

# DATABASE

## Progettare una base di dati

### Database fisico e database logico

- Un DB è una collezione di tabelle, le cui proprietà sono specificate dai metadati
- Attraverso le operazioni sulle tabelle è possibile ottenere **tabelle temporanee** partendo da quelle esistenti
- Queste considerazioni portano a considerare separatamente il **database fisico** dal **database logico**
  - **DB fisico**: formato dai dati effettivamente memorizzati nelle tabelle esistenti
  - **DB logico**: formato da tabelle temporanee (**viste**) create dinamicamente in modo da adattarsi alle esigenze degli utenti

## Il database fisico

- Costituito dalle tabelle effettivamente scritte su disco (non quelle temporanee)
- Criterio importante nel DB fisico = **EVITARE LA RIDONDANZA**
  - Non si devono mai replicare informazioni scrivendole in più posti diversi nel database
    - Rischio di inconsistenza tra le copie
      - Potremmo cambiare un dato in un posto e dimenticare di cambiarlo in un altro
    - I dati incoerenti sono chiamati **garbage**
      - Spazzatura → la loro presenza è peggio della mancanza assoluta di dati

Copyright © 2006 Pearson Education. All rights reserved.

13-3

## Assenza di ridondanza

- **Possibile problema:** l'informazione potrebbe essere necessaria in più parti del DB
  - Meglio tenere una **copia singola** e permettere agli altri di accedervi
- **Soluzione:** L'unica copia dei dati è mantenuta in una **tabella separata** a cui si accede tramite una chiave
  - Tale chiave viene memorizzata in qualsiasi punto in cui avremmo dovuto ripetere l'informazione - detta **chiave esterna**

Copyright © 2006 Pearson Education. All rights reserved.

13-4

## Esempio di chiave esterna

| Nome    | Categoria | Formula | Peso atomico | Viscosità |
|---------|-----------|---------|--------------|-----------|
| benzene | 1         | C6H6    | 78,12        | 0,6028    |
| pentane | 2         | C5H12   | 72,17        | 0,225     |
| hexane  | 2         | C6H14   | 86,2         | 0,294     |
| ...     | ...       | ...     | ...          | ...       |

L'attributo Categoria nella tabella Solventi è una chiave esterna che riferenzia l'attributo Id\_categoria nella tabella Categorie

Categorie

| Id categoria | Nome                  |
|--------------|-----------------------|
| 1            | alcohol               |
| 2            | aliphatic hydrocarbon |
| 3            | .....                 |

Una chiave esterna è la chiave primaria di un'altra tabella

Copyright © 2006 Pearson Education. All rights reserved.

## Lo schema di un database

- **Lo schema del database** è formato dall'insieme dei **metadati** delle tabella che lo costituiscono
  - **Struttura e relazioni tra tabelle**
  - **NOTA: Le relazioni fanno parte dello schema del DB (metadati)**
- Per illustrare la strategia di eliminazione della ridondanza, consideriamo un DB universitario con due tabelle, **Studente** e **Residenza**

Copyright © 2006 Pearson Education. All rights reserved.

13-6

## Database universitario

```

Studente
  Num_Matricola   Integer      8 cifre
  Nome            Character, 25
  Cognome         Character, 25
  Data_Nascita    Data
  Fuori corso     Boolean        0 = no;
                                       1 = si
Chiave primaria: Num_Matricola

```

```

Residenza
  Num_Matricola   Integer      8 cifre
  Indirizzo       Character, 100 tutto ciò che precede la città
  Città           Character, 25  senza abbreviazioni
  Provincia       Character, 25
  CAP             Character, 5
Chiave primaria: Num_Matricola

```

Copyright © 2006 Pearson Education. All rights reserved.

13-7

## Collegare tabelle mediante relazioni

- Le due tabelle sono ***distinte, ma non indipendenti***
- Il Num\_Matricola collega ogni riga di Studente al rispettivo indirizzo in Residenza
  - Si dice che tra le due entità c'è una ***relazione***
    - Corrispondenza tra le righe di una tabella e quelle di un'altra
    - La relazione è uno-a-uno poiché Num\_matricola è la chiave di entrambe le tabelle
    - La relazione è bidirezionale (possiamo trovare l'indirizzo per ogni studente e anche lo studente per ogni indirizzo)

Copyright © 2006 Pearson Education. All rights reserved.

