pneumatico varchar(20) NOT NULL
* Partecipante
  + Codice\_fiscale varchar(20) PRIMARY KEY
  + Vettura varchar(60) PRIMARY KEY
  + Numero\_in\_gara int
* Gara
  + Anno int PRIMARY KEY
  + Gara\_num PRIMARY KEY
  + Nome\_gara varchar(60) NOT NULL
* Pilota del giorno
  + Anno int PRIMARY KEY
  + Gara\_num int PRIMARY KEY
  + Pilota\_del\_giorno varchar(20) NOT NULL
* Pilota veloce
  + Anno int PRIMARY KEY
  + Gara\_num int PRIMARY KEY
  + Pilota\_veloce varchar(20) NOT NULL
  + Hot lap time NOT NULL
* Prestazione (PRIMARY KEY sottesa)
  + Posizione\_arrivo int NOT NULL
* Ritiro (PRIMARY KEY sottesa)
  + Ritiro varchar(1) NOT NULL
* Qualifica (PRIMARY KEY sottesa)
  + Tempo q1 time
  + Tempo q2 time
  + Tempo q3 time

Tabella delle Relazioni

|  |  |  |  |
| --- | --- | --- | --- |
| Relazioni | Entità coinvolte | Descrizione | Attributi |
| Usa | Motore (1,N)  Autovettura (1,1) | Un motore può essere montato da più autovetture anche di squadre diverse ma una vettura monta un solo motore | N/A |
| Monta | Pneumatico (1,N)  Autovettura (1,1) | Una marca di pneumatici può fornire più autovetture ma una autovettura monta pneumatici di un solo brand | N/A |
| Fatta | Squadra (1,N)  Autovettura (1,1) | Una squadra realizza più vetture ma una vettura è realizzata da un solo team | N/A |
| Ha | Autovettura (1,N)  Partecipante (1,1) | Ogni autovettura può essere guidata da più piloti ( di solito 2) mentre un pilota di una determinata stagione guida solo una vettura | N/A |
| È | Partecipante (1,1)  Pilota (1,N) | Un partecipante(ad una stagione) è un pilota ma un pilota può partecipare più volte | N/A |
| Dove | Gara (1,1)  Circuito (0,N) | Una gara si disputa su un solo circuito ma un circuito può ospitare nessuna o più gare | N/A |
| Premiato | Pilota (0,N)  Pilota del giorno (1,1)  Pilota (0,N)  Pilota veloce (1,1) | Ogni pilota può essere premiato mai o più volte sia come pilota del giorno che come pilota veloce, ma il pilota del giorno e il pilota veloce è uno solo a gara | N/A |

