

Viste in MySQL

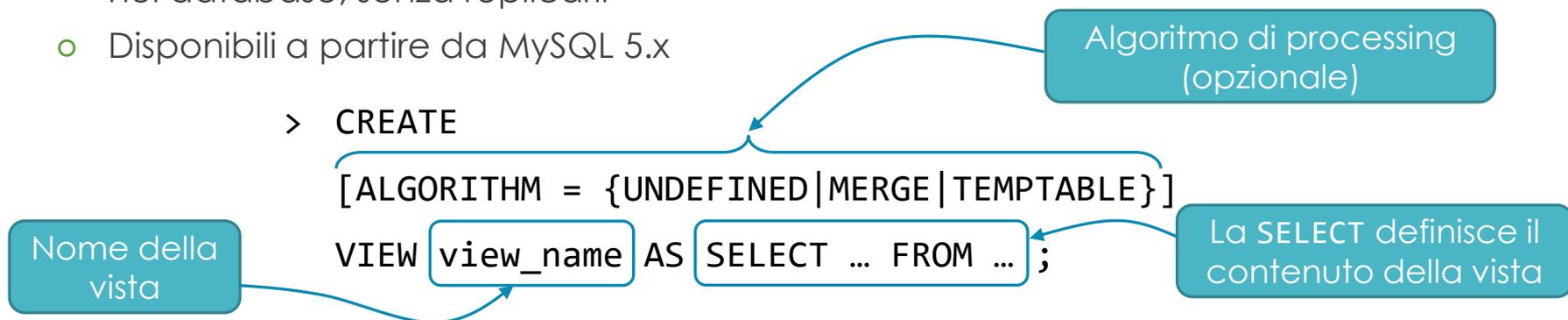
LABORATORIO DI BASI DI DATI
A.A. 2019/2020

Dott. Marco Savi

Contenuti riadattati a partire da slide gentilmente concesse
dai **Dott. Paolo Napoletano** e **Claudio Venturini**

Viste in MySQL

- × Sono "tabelle virtuali"
 - Definite con lo statement `CREATE VIEW`, tramite un'interrogazione `SELECT` il cui risultato è utilizzabile come se fosse una tabella
 - Consentono di definire **diverse "prospettive" di rappresentazione dei dati** memorizzati nel database, senza replicarli
 - Disponibili a partire da MySQL 5.x



- L'algoritmo utilizzato definisce la modalità di processing delle query sulla vista
- × *Normalmente* le viste sono utilizzabili **solo in lettura**, cioè *non sono aggiornabili*
 - Non supportano `INSERT`, `UPDATE` e `DELETE`
- × Per approfondire: <https://dev.mysql.com/doc/refman/5.7/en/views.html>

Viste in MySQL

- × Esempio:

Definiamo una vista che rappresenti la media dei voti degli studenti

> CREATE VIEW media_voti AS

SELECT che
definisce il
contenuto

```
SELECT s.matricola, s.cognome, s.nome, AVG(e.voto) AS media
FROM studente AS s
      INNER JOIN esame AS e ON s.matricola = e.studente
GROUP BY s.matricola, s.cognome, s.nome;
```

- × La vista `media_voti` è interrogabile come qualsiasi tabella

- × Esempio:

Selezionare gli studenti che hanno una media dei voti maggiore di 25

> SELECT * FROM media_voti WHERE media > 25;

Viste in MySQL – Algoritmi di processing

- × Gli algoritmi di processing definiscono la modalità di processing delle query effettuate sulla vista
- × **MERGE**
 - La query viene riscritta effettuando il merge tra essa e la SELECT che definisce la vista
- × **TEMPTABLE**
 - Il DBMS esegue prima l'interrogazione che definisce la vista e inserisce il risultato in una tabella temporanea
 - La query viene poi eseguita sulla tabella temporanea, che viene poi eliminata
- × **MERGE** è più efficiente ma utilizzabile solo se la vista non è definita con:
 - Operatori di aggregazione: `COUNT()`, `MIN()`, `MAX()`, `AVG()`, ...
 - Raggruppamenti: `GROUP BY`
 - Operatori insiemistici: `UNION`, `UNION ALL`
 - Clausole `DISTINCT`, `HAVING` e `LIMIT`
- × Se l'algoritmo è `UNDEFINED`, MySQL utilizza, se possibile, **MERGE**

