

Parte 1: Interrogazioni di Base

Esercizio 1.1: Database Biblioteca

Data la tabella:

```
LIBRI (Codice, Titolo, Autore, AnnoPubblicazione, Editore, Genere,
      NumPagine, Prezzo)
```

Scrivere le seguenti query SQL:

1. Elencare tutti i libri pubblicati dopo il 2020.
2. Visualizzare titolo e autore dei libri con prezzo superiore a 25 euro.
3. Contare quanti libri di genere "Fantasy" sono presenti nella biblioteca.
4. Elencare tutti i libri ordinati per autore e, a parità di autore, per titolo.
5. Trovare il prezzo medio dei libri per ogni editore.

Esercizio 1.2: Database Università

Date le seguenti tabelle:

```
STUDENTI (Matricola, Nome, Cognome, DataNascita, CorsoDiLaurea)
ESAMI (CodEsame, NomeEsame, CFU, DocenteResponsabile)
VOTI (Matricola*, CodEsame*, Data, Voto)
```

Scrivere le seguenti query SQL:

1. Elencare tutti gli studenti iscritti al corso di laurea "Informatica".
2. Trovare il voto medio ottenuto in ogni esame.
3. Visualizzare nome e cognome degli studenti che hanno superato l'esame di "Basi di Dati" con un voto superiore a 27.
4. Contare quanti esami ha sostenuto ogni studente.
5. Trovare gli studenti che hanno sostenuto tutti gli esami previsti.

Parte 2: Join e Subquery

Esercizio 2.1: Database E-commerce

Date le seguenti tabelle:

```
CLIENTI (CodCliente, Nome, Cognome, Email, Indirizzo, Città, CAP)
ORDINI (NumOrdine, CodCliente*, Data, Stato, ImportoTotale)
DETTAGLI_ORDINE (NumOrdine*, CodProdotto*, Quantità, PrezzoUnitario)
PRODOTTI (CodProdotto, Nome, Descrizione, Categoria, Prezzo, Giacenza)
```

Scrivere le seguenti query SQL:

1. Elencare tutti i clienti che hanno effettuato almeno un ordine nell'ultimo mese.
2. Trovare i prodotti più venduti (in termini di quantità).
3. Visualizzare per ogni cliente il numero di ordini effettuati e l'importo totale speso.
4. Identificare i clienti che hanno acquistato il prodotto più costoso.
5. Trovare i prodotti mai ordinati.

Esercizio 2.2: Database Cinema

Date le seguenti tabelle:

```
FILM (CodFilm, Titolo, Regista, AnnoUscita, Genere, Durata)
ATTORI (CodAttore, Nome, Cognome, DataNascita, Nazionalità)
RECITA (CodFilm*, CodAttore*, Ruolo)
PROIEZIONI (CodProiezione, CodFilm*, DataOra, Sala, PrezzoIntero)
BIGLIETTI (CodBiglietto, CodProiezione*, TipoBiglietto, Prezzo)
```

Scrivere le seguenti query SQL:

1. Elencare tutti i film in cui ha recitato un determinato attore.
2. Trovare gli attori che hanno recitato in più di 3 film.
3. Visualizzare il film che ha generato il maggior incasso.
4. Contare il numero di biglietti venduti per ogni film.
5. Trovare gli attori che hanno recitato insieme in almeno un film.

Parte 3: Funzioni di Aggregazione e Raggruppamento

Esercizio 3.1: Database Azienda

Date le seguenti tabelle:

```
DIPENDENTI (CodDipendente, Nome, Cognome, DataAssunzione, Dipartimento, Stipendio)
PROGETTI (CodProgetto, Nome, DataInizio, DataFine, Budget)
ASSEGNAZIONI (CodDipendente*, CodProgetto*, OrePreviste, OreEffettive)
```

Scrivere le seguenti query SQL:

1. Calcolare lo stipendio medio per ogni dipartimento.
2. Trovare il dipendente con lo stipendio più alto in ogni dipartimento.
3. Visualizzare i progetti con più ore di lavoro effettive rispetto a quelle previste.
4. Contare quanti dipendenti lavorano su ciascun progetto.
5. Trovare i dipartimenti con più di 5 dipendenti assunti negli ultimi 2 anni.

