



SAPIENZA  
UNIVERSITÀ DI ROMA

# Fly Competition - Query



*Gabriele Rodonò (1916563)*  
*Michele Palma (1849661)*

**Data di Consegna:**  
23/06/2022

**Basi di Dati (Modulo 2), CdL Informatica**

Sapienza Università di Roma

A.A 2021/2022, 2° anno, 2° semestre

Professore Giuseppe Pirrò

## 1) Tabella delle Operazioni:

Riportiamo anche in questo documento le query che abbiamo elaborato per il nostro progetto.

Operaz.	Specifica	Tipo	Frequenza
Op1	Calcolare il numero di Km totali fatti da ogni pilota nell'arco di un anno	B	5/anno
Op2	Visualizzare il totale delle iscrizioni per ogni competizione sportiva ed il budget raggiunto grazie al pagamento delle quote di iscrizione ed il numero degli sponsor della competizione	B	2/anno
Op3	Trovare per ogni pilota i dati del mezzo con il quale ha effettuato più km	I	5/mese
Op4	Ottenere la classifica della gara cross country X-Alps tenutasi nel 2019 e visualizzare per ogni pilota il mezzo con il quale si è iscritto alla competizione e il tempo impiegato	I	2/anno
Op5	Per ogni streamer, trovare il numero di trasmissioni effettuate durante la propria carriera	B	3/anno
Op6	Calcolare il numero di task di cui è composta una Gara di Campionato e il numero totale di punti che essa attribuisce. In particolare, sono richieste le Gare di Campionato tenutesi dal 2015 in poi, e si vuole sapere anche il luogo della Competizione.	I	10/mese
Op7	Data una traccia qualsiasi, scoprire a quale task (relativa ad una gara di campionato) della gara corrisponde, ed il numero di boe che prevede.	I	15/mese
Op8	Trovare la classifica dei modelli dei mezzi relative alle vincite avvenute nelle competizioni del 2022	B	1/anno
Op9	Identificare il Direttore di Gara che ha maturato la media annuale del numero di Competizioni dirette maggiore	I	2/anno
Op10	Ottenere la targa dell'ambulanza che è stata incaricata ad intervenire nel maggior numero di Competizioni Sportive	B	1/anno
Op11	Trovare il montepremi assoluto vinto negli anni da piloti che attualmente hanno un'età compresa tra i 18 e i 25 anni	I	7/mese
Op12	Ottenere lo sponsor che in assoluto ha sponsorizzato il maggior numero di competizioni nello stesso anno.	I	1/anno
Op13	Per ogni marca che produce parapendio, calcolare la somma dei km totali effettuati	I	1/mese
Op14	Trovare il Nome e l'edizione della Gara di Campionato composta da più task, ed il relativo sponsor.	I	1/mese
Op15	Ottenere l'età e il nickname dei piloti iraniani che si sono iscritti a più di 3 competizioni negli ultimi 5 anni e che hanno vinto complessivamente più di 200 euro	I	1/anno

## 2) Implementazione delle Query SQL:

Di seguito abbiamo riportato le implementazioni delle query SQL principali del nostro progetto, tralasciando quelle di inserimento che abbiamo utilizzato per popolare il nostro DB e poter testare quanto descritto di seguito.

### Op1:

“Calcolare il numero di Km totali fatti da un pilota nell'arco di un anno”

```
SELECT P.cf, Year(T.data) as anno, sum(T.km_percorsi) as kmPercorsi
FROM traccia T, pilota_dati_da_gara P
WHERE P.cf = T.cf_pilota
GROUP BY P.cf, Year(T.data)
```

### Op2:

“Visualizzare il totale delle iscrizioni per ogni competizione sportiva ed il budget raggiunto grazie al pagamento delle quote di iscrizione ed il numero degli sponsor della competizione”

```
CREATE OR REPLACE VIEW iscritti_competizioni as
SELECT C.nome, C.edizione, count(*) as iscritti, sum(costo_iscrizione) as incassi
FROM iscrizione I, competizione C
WHERE I.nome_competizione = C.nome AND I.edizione_competizione = C.edizione
GROUP BY C.nome, C.edizione
ORDER BY iscritti DESC;
```

```
CREATE OR REPLACE VIEW sponsor_competizioni as
SELECT C.nome, C.edizione, count(*) as num_sponsor
FROM competizione C, spon_competizione SC
WHERE C.nome = SC.nome_competizione AND C.edizione = SC.edizione_competizione
GROUP BY C.nome, C.edizione;
```

```
SELECT IC.nome, IC.edizione, SC.num_sponsor, IC.incassi
FROM iscritti_competizioni IC, sponsor_competizioni SC
WHERE IC.nome = SC.nome AND IC.edizione = SC.edizione;
```

