

Esercizio 1

Sia dato il seguente schema

```
PERSONA (id_persona, nome, cognome, telefono, nazionalità, sesso, anno_nascita,
primo_anno_attività)
FILM (titolo, id_regista, genere, anno, durata)
PARTECIPAZIONE (id_attore, titolo_film, ruolo, compenso)
```

1

Trovare gli attori (nome, cognome) di nazionalità tedesca che hanno recitato come comparse in film il cui regista è di nazionalità francese e non hanno mai recitato come attori protagonisti in alcun film

```
SELECT Att.nome, Att.cognome
FROM Persona Att
JOIN Partecipazione P ON Att.id_persona = P.id_attore
JOIN Film F ON F.titolo = P.titolo_film
JOIN Persona Reg ON F.id_regista = Reg.id_persona
WHERE Att.nazionalità = "Tedesca"
AND Reg.nazionalità = "Francese"
AND Att.ruolo = "Comparsa"
AND Att.id_persona NOT IN (
    SELECT DISTINCT A.id_persona
    FROM Persona A JOIN Partecipazione P ON A.id_persona = P.id_attore
    WHERE A.ruolo = "Protagonista"
)
```

2

Trovare i film di ciascun regista per i quali l'insieme di attori coinvolti è stato pagato di più rispetto agli altri film dello stesso regista

```
SELECT P.titolo_film
FROM Partecipazione P
JOIN Film F ON F.titolo = P.titolo_film
GROUP BY P.titolo_film
HAVING SUM(P.compenso) >= ALL(
    SELECT SUM(compenso)
    FROM Partecipazione P1
    JOIN Film F2 ON F2.titolo = P1.titolo_film
    WHERE F.id_regista = F2.id_regista
    GROUP BY P2.titolo_film
)
```

Esercizio 2

Si consideri il seguente schema.

```
Film(ID, Titolo, AnnoProduzione, Incasso)
Partecipa(IDAttore, IDFilm, Ruolo, Compenso)
Attore(ID, Nome, AnnoNascita, LuogoNascita, Nazionalità)
```

Si suppone che ogni attore abbia avuto un solo ruolo in ciascun film.

1

Trovare i film in cui recitano almeno due attori inglesi ma non recita alcun attore americano.

```
SELECT F.titolo, F.ID
FROM Attore A JOIN Partecipa P ON A.ID = P.IDAttore
JOIN Film F ON F.ID = P.IDFilm
WHERE A.Nazionalità = "Inglese"
AND F.ID NOT IN (
  SELECT F2.ID
  FROM Film F2 JOIN Partecipa P2 ON F2.ID = P2.IDFilm
  JOIN Attore A2 ON A2.ID = P2.IDAttore
  WHERE A2.Nazionalità = "Americana"
)
GROUP BY F.titolo, F.ID
```

2

Trovare gli attori che nell'anno di produzione di almeno un film in cui hanno recitato come protagonisti avevano 50 anni

```
SELECT DISTINCT A.Nome, A.ID
FROM Attore A
JOIN Partecipa P ON A.ID = P.IDAttore
JOIN Film F ON F.ID = P.IDFilm
WHERE F.AnnoProduzione - A.AnnoNascita >= 50
```

3

Trovare i titoli dei film nei quali il compenso pagato per tutte le comparse sia superiore al compenso pagato per tutti i protagonisti

```
SELECT F.titolo
FROM Film F JOIN Partecipa P ON F.ID = P.IDFilm
WHERE P.Ruolo = "Comparsa"
GROUP BY F.titolo
HAVING SUM(P.Compenso) >= (
  SELECT SUM(P2.Compenso)
  FROM Partecipa P2
  WHERE P2.Ruolo = "Protagonista"
  AND P2.IDFilm = P.IDFilm
)
```

Esercizio 3

Si consideri il seguente schema che rappresenta una base di dati impiegata in una catena di librerie distribuite su tutto il territorio nazionale. In particolare oltre ai dati sulle singole librerie, sono archiviati i dati del personale e dei libri in vendita. Si osservi che possono esistere diverse versioni dello stesso libro (autore e titolo), edite da case editrici diverse. Infine attraverso la relazione Catalogo è possibile conoscere in quali librerie una determinata versione di un libro sia disponibile per essere venduta, il numero di copie a disposizione, ed il costo.

```
LIBRERIA(idLibreria, nome, indirizzo, città, oraApertura, oraChiusura,
turnoChiusura)
```

```
PERSONALE(idPersona, cognome, nome, idLibreria, reparto, turno)
CATALOGO(idLibro, idLibreria, reparto, numerocopie, costo)
LIBRO(idLibro, titolo, autori, casaEditrice, annoPubblicazione)
```