Esercizio 3.2: Database Meteo

Data la tabella:

```
RILEVAZIONI (ID, Stazione, Data, Ora, Temperatura, Umidità, Pressione, VelocitàVento, Precipitazioni)
```

Scrivere le seguenti query SQL:

1. Calcolare la temperatura media giornaliera per ogni stazione.
2. Trovare la stazione con le precipitazioni massime in un determinato mese.
3. Visualizzare per ogni stazione il numero di giorni con temperatura superiore a 30°C.
4. Calcolare l'escursione termica (differenza tra temperatura massima e minima) giornaliera per ogni stazione.
5. Trovare le stazioni che hanno registrato almeno un giorno di "tempesta" (definita come precipitazioni > 50mm e velocità del vento > 80km/h).

Parte 4: Query Avanzate e Casi Complessi

Esercizio 4.1: Database Sanità

Date le seguenti tabelle:

```
PAZIENTI (CodPaziente, CodiceFiscale, Nome, Cognome, DataNascita, Indirizzo, Città)
MEDICI (CodMedico, Nome, Cognome, Specializzazione, AnniEsperienza)
VISITE (CodVisita, CodPaziente*, CodMedico*, DataOra, Motivo, Diagnosi, Costo)
FARMACI (CodFarmaco, Nome, PrincipioAttivo, Dosaggio, Prezzo)
PRESCRIZIONI (CodPrescrizione, CodVisita*, CodFarmaco*, Quantità, Posologia)
```

Scrivere le seguenti query SQL:

1. Trovare i pazienti che hanno effettuato più di 3 visite nell'ultimo anno.
2. Visualizzare il medico che ha prescritto più farmaci in totale.
3. Calcolare il costo medio delle visite per ogni specializzazione medica.

4. Identificare i pazienti che hanno ricevuto prescrizioni per almeno due farmaci con lo stesso principio attivo.
5. Trovare i farmaci mai prescritti.

Esercizio 4.2: Database Trasporti

Date le seguenti tabelle:

```
VEICOLI (Targa, Modello, Marca, AnnoImmatricolazione, Capacità, Consumo)
AUTISTI (CodAutista, Nome, Cognome, DataAssunzione, NumPatente, TipoPatente)
TRATTE (CodTratta, Partenza, Arrivo, Distanza, TempoStimato)
VIAGGI (CodViaggio, CodTratta*, Targa*, CodAutista*, DataPartenza,
OraPartenza, NumPasseggeri)
```

Scrivere le seguenti query SQL:

1. Calcolare la distanza totale percorsa da ogni veicolo.
2. Trovare le tratte più frequenti (in termini di numero di viaggi).
3. Visualizzare gli autisti che hanno guidato tutti i tipi di veicoli disponibili.
4. Identificare i viaggi in cui il numero di passeggeri ha superato l'80% della capacità del veicolo.
5. Trovare la tratta con il maggior ritardo medio rispetto al tempo stimato.

Parte 5: DDL e Gestione del Database

Esercizio 5.1: Creazione Database Scuola

Definire lo schema SQL per un database che gestisca una scuola con le seguenti entità:

- Studenti (con attributi: ID, Nome, Cognome, DataNascita, Classe)
- Insegnanti (con attributi: ID, Nome, Cognome, Materia, AnniServizio)
- Classi (con attributi: ID, Anno, Sezione, Indirizzo)
- Lezioni (con attributi: ID, IDInsegnante, IDClasse, Giorno, OraInizio, OraFine, Aula)
- Voti (con attributi: ID, IDStudente, IDInsegnante, Materia, Data, Valore, Tipo)

Definire le chiavi primarie, le chiavi esterne e i vincoli di integrità appropriati.

Esercizio 5.2: Modifica di uno Schema Esistente

Data la struttura seguente:

```
PRODOTTI (CodProdotto, Nome, Prezzo, Categoria)
MAGAZZINI (CodMagazzino, Nome, Indirizzo, Capienza)
```

GIACENZE (CodProdotto*, CodMagazzino*, Quantità)

Scrivere le istruzioni SQL per:

1. Aggiungere un attributo "Fornitore" alla tabella PRODOTTI.
2. Creare un indice sulla colonna "Categoria" della tabella PRODOTTI.
3. Aggiungere un vincolo di controllo in modo che il Prezzo non possa essere negativo.
4. Modificare la tabella GIACENZE in modo che la Quantità abbia un valore di default pari a 0.
5. Creare una vista che mostri, per ogni magazzino, il valore totale delle merci contenute (Quantità * Prezzo).