

Esercitazione 2

Esercizio 1

CORSO (CodCorso, NomeC, Anno, Semestre)

ORARIO-LEZIONI (CodCorso, GiornoSettimana, OraInizio, OraFine, Aula)

- a) Trovare le aule in cui non si tengono mai lezioni di corsi del primo anno.
- b) Trovare le aule in cui si tengono solo lezioni del primo anno.
- c) Trovare codice corso, nome corso e numero totale di ore di lezione settimanali per i corsi del terzo anno per cui il numero complessivo di ore di lezione settimanali è superiore a 10 e le lezioni sono in più di tre giorni diversi della settimana.

Esercizio 1.a Soluzione (NOT IN)

CORSO (CodCorso, NomeC, Anno, Semestre)

ORARIO-LEZIONI (CodCorso, GiornoSettimana, OraInizio, OraFine, Aula)

a) Trovare le aule in cui non si tengono mai lezioni di corsi del primo anno.

```
SELECT DISTINCT Aula
FROM ORARIO-LEZIONI
WHERE Aula NOT IN
(
    SELECT Aula
    FROM ORARIO_LEZIONI OL, CORSO C
    WHERE OL.CodCorso=C.CodCorso AND C.Anno=1
);
```

Esercizio 1.a Soluzione (NOT EXIST)

CORSO (CodCorso, NomeC, Anno, Semestre)

ORARIO-LEZIONI (CodCorso, GiornoSettimana, OraInizio, OraFine, Aula)

a) Trovare le aule in cui non si tengono mai lezioni di corsi del primo anno.

```
SELECT DISTINCT Aula
FROM ORARIO-LEZIONI OL1
WHERE NOT EXISTS
(
    SELECT *
    FROM ORARIO-LEZIONI OL2, CORSO C
    WHERE OL2.CodCorso=C.CodCorso AND C.Anno=1
    AND OL2.Aula=OL1.Aula
);
```

Esercizio 1.b Soluzione (NOT IN)

CORSO (CodCorso, NomeC, Anno, Semestre)

ORARIO-LEZIONI (CodCorso, GiornoSettimana, OraInizio, OraFine, Aula)

b) Trovare le aule in cui si tengono solo lezioni del primo anno.

```
SELECT DISTINCT Aula
FROM ORARIO-LEZIONI
WHERE Aula NOT IN
(
    SELECT Aula
    FROM ORARIO_LEZIONI OL, CORSO C
    WHERE OL.CodCorso=C.CodCorso AND C.Anno<>1
);
```

Esercizio 1.b Soluzione (NOT EXIST)

CORSO (CodCorso, NomeC, Anno, Semestre)

ORARIO-LEZIONI (CodCorso, GiornoSettimana, OraInizio, OraFine, Aula)

b) Trovare le aule in cui si tengono solo lezioni del primo anno.

```
SELECT DISTINCT Aula
FROM ORARIO-LEZIONI OL1
WHERE NOT EXISTS
(
    SELECT *
    FROM ORARIO-LEZIONI OL2, CORSO C
    WHERE OL2.CodCorso=C.CodCorso AND C.Anno<>1
    AND OL2.Aula=OL1.Aula
);
```

Esercizio 1.c Soluzione

CORSO (CodCorso, NomeC, Anno, Semestre)

ORARIO-LEZIONI (CodCorso, GiornoSettimana, Orainizio, OraFine, Aula)

- c) Trovare codice corso, nome corso e numero totale di ore di lezione settimanali per i corsi del terzo anno per cui il numero complessivo di ore di lezione settimanali è superiore a 10 e le lezioni sono in più di tre giorni diversi della settimana.

```
SELECT C.CodCorso, C.NomeC, SUM(OraFine-Orainizio)
FROM CORSO C, ORARIO-LEZIONI OL
WHERE C.CodCorso=OL.CodCorso AND C.Anno = 3
GROUP BY C.Corso, C.NomeC
HAVING SUM(OraFine-Orainizio)>10
AND COUNT(DISTINCT GiornoSettimana)>3;
```

Esercizio 2

ALLOGGIO (CodA, Indirizzo, Città, Superficie, CostoAffittoMensile)

CONTRATTO-AFFITTO (CodC, DataInizio, DataFine, CodFisc, CodA)

PERSONA(CodFisc, NomePersona)

N.B. Superficie espressa in metri quadri. Per i contratti in corso, DataFine è NULL.