### Op3:

“Trovare per ogni pilota i dati del mezzo con il quale ha effettuato più km”

```
CREATE OR REPLACE VIEW km_per_mezzo as
SELECT T.cf_pilota, T.mezzo, sum(km_percorsi) as km_effettuati
FROM traccia T, produzione P
WHERE T.mezzo = P.numero_seriale_mezzo
GROUP BY T.cf_pilota, T.mezzo;
```

```
CREATE OR REPLACE VIEW miglior_mezzo_per_pilota as
SELECT KM.cf_pilota, KM.mezzo as seriale_mezzo, KM.km_effettuati as km_totali
FROM km_per_mezzo KM
WHERE KM.km_effettuati = (SELECT MAX(km_effettuati) FROM km_per_mezzo KM1
                        WHERE KM.cf_pilota = KM1.cf_pilota);
```

```
SELECT MMPP.cf_pilota, MMPP.km_totali, PR.nome as marca, P.modello, MMPP.seriale_mezzo
FROM miglior_mezzo_per_pilota MMPP, produzione P, produttore PR
WHERE P.numero_seriale_mezzo = MMPP.seriale_mezzo AND PR.piva = P.piva;
```

### Op4:

“Ottenere la classifica della gara cross country X-Alps tenutasi nel 2019 e visualizzare per ogni pilota il mezzo con il quale si è iscritto alla competizione e il tempo impiegato”

```
CREATE OR REPLACE VIEW iscritti_XAlps_2019 as
SELECT I.cf_pilota, I.num_seriale_mezzo, GCC.nome_competizione, GCC.edizione_competizione
FROM gara_cross_country GCC, iscrizione I, pilota_dati_da_gara PDG
WHERE (I.nome_competizione = GCC.nome_competizione AND
      I.edizione_competizione = GCC.edizione_competizione AND
      I.cf_pilota = PDG.cf) AND
      GCC.nome_competizione = 'XAlps' AND GCC.edizione_competizione = 2019;
```

```
CREATE OR REPLACE VIEW tracce_XAlps_2019 as
SELECT ISC.cf_pilota, T.data, T.ora_partenza, T.ora_arrivo, TIMESTAMPDIFF(MINUTE,
T.ora_partenza, T.ora_arrivo) / 60 as tempo
FROM iscritti_XAlps_2019 ISC, traccia T, gara_cross_country_finita GCCF
WHERE ISC.cf_pilota = T.cf_pilota AND
      ISC.nome_competizione = GCCF.nome_competizione AND
      ISC.edizione_competizione = GCCF.edizione_competizione AND
      T.data >= GCCF.data_inizio AND T.data <= GCCF.data_fine;
```

```
CREATE OR REPLACE VIEW classifica_XAlps_2019 as
SELECT ROW_NUMBER() OVER() AS posizione,
      ISC.cf_pilota, M.modello, sum(tempo) * 60 as minuti_totali
FROM iscritti_XAlps_2019 ISC, mezzo M, tracce_XAlps_2019 TR
WHERE ISC.num_seriale_mezzo = M.num_seriale AND
      ISC.cf_pilota = TR.cf_pilota
GROUP BY ISC.cf_pilota
ORDER BY minuti_totali ASC;
```

### Op5:

“Per ogni streamer, trovare il numero di trasmissioni effettuate durante la propria carriera”

```
SELECT SO.nome_streamer, count(*) as num_stream
FROM streaming_online SO, trasmissione T, competizione C
WHERE C.nome = T.nome_competizione AND C.edizione = T.edizione_competizione AND
      T.data_orario_fine = SO.data_orario_fine AND T.data_orario_inizio = SO.data_orario_inizio
GROUP BY SO.nome_streamer;
```

### Op6:

“Calcolare il numero di task di cui è composta una Gara di Campionato e il numero totale di punti che essa attribuisce. In particolare, sono richieste le Gare di Campionato tenutesi dal 2015 in poi, e si vuole sapere anche il luogo della Competizione.”

```
SELECT GC.nome_competizione, GC.edizione_competizione, GC.data_inizio,
      count(*) as num_task, GC.luogo, sum(T.punteggio_attribuito) as punti_tot
FROM task T, gara_di_campionato GC
WHERE YEAR(GC.data_inizio) >= 2015 AND
      GC.nome_competizione = T.nome_competizione AND
      GC.edizione_competizione = T.edizione_competizione AND
      T.data_inizio = GC.data_inizio AND T.data_task >= GC.data_inizio
GROUP BY GC.nome_competizione, GC.edizione_competizione, GC.data_inizio, GC.luogo;
```