Viste in MySQL – Viste aggiornabili

- × Una vista è aggiornabile se c'è un mapping 1 a 1 tra le sue tuple e quelle di una tabella fisica
- × In pratica, in MySQL una vista è **aggiornabile** solo se usa l'algoritmo di processing MERGE e se la definizione non contiene:
 - Operatori di aggregazione: COUNT(), MIN(), MAX(), AVG(), ...
 - Raggruppamenti: GROUP BY
 - Operatori insiemistici: UNION, UNION ALL
 - Clausole DISTINCT e HAVING
 - Subquery nella clausola WHERE se si tratta di *subquery correlate*
 - Riferimenti ad altre viste non aggiornabili nella clausola FROM
 - Outer join: LEFT JOIN, RIGHT JOIN
- × Inoltre, la INSERT è supportata solo se la vista seleziona tutte le colonne delle tabelle coinvolte che non hanno un valore di default, e se le colonne selezionate *non sono* espressioni definite sulle colonne delle tabelle referenziate
- × Solo una delle tabelle coinvolte nella vista può essere aggiornata:
 - Nel caso di UPDATE, la clausola SET può fare riferimento solamente a colonne di una sola delle tabelle

Viste in MySQL – Modifica/eliminazione

- × Per **modificare** una vista si usa lo statement ALTER VIEW
 - <https://dev.mysql.com/doc/refman/5.7/en/alter-view.html>
 - La sintassi è identica a quella dello statement CREATE VIEW
 - Poiché le viste sono "virtuali", e quindi non contengono dati, è anche **possibile semplicemente eliminare la vista e ricrearla**

> ALTER

```
[ALGORITHM = {UNDEFINED|MERGE|TEMPTABLE}]
```

```
VIEW view_name AS SELECT ... FROM ... ;
```

- × Per **eliminare** una vista si usa lo statement DROP VIEW
 - Specificando la clausola IF EXISTS la vista viene eliminata solo se esiste
 - <https://dev.mysql.com/doc/refman/5.7/en/drop-view.html>

> DROP VIEW [IF EXISTS] media_voti;