- a) Trovare, per le città in cui sono stati stipulati almeno 100 contratti, la città il costo mensile massimo degli affitti, il costo mensile medio degli affitti, la durata massima dei contratti, la durata media dei contratti e il numero totale di contratti stipulati.
- b) Trovare il nome delle persone che non hanno mai affittato alloggi con superficie superiore a 80 metri quadri. Se ci sono più persone con lo stesso nome visualizzare solo una volta il nome.

Esercizio 2.a Soluzione

ALLOGGIO (CodA, Indirizzo, Città, Superficie, CostoAffittoMensile)

CONTRATTO-AFFITTO (CodC, DataInizio, DataFine, NomePersona, CodA)

PERSONA(CodFisc, NomePersona)

N.B. Superficie espressa in metri quadri. Per i contratti in corso, DataFine è NULL.

- a) Trovare, per le città in cui sono stati stipulati almeno 100 contratti, la città, il costo mensile massimo degli affitti, il costo mensile medio degli affitti, la durata massima dei contratti, la durata media dei contratti e il numero totale di contratti stipulati.

```
SELECT Citta, MAX(CostoAffittoMensile), AVG(CostoAffittoMensile), MAX(DataFine-
DataInizio), AVG(DataFine-DataInizio), COUNT(*)
FROM ALLOGGIO A, CONTRATTO-AFFITTO C
WHERE A.CodA=C.CodA
GROUP BY Citta
HAVING COUNT(*)>=100;
```

Esercizio 2.b Soluzione

ALLOGGIO (CodA, Indirizzo, Città, Superficie, CostoAffittoMensile)

CONTRATTO-AFFITTO (CodC, DataInizio, DataFine, CodFisc, CodA)

PERSONA(CodFisc, NomePersona)

N.B. Superficie espressa in metri quadri. Per i contratti in corso, DataFine è NULL.

- b) Trovare il nome delle persone che non hanno mai affittato alloggi con superficie superiore a 80 metri quadri. Se ci sono più persone con lo stesso nome visualizzare solo una volta il nome.

```
SELECT DISTINCT NomePersona
FROM PERSONA
WHERE CodFisc NOT IN
(
    SELECT CodFisc
    FROM CONTRATTO-AFFITTO C, ALLOGGIO A
    WHERE C.CodA=A.CodA
    AND Superficie>80
);
```

Esercizio 3

ORCHESTRA(CodO, NomeO, NomeDirettore, numElementi)

CONCERTI(CodC, Data, CodO, CodS, PrezzoBiglietto)

SALE(CodS, NomeS, Città, Capienza)

N.B.: PrezzoBiglietto è espresso in euro

- a) Trovare il codice e il nome delle orchestre con più di 30 elementi che hanno tenuto concerti sia a Torino, sia a Milano e non hanno mai tenuto concerti a Bologna.
- b) Mostrare il codice ed il nome delle orchestre che il 1 Gennaio 2017 si sono esibite nelle sale in cui il prezzo medio dei biglietti è maggiore di 100 euro.