13-8

## Join e chiavi esterne

- Num\_Matricola è una chiave esterna
- La relazione tra due tabelle permette di costruirne una terza temporanea (Lista\_Studenti) che contiene le informazioni combinate di entrambe
  - Utilizziamo l'operazione di **join** descritta nella lezione precedente

```
Lista_Studenti = select * from Studente JOIN Residenza
on Studente.Num_Matricola = Residenza.Num_Matricola
```

Copyright © 2006 Pearson Education. All rights reserved.

13-9

## Progettare uno schema di database

- Potremmo progettare altre 2 tabelle da cui si può accedere all'informazione sull'indirizzo di uno studente (senza replicarla)
  - Definiamo due tabelle **Borsista e Sportivo** senza l'indirizzo, ma con Num\_Matricola come chiave (primaria e anche esterna)
    - Borsista (Num\_Matricola, Dipartimento, Media\_voti)
    - Sportivo (Num\_Matricola, Num\_Armadietto, Disciplina)
  - Ogni nuova tabella ha una relazione uno-a-uno con la tabella Residenza e Studente

**Abbiamo 4 tabelle che non replicano alcun dato**

Copyright © 2006 Pearson Education. All rights reserved.

13-10

## Database logici e fisici

- Partendo da queste **4 tabelle fisiche**, ne possiamo creare altre temporanee e *personalizzate* per i vari dipartimenti e uffici (come Lista\_Studenti)
- L'insieme delle tabelle temporanee va sotto il nome di **database logico (tabelle logiche)**
  - *Il database logico non esiste fisicamente*
  - È ricreato ogni volta che è necessario, usando i valori correnti (→ aggiornati) del DB fisico
  - Contiene informazioni duplicate → non lo memorizziamo in modo permanente per non creare ridondanza
  - Le tabelle logiche sono anche chiamate **viste, perché danno una visuale personalizzata dei dati nel DB**

Copyright © 2006 Pearson Education. All rights reserved.

## Creare la vista del Preside

- Supponiamo che il Preside voglia visualizzare:  
Nome Cognome Città Dipartimento e Media\_voti
- Contiene informazioni selezionate dalle tabelle fisiche

| Vista_Preside | Tabella Sorgente |  |
|---------------|------------------|--|
| Nome          | Studente         |  |
| Cognome       | Studente         |  |
| Città         | Residenza        | <i>Non è necessario conoscere l'indirizzo completo</i> |
| Dipartimento  | Borsista         | <i>Indica la specializzazione dello studente</i>       |
| Media_Voti    | Borsista         | <i>Quanto è bravo, precisamente</i>                    |

Figura 13.2 Vista\_Preside combina informazioni prese da diverse tabelle.

Copyright © 2006 Pearson Education. All rights reserved.

13-12

## Riunire tre tabelle con un join

- Come si scrive la query per creare la vista del Preside?
- Primo passo: notate che la vista del Preside contiene informazioni prese da **tre tabelle**
- L'operazione di **join** associa l'informazione per ogni studente attraverso **Num\_Matricola**
- Per ogni studente si possono riunire le informazioni prese da tutte e tre le tabelle

Dati\_Preside =

```
Select * from Borsista JOIN (Studiante JOIN Residenza on
Studiante.Num_Matricola = Residenza.Num_Matricola) on
Borsista.Num_Matricola =Studiante.Num_Matricola
```

Copyright © 2006 Pearson Education. All rights reserved.

13-13

## “Sfoltire” i dati nella tabella

- Recuperare solo le colonne che il Preside vuole vedere
- La strategia del “join più sfoltimento” è un approccio standard
  - Si crea una “supertabella” (es. Dati\_Preside) mediante il join di diverse tabelle fisiche
  - Quindi si estraggono solo le informazioni che interessano l'utente tramite proiezione

```
Vista_Preside = Select Nome, Cognome, Città,
Dipartimento, Media_Voti From Dati_Preside
```

Copyright © 2006 Pearson Education. All rights reserved.

