

# Sistemi Informativi

Sistemi Informativi .....	1
Esercizi .....	1
.1 Gara di sci .....	1
.2 Filiali.....	1
.3 Filiali (SQL - risposta aperta).....	1
.4 Filiali (AR - risposta multipla) .....	2
.5 Filiali (SQL - risposta aperta).....	2
.6 Filiali Bis (SQL - risposta aperta).....	3
.7 Dischi.....	3
.8 Dischi: cantautori.....	3
.9 Dischi: anno di registrazione brani .....	3
.10 Dischi: stesso titolo .....	3
.11 Dischi: autori non cantanti.....	3
.12 Dischi: autori e cantanti "puri" .....	4
.13 Dischi: elenco brani .....	4
.14 Dischi: elenco brani e cantanti.....	4
.15 Dischi: coautori di Mogol.....	4
.16 Dischi: numero brani di ogni album.....	4
.17 Domande teoriche varie.....	4
Soluzioni .....	5

## Esercizi

### .1 Gara di sci

Descrivere in linguaggio naturale le informazioni organizzate nel seguente schema di base di dati:

**Fondista**(Nome, Nazione, Età)

**Gareggia**(NomeFondista, NomeGara, Piazzamento)

**Gara**(Nome, Luogo, Nazione, Lunghezza)

Individuare le chiavi primarie, i vincoli di integrità referenziale, gli attributi sui quali è sensato ammettere valori nulli.

### .2 Filiali

Si sintetizzi uno schema di base di dati che contenga le informazioni necessarie alla gestione di un insieme di dipendenti, con codice fiscale, cognome, nome, data di nascita, luogo di nascita, indirizzo, filiale di appartenenza, e stipendio. Inoltre per ogni filiale c'è un solo direttore, e una città in cui è sita. Si vuole poter rimuovere un dipendente dalla banca dati (cancellando in suo stipendio e la sua appartenenza ad una filiale) senza che i dati sulla persona vengano cancellati. Si determinino le chiavi primarie e i vincoli di integrità referenziale.

### .3 Filiali (SQL - risposta aperta)

Considerare uno schema di base di dati così concepito:

**Persone**(CodiceFiscale, Cognome, Nome, DataDiNascita, LuogoDiNascita, Indirizzo)

**Dipendenti**(CodiceFiscale, Filiale, Stipendio)

con vincoli di integrità referenziale:

fra l'attributo Filiale e la relazione **Filiali**

fra l'attributo CodiceFiscale e la relazione **Persone**

**Filiali**(Codice, Città, Direttore)

con vincolo di integrità referenziale:

fra l'attributo Direttore e la relazione **Dipendenti**

Formulare in linguaggio SQL una interrogazione che trova i dipendenti che guadagnano di più dei loro direttori

#### .4 Filiali (AR - risposta multipla)

Con riferimento allo schema di base di dati dell'esercizio .3, quale (o quali) delle seguenti interrogazioni in algebra relazionale trova i cognomi dei dipendenti che guadagnano di più dei loro direttori?

