

Relazione sul progetto di Basi di Dati

Tommaso Frassetto,
matr. 1007883

26 giugno 2012

Abstract

Il progetto modella il processo di gestione di uno studio legale. In particolare, vengono gestite le cause e, per ciascuna causa, le informazioni relative alle parti coinvolte (persone, società e istituzioni), il giudice e il tribunale a cui è assegnata, le udienze in cui è stata discussa e i collaboratori che hanno partecipato ad ogni udienza. Le operazioni tipiche sono creazione e modifica dei *record* anagrafici, creazione di cause e aggiunta di udienze associate ad una causa.

1 Descrizione dei requisiti

Il progetto consiste nella realizzazione di una base di dati che modelli alcune classi coinvolte nella gestione di uno studio legale; il personale interagisce con la base di dati tramite una interfaccia *web*, che non è destinata ai clienti.

L'entità principale che interessa modellare sono le cause. Di ognuna interessa il numero di Registro Generale (RG), che identifica la causa all'interno di ogni tribunale, lo stato attuale, gli enti coinvolti come parti nella causa, il loro ruolo nella causa e se siano o meno clienti dello studio in relazione a quella causa; inoltre, interessano il giudice e il tribunale incaricati della causa e le (zero o più) udienze in cui è stata discussa. Una causa può essere affidata solo ad un giudice che lavora presso il tribunale dove si celebra la causa.

Di tutti gli enti interessa il nome. Gli enti possono essere persone fisiche, società o istituzioni; delle prime interessano codice fiscale, data e luogo di nascita; delle seconde interessano partita IVA e nome del legale rappresentante; delle ultime interessano tipo di istituzione (comune, provincia e così via) e nome del responsabile. Inoltre, ad ogni ente sono associati zero o più recapiti. Ogni recapito è corredato da una descrizione e può essere un numero di telefono, un indirizzo fisico o di posta elettronica.

Dei giudici interessano, oltre agli attributi relativi alle persone fisiche, il tipo (togato o onorario) e i tribunali in cui lavorano (un giudice lavora in uno o più tribunali e un tribunale ha uno o più giudici). Un giudice togato non può lavorare presso un ufficio del Giudice di Pace.

Dei tribunali sono rilevanti il tipo di tribunale, il luogo, l'indirizzo e il telefono. In ogni luogo c'è al più un tribunale di ogni tipo.

Delle udienze interessano la data e il tipo, oltre ai collaboratori che le hanno discusse (almeno uno).

Dei collaboratori sono rilevanti, oltre agli attributi relativi alle persone fisiche, la qualifica (Avvocato, Praticante con patrocinio, Praticante senza patrocinio e Segre-

tario) e la retribuzione (convenzionalmente pari a -1 per indicare i soci). Ad ogni collaboratore è inoltre associata una *password* di accesso al sistema.

Le operazioni tipiche sono creazione, aggiornamento ed eliminazione delle entità sopra nominate. Gli avvocati possono compiere tutte le operazioni; i praticanti con e senza patrocinio possono compiere tutte le operazioni di creazione e modifica, ma possono cancellare solo recapiti di un ente e partecipazioni di un collaboratore ad un'udienza. I segretari possono solo consultare il sistema.

Nota: per accedere il sistema i codici utente validi vanno dall'81 al 90; gli avvocati sono quelli con i numeri più bassi, mentre quelli con i codici più alti sono i segretari (che possono solo consultare). Ogni account ha la password uguale al codice utente.

2 Progettazione concettuale

2.1 Lista delle classi

ENTI: modella un generico ente.

- Nome: *string*

Sono definite le seguenti sottoclassi (con vincolo di partizionamento).

SOCIETÀ: rappresenta una società.

- PartitaIVA: *string*
- LegaleRappresentante: *string*

ISTITUZIONI: rappresenta una istituzione.

- Tipo: *string*
- PartitaIVA: *string*
- NomeResponsabile: *string*

PERSONEFISICHE: rappresenta una persona fisica.

- CodiceFiscale: *string*
- DataNascita: *date*
- LuogoNascita: *string*

Sottoclassi (senza alcun vincolo):

GIUDICI: modella un giudice.

- Tipo: (*Togato; Onorario*)

COLLABORATORI: modella un collaboratore.

- Qualifica: (*Avvocato; PraticanteConPatrocinio; Praticante; Segretario*)
- Retribuzione: *real* ≥ -1
- Password: *string*

RECAPITI: rappresenta i recapiti di un ente.

- TipoRecapito: (*Telefono; Indirizzo; Email*)
- Descrizione: *string*
- Recapito: *string*

CAUSE: modella una causa.

- RG: *string*
- Stato: *string*

RG identifica la causa all'interno del tribunale.

TRIBUNALI: rappresenta un tribunale.

- Luogo: *string*
- Tipo: *string*
- Indirizzo: *string*
- Telefono: *string*

In ogni luogo c'è al più un solo tribunale per tipo.

UDIENZE: contiene lo storico delle udienze.

- Data: *date*
- Tipo: *string*

2.2 Lista delle associazioni

- **RECAPITI-ENTI: è recapito di**
 - Ogni ente ha zero o più recapiti. Ogni recapito è di esattamente un ente.
 - Molteplicità: N:1
 - Totalità: totale verso RECAPITI, parziale verso ENTI
- **ENTI-CAUSE: è coinvolto in**
 - Ogni ente è coinvolto in zero o più cause. In ogni causa è coinvolto almeno un ente.
 - Molteplicità: N:M
 - Totalità: totale verso ENTI, parziale verso CAUSE
 - Attributi:
 - * Ruolo: *string*
 - * Cliente: *bool*
- **GIUDICI-CAUSE: giudica**
 - Ogni giudice giudica zero o più cause. Ogni causa ha esattamente un giudice. Il giudice di una causa deve lavorare presso il tribunale di quella causa.
 - Molteplicità: 1:N
 - Totalità: totale verso GIUDICI, parziale verso CAUSE
- **GIUDICI-TRIBUNALI: lavora in**
 - Ogni giudice lavora in uno o più tribunali. Ogni tribunale ha almeno un giudice. I tribunali di tipo 'Giudice di Pace' non possono avere giudici togati.

- Molteplicità: N:M
- Totalità: totale in entrambi i versi
- TRIBUNALI-CAUSE: **ospita**
 - Ogni tribunale ospita zero o più cause. Ogni causa è ospitata da esattamente un tribunale.
 - Molteplicità: 1:N
 - Totalità: totale verso TRIBUNALI, parziale verso CAUSE
- CAUSE-UDIENZE: **svilupata in**
 - Ogni causa è stata discussa in zero o più udienze. Ogni udienza è di esattamente una causa.
 - Molteplicità: 1:N
 - Totalità: totale verso CAUSE, parziale verso UDIENZE
- UDIENZE-COLLABORATORI: **discussa da**
 - Ogni udienza è discussa da uno o più collaboratori. Ogni collaboratore ha discusso zero o più udienze.
 - Molteplicità: N:M
 - Totalità: parziale verso UDIENZE, totale verso COLLABORATORI

Non è esplicitamente presente alcuna associazione di molteplicità 1:1 nel modello concettuale; alcune associazioni di questo tipo compaiono però nel modello logico tra le componenti della gerarchia degli enti.

3 Progettazione logica

3.1 Gerarchie

La gerarchia della classe ENTI è stata implementata con partizionamento verticale visto che la classe radice ha delle associazioni proprie e che le sottoclassi hanno numerosi attributi autonomi e associazioni proprie. La gerarchia completa viene quindi implementata usando le seguenti classi:

ENTI: viene istituita una nuova chiave sintetica intera. Non vengono usati codice fiscale o partita IVA perché potrebbero essere sconosciuti, renderebbero la chiave non uniforme e ridurrebbero l'efficienza.

- CodEnte: *int* «PK»
- Nome: *string* «NOT NULL»

SOCIETÀ: si utilizza CodEnte come chiave esterna verso ENTI.

- CodEnte: *int* «PK» «FK(ENTI)»
- PartitaIVA: *string*
- LegaleRappresentante: *string*

ISTITUZIONI: come sopra.

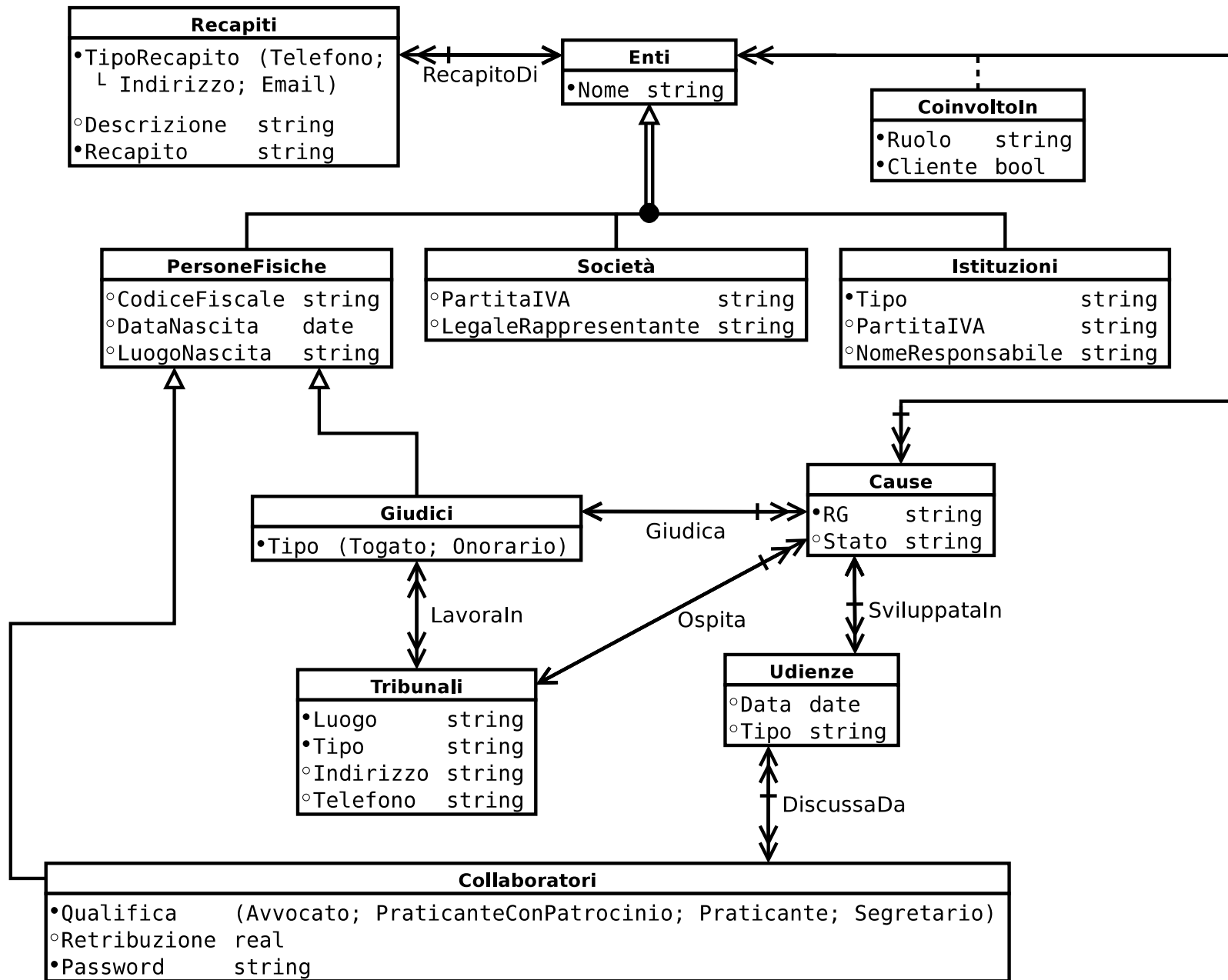


Figura 1: Schema concettuale della base di dati

- CodEnte: *int* «PK» «FK(ENTI)»
- Tipolstituzione: *string* «NOT NULL»
- PartitalVA: *string*
- NomeResponsabile: *string*

PERSONEFISCHE: come sopra.

- CodEnte: *int* «PK» «FK(PERSONEFISCHE)»
- CodiceFiscale: *string*
- DataNascita: *date*
- LuogoNascita: *string*

GIUDICI: qui CodEnte è chiave esterna verso PERSONEFISCHE.

- CodEnte: *int* «PK» «FK(PERSONEFISCHE)»
- Tipo: (*Togato; Onorario*) «NOT NULL»

COLLABORATORI: come sopra.

- CodEnte: *int* «PK» «FK(PERSONEFISCHE)»
- Qualifica: (*Avvocato; PraticanteConPatrocinio; Praticante; Segretario*) «NOT NULL»
- Retribuzione: *real* ≥ -1
- Password: *string*

3.2 Chiavi sintetiche

Sono state aggiunte chiavi primarie sintetiche intere ad alcune tabelle, in particolare CodUdienza a UDIENZE, CodTribunale a TRIBUNALI, CodCausa a CAUSE. Quest'ultima tabella avrebbe già la chiave (RG, CodTribunale), ma è stata inserita una chiave sintetica perché RG potrebbe essere sconosciuto o non assegnato.

3.3 Associazioni

- **Recapiti-ENTI: è recapito di**
 - Ogni ente ha zero o più recapiti. Ogni recapito è di esattamente un ente.
 - Molteplicità: N:1
 - Totalità: totale verso RECAPITI, parziale verso ENTI
 - Chiave esterna non-nulla in RECAPITI verso ENTI
- **ENTI-CAUSE: è coinvolto in**
 - Ogni ente è coinvolto in zero o più cause. In ogni causa è coinvolto almeno un ente.
 - Molteplicità: N:M
 - Totalità: totale verso ENTI, parziale verso CAUSE
 - Nuova tabella **COINVOLTOIN**, con i seguenti attributi:

- * CodEnte: *int* «PK» «FK(ENTI)»
- * CodCausa: *int* «PK» «FK(CAUSE)»
- * Ruolo: *string* «NOT NULL»
- * Cliente: *bool* «NOT NULL»

- GIUDICI-CAUSE: **giudica**

- Ogni giudice giudica zero o più cause. Ogni causa ha esattamente un giudice. Il giudice di una causa deve lavorare presso il tribunale di quella causa.
- Molteplicità: 1:N
- Totalità: totale verso GIUDICI, parziale verso CAUSE
- Chiave esterna non-nulla in CAUSE verso GIUDICI

- GIUDICI-TRIBUNALI: **lavora in**

- Ogni giudice lavora in uno o più tribunali. Ogni tribunale ha almeno un giudice. I tribunali di tipo ‘Giudice di Pace’ non possono avere giudici togati.
- Molteplicità: N:M
- Totalità: totale in entrambi i versi
- Nuova tabella **LAVORAIN**, con i seguenti attributi:
 - * CodEnte: *int* «PK» «FK(GIUDICI)»
 - * CodTribunale: *int* «PK» «FK(TRIBUNALI)»

- TRIBUNALI-CAUSE: **ospita**

- Ogni tribunale ospita zero o più cause. Ogni causa è ospitata da esattamente un tribunale.
- Molteplicità: 1:N
- Totalità: totale verso TRIBUNALI, parziale verso CAUSE
- Chiave esterna non-nulla in CAUSE verso TRIBUNALI

- CAUSE-UDIENZE: **sviluppata in**

- Ogni causa è stata discussa in zero o più udienze. Ogni udienza è di esattamente una causa.
- Molteplicità: 1:N
- Totalità: totale verso CAUSE, parziale verso UDIENZE
- Chiave esterna non-nulla in UDIENZE verso CAUSE

- UDIENZE-COLLABORATORI: **discussa da**

- Ogni udienza è discussa da uno o più collaboratori. Ogni collaboratore ha discusso zero o più udienze.
- Molteplicità: N:M
- Totalità: parziale verso UDIENZE, totale verso COLLABORATORI
- Nuova tabella **DISCUSSADA**, con i seguenti attributi:
 - * CodUdienza: *int* «PK» «FK(Udienza)»
 - * CodEnte: *int* «PK» «FK(COLLABORATORI)»

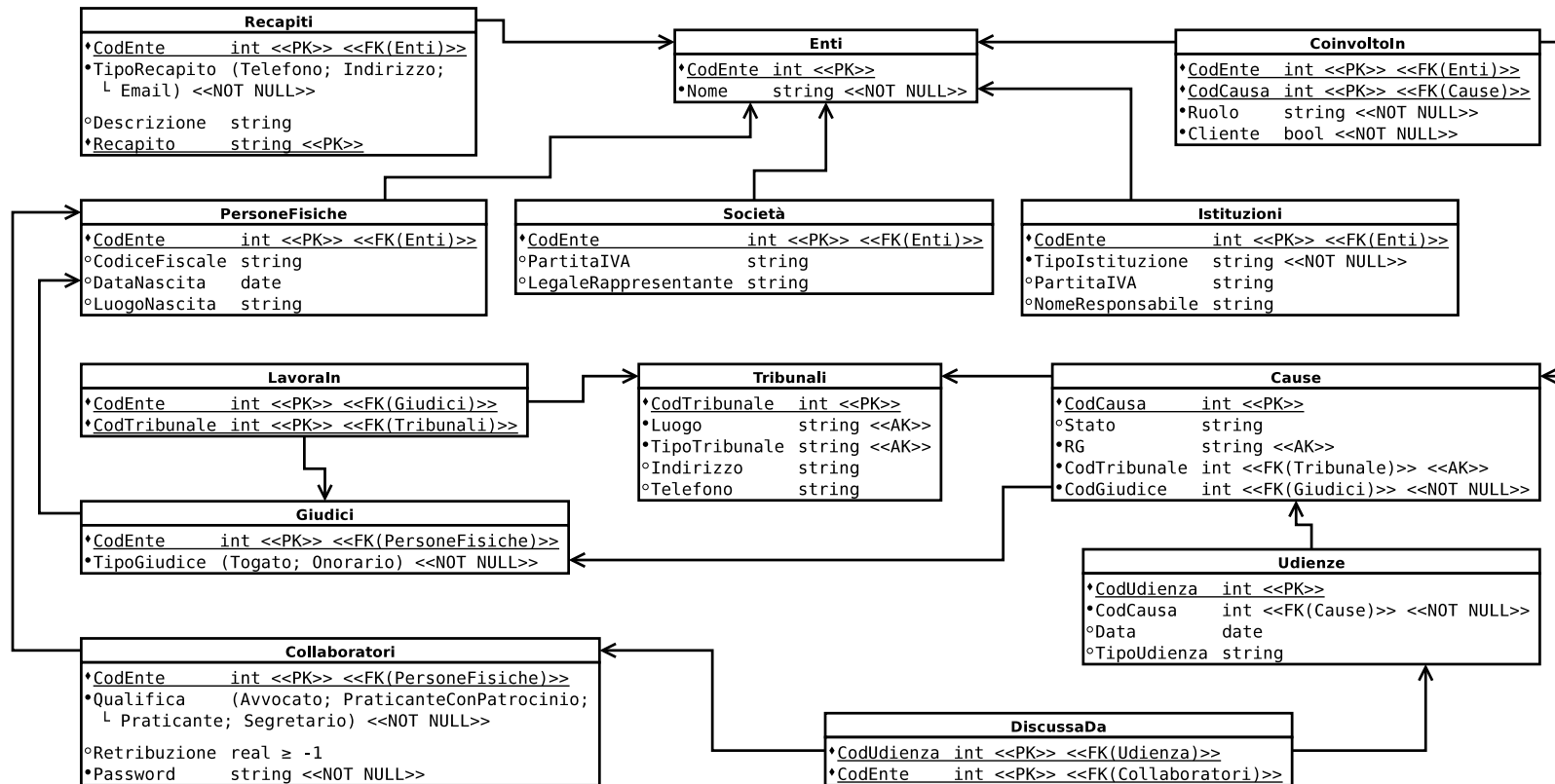


Figura 2: Schema logico della base di dati

4 Implementazione della base di dati

È stata implementata una tabella aggiuntiva, **ERRORI**, che contiene la descrizione degli errori rilevati dai *trigger*.

