

1 Query semplici

1.1 Determinare nome, cognome e codice fiscale degli Impiegati il cui codice fiscale inizia per L e termina per M

```
SELECT nome, cognome, cf
FROM Impiegati
WHERE cf LIKE 'L%M';
```

1.2 Selezionare la Partita Iva dei Fornitori che pagano tramite 'bonifico 60 giorni'

```
SELECT piva
FROM Fornitore
WHERE pagamento='bonifico_60_giorni';
```

1.3 Selezionare nome e codice di tutti i Prodotti che presentano un prezzo minore o uguale di '4 euro'

```
SELECT nome, codice
FROM Prodotti
WHERE prezzo <= 4.00;
```

1.4 Selezionare il responsabile del reparto '5', con il nome di quest'ultimo

```
SELECT responsabile
FROM Reparti
WHERE id=5;
```

1.5 Conoscere i dati anagrafici degli Impiegati nati in provincia di Torino

```
SELECT *
FROM Impiegati
WHERE provinciaNascita='Torino';
```

1.6 Determinare la quantità e la scadenza dei Prodotti appartenenti al genere 'Alimentare'

```
SELECT nome, quantita, scadenza
FROM Prodotti
WHERE genere='Alimentare';
```

1.7 Selezionare nome e genere del Prodotto con codice 77

```
SELECT nome, genere
FROM Prodotti
WHERE codice=77;
```

1.8 Determinare l'indirizzo, la provincia ed il comune di nascita degli Impiegati nati tra ' e '

```
SELECT comuneNascita, provinciaNascita, indirizzo
FROM Impiegati
WHERE dataNascita between '1975-01-06' AND '1990-06-29';
```

2 Query con JOIN

2.1 Determinazione dei dati delle vendite nel giorno '2016-01-01'

```
SELECT s1.nScontrino, s1.cliente, a1.codiceProdotto,
a1.prezzo, a1.quantita
FROM Acquisti as a1 JOIN Scontrini as s1 on
a1.nScontrino=s1.nScontrino
WHERE s1.dataAcquisto='2016-01-01';
```

2.2 Trovare gli Impiegati (CF) che lavorano nel reparto 10

```
SELECT a1.impiegato , r1.nome
FROM Afferenza as a1 JOIN Reparti as r1 on a1.reparto=r1.id
WHERE r1.id=10;
```

2.3 Selezionare nome e codice prodotto di un Premio, al quale un cliente ha diritto con i suoi punti tessera

```
SELECT p1.id as codicePremio , p2.nome, p1.prodotto
FROM Clienti JOIN Premi as p1 JOIN Prodotti as p2
on p1.prodotto=p2.codice
WHERE Clienti.punti>=p1.puntinecessari
AND Clienti.tessera=10;
```

2.4 Determinare lo Stipendio di un responsabile del reparto 7

```
SELECT i1.nome, i1.cognome, r1.nome, s1.retribuzione
FROM Impiegati as i1 JOIN Reparti as r1
on i1.cf=r1.responsabile JOIN Stipendi as s1
on i1.livello=s1.livelloQualifica
WHERE r1.id=7;
```

2.5 Selezionare nome, cognome e punti ottenuti da un cliente che ha effettuato un acquisto nel giorno ”

```
SELECT c1.nome, c1.cognome
sum(p1.puntiOttenibili)*a1.quantita as puntiOttenuti ,
p1.prodotto , s1.dataAcquisto
FROM Scontrini as s1 JOIN Acquisti as a1
on s1.nScontrino=a1.nScontrino JOIN Clienti as c1
on s1.cliente=c1.cf JOIN ProdottiPunti as p1
on a1.codiceProdotto=p1.prodotto WHERE c1.tessera=10
AND DATE(s1.dataAcquisto)=DATE(NOW());
```

