

(A) CONOSCENZA TERMINOLOGICA

Dare una breve descrizione dei termini introdotti

- Clausola **LEFT JOIN**
- Clausola **RIGHT JOIN**
- Clausola **FULL JOIN**

(B) CONOSCENZA E COMPETENZA

Rispondere alle seguenti domande producendo anche qualche esempio

B1) Conoscenze

1. Cosa vuol dire *interrogazioni composte*?
2. Quale operazione svolge l'operatore *congiunzione*?

B2) Competenze

1. Qual è la sintassi per realizzare la *congiunzione* fra tabelle?
2. Qual è la clausola che indica il *prodotto cartesiano*?
3. Qual è la clausola che indica la *selezione*?
4. Qual è la clausola che indica la *proiezione*?

(C) ESERCIZI DI COMPrensIONE

1. La sintassi della congiunzione è:
SELECT a FROM b [WHERE c];
dove *a* indica una lista di da visualizzare, *b* indica la lista delle da porre in congiunzione e *c* indica un'eventuale di selezione.
2. La sintassi seguente:
SELECT lista_attributi FROM tab1 JOIN tab2 ON condizione;
indica il *join*.....
3. Consideriamo i dati presenti nelle tabelle *Clienti* e *Dipendenti* riportate di seguito:

IDCliente	Cognome	Nome	Indirizzo	Città	Paese	Telefono	Agente
1	Rossi	Mario	v. Padova, 67	Roma	Italia	06/566756	1
2	Bianchi	Piero	v. Alfieri, 10	Milano	Italia	02/744192	2
3	Smith	Davide	v. Del Corso	Roma	Italia	06/447643	1
4	Rossi	Fabio	v. Roma, 1/a	Prato	Italia	050/43324	3
5	Neri	Renzo	v. Po, 56	Milano	Italia	02/4354434	2
6	Verdi	Enza	v. Pazzi, 17	Roma	Italia	06/545432	1
7	Paneri	Riccardo	v. del Grano, 34	Roma	Italia	06/767654	1
8	Manzi	Elisa	L.go Aironi, 12	Firenze	Italia	055/34356	3

IDDipendente	Cognome	Nome	Interno	Telefono	Zona
1	Fermani	Jolanda	1	06-233445	Roma
2	Rinaldi	Rosanna	2	02-229707	Milano
3	Di Bello	Mario	3	055-765857	Firenze
4	De Giovanni	Emilio	4	081-567568	Napoli
5	Simoni	Gabriella	5	050-78656	Bologna
6	Grandi	Elio	6	06-657657	Roma

Scrivere i comandi SQL per realizzare le seguenti richieste:

- a. elencare cognome e nome di tutti i clienti, con a fianco il cognome dell'agente rappresentante:
- b. elencare cognome e nome dei clienti residenti a Roma, con a fianco il cognome dell'agente, nominando *Agente* la sua colonna:

- c. elencare cognome e nome dei dipendenti, riportando a fianco il cognome del cliente seguito da ciascuno di essi (rinominato *Cliente*):

- d. fornire lo stesso elenco della domanda precedente, ma elencando anche eventuali agenti che non seguono nessun cliente (come ad esempio avviene per `IDDipendente=6`):

4. Dati gli schemi:
Pacchetti (*IDPacchetto*, *DataPartenza*, *DataRientro*, *Descrizione*, *Prezzo*, *CodNaz*)
Nazioni (*Codice*, *Denominazione*)
 scrivere in SQL le interrogazioni per ottenere:
- l'elenco *IDPacchetto*, *Descrizione* e *Prezzo* di tutti i pacchetti viaggio aventi destinazione "Spagna";
 - l'elenco di tutti i viaggi (*IDPacchetto*, *DataPartenza*, *DataRientro*) verso una destinazione X, con prezzo compreso in un dato intervallo (Y, Z);
5. Dati gli schemi:
Fornitori (*IDFornitore*, *Nome*, *Indirizzo*, *Città*)
Prodotti (*IDProdotto*, *Nome*, *Marca*, *Modello*)
Catalogo (*IDFornitore*, *IDProdotto*, *Costo*)
 scrivere in SQL le interrogazioni per ottenere:
- l'elenco *Nome*, *Marca* e *Modello* dei prodotti aventi costo minore di 2000 euro

```
[R. SELECT prodotti.IDProdotto, Nome, Marca, Modello, Costo
FROM listino JOIN prodotti WHERE costo<2000 AND
(IDFornitore=listino.IDFornitore) AND
(listino.IDProdotto =listino.IDProdotto) ]
```
 - l'elenco dei nomi dei fornitori che distribuiscono prodotti di marca "IBM"

```
[R. SELECT DISTINCT fornitori.Nome, Marca FROM prodotti
JOIN (fornitori JOIN listino) WHERE Marca='IBM' ]
```
 - l'elenco dei codici di tutti i prodotti che sono forniti almeno da due fornitori;
 - i nomi dei fornitori che distribuiscono tutti i prodotti presenti nel catalogo.