13-14

## Query su tabelle logiche

- **Le tabelle logiche possono poi essere usate come quelle fisiche**
- Se il Preside volesse premiare gli studenti con media superiore a 28  
Lista\_premiati =  
Select Num\_Matricola, Nome, Cognome From  
Vista\_Preside Where Media\_Voti > 28
- Lista\_premiati sarà anch'essa una tabella logica

Copyright © 2006 Pearson Education. All rights reserved.

## Le fasi di progettazione di un database



## Passi per la costruzione di un DB

- Effettuare un'analisi dei requisiti
  - Occorre capire in che modo il database sarà utilizzato, quali tipi di dati verranno immessi e quali relazioni ci sono sui dati
- Scrivere e raffinare un prototipo di progetto fisico
- Implementare il progetto fisico
- Progettare il database logico
- Implementare il database logico
- Implementare le interfacce grafiche (es. maschere per inserimento dati)
- Valutare l'efficacia del prodotto finale e (eventualmente) rivedere il progetto

Copyright © 2006 Pearson Education. All rights reserved.

14-17

## Esempio: noleggio attrezzatura sportiva

- Una società che noleggia equipaggiamenti sportivi per diverse attività sportive
- Il progetto del database deve riguardare:
  - Anagrafica clienti
  - Anagrafica dipendenti
  - I noleggi dell'attrezzatura sportiva

Copyright © 2006 Pearson Education. All rights reserved.

14-18

## Operazioni dei clienti

- Esempio: noleggio attrezzatura sciistica
- Un cliente attraversa tipicamente le fasi:
  - **Registrazione.** Il cliente fornisce all'addetto il nome, l'indirizzo di residenza e il domicilio locale
  - **Prova.** Un dipendente ottiene informazioni sul peso e l'abilità sciistica per la scelta degli attacchi. I numeri di matricola sono registrati e l'equipaggiamento viene consegnato al cliente
  - **Pagamento.** Il cliente firma il contratto di noleggio e paga

Copyright © 2006 Pearson Education. All rights reserved.

14-19

## Analisi dei requisiti

- Considerare i dati creati durante una operazione e le relazioni necessarie
- Dati creati:
  - Informazioni personali del cliente
  - Dati su equipaggiamento noleggiato e dipendente
- Relazioni necessarie:
  - Associazione di dati personali cliente con dati tecnici relativi all'attrezzatura e dipendente assegnato (contratto di noleggio)
- Idea: costruire una unica tabella Noleggi con un campo per ogni informazione necessaria nel contratto
  - Ogni riga rappresenta un contratto di noleggio

Copyright © 2006 Pearson Education. All rights reserved.

14-20

## Analisi dei requisiti (cont.)

- Ogni contratto deve contenere dati del cliente, come il nome, l'indirizzo e il numero di telefono del cliente, etc...
- Se il cliente ritorna, queste informazioni saranno ripetute in tante righe della tabella Noleggi → **ridondanza! NON VA BENE**
- Definiamo quindi una tabella separata per i clienti (**Clients**)
- Definiamo una tabella anche per i dipendenti dell'agenzia (**Squadra**)
- La tabella dei **Noleggi** registra la data del noleggio, l'equipaggiamento noleggiato, un identificativo del cliente, etc.
- Il DB ha ora tre tabelle: **Clients**, **Squadra** e **Noleggi**
  - ci sono relazioni tra **Clients** e **Noleggi**
  - ci sono relazioni tra **Squadra** e **Noleggi**

Copyright © 2006 Pearson Education. All rights reserved.