Esercizio 3.a Soluzione

ORCHESTRA(CodO, NomeO, NomeDirettore, NumElementi)

CONCERTI(CodC, Data, CodO, CodS, PrezzoBiglietto)

SALE(CodS, NomeS, Città, Capienza)

N.B.: PrezzoBiglietto è espresso in euro

- a) Trovare il codice e il nome delle orchestre con più di 30 elementi che hanno tenuto concerti sia a Torino, sia a Milano e non hanno mai tenuto concerti a Bologna.

```
SELECT O.CodO, O.NomeO FROM ORCHESTRA O
WHERE O.NumElementi>30
AND O.CodO IN (
    SELECT C1.CodO
    FROM CONCERTI C1 WHERE Città = 'Torino' )
AND O.CodO IN (
    SELECT C2.CodO
    FROM CONCERTI C1 WHERE Città = 'Milano')
AND O.CodO NOT IN (
    SELECT C3.CodO
    FROM CONCERTI C1 WHERE Città = 'Bologna' );
```

Esercizio 3.b Soluzione

ORCHESTRA(CodO, NomeO, NomeDirettore, NumElementi)

CONCERTI(CodC, Data, CodO, CodS, PrezzoBiglietto)

SALE(CodS, NomeS, Città, Capienza)

N.B.: PrezzoBiglietto è espresso in euro

- b) Mostrare il codice ed il nome del direttore delle orchestre che il 1 Gennaio 2017 si sono esibite nelle sale in cui il prezzo medio dei biglietti è maggiore di 50 euro.

```
SELECT CodO, NomeDirettore
```

```
FROM ORCHESTRA O, CONCERTI C
```

```
WHERE O.CodO=C.CodO AND Data='1/1/2017' AND CodS IN (
```

```
    SELECT CodS
```

```
    FROM CONCERTI
```

```
    GROUP BY CodS
```

```
    HAVING AVG(PrezzoBiglietto) > 50
```

```
)
```

Esercizio 4

PERSONA(Nome, Sesso, Età)

GENITORE(Nome-Gen, Nome-Figlio)

Trovare il nome di tutte le persone con età inferiore a 10 anni che sono figli unici.

Esercizio 5

CLIENTE(Cod-Cli, Nome)

CONTO(Cod-Conto, Saldo, Agenzia, Stato)

CONTO-CLIENTE(Cod-Conto, Cod-Cli)

Trovare tutte le agenzie che hanno almeno un cliente titolare da solo (senza cointestatari) di un unico conto corrente (cliente a cui non è intestato nessun altro conto corrente)

Esercizio 4 Soluzione

PERSONA(Nome, Sesso, Età)

GENITORE(Nome-Gen, Nome-Figlio)

Trovare il nome di tutte le persone con età inferiore a 10 anni che sono figli unici.

```
SELECT DISTINCT NOME
```

```
FROM PERSONA P, GENITORE G
```

```
WHERE P.NOME=G.NOME-FIGLIO
```

```
AND P.ETA<10
```

```
AND P.NOME NOT IN
```

```
(SELECT G1.NOME-FIGLIO FROM GENITORE G1, GENITORE G2
```

```
WHERE G1.NOME-GEN=G2.NOME-GEN AND G1.NOME-FIGLIO<>G2.NOME-FIGLIO);
```

Esercizio 5 Soluzione

CLIENTE(Cod-Cli, Nome)

CONTO(Cod-Conto, Saldo, Agenzia, Stato)

CONTO-CLIENTE(Cod-Conto, Cod-Cli)

Trovare tutte le agenzie che hanno almeno un cliente titolare da solo (senza cointestatari) di un unico conto corrente (cliente a cui non è intestato nessun altro conto corrente)

```
SELECT DISTINCT C.Agenzia
```

```
FROM Conto C, CONTO-CLIENTE CL
```

```
WHERE C.COD-CONTO=CL.COD-CONTO
```

```
AND NOT EXISTS
```

```
(SELECT * FROM CONTO-CLIENTE CL2
```

```
WHERE CL2.DOC-CONTO=CL.COD-CONTO AND CL2.COD-CLI<>CL.COD-CLI)
```

```
AND NOT EXISTS
```

```
(SELECT * FROM CONTO-CLIENTE CL3
```

```
WHERE CL3.COD-CLI=CL.COD-CLI AND CL3.COD-CONTO<>CL.COD-CONTO);
```


Esercizio 6

QUIZ(CodQuiz, Argomento, Punteggio)

STUDENTE(Matricola, Nome, Indirizzo, Città)

RISULTATO_TEST(Matricola, CodQuiz, RispostaCorretta)

Il quiz è una sola domanda; RispostaCorretta può assumere valore 'si' (risposta corretta) o 'no' (risposta errata)

- a) Trovare il nome degli studenti che non hanno risposto correttamente a nessun quiz di matematica.
- b) Trovare il nome degli studenti di Torino che hanno conseguito il punteggio massimo possibile nei quiz di matematica.