Esercizi

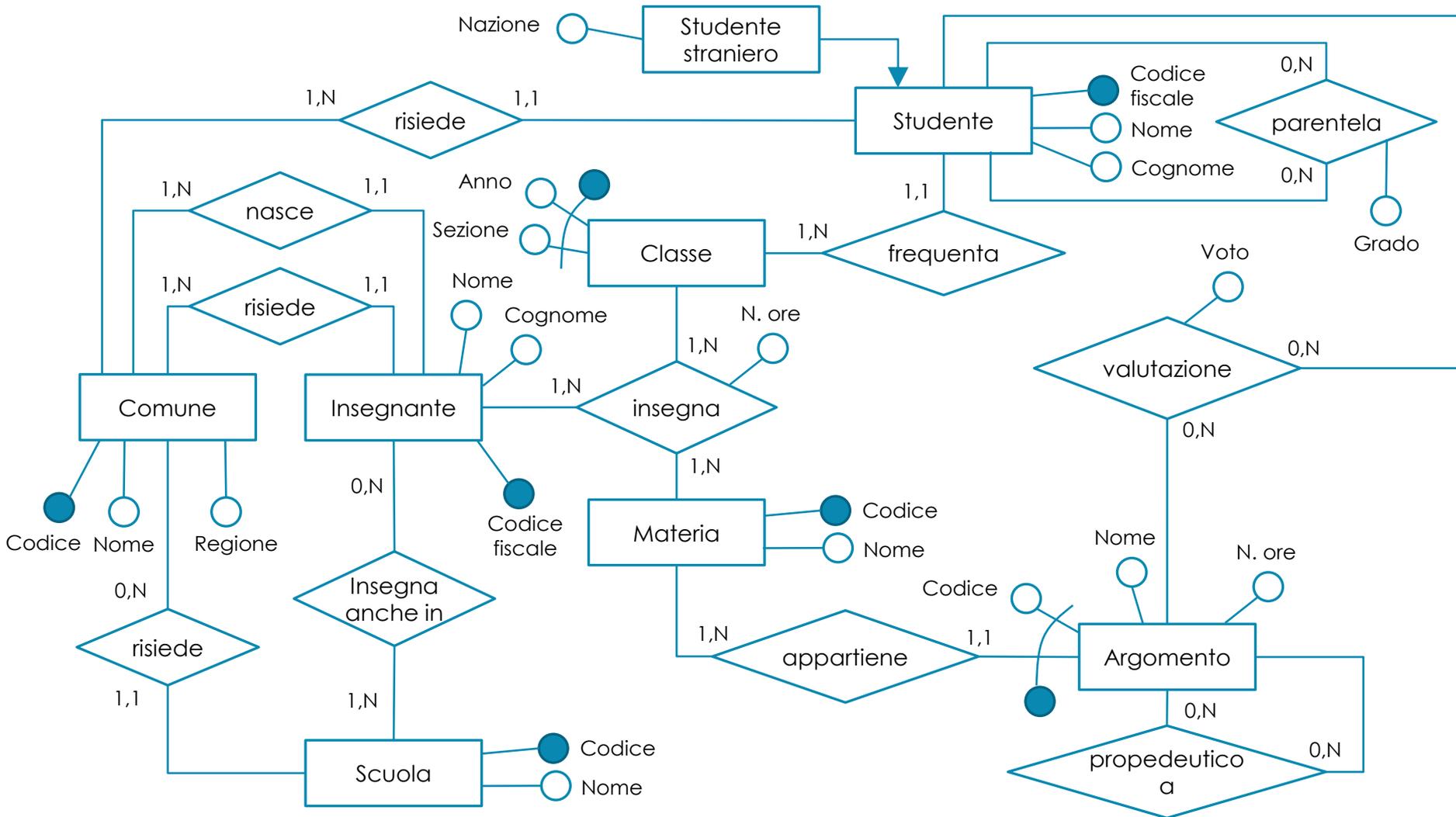
Esercizio 12

- × Utilizzare il database progettato nell'Esercizio 7 e popolato nell'Esercizio 11 (scaricabile dalla piattaforma di eLearning con il nome `es11_popolato.sql`):
Scuola Secondaria