**Vincolo non rappresentabili tramite grafico E-R**

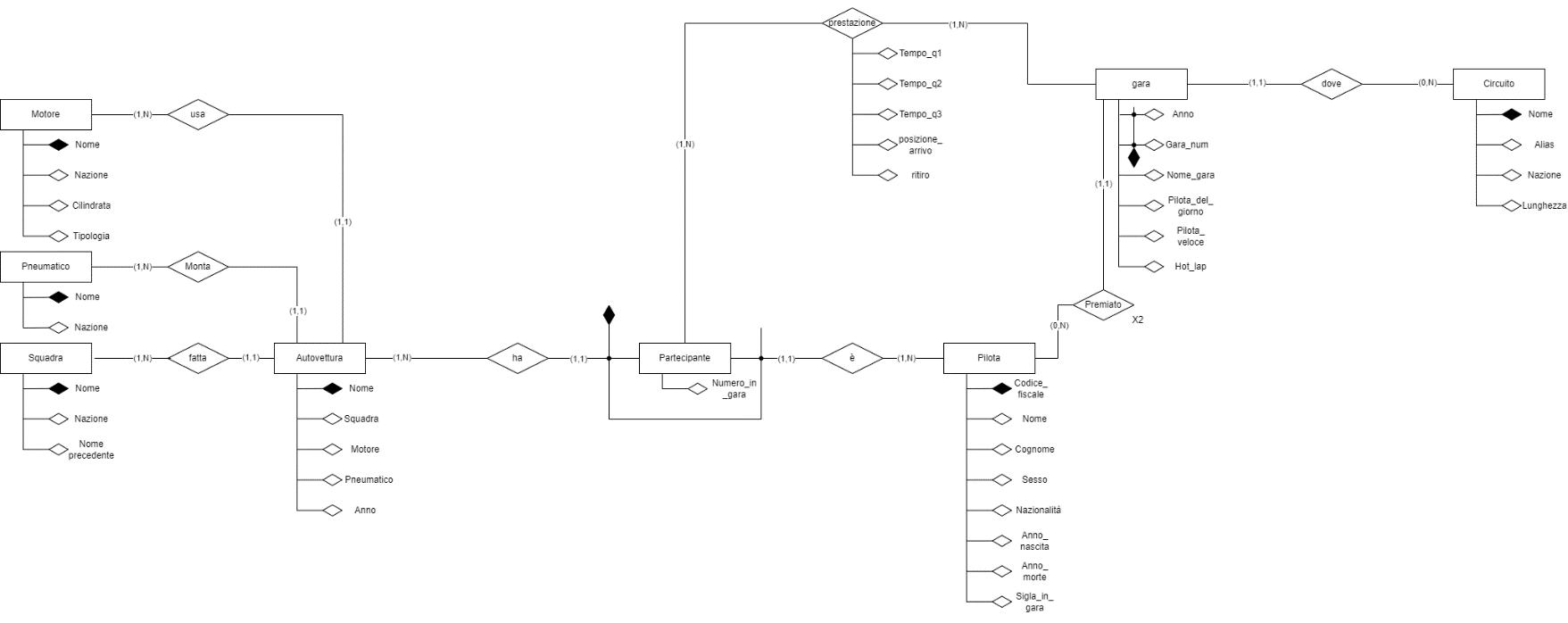
Per ottenere i premi di pilota veloce e di pilota del giorno il pilota deve essere un partecipante

Schema concettuale

Progettazione Logica - Ristrutturazione

Eliminazione delle generalizzazioni

|  |  |
| --- | --- |
| Generalizzazione | Risoluzione |
| Pilota 🡨 uomo/donna | Piloti donna nonostante più rari rispetto agli uomini detengono le medesime proprietà per cui vengono accorpati in un'unica tabella e viene aggiunto un tag identificativo |
| Gara 🡨Pilota\_del\_giorno e( Pilota\_veloce Hot\_lap) | Entrambi questi dati condividono la medesima chiave primaria di gara per cui vengono accorpati |
| Prestazione 🡨 qualifiche,ritiro | Condividono l’intento del descrivere il weekend di gara per cui vengono accorpate al risultato della gara |
| Motore <- (cilindrata) | Può essere condensato come caratteristica del motore direttamente in esso |

Schema E-R ristrutturato

Indice

Questo database presente nel sito ufficiale della F1 viene usato in grandissima misura soprattutto in lettura dai milioni di fan dello sport, l’informazione più importante è il pilota che in una determinata gara ottiene un certo piazzamento per cui si indicizza così:

create index "posizioni\_arrivo" on prestazione(anno,gara\_num,codice\_fiscale,posizione\_arrivo);

Creazione Tabelle

Circuito(nome,alias,nazione,lunghezza)

Squadra(nome,nazione,nome\_precedente)

Pilota(codice\_fiscale,nome,cognome,sesso,nazionalità,anno\_nascita,anno\_morte,sigla\_in\_gara)

Motore(nome,nazione,cilindrata,tipologia)

Pneumatico(nome,nazione)

Autovettura(nome,squadra,anno,motore,pneumatico)

-autovettura.squadra 🡨 squadra.nome

-autovettura.motore 🡨 motore.nome

-autovettura.pneumatico 🡨 pneumatico.nome

Partecipante(codice\_fiscale,vettura,numero\_in\_gara)