### Op7:

“Data una traccia qualsiasi, scoprire a quale task (relativa ad una gara di campionato) della gara corrisponde, ed il numero di boe che prevede.”

```
CREATE OR REPLACE VIEW pilota_con_tracce_camp as
SELECT PDG.cf, T.data, T.ora_partenza, T.ora_arrivo, T.tipologia_traccia
FROM traccia T, pilota_dati_da_gara PDG, iscrizione I
WHERE T.cf_pilota = PDG.cf AND PDG.cf = I.cf_pilota AND T.tipologia_traccia = 'G' AND
      (T.cf_pilota, I.nome_competizione, I.edizione_competizione) IN
      (
        SELECT ISR.cf_pilota, GDC.nome_competizione, GDC.edizione_competizione
        FROM iscrizione ISR, gara_di_campionato GDC
        WHERE ISR.nome_competizione = GDC.nome_competizione AND
              ISR.edizione_competizione = GDC.edizione_competizione
      );

SELECT G.nome_competizione, G.edizione_competizione, T.data_task, T.numero_boe
FROM gara_di_campionato G, task T
WHERE G.data_inizio <= T.data_task AND
      T.data_task IN (SELECT data FROM pilota_con_tracce_camp)
```

### Op8:

“Trovare la classifica dei modelli dei mezzi relative alle vincite avvenute nelle competizioni del 2022 ”

```
SELECT ROW_NUMBER() OVER(ORDER BY sum(P.vincita) DESC) AS posizione,  
       M.modello, sum(P.vincita) as montepremi  
FROM iscrizione I, mezzo M, Premio P, competizione C  
WHERE I.num_seriale_mezzo = M.num_seriale AND  
       I.nome_competizione = C.nome AND I.edizione_competizione = C.edizione AND  
       C.edizione = 2022 AND  
       P.nome_competizione = C.nome AND P.edizione_competizione = C.edizione AND  
       P.cf_pilota = I.cf_pilota  
GROUP BY M.modello;
```

### Op9:

“Identificare il Direttore di Gara che ha maturato la media annuale del numero di Competizioni dirette maggiore”

```
CREATE OR REPLACE VIEW direttore_con_num_direzioni AS  
SELECT DG.cf, D.edizione_competizione, count(*) as direz_per_anno  
FROM direttore_di_gara DG, direzione D  
WHERE DG.cf = D.cf_direttore  
GROUP BY D.cf_direttore, D.edizione_competizione;
```

```
CREATE OR REPLACE VIEW direct_con_media AS  
SELECT DG.nome, DG.cognome, count(*) as anni_diretti, AVG(direz_per_anno) as media  
FROM direttore_con_num_direzioni DND, direttore_di_gara DG  
WHERE DND.cf = DG.cf  
GROUP BY DG.cf, DG.nome, DG.cognome;
```

```
SELECT nome, cognome, anni_diretti, media  
FROM direct_con_media  
WHERE media = (SELECT MAX(media) FROM direct_con_media);
```

### Op10:

“Ottenere la targa dell’ambulanza che è stata incaricata ad intervenire nel maggior numero di Competizioni Sportive”

```
CREATE VIEW ambulanze_with_num_comp AS
SELECT PM.targa_ambulanza, count(*) as num_presidi
FROM presidio_medico PM
GROUP BY PM.targa_ambulanza;

SELECT targa_ambulanza, num_presidi
FROM ambulanze_with_num_comp
WHERE num_presidi = (SELECT MAX(num_presidi) FROM ambulanze_with_num_comp);
```

### Op11:

“Trovare il montepremi assoluto vinto negli anni da piloti che attualmente hanno un'età compresa tra i 18 e i 25 anni”

```
CREATE OR REPLACE VIEW pilota_anni as
SELECT cf, ((YEAR(curdate()) - YEAR(data_nascita)) -
            (RIGHT(curdate(),5) < RIGHT(data_nascita, 5))) as eta
FROM pilota_dati_anagrafici;

CREATE OR REPLACE VIEW piloti_18_25 as
SELECT PDA.nome, PDA.cognome, PDG.soprannome, PANNI.eta, PANNI.cf
FROM pilota_dati_anagrafici PDA, pilota_dati_da_gara PDG, pilota_anni PANNI
WHERE PDA.cf = PDG.cf AND PANNI.cf = PDA.cf AND PANNI.eta BETWEEN 18 AND 25;

SELECT P.cf, PDG.soprannome, P.eta_attuale, P.vincita
FROM pilota_dati_da_gara PDG, (SELECT PI.cf, PI.eta as eta_attuale, sum(PR.vincita) as vincita
                                FROM piloti_18_25 PI, premio PR
                                WHERE PI.cf = PR.cf_pilota
                                GROUP BY PI.cf, PI.eta) as P
WHERE P.cf = PDG.cf;
```