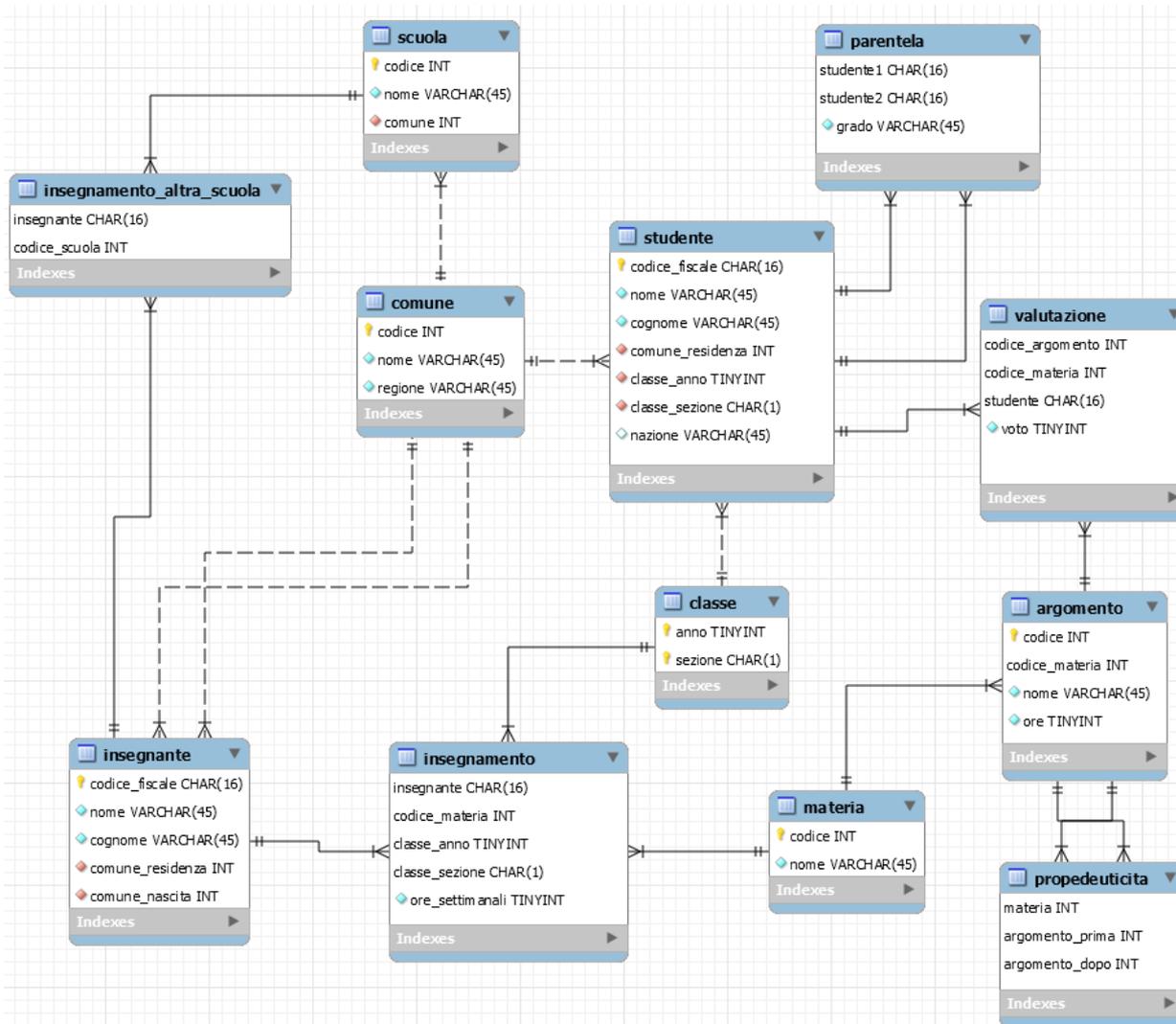
- × **Creazione di viste**
 1. Costruire una vista che selezioni la valutazione media degli studenti di ogni classe (ad es. 2° B → 6.8, 3° A → 7.1) e ordini le classi in ordine decrescente per media
 2. Costruire una vista che selezioni gli studenti (con nome, cognome, codice fiscale e comune di residenza) che *non* hanno ottenuto valutazioni insufficienti

- × **Interrogazione sulle viste**
 1. Selezionare il numero di studenti che *non* hanno ottenuto insufficienze divisi per comune di residenza
 2. Selezionare gli insegnanti (nome e cognome) che insegnano in classi con valutazioni medie minori di 8.0

Esercizio 12 – Modello ER



Esercizio 12 – Modellazione in Workbench



Esercizio 13 – Tema d'esame 2013

- × Tempo di svolgimento: 4 ore

- × **Svolgimento esame**
 - Si richiede di implementare con l'utilizzo di MySQL Workbench il tema sotto riportato, secondo lo schema E-R dato
 1. Definire lo schema logico del database mediante l'EER Diagram di Workbench;
 2. Creare il database e le relative tabelle;
 3. Popolare il database secondo quanto specificato di seguito;
 4. Implementare ed eseguire almeno 3 delle 8 query riportate.