- 1)  $A = \rho_{CF \leftarrow \text{CodiceFiscale}, C \leftarrow \text{Cognome}, N \leftarrow \text{Nome}, D \leftarrow \text{DataDiNascita}, L \leftarrow \text{LuogoDiNascita}, I \leftarrow \text{Indirizzo}} (\text{Persone})$   
 $B = \rho_{CF2 \leftarrow \text{CodiceFiscale}, F \leftarrow \text{Filiale}, S \leftarrow \text{Stipendio}} (\text{Dipendenti})$   
 $C = \pi_{C,F,S} (A \bowtie_{CF=CF2} B)$   
 $D = C \bowtie_{F=\text{Codice}} \text{Filiali} \bowtie_{\text{Direttore}=\text{CodiceFiscale}} \text{Dipendenti} \bowtie_{\text{CodiceFiscale}=CF3} (\rho_{CF3 \leftarrow \text{CodiceFiscale}} (\text{Persone}))$   
 $\pi_C (D)$
- 2)  $A = \rho_{CF \leftarrow \text{CodiceFiscale}, C \leftarrow \text{Cognome}, N \leftarrow \text{Nome}, D \leftarrow \text{DataDiNascita}, L \leftarrow \text{LuogoDiNascita}, I \leftarrow \text{Indirizzo}} (\text{Persone})$   
 $B = \rho_{CF2 \leftarrow \text{CodiceFiscale}, F \leftarrow \text{Filiale}, S \leftarrow \text{Stipendio}} (\text{Dipendenti})$   
 $C = A \bowtie_{CF=CF2} B \bowtie_{F=\text{Codice}} \text{Filiali}$   
 $D = C \bowtie_{\text{Direttore}=\text{CodiceFiscale}, S > \text{Stipendio}} \text{Dipendenti}$   
 $\pi_C (D)$
- 3)  $A = \rho_{CF \leftarrow \text{CodiceFiscale}, C \leftarrow \text{Cognome}, N \leftarrow \text{Nome}, D \leftarrow \text{DataDiNascita}, L \leftarrow \text{LuogoDiNascita}, I \leftarrow \text{Indirizzo}} (\text{Persone})$   
 $B = \rho_{CF2 \leftarrow \text{CodiceFiscale}, F \leftarrow \text{Filiale}, S \leftarrow \text{Stipendio}} (\text{Dipendenti})$   
 $C = \pi_{C,S,\text{Direttore}} (A \bowtie_{CF=CF2} B \bowtie_{F=\text{Codice}} \text{Filiali})$   
 $D = C \bowtie_{\text{Direttore}=\text{CodiceFiscale}} \text{Dipendenti}$   
 $\pi_C (\sigma_{S > \text{Stipendio}} D)$

#### .5 Filiali (SQL - risposta aperta)

Con riferimento allo schema di base di dati dell'esercizio .3 quale (o quali) delle seguenti interrogazioni SQL trovano i direttori che guadagnano di più di tutti i loro dipendenti

- 1) 

```
SELECT Cognome
FROM Persona, Dipendente, Filiali
WHERE Dipendente.Stipendio > ALL (SELECT Stipendio
                                  FROM Dipendente
                                  WHERE Dipendente.Filiale = Filiali.Codice)
```
- 2) 

```
SELECT Cognome
FROM Persona, Dipendente, Filiali
WHERE Persona.CodiceFiscale = Dipendente.CodiceFiscale OR
      Dipendente.CodiceFiscale = Filiali.Direttore AND
      Dipendente.Stipendio > ANY (SELECT Stipendio
                                  FROM Dipendente
                                  WHERE Dipendente.Filiale = Filiali.Codice)
```
- 3) 

```
SELECT Cognome
FROM Persona, Dipendente, Filiali
WHERE Persona.CodiceFiscale = Dipendente.CodiceFiscale AND
      Dipendente.CodiceFiscale = Filiali.Direttore AND
      Dipendente.Stipendio > ALL (SELECT Stipendio
```

```
FROM Dipendente
WHERE D2.Filiale = Filiali.Codice AND
Dipendente.CodiceFiscale <> Persona.CodiceFiscale)
```

## .6 Filiali Bis (SQL - risposta aperta)

Con riferimento allo schema di base di dati dell'esercizio .3 formulare una interrogazione SQL che trovi i cognomi dei dipendenti che sono nati nella stessa città del loro direttore

## .7 Dischi

Descrivere in linguaggio naturale le informazioni organizzate nel seguente schema di base di dati:

**Esecuzione**(CodiceReg, TitoloCanz, Anno)  
**Disco**(NroSerie, TitoloAlbum, Anno, Prezzo)  
**Contiene**(NroSerieDisco, CodiceReg, NroProgr)  
**Autore**(Nome, TitoloCanzone)  
**Cantante**(NomeCantante, CodiceReg)

Individuare le chiavi primarie, i vincoli di integrità referenziale e gli attributi sui quali è sensato ammettere valori nulli. Discutere i seguenti punti:

1. Può un disco contenere due volte lo stesso brano? Quale variazione allo schema bisognerebbe fare per consentire questa eventualità?
2. E' ottimale la rappresentazione data per i nomi degli autori e dei cantanti? Quante volte occorre inserire il nome "Lucio Dalla" se nella base di dati è presente un suo album con 20 canzoni? Come si potrebbe migliorare lo schema?
3. Possono esistere due esecuzioni con lo stesso titolo? Può succedere che la prima esecuzione abbia un autore e la seconda ne abbia un altro?
4. Come si potrebbe imporre che il TitoloCanzone di Autore sia coerente con il TitoloCanz di Esecuzione?
5. Come si potrebbe semplificare lo schema se ogni esecuzione avesse sempre un solo interprete?