14-21

## Un primo prototipo

- Specifica della tabella **Clients**

| Clients                     |               |                                 |
|-----------------------------|---------------|---------------------------------|
| ID_Cliente                  | Integer       | <i>identificatore unico</i>     |
| Nome                        | Character, 20 |                                 |
| Cognome                     | Character, 30 |                                 |
| Data_Nascita                | Data          | <i>data di nascita</i>          |
| Indirizzo                   | Character, 30 | <i>indirizzo di casa</i>        |
| Città                       | Character, 20 |                                 |
| Provincia                   | Character, 2  | <i>in forma abbreviata</i>      |
| CAP                         | Character, 5  |                                 |
| Paese                       | Character, 10 |                                 |
| Tel_casa                    | Character, 20 | <i>telefono di casa</i>         |
| Tel_cellulare               | Character, 20 |                                 |
| Email                       | Character, 40 |                                 |
| Indir_locale                | Character, 40 | <i>domicilio locale (hotel)</i> |
| Chiave Primaria: ID_Cliente |               |                                 |

Copyright © 2006 Pearson Education. All rights reserved.

14-22

## Un primo prototipo

- Specifica della tabella **Squadra**

| Squadra                     |                |   |
|-----------------------------|----------------|---|
| Soprannome                  | Character, 10  |   |
| Nome                        | Character, 20  |   |
| Cognome                     | Character, 30  |   |
| Data_Nascita                | Data           | <i>data di nascita</i>                  |
| Numero_Assicur              | ddd-dd-dddd    | <i>assicurazione infortuni</i>          |
| Tel_cellulare               | Character, 20  |   |
| Email                       | Character, 40  |   |
| Certificato S/N             |                | <i>ha una certificazione?</i>           |
| Cert_Termine                | Data           | <i>quando scade la certificazione?</i>  |
| Cert_Dettagli               | Character, 255 | <i>descrizione della certificazione</i> |
| Chiave Primaria: Soprannome |                |   |

**NOTA** I dipendenti potrebbero avere due recapiti: uno permanente e uno temporaneo

Copyright © 2006 Pearson Education. All rights reserved.

14-23

## Un primo prototipo

- **Schema** dei recapiti degli impiegati

| Recapito                 |               |                             |
|--------------------------|---------------|-----------------------------|
| SopraNN                  | Character, 10 | <i>identificatore unico</i> |
| Indirizzo                | Character, 30 | <i>indirizzo di casa</i>    |
| Città                    | Character, 20 |                             |
| Provincia                | Character, 2  | <i>in forma abbreviata</i>  |
| CAP                      | Character, 5  |                             |
| Paese                    | Character, 10 |                             |
| Tel_casa                 | Character, 20 | <i>telefono di casa</i>     |
| Chiave Primaria: SopraNN |               |                             |

- Lo stesso **schema** può anche essere usato per creare due tabelle, In\_Casa e Indirizzo\_Locale (es. per il recapito permanente e temporaneo)
- SopraNN corrisponde a Soprannome in **Squadra**: serve per mettere le tabelle in *una relazione uno-a-uno*

Copyright © 2006 Pearson Education. All rights reserved.

14-24

## Un primo prototipo

- Specifica della tabella **Noleggi**
  - ID noleggio (chiave primaria)
  - chiave della tabella **Clients (chiave esterna)**
  - chiave della tabella **Squadra (chiave esterna)**
  - Data noleggio
  - Altri dati necessari riguardanti l'attrezzatura noleggiata:
    - Num\_scarp – numero di serie scarponi
    - Num\_sci – numero di serie sia per sci che per tavole
    - Sci? – noleggia sci o tavola (S/N)
    - Abilità, peso – informazioni sul cliente
    - Att\_sx, att\_dx – configurazione attacchi

Copyright © 2006 Pearson Education. All rights reserved.

14-25

## Un primo prototipo

- Specifica delle relazioni tra tabelle
  - Relazioni uno-a-molti
    - **Noleggia**, la relazione tra Clients e Noleggi
    - **Gestisce**, la relazione tra Squadra e Noleggi
  - Relazioni uno-a-uno
    - **Casa\_Di**, la relazione Ind\_Casa:Squadra
    - **Domicilio\_Di**, la relazione Ind\_Locale:Squadra

Copyright © 2006 Pearson Education. All rights reserved.

14-26

## Rivedere il progetto fisico?

- Valutare il progetto fisico e verificare se soddisfa i requisiti degli utenti
- Nel database non c'è ridondanza (progetto corretto), ma il progetto può essere esteso ad altre attività sportive?
  - Potenziale problema: i dettagli dell'attrezzatura devono essere specializzati per la nuova attività
- Idea: i dettagli sull'attrezzatura noleggiata potrebbero essere spostati in una tabella separata, di nome **Attrezzatura**

Copyright © 2006 Pearson Education. All rights reserved.