I vincoli di integrità del tipo ‘per ogni *A* almeno un *B*’ non sono stati implementati visto che sono di scarsa importanza e avrebbero reso necessario parecchio codice per gestire la creazione congiunta di un *record* principale e del primo *record* correlato, come anche per impedire la cancellazione di *record* correlati se sono gli unici. L’implementazione consente invece di creare separatamente *record* principale e un numero arbitrario di *record* correlati.

Il listato seguente implementa lo schema descritto in precedenza.

```
0 SET NAMES 'utf8';

USE tfrasset-PR;

DROP TABLE IF EXISTS DISCUSSADA;
DROP TABLE IF EXISTS UDienze;
DROP TABLE IF EXISTS COINVOLTOIn;
DROP TABLE IF EXISTS CAUSE;
DROP TABLE IF EXISTS LAVORAIN;
DROP TABLE IF EXISTS TRIBUNALI;
10 DROP TABLE IF EXISTS COLLABORATORI;
DROP TABLE IF EXISTS GIUDICI;
DROP TABLE IF EXISTS PERSONEFISICHE;
DROP TABLE IF EXISTS SOCIETA;
DROP TABLE IF EXISTS ISTITUZIONI;
DROP TABLE IF EXISTS RECAPITI;
DROP TABLE IF EXISTS ENTI;

DROP TABLE IF EXISTS ERRORI;
CREATE TABLE ERRORI (
20   CodErr      INT PRIMARY KEY,
   Descrizione VARCHAR(255)
) ENGINE=InnoDB;
INSERT INTO ERRORI(CodErr, Descrizione) VALUES
(24, 'Un giudice togato non può lavorare come giudice di pace'),
(25, 'Il giudice di una causa deve lavorare nel tribunale della causa');

CREATE TABLE ENTI (
30   CodEnte INT PRIMARY KEY AUTO_INCREMENT,
   Nome   VARCHAR(255) NOT NULL
) ENGINE=InnoDB;

CREATE TABLE RECAPITI (
40   CodEnte      INT,
   TipoRecapito  ENUM('Telefono', 'Indirizzo', 'Email') NOT NULL,
   Descrizione   VARCHAR(100),
   Recapito      VARCHAR(100),
   PRIMARY KEY (CodEnte, Recapito),
   CONSTRAINT fk_Recapiti_Enti FOREIGN KEY (CodEnte) REFERENCES ENTI(CodEnte) ON
      DELETE CASCADE
) ENGINE=InnoDB;

CREATE TABLE ISTITUZIONI (
   CodEnte      INT PRIMARY KEY,
   Tipolstituzione VARCHAR(100) NOT NULL,
   PartitaIVA    CHAR(11),
   NomeResponsabile VARCHAR(100),
```

```

    CONSTRAINT fk_Istituzioni_Enti FOREIGN KEY (CodEnte) REFERENCES ENTI(CodEnte) ON
    DELETE CASCADE
) ENGINE=InnoDB;

DELIMITER $
50 DROP PROCEDURE IF EXISTS InserisciIstituzione $
CREATE PROCEDURE InserisciIstituzione (IN
    ilNome          VARCHAR(255),
    ilTipolstituzione VARCHAR(100),
    laPartitaIVA     CHAR(11),
    ilNomeResponsabile VARCHAR(100) )
BEGIN
    DECLARE oldac BOOL;
    SELECT @@AUTOCOMMIT INTO oldac;
    SET AUTOCOMMIT=0;
    START TRANSACTION;
    60 INSERT INTO ENTI(Nome) VALUE (ilNome);
    INSERT INTO Istituzioni(CodEnte, Tipolstituzione, PartitaIVA, NomeResponsabile) VALUE
        (LAST_INSERT_ID(), ilTipolstituzione, laPartitaIVA, ilNomeResponsabile);
    COMMIT;
    SET AUTOCOMMIT=oldac;
    END $
    DELIMITER ;

CREATE TABLE SOCIETA (
    CodEnte          INT PRIMARY KEY,
    70 PartitaIVA      CHAR(11),
    LegaleRappresentante VARCHAR(100),
    CONSTRAINT fk_Societa_Enti FOREIGN KEY (CodEnte) REFERENCES ENTI(CodEnte) ON
    DELETE CASCADE
) ENGINE=InnoDB;

DELIMITER $
DROP PROCEDURE IF EXISTS InserisciSocieta $

CREATE PROCEDURE InserisciSocieta (IN
    80 ilNome          VARCHAR(255),
    laPartitaIVA      CHAR(11),
    ilLegaleRappresentante VARCHAR(100) )
BEGIN
    DECLARE oldac BOOL;
    SELECT @@AUTOCOMMIT INTO oldac;
    SET AUTOCOMMIT=0;
    START TRANSACTION;
    INSERT INTO ENTI(Nome) VALUE (ilNome);
    INSERT INTO SOCIETA(CodEnte, PartitaIVA, LegaleRappresentante) VALUE
        (LAST_INSERT_ID(), laPartitaIVA, ilLegaleRappresentante);
    90 COMMIT;
    SET AUTOCOMMIT=oldac;
    END $
    DELIMITER ;

CREATE TABLE PERSONEFISCHE (
    CodEnte          INT PRIMARY KEY,
    CodiceFiscale    CHAR(16),
    DataNascita      DATE,
    LuogoNascita     VARCHAR(100),
    CONSTRAINT fk_PersoneFisiche_Enti FOREIGN KEY (CodEnte) REFERENCES ENTI(CodEnte)
    ON DELETE CASCADE
100 ) ENGINE=InnoDB;

DELIMITER $
DROP PROCEDURE IF EXISTS InserisciPersonaFisica $

```

```

110 CREATE PROCEDURE InserisciPersonaFisica (IN
    ilNome          VARCHAR(255),
    ilCodiceFiscale CHAR(16),
    laDataNascita   DATE,
    ilLuogoNascita  VARCHAR(100) )
BEGIN
    DECLARE oldac BOOL;
    SELECT @@AUTOCOMMIT INTO oldac;
    SET AUTOCOMMIT=0;
    START TRANSACTION;
    INSERT INTO ENTI(Nome) VALUE (ilNome);
    INSERT INTO PERSONEFISCHE(CodEnte, CodiceFiscale, DataNascita, LuogoNascita) VALUE
        (LAST_INSERT_ID(), ilCodiceFiscale, laDataNascita, ilLuogoNascita);
    COMMIT;
    SET AUTOCOMMIT=oldac;
END $

120 CREATE TRIGGER CodFiscMaiuscI
BEFORE INSERT ON PERSONEFISCHE
FOR EACH ROW
BEGIN
    SET NEW.CodiceFiscale=UPPER(NEW.CodiceFiscale);
END $
CREATE TRIGGER CodFiscMaiuscU
BEFORE UPDATE ON PERSONEFISCHE
FOR EACH ROW
130 BEGIN
    SET NEW.CodiceFiscale=UPPER(NEW.CodiceFiscale);
END $
DELIMITER ;

CREATE TABLE GIUDICI (
    CodEnte    INT PRIMARY KEY,
    TipoGiudice ENUM('Togato', 'Onorario') NOT NULL,
    CONSTRAINT fk_Giudici_PersoneFisiche FOREIGN KEY (CodEnte) REFERENCES
        PERSONEFISCHE(CodEnte) ON DELETE CASCADE
) ENGINE=InnoDB;

140 CREATE TABLE COLLABORATORI (
    CodEnte    INT PRIMARY KEY,
    Qualifica  ENUM('Avvocato', 'PraticanteConPatrocinio', 'Praticante', 'Segretario')
        NOT NULL,
    Retribuzione DOUBLE DEFAULT 0,
    Password   CHAR(41) NOT NULL,
    CONSTRAINT fk_Collaboratori_PersoneFisiche FOREIGN KEY (CodEnte) REFERENCES
        PERSONEFISCHE(CodEnte) ON DELETE CASCADE
) ENGINE=InnoDB;

```

Fin qui sono state create le tabelle ERRORI, RECAPITI e quelle relative alla gerarchia degli ENTI. Sono state definiti i *trigger* CodFiscMaiuscI e CodFiscMaiuscU, per assicurarsi che il codice fiscale sia in maiuscolo, e le procedure InserisciIstituzione, InserisciSocieta e InserisciPersonaFisica di evidente scopo.

```

150 CREATE TABLE TRIBUNALI (
    CodTribunale INT PRIMARY KEY AUTO_INCREMENT,
    Luogo        VARCHAR(100) NOT NULL,
    TipoTribunale VARCHAR(100) NOT NULL,
    Indirizzo    VARCHAR(100),
    Telefono     VARCHAR(100),
    UNIQUE (Luogo, TipoTribunale)
) ENGINE=InnoDB;

```

```

CREATE TABLE LAVORAlN (
  CodEnte INT,
  CodTribunale INT,
  PRIMARY KEY (CodEnte, CodTribunale),
  CONSTRAINT fk_LavoralN_Giudici FOREIGN KEY (CodEnte) REFERENCES GIUDICI(CodEnte) ON
160   DELETE CASCADE,
  CONSTRAINT fk_LavoralN_Tribunali FOREIGN KEY (CodTribunale) REFERENCES
    TRIBUNALI(CodTribunale)
) ENGINE=InnoDB;

DELIMITER $
DROP PROCEDURE IF EXISTS ContolloLavoralN;
CREATE PROCEDURE ContolloLavoralN (IN TipoGiudice VARCHAR(100), TipoTribunale
  VARCHAR(100))
170 BEGIN
  IF TipoGiudice='Togato' AND TipoTribunale='Giudice di Pace'
    THEN INSERT INTO ERRORI(CodErr) VALUE (24);
  END IF;
END $

CREATE TRIGGER ContolloLavoralNl
BEFORE INSERT ON LAVORAlN
FOR EACH ROW
CALL ContolloLavoralN(
  (SELECT g.TipoGiudice FROM GIUDICI g WHERE g.CodEnte=NEW.CodEnte),
  (SELECT t.TipoTribunale FROM TRIBUNALI t WHERE t.CodTribunale=NEW.CodTribunale)) $
180 CREATE TRIGGER ContolloLavoralNu
BEFORE UPDATE ON LAVORAlN
FOR EACH ROW
CALL ContolloLavoralN(
  (SELECT g.TipoGiudice FROM GIUDICI g WHERE g.CodEnte=NEW.CodEnte),
  (SELECT t.TipoTribunale FROM TRIBUNALI t WHERE t.CodTribunale=NEW.CodTribunale)) $

CREATE TRIGGER ContolloGiudiceU
BEFORE UPDATE ON GIUDICI
FOR EACH ROW
190 BEGIN
  DECLARE unCodTribunale INT;
  DECLARE Fatto BOOL DEFAULT 0;
  DECLARE curs CURSOR FOR
    (SELECT li.CodTribunale
     FROM LAVORAlN li
     WHERE li.CodEnte=NEW.CodEnte);
  DECLARE CONTINUE HANDLER FOR NOT FOUND SET Fatto=1;
  OPEN curs;
  REPEAT
    200   FETCH curs INTO unCodTribunale;
    CALL ContolloLavoralN(NEW.TipoGiudice,
      (SELECT t.TipoTribunale FROM TRIBUNALI t WHERE t.CodTribunale=unCodTribunale));
  UNTIL Fatto END REPEAT;
  CLOSE curs;
END $

CREATE TRIGGER ContolloTribunaleU
BEFORE UPDATE ON TRIBUNALI
FOR EACH ROW
210 BEGIN
  DECLARE unCodEnte INT;
  DECLARE Fatto BOOL DEFAULT 0;
  DECLARE curs CURSOR FOR
    (SELECT li.CodEnte
     FROM LAVORAlN li

```

```

WHERE li.CodTribunale=NEW.CodTribunale);
DECLARE CONTINUE HANDLER FOR NOT FOUND SET Fatto=1;
OPEN curs;
REPEAT
220   FETCH curs INTO unCodEnte;
      CALL ControlloLavoralN(
        (SELECT g.TipoGiudice FROM GIUDICI g WHERE g.CodEnte=unCodEnte),
        NEW.TipoTribunale);
      UNTIL Fatto END REPEAT;
      CLOSE curs;
END $
DELIMITER ;

```

In questo frammento sono state definite le tabelle TRIBUNALI e LAVORALN; inoltre, è stata definita la procedura ControlloLavoralN che accetta come parametri un tipo di giudice e un tipo di tribunale e, se sono incompatibili (cioè, corrispondenti rispettivamente a 'Togato' e 'Giudice di Pace') genera un errore (inserendo una chiave duplicata nella tabella Errori); infine, vengono definiti quattro *trigger* che invocano la funzione di cui sopra per controllare il rispetto del vincolo tra giudici e tribunali in caso di inserimento o aggiornamento di LAVORALN (rispettivamente ControlloLavoralNl e ControlloLavoralNu), aggiornamento di GIUDICI (ControlloGiudiceU) e aggiornamento di TRIBUNALI (ControlloTribunaleU).

```

230 CREATE TABLE CAUSE (
      CodCausa      INT PRIMARY KEY AUTO_INCREMENT,
      Stato         VARCHAR(100),
      RG            VARCHAR(11) NOT NULL,
      CodTribunale  INT NOT NULL,
      CodGiudice    INT NOT NULL,
      UNIQUE (RG, CodTribunale),
      CONSTRAINT fk_Cause_Tribunali FOREIGN KEY (CodTribunale) REFERENCES
        TRIBUNALI(CodTribunale),
      CONSTRAINT fk_Cause_Giudici FOREIGN KEY (CodGiudice) REFERENCES GIUDICI(CodEnte)
) ENGINE=InnoDB;

240 DELIMITER $
      CREATE TRIGGER ControlloCausaGiudiceI
      BEFORE INSERT ON CAUSE
      FOR EACH ROW
      BEGIN
        IF (SELECT MAX(l.CodTribunale=NEW.CodTribunale)
            FROM LAVORALN l
            WHERE l.CodEnte=NEW.CodGiudice)=0 THEN
          INSERT INTO ERRORI(CodErr) VALUE(25);
        END IF;
      END $
250 CREATE TRIGGER ControlloCausaGiudiceU
      BEFORE UPDATE ON CAUSE
      FOR EACH ROW
      BEGIN
        IF (SELECT MAX(l.CodTribunale=NEW.CodTribunale)
            FROM LAVORALN l
            WHERE l.CodEnte=NEW.CodGiudice)=0 THEN
          INSERT INTO ERRORI(CodErr) VALUE(25);
        END IF;
      END $
260 CREATE TRIGGER ControlloLavoralND
      BEFORE DELETE ON LAVORALN
      FOR EACH ROW
      BEGIN

```

```

IF (SELECT COUNT(*)
FROM CAUSE c
WHERE c.CodTribunale=OLD.CodTribunale AND c.CodGiudice=OLD.CodEnte)!=0 THEN
INSERT INTO ERRORI(CodErr) VALUE(25);
END IF;
END $
DELIMITER ;