2.6 Determinare il reparto, nome del reparto, nome, cognome e codice fiscale del direttore del supermercato, sapendo che un direttore ha livello di qualifica pari a 7

```
SELECT a.reparto , r.nome, i.nome, i.cognome, i.cf
FROM Impiegati as i JOIN Afferenza as a
on i.cf=a.impiegato JOIN Reparti as r
on i.cf=r.responsabile
WHERE livello=7;
```

2.7 Determinare il nome ed il codice del prodotto composto 'Pasta al Forno' e delle sue componenti, con le relative quantità

```
SELECT p1.codice , p1.nome, c.componente, p2.nome, c.quantita
FROM Composizione as c JOIN Prodotti as p1
on c.composto=p1.codice JOIN Prodotti as p2
on c.componente=p2.codice WHERE p1.nome='Pasta_al_Forno';
```

2.8 Determinare il nome del responsabile del reparto che possiede il prodotto con il prezzo più alto

```
SELECT i1.nome, p1.nome, p1.nome, p1.prezzo
FROM Reparti as r1 JOIN Impiegati as i1
on r1.responsabile=i1.cf JOIN Prodotti as p1
on r1.id=p1.reparto order by p1.prezzo DESC LIMIT 1;
```

3 Query con operatori aggregati

3.1 Conteggia tutti i Prodotti riforniti da '07273008006'

```
SELECT count(*) as contatore
FROM Fornitura
WHERE fornitore=07273008006;
```

3.2 Seleziona il massimo costo di un prodotto presente nel reparto 12

```
SELECT max(prezzo) as prezzoMassimo
FROM Prodotti
WHERE reparto=12;
```

3.3 Determinare la media degli Stipendi di tutti gli Impiegati

```
SELECT avg(s1.retribuzione) as mediaRetribuzione
FROM Impiegati as i1 JOIN Stipendi as s1
on i1.livello=s1.livelloQualifica;
```

3.4 Determinare il prezzo più basso tra i Prodotti forniti che appartengono al reparto 4

```
SELECT min(p1.prezzo)
FROM Prodotti as p1 JOIN Fornitura as f1
on p1.codice=f1.prodotto JOIN Reparti as r1
on r1.id=p1.reparto
WHERE r1.id=4;
```

4 Query con l'uso di viste

4.1 Determinazione del prodotto più venduto nel reparto 10

```
CREATE VIEW Vista(count, prodotto) as
SELECT count(distinct a1.codiceProdotto)*sum(a1.quantita),
p1.codice
FROM Acquisti as a1 JOIN Scontrini as s1
on a1.nScontrino=s1.nScontrino JOIN Prodotti as p1
on a1.codiceProdotto=p1.codice
WHERE p1.reparto=10
GROUP BY a1.codiceProdotto;
```

```
SELECT v.count, v.prodotto, p1.nome, p1.reparto
FROM Vista as v JOIN Prodotti as p1
on v.prodotto=p1.codice order by p1.prezzo DESC LIMIT 1;
```

4.2 Determinare in quale reparto

```
CREATE VIEW Vista(count, reparto) as
SELECT count(*), r1.id
FROM acquisti as a1 join prodotti as p1
on a1.codiceProdotto = p1.codice join reparti as r1
on p1.reparto = r1.id join scontrini as s1
on a1.nScontrino = s1.nScontrino
WHERE date(a1.dataAcquisto) = '2016-05-11'
GROUP BY r1.id;
```

```
SELECT count as contatore, reparto
FROM Vista
ORDER BY contatore DESC LIMIT 1;
```

5 Query annidate

5.1 Determinare gli Impiegati con nome e cognome differente che possiedono la stessa provincia di nascita

```
SELECT nome, cognome, il.provinciaNascita
FROM Impiegati as i1
WHERE EXISTS( SELECT nome, cognome FROM Impiegati as i2
WHERE i1.nome <> i2.nome AND i1.cognome <> i2.cognome
AND i1.provinciaNascita=i2.provinciaNascita)
order by provinciaNascita;
```