## .8 Dischi: cantautori

Con riferimento allo schema descritto nell'esercizio .7 formulare una interrogazione SQL che permetta di **determinare i cantautori** (persone che hanno scritto e cantato la stessa canzone) **il cui nome inizia per 'D'**.

## .9 Dischi: anno di registrazione brani

Con riferimento allo schema descritto nell'esercizio .7 formulare una interrogazione SQL che permetta di **determinare i titoli dei dischi che contengono canzoni di cui non si conosce l'anno di registrazione.**

## .10 Dischi: stesso titolo

Con riferimento allo schema descritto nell'esercizio .7 formulare una interrogazione SQL che permetta di **determinare i dischi che hanno lo stesso titolo di un brano all'interno.**

## .11 Dischi: autori non cantanti

Con riferimento allo schema descritto nell'esercizio .7 formulare una interrogazione SQL che permetta di **determinare gli autori che non hanno mai registrato una canzone.**

Si ricorda che gli operatori insiemistici in SQL sono UNION (unione), INTERSECT (intersezione) ed EXEPT (differenza).

## .12 Dischi: autori e cantanti "puri"

Con riferimento allo schema descritto nell'esercizio .7 formulare una interrogazione SQL che permetta di **determinare gli autori e i cantanti "puri", cioè gli autori che non hanno mai registrato una canzone e i cantanti che non ne hanno mai scritta una.**

## .13 Dischi: elenco brani

Con riferimento allo schema descritto nell'esercizio .7 formulare una interrogazione SQL che permetta di **determinare i brani del disco con numero di serie 1, nell'ordine in cui compaiono nel disco.**

## .14 Dischi: elenco brani e cantanti

Con riferimento allo schema descritto nell'esercizio .7 formulare una interrogazione SQL che permetta di **determinare i brani del disco con numero di serie 1, nell'ordine in cui compaiono nel disco, con indicazione degli interpreti per i brani che hanno associato un cantante.**

## .15 Dischi: coautori di Mogol

Con riferimento allo schema descritto nell'esercizio .7 formulare una interrogazione SQL che permetta di **determinare i coautori di Mogol.**

## .16 Dischi: numero brani di ogni album

Con riferimento allo schema descritto nell'esercizio .7 formulare una interrogazione SQL che permetta di **ottenere una lista dei numeri di serie degli album accompagnati dal numero di brani presenti nell'album e ordinati in maniera decrescente per numero brani.**

## .17 Domande teoriche varie

1. La differenza principale tra il modello relazionale dei dati e il modello reticolare
2. Perché si usano i due termini "sistema informativo" e "sistema informatico" e con quale significato?
3. La differenza principale tra una relazione in senso matematico e una relazione nel modello relazionale dei dati
4. Quando un vincolo di integrità è detto vincolo di tupla. Una chiave è un vincolo di tupla?
5. Che differenza c'è tra una chiave e una superchiave?
6. Chi è il DataBase Administrator? Cosa fa?
7. Che differenza c'è tra una chiave primaria e un'altra chiave della relazione?
8. Cosa vuol dire che il modello relazionale "non è posizionale"?
9. Cosa vuol dire che l'algebra relazionale è "procedurale" e non "dichiarativa"?
10. Una relazione può avere nomi di attributi uguali? (motivare la risposta)
11. Che cosa è un DBMS e come si può interagire con esso?
12. Cosa vuol dire che il modello relazionale usa riferimenti per valore? Qual'è la più immediata conseguenza di questa caratteristica?
13. Quando un vincolo di integrità è detto "di dominio"? Se ne fornisca un esempio.
14. Che differenza c'è fra l'equi-join e il theta-join
15. Perché sono stati introdotti i valori nulli? Da quali circostanze reali possono essere generati? Quanti tipi di valori nulli sono definiti nel modello relazionale?
16. Quando un join naturale è detto "prodotto cartesiano"
17. Che differenza c'è tra una istanza e uno schema di relazione
18. Cosa si intende per "indipendenza dei dati" quando si parla di schemi esterni (cioè "viste"), schemi logici e schemi fisici?