```

Qui è stata definita la tabella CAUSE, oltre ad alcuni *trigger* che controllano il vincolo che un giudice di una causa lavori nel tribunale in cui si celebra quella causa; essi si attivano in caso di inserimento e aggiornamento di CAUSE (rispettivamente ControlloCausaGiudiceI e ControlloCausaGiudiceU) e cancellazione in LAVORALN (ControlloLavoralND).

```

270 CREATE TABLE COINVOLTOIN (
CodEnte INT,
CodCausa INT,
Ruolo VARCHAR(20) NOT NULL,
Cliente BOOL NOT NULL,
PRIMARY KEY (CodEnte, CodCausa),
CONSTRAINT fk_Coinvoltoln_Enti FOREIGN KEY (CodEnte) REFERENCES ENTI(CodEnte),
CONSTRAINT fk_Coinvoltoln_Cause FOREIGN KEY (CodCausa) REFERENCES
CAUSE(CodCausa) ON DELETE CASCADE
) ENGINE=InnoDB;

280 CREATE TABLE UDIENZE (
CodUdienza INT PRIMARY KEY AUTO_INCREMENT,
CodCausa INT NOT NULL,
Data DATE,
TipoUdienza VARCHAR(100),
CONSTRAINT fk_Udienze_Cause FOREIGN KEY (CodCausa) REFERENCES CAUSE(CodCausa)
ON DELETE CASCADE
) ENGINE=InnoDB;

CREATE TABLE DISCUSSADA (
290 CodUdienza INT,
CodEnte INT,
PRIMARY KEY (CodUdienza, CodEnte),
CONSTRAINT fk_DiscussaDa_Udienze FOREIGN KEY (CodUdienza) REFERENCES
UDIENZE(CodUdienza) ON DELETE CASCADE,
CONSTRAINT fk_DiscussaDa_Collaboratori FOREIGN KEY (CodEnte) REFERENCES
COLLABORATORI(CodEnte) ON DELETE CASCADE
) ENGINE=InnoDB;

DELIMITER $
DROP FUNCTION IF EXISTS DatiEnte $
CREATE FUNCTION DatiEnte(unCodEnte INT) RETURNS VARCHAR(255) DETERMINISTIC
BEGIN
300 DECLARE ris VARCHAR(255);
SELECT CASE

WHEN s.CodEnte IS NOT NULL THEN CONCAT( -- società
'<br />(Società) ',
IF(s.PartitaIVA IS NOT NULL, CONCAT('Partita IVA: ', s.PartitaIVA, ', ', '), ''),
IF(s.LegaleRappresentante IS NOT NULL, CONCAT('Legale Rappresentante: ',
s.LegaleRappresentante, ', ', '), ''))
WHEN i.CodEnte IS NOT NULL THEN CONCAT( -- istituzione
'<br />(Istituzione) ',
IF(i.Tipolstituzione IS NOT NULL, CONCAT('Tipo Istituzione: ', i.Tipolstituzione, ', ', '),
''),
310 IF(i.PartitaIVA IS NOT NULL, CONCAT('Partita IVA: ', i.PartitaIVA, ', ', '), ''),

```

```

        IF(i.NomeResponsabile IS NOT NULL, CONCAT('Nome Responsabile: ',
            i.NomeResponsabile, ', ', '), ''))
    WHEN p.CodEnte IS NOT NULL THEN CONCAT( -- persona fisica
        '<br />(Persona Fisica) ',
        IF(p.CodiceFiscale IS NOT NULL, CONCAT('Codice Fiscale: ', p.CodiceFiscale, ', ', '), ''),
        IF(p.DataNascita IS NOT NULL, CONCAT('Data Nascita: ', DATE_FORMAT(p.DataNascita,
            '%d/%m/%Y'), ', ', '), ''),
        IF(p.LuogoNascita IS NOT NULL, CONCAT('Luogo Nascita: ', p.LuogoNascita, ', ', '), ''),
        IF(c.CodEnte IS NOT NULL, CONCAT( -- collaboratore
            '<br />(Collaboratore) ',
            IF(c.Qualifica IS NOT NULL, CONCAT('Qualifica: ', c.Qualifica, ', ', '), ''),
            IF(c.Retribuzione IS NOT NULL, CONCAT('Retribuzione: ', c.Retribuzione, ', ', '), '')),
        IF(g.CodEnte IS NOT NULL, CONCAT( -- giudice
            '<br />(Giudice) ',
            IF(g.TipoGiudice IS NOT NULL, CONCAT('TipoGiudice: ', g.TipoGiudice, ', ', '), '')),
            ''))

    END INTO ris
    FROM ENTI e
        LEFT JOIN SOCIETA s USING (CodEnte)
        LEFT JOIN ISTITUZIONI i USING (CodEnte)
        LEFT JOIN PERSONEFISCHE p USING (CodEnte)
        LEFT JOIN COLLABORATORI c USING (CodEnte)
        LEFT JOIN GIUDICI g USING (CodEnte)
    WHERE e.CodEnte=unCodEnte;
    RETURN SUBSTRING(ris FROM 7 FOR LENGTH(ris)-8);
END $

DROP FUNCTION IF EXISTS TipoEnte $
CREATE FUNCTION TipoEnte(unCodEnte INT) RETURNS CHAR(3) DETERMINISTIC
BEGIN
    DECLARE ris CHAR(3);
    SELECT CASE
        WHEN s.CodEnte IS NOT NULL THEN 's'
        WHEN i.CodEnte IS NOT NULL THEN 'i'
        WHEN p.CodEnte IS NOT NULL THEN CONCAT('p',
            IF(c.CodEnte IS NOT NULL, 'c', ''),
            IF(g.CodEnte IS NOT NULL, 'g', ''))
    END INTO ris
    FROM ENTI e
        LEFT JOIN SOCIETA s USING (CodEnte)
        LEFT JOIN ISTITUZIONI i USING (CodEnte)
        LEFT JOIN PERSONEFISCHE p USING (CodEnte)
        LEFT JOIN COLLABORATORI c USING (CodEnte)
        LEFT JOIN GIUDICI g USING (CodEnte)
    WHERE e.CodEnte=unCodEnte;
    RETURN ris;
END $

DROP FUNCTION IF EXISTS DistMediaUdienze $
CREATE FUNCTION DistMediaUdienze(unCodCausa INT) RETURNS INT DETERMINISTIC
BEGIN
    DECLARE fatto BOOL DEFAULT FALSE;
    DECLARE olddat DATE;
    DECLARE newdat DATE;
    DECLARE num INT;
    DECLARE somma INT DEFAULT 0;
    DECLARE curs CURSOR FOR
        SELECT u.Data
        FROM UDIENZE u
        WHERE u.CodCausa=unCodCausa
        ORDER BY u.Data;

```

```

370 DECLARE CONTINUE HANDLER FOR NOT FOUND
    SET fatto=TRUE;

    SET num = (SELECT COUNT(*)
        FROM UDIENZE u
        WHERE u.CodCausa=unCodCausa);
    IF num <= 1 THEN RETURN 0; END IF;

    OPEN curs;
    FETCH curs INTO newdat;
380 REPEAT
        SET olddat = newdat;
        FETCH curs INTO newdat;
        SET somma = somma + DATEDIFF(newdat, olddat);
    UNTIL fatto END REPEAT;
    RETURN somma/num;
END $

DELIMITER ;

390 CREATE OR REPLACE VIEW VENTI AS
    SELECT *, TipoEnte(CodEnte) AS Tipo, DatiEnte(CodEnte) AS Dati
    FROM ENTI;
CREATE OR REPLACE VIEW VTRIBUNALI AS
    SELECT t.*, COUNT(DISTINCT li.CodEnte) AS 'NumeroGiudici', COUNT(DISTINCT
        c.CodCausa) AS 'NumeroCause'
    FROM TRIBUNALI t NATURAL LEFT JOIN LAVORAIN li NATURAL LEFT JOIN CAUSE c
    GROUP BY t.CodTribunale, t.Luogo, t.TipoTribunale, t.Indirizzo, t.Telefono;
CREATE OR REPLACE VIEW VCAUSE AS
    SELECT c.*, COUNT(e.CodEnte) AS NumeroParti, GROUP_CONCAT(IF(ci.Cliente, '<b>', ''),
        e.Nome, IF(ci.Cliente, '</b>', ''), ' (', ci.Ruolo, ') ' ORDER BY ci.Ruolo, e.Nome
        SEPARATOR ', ') AS DatiParti
    FROM CAUSE c NATURAL LEFT JOIN COINVOLTOIN ci NATURAL LEFT JOIN ENTI e
400 GROUP BY c.CodCausa, c.Stato, c.RG, c.CodTribunale, c.CodGiudice
    ORDER BY c.CodCausa;
CREATE OR REPLACE VIEW VUDIENZE AS
    SELECT u.*, GROUP_CONCAT(e.Nome, ' (', e.CodEnte, ') ' ORDER BY e.CodEnte
        SEPARATOR ', ') AS Partecipanti
    FROM UDIENZE u NATURAL LEFT JOIN DISCUSSADA dd NATURAL LEFT JOIN ENTI e
    GROUP BY u.CodUdienza, u.CodCausa, u.Data, u.TipoUdienza
    ORDER BY u.CodCausa;

REHASH;

410 SOURCE insert.mysql; -- popola le tabelle

```

In questo frammento, che conclude l'implementazione della base di dati, vengono definite le ultime tabelle (COINVOLTOIN, UDIENZE e DISCUSSADA) ed alcune funzioni. La funzione DatiEnte prende come parametro un CodEnte e restituisce una stringa formattata in HTML che descrive l'ente identificato da quel codice (in particolare le specializzazioni relative alle sottoclassi); la funzione TipoEnte funziona in maniera simile, ma si limita a restituire un gruppo di caratteri contenenti 's' se l'ente identificato dal codice è una società, 'i' se è una istituzione e così via.

La funzione DistMediaUdienze è un esempio di implementazione di una ricerca con la logica a cursore. Restituisce la distanza media tra due udienze della causa identificata dal codice passato come argomento. È quindi equivalente a

```

SELECT DATEDIFF(MAX(Data), MIN(Data))/COUNT(*)
FROM UDIENZE

```


WHERE CodCausa=unCodCausa

Infine, vengono definite alcune *view* utili per l'interfaccia. Lo *script* insert.mysql contiene le istruzioni per popolare le tabelle, che non sono qui riportate per brevità.

5 Query

Seguono alcuni esempi di *query*.

1. Per quei collaboratori che hanno discusso da soli un'udienza nel giorno del loro compleanno, nome di ogni cliente e del collaboratore e data.

```
SELECT e.Nome AS Cliente, ec.Nome AS Collaboratore, u.Data
FROM DISCUSSADA dd NATURAL JOIN UDienze u
JOIN COINVOLTOIN ci ON (ci.Cliente AND ci.CodCausa=u.CodCausa)
JOIN ENTI e ON ci.CodEnte=e.CodEnte
JOIN PERSONEFISCHE cpf ON cpf.CodEnte=dd.CodEnte
JOIN ENTI ec ON ec.CodEnte=dd.CodEnte
WHERE (DAY(u.Data)=DAY(cpf.DataNascita)) AND (MONTH(u.Data)=MONTH(cpf.DataNascita))
AND NOT EXISTS
(SELECT *
FROM DISCUSSADA dd2
WHERE dd2.CodUdienza=dd.CodUdienza AND dd.CodEnte!=dd2.CodEnte);
```

Output:

```
+-----+-----+-----+
| Cliente          | Collaboratore    | Data          |
+-----+-----+-----+
| Pircher S.a.s.   | Trabalza Lamberto | 2005-12-16 |
| Marchesini Bino | Capovilla Nicla   | 2011-05-03 |
+-----+-----+-----+
2 rows in set (0,01 sec)
```

2. Nome e recapiti telefonici per quegli enti che hanno un telefono del distretto di Padova (prefisso: 049) e compaiono insieme a 'Busana Astianatte' (CodEnte=42) in una causa con un ruolo diverso dal suo.

```
SELECT e.Nome, r.Recapito
FROM COINVOLTOIN ci1
JOIN COINVOLTOIN ci2 ON (ci1.CodCausa=ci2.CodCausa AND ci1.Ruolo!=ci2.Ruolo)
JOIN ENTI e ON ci2.CodEnte=e.CodEnte JOIN RECAPITI r ON (r.CodEnte=ci2.CodEnte)
WHERE ci1.CodEnte=42 AND r.TipoRecapito='Telefono' AND r.Recapito LIKE '049%';
```

Output:

```
+-----+-----+
| Nome          | Recapito         |
+-----+-----+
| Cantarella S.p.A. | 0496525853 |
| Valdo S.n.c.    | 0492759261 |
+-----+-----+
2 rows in set (0,00 sec)
```

3. Per quei giudici che giudicano cause in tribunali diversi che coinvolgono lo stesso ente, nome del giudice e dell'ente.

```
CREATE VIEW GiudiciEntiDueTrib AS
(SELECT g.CodEnte AS CodGiudice, ci1.CodEnte
FROM GIUDICI g, COINVOLTOIN ci1, COINVOLTOIN ci2
WHERE ci1.CodEnte=ci2.CodEnte AND EXISTS
(SELECT *
FROM CAUSE c1
WHERE c1.CodGiudice=g.CodEnte AND c1.CodCausa=ci1.CodCausa AND EXISTS
(SELECT *
FROM CAUSE c2
WHERE c1.CodTribunale>c2.CodTribunale AND c2.CodGiudice=g.CodEnte AND
c2.CodCausa=ci2.CodCausa)));
SELECT eg.Nome AS NomeGiudice, ee.Nome AS NomeParte
FROM GiudiciEntiDueTrib ge NATURAL JOIN ENTI ee
JOIN ENTI eg ON ge.CodGiudice=eg.CodEnte;
DROP VIEW GiudiciEntiDueTrib;
```

Output:

```
+-----+-----+
| NomeGiudice | NomeParte |
+-----+-----+
| Nadalino Telica | Savella Fiorella |
+-----+-----+
1 row in set (0,01 sec)
```

4. CodCausa, RG e nome del cliente per quelle cause alle cui udienze di tipo 'Comparizione' hanno partecipato almeno 4 diversi collaboratori.

```
SELECT u.CodCausa, c.RG, e.Nome AS NomeCliente
FROM UDIENZE u NATURAL JOIN DISCUSSADA dd NATURAL JOIN CAUSE c JOIN COINVOLTOIN ci ON
(ci.Cliente AND ci.CodCausa=c.CodCausa)
JOIN ENTI e ON (ci.CodEnte=e.CodEnte)
WHERE u.TipoUdienza='Comparizione'
GROUP BY u.CodCausa
HAVING COUNT(DISTINCT dd.CodEnte)>=4;
```

Output:

```
+-----+-----+-----+
| CodCausa | RG | NomeCliente |
+-----+-----+-----+
| 4 | 6170/2005 | Puglioli Aristo |
| 5 | 8523/2005 | Somaschi Cristaldo |
| 7 | 20283/2005 | Comune di Dolo |
| 11 | 1202/2005 | Manfron Amadeo |
| 14 | 5156/2005 | Savella Fiorella |
| 15 | 915/2005 | Campus Liana |
| 16 | 27581/2007 | Marizza Eutalia |
| 17 | 23993/2005 | Pittaro S.r.l. |
| 20 | 9318/2005 | Marchesini Bino |
+-----+-----+-----+
9 rows in set (0,01 sec)
```

5. Per quelle cause che coinvolgono esattamente due istituzioni e nessun altro, nome delle due istituzioni e codice della causa.

```

CREATE VIEW IstituzioniCoinvolte AS
SELECT *
FROM COINVOLTOIN ci NATURAL JOIN ISTITUZIONI i;
CREATE VIEW IstituzioniCoinvolteSole AS
(SELECT ic1.CodEnte AS Cod1, ic2.CodEnte AS Cod2, ic1.CodCausa
FROM IstituzioniCoinvolte ic1 JOIN IstituzioniCoinvolte ic2 ON (ic1.CodCausa=ic2.CodCausa
AND ic1.CodEnte>ic2.CodEnte)
WHERE NOT EXISTS
(SELECT *
FROM COINVOLTOIN ci
WHERE ci.CodCausa=ic1.CodCausa AND ci.CodEnte!=ic1.CodEnte AND
ci.CodEnte!=ic2.CodEnte));
SELECT e1.Nome AS PrimaParte, e2.Nome AS SecondaParte, ics.CodCausa
FROM IstituzioniCoinvolteSole ics
JOIN ENTI e1 ON (ics.Cod1=e1.CodEnte)
JOIN ENTI e2 ON (ics.Cod2=e2.CodEnte);
DROP VIEW IstituzioniCoinvolte, IstituzioniCoinvolteSole;

```

Output:

```

+-----+-----+-----+
| PrimaParte | SecondaParte | CodCausa |
+-----+-----+-----+
| Provincia di Venezia | Comune di Dolo | 12 |
+-----+-----+-----+
1 row in set (0,00 sec)

```

6. Per ogni causa, nome dei collaboratori che hanno partecipato a più udienze e loro numero di partecipazioni ad udienze di quella causa, o NULL per le cause senza udienze.

```

CREATE VIEW CollaboratoriCause AS
SELECT u.CodCausa, dd.CodEnte, COUNT(*) AS NumPartecipazioni
FROM DISCUSSADA dd NATURAL JOIN UDIENZE u GROUP BY u.CodCausa, dd.CodEnte;
SELECT cc1.CodCausa, e.Nome, NumPartecipazioni
FROM CollaboratoriCause cc1 NATURAL JOIN ENTI e
WHERE NOT EXISTS
(SELECT *
FROM CollaboratoriCause cc2
WHERE cc1.CodCausa=cc2.CodCausa AND cc2.NumPartecipazioni>cc1.NumPartecipazioni)
UNION
SELECT c.CodCausa, NULL, NULL
FROM CAUSE c NATURAL LEFT JOIN UDIENZE u
WHERE u.CodUdienza IS NULL
ORDER BY CodCausa;
DROP VIEW CollaboratoriCause;

```

Output:

```

+-----+-----+-----+
| CodCausa | Nome | NumPartecipazioni |
+-----+-----+-----+
| 1 | Trabalza Lamberto | 3 |
| 1 | Capovilla Nicla | 3 |
| 1 | Gavazzi Ataleo | 3 |
| 2 | Lomi Adina | 5 |
| 3 | NULL | NULL |

```

4	Lomi Adina	7
5	Gavazzi Ataleo	6
6	Sansone Iginio	5
6	Gavazzi Ataleo	5
7	Sansone Iginio	6
8	Trabalza Lamberto	3
9	Spaccatrosi Abramo	6
10	Capovilla Nicla	5
11	Trabalza Lamberto	5
11	Sansone Iginio	5
11	Gavazzi Ataleo	5
12	Sansone Iginio	4
13	NULL	NULL
14	Trabalza Lamberto	5
14	Gavazzi Ataleo	5
15	Sansone Iginio	5
15	Gavazzi Ataleo	5
16	Trabalza Lamberto	4
16	Tassone Delfina	4
17	Capovilla Nicla	5
18	NULL	NULL
19	Spaccatrosi Abramo	4
20	Tassone Delfina	5
20	Capovilla Nicla	5

-----+

29 rows in set (0,01 sec)

6 Interfaccia web

L'interfaccia usa tre tipi di *file* sorgente: librerie (che risiedono nella cartella `php-incl`), pagine di servizio (identificate dal prefisso `'_'`) e pagine di interfaccia vere e proprie.