Esercizio 13 – Tema d'esame 2013

× Tema del database

- Il database “Mondiale2009” contiene i dati relativi ai risultati del campionato mondiale di Formula 1 della stagione 2009.
- La base di dati dovrà contenere alcune semplici informazioni sui piloti che hanno preso parte alle gare del mondiale di Formula 1, anno 2009 (codice pilota, cognome, nome e nazionalità).
- Inoltre, per le squadre si dovrà specificare anche il nome della squadra, il motore montato sulle autovetture (si suppone che tutte le macchine di ogni squadra montino lo stesso motore) e un codice identificativo.
- Dovranno essere registrate le informazioni relative ai singoli Gran Premi che sono stati disputati. Per ogni gran premio si dovrà tenere traccia, in particolare, della nazione in cui si è svolto, del nome, del circuito, del numero di giri che i piloti devono compiere per arrivare al traguardo, e della data in cui si svolge.
- Sarà poi necessario tenere traccia dei risultati ottenuti dai piloti in ciascun gran premio disputato, in particolare la data di svolgimento della gara, la posizione di arrivo, l'eventuale motivo del ritiro, i giri effettuati, i punti guadagnati.

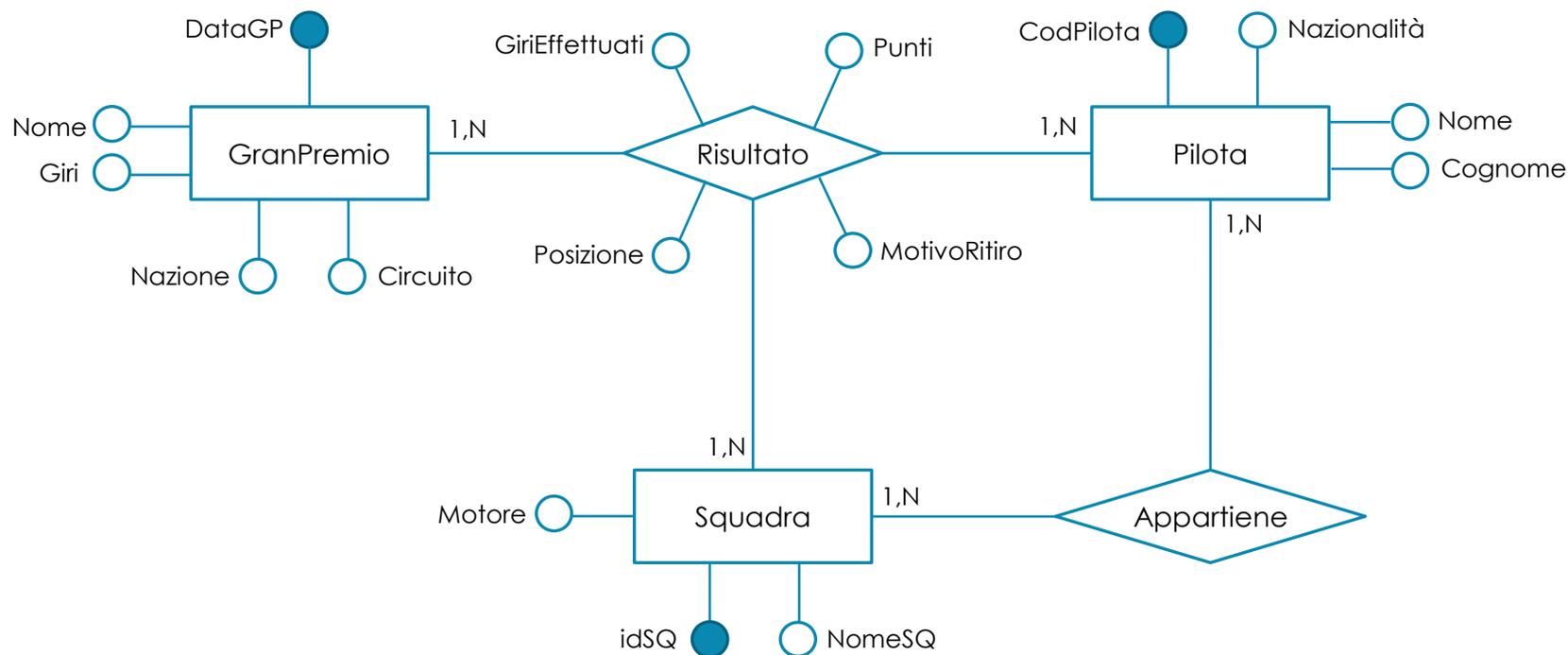
Esercizio 13 – Tema d'esame 2013

× Tema del database (continua...)

- Per la relazione Risultato si noti che:
 - l'attributo relativo alla posizione di arrivo deve essere nullo nel caso in cui il pilota non abbia terminato la gara;
 - l'attributo relativo ai giri compiuti dal pilota è inferiore o uguale al numero di giri totali del gran premio;
 - l'attributo relativo al motivo del ritiro è nullo quando il pilota termina regolarmente la gara.

Esercizio 13 – Tema d'esame 2013

× Modello E-R



Esercizio 13 – Tema d'esame 2013

× Query

1. Stilare una classifica dei piloti (composta da codice pilota, nome, cognome, nazionalità e totale dei punti), ordinandola per punteggio decrescente;
2. Stilare una classifica delle squadre (composta da id squadra, nome, motore e totale dei punti), ordinandola per punteggio decrescente;
3. Relativamente ai piloti che hanno partecipato a meno di 4 gran premi, selezionare il nome dei piloti e il numero di gran premi disputati;