14-27

## Rivedere il progetto fisico?

- Chiave della tabella **Attrezzatura**:  
ID\_Attrezzatura
- **Noleggi** non contiene più i dettagli dell'attrezzatura, ma solo ID\_Attrezzatura
- Si introduce una nuova relazione
  - Si crea una relazione uno-a-molti tra le due tabelle **Attrezzatura e Noleggi**
- Assicurarsi che i campi siano unici
  - In caso di più negozi che effettuano noleggio per l'agenzia, gestire il campo chiave di **Noleggi** in modo che identifichi **univocamente** un noleggio

Copyright © 2006 Pearson Education. All rights reserved.

14-28

## Implementazione del database fisico

- Definire le tabelle
- Definire le relazioni

Copyright © 2006 Pearson Education. All rights reserved.

14-29

The figure displays four screenshots of the OpenOffice.org Base Table Design interface, showing the field definitions for four tables: Clienti, Noleggi, Squadra, and Attrezz\_Sci.

**Clienti - OpenOffice.org Base: Table Design**

| Field Name    | Field Type          | Description                       |
|---------------|---------------------|-----------------------------------|
| ID_Cliente    | Integer [ INTEGER ] | identificatore unico              |
| Nome          | Text [ VARCHAR ]    |                                   |
| Cognome       | Text [ VARCHAR ]    |                                   |
| Data_Nascita  | Date [ DATE ]       |                                   |
| Indirizzo     | Text [ VARCHAR ]    | di casa: si presume non cambi mai |
| Città         | Text [ VARCHAR ]    |                                   |
| Provincia     | Text [ VARCHAR ]    | in forma abbreviata (sigla)       |
| CAP           | Text [ VARCHAR ]    |                                   |
| Paese         | Text [ VARCHAR ]    |                                   |
| Tel_Casa      | Text [ VARCHAR ]    |                                   |
| Tel_Cellulare | Text [ VARCHAR ]    |                                   |
| Email         | Text [ VARCHAR ]    |                                   |
| Indr_Locale   | Text [ VARCHAR ]    | residenza locale (albergo)        |

**Noleggi - OpenOffice.org Base: Table Design**

| Field Name  | Field Type          | Description  |
|-------------|---------------------|--|
| ID_Regione  | Text [ VARCHAR ]    | nome abbreviato (senza) del regione                    |
| ID_Noleggio | Integer [ INTEGER ] | numero (distinto da tutti gli altri) del noleggio      |
| Data        | Date [ DATE ]       | data della transazione                                 |
| Cliente     | Integer [ INTEGER ] | chiave della tabella Clienti                           |
| Espresso    | Text [ VARCHAR ]    | chiave della tabella Squadra                           |
| Attrezz     | Integer [ INTEGER ] | chiave della tabella Attrezz_sci o di un'altra tabella |
| Pagamento   | Real [ REAL ]       | pagamento del noleggio                                 |

**Squadra - OpenOffice.org Base: Table Design**

| Field Name     | Field Type         | Description                      |
|----------------|--------------------|----------------------------------|
| ID_Sopranome   | Text [ VARCHAR ]   | usato come chiave primaria       |
| Nome           | Text [ VARCHAR ]   |                                  |
| Cognome        | Text [ VARCHAR ]   |                                  |
| Data_Nascita   | Date [ DATE ]      | data di nascita                  |
| Numero_assicur | OTHER [ OTHER ]    | assicurazione infortuni          |
| Tel_Cellulare  | Text [ VARCHAR ]   |                                  |
| Email          | Text [ VARCHAR ]   | indirizzo di posta elettronica   |
| Certificato    | Yes/No [ BOOLEAN ] | ha una certificazione? (S/N)     |
| Cert_Termine   | Date [ DATE ]      | scadenza della certificazione    |
| Cert_Dettagli  | Text [ VARCHAR ]   | descrizione della certificazione |