Le librerie sono semplici raccolte di funzioni; vengono incluse dagli altri *files* sorgente quando richieste.

Le pagine di servizio sono `_insert.php`, `_update.php`, `_delete.php`, `_log-check.php`, `_cambiapassword.php` e `_nuovoente.php`. Esse accettano parametri passati tramite metodo `POST`, compiono (se necessario) operazioni varie sulla base di dati, e restituiscono il risultato alla pagina che le ha richiamate reindirizzando il *browser* al *referer*, passando il risultato tramite il parametro `GET errore`; nessuna di queste pagine produce direttamente un *output*.

Infine, le pagine di interfaccia producono l'interfaccia *web* che vede l'utente.

L'architettura dell'interfaccia *web* è qui descritta *file* per *file*, a partire dalle librerie.

6.1 Librerie

6.1.1 baseurl.php

Il *file* `baseurl.php` contiene solo una semplice funzione che restituisce l'indirizzo della cartella di base del sito.

```
<?php
function urlbasesito() {
```

```

        return 'http' . (isset($_SERVER['HTTPS']) ? 's':'') . '://' . $_SERVER['HTTP_HOST'] .
            '/basidati/~tfrasset';
    }
    ?>

```

6.1.2 misc.php

Il file di libreria `misc.php` contiene varie funzioni. In particolare:

- `head_trans($title)`: manda in *output* l'intestazione della pagina, con titolo `$title`, con informazioni sugli stili e uno *script javascript* che mostra o nasconde un elemento `<div>`; inoltre produce le informazioni riguardo all'utente collegato (il numero tra parentesi indica il livello di autorizzazione corrente, 1 per i segretari, 2 per i praticanti, 3 per gli avvocati) e la barra di navigazione (chiamando la `nav()`).
- `tail_trans()`: termina la sezione dedicata al contenuto e chiude la pagina.
- `nav()`: produce la barra di navigazione a sinistra.
- `afieldset($legend, $divid=NULL, $stile='')`: inizia un elemento `<div>` con l'*id* e lo stile specificato, ed all'interno un elemento `<fieldset>` di legenda specificata.
- `cfieldset()`: chiude i *tag* aperti dalla funzione precedente.
- `dbconn(&$conn)`: crea una connessione al database e la associa al parametro passato per riferimento.
- `get_warn($conn)`: interroga il DBMS sugli eventuali *warning* prodotti dalla ultima interrogazione, produce in alcuni casi messaggi di errore più chiari e restituisce il risultato come stringa.
- `err($err, $ok='Query OK')`: reindirizza il *browser* alla pagina precedente (*referer*) impostando il parametro GET errore a `$err` o, se quest'ultima è vuota, a `$ok`.
- `stampa_err()`: mostra i messaggi passati dalla funzione precedente dentro ad un elemento `<fieldset>`.
- `queryset($vecchiaurl, $chiave, $valore)`: sostituisce o inserisce il parametro `$chiave` impostandolo a `$valore` in `$vecchiaurl`.
- `ris2array($ris)`: restituisce un *array* contenente il risultato di una *query*; il parametro di *input* è una *resource* restituita da una **mysql_query**.
- `auth()`: restituisce il livello di autorizzazione dell'utente se è *loggato*, altrimenti reindirizza il *browser* alla pagina di *login*.

```

0 <?php
  require_once('../baseurl.php');

  function head_trans($title) {
    echo<<<END
    <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
      "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

```

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>$title</title>
10 <script type="text/javascript">
    function toggle(id) {
        var e = document.getElementById(id);
        if (e.style.display == '')
            e.style.display = 'none';
        else
            e.style.display = '';
        }
    </script>
    <style type="text/css">
20     a {
        color:blue
    }
    </style>
</head>

<body>
<div id="container">\n
END;

30 if (isset($_SESSION['login'])) {
    echo "    <p style=\"text-align:right\">Salve <a href=\"dettagliente.php?CodEnte=\" .
        $_SESSION['login'] . "\"><b>\" . $_SESSION['nome'] . \"</b></a><sup>{ \" .
        $_SESSION['livello'] . \"}</sup> (<a href=\"\" . urlbasesito() . \"/logout.php\">logout</a>)<\/p>\n";
    echo "    <div id=\"nav\" style=\"float:left; width:150px\">\n";
    nav();
    echo<<<END
        </div>
        <div id="cont" style="margin-left:160px">\n
    END;
    }
    }
40

function tail_trans() {
    if (isset($_SESSION['login']))
    echo<<<END
        </div>
        <div id="foot" style="clear:both">
        <p>
            <a href="http://validator.w3.org/check?uri=referer"></a>
        </p>
        </div>\n
50 END;
    echo<<<END

        </div>
    </body>
</html>
END;
}

60 function nav() {
    echo "\n\n";
    $lista = array(
        #array(0=>nome, 1=>indirizzo, 2=>indentazione, 3=>visibile, 4=>livelloautorizzazione)
        array('Cause', 'listacause.php', 0, true, 1),

```

```

        array('&#x2295; Nuova', 'nuovacausa.php', 1, true, 2),
        array('Dettagli Causa', 'dettagli_causa.php', 1, false, 1),
array('Enti', 'listaenti.php', 0, true, 1),
        array('Istituzioni', 'listaenti.php?Tipo=Istituzione', 1, true, 1),
        array('Società', 'listaenti.php?Tipo=Società', 1, true, 1),
        array('Persone Fisiche', 'listaenti.php?Tipo=Persona+Fisica', 1, true, 1),
70         array('Giudici', 'listaenti.php?Tipo=Giudice', 1, true, 1),
        array('Collaboratori', 'listaenti.php?Tipo=Collaboratore', 1, true, 1),
        array('&#x2295; Nuovo', 'nuovoente.php', 1, true, 2),
        array('Dettagli Ente', 'dettagliente.php', 1, false, 1),
        array('Tribunali', 'listatribunali.php', 0, true, 1),
        array('&#x2295; Nuovo', 'nuovotribunale.php', 1, true, 2),
        array('Dettagli Tribunale', 'dettagli_tribunale.php', 1, false, 1)
    );
    $self = end(explode('/', $_SERVER['PHP_SELF']));
    $base = urlbasesito();
80     foreach ($lista as $i) if ($i[4] <= $_SESSION['livello']) {
        $dim = ($i[2] ? '' : 'font-size:1.2em;');
        $ind = ($i[2] ? 'margin-left:'. $i[2]*20 . 'px; ' : '');
        $wid = 150 - $i[2]*20 . 'px';
        $col = '#' . dechex(hexdec('555555') + hexdec('333333')*$i[2]);
        if ($self === $i[1] && !$i[3]) {
            echo "<div style=\"$dim font-weight:bold; $ind\">$i[0]</div>\n";
        } else if ($self === $i[1]) {
            echo "<a href=\"$base/$i[1]\" style=\"$dim font-weight:bold; color:black;
                display:block; width:$wid; $ind\">$i[0]</a>\n";
        } else if ($i[3]) {
90         echo "<a href=\"$base/$i[1]\" style=\"$dim color:$col; display:block;
                width:$wid; $ind\">$i[0]</a>\n";
        }
    }
    echo "\n\n";
}

function afieldset($legend, $divid=NULL, $stile='') {
    $id = (isset($divid) ? "id=\"$divid\"" : '');
    if ($stile) $stile = 'style="'. $stile . '";
    echo<<<END
100         <div $id $stile>
        <fieldset style="margin:10px 0px">
        <legend>$legend</legend>

    END;
}

function cfieldset() {
    echo "        </fieldset>\n";
    echo "    </div>\n";
110 }

function dbconn(&$conn) {
    $conn=mysql_connect('localhost', 'tfrasset', '67wyBK0u') or err('Connessione a mysql
        fallita');
    mysql_set_charset('utf8');
    mysql_select_db('tfrasset-PR');
}

function get_warn($conn) {
    $ris = mysql_query('SHOW WARNINGS;', $conn);
    $warn = '';
120     while ($rig = mysql_fetch_row($ris)) {
        if ($rig[1] == 1062) { #errore da trigger

```

```

$errn = preg_replace("/Duplicate entry '([0-9]+)' for key 'PRIMARY'/" , '\1',
    $rig[2], -1, $count);
if ($count) {
    $seris = mysql_query("SELECT e.Descrizione FROM Errori e WHERE
        e.CodErr=$errn;" , $conn);
    $serig = mysql_fetch_row($seris);
    $swarn .= "Errore di consistenza ($errn) : $serig[0] <br /> \n";
} else {
    $swarn .= "$rig[0] ($rig[1]): $rig[2]<br />\n";
}
130 } else if ($rig[1] = 1451) { #foreign key check failed
    $const = preg_replace("/^.*CONSTRAINT `([^\`]+)`.*$/", '\1', $rig[2]);
    switch($const) {
        case 'fk_CoinvoltoIn_Enti':
            $swarn .= "Cancella prima le Cause in cui questo Ente è coinvolto <br />
                \n";
            break;
        case 'fk_LavoraIn_Tribunali':
            $swarn .= "Cancella prima i Giudici di questo Tribunale <br /> \n";
            break;
140 case 'fk_Cause_Tribunali':
            $swarn .= "Cancella prima le Cause di questo Tribunale <br /> \n";
            break;
        case 'fk_Cause_Giudici':
            $swarn .= "Cancella prima le Cause di questo Giudice <br /> \n";
            break;
    }
    $swarn .= "$rig[0] ($rig[1]): $rig[2]<br />\n";
} else {
    $swarn .= "$rig[0] ($rig[1]): $rig[2]<br />\n";
150 }
}
return $swarn;
}

function err($err, $ok='Query OK') {
    if (!$err) $err=$ok;
    if (isset($_SERVER['HTTP_REFERER'])) {
        header("Location: " . queryset($_SERVER['HTTP_REFERER'], 'Errore', $err));
        die();
160 } else {
        die($err);
    }
}

function stampa_err() {
    if (isset($_GET['Errore'])) {
        echo "<fieldset>\n";
        $param = array(); #parametri per il link x
        foreach ($_GET as $k => $i)
170 if ($k != 'Errore') $param[$k] = $i;
        echo "<div style='text-align:right; float:right'> <a
            href='\"".$_SERVER['PHP_SELF']."'\"?\".http_build_query($param, '',
                '&';')."\"><b>&times;</b></a> </div>\n";
        echo " <legend>Messaggi</legend>\n";
        echo " <h3>\" . stripslashes($_GET['Errore']) . \"</h3>\n";
        echo "</fieldset>\n";
    }
}

function queryset($vecchiaurl, $chiave, $valore) {
    $urla = explode('?', $vecchiaurl, 2);
180 if (!isset($urla[1])) $urla[1] = '';

```



```

    parse_str($urla[1], $param);
    $param[$chiave] = $valore;
    return $urla[0] . '?' . http_build_query($param);
}

function ris2array($ris) {
    $rit = array();
    if (mysql_num_fields($ris) > 1) {
        while ($rig = mysql_fetch_assoc($ris))
            $rit[] = $rig;
    } else {
        while ($rig = mysql_fetch_row($ris))
            $rit[] = $rig[0];
    }
    return $rit;
}

function auth() {
    session_start();

    if (!isset($_SESSION['login'])) {
        header('Location: ' . urlbasecito() . '/login.php');
        die();
    }

    return $_SESSION['live110'];
}
?>

```

6.1.3 potab.php

Il file potab.php contiene le seguenti funzioni.

- sololettere(\$stringa): restituisce i caratteri alfabetici presenti nella stringa in *input*, nell'ordine.
- solonumeri(\$stringa): restituisce i caratteri numerici presenti nella stringa in *input*, nell'ordine.
- nonvirgolette(\$stringa): restituisce i caratteri 'sicuri' presenti nella stringa in *input*.
- ckchiavi(\$nometabella, \$chiavi): data una tabella e un *array* associativo, controlla se le chiavi dell'*array* sono le chiavi giuste per quella tabella.
- cklivello (\$nometabella, \$liv, \$op): controlla se il livello di autorizzazione \$liv è sufficiente per eseguire una operazione \$op sulla tabella \$nometabella.

```

0 <?php
function sololettere($stringa) {
    return preg_replace('/[^A-Za-z]/', '', $stringa);
}

function solonumeri($stringa) {
    return preg_replace('/[^0-9]/', '', $stringa);
}

10 function nonvirgolette($stringa) {
    return preg_replace("/['\"`\\n\\r\\\\\\\\]/", '', $stringa);
}

```

```

    }

    function ckchiavi($nometabella, $chiavi) {
        switch ($nometabella) {
            case 'Enti': case 'Societa': case 'Istituzioni':
            case 'PersoneFisiche': case 'Giudici': case 'Collaboratori':
                if (array_keys($chiavi) !== array('CodEnte')) err('Errore di selezione chiavi');
                break;
            case 'Recapiti':
20         if (!ksort($chiavi) || (array_keys($chiavi) !== array('CodEnte', 'Recapito'))) err('Errore
                di selezione chiavi');
                break;
            case 'Tribunali':
                if (array_keys($chiavi) !== array('CodTribunale')) err('Errore di selezione chiavi');
                break;
            case 'Cause':
                if (array_keys($chiavi) !== array('CodCausa')) err('Errore di selezione chiavi');
                break;
            case 'Udienze':
30         if (array_keys($chiavi) !== array('CodUdienza')) err('Errore di selezione chiavi');
                break;
            case 'LavoraIn':
                if (!ksort($chiavi) || (array_keys($chiavi) !== array('CodEnte', 'CodTribunale')))
                    err('Errore di selezione chiavi');
                break;
            case 'CoinvoltoIn':
                if (!ksort($chiavi) || (array_keys($chiavi) !== array('CodCausa', 'CodEnte'))) err('Errore
                    di selezione chiavi');
                break;
            case 'DiscussaDa':
                if (!ksort($chiavi) || (array_keys($chiavi) !== array('CodEnte', 'CodUdienza')))
                    err('Errore di selezione chiavi');
                break;
40         default:
            err('Tabella sconosciuta');
            break;
        }
    }

    function cklivello($nometabella, $liv, $op) {
        # si assume liv almeno 2
        if ($op === 'delete') {
50         switch ($nometabella) {
            case 'Recapiti':
            case 'DiscussaDa':
                break;
            case 'Societa':
            case 'Istituzioni':
            case 'PersoneFisiche':
                err('Livello di autorizzazione insufficiente');
                break;
            default:
                if ($liv <= 2) err('Livello di autorizzazione insufficiente');
60         }
        } else {
            switch ($nometabella) {
            case 'Collaboratori':
                if ($liv <= 2) err('Livello di autorizzazione insufficiente');
            default:
                break;
            }
        }
    }
}

```

6.1.4 arrayedit.php

Il file `arrayedit.php` definisce due funzioni. La seconda, `arrayedit`, prende in *input* un *array* che descrive un *record* (cioè un insieme di campi, ognuno dei quali ha un nome e un valore, e può avere altre caratteristiche come un'etichetta, un insieme di valori tra cui scegliere o il fatto di essere di tipo data), oltre ad un altro *array* di campi da passare come campi nascosti e altri parametri di visualizzazione; quando viene invocata, la funzione stampa il codice necessario per visualizzare il *form* descritto dai parametri di *input*. La maggior parte dei *form* presenti nell'interfaccia sono generati da questa funzione.

La prima funzione presente nel file, `assoc2proplist`, prende in *input* un *array* associativo come quelli restituiti dalla `mysql_fetch_assoc` e genera in *output* un *array* nel formato utilizzato dalla funzione `arrayedit`, pronto per memorizzare altre caratteristiche dei campi.