4. Per ciascun pilota, selezionare il nome e il numero di gare in cui ha effettuato almeno il 50% dei giri totali;
5. Per ciascun gran premio, indicarne il nome, la nazione, il circuito e il numero di piloti che si sono ritirati;
6. Effettuare una classifica delle cause di ritiro (escludendo il valore NULL);
7. Individuare i piloti che hanno corso per due squadre diverse. Stampare il nome e cognome dei piloti ed il nome delle due squadre;
8. Individuare le squadre per cui hanno corso più di due piloti. Stampare il nome della squadra ed il numero di piloti corrispondente.

Esercizio 13 – Tema d'esame 2013

× **Avvertenze per il popolamento del database**

- Per il popolamento del database potete utilizzare dati reali relativi al mondiale di F1 del 2009 presenti in internet sui vari siti specializzati, per esempio http://it.wikipedia.org/wiki/Campionato_mondiale_di_Formula_1_2009
- In ogni caso, il popolamento del database dovrà tenere conto delle seguenti informazioni:
 - Una squadra può cambiare i propri piloti nel corso della stagione e, analogamente, un pilota può correre per squadre diverse (in gare diverse, ovviamente).
 - La tabella che si riferisce alle squadre dovrà avere almeno 5 record.
 - Ogni squadra concorre in ogni gran premio con due autovetture.
 - In totale sono stati disputati almeno 6 gran premi.

Esercizio 13 – Tema d'esame 2013

× **Modalità di consegna**

- Lo schema logico dovrà essere salvato in formato .mwb, il file dovrà essere nominato “Matricola_Cognome_Nome.mwb”;
- Il database dovrà essere esportato mediante il tool di Data Export (Server → Data Export) in un singolo file self-contained e dovrà contenere gli script sql per la creazione e il popolamento del database. Il file dovrà essere nominato “Matricola_Cognome_Nome.sql”;
- Ogni query dovrà essere salvata in un singolo file sql nominato con il numero della query stessa (es. per la query 1 sarà “Matricola_Cognome_Nome_Q1.sql”).
- Tutti i file di cui sopra dovranno essere compressi in un unico file nominato: “Matricola_Cognome_Nome.zip”