

**Scrivete le soluzioni delle due parti su DUE FOGLI DISTINTI, entrambi intestati**

**Parte 1: LINGUAGGI DI INTERROGAZIONE (su un foglio a parte rispetto alla Parte 2)**

AUTOBUS(TARGA, ANNOIMMATIC, MODELLO, MARCA, NO\_POSTI)

CONDUCENTE(MATRCOND, NOMECONDUCENTE, TELEFONO)

CORSA(IDCORSA, CITTÀPARTENZA, CITTÀARRIVO, ORAPARTENZA, TEMPOSTIMATO)

TURNO (IDCORSA, DATA, MATRCOND, BUS)

Lo schema soprastante descrive una rete di trasporti via autobus su lunghe distanze. Ogni Bus in una certa giornata percorre una o più volte la medesima tratta, in entrambe le direzioni (andata e ritorno, ad es: Milano-Roma e poi Roma-Milano).

**A. SQL (10 punti per le query + 3 punti per DDL e vincolo)**

1. Specificare i comandi di creazione delle tabelle TURNO e CORSA, definendo i vincoli di tupla e di dominio ritenuti opportuni ed esprimendo eventuali vincoli di integrità referenziale verso le altre tabelle. (1 p.)
2. Estrarre la targa dei bus con almeno 50 posti di capienza che hanno viaggiato a luglio 2018 ma nel luglio 2018 non sono partiti da Milano. (3 p.)
3. Estrarre targa e anno di immatricolazione dei bus che sono stati allocati il maggior numero di volte su tratte con arrivo a Roma e sono stati guidati almeno due volte da “Marco Rossi”. (4 p.)
4. Estrarre, per ogni autista che ha almeno 300 giornate di servizio, il numero di autobus distinti che ha guidato. (3 p.)
5. Esprimere il vincolo che verifica che nessun bus sia assegnato a tratte diverse in una stessa giornata. (2 p.)

**B. Linguaggi formali (6 punti)**

6. Formulare in Algebra Relazionale, Calcolo Relazionale e Datalog l’interrogazione che estrae i nomi dei conducenti che hanno guidato almeno due volte una tratta con arrivo a Milano ma mai tratte con arrivo a Roma. (2 + 2 + 2 p.)

**Parte 2: PROGETTAZIONE E TEORIA (su un foglio a parte rispetto alla Parte 1)**

**C. Progetto Concettuale e Logico (13 punti)**

Un gruppo di ricercatori organizza ogni anno un evento internazionale, finanziato da una organizzazione scientifica. Quattro ricercatori formano un comitato organizzatore e sono caratterizzati da nome, affiliazione, email e sito web. L’evento si ripete ogni anno, in una particolare località, e ha una trentina di partecipanti, tra cui anche gli organizzatori, caratterizzati da nome, affiliazione ed email.

Ogni anno, circa la metà dei partecipanti dell’edizione precedente viene confermata, e nuovi partecipanti vengono invitati. Tutti coloro che hanno partecipato ad una edizione divengono membri di un “gruppo di interesse” e ricevono una newsletter. Ogni evento annuale si tiene in due giornate, tipicamente all’inizio di settembre, ed ha un tema e una agenda, nella quale si individua una sequenza di sessioni, caratterizzate da un titolo e un orario di inizio e di fine; in alcune sessioni c’è uno speaker, altre sessioni sono lasciate per discussioni, talvolta organizzate per gruppi, e nel programma compare l’elenco degli animatori. Speaker e animatori fanno parte dei partecipanti.

L’organizzazione gestisce i pernottamenti dei partecipanti. Ogni partecipante indica il giorno di arrivo e di partenza e può essere alloggiato in uno dei tre alberghi messi a disposizione dall’organizzazione. I costi di pernottamento relativi alle due notti che precedono i due giorni dell’evento vengono gestiti dalla organizzazione, mentre eventuali pernottamenti aggiuntivi sono a carico dei partecipanti. Sei o sette partecipanti, tra cui gli organizzatori, si fermano un terzo giorno per scrivere i principali risultati prodotti durante l’evento; per loro l’organizzazione copre le spese di tre notti.

- Il progetto è valutato per completezza, correttezza, leggibilità, minimalità e autoesplicitività. Si ricordi che anche il progetto logico è un **grafo** i cui nodi e archi devono essere disposti coerentemente con quelli del progetto concettuale.
- Ricordare anche di specificare gli **identificatori** di tutte le entità e le **cardinalità** di tutte le associazioni, e di disporre i due grafi su due facciate affiancate, in modo da poterli osservare simultaneamente.

**D. Teoria (1 punto):**

Confrontare molto brevemente dal punto di vista del potere espressivo l’algebra relazionale con l’SQL

A.1

create table Corsa (

IdCorsa integer primary key,  
CittàPartenza varchar(30),  
CittàArrivo varchar(30),  
OraPartenza time,  
TempoStimato interval hour to minute

)

create table Turno (

IdCorsa integer references Corsa( IdCorsa ) on delete no action on update cascade,  
Data Date,  
MatrCond integer references Conducente( MatrCond ) on delete no action on update cascade,  
Bus char(7) references Autobus( Targa ) on delete no action on update cascade,  
primary key( IdCorsa, Data )

)

A.2

select Targa

from Autobus join Turno on Bus = Targa

where No\_Posti >= 50 and Data between 1/7/18 and 31/7/18

and Targa not in ( select Bus

from Corsa natural join Turno

where Data between 1/7/18 and 31/7/18 and CittàPartenza = "Milano" )