```

0  <?php
function assoc2proplist($arr) {
    $ris = array();
    foreach ($arr as $k => $i) {
        $ris[$k] = array ('valore' => $i);
    }
    return $ris;
}

function arrayedit ($arr, $altriparametri=array(), $abilitato=true, $nomepulsante='Aggiorna',
    $action='_update.php', $method='post') {
10  # arr è un array associativo di proprietà dei campi del tipo nomecampo => array('valore' => "",
    'data' => true, 'sceltamultipla' => array(...), 'menu' => array(...), 'etichetta' => "",
    'obbligatorio' => true, 'lunghezza' => num, 'maxlunghezza' => num, 'password' => true); solo
    'valore' è obbligatorio
    # menu e sceltamultipla sono associativi valore=>testo da mostrare
    # altriparametri è un array associativo di altri parametri da passare come campi nascosti

    $disab = $abilitato ? '' : 'disabled="disabled"';

    echo "<form action=\"$action\" method=\"$method\" style=\"margin:10px 0px\">\n";
    foreach ($altriparametri as $k => $i) {
        echo " <input type=\"hidden\" name=\"$k\" value=\"$i\" /> \n";
    }
20  echo " <table style=\"margin:10px 0px\">\n";

    foreach ($arr as $k => $c) {
        $i = $c['valore'];
        $obbl = (isset($c['obbligatorio']) ? '<sup>*</sup>' : '');
        $lung = (isset($c['lunghezza']) ? ('size="'. $c['lunghezza']. '"): ');
        $mlung = (isset($c['maxlunghezza']) ? ('maxlength="'. $c['maxlunghezza']. '"): ');
        $n = (isset($c['etichetta']) ? $c['etichetta']: $k);
        $rn = mt_rand(0, 999);

30  echo " <tr>\n";
    if (isset($c['sceltamultipla'])) { #scelta multipla
        echo " <td> $n:$obbl </td>\n";
        echo " <td>\n";
        foreach ($c['sceltamultipla'] as $c => $j) {
            $check = ((i==$c) ? "checked=\"checked\"": "");
            echo " <input $disab type=\"radio\" name=\"$k\" value=\"$c\"
                id=\"$k.$c.$rn\" $check /> <label for=\"$k.$c.$rn\">$j</label>\n";
        }
    }
}

```

```

    }
    echo "    </td>\n";
40 } else if (isset($c['menu'])) { #menu tendina
    echo "    <td> <label for=\"\$k.\$rn\">\$n:\$obb1</label> </td>\n";
    echo "    <td> <select \$disab name=\"\$k\" id=\"\$k.\$rn\"> \n";
    foreach ($c['menu'] as $c => $j) {
        $check = (($i==$c) ? "selected=\"selected\" : \"");
        echo "        <option value=\"\$c\" \$check>\$j</option>";
    }
    echo "    </select> </td>\n";

50 } else if (isset($c['data'])) { #data
    $i = ($i ? date('d/m/Y', strtotime($i)) : '');
    echo "    <td> <label for=\"\$k.\$rn\">\$n:\$obb1</label> </td>\n <td> <input
        \$disab type=\"text\" value=\"\$i\" id=\"\$k.\$rn\" name=\"-D-\$k\" \$lung
        maxlength=\"10\" /> (GG/MM/AAAA) </td> \n";

    } else if (isset($c['password'])) { #password
        echo "    <td> <label for=\"\$k.\$rn\">\$n:\$obb1</label> </td>\n <td> <input
            \$disab type=\"password\" value=\"\$i\" id=\"\$k.\$rn\"
            name=\"---password\" \$lung \$mlung /> </td> \n";

    } else { #testo
        echo "    <td> <label for=\"\$k.\$rn\">\$n:\$obb1</label> </td>\n <td> <input
            \$disab type=\"text\" value=\"\$i\" id=\"\$k.\$rn\" name=\"\$k\" \$lung
            \$mlung /> </td> \n";
    }
    echo "    </tr>\n";
60 }
echo " </table>";
if ($abilitato) echo " <input type=\"submit\" value=\"\$nomepulsante\" />\n";
echo "</form>\n";
}
?>

```

6.1.5 query2tab.php

Anche il file `query2tab.php` introduce due funzioni. La seconda, `query2tab`, prende in *input* il testo di una *query* e una connessione al DBMS e stampa il risultato della *query* in formato tabulare, rendendo le intestazioni delle colonne dei *link* per ordinare le righe della tabella. La funzione prende in *input* il testo della *query* e non il risultato della medesima perché così può richiedere al DBMS di applicare un ordinamento. I parametri di ordinamento sono passati tramite il metodo `GET`. Al fine di permettere più tabelle di questo tipo nella stessa pagina, ad ognuna istanza viene assegnato un numero che essa concatena al nome dei parametri per distinguere quelli destinati a sé da quelli destinati ad un'altra istanza di tabella. Inoltre, la funzione permette di aggiungere dei pulsanti alla destra della tabella, ripetuti per ogni riga, che compiono azioni prefissate applicate al contesto della riga in cui si trovano; i dati di questi pulsanti vengono passati come parametri.

La prima funzione, `creapulsante`, serve appunto per stampare il codice necessario per questi pulsanti (che sono in realtà dei *form* con tutti i parametri nascosti e solo il pulsante visibile).

```

0 <?php
require_once('potab.php');

function creapulsante(&$action, &$method, &$nomecampi, &$valcampi, &$testosulbottone) {

```

```

# crea un pulsante per ogni riga della tabella per fare qualcosa
echo "<td> <form action=\"\$action\" method=\"\$method\" style=\"margin:0\">";
$n = count($nomecampi);
for ($i = 0; $i < $n; $i++) {
    echo " <input type=\"hidden\" name=\"\$nomecampi[$i]\" value=\"\$valcampi[$i]\"
        />";
}
10 echo " <input type=\"submit\" value=\"\$testosulbottone\" /> </form> </td>";
}

function query2tab($query, &$conn, $tabnum='', $datibottoni=NULL, $campidapassare=array()) {
# tabnum: identificativo della tabella nella pagina (ne permette >1)
# datibottoni: array di array('action'=>., 'method'=>., 'etichetta'=>., 'parametri' => array())
# campidapassare: elenco di nomi dei campi da passare allo script del bottone
# parametri: array associativo di campi prefissati da passare allo script del bottone

if (isset($datibottoni)) { #sistema datibottoni
20     for ($i = 0; $i < count($datibottoni); $i++) {
        if (!isset($datibottoni[$i]['action'])) {
            unset($datibottoni[$i]);
        } else {
            if (!isset($datibottoni[$i]['method']))
                $datibottoni[$i]['method'] = 'post';
            if (!isset($datibottoni[$i]['etichetta']))
                $datibottoni[$i]['etichetta'] = 'Modifica';
            if (!isset($datibottoni[$i]['parametri']))
                $datibottoni[$i]['parametri'] = array();
30         }
    }
    if (count($datibottoni) < 1) unset($datibottoni);
}

if (isset($_GET["tableorderby$tabnum"])) {
    $orderby = sololettere($_GET["tableorderby$tabnum"]);
    $query .= ' ORDER BY ' . $orderby;
    if (isset($_GET["tableorderDESC$tabnum"])) {
40         $DESC = TRUE;
        $query .= ' DESC';
    }
}

$res = mysql_query($query . ' ', $conn) or die("Query fallita: ". mysql_error($conn));

if (!mysql_num_rows($res)) { #caso senza risultati
    print "Nessun risultato\n";
    return;
}

50 $baseurl = $_SERVER['SCRIPT_NAME'] . '?'; #sistema altri parametri
foreach ($_GET as $k => $i) {
    if ($k != "tableorderby$tabnum" && $k != "tableorderDESC$tabnum") $baseurl .=
        "$k=$i&";
}

$numcampi = mysql_num_fields($res);
$indicampidata = array();
$nomiampidapassare = array();
$indicampidapassare = array();
60 echo "<table border=\"1\" cellspacing=\"0\" style=\"margin:10px 0px\">". "\n"; #intestazione
echo " <tr>\n";
for ($i = 0; $i < $numcampi; $i++) {
    $nome = mysql_field_name($res, $i);
    if (mysql_field_type($res, $i) === 'date') { #controllo se data
        $indicampidata[] = $i;
    }
}

```

```

    }
    if (isset($datibottoni) && (in_array($nome, $campidapassare))) { #controlla se è un campo da
        passare allo script del bottone
        $indicicampidapassare[] = $i;
        $nomicampidapassare[] = $nome;
    }
70    if (!isset($orderby) or ($nome != $orderby) or isset($DESC)) { #stampa intestazione
        echo " <th><a href=\"\$baseurl\".\"tableorderby\$tabnum=\$nome\">\$nome</a></th>";
    } else {
        echo " <th><a href=\"\$baseurl\".\"tableorderby\$tabnum=\$nome&amp;\".
            \"tableorderDESC\$tabnum\">\$nome</a></th>";
    }
}
if (isset($datibottoni)) echo "\n <th colspan=\"\" . count($datibottoni) . "\">Azioni</th>";
echo "\n </tr>\n";

80    while ($rig = mysql_fetch_row($res)) { #righe
        echo "<tr>\n";
        $valoricampidapassare = array();
        for ($i=0; $i < $numcampi; $i++) { #campi
            if (isset($datibottoni) && in_array($i, $indicicampidapassare)) {
                $valoricampidapassare[] = $rig[$i];
            }
            if ($rig[$i]=='') {
                $rig[$i] = '<center>&#8226;</center>';
            } else if (in_array($i, $indicicampidata)) {
                $rig[$i] = date('d/m/Y', strtotime($rig[$i]));
90            }
            echo " <td>\$rig[\$i]</td>";
        }
        if (isset($datibottoni)) {
            for ($i = 0; $i < count($datibottoni); $i++) {
                creapulsante($datibottoni[$i]['action'], $datibottoni[$i]['method'],
                    array_merge(array_keys($datibottoni[$i]['parametri']), $nomicampidapassare),
                    array_merge(array_values($datibottoni[$i]['parametri']),
                        $valoricampidapassare), $datibottoni[$i]['etichetta']);
            }
        }
        echo "\n </tr>\n";
100    }
    echo "</table>\n";
}
?>

```

6.2 Pagine di servizio

6.2.1 _delete.php, _insert.php e _update.php

I tre *script* `_delete.php`, `_insert.php` e `_update.php` servono ad eseguire le funzioni da cui traggono il nome. Prendono in *input* (con metodo `POST`) una lista di nomi e valori di campi che descrivono i dati su cui va effettuata l'operazione. Qui ed altrove, i nomi dei parametri sono filtrati dalla funzione `sololettere`, mentre il contenuto viene filtrato dalla `nonvirgolette`, per evitare fenomeni di *code injection*. I nomi dei campi chiave, ove presenti, vengono inoltre validati dalla funzione `ckchiavi` per evitare che possano essere eseguite operazioni di massa. Infine, viene invocata la funzione `cklivello` per verificare che l'utente attuale abbia il permesso di eseguire l'operazione che ha richiesto. Alcuni parametri sono speciali: tutti e tre gli *script* ricevono ad esempio il parametro `---nometabella` che determina la tabella su cui va eseguita l'operazione.

Lo *script* `_delete.php` ha solo un altro parametro speciale, `---` conferma, che, se il suo valore è diverso da `'conferma'`, impedisce il compimento dell'operazione. Lo *script* assume che tutti i parametri normali che riceve siano chiavi.

```

0  <?php
   require_once('../php-incl/potab.php');
   require_once('../php-incl/misc.php');

   $liv = auth();
   if ($liv <= 1) err('Livello di autorizzazione insufficiente');

   $chiavi = array();
   foreach ($_POST as $k => $i) {
10      switch(substr($k, 0, 3)) {
         case '---':
            switch($k) {
               case '---nometabella':
                  $nometabella = sololettere($i);
                  break;
               case '---conferma':
                  if ($i = 'conferma') $conf = true;
                  break;
            }
            break;
20      default:
         $chiavi[sololettere($k)] = nonvirgolette($i);
         break;
      }
   }

   if (!isset($conf)) err('Non hai confermato la cancellazione');
   if (!isset($nometabella)) err('Errore di selezione tabella');
   ckchiavi($nometabella, $chiavi);
   cklivello($nometabella, $liv, 'delete');
30

   dbconn($conn);
   $query = "DELETE FROM $nometabella WHERE TRUE";
   foreach ($chiavi as $k => $i) {
      $query .= " AND $k='$i' ";
   }
   $query .= ' ';
   mysql_query($query, $conn);
   err(get_warn($conn), 'Cancellazione eseguita con successo');
   ?>

```

Lo *script* `_insert.php` può ricevere come parametri in *input* anche date, formattate nella maniera usuale italiana (GG/MM/AAAA); i nomi dei campi di tipo data devono essere preceduti da `-D-`. Se la data inserita non è valida viene prodotto un errore.

```

0  <?php
   require_once('../php-incl/potab.php');
   require_once('../php-incl/misc.php');

   $liv = auth();
   if ($liv <= 1) err('Livello di autorizzazione insufficiente');

   $chiavi = array();
   $campi = array();
10  foreach ($_POST as $k => $i) {
      switch(substr($k, 0, 3)) {
         case '---':

```

```

        if($k == '---nometabella') $nometabella = sololettere($i);
        if($k == '---password') $campi['Password'] = SHA1($i);
        break;
    case '-D-':
        if ($i != '') {
            $i = preg_replace('/^([0-3]?[0-9])/([01]?[0-9])/([0-9]{2,4})$/','\3-\2-\1',$i,
                1, $s);
            $i = strtotime($i);
            if (( $i == NULL) || ($s != 1)) err('Errore nella data');
            $campi[sololettere(substr($k, 3))] = date('Y-m-d', $i);
        }
        break;
    default:
        $campi[sololettere($k)] = nonvirgolette($i);
        break;
    }
}

if (!isset($nometabella)) err('Errore di selezione tabella');
30 cklivello ($nometabella, $liv, 'insert');

foreach ($campi as $k=>$i) { #sostituisce stringhe vuote con NULL
    if ($i === '') {
        $campi[$k] = 'NULL';
    } else {
        $campi[$k] = "'$i'";
    }
}

40 dbconn($conn);
$query = '';
foreach (array_keys($campi) as $k) {
    $query .= ", $k";
}
$query = "INSERT INTO $nometabella(" . substr($query, 2);
$query .= ") VALUE (";
foreach ($campi as $i) {
    $query .= "$i, ";
}
50 $query = substr($query, 0, -2) . ')';
mysql_query($query, $conn);
err(get_warn($conn), 'Inserimento eseguito con successo');
?>

```

Infine, lo *script* `_update.php` deve distinguere, oltre alle date (indicate con lo stesso prefisso `-D-`), quali campi in ingresso sono chiave; a questo fine, i nomi dei campi chiave devono essere preceduti da `-K-`. I valori dei campi chiave servono ad identificare il *record*, mentre gli altri sono i nuovi valori degli altri campi.

```

0 <?php
require_once('../php-incl/potab.php');
require_once('../php-incl/misc.php');

$liv = auth();
if ($liv <= 1) err('Livello di autorizzazione insufficiente');

$chiavi = array();
$campi = array();
10 foreach ($_POST as $k => $i) {
    switch(substr($k, 0, 3)) {
        case '---':
            if($k == '---nometabella') $nometabella = sololettere($i);

```



```

        break;
    case '-K-':
        $chiavi[sololettere(substr($k, 3))] = nonvirgolette($i);
        break;
    case '-D-':
        if ($i !== '') {
            $i = preg_replace('/^([0-3]?[0-9])\/([01]?[0-9])\/([0-9]{2,4})$|', '\3-\2-\1', $i,
20             1, $s);
            $i = strtotime($i);
            if (($i == NULL) || ($s != 1)) err('Errore nella data');
            $campi[sololettere(substr($k, 3))] = date('Y-m-d', $i);
        }
        break;
    default:
        $campi[sololettere($k)] = nonvirgolette($i);
        break;
    }
}
30
if (!isset($nometabella)) err('Errore di selezione tabella');
ckchiavi($nometabella, $chiavi);
cklivello($nometabella, $liv, 'update');

dbconn($conn);
$query = "";
foreach ($campi as $k => $i) {
    if ($i === '') {
        $query .= " , $k=NULL";
    } else {
40         $query .= " , $k='$i'";
    }
}
$query = "UPDATE $nometabella SET " . substr($query, 2);
$query .= " WHERE TRUE";
foreach ($chiavi as $k => $i) {
    $query .= " AND $k='$i'";
}
$query .= ' ';
50 mysql_query($query, $conn);
err(get_warn($conn), 'Aggiornamento eseguito con successo');
?>

```

6.2.2 _cambiapassword.php

Il file `_cambiapassword.php` contiene le istruzioni per cambiare la *password* dell'utente corrente.

```

0 <?php
require_once('../php-incl/potab.php');
require_once('../php-incl/misc.php');

$liv = auth();

$old = SHA1($_POST['oldpass']);
$pass = SHA1($_POST['password']);
$conf = SHA1($_POST['conferma']);
$login = $_SESSION['login'];
10

dbconn($conn);
$query = "SELECT * FROM Collaboratori c NATURAL JOIN Enti e WHERE c.CodEnte='$login'
        AND c.Password='$old'";

```

```

$ris = mysql_query($query, $conn);
if (mysql_num_rows($ris) != 1) err("Password vecchia sbagliata");
if ($pass != $conf) err("La nuova password non è la stessa");

$query = "UPDATE Collaboratori c SET c.Password='$pass' WHERE c.CodEnte='$login' AND
        c.Password='$old'";
mysql_query($query, $conn);
err(get_warn($conn), 'Password cambiata correttamente');
20 ?>

```

6.2.3 _logcheck.php

Lo script `_logcheck.php` riceve le credenziali di *login* dall'utente e le valida; in caso di errore produce un messaggio di errore, mentre in caso di *login* effettuato correttamente imposta le variabili di sessione ai loro valori corretti.

```

0 <?php
require_once(' ../php-incl/misc.php');
require_once(' ../php-incl/potab.php');

if (!isset($_POST['login']) || !isset($_POST['password'])) err('Problemi di comunicazione');

$login = solonumeri($_POST['login']);
$pass = SHA1($_POST['password']);

dbconn($conn);
10 $query = "SELECT * FROM Collaboratori c NATURAL JOIN Enti e WHERE c.CodEnte='$login'
        AND c.Password='$pass'";
$ris = mysql_query($query, $conn);
if (mysql_num_rows($ris) != 1) err("Codice utente o password sbagliata");

$rig = mysql_fetch_assoc($ris);
switch($rig['Qualifica']) {
case 'Segretario':
    $liv = 1;
    break;
20 case 'Praticante':
case 'PraticanteConPatrocinio':
    $liv = 2;
    break;
case 'Avvocato':
    $liv = 3;
    break;
}
$nome = $rig['Nome'];

30 session_start();
$_SESSION['login'] = $login;
$_SESSION['livello'] = $liv;
$_SESSION['nome'] = $nome;

header('Location: ' . urlbasesito() . '/index.php');
?>

```

6.2.4 _nuovoente.php

Il file `_nuovoente.php` è una versione modificata di `_insert.php` che, invece di eseguire una generica operazione INSERT, chiama una delle procedure per creare enti, ovvero `InserisciIstituzione`, `InserisciSocieta` o `InserisciPersonaFisica`.