-Partecipante.codice\_fiscale 🡨 pilota.codice\_fiscale

-partecipante.vettura 🡨 autovettura.nome

Gara(anno,gara\_num,nome\_gara,pilota\_del\_giorno,pilota\_veloce,hot\_lap)

-gara.pilota\_del\_giorno 🡨 pilota.codice\_fiscale

-gara.pilota\_veloce 🡨 pilota.codice\_fiscale

Gara.nome\_gara 🡨 circuito.nome

Prestazione(anno,gara\_num,codice\_fiscale,vettura,tempo\_q1,tempo\_q2,tempo\_q3,posizione\_arrivo,ritiro)

-(prestazione.anno,prestazione.gara\_num) 🡨 (gara.anno,gara.gara\_num)

-(prestazione.vettura,prestazione.codice\_fiscale)🡨(partecipante.vettura,partecipante.codice\_fiscale)

Query

Tutte le query presenti sono interattive e richiedono all’utente l’input di dati per ottenere i risultati richiesti, nel codice c++ sono riportati con i placeholder $x::tipo nel quale x rappresenta il numero di parametro mentre il tipo, la tipologia di dato da rimpiazzare. Questi placeholder poi vengono rimpiazzati dal codice con i dati immessi

Query 1:

Hot\_lap: visualizza chi ha realizzato il miglior giro in assoluto in una determinata pista, in quale anno e con quale vettura

void hotlap(PGconn\* conn)

{

    string pista;

    cout<<"Inserire il nome della pista tra quelle presenti: ";

    string query = "SELECT circuito.nome FROM circuito;";

    PGresult\* res = PQexec(conn,query.c\_str());

    checkResults(res,conn);

    stampatuple(res);

    cout<<"nome selezionato: ";

    cin>>pista;

    query = "

SELECT circuito.nome,gara.anno,gara.hot\_lap,pilota.nome,pilota.cognome,prestazione.vettura

FROM circuito,gara,pilota,prestazione

WHERE circuito.nome = $1 AND

gara.nome\_gara = circuito.nome AND

gara.pilota\_veloce = pilota.codice\_fiscale AND

gara.hot\_lap = (SELECT MIN(gara.hot\_lap) FROM gara WHERE gara.nome\_gara = $1) AND

gara.pilota\_veloce = prestazione.codice\_fiscale AND gara.anno = prestazione.anno AND

gara.gara\_num = prestazione.gara\_num;";

    const char\* parametro1 = pista.c\_str();

    res = PQprepare(conn,"hot\_lap",query.c\_str(),1,NULL);

    res = PQexecPrepared(conn,"hot\_lap",1,&parametro1,NULL,0,0);

    checkResults(res,conn);

    stampatuple(res);

}

OUTPUT:

Immagine che contiene testo, elettronico, screenshot, schermo

Descrizione generata automaticamente

…

Immagine che contiene testo

Descrizione generata automaticamente

Query 2:

Classifica: visualizza la classifica piloti o costruttori di un anno selezionato

void classifica(PGconn\* conn)

{

    int anno;

    int pil\_cos;

    cout<<"inserire l'anno di cui si vuole visualizzare la classifica: ";

    cin>>anno;

    string supp = to\_string(anno);

    const char\* parametro1 = supp.c\_str();

    cout<<"0 per classifica piloti, 1 per classifica costruttori: ";

    cin>>pil\_cos;

    string query\_costruttori = "