Esercizio 6.a Soluzione

QUIZ(CodQuiz, Argomento, Punteggio)

STUDENTE(Matricola, Nome, Indirizzo, Città)

RISULTATO_TEST(Matricola, CodQuiz, RispostaCorretta)

- a) Trovare il nome degli studenti che non hanno risposto correttamente a nessun quiz di matematica.

```
SELECT Nome
```

```
FROM STUDENTE S
```

```
WHERE Matricola NOT IN (
```

```
    SELECT Matricola FROM RISULTATO_TEST R, QUIZ Q
```

```
    WHERE R.CodQuiz=Q.CodQuiz AND RispostaCorretta='si'
```

```
    AND Argomento='matematica'
```

```
);
```

Esercizio 6.b Soluzione Alternativa

QUIZ(CodQuiz, Argomento, Punteggio)

STUDENTE(Matricola, Nome, Indirizzo, Città)

RISULTATO_TEST(Matricola, CodQuiz, RispostaCorretta)

b) Trovare il nome degli studenti di Torino che hanno conseguito il punteggio massimo possibile nei quiz di matematica.

```
SELECT Nome
```

```
FROM STUDENTE S, RISULTATO_TEST R
```

```
WHERE S.Matricola=R.Matricola AND Citta='Torino' AND
```

```
RispostaCorretta='si' AND Argomento='matematica'
```

```
GROUP BY S.Matricola,Nome
```

```
HAVING COUNT(CodQuiz) = (
```

```
SELECT COUNT(CodQuiz)
```

```
FROM QUIZ
```

```
WHERE Argomento='matematica');
```

Esercizio 6.b Soluzione Alternativa (Attenzione! Richiede una tabella in più!)

QUIZ(CodQuiz, Argomento, Punteggio)

STUDENTE(Matricola, Nome, Indirizzo, Città)

RISULTATO_TEST(Matricola, CodQuiz, RispostaCorretta)

b) Trovare il nome degli studenti di Torino che hanno conseguito il punteggio massimo possibile nei quiz di matematica.

```
SELECT Nome
FROM STUDENTE S,RISULTATO_TEST R,QUIZ Q
WHERE S.Matricola=R.Matricola AND Q.CodQuiz=R.CodQuiz
AND Città='Torino' AND RispostaCorretta='si'
AND Argomento='matematica'
GROUP BY S.Matricola,Nome
HAVING SUM(Punteggio)=(SELECT SUM(Punteggio) FROM QUIZ
WHERE Argomento='matematica');
```

Esercizio 7

ALLOGGIO(CodA, Indirizzo, Città, Superficie, CostoAffittoMensile)

CONTRATTO-AFFITTO(CodC, DataInizio, DataFine, NomePersona, CodA)

Trovare il codice, l'indirizzo e la città degli alloggi che hanno una superficie superiore alla superficie media degli alloggi delle città in cui si trovano.

Esercizio 7 Soluzione

ALLOGGIO(CodA, Indirizzo, Città, Superficie, CostoAffittoMensile)

CONTRATTO-AFFITTO(CodC, DataInizio, DataFine, NomePersona, CodA)

Trovare il codice, l'indirizzo e la città degli alloggi che hanno una superficie alla superficie media degli alloggi delle città in cui si trovano.

```
SELECT CodA, Indirizzo, Citta FROM ALLOGGIO A1
WHERE Superficie > (
    SELECT AVG(Superficie) FROM ALLOGGIO A2
    WHERE A2.Citta=A1.Citta
);
```