```

0  <?php
    require_once(' ../php-inc1/potab.php');
    require_once(' ../php-inc1/misc.php');

    $liv = auth();
    if ($liv <= 1) err('Livello di autorizzazione insufficiente');

    $chiavi = array();
    $campi = array();
    foreach ($_POST as $k => $i) {
10      switch(substr($k, 0, 3)) {
        case '---':
            if ($k == '---tipo') $tipo = $i;
            break;
        case '-D-':
            if ($i != '') {
                $i = preg_replace('/^([0-3]?[0-9])/([01]?[0-9])/([0-9]{2,4})$/',' \3-\2-\1', $i,
20                1, $s);
                $i = strtotime($i);
                if (($i == NULL) || ($s != 1)) err('Errore nella data');
                $campi[sololettere(substr($k, 3))] = date('Y-m-d', $i);
            }
            break;
        default:
            $campi[sololettere($k)] = nonvirgolette($i);
            break;
    }
}

    if (!isset($tipo) || !in_array($tipo, array('PersonaFisica', 'Societa', 'Istituzione')))
        err('Errore di selezione tabella');
    cklivello('Enti', $liv, 'insert');
30
    foreach ($campi as $k=>$i) { #sostituisce stringhe vuote con NULL
        if ($i === '') {
            $campi[$k] = 'NULL';
        } else {
            $campi[$k] = "'$i'";
        }
    }

    dbconn($conn);
40    switch($tipo) {
        case 'Istituzione':
            foreach (array('Nome', 'TipoIstituzione', 'PartitaIVA', 'NomeResponsabile') as $k)
                if (!isset($campi[$k])) $campi[$k] = "";
            $query = 'CALL InserisciIstituzione(' . $campi['Nome'] . ', ' . $campi['TipoIstituzione'] .
                ', ' . $campi['PartitaIVA'] . ', ' . $campi['NomeResponsabile'] . ');';
            break;
        case 'Societa':
            foreach (array('Nome', 'PartitaIVA', 'LegaleRappresentante') as $k)
                if (!isset($campi[$k])) $campi[$k] = "";
            $query = 'CALL InserisciSocieta(' . $campi['Nome'] . ', ' . $campi['PartitaIVA'] . ', ' .
                $campi['LegaleRappresentante'] . ');';
50            break;
        case 'PersonaFisica':
            foreach (array('Nome', 'CodiceFiscale', 'DataNascita', 'LuogoNascita') as $k)
                if (!isset($campi[$k])) $campi[$k] = "";
            $query = 'CALL InserisciPersonaFisica(' . $campi['Nome'] . ', ' . $campi['CodiceFiscale'] .
                ', ' . $campi['DataNascita'] . ', ' . $campi['LuogoNascita'] . ');';
            break;
    }
}

```

```

mysql_query($query, $conn);
err(get_warn($conn), 'Inserimento eseguito con successo');
?>
60

```

6.3 Pagine di interfaccia

6.3.1 login.php

Il file `login.php` contiene il *form* di *login*.

```

0  <?php
    require_once('../php-incl/misc.php');

    head_trans('Login');

    stampa_err();

    echo<<<END
    <form action="_logcheck.php" method="post" style="margin:10px 0px">
      <fieldset>
        <legend>Login</legend>
10  <table>
      <tr>
        <td> <label for="login">Codice utente:</label> </td>
        <td> <input type="text" name="login" id="login" maxlength="6" /> </td>
      </tr>
      <tr>
        <td> <label for="password">Password:</label> </td>
        <td> <input type="password" name="password" id="password" /> </td>
20  </tr>
    </table>
    <input type="submit" value="Entra" />
  </fieldset>
</form>
END;

    tail_trans();
?>

```

6.3.2 logout.php

Lo *script* `logout.php` si limita a distruggere le variabili di sessione.

```

0  <?php
    session_start();
    $_SESSION = array();
    session_destroy();

    require_once('../php-incl/misc.php');

    head_trans('Logout');
    echo "<h3>Logout effettuato</h3>\n";
    echo "Torna al <a href=\"".urlbasesito()."/login.php\">login</a>.";
10  tail_trans();
?>

```

6.3.3 index.php

Il file `index.php` è un mero *redirect*.

```
<?php
require_once('../php-incl/misc.php');
header('Location: ' . urlbaseito() . '/listacause.php');
?>
```

6.3.4 nuovacausa.php, nuovotribunale.php e nuovoente.php

Lo script `nuovacausa.php` contiene un *form* per creare una nuova causa.

```
0 <?php

require_once('../php-incl/query2tab.php');
require_once('../php-incl/arrayedit.php');
require_once('../php-incl/misc.php');

$liv = auth();

dbconn($conn);

10 head_trans('Nuova Causa');

stampa_err(); #Messaggi

#recupero lista tribunali
$ris = mysql_query("SELECT t.CodTribunale, CONCAT(t.TipoTribunale, ' di ', t.Luogo, '
        (', t.CodTribunale, ')') AS Nome FROM Tribunali t ORDER BY t.Luogo", $conn);
$risarray = ris2array($ris);
$tribl = array('' => 'scegli...');
foreach ($risarray as $i)
    $tribl[$i['CodTribunale']] = $i['Nome'];

20 #recupero lista giudici
$ris = mysql_query("SELECT e.CodEnte, CONCAT(e.Nome, ' (', e.CodEnte, ')') AS Nome FROM
        VEnti e WHERE e.Tipo LIKE '%g%' ORDER BY e.Nome", $conn);
$risarray = ris2array($ris);
$giudl = array('' => 'scegli...');
foreach ($risarray as $i)
    $giudl[$i['CodEnte']] = $i['Nome'];

afieldset('Nuova causa', 'ncausa');
$rig = array(
30 'Stato' => array('valore' => ''),
    'RG' => array('valore' => '', 'maxlung' => 11, 'obbligatorio' => true),
    'CodTribunale' => array('valore' => '', 'etichetta' => 'Tribunale', 'menu' => $tribl,
        'obbligatorio' => true),
    'CodGiudice' => array('valore' => '', 'etichetta' => 'Giudice', 'menu' => $giudl,
        'obbligatorio' => true);

arrayedit($rig, array('---nometabella' => 'Cause'), ($liv>1), 'Inserisci', '_insert.php');
cfieldset();

tail_trans();
?>
```

Il file `nuovotribunale.php` produce un *form* di creazione di un nuovo tribunale.

```
0 <?php
```

```

require_once(' ../php-incl/query2tab.php');
require_once(' ../php-incl/arrayedit.php');
require_once(' ../php-incl/misc.php');

$liv = auth();

dbconn($conn);

10 head_trans('Nuovo Tribunale');

stampa_err(); #Messaggi

afieldset('Nuovo tribunale', 'ntrib');
$rig = array(
    'Luogo' => array('valore' => '', 'obbligatorio' => true),
    'TipoTribunale' => array('valore' => '', 'etichetta' => 'Tipo Tribunale', 'obbligatorio'
        => true),
    'Indirizzo' => array('valore' => '', 'lunghezza' => 50),
    'Telefono' => array('valore' => '');
20 arrayedit($rig, array('---nometabella' => 'Tribunali'), ($liv>1), 'Inserisci', '_insert.php');
cfieldset();

tail_trans();
?>

```

Lo *script* nuovoente.php genera tre *form* che permettono di creare rispettivamente una istituzione, una società o una persona fisica. I tre form sono nascosti da un attributo di stile e vengono mostrati dalla funzione *javascript* toggle quando l'utente utilizza uno dei *link* appositi.

```

0 <?php

require_once(' ../php-incl/query2tab.php');
require_once(' ../php-incl/arrayedit.php');
require_once(' ../php-incl/misc.php');

$liv = auth();

dbconn($conn);

10 head_trans('Nuovo ente');

stampa_err(); #Messaggi

afieldset('Nuovo ente', 'nente');
echo "<a href=\"#\" onclick=\"toggle('nist')\">Nuova istituzione</a> &mdash; \n";
echo "<a href=\"#\" onclick=\"toggle('nsoc')\">Nuova società</a> &mdash; \n";
echo "<a href=\"#\" onclick=\"toggle('npers')\">Nuova persona fisica</a>\n";
cfieldset();

20 afieldset('Nuova Istituzione', 'nist', 'display:none');
$rig = array(
    'Nome' => array('valore' => '', 'obbligatorio' => true),
    'TipoIstituzione' => array('valore' => '', 'etichetta' => 'Tipo Istituzione',
        'obbligatorio' => true),
    'PartitaIVA' => array('valore' => '', 'etichetta' => 'Partita IVA'),
    'NomeResponsabile' => array('valore' => '', 'etichetta' => 'Nome Responsabile');
arrayedit($rig, array('---tipo' => 'Istituzione'), ($liv>1), 'Inserisci', '_nuovoente.php');
cfieldset();

```

```

30  afieldset('Nuova Società', 'nsoc', 'display:none');
    $rig = array(
        'Nome' => array('valore' => '', 'obbligatorio' => true),
        'PartitaIVA' => array('valore' => '', 'etichetta' => 'Partita IVA'),
        'LegaleRappresentante' => array('valore' => '', 'etichetta' => 'Legale Rappresentante');
    arrayedit($rig, array('---tipo' => 'Società'), ($liv>1), 'Inserisci', '_nuovoente.php');
    cfieldset();

    afieldset('Nuova Persona Fisica', 'npers', 'display:none');
    $rig = array(
        'Nome' => array('valore' => '', 'obbligatorio' => true),
40    'CodiceFiscale' => array('valore' => '', 'etichetta' => 'Codice Fiscale'),
        'DataNascita' => array('valore' => '', 'etichetta' => 'Data di nascita', 'data' => true),
        'LuogoNascita' => array('valore' => '', 'etichetta' => 'Luogo di nascita');
    arrayedit($rig, array('---tipo' => 'PersonaFisica'), ($liv>1), 'Inserisci', '_nuovoente.php');
    cfieldset();

    tail_trans();
    ?>

```

6.3.5 listacause.php, listaenti.php e listatribunali.php

I files `listacause.php`, `listaenti.php` e `listatribunali.php` consentono di eseguire ricerche di varie entità (realizzate passando parametri GET agli *script* medesimi) e di visualizzare i risultati tramite la funzione `query2tab`.

Seguono gli *script* citati, nell'ordine.

```

0  <?php
    require_once(' ../php-incl/potab.php');
    require_once(' ../php-incl/misc.php');
    require_once(' ../php-incl/query2tab.php');
    require_once(' ../php-incl/arrayedit.php');

    auth();

    head_trans('Lista Cause');
10  dbconn($conn);

    $cod = (isset($_GET['CodCausa']) ? solonumeri($_GET['CodCausa']): '');
    $stato = (isset($_GET['Stato']) ? nonvirgolette($_GET['Stato']): '');
    $rg = (isset($_GET['RG']) ? nonvirgolette($_GET['RG']): '');
    $tribc = (isset($_GET['CodTribunale']) ? solonumeri($_GET['CodTribunale']): '');
    $giudc = (isset($_GET['CodGiudice']) ? solonumeri($_GET['CodGiudice']): '');
    $datip = (isset($_GET['Dati_Partii']) ? nonvirgolette($_GET['Dati_Partii']): '');

    echo "<div id=\"causas\" style=\"display:none\"> <a href=\"#\"
        onclick=\"toggle('causaq'); toggle('causas')\">Parametri di ricerca</a> </div>\n";
20  afieldset('Ricerca cause', 'causaq');
    echo "<div style=\"text-align:right; float:right\"> <a href=\"#\"
        onclick=\"toggle('causaq'); toggle('causas')\"><b>&times;</b></a> </div>\n";

    #recupero lista tribunali
    $ris = mysql_query("SELECT t.CodTribunale, CONCAT(t.TipoTribunale, ' di ', t.Luogo, '
        (' , t.CodTribunale, ')') AS Nome FROM Tribunali t ORDER BY t.Luogo", $conn);
    $tribarray = ris2array($ris);
    $tribl = array('' => '*');
    foreach ($tribarray as $i)
        $tribl[$i['CodTribunale']] = $i['Nome'];

30  #recupero lista giudici

```

```

$ris = mysql_query("SELECT e.CodEnte, CONCAT(e.Nome, ' (', e.CodEnte, ')') AS Nome FROM
    VEnti e WHERE e.Tipo LIKE '%g%' ORDER BY e.Nome", $conn);
$giudarray = ris2array($ris);
$giudl = array('' => '*');
foreach ($giudarray as $i)
    $giudl[$i['CodEnte']] = $i['Nome'];

$rig = array(
    'CodCausa' => array('valore' => $cod, 'maxlung' => 6),
    'Stato' => array('valore' => $stato),
40    'RG' => array('valore' => $rg),
    'CodTribunale' => array('valore' => $tribc, 'menu' => $tribl, 'etichetta' => 'Tribunale'),
    'CodGiudice' => array('valore' => $giudc, 'menu' => $giudl, 'etichetta' => 'Giudice'),
    'Dati_Part' => array('valore' => $datip, 'lunghezza' => 50, 'etichetta' => 'Dati Parti');
arrayedit($rig, array(), true, 'Cerca', 'listacause.php', 'get');
cfieldset(); #fine ricerca

$query = "SELECT c.CodCausa, c.Stato, c.RG, CONCAT(t.TipoTribunale, ' di ', t.Luogo) AS
    Tribunale, e.Nome AS Giudice, DatiParti FROM VCause c NATURAL JOIN Tribunali t
    JOIN Enti e ON c.CodGiudice=e.CodEnte WHERE TRUE";
if ($cod) $query.= " AND CodCausa=$cod";
if ($stato) $query.= " AND Stato REGEXP '$stato'";
50 if ($rg) $query.= " AND RG REGEXP '$rg'";
if ($tribc) $query.= " AND CodTribunale=$tribc";
if ($giudc) $query.= " AND CodGiudice=$giudc";
if ($datip) $query.= " AND DatiParti REGEXP '$datip'";

query2tab($query, $conn, 1, array(array('etichetta' => 'Dettagli', 'action' =>
    'dettaglicausa.php', 'method' => 'get')), array('CodCausa'));
echo "<small>(Le parti in <b>grassetto</b> sono i clienti)</small>\n";

tail_trans();
?>

```

```

0 <?php
require_once('../php-incl/potab.php');
require_once('../php-incl/misc.php');
require_once('../php-incl/query2tab.php');
require_once('../php-incl/arrayedit.php');

auth();

head_trans('Lista Enti');

10 $cod = (isset($_GET['CodEnte']) ? solonumeri($_GET['CodEnte']): '');
$nome = (isset($_GET['Nome']) ? nonvirgolette($_GET['Nome']): '');
$tipo = (isset($_GET['Tipo']) ? nonvirgolette($_GET['Tipo']): 'Ente');
$det = (isset($_GET['Altri_dettagli']) ? nonvirgolette($_GET['Altri_dettagli']): '');

echo "<div id=\"entis\" style=\"display:none\"> <a href=\"#\"
    onclick=\"toggle('entiq'); toggle('entis')\">Parametri di ricerca</a> </div>\n";
afieldset('Ricerca enti', 'entiq');
echo "<div style=\"text-align:right; float:right\"> <a href=\"#\"
    onclick=\"toggle('entiq'); toggle('entis')\"><b>&times;</b></a> </div>\n";
$rig = array(
    'CodEnte' => array('valore' => $cod, 'maxlung' => 4),
20    'Nome' => array('valore' => $nome),
    'Tipo' => array('valore' => $tipo, 'menu' => array('Ente' => 'Ente', 'Istituzione' =>
        'Istituzione', 'Società' => 'Società', 'Persona Fisica' => 'Persona Fisica',
        'Giudice' => 'Giudice', 'Collaboratore' => 'Collaboratore')),
    'Altri_dettagli' => array('valore' => $det, 'lunghezza' => 50));
arrayedit($rig, array(), true, 'Cerca', 'listaenti.php', 'get');

```



```

cfieldset();

$query = 'SELECT CodEnte, Nome, Dati FROM VEnti WHERE TRUE';
if ($cod) $query.= " AND CodEnte='$cod'";
if ($nome) $query.= " AND Nome REGEXP '$nome'";
if ($det) $query.= " AND Dati REGEXP '$det'";
30 if ($tipo != 'Ente') $query.= " AND Tipo LIKE '%" . strtolower($tipo[0]) . "%'";

dbconn($conn);
query2tab($query, $conn, 1, array(array('etichetta' => 'Dettagli', 'action' =>
    'dettagliente.php', 'method' => 'get')), array('CodEnte'));

tail_trans();
?>

```

```

0 <?php
require_once('../php-incl/potab.php');
require_once('../php-incl/misc.php');
require_once('../php-incl/query2tab.php');
require_once('../php-incl/arrayedit.php');

auth();

head_trans('Lista Tribunali');

10 $cod = (isset($_GET['CodTribunale']) ? solonumeri($_GET['CodTribunale']) : '');
$luogo = (isset($_GET['Luogo']) ? nonvirgolette($_GET['Luogo']) : '');
$tipo = (isset($_GET['TipoTribunale']) ? nonvirgolette($_GET['TipoTribunale']) : '');
$det = (isset($_GET['Altri_dettagli']) ? nonvirgolette($_GET['Altri_dettagli']) : '');

echo "<div id=\"tribs\" style=\"display:none\"> <a href=\"#\"
    onclick=\"toggle('tribq'); toggle('tribs')\">Parametri di ricerca</a> </div>\n";
afieldset('Ricerca tribunali', 'tribq');
echo "<div style=\"text-align:right; float:right\"> <a href=\"#\"
    onclick=\"toggle('tribq'); toggle('tribs')\"><b>&times;</b></a> </div>\n";
$rig = array(
20 'CodTribunale' => array('valore' => $cod, 'maxlung' => 4),
    'Luogo' => array('valore' => $luogo),
    'TipoTribunale' => array('valore' => $tipo, 'etichetta' => 'Tipo Tribunale'),
    'Altri_dettagli' => array('valore' => $det, 'lunghezza' => 50, 'etichetta' => 'Altri
        dettagli');
arrayedit($rig, array(), true, 'Cerca', 'listatribunali.php', 'get');
cfieldset();

$query = "SELECT * FROM VTribunali WHERE TRUE";
if ($cod) $query.= " AND CodTribunale='$cod'";
if ($luogo) $query.= " AND Luogo REGEXP '$luogo'";
if ($tipo) $query.= " AND TipoTribunale REGEXP '$tipo'";
30 if ($det) $query.= " AND CONCAT(Indirizzo, Telefono) REGEXP '$det'";

dbconn($conn);
query2tab($query, $conn, 1, array(array('etichetta' => 'Dettagli', 'action' =>
    'dettaglitribunale.php', 'method' => 'get')), array('CodTribunale'));

tail_trans();
?>

```

6.3.6 dettaglicausa.php

Il file `dettaglicausa.php` permette di visualizzare e modificare (se si possiede il livello di autorizzazione adeguato) le proprietà di una causa e delle udienze relative.

La prima sezione consente di modificare le proprietà facenti parte della tabella CAUSE, ed ha un riquadro nascosto tramite un attributo di stile che permette di eliminare la causa (il riquadro può essere mostrato tramite un *link* che invoca la funzione *javascript toggle*).

La seconda sezione mostra le parti coinvolte nella causa. Ha un riquadro nascosto per aggiungere altre parti; inoltre, per ogni riga una coppia di pulsanti permette di passare alla scheda della parte e di modificare l'associazione di quell'ente con la causa in esame. Il riquadro di modifica e rimozione non è già presente e nascosto, bensì è richiesto al *server* quando serve richiedendo la pagina con un parametro GET aggiuntivo.

In maniera molto simile, la terza sezione mostra le udienze collegate, ha un riquadro nascosto per aggiungerne e usa un parametro GET per richiedere il riquadro per modificare o eliminare le udienze e per associare i collaboratori ad ognuna di esse.