SELECT squadra.nome, SUM(punti\_gara.punti) as punti\_totali

FROM(SELECT prestazione.codice\_fiscale,prestazione.posizione\_arrivo,

CASE

WHEN prestazione.posizione\_arrivo = 1 THEN 25

WHEN prestazione.posizione\_arrivo = 2 THEN 18

WHEN prestazione.posizione\_arrivo = 3 THEN 15

WHEN prestazione.posizione\_arrivo = 4 THEN 12

WHEN prestazione.posizione\_arrivo = 5 THEN 10

WHEN prestazione.posizione\_arrivo = 6 THEN 8

WHEN prestazione.posizione\_arrivo = 7 THEN 6

WHEN prestazione.posizione\_arrivo = 8 THEN 4

WHEN prestazione.posizione\_arrivo = 9 THEN 2

WHEN prestazione.posizione\_arrivo = 10 THEN 1

ELSE 0 END punti

FROM prestazione

WHERE prestazione.anno = $1::int

ORDER BY prestazione.gara\_num,prestazione.posizione\_arrivo)AS punti\_gara,

pilota,partecipante,autovettura,squadra

WHERE punti\_gara.codice\_fiscale = partecipante.codice\_fiscale AND

partecipante.vettura = autovettura.nome AND

autovettura.anno = $1::int AND

autovettura.squadra = squadra.nome AND

partecipante.codice\_fiscale = pilota.codice\_fiscale GROUP BY squadra.nome ORDER BY punti\_totali DESC;";

    string query\_piloti      = "

SELECT pilota.nome,pilota.cognome, SUM(punti\_gara.punti) as punti\_totali

FROM(SELECT prestazione.codice\_fiscale,prestazione.posizione\_arrivo,

CASE WHEN prestazione.posizione\_arrivo = 1 THEN 25

WHEN prestazione.posizione\_arrivo = 2 THEN 18

WHEN prestazione.posizione\_arrivo = 3 THEN 15

WHEN prestazione.posizione\_arrivo = 4 THEN 12

WHEN prestazione.posizione\_arrivo = 5 THEN 10

WHEN prestazione.posizione\_arrivo = 6 THEN 8

WHEN prestazione.posizione\_arrivo = 7 THEN 6

WHEN prestazione.posizione\_arrivo = 8 THEN 4

WHEN prestazione.posizione\_arrivo = 9 THEN 2

WHEN prestazione.posizione\_arrivo = 10 THEN 1

ELSE 0 END punti

FROM prestazione

WHERE prestazione.anno = $1::int

ORDER BY prestazione.gara\_num,prestazione.posizione\_arrivo)AS punti\_gara,  
pilota,partecipante,autovettura

WHERE punti\_gara.codice\_fiscale = partecipante.codice\_fiscale AND

partecipante.vettura = autovettura.nome AND

autovettura.anno = $1::int AND

partecipante.codice\_fiscale = pilota.codice\_fiscale GROUP BY pilota.codice\_fiscale ORDER BY punti\_totali DESC;";

    PGresult\* res;

    if(pil\_cos == 1)

    {

        res = PQprepare(conn,"classifica\_costruttori",query\_costruttori.c\_str(),1,NULL);

        res = PQexecPrepared(conn,"classifica\_costruttori",1,&parametro1,NULL,0,0);

    }

    else

    {

        res = PQprepare(conn,"classifica\_piloti",query\_piloti.c\_str(),1,NULL);

        res = PQexecPrepared(conn,"classifica\_piloti",1,&parametro1,NULL,0,0);

    }

    checkResults(res,conn);

    stampatuple(res);

}

OUTPUT

Immagine che contiene testo, screenshot, elettronico, computer

Descrizione generata automaticamente

Query 3:

DNF: visualizza quanti piloti hanno subito un numero di DNF a scelta in una stagione a scelta

{

    int anno;

    int n\_ritiri;

    cout<<"inserire l'anno interessato: ";

    cin>>anno;

    string supp1 = to\_string(anno);

    const char\* parametro1 = supp1.c\_str();

    cout<<"inserire il numero minimo di ritiri: ";

    cin>>n\_ritiri;

    string supp2 = to\_string(n\_ritiri);

    const char\* parametro2 = supp2.c\_str();

    string query = "