A.3

create view BusGuidatiDueVolteDaMarcoRossi( Bus ) as (

select Bus  
from Turno natural join Conducente  
where NomeConducente = "Mario Rossi"  
group by Bus  
having count(\*) > 1

)

create view NumeroTurniVersoRomaDeiBusGuidatiDaRossi( Targa, QteVolte ) as (

select Bus, count( \* )  
from BusGuidatiDueVolteDaMarcoRossi natural join Turno natural join Corsa  
where CittàArrivo = Roma  
group by Bus

)

select Targa, AnnoImmatricolazione

from Autobus

where Targa in ( select Targa

from NumeroTurniVersoRomaDeiBusGuidatiDaRossi

where QteVolte = ( select max( QteVolte )

from NumeroTurniVersoRomaDeiBusGuidatiDaRossi ) )

A.4

```
select MatrCond, count( distinct Bus ) as NumeroBusDistintiGuidati
from Turno
group by MatrCond
having count( distinct Data ) >= 300
```

A.5

```
create assertion StessaTrattaTuttoIlGiorno
check not exists ( select Bus, Data
                    from Turno natural join Corsa
                    group by Bus, Data
                    having count( distinct CittàPartenza ) > 1 or count( distinct CittàArrivo ) > 1 )
```

B.

```
UnaVoltaVersoRoma( MatrCond ) :- Turno( IC, _, MatrCond, _ ), Corsa( Ic, _, "Roma", _, _ )
DueVolteVersoMilano( MatrCond ) :- Turno( IdC1, _, MatrCond, _ ), Turno( IdC2, _, MatrCond, _ ),
                                   Corsa( IdC1, _, "Milano", _, _ ), Corsa( IdC1, _, "Milano", _, _ ), IdC1 ≠ IdC2
DueVolteVersoMilano( MatrCond ) :- Turno( IdC, Data1, MatrCond, _ ), Turno( IdC, Data2, MatrCond, _ ),
                                   Corsa( IdC, _, "Milano", _, _ ), Data1 ≠ Data2
MilanoSiRomaNo( M, N ) :- Conducente( M, N, _ ), DueVolteVersoMilano( M ), ¬ UnaVoltaVersoRoma( M )
? - MilanoSiRomaNo( Matr, Nome )
```

$$\text{UnaRoma} = \prod_{\text{MatrCond}} ( \text{Turno} \bowtie \sigma_{\text{CittàArrivo}=\text{"Roma"}} \text{Corsa} )$$
$$\text{DueMilano} = \prod_{\text{MatrCond}} ( ( \text{Turno} \bowtie \sigma_{\text{CittàArrivo}=\text{"Milano"}} \text{Corsa} )$$
$$\bowtie_{\text{MatrCond}} = \text{MC} \wedge ( \text{IdCorsa} \neq \text{IC} \vee \text{Data} \neq \text{D} )$$
$$\rho_{\text{IC,D,MC} \leftarrow \text{IdCorsa,Data,MatrCond}} ( \text{Turno} \bowtie \sigma_{\text{CittàArrivo}=\text{"Milano"}} \text{Corsa} )$$
$$\prod_{\text{NomeConducente}} ( \text{Conducente} \bowtie ( \text{DueMilano} - \text{UnaRoma} )$$
$$\{ t \mid \exists t_A \in \text{Conducente}, \exists t_C \in \text{Corsa}, \exists t_T \in \text{Turno} \mid$$
$$t[\text{Matricola}, \text{Nome}] = t_A[\text{MatrCond}, \text{NomeConducente}] \wedge$$
$$t_A[\text{MatrCond}] = t_T[\text{MatrCond}] \wedge t_T[\text{IdCorsa}] = t_C[\text{IdCorsa}] \wedge t_C[\text{CittàArrivo}] = \text{"Milano"} \wedge$$
$$( \exists t_{C2} \in \text{Corsa}, \exists t_{T2} \in \text{Turno} \mid$$
$$t_{T2}[\text{MatrCond}] = t_A[\text{MatrCond}] \wedge t_{T2}[\text{IdCorsa}] = t_{C2}[\text{IdCorsa}] \wedge t_{C2}[\text{CittàArrivo}] = \text{"Milano"} \wedge$$
$$( t_{C2}[\text{IdCorsa}] \neq t_C[\text{IdCorsa}] \vee t_{T2}[\text{Data}] \neq t_T[\text{Data}] ) ) \wedge$$
$$\neg ( \exists t_{CR} \in \text{Corsa}, \exists t_{TR} \in \text{Turno} \mid$$
$$t_A[\text{MatrCond}] = t_{TR}[\text{MatrCond}] \wedge t_{TR}[\text{IdCorsa}] = t_{CR}[\text{IdCorsa}] \wedge t_{CR}[\text{CittàArrivo}] = \text{"Roma"} )$$
$$\}$$

C.

...

D.

*Schematicamente, le principali differenze:*

ALGEBRA

Linguaggio operativo (la query è un query plan)

Semantica Set-oriented (elimina i duplicati)

Non supporta l'aggregazione (nella versione studiata)

SQL

Linguaggio dichiarativo (richiede la generazione di un plan)

Semantica Bag-oriented (non li elimina automaticamente)

Ha raggruppamento e aggregati

Non supporta la ricorsione

Supporta (limitatamente) la ricorsione

In conclusione, l'SQL è strettamente più espressivo per almeno due ragioni rispetto all'algebra relazionale (nella versione studiata nel corso).