(D) ESERCIZI DI APPLICAZIONE

1. **Esercizio risolto** Dati gli schemi
Maternità (*Madre*, *Figlio*)
Paternità (*Padre*, *Figlio*)
Persone (*Nome*, *Età*, *Reddito*)
- Elencare i padri delle persone che guadagnano più di 10000.
 In algebra, l'interrogazione si esprime con:
PROJ *Padre* (**SEL** *Reddito* > 1000 **JOIN** *Figlio*=*Nome* *Paternità*);
 In SQL abbiamo quindi:
SELECT DISTINCT Padre FROM Persone, Paternità
WHERE Figlio=Nome AND Reddito > 1000;
 - Elencare i figli con i rispettivi genitori.
 In algebra, il comando si esprime con:
PROJ *Figlio, Padre, Madre* (*Maternità JOIN Paternità*)
 che in SQL diventa:
SELECT Paternità.Figlio, Padre, Madre FROM Maternità, Paternità
WHERE Paternità.Figlio = Maternità.Figlio;
 - Elencare i nomi delle persone di età maggiore di 50 anni in ordine alfabetico:
SELECT Nome, Reddito FROM Persone
WHERE Età > 50 ORDER BY Nome;
2. Dati gli schemi
Studenti (*Matricola*, *Cognome*, *Nome*, *Sede*)
Esami (*IDesame*, *IDStudente*, *Voto*, *Data*)
 scrivere in SQL le interrogazioni corrispondenti alle seguenti richieste:
- Elencare gli *IDesame* di una data giornata;
 - Elencare *Cognome* e *Nome* degli studenti esaminati in una data giornata;
 - Elencare *Cognome* e *Nome* di tutti gli studenti che sono stati respinti in un dato giorno.

(E) ESERCITAZIONI PRATICHE

1. **Gestione medici e pazienti.**
- Scrivere i comando SQL per creare i seguenti schemi:
Pazienti (*IDPaziente*, *Cognome*, *Nome*, *Indirizzo*, *Telefono*, *IDMedico*)
Medici (*IDMedico*, *Cognome*, *Nome*, *Indirizzo*, *Telefono*, *Specialità*)
 - Popolare le tabelle con i seguenti dati::

IDPaziente	Cognome	Nome	Indirizzo	Telefono	IDMedico
1	Bianchi	Ennio	via Roma 1	56476745	100
2	Pandolfi	Simone	via Milano 16	76567767	105
3	Pacifici	Eda	l.go Paci 4	65565756	100
4	Arduini	Ada	c.so Vittorio	86796866	103

IDMedico	Cognome	Nome	Indirizzo	Telefono	Specialità
100	Simoni	Edoardo	via Milano 35	6576545	otorino
101	Carli	Lucio	l.go Aironi 6	5675657	odontoiatra
102	Balzani	Giuseppe	v.le Meduse 48	8789789	neurologo
103	Peri	Antonio	c.so Garibaldi	5685787	otorino

- c. Svolgere le seguenti interrogazioni:
- elencare *Cognome, Nome, Indirizzo e Telefono* dei pazienti del medico X;
 - elencare *Cognome, Nome, Indirizzo e Telefono* del medico che segue il paziente X;
 - trovare il numero di telefono del medico che segue il paziente X.
2. Un'agenzia immobiliare vuole realizzare una applicazione software per la realizzazione di un archivio delle case disponibili. Per ogni casa sono disponibili le seguenti informazioni:
- la zona;
 - il costo;
 - la tipologia dell'abitazione
 - superficie in metri quadrati;
 - l'anno di costruzione;
 - una breve descrizione di non più di 100 parole;
 - i dati del proprietario;
 - i dati dei potenziali acquirenti per ogni appartamento
 - eventualmente una fotografia che non necessariamente esiste per tutte le case.

Dopo aver sviluppato l'analisi del problema e il relativo diagramma E-R, realizzare il progetto logico mediante costruzione delle relative tabelle. Successivamente, formulare in SQL le opportune istruzioni per:

- introdurre le informazioni relative a una nuova casa
- cancellare le informazioni di quelle di una casa non più disponibile,;
- ricercare una casa in archivio in base a uno o più dei seguenti elementi di ricerca: fascia di prezzo, superficie, zona, tipologia;
- elencare i dati di tutti i potenziali clienti di una data abitazione
- elencare i dati dei proprietari di case appartenenti ad una data zona
- elencare i dati dei proprietari di case appartenenti ad una data tipologia