1

Il nome e la città delle librerie dove siano presenti almeno una copia del libro "le basi dell'informatica" e una copia del libro "qualità dei dati: concetti metodologie e tecniche", indipendentemente dalle diverse versioni dei libri stampate da più case editrici

```
CREATE VIEW HaBasiInformatica AS
SELECT DISTINCT L.nome, L.città
FROM Libreria L
JOIN Catalogo C ON L.idLibreria = C.idLibreria
JOIN Libro LB ON C.idLibro = LB.idLibro
WHERE LB.titolo = "le basi dell'informatica"

SELECT L.nome, L.città
FROM Libreria L
JOIN Catalogo C ON L.idLibreria = C.idLibreria
JOIN Libro LB ON C.idLibro = LB.idLibro
WHERE LB.titolo = "qualità dei dati: concetti metodologie e tecniche"
AND (L.nome, L.città) IN HaBasiInformatica
```

2

La libreria di Milano con il numero maggiore di libri in vendita (contando quindi tutte le copie di ogni libro)

```
SELECT L.nome, L.idLibreria
FROM Libreria L
JOIN Catalogo C ON L.idLibreria = C.idLibreria
WHERE L.città = "Milano"
GROUP BY L.nome, L.idLibreria
HAVING SUM(c.numerocopie) >= ALL(
    SELECT SUM(C1.numerocopie)
    FROM Libreria L1 JOIN Catalogo C1 ON L1.idLibreria = C1.idLibreria
    WHERE L1.città = "Milano"
    GROUP BY L1.idLibreria
)
```

Esercizio 4

Data la seguente base di dati che descrive un campionato invernale di sci maschile

```
LUOGOGARA(nome, altitudine, nazione)
GARA (specialità, luogo, dataGara, lunghezza, dislivello, premiovincitore)
POSIZIONEGARA (idatleta, dataGara, numeroPettorina, specialità, luogo,
posizionearrivo)
ATLETA (idatleta, nome, cognome, nazione, datadinascita)
```

1

Trovare per ogni nazione l'atleta che ha vinto il maggior numero di gare (assumendo che i valori di posizionearrivo siano specificati da interi)

```

SELECT A.idAtleta, A.nazione
FROM Atleta A JOIN PosizioneGara P ON P.idAtleta = A.idAtleta
WHERE posizionearrivo = 1
GROUP BY A.idAtleta, A.nazione
HAVING COUNT(*) >= ALL(
    SELECT COUNT(*)
    FROM Atleta A2 JOIN PosizioneGara P2 ON P2.idAtleta = A2.idAtleta
    WHERE posizionearrivo = 1
    AND A.nazione = A2.nazione
    GROUP BY A2.idAtleta
)

```

2

Trovare le nazioni che hanno almeno un'atleta che ha vinto almeno una gara in patria

```

SELECT DISTINCT A.nazione
FROM Atleta A JOIN PosizioneGara P ON A.idAtleta = P.idAtleta
JOIN LuogoGara L ON L.nome = P.luogo
WHERE A.nazione = L.nazione
AND P.posizionearrivo = 1

```

Esercizio 5

Cliente(CodiceFiscale, Nome, Cognome, Nazionalità, Telefono, e-mail) Stanza(CodiceStanza, Albergo, Tipo, MetriQuadri, Locali, Piano) Soggiorno(CodiceStanza, Albergo, DataInizio, DataFine, Agente, CFCliente, CostoGiornaliero) Albergo(CodiceAlbergo, NumeroDiStelle, Indirizzo, Città, Stato) Agente(CodiceAgente, Nome, Cognome, DataDiNascita).

1

Trovare, specificando nome cognome e codice fiscale, i clienti omonimi (stesso nome e cognome e codice fiscale diverso) che abbiano soggiornato in una qualunque stanza di albergo nella medesima città.

```

SELECT C.Nome, C.Cognome, C.CodiceFiscale
FROM Cliente C JOIN Soggiorno S ON S.CFCliente = C.CodiceFiscale
JOIN Albergo A ON S.Albergo = A.CodiceAlbergo
WHERE (C.Nome, C.Cognome) IN(
    SELECT C1.Nome, C1.Cognome
    FROM Cliente C1 JOIN Soggiorno S1 ON S1.CFCliente = C1.CodiceFiscale
    JOIN Albergo A1 ON S1.Albergo = A1.CodiceAlbergo
    WHERE A.Città = A1.Città
    AND C.Nome = C1.Nome
    AND C.Cognome = C1.Cognome
    AND C.CodiceFiscale <> C1.CodiceFiscale
)

```

2

Trovare l'agente (codice, nome, cognome) che ha stipulato più soggiorni per l'anno 2009 (cominciati nel 2009) in alberghi con più di 3 stelle

```

SELECT AG.Nome, AG.Cognome, AG.CodiceAgente
FROM Agente AG
JOIN Soggiorno S ON AG.CodiceAgente = S.Agente
JOIN Albergo A ON S.Albergo = A.CodiceAlbergo

```

```

WHERE S.DataInizio >= 1-1-2009
AND A.NumeroDiStelle > 3
GROUP BY AG.Nome, AG.Cognome, AG.CodiceAgente
HAVING COUNT(*) >= ALL(
    SELECT COUNT(*)
    FROM Agente AG1
    JOIN Soggiorno S1 ON AG1.CodiceAgente = S1.Agente
    JOIN Albergo A1 ON S1.Albergo = A1.CodiceAlbergo
    WHERE S1.DataInizio >= 1-1-2009
    AND A1.NumeroDiStelle > 3
    GROUP BY AG.CodiceAgente
)