## Soluzioni

### .1 Gara di sci

La base di dati descrive le informazioni inerenti alla partecipazione di atleti a diverse gare di sci: la relazione **Fondista** specifica nome, nazione ed età di ogni partecipante; la relazione **Gareggia** descrive ogni partecipazione ad una gara di un atleta specificando il suo piazzamento; la relazione **Gara** specifica nome, luogo, nazione e lunghezza di ogni competizione.

Chiavi primarie: Nome per **Fondista**; NomeFondista e NomeGara per **Gareggia**; Nome per **Gara**.

Vincoli di integrità referenziale: tra l'attributo NomeFondista di **Gareggia** e la relazione **Fondista**; tra l'attributo NomeGara di **Gareggia** e la relazione **Gara**.

Valori nulli sicuramente ammissibili per Piazzamento di **Gareggia**.

### .2 Filiali

Un possibile schema è il seguente:

**Persone**(CodiceFiscale, Cognome, Nome, DataDiNascita, LuogoDiNascita, Indirizzo)

**Dipendenti**(CodiceFiscale, Filiale, Stipendio)

con vincoli di integrità referenziale:

fra l'attributo Filiale e la relazione **Filiali**

fra l'attributo CodiceFiscale e la relazione **Persone**

**Filiali**(Codice, Città, Direttore)

con vincolo di integrità referenziale:

fra l'attributo Direttore e la relazione **Dipendenti**

### .3 Filiali (SQL - risposta aperta)

Una soluzione può essere:

```
SELECT Cognome
FROM Persone, Dipendenti AS D1, Filiali, Dipendenti AS D2
WHERE Persone.CodiceFiscale = D1.CodiceFiscale AND
      D1.Filiale = Filiali.Codice AND
      Filiali.Direttore = D2.CodiceFiscale AND
      D1.Stipendio > D2.Stipendio
```

### .4 Filiali (AR - risposta multipla)

Le risposte 2) e 3) sono corrette. Nella 1) manca la condizione sugli stipendi e viene chiamata in causa immotivatamente per la seconda volta la relazione Persone

### .5 Filiali (SQL - risposta aperta)

Nella risposta 1) mancano (almeno) le condizioni di join. La risposta 2) non funziona perchè il direttore è un dipendente della filiale di cui è a capo. La risposta 3) è corretta.

### .6 Filiali Bis (SQL - risposta aperta)

Una soluzione può essere:

```
SELECT Persone.Cognome
FROM Persone, Dipendenti, Filiali, Persone AS P, Dipendenti as D
WHERE Persone.CodiceFiscale = Dipendenti.CodiceFiscale AND
      Dipendenti.Filiale = Filiali.Codice AND
      P.CodiceFiscale = Filiali.Direttore AND
      D.CodiceFiscale = P.CodiceFiscale AND
      Persone.LuogoDiNascita = P.LuogoDiNascita
```

## .7 Dischi

La base di dati descrive le informazioni inerenti a dei dischi, composti da un numero variabile di brani, di cui si vuole ricordare l'autore (o gli autori) e l'interprete (o gli interpreti).

La relazione **Esecuzione** descrive un singolo brano, specificandone il titolo e l'anno di registrazione. Diverse registrazioni dello stesso brano avranno diverse tuple in questa relazione. La chiave primaria è il CodiceReg.

La relazione **Disco** contiene le informazioni inerenti ad un album: il titolo, l'anno ed il prezzo. La chiave primaria è NroSerie. La relazione **Contiene** associa le esecuzioni agli album, specificando qual'è il numero progressivo della esecuzione nell'album. La chiave primaria è (NroSerieDisco, CodiceReg), ammesso che (come è plausibile) nessun album contenga due volte la stessa esecuzione. La relazione **Autore** associa ad ogni titolo di canzone il nome (o i nomi) degli autori. La chiave primaria è (Nome, TitoloCanzone). La relazione **Cantante** associa ad ogni esecuzione uno o più interpreti. La chiave primaria è (NomeCantante, CodiceReg).

**Esecuzione**(CodiceReg, TitoloCanz, Anno)

**Disco**(NroSerie, TitoloAlbum, Anno, Prezzo)

**Contiene**(NroSerieDisco, CodiceReg, NroProgr)

**Autore**(Nome, TitoloCanzone)

**Cantante**(NomeCantante, CodiceReg)

I vincoli di integrità referenziale sono tra:

- l'attributo NroSerieDisco di **Contiene** e la relazione **Disco**
- l'attributo CodiceReg di **Contiene** e la relazione **Esecuzione**
- l'attributo CodiceReg di **Cantante** e la relazione **Esecuzione**

Si noti che non è definibile un vincolo di integrità referenziale tra l'attributo TitoloCanzone di **Autore** e l'attributo TitoloCanz di **Esecuzione** (né in un verso né nell'altro) perché i due attributi non sono chiave primaria.