5.2 Selezionare i componenti e la quantità necessaria per creare la 'Pizza Popeye' in laboratorio

```
SELECT p1.nome, c1.componente, c1.quantita
FROM Composizione as c1 JOIN Prodotti as p1
on c1.composto=p1.codice
WHERE c1.composto = ANY (SELECT codice FROM Prodotti
WHERE nome='Pizza_Popeye');
```

5.3 Determinare gli Impiegati con nome, reparto e livello che non sono responsabili in nessun reparto

```
SELECT i1.nome, a1.reparto, i1.livello
FROM Impiegati as i1 JOIN Afferenza as a1
on i1.cf=a1.impiegato
WHERE NOT EXISTS
(SELECT responsabile FROM Reparti WHERE responsabile=i1.cf);
```

5.4 Determinare i dati degli Impiegati(compreso di livello) con livello più basso tra gli Impiegati

```
SELECT cf, nome, cognome, livello
FROM Impiegati
WHERE livello <= ALL(SELECT livelloQualifica FROM Stipendi);
```

5.5 Determinare il nome, cognome, codice fiscale del responsabile del reparto che possiede il prodotto “composto” con componente 'farina grano duro' con il prezzo più alto

```
SELECT cf, nome, cognome FROM Impiegati
WHERE cf = (SELECT responsabile FROM reparti
WHERE id = (SELECT reparto FROM prodotti
WHERE codice in (SELECT composto FROM composizione
WHERE componente = (SELECT codice FROM prodotti
WHERE nome='farina_grano_duro')) AND
prezzo=(SELECT max(prezzo)
FROM prodotti WHERE codice in (SELECT composto
FROM composizione
WHERE componente=(SELECT codice FROM prodotti
WHERE nome='farina_grano_duro'))))));
```

6 Query con operatori insiemistici

6.1 Determinare codice fiscale, nome, cognome di tutti gli Impiegati e l'id, nome e responsabile di tutti i Reparti

```
SELECT cf as campo1, nome as campo2, cognome as campo3
FROM Impiegati
UNION
SELECT impiegato, reparto, dataAssunzione
FROM Afferenza
ORDER BY campo1;
```


7 Procedure

7.1 Aggiorna responsabile di un reparto

call AggiornaResponsabile(vecchioResponsabile, nuovoResponsabile, reparto);

La procedure permette di aggiornare il responsabile di un reparto, partendo dalla conoscenza del vecchio.

7.2 Aggiungi un acquisto

call AggiungiAcquisto(codiceProdotto, quantita, tesseraCliente, modalita);

La procedure permette di creare un nuovo acquisto generando un nuovo scontrino (modalità 'n'), oppure aggiungere un nuovo prodotto ad uno scontrino generato precedentemente (modalità 'o').

Il parametro 'tesseraCliente', passato alla funzione, può essere NULL se il cliente non è tesserato. In questo modo, ogni acquisto è comunque registrato.

Inoltre, se il prodotto acquistato prevede ottenimento di punti, questi sono caricati sulla tessera del cliente.

Infine, è decrementata la quantità di prodotto disponibile.

7.3 Modifica appartenenza di un impiegato

call ModificaAfferenza(cfImpiegato, nuovoReparto);

La procedure permette di modificare il reparto di appartenenza di un impiegato.

7.4 Aggiungi un prodotto fornito

call AggiungiProdotto(nomeProdotto, reparto, genere, prezzo, scadenza, fornitore);

La procedure permette di aggiungere un nuovo prodotto fornito alla tabella Prodotti e successivamente alla tabella Fornitura, tenendo conto del fornitore e del particolare prezzo di fornitura.

7.5 Ritiro di un premio

call RitiraPremio(codicePremio, cliente);

La procedure permette di ritirare un premio, specificando il suo codice ed il cliente, sottraendo i punti necessari dalla sua tessera; contemporaneamente la quantità di prodotto disponibile è decrementata.