```

Esercizio 6

Si consideri il seguente schema di basi di dati che descrive le prenotazioni on-line di una compagnia di traghetti nazionali che effettua diverse tratte. La compagnia dispone di più navi. Ogni cliente al primo accesso deve registrarsi ed ottenere il proprio codice, valido per gli accessi successivi. Effettuando la prenotazione il cliente indica il tipo di sistemazione (cabina, poltrona, passaggio ponte, ...) ed in caso di trasporto di auto ne indica la targa.

```

CLIENTE (codice_cliente, nome, cognome, telefono, primo_accesso, ultimo_accesso)
PRENOTAZIONE(codice_prenotazione, codice_cliente, portopartenza, portoarrivo,
nome_nave, data_prenotazione, data_partenza, numero_passeggeri, sistemazione,
auto, costo)
NAVE (nome_nave, anno_immatricolazione, numero_cabine)
TRATTE(portopartenza, portoarrivo, distanza)

```

1

Trovare i clienti (codice_cliente, nome e cognome) che si sono registrati ma non hanno mai effettuato prenotazioni

```

SELECT C.codice_cliente, C.nome, C.cognome
FROM Cliente C
WHERE C.codice_cliente NOT IN (
    SELECT codice_cliente FROM Prenotazione
)

```

2

Trovare i clienti (codice_cliente, nome e cognome) che hanno effettuato nel 2009 un viaggio di andata e ritorno

```

SELECT C.codice_cliente, C.nome, C.cognome
FROM Cliente C
JOIN Prenotazione P ON C.codice_cliente = P.codice_cliente
WHERE P.data_partenza BETWEEN 1-1-2009 AND 31-12-2009
AND C.codice_cliente IN (
    SELECT C1.codice_cliente
    FROM Cliente C1 JOIN Prenotazione P1 ON C1.codice_cliente = P1.codice_cliente
    WHERE P1.portopartenza = P1.portoarrivo
    AND P1.data_partenza BETWEEN 1-1-2009 AND 31-12-2009
)

```

3

Trovare la nave che ha trasportato più passeggeri nel 2009

```
SELECT N.nome_nave
FROM Nave N
JOIN Prenotazione P ON N.nome_nave = P.nome_nave
WHERE P.data_partenza BETWEEN 1-1-2009 AND 31-12-2009
GROUP BY N.nome_nave
HAVING SUM(P.numero_passeggeri) >= ALL(
    SELECT SUM(P1.numero_passeggeri)
    FROM Nave N1
    JOIN Prenotazione P1 ON N1.nome_nave = P1.nome_nave
    WHERE P1.data_partenza BETWEEN 1-1-2009 AND 31-12-2009
)
```

Esercizio 7

Si consideri il seguente schema che rappresenta una base di dati impiegata in una catena di librerie distribuite su tutto il territorio nazionale. In particolare oltre ai dati sulle singole librerie, sono archiviati i dati del personale e dei libri in vendita. Si osservi che possono esistere diverse versioni dello stesso libro (autore e titolo), edite da case editrici diverse. Infine attraverso la relazione Catalogo è possibile conoscere in quali librerie una determinata versione di un libro sia disponibile per essere venduta, il numero di copie a disposizione, ed il costo.

```
LIBRERIA(idLibreria, nome, indirizzo, città, oraApertura, oraChiusura,
turnoChiusura)
PERSONALE(idPersona, cognome, nome, idLibreria, reparto, turno)
CATALOGO(idLibro, idLibreria, reparto, numerocopie, costo)
LIBRO(idLibro, titolo, autori, casaEditrice, annoPubblicazione)
```

1

Il nome e le città delle librerie dove siano presenti almeno 10 copie del libro "Delitto e castigo", tenendo conto che possono esistere più versioni del libro poiché stampato da più case editrici

```
SELECT L.nome, L.città
FROM Libreria L
JOIN Catalogo C ON L.idLibreria = C.idLibreria
JOIN Libro LI ON C.idLibro = LI.idLibro
WHERE LI.nome = "Delitto e castigo"
GROUP BY (L.nome, L.città)
HAVING SUM(C.numerocopie) >= 10
```

2

La libreria di Milano con il numero maggiore di libri in vendita (contando quindi tutte le copie di ogni libro)

```
SELECT DISTINCT L.idLibreria, L.nome
FROM Libreria L
JOIN Catalogo C ON L.idLibreria = C.idLibreria
WHERE L.città = "Milano"
GROUP BY (L.idLibreria, L.nome)
HAVING SUM(C.numerocopie) >= ALL(
    SELECT SUM(C2.numerocopie)
    FROM Libreria L2
    JOIN Catalogo C2 ON L2.idLibreria = C2.idLibreria
    WHERE L2.città = "Milano"
    GROUP BY (L2.idLibreria)
```