SELECT pilota.nome,pilota.cognome, COUNT(\*) AS ritiri\_stagionali

FROM prestazione,partecipante,pilota

WHERE prestazione.anno = $1::int AND

prestazione.ritiro = 'Y' AND

prestazione.codice\_fiscale = partecipante.codice\_fiscale AND

prestazione.vettura = partecipante.vettura AND

partecipante.codice\_fiscale = pilota.codice\_fiscale GROUP BY pilota.codice\_fiscale HAVING COUNT(\*) >= $2::int ORDER BY ritiri\_stagionali DESC;";

    PGresult\* res = PQprepare(conn,"DNF",query.c\_str(),2,NULL);

    const char\* parametri[2] = {parametro1,parametro2};

    res = PQexecPrepared(conn,"DNF",2,parametri,NULL,0,0);

    checkResults(res,conn);

    stampatuple(res);

}

OUTPUT

Immagine che contiene testo

Descrizione generata automaticamente

Query 4:

Info: visualizza le informazioni riguardo ai veicoli(e ai loro dettagli) di ogni squadra

void info(PGconn\*conn)

{

    cout<<"Selezionare un team da quelli sottostanti\n";

    PGresult\* res = PQexec(conn,"SELECT squadra.nome FROM squadra");

    checkResults(res,conn);

    stampatuple(res);

    string squadra;

    cout<<"inserire il nome: ";

    cin>>squadra;

    const char\* parametro1 = squadra.c\_str();

    res = PQprepare(conn,"seleziona\_anno","

SELECT autovettura.anno AS anni\_selezionabili

FROM squadra,autovettura

WHERE squadra.nome = autovettura.squadra AND

squadra.nome = $1::varchar;",1,NULL);

    res = PQexecPrepared(conn,"seleziona\_anno",1,&parametro1,NULL,0,0);

    checkResults(res,conn);

    stampatuple(res);

    int anno;

    cout<<"inserire l'anno: ";

    cin>>anno;

    string supp = to\_string(anno);

    const char\* parametro2 = supp.c\_str();

    string query = "

SELECT autovettura.nome,autovettura.squadra,autovettura.motore,autovettura.pneumatico,

pilota.nome,pilota.cognome

FROM autovettura,partecipante,squadra,pilota

WHERE squadra.nome = $1::varchar AND

autovettura.squadra = squadra.nome AND

autovettura.anno = $2::int AND

partecipante.vettura = autovettura.nome AND

partecipante.codice\_fiscale = pilota.codice\_fiscale;";

    res = PQprepare(conn,"info1",query.c\_str(),2,NULL);

    const char\* parametri[2] = {parametro1,parametro2};

    res = PQexecPrepared(conn,"info1",2,parametri,NULL,0,0);

    checkResults(res,conn);

    stampatuple(res);

    cout<<"Selezionare il campo del quale si vogliono maggiori informazioni:\n";

    cout<<"[1] info squadra\n"<<"[2] info motore\n"<<"[3] info pneumatico\n"<<"[0]chiudi query\n";

    cout<<"inserire il valore desiderato: ";

    int menu;

    bool input\_corretto = false;

    do{

        cin>>menu;

        if(menu<0||menu>3)

        {

            cout<<"inserire un input corretto tra 0 e 3: ";

            input\_corretto = false;

        }

        else

            input\_corretto = true;

    }while(!input\_corretto);

    switch (menu)

    {

    case 1:

        query = "SELECT squadra.\* FROM squadra WHERE squadra.nome = $1::varchar;";

        res = PQprepare(conn,"info\_squadra",query.c\_str(),1,NULL);

        res = PQexecPrepared(conn,"info\_squadra",1,&parametro1,NULL,0,0);

        checkResults(res,conn);

        stampatuple(res);

        int select;

        cout<<endl<<"premere 1 se si vuole visualizzare tutti i nomi precedenti del team se presenti :";

        cin>>select;

        if(select == 1)

        {

            query = "

WITH RECURSIVE nomi AS (SELECT s1.nome, s1.nome\_precedente

FROM squadra s1

WHERE s1.nome = $1::varchar

UNION ALL

SELECT s2.nome, s2.nome\_precedente

FROM nomi JOIN squadra s2 ON nomi.nome\_precedente = s2.nome)