**Attrezz\_Sci - OpenOffice.org Base: Table Design**

| Field Name | Field Type          | Description                                   |
|------------|---------------------|---|
| ID_Sci     | Integer [ INTEGER ] | identificatore unico                          |
| Num_scarp  | Text [ VARCHAR ]    | numero di serie degli scarponi                |
| Num_sci    | Text [ VARCHAR ]    | numero di serie di sci o tavola               |
| Sci?       | Yes/No [ BOOLEAN ]  | il noleggio coinvolge attacchi da sci?        |
| Abilita    | Text [ VARCHAR ]    | Principiante/Medio/Espresso                   |
| Peso       | Integer [ INTEGER ] | peso dello sciatore                           |
| Att_sin    | Text [ VARCHAR ]    | configurazione attacco sinistro               |
| Att_dir    | Text [ VARCHAR ]    | configurazione attacco destro                 |
| Insellanti | Text [ VARCHAR ]    | sigla e lunghezza dei bastoncini              |
| Note       | Text [ VARCHAR ]    | commenti riguardanti il noleggio o il cliente |

Figura 14.1 Quattro delle sei tabelle che compongono il progetto finale del database fisico, nella sua versione per il negozio di sci Snow Machine. Le tabelle Ind\_Casa e Ind\_Locale hanno la stessa struttura e contengono solo indirizzi: i loro campi sono identici ai campi 5-10 della tabella Clienti.

Copyright © 2006 Pearson Education. All rights reserved.

14-30

## Il database logico

- Scrivere query (viste) per un database:
  - si analizza l'attività della società per identificare le operazioni necessarie
  - si determinano le informazioni che servono per ogni operazione e si identificano le tabelle fisiche in cui sono memorizzate
  - si scrive una query SQL per una nuova tabella (vista) che contenga i dati necessari per ciascuna operazione
- *NOTA: in fase di inserimento di una entità in una tabella non è detto che tutti i campi della tabella debbano essere riempiti nello stesso momento*

Copyright © 2006 Pearson Education. All rights reserved.

14-31

## Il database logico (cont.)

- Esempio
 

Una vista relativa all'inserimento di nuovi impiegati:

  - **Assunzione.** Le informazioni di ogni nuovo "esperto" vanno inserite nel database al momento dell'ingaggio. Quest'operazione coinvolge tre tabelle, **Squadra, Ind\_Casa e Ind\_Locale**, di cui servono tutti i campi
- **NOTA:** Le viste possono essere usate per l'inserimento di dati in più tabelle contemporaneamente

Copyright © 2006 Pearson Education. All rights reserved.

14-32



## Il database logico (cont.)

- Tre viste legate al rapporto con la clientela:
  - **Registrazione.** La registrazione dà l'avvio a un noleggio, per cui la dobbiamo immaginare come una nuova riga nella tabella **Noleggi**, di cui servono i campi **ID\_noleggio** e **Cliente**. Son richiesti anche tutti i campi della tabella **Clients** (da inserire se è un nuovo cliente)
  - **Configurazione.** L'esperto aiuta il cliente a scegliere e configurare l'attrezzatura: servono tutti i campi di **Attrezzatura**, **Nome** e **Cognome** da **Clients**, **ID\_Noletggio**, **Cliente** ed **Esperto** da **Noleggi**.
  - **Contratto.** Il cliente passa alla firma del contratto. Per la creazione del contratto occorre combinare e stampare informazioni prese dalle tabelle **Noleggi** e **Attrezzatura** (**tutti i campi**), **Squadra** (**Nome e Cognome**), e **Clients** (**tutti tranne Email e Indir\_Locale**)

Copyright © 2006 Pearson Education. All rights reserved.