```
0  <?php
    require_once(' ../php-incl/query2tab.php');
    require_once(' ../php-incl/arrayedit.php');
    require_once(' ../php-incl/misc.php');

    $liv = auth();

    dbconn($conn);
    if (!isset($_GET['CodCausa'])) die($_SERVER['PHP_SELF'].' Non so di chi parli');
10  $cod = solonumeri($_GET['CodCausa']); #Causa
    $risul = mysql_query("SELECT c.Stato, c.RG, c.CodTribunale, c.CodGiudice FROM Cause c
        WHERE c.CodCausa=$cod;", $conn) or die("Query fallita: ".mysql_error($conn));
    if (!mysql_num_rows($risul)) {
        header('Location: '.urlbasesito().'/listacause.php');
        die();
    }

    head_trans("Dettagli Causa $cod");

    stampa_err(); #Messaggi
20  afieldset('Causa', 'causa');
    if ($liv > 2) { #cancella causa
        afieldset('Elimina Causa', 'rmcausa', 'float:right; display:none; margin-left:20px');
        echo <<<END
        <form action="_delete.php" method="post">
            <input type="hidden" name="---nometabella" value="Cause" />
            <input type="hidden" name="CodCausa" value="$cod" />
            <input type="checkbox" name="---conferma" id="confermac" />
            <label for="confermac">Conferma</label>
30  <input type="submit" value="Elimina causa" />
        </form>\n
        END;
        cfieldset();
    }

    #recupero lista tribunali
    $ris = mysql_query("SELECT t.CodTribunale, CONCAT(t.TipoTribunale, ' di ', t.Luogo, '
        (', t.CodTribunale, ')') AS Nome FROM Tribunali t ORDER BY t.Luogo", $conn);
    $risarray = ris2array($ris);
    $tribl = array(' => 'scegli...');
```

```

40 foreach ($risarray as $i)
    $tribl[$i['CodTribunale']] = $i['Nome'];

#recupero lista giudici
$ris = mysql_query("SELECT e.CodEnte, CONCAT(e.Nome, ' (', e.CodEnte, ')') AS Nome FROM
    VEnti e WHERE e.Tipo LIKE '%g%' ORDER BY e.Nome", $conn);
$risarray = ris2array($ris);
$giudl = array('' => 'scegli...');
foreach ($risarray as $i)
    $giudl[$i['CodEnte']] = $i['Nome'];

50 $ris = $risul;
$rig = mysql_fetch_assoc($ris);
$rig = assoc2proplist($rig);
$rig['RG']['maxlung'] = 11;
$rig['RG']['obbligatorio'] = true;
$rig['CodTribunale']['etichetta'] = 'Tribunale';
$rig['CodTribunale']['menu'] = $tribl;
$rig['CodTribunale']['obbligatorio'] = true;
$rig['CodGiudice']['etichetta'] = 'Giudice';
$rig['CodGiudice']['menu'] = $giudl;
60 $rig['CodGiudice']['obbligatorio'] = true;

arrayedit($rig, array('---nometabella' => 'Cause', '-K-CodCausa' => $cod), ($liv>1));
if ($liv > 2) {
    echo "<a href=\"\#\" onclick=\"toggle('rmcausa')\">Elimina causa</a>\n";
}
cfieldset();

70 afieldset('Parti', 'par');
echo "<div style=\"float:right; margin-left:20px\">\n";
if (isset($_GET['CodEnte']) && $liv>1) { #modifica associazione
    $codente = solonumeri($_GET['CodEnte']);
    $ris = mysql_query("SELECT ci.Ruolo, ci.Cliente FROM CoinvoltoIn ci WHERE
        ci.CodEnte=$codente AND ci.CodCausa=$cod", $conn);
    if (mysql_num_rows($ris)) {
        afieldset("Modifica associazione con ente $codente", 'edpar');
        $param = array(); #parametri per il link x
        foreach ($_GET as $k => $i)
            if ($k != 'CodEnte') $param[$k] = $i;
80 echo "<div style=\"text-align:right; float:right\"> <a
            href=\"dettaglicausa.php?\".http_build_query($param, '',
                '&');\"><b>&times;</b></a> </div>\n";

        $rig = mysql_fetch_assoc($ris);
        $rig = assoc2proplist($rig);
        $rig['Cliente']['sceltamultipla'] = array(0 => 'no', 1 => 'si');
        $rig['Cliente']['obbligatorio'] = true;
        $rig['Ruolo']['obbligatorio'] = true;
        arrayedit($rig, array('---nometabella' => 'CoinvoltoIn', '-K-CodCausa' => $cod,
            '-K-CodEnte' => $codente), ($liv>1));
        if ($liv>2)
90 echo <<<END
    <hr />
    <form action="_delete.php" method="post">
    <input type="hidden" name="---nometabella" value="CoinvoltoIn" />
    <input type="hidden" name="CodCausa" value="$cod" />
    <input type="hidden" name="CodEnte" value="$codente" />
    <input type="checkbox" name="---conferma" id="confermap" />
    <label for="confermap">Conferma</label>
    <input type="submit" value="Elimina associazione" />

```

```

100 </form>\n
    END;
        cfieldset();
    }
}

if ($liv>1) { #nuova associazione parte
    afieldset('Nuova associazione', 'addpar', 'display:none');

    #recupero lista enti
    $ris = mysql_query("SELECT e.CodEnte, CONCAT(e.Nome, ' (', e.CodEnte, ')') AS Nome
        FROM Enti e WHERE e.CodEnte NOT IN (SELECT ci.CodEnte FROM CoinvoltoIn ci
        WHERE ci.CodCausa=$cod) ORDER BY e.Nome", $conn);
    $risarray = ris2array($ris);
    $sentil = array('' => 'scegli...');
110 foreach ($risarray as $i)
        $sentil[$i['CodEnte']] = $i['Nome'];
    $rig = array(
        'CodEnte' => array('valore' => '', 'menu' => $sentil, 'obbligatorio' => true, 'etichetta'
            => 'Ente'),
        'Ruolo' => array('valore' => '', 'obbligatorio' => true),
        'Cliente' => array('valore' => 0, 'sceltamultipla' => array(0 => 'no', 1 => 'si'),
            'obbligatorio' => true);
    arrayedit($rig, array('---nometabella' => 'CoinvoltoIn', 'CodCausa' => $cod), ($liv>1),
        'Inserisci', '_insert.php');
    cfieldset();
}
120 echo " </div>\n";

$bott = array(array('etichetta' => 'Dettagli ente', 'method' => 'get', 'action' =>
    'dettagliente.php'));
if ($liv > 1) {
    $param = array();
    foreach ($_GET as $k => $i)
        if ($k != 'CodEnte') $param[$k] = $i;
    $bott[] = array('etichetta' => 'Modifica associazione', 'method' => 'get', 'action' =>
        'dettagliacausa.php', 'parametri' => $param);
}
query2tab("SELECT e.CodEnte, CONCAT(IF(ci.Cliente, '<b>', ''), e.Nome, IF(ci.Cliente,
    '</b>', '')) AS Nome, ci.Ruolo FROM CoinvoltoIn ci NATURAL JOIN Enti e WHERE
    ci.CodCausa=$cod", $conn, 1, $bott, array('CodEnte'));
130 if ($liv > 1) {
    echo "<a href='#' onclick='toggle('addpar')'>Aggiungi parte</a> &mdash;\n";
}
echo "<small>(Le parti in <b>grassetto</b> sono i clienti)</small>\n";
cfieldset();

afieldset('Udienze', 'udi');
echo " <div style='float:right; margin-left:20px'>\n";
140 if (isset($_GET['CodUdienza']) && $liv>1) { #modifica udienza
    $codudienza = solonumeri($_GET['CodUdienza']);
    $ris = mysql_query("SELECT u.Data, u.TipoUdienza FROM VUdienze u WHERE
        u.CodUdienza=$codudienza", $conn);
    if (mysql_num_rows($ris)) {
        afieldset("Modifica udienza $codudienza", 'edudi');
        $param = array(); #parametri per il link x
        foreach ($_GET as $k => $i)
            if ($k != 'CodUdienza') $param[$k] = $i;
        echo "<div style='text-align:right'> <a
            href='dettagliacausa.php?'.http_build_query($param, '',
                '&')." "><b>&times;</b></a> </div>\n";
    }
}

```

```

150      $rig = mysql_fetch_assoc($ris);
      $rig = assoc2proplist($rig);
      $rig['Data']['data'] = true;
      $rig['Data']['obbligatorio'] = true;
      $rig['TipoUdienza']['etichetta'] = 'Tipo Udienza';
      $rig['TipoUdienza']['obbligatorio'] = true;
      arrayedit($rig, array('---nometabella' => 'Udienze', '-K-CodUdienza' => $codudienza),
        ($liv>1));
      #partecipanti
      echo "<hr />\n<b>Partecipanti</b> <br />\n";
      $query = "SELECT e.* FROM DiscussaDa dd NATURAL JOIN Enti e WHERE
        dd.CodUdienza=$codudienza";
160      $bott = array(array('etichetta' => 'Rimuovi', 'action' => '_delete.php', 'method' =>
        'post', 'parametri' => array('---nometabella' => 'DiscussaDa', '---conferma' =>
        'conferma', 'CodUdienza' => $codudienza)));
      query2tab($query, $conn, 3, $bott, array('CodEnte'));
      echo <<<END
      <form action="_insert.php" method="post">
      <input type="hidden" name="---nometabella" value="DiscussaDa" />
      <input type="hidden" name="CodUdienza" value="$codudienza" />
      <select id="selcoll" name="CodEnte">
      END;
      #recupero lista collaboratori
      $ris = mysql_query("SELECT e.CodEnte, CONCAT(e.Nome, ' (', e.CodEnte, ')') AS
        Nome FROM VEnti e WHERE Tipo LIKE '%c%' AND CodEnte NOT IN (SELECT
        dd.CodEnte FROM DiscussaDa dd WHERE dd.CodUdienza=$codudienza) ORDER BY
        e.Nome", $conn);
170      $risarray = ris2array($ris);
      foreach ($risarray as $i)
        echo ' <option value="' . $i['CodEnte'] . '">' . $i['Nome'] . "</option>\n";
      echo <<<END
      </select>
      <input type="submit" value="Inserisci" />
      </form>\n
      END;
      if ($liv>2) #elimina udienza
      echo <<<END
180      <hr />
      <b>Elimina</b> <br />
      <form action="_delete.php" method="post">
      <input type="hidden" name="---nometabella" value="Udienze" />
      <input type="hidden" name="CodUdienza" value="$codudienza" />
      <input type="checkbox" name="---conferma" id="confermau" />
      <label for="confermau">Conferma</label>
      <input type="submit" value="Elimina udienza" />
      </form>\n
      END;
190      cfieldset ();
      }
    }

    if ($liv>1) { #nuova udienza
      afieldset('Aggiungi udienza', 'addudi', 'display:none');

      $rig = array(
        'Data' => array('valore' => '', 'obbligatorio' => true, 'data' => true),
        'TipoUdienza' => array('valore' => '', 'obbligatorio' => true));
200      arrayedit($rig, array('---nometabella' => 'Udienze', 'CodCausa' => $cod), ($liv>1),
        'Inserisci', '_insert.php');
      echo "<small>(I partecipanti potranno essere aggiunti quando l'udienza è stata
        inserita)</small> <br />\n";
      cfieldset ();
    }
  }
}

```

```

    }
    echo " </div>\n";

    $bott = array();
    if ( $liv > 1 ) {
        $param = array();
        foreach ( $_GET as $k => $i )
            if ( $k != 'CodUdienza' ) $param[$k] = $i;
        $bott[] = array('etichetta' => 'Modifica', 'method' => 'get', 'action' =>
            'dettagli causa.php', 'parametri' => $param);
    }
    query2tab("SELECT u.CodUdienza, u.Data, u.TipoUdienza, u.Partecipanti FROM VUdienze u
        WHERE u.CodCausa=$cod", $conn, 2, $bott, array('CodUdienza'));
    if ( $liv > 1 ) {
        echo "<a href='#\" onclick='\"toggle('addudi')\">Aggiungi udienza</a>\n";
    }
    cfieldset();

    tail_trans();
    ?>

```

6.3.7 dettaglitribunale.php

Lo script `dettaglitribunale.php` permette di visualizzare e modificare (avendone i permessi) le proprietà di un tribunale.

La prima sezione permette di modificare i campi della tabella `TRIBUNALI` e di eliminare un tribunale tramite un riquadro nascosto.

La seconda sezione elenca i giudici associati al tribunale e permette di aggiungere un giudice (tramite un riquadro nascosto contenente un apposito *form*) e di rimuovere una associazione o passare alla scheda del giudice tramite i pulsanti della tabella `query2tab`.

La terza sezione elenca le cause presenti nel tribunale in esame e permette di passare alle schede delle singole cause.

```

0  <?php

    require_once(' ../php-incl/query2tab.php');
    require_once(' ../php-incl/arrayedit.php');
    require_once(' ../php-incl/misc.php');

    $liv = auth();

    dbconn($conn);
    if (!isset($_GET['CodTribunale'])) die($_SERVER['PHP_SELF'].' Non so di chi parli');
10  $cod = solonumeri($_GET['CodTribunale']); #Tribunale
    $ris = mysql_query("SELECT t.Luogo, t.TipoTribunale, t.Indirizzo, t.Telefono FROM
        Tribunali t WHERE t.CodTribunale=$cod;", $conn) or die("Query fallita: ").
        mysql_error($conn);
    if (!mysql_num_rows($ris)) {
        header('Location: ' . urlbasesito() . '/listatribunali.php');
        die();
    }

    head_trans("Dettagli Tribunale $cod");

    stampa_err(); #Messaggi
20  afieldset('Tribunale', 'trib');
    if ( $liv > 2 ) { #cancella tribunale

```

```

        afieldset('Elimina Tribunale', 'rmtrib', 'display:none; float:right; margin:10px');
echo <<<END
<form action="_delete.php" method="post">
    <input type="hidden" name="---nometabella" value="Tribunali" />
    <input type="hidden" name="CodTribunale" value="$cod" />
    <input type="checkbox" name="---conferma" id="confermat" />
    <label for="confermat">Conferma</label>
30    <input type="submit" value="Elimina tribunale" />
</form>\n
END;
    cfieldset();
}

$rig = mysql_fetch_assoc($ris);
$rig = assoc2proplist($rig);
$rig['TipoTribunale']['etichetta'] = 'Tipo Tribunale';
$rig['TipoTribunale']['obbligatorio'] = true;
40 $rig['Luogo']['obbligatorio'] = true;
$rig['Indirizzo']['lunghezza'] = 50;

arrayedit($rig, array('---nometabella' => 'Tribunali', '-K-CodTribunale' => $cod), ($liv>1));
if ($liv > 2) {
    echo "<a href=\"#\" onclick=\"toggle('rmtrib')\">Elimina tribunale</a>\n";
}
cfieldset();

afieldset('Giudici', 'giud');
50 if ($liv > 1) {
    afieldset('Associa giudice', 'addgiu', 'float:right; display:none; margin:10px');
echo<<<END
<form action="_insert.php" method="post">
    <input type="hidden" name="---nometabella" value="LavoraIn" />
    <input type="hidden" name="CodTribunale" value="$cod" />
    <label for="selgiu">Giudice:<sup>*</sup></label>
    <select id="selgiu" name="CodEnte">\n
END;
    $ris = mysql_query("SELECT e.* FROM Enti e NATURAL JOIN Giudici g WHERE e.CodEnte
        NOT IN (SELECT li.CodEnte FROM LavoraIn li WHERE li.CodTribunale=$cod) ORDER
        BY e.Nome;", $conn);
60 $ris = ris2array($ris);
    foreach ($ris as $i) {
        echo ' <option value="'. $i['CodEnte']. '">'. $i['Nome']. ' ('. $i['CodEnte'].
            ')</option>\n';
    }
echo<<<END
</select> <br />
    <input type="submit" value="Aggiungi" />
</form>\n
END;
    cfieldset();
70 }
$bott = array(array('etichetta' => 'Dettagli', 'action' => 'dettagliante.php', 'method' =>
    'get'));
if ($liv > 2) $bott[] = array('etichetta' => 'Rimuovi associazione', 'action' =>
    '_delete.php', 'method' => 'post', 'parametri' => array('---nometabella' => 'LavoraIn',
    '---conferma' => 'conferma', 'CodTribunale' => $cod));
query2tab("SELECT g. CodEnte, e.Nome, g.TipoGiudice FROM LavoraIn li NATURAL JOIN
    Giudici g NATURAL JOIN Enti e WHERE li.CodTribunale='$cod'", $conn, 1, $bott,
    array('CodEnte'));
if ($liv > 1) {
    echo "<a href=\"#\" onclick=\"toggle('addgiu')\">Associa giudice</a>\n";
}
cfieldset();

```

```

80 afieldset('Cause', 'cause');#Cause
query2tab("SELECT CodCausa, Stato, RG, DatiParti FROM VCause c WHERE
      c.CodTribunale=$cod", $conn, 2, array(array('etichetta' => 'Dettagli', 'method' => 'get',
      'action' => 'dettagliacausa.php')), array('CodCausa'), array());
cfieldset();

tail_trans();
?>

```

6.3.8 dettagliente.php

Il file `dettagliente.php` permette di visualizzare e modificare (se si hanno i permessi) le informazioni relative ai vari enti.

La prima sezione permette di modificare il nome dell'ente e di cancellarlo (tramite il classico riquadro nascosto).

La seconda sezione mostra e permette la modifica di dati specifici per istituzioni, società e persone fisiche.

Per i giudici, la sezione successiva permette di accedere al tipo di giudice e di rimuovere lo status di giudice (quindi rendendo l'ente una normale persona fisica); inoltre visualizza la lista di tribunali con cui il giudice è associato permettendo di rimuoverli e aggiungerne; infine elenca le cause giudicate dal giudice in questione.

Per i collaboratori, la sezione successiva permette di cambiarne qualifica e retribuzione. Per i collaboratori diversi da quello *loggato* il sistema permette, se si hanno i permessi, di rimuovere lo status di collaboratore. Ogni collaboratore può, infine, cambiare la propria *password*, mentre non può rimuovere il proprio status di collaboratore.

Per le persone fisiche che non sono collaboratori o non sono giudici il riquadro successivo permette di assegnare loro lo status di collaboratori o giudici, rispettivamente. Anche qui i *form* sono contenuti in riquadri nascosti.

Infine, per tutti gli enti sono elencate le cause in cui sono coinvolti ed i recapiti, che possono essere rimossi mentre altri recapiti possono venire aggiunti.