SELECT nomi.nome

FROM nomi;";

            res = PQprepare(conn,"info\_ricorsiva",query.c\_str(),1,NULL);

            res = PQexecPrepared(conn,"info\_ricorsiva",1,&parametro1,NULL,0,0);

            checkResults(res,conn);

            stampatuple(res);

        }

        break;

    case 2:

        query = "

SELECT motore.\*

FROM motore,squadra,autovettura

WHERE squadra.nome = $1::varchar AND

autovettura.squadra = squadra.nome AND

autovettura.anno = $2::int AND

autovettura.motore = motore.nome;";

        res = PQprepare(conn,"info\_motore",query.c\_str(),2,NULL);

        res = PQexecPrepared(conn,"info\_motore",2,parametri,NULL,0,0);

        checkResults(res,conn);

        stampatuple(res);

        break;

    case 3:

        query = "

SELECT pneumatico.\*

FROM pneumatico,autovettura,squadra

WHERE squadra.nome = $1::varchar AND

autovettura.squadra = squadra.nome AND

autovettura.anno = $2::int AND

autovettura.pneumatico = pneumatico.nome;";

        res = PQprepare(conn,"info\_pneumatico",query.c\_str(),2,NULL);

        res = PQexecPrepared(conn,"info\_pneumatico",2,parametri,NULL,0,0);

        checkResults(res,conn);

        stampatuple(res);

        break;

    default:

        break;

    }

    cout<<endl<<"chiusura menu info\n";

}

OUTPUT

Immagine che contiene testo, elettronico, screenshot, schermo

Descrizione generata automaticamente

…

Immagine che contiene testo, screenshot, monitor, nero

Descrizione generata automaticamente

Query 5:

Risultati Qualifica: visualizza la griglia di partenza di una determinata gara

void qualifica(PGconn\* conn)

{

    int anno;

    cout<<"inserire l'anno interessato: ";

    cin>>anno;

    string supp1 = to\_string(anno);

    const char\* parametro1 = supp1.c\_str();

    cout<<"selezionare il numero della gara della quale si vuole sapere la griglia di partenza: \n";

    string query = " SELECT gara.gara\_num AS numero\_gara,gara.nome\_gara FROM gara WHERE gara.anno = $1::int;";

    PGresult\* res = PQprepare(conn,"calendario",query.c\_str(),1,NULL);

    res = PQexecPrepared(conn,"calendario",1,&parametro1,NULL,0,0);

    checkResults(res,conn);

    stampatuple(res);

    int gara\_num;

    cout<<"inserire il numero: ";

    cin>>gara\_num;

    string supp2 = to\_string(gara\_num);

    const char\* parametro2 = supp2.c\_str();

    const char\* parametri[2] = {parametro1,parametro2};

    query = "

SELECT ROW\_NUMBER() OVER

(ORDER BY prestazione.tempo\_q3,prestazione.tempo\_q2,prestazione.tempo\_q1) AS Posizione\_Partenza,pilota.nome,pilota.cognome,pilota.sigla\_in\_gara,partecipante.numero\_in\_gara FROM pilota,partecipante,prestazione,autovettura

WHERE prestazione.codice\_fiscale = partecipante.codice\_fiscale AND

prestazione.vettura = prestazione.vettura AND

partecipante.codice\_fiscale = pilota.codice\_fiscale AND

prestazione.anno = $1::int AND

prestazione.gara\_num = $2::int AND

autovettura.anno = $1::int AND

partecipante.vettura = autovettura.nome;";

    res = PQprepare(conn,"griglia",query.c\_str(),2,NULL);

    res = PQexecPrepared(conn,"griglia",2,parametri,NULL,0,0);

    checkResults(res,conn);

    stampatuple(res);

}

OUTPUT

Immagine che contiene testo, screenshot, elettronico, computer

Descrizione generata automaticamente