14-33

## Riassunto dei campi richiesti

| Viste                 | Clients                              | Squadra         | Attrezzatura | Ind_Casa | Ind_Locale | Noleggi                            |
|-----------------------|--------------------------------------|-----------------|--------------|----------|------------|------------------------------------|
| <i>Registrazione</i>  | Tutti                                |                 |              |          |            | ID_Noletggio<br>Cliente            |
| <i>Configurazione</i> | Nome<br>Cognome                      |                 | Tutti        |          |            | ID_Noletggio<br>Cliente<br>Esperto |
| <i>Contratto</i>      | Tutti tranne:<br>Email<br>Ind_Locale | Nome<br>Cognome | Tutti        |          |            | Tutti                              |
| <i>Assunzione</i>     |                                      | Tutti           |              | Tutti    | Tutti      |                                    |

Tabella 14.1 Riepilogo dei campi richiesti per le viste sul database della Snow Machine.

Copyright © 2006 Pearson Education. All rights reserved.

14-34

## Scrivere le interrogazioni che creano le viste

- Per creare le viste scriveremo alcune query
  - Approccio “join più sfoltimento”: si crea una “supertabella” e si estraggono gli elementi necessari alla vista
- Query per la **registrazione**:
  - Dà l’avvio a un noleggio (crea una nuova riga nella tabella **Noleggi**)
  - Vengono impostati i valori ID\_Noleggio e Cliente di Noleggi (e tutti i campi di Clienti se è un nuovo cliente)

Copyright © 2006 Pearson Education. All rights reserved.

14-35

## Scrivere le interrogazioni che creano le viste (cont.)

- Query per la **configurazione**:
  - Servono ID\_Noleggi, Esperto e Cliente dalla tabella Noleggi; tutti i campi dalla tabella Attrezzatura; nome e cognome dalla tabella Cliente
- Query per il **contratto**:
  - Dalle prime tre tabelle servono tutti i campi eccetto Clienti.Email e Clienti.Ind\_Locale; dalla tabella Squadra solo nome e cognome dell’Esperto
- Query per **l’assunzione**:
  - Tutti i campi delle tre tabelle Squadra, Ind\_Casa e Ind\_Locale

Copyright © 2006 Pearson Education. All rights reserved.

14-36

## Query SQL per l'assunzione dei dipendenti

- Codifica della query per l'assunzione dei dipendenti
  - consiste nel join di tre tabelle

```
SELECT Squadra.Soprannome, Squadra.Nome,
       Squadra.Cognome, Squadra.Numero_assicur,
       Squadra.Tel_cellulare, Squadra.Email,
       Squadra.Certificato, Squadra.Cert_termine,
       Squadra.Cert_Dettagli, Ind_Casa.Indirizzo,
       Ind_Casa.Città...
FROM Ind_Locale JOIN (Ind_Casa JOIN Squadra
                     ON Ind_Casa.SopraNN=Squadra.Soprannome)
                     ON Ind_Locale.SopraNN=Squadra.Soprannome;
```

Copyright © 2006 Pearson Education. All rights reserved.

## Implementazione del database logico (cont.)

- Codifica della query per la registrazione dei clienti e della query per la configurazione dell'attrezzatura
  - segue le stesse linee guida
- Codifica della query per la generazione del contratto
  - Coninvolge quattro tabelle
  - Tuttavia, dato che utilizza tutti i campi (tranne due) delle prime tre tabelle e solo tre campi dell'ultima, non è molto difficile concettualmente

Copyright © 2006 Pearson Education. All rights reserved.

14-38

## Query SQL per la generazione del contratto

```

SELECT ..... (elenco di tutti i campi richiesti)...
FROM
Attrezzatura JOIN
  (Clienti JOIN
    (Squadra JOIN Noleggi
      ON Squadra.Soprannome = Noleggi.Esperto)
    ON Clienti.ID_Cliente = Noleggi.Cliente)
ON Attrezzatura.ID_Attrezzatura=Noleggi.Attrezz;

```

Copyright © 2006 Pearson Education. All rights reserved.

## QBE: Query by Example

- Sebbene formulare le query in SQL non sia complicatissimo, la **tecnica QBE** lo rende ancor più semplice
  - *Disponibile in Microsoft Access*
- **Idea:** l'utente fornisce un esempio della tabella finale che desidera ottenere specificando i campi in una tabella vuota
- *Il software genera il codice SQL necessario per ottenere una tabella (vista) identica a quella indicata*

Copyright © 2006 Pearson Education. All rights reserved.