### Op12:

“Ottenere lo sponsor che in assoluto ha sponsorizzato il maggior numero di competizioni nello stesso anno.”

```
CREATE OR REPLACE VIEW sponsor_with_num_sponsorizzate AS
SELECT S.nome, SC.piva, SC.edizione_competizione, count(*) as num_comp_sponsorizz
FROM spon_competizione SC, sponsor S
WHERE SC.piva = S.piva
GROUP BY SC.piva, SC.edizione_competizione, S.nome;

SELECT nome, piva, edizione_competizione, num_comp_sponsorizz
FROM sponsor_with_num_sponsorizzate
WHERE num_comp_sponsorizz = (SELECT MAX(num_comp_sponsorizz) FROM
sponsor_with_num_sponsorizzate);
```

### Op13:

“Per ogni marca che produce parapendio, calcolare la somma dei km totali effettuati”

```
SELECT prod_info.nome as nome_prodotto, sum(T.km_percorsi) as km_totali
FROM traccia as T
    INNER JOIN produzione as prod ON T.mezzo = prod.numero_seriale_mezzo
    INNER JOIN modello_prodotto as build_info ON prod.modello = build_info.modello
    INNER JOIN produttore as prod_info ON prod.piva = prod_info.piva
WHERE categoria = 'P'
GROUP BY prod_info.nome
ORDER BY km_totali DESC;
```

### Op14:

“Trovare il Nome e l'edizione della Gara di Campionato composta da più task, ed il relativo sponsor.”

```
CREATE OR REPLACE VIEW TaskPerCompetizione AS
SELECT nome_competizione, edizione_competizione, count(*) as numTask
FROM task
GROUP BY nome_competizione, edizione_competizione;

SELECT T.nome_competizione, T.edizione_competizione, T.numTask, SP.nome
FROM TaskPerCompetizione T JOIN spon_competizione S JOIN sponsor SP ON
    T.nome_competizione = S.nome_competizione AND
    T.edizione_competizione = S.edizione_competizione
    AND S.piva = SP.piva
WHERE T.numTask = (SELECT MAX(numTask) FROM TaskPerCompetizione);
```



### Op15:

“Ottenere l'età e il nickname dei piloti iraniani che si sono iscritti a più di 3 competizioni negli ultimi 5 anni e che hanno vinto complessivamente più di 200 euro”

```
SELECT PDA.cf, PDG.soprannome, ((YEAR(curdate()) - YEAR(PDA.data_nascita))) as eta,  
       count(*) as num_iscrizioni  
FROM pilota_dati_anagrafici PDA, pilota_dati_da_gara PDG, iscrizione I  
WHERE PDA.nazionalita = 'iraniana' AND  
       PDG.cf = PDA.cf AND I.cf_pilota = PDA.cf AND  
       I.edizione_competizione >= YEAR(current_date()) - 5 AND  
       PDA.cf IN (SELECT P.cf_pilota  
                  FROM premio P  
                  WHERE P.vincita > 200)  
GROUP BY PDA.cf, PDG.soprannome, PDA.data_nascita  
HAVING count(*) >= 3;
```

## 3) Soddisfazione dei Requisiti:

Come richiesto dalle specifiche, riportiamo di seguito una tabella che riporta come abbiamo soddisfatto ogni requisito del punto H:

Punto	Requisito Minimo	Fly Competition
I	Devono richiedere l'utilizzo degli operatori aggregati	Tutte le query fanno uso degli operatori aggregati
II	Devono richiedere l'utilizzo di query annidate (di cui una con almeno tre livelli)	La query <b>numero 7</b> è a 3 livelli, le altre o utilizzano delle query annidate o fanno uso di viste per facilitare la leggibilità
III	La metà delle query deve essere scritta utilizzando le viste	9 query sono state scritte utilizzando le viste
IV	Devono coinvolgere almeno due tabelle	Ogni query fa utilizzo di almeno due tabelle