```

0  <?php

require_once('../php-incl/query2tab.php');
require_once('../php-incl/arrayedit.php');
require_once('../php-incl/misc.php');

$liv = auth();

dbconn($conn);

10 if (!isset($_GET['CodEnte'])) die($_SERVER['PHP_SELF'].' Non so di chi parli');
   $cod = solonumeri($_GET['CodEnte']);
   $ris = mysql_query("SELECT e.CodEnte, e.Nome, TipoEnte(e.CodEnte) AS t FROM Enti e
      WHERE e.CodEnte=$cod;", $conn) or die("Query fallita: ".mysql_error($conn));
   if (!mysql_num_rows($ris)) {
       header('Location: '.urlbasesito().'/listaenti.php');
       die();
   }

   head_trans("Dettagli Ente $cod");

20 stampa_err();#Messaggi

   afieldset('Ente', 'ente');#Ente

```



```

if ($liv > 2) { #cancella ente
afieldset('Elimina ente', 'rmente', 'display:none; float:right; margin-left:20px');
echo <<<END
<form action="_delete.php" method="post">
  <input type="hidden" name="---nometabella" value="Enti" />
  <input type="hidden" name="CodEnte" value="$cod" />
30  <input type="checkbox" name="---conferma" id="confermae" />
    <label for="confermae">Conferma</label>
  <input type="submit" value="Elimina ente" />
</form>\n
END;
cfieldset();
}

$rig = mysql_fetch_assoc($ris);
$t = str_split($rig['t'], 1);
40 $rig = array('Nome' => array('valore' => $rig['Nome']));
$rig['Nome']['obbligatorio'] = true;

arrayedit($rig, array('---nometabella' => 'Enti', '-K-CodEnte' => $cod), ($liv>1));
if ($liv > 2) {
  echo "<a href=\"\#\" onclick=\"toggle('rmente')\">Elimina ente</a>\n";
}

cfieldset();

50 foreach($tas as $i) switch ($i) {
case 's': #Società
  $ris = mysql_query("SELECT s.PartitaIVA, s.LegaleRappresentante FROM Societa s WHERE
    s.CodEnte=$cod;", $conn) or die("Query fallita: ". mysql_error($conn));
  $rig = mysql_fetch_assoc($ris);
  $rig = assoc2proplist($rig);
  $rig['PartitaIVA']['etichetta'] = 'Partita IVA';
  $rig['LegaleRappresentante']['etichetta'] = 'Legale Rappresentante';
  afieldset('Società', 'soc');
  arrayedit($rig, array('---nometabella' => 'Società', '-K-CodEnte' => $cod), ($liv>1));
  cfieldset();
60  break;

case 'i': #Istituzione
  $ris = mysql_query("SELECT i.TipoIstituzione, i.PartitaIVA, i.NomeResponsabile FROM
    Istituzioni i WHERE i.CodEnte=$cod;", $conn) or die("Query fallita: ".
    mysql_error($conn));
  $rig = mysql_fetch_assoc($ris);
  $rig = assoc2proplist($rig);
  $rig['TipoIstituzione']['etichetta'] = 'Tipo Istituzione';
  $rig['TipoIstituzione']['obbligatorio'] = true;
  $rig['PartitaIVA']['etichetta'] = 'Partita IVA';
  $rig['NomeResponsabile']['etichetta'] = 'Nome Responsabile';
70  afieldset('Istituzione', 'ist');
  arrayedit($rig, array('---nometabella' => 'Istituzioni', '-K-CodEnte' => $cod), ($liv>1));
  cfieldset();
  break;

case 'p': #Persona Fisica
  $pers = true;
  $ris = mysql_query("SELECT p.CodiceFiscale, p.DataNascita, p.LuogoNascita FROM
    PersoneFisiche p WHERE p.CodEnte=$cod;", $conn) or die("Query fallita: ".
    mysql_error($conn));
  $rig = mysql_fetch_assoc($ris);
  $rig = assoc2proplist($rig);
80  $rig['CodiceFiscale']['etichetta'] = 'Codice Fiscale';

```

```

$rig['DataNascita']['etichetta'] = 'Data di nascita';
$rig['DataNascita']['data'] = true;
$rig['LuogoNascita']['etichetta'] = 'Luogo di nascita';
afieldset('Persona Fisica', 'pers');
arrayedit($rig, array('---nometabella' => 'PersoneFisiche', '-K-CodEnte' => $cod), ($liv>1));
cfieldset();
break;

90 case 'c': #Collaboratore
    $coll = true;
    afieldset('Collaboratore', 'coll');

    if (($liv > 2) && ($cod != $_SESSION['login'])) { #rimuovi status di collaboratore
        afieldset('Rimuovi status di collaboratore', 'rmcoll', 'clear:right; display:none;
            float:right; margin-left:20px');
        echo <<<END
        <form action="_delete.php" method="post">
        <input type="hidden" name="---nometabella" value="Collaboratori" />
        <input type="hidden" name="CodEnte" value="$cod" />
        <input type="checkbox" name="---conferma" id="confermac" />
100 <label for="confermac">Conferma</label>
        <input type="submit" value="Rimuovi" />
        </form>\n
        END;
        cfieldset();
        }

        if ($cod == $_SESSION['login']) { #cambia password
            afieldset('Cambia password', 'chpwd', 'clear:right; display:none; float:right;
                margin-left:20px');
            echo <<<END
110 <form action="_cambiapassword.php" method="post">
            <table>
            <tr>
            <td> <label for="oldpass">Vecchia password:<sup>*</sup></label> </td>
            <td> <input type="password" name="oldpass" id="oldpass" /> </td>
            </tr>
            <tr>
            <td> <label for="password">Nuova password:<sup>*</sup></label> </td>
            <td> <input type="password" name="password" id="password" /> </td>
            </tr>
120 <tr>
            <td> <label for="confermap">Conferma:<sup>*</sup></label> </td>
            <td> <input type="password" name="conferma" id="confermap" /> </td>
            </tr>
            </table>
            <input type="submit" value="Procedi" />
            </form>\n
            END;
            cfieldset();
            }

130 #form principale
            $ris = mysql_query("SELECT c.Qualifica, c.Retribuzione FROM Collaboratori c WHERE
                c.CodEnte=$cod;", $conn) or die("Query fallita: ".mysql_error($conn));
            $rig = mysql_fetch_assoc($ris);
            $rig = assoc2proplist($rig);
            $rig['Qualifica']['sceltamultipla'] = array('Avvocato' => 'Avvocato',
                'PraticanteConPatrocinio' => 'Praticante con patrocinio', 'Praticante' =>
                'Praticante', 'Segretario' => 'Segretario');
            arrayedit($rig, array('---nometabella' => 'Collaboratori', '-K-CodEnte' => $cod), ($liv > 2));

            #link di visualizzazione div

```

```

140     if ($cod == $_SESSION['login']) {
        echo "<a href=\"#\" onclick=\"toggle('chpwd')\">Cambia password</a>\n";
    }
    if (( $liv > 2) && ($cod != $_SESSION['login'])) {
        echo "<a href=\"#\" onclick=\"toggle('rmcoll')\">Rimuovi status di
        collaboratore</a>\n";
    }
    cfieldset();
    break; #fine collaboratore

case 'g': #Giudice
    $giud = true;
150     afieldset('Giudice', 'giud');

    if (( $liv > 2) && ($cod != $_SESSION['login'])) { #rimuovi status di giudice
afieldset('Rimuovi status di giudice', 'rmgiud', 'display:none; float:right;
    margin-left:20px');
echo <<<END
<form action="_delete.php" method="post">
<input type="hidden" name="---nometabella" value="Giudici" />
<input type="hidden" name="CodEnte" value="$cod" />
<input type="checkbox" name="---conferma" id="confermag" />
<label for="confermag">Conferma</label>
160 <input type="submit" value="Rimuovi" />
</form>\n
END;
cfieldset();
    }

    #form principale
    $ris = mysql_query("SELECT g.TipoGiudice FROM Giudici g WHERE g.CodEnte=$cod;", $conn)
        or die("Query fallita: ".mysql_error($conn));
    $rig = mysql_fetch_assoc($ris);
    $rig = assoc2proplist($rig);
170 $rig['TipoGiudice']['etichetta'] = 'Tipo Giudice';
    $rig['TipoGiudice']['obbligatorio'] = true;
    $rig['TipoGiudice']['sceltamultipla'] = array('Togato' => 'Togato', 'Onorario' =>
        'Onorario');
    arrayedit($rig, array('---nometabella' => 'Giudici', '-K-CodEnte' => $cod), ($liv > 1));
    if ($liv > 2) {
        echo "<a href=\"#\" onclick=\"toggle('rmgiud')\">Rimuovi status di
        giudice</a>\n";
    }

    afieldset('Tribunali associati', 'trib'); #tribunali associati
    if ($liv > 1) {
180     afieldset('Associa tribunale', 'addtrib', 'float:right; display:none;
        margin-left:20px');
echo <<<END
<form action="_insert.php" method="post">
<input type="hidden" name="---nometabella" value="LavoraIn" />
<input type="hidden" name="CodEnte" value="$cod" />
<label for="seltrib">Tribunale:<sup>*</sup></label>
<select id="seltrib" name="CodTribunale">\n
END;
    $ris = mysql_query("SELECT t.CodTribunale, t.Luogo, t.TipoTribunale FROM
        Tribunali t WHERE t.CodTribunale NOT IN (SELECT li.CodTribunale FROM
        LavoraIn li WHERE li.CodEnte=$cod) ORDER BY t.Luogo;", $conn);
    $ris = ris2array($ris);
190     foreach ($ris as $i) {
        echo ' <option value="' . $i['CodTribunale'] . '">' . $i['Luogo'] . ' - ' .
            $i['TipoTribunale'] . ' (' . $i['CodTribunale'] . ')</option>\n';
    }

```

```

echo<<<END
</select> <br />
<input type="submit" value="Aggiungi" />
</form>\n
END;
    cfieldset ();
}
200 $bott = array(array('etichetta' => 'Dettagli', 'action' => 'dettaglitribunale.php',
    'method' => 'get'));
    if ($liv > 2) $bott[] = array('etichetta' => 'Rimuovi associazione', 'action' =>
        '_delete.php', 'method' => 'post', 'parametri' => array('---nometabella' =>
            'LavoraIn', '---conferma' => 'conferma', 'CodEnte' => $cod));
    query2tab("SELECT t.CodTribunale, t.Luogo, t.TipoTribunale FROM LavoraIn li NATURAL
        JOIN Tribunali t WHERE li.CodEnte=$cod", $conn, 2, $bott, array('CodTribunale'));
    if ($liv > 1) {
        echo "<a href=\"#\" onclick=\"toggle('addtrib')\">Associa tribunale</a>\n";
    }
    cfieldset (); #fine tribunali associati

    afieldset('Cause giudicate', 'cause'); #cause associate
210 query2tab("SELECT CodCausa, Stato, RG, DatiParti FROM VCause c WHERE
        c.CodGiudice=$cod", $conn, 3, array(array('etichetta' => 'Dettagli', 'method' =>
            'get', 'action' => 'dettaglicausa.php')), array('CodCausa'), array());
    cfieldset ();

    cfieldset ();
    break; #fine giudice
}

$giud = !isset($giud) && isset($pers) && $liv>1;
$coll = !isset($coll) && isset($pers) && $liv>2;
220 if ($giud || $coll) { #Promozioni
    afieldset('Promozioni', 'prom');
    echo " <div style=\"float:right; margin-left:20px\">\n";
    if ($giud) {
        afieldset('Aggiungi status di giudice', 'addgiud', 'display:none');
        $rig = array('TipoGiudice' => array('valore' => 'Togato', 'sceltamultipla' =>
            array('Togato' => 'Togato', 'Onorario' => 'Onorario'), 'etichetta' => 'Tipo
            Giudice', 'obbligatorio' => true));
        arrayedit($rig, array('---nometabella' => 'Giudici', 'CodEnte' => $cod), ($liv>1),
            'Aggiungi status', '_insert.php');
        cfieldset ();
    }
    if ($coll) {
230 afieldset('Aggiungi status di collaboratore', 'addcoll', 'display:none');
        $rig = array(
            'Qualifica' => array('valore' => 'Segretario', 'sceltamultipla' =>
                array('Avvocato' => 'Avvocato', 'PraticanteConPatrocinio' => 'Praticante
                con patrocinio', 'Praticante' => 'Praticante', 'Segretario' =>
                'Segretario'), 'obbligatorio' => true),
            'Ripartizione' => array('valore' => ''),
            'password' => array('valore' => '', 'obbligatorio' => 'true', 'etichetta' =>
                'Password', 'password' => true));
        arrayedit($rig, array('---nometabella' => 'Collaboratori', 'CodEnte' => $cod), ($liv>2),
            'Aggiungi status', '_insert.php');
        cfieldset ();
    }
    echo " </div>\n";
    if ($giud) {
240 echo "<a href=\"#\" onclick=\"toggle('addgiud')\">Aggiungi status di
        giudice</a>\n";
    }
}

```

```

    }
    if ($giud && $coll) echo " &mdash; \n";
    if ($coll) {
        echo "<a href=\"#\" onclick=\"toggle('addcoll')\">Aggiungi status di
            collaboratore</a>\n";
    }
    cfieldset();
}

afieldset('Cause', 'causep'); #Cause
250 query2tab("SELECT c.CodCausa, ci.Ruolo, c.Stato, c.RG, CONCAT(t.TipoTribunale, ' di ',
    t.Luogo) AS Tribunale, e.Nome AS NomeGiudice FROM CoinvoltoIn ci NATURAL JOIN
    Cause c NATURAL JOIN Tribunali t JOIN Enti e ON e.CodEnte=c.CodGiudice WHERE
    ci.CodEnte=$cod", $conn, 4, array(array('action'=>'dettaglicausa.php', 'method'=>'get',
    'etichetta'=>'Dettagli')), array('CodCausa'));
cfieldset();

afieldset('Recapiti', 'rec'); #Recapiti
if ($liv > 1) { #nuovo recapito
    $rig = array(
        'TipoRecapito' => array('valore' => 'Telefono', 'menu' => array('Telefono' =>
        'Telefono', 'Indirizzo' => 'Indirizzo', 'Email' => 'Email'), 'etichetta' =>
        'Tipo Recapito', 'obbligatorio' => true),
        'Descrizione' => array('valore' => ''),
        'Recapito' => array('valore' => '', 'lunghezza' => 50, 'obbligatorio' => true));
    afieldset('Nuovo recapito', 'nrec', 'float:right; display:none; margin-left:20px');
260 arrayedit($rig, array('---nometabella' => 'Recapiti', 'CodEnte' => $cod), ($liv>1),
    'Inserisci nuovo', '_insert.php');
    cfieldset();
}
query2tab("SELECT TipoRecapito, Descrizione, Recapito FROM Recapiti r WHERE
    r.CodEnte=$cod", $conn, 1,
    (( $liv>1) ? array(array('action'=> '_delete.php', 'method'=> 'post',
        'etichetta'=> 'Elimina', 'parametri' => array('---nometabella'=>'Recapiti',
        '---conferma'=>'conferma', 'CodEnte'=>$cod))) : array()),
    array('Recapito'));
if ($liv > 1) echo "<a href=\"#\" onclick=\"toggle('nrec')\"> Aggiungi recapito </a>\n";
cfieldset();

tail_trans();
270 ?>

```