E' plausibilmente lecito ammettere valori nulli solo su Anno di **Esecuzione** e Prezzo di **Disco**.

Discussione dei punti suggeriti:

1. Come già detto la stessa esecuzione non può essere contenuta due volte nello stesso album. Se si volesse consentire ciò occorrerebbe includere NroProgr nella chiave primaria di **Contiene**.
2. La rappresentazione dei nomi degli autori e dei cantanti è ridondante: il nome "Lucio Dalla" dovrebbe essere inserito quaranta volte quando si inserisce un suo album di venti canzoni di cui è autore e interprete. Occorrerebbe introdurre una relazione **Persona** con un CodicePersona come chiave primaria e rimandare a questa relazione sia da **Autore** che da **Cantante**. Si dovrebbe imporre un vincolo di integrità referenziale tra l'attributo CodiceAutore di **Autore** e la relazione **Persona** e tra l'attributo CodiceCantante di **Cantante** e la relazione **Persona**.
3. Sì, il titolo va inserito due volte. No, gli autori sono collegati alle esecuzioni tramite il valore dell'attributo TitoloCanzone.
4. Occorrerebbe creare una nuova relazione **Branzo**, in cui compare un CodiceBranzo (chiave primaria) e un TitoloBranzo, e imporre un vincolo di integrità referenziale tra CodiceBranzo di **Esecuzione** e la relazione **Branzo** e tra CodiceBranzo di **Autore** e la relazione **Branzo**.
5. Si potrebbe rinunciare alla relazione **Cantante**, e specificare il nome dell'interprete direttamente nella tupla dell'esecuzione.

## .8 Dischi: cantautori

Una risposta corretta è la seguente:

```
SELECT DISTINCT Autore.Nome
FROM Esecuzione, Autore, Cantante
WHERE Esecuzione.TitoloCanz = Autore.TitoloCanzone AND
      Autore.Nome = Cantante.NomeCantante AND
      Autore.Nome like "D%"
```

## .9 Dischi: anno di registrazione brani

```
SELECT DISTINCT TitoloAlbum
```

```

FROM Disco, Esecuzione, Contiene
WHERE Disco.NroSerie = Contiene.NroSerieDisco AND
      Contiene.CodiceReg = Esecuzione.CodiceReg AND
      Esecuzione.Anno = NULL

```

**.10 Dischi: stesso titolo**

```

SELECT TitoloAlbum
FROM Disco, Esecuzione, Contiene
WHERE Disco.NroSerie= Contiene.NroSerieDisco AND
      Contiene.CodiceReg = Esecuzione.CodiceReg AND
      Disco.TitoloAlbum = Esecuzione.TitoloCanz

```

**.11 Dischi: autori non cantanti**

```

SELECT DISTINCT Nome
FROM Autore
      EXEPT
SELECT DISTINCT NomeCantante
FROM Cantante

```

**.12 Dischi: autori e cantanti "puri"**

```

(SELECT DISTINCT Nome
FROM Autore
      EXEPT
SELECT DISTINCT NomeCantante
FROM Cantante)
UNION
(SELECT DISTINCT NomeCantante
FROM Cantante
      EXEPT
SELECT DISTINCT Nome
FROM Autore)

```

**.13 Dischi: elenco brani**

```

SELECT TitoloCanzone
FROM Disco, Contiene, Esecuzione
WHERE Disco.NroSerie= Contiene.NroSerieDisco AND
      Contiene.CodiceReg = Esecuzione.CodiceReg AND
      Disco.NroSerie = 1
ORDER BY NroProgr

```

**.14 Dischi: elenco brani e cantanti**

```

SELECT TitoloCanzone, NomeCantante
FROM Disco, Contiene, Esecuzione LEFT OUTER JOIN Cantante
      ON Esecuzione.CodiceReg = Cantante.CodiceReg
WHERE Disco.NroSerie= Contiene.NroSerieDisco AND
      Contiene.CodiceReg = Esecuzione.CodiceReg AND
      Disco.NroSerie = 1
ORDER BY NroProgr

```

**.15 Dischi: coautori di Mogol**

```

SELECT Coautore.Nome
FROM Autore, Autore AS Coautore
WHERE Autore.TitoloCanzone = Coautore.TitoloCanzone AND
      Autore.Nome = "Mogol" AND

```

Coautore.Nome <> "Mogol"

### **.16 Dischi: numero brani di ogni album**

```
SELECT Count (*), NroSerieDisco  
FROM Contiene  
GROUP BY NroSerieDisco  
ORDER BY Count (*)
```

### **.17 Domande teoriche varie**

1. Il modello reticolare fa uso esplicito di puntatori. Il modello relazionale usa la coincidenza di valori per esprimere i riferimenti tra le varie informazioni
2. Il sistema informativo è la collezione delle informazioni a disposizione di un'organizzazione. Il sistema informatico è la parte automatizzata del sistema informativo grazie alla tecnologia informatica.
3. La relazione in senso matematico assume un ordinamento tra i domini coinvolti. La relazione nel modello relazionale associa ad ogni dominio un nome di attributo univoco e non necessita di un ordinamento tra gli attributi.
4. Un vincolo di integrità è detto "di tupla" quando è possibile verificare il soddisfacimento del vincolo osservando una singola tupla alla volta. Una chiave non è un vincolo di tupla perché per vedere se un'istanza di base di dati soddisfa il vincolo è necessario confrontare tutte le tuple di una relazione.
5. La chiave è minimale, cioè qualsiasi sottoinsieme degli attributi che la compongono non è superchiave
6. È il progettista e l'amministratore della base di dati. Si occupa della modifica degli schemi logici e degli schemi fisici al fine di ottimizzare le operazioni più usuali. Gestisce gli utenti e i loro privilegi.
7. La chiave primaria è stata scelta tra le altre per riferire le tuple della relazione. Dunque non si accettano valori nulli sulla attributi della chiave primaria.
8. Che l'ordine degli attributi di una relazione non è significativo
9. Che una espressione in algebra relazionale contiene una descrizione rigorosa delle operazioni che occorre eseguire per avere il risultato. I linguaggi dichiarativi danno solamente una descrizione delle proprietà del risultato senza descrivere implicitamente le operazioni necessarie per eseguirlo.
10. No, perché le tuple sono funzioni che associano ad ogni nome di attributo un valore nel suo dominio. Se due attributi avessero lo stesso nome, le tuple sarebbero equivoche, in quanto non si capirebbe a quale dei due attributi associare i valori specificati.
11. Il DBMS è un sistema automatico di gestione di basi di dati. Si può interagire con esso direttamente tramite interrogazioni con linguaggi interattivi tipo SQL, immergendo tali interrogazioni in altri linguaggi di programmazione (come C, pascal, Java), oppure tramite interfacce grafiche.
12. Vuol dire che non usa puntatori ma coincidenze di valori per stabilire riferimenti tra le informazioni. La più immediata conseguenza è che non si possono ammettere nella stessa relazione due tuple con gli stessi valori senza ingenerare equivoci.
13. Un vincolo di integrità è detto di dominio quando per verificare se è soddisfatto è sufficiente controllare il valore di un dominio di una tupla alla volta. Esempio: un vincolo che imponga che il voto sia sempre inferiore a 31.
14. L'equi-join è un theta-join in cui le condizioni di selezione sono tutte ugualianze.
15. I valori nulli sono stati introdotti perché alcune informazioni possono essere non attualmente disponibili, inesistenti, oppure potrebbe non essere noto se tali informazioni esistono e non sono note o non esistono affatto. Il modello relazionale definisce un solo tipo di valore nullo
16. Quando gli insiemi degli attributi delle due relazioni sono disgiunti.
17. Lo schema di una relazione è la parte invariante dell'informazione descritta dalla relazione: esso è composto da un nome di relazione e una serie di nomi e domini per gli attributi. L'istanza di relazione è la collezione delle tuple della relazione stessa. Essa varia con gli aggiornamenti della base di dati.
18. Per indipendenza dei dati si intende la possibilità di modificare uno schema ad uno dei tre livelli senza necessariamente dover modificare tutti gli schemi agli altri livelli. Per esempio si può modificare lo schema fisico lasciando inalterato lo schema logico e quindi tutti gli schemi esterni. Si può modificare una vista senza dover modificare lo schema logico.