www.ipdigital.com
info@ipdigital.com
(+39) 049 000 0000

Creation of SQL tables

```
CREATE TABLE NonComplianceList (
   code                 int AUTO_INCREMENT,
   name                 varchar(32) NOT NULL,
   description          varchar(128),
   PRIMARY KEY(code)
) ENGINE = INNODB;

CREATE TABLE CorrectiveAction (
   code                 int AUTO_INCREMENT,
   name                 varchar(32) NOT NULL,
   description          varchar(128),
   PRIMARY KEY(code)
) ENGINE = INNODB;

CREATE TABLE Company (
   vatNum               char(11),
   name                 varchar(32) NOT NULL,
   address              varchar(96),
   PRIMARY KEY(vatNum)
) ENGINE = INNODB;

CREATE TABLE Supplier (
   vatNum               char(11),
   ISO9001              char(4),
   PRIMARY KEY(vatNum),
   FOREIGN KEY(vatNum) REFERENCES Company(vatNum) ON UPDATE CASCADE ON DELETE
CASCADE
) ENGINE = INNODB;
```

```
CREATE TABLE Customer (
   vatNum               char(11),
   PRIMARY KEY(vatNum),
   FOREIGN KEY(vatNum) REFERENCES Company(vatNum) ON UPDATE CASCADE ON DELETE
CASCADE
) ENGINE = INNODB;

CREATE TABLE CurrentCompany (
   vatNum               char(11),
   PRIMARY KEY(vatNum),
   FOREIGN KEY(vatNum) REFERENCES Company(vatNum) ON UPDATE CASCADE ON DELETE
CASCADE
) ENGINE = INNODB;

CREATE TABLE PersonalData (
   fiscalCode           char(16),
   firstName            varchar(32),
   lastName             varchar(32),
   PRIMARY KEY(fiscalCode)
) ENGINE = INNODB;

CREATE TABLE Employee (
   fiscalCode           char(16),
   job                  varchar(32) NOT NULL,
   role                 varchar(32) NOT NULL,
   company              char(11),
   department           int,
      PRIMARY KEY(fiscalCode),
   FOREIGN KEY(fiscalCode) REFERENCES PersonalData(fiscalCode) ON UPDATE CASCADE ON
DELETE CASCADE,
   FOREIGN KEY(company) REFERENCES CurrentCompany(vatNum) ON UPDATE CASCADE ON
DELETE SET NULL
) ENGINE = INNODB;
```

```
CREATE TABLE Department (
   code                 int AUTO_INCREMENT,
   name                 varchar(32) NOT NULL,
   directorFiscalCode   char(16),
   PRIMARY KEY(code),
   FOREIGN KEY(directorFiscalCode) REFERENCES Employee(fiscalCode) ON UPDATE
CASCADE ON DELETE SET NULL
) ENGINE = INNODB;

/*
 MySQL non permette di inserire un vincolo di integrità relazionale verso una
 tabella che non è ancora stata creata, perciò lo aggiungiamo successivamente.
*/
ALTER TABLE Employee ADD CONSTRAINT FOREIGN KEY(department) REFERENCES
Department(code);

CREATE TABLE User (
   fiscalCode           char(16),
   email                varchar(254) UNIQUE NOT NULL,
   password             char(64),
   PRIMARY KEY(fiscalCode),
   FOREIGN KEY(fiscalCode) REFERENCES Employee(fiscalCode) ON UPDATE CASCADE ON
DELETE CASCADE
) ENGINE = INNODB;

CREATE TABLE Process (
   code                 int AUTO_INCREMENT,
   departmentCode       int,
   name                 varchar(32) NOT NULL,
   description          varchar(128),
   PRIMARY KEY(code),
   FOREIGN KEY(departmentCode) REFERENCES Department(code) ON UPDATE CASCADE ON
DELETE SET NULL
) ENGINE = INNODB;
```

```
CREATE TABLE Product (
   code                 int AUTO_INCREMENT,
   name                 varchar(32) NOT NULL,
   price                double,
   producedByProcess    int,
   type                 varchar(32),
   PRIMARY KEY(code),
   FOREIGN KEY(producedByProcess) REFERENCES Process(code) ON UPDATE CASCADE ON
DELETE SET NULL
) ENGINE = INNODB;

CREATE TABLE ProcessProduct (
   processCode          int,
   productCode          int,
   PRIMARY KEY(processCode, productCode),
   FOREIGN KEY(processCode) REFERENCES Process(code) ON UPDATE CASCADE ON DELETE
CASCADE,
   FOREIGN KEY(productCode) REFERENCES Product(code) ON UPDATE CASCADE ON DELETE
CASCADE
) ENGINE = INNODB;

CREATE TABLE Orders (
   invoiceNum           int AUTO_INCREMENT,
   vatNum               char(11),
   product              int,
   orderDate            date NOT NULL,
   quantity             int DEFAULT 1,
   PRIMARY KEY(invoiceNum),
   FOREIGN KEY(vatNum) REFERENCES Company(vatNum) ON UPDATE CASCADE ON DELETE SET
NULL,
   FOREIGN KEY(product) REFERENCES Product(code) ON UPDATE CASCADE
) ENGINE = INNODB;
```

```
CREATE TABLE Lot (
   shippingCode          char(10),
   orderCode             int NOT NULL,
   deliveryDate          date,
   quantity              int DEFAULT 1,
   PRIMARY KEY(shippingCode),
   FOREIGN KEY(orderCode) REFERENCES Orders(invoiceNum) ON UPDATE CASCADE ON DELETE
CASCADE
) ENGINE = INNODB;

CREATE TABLE NonCompliance (
   code                  int AUTO_INCREMENT,
   lot                   char(10),
   processOrigin         int,
   type                  int,
   repEmployee           char(16),
   date                  date NOT NULL,
   comment               varchar(128),
   PRIMARY KEY(code),
   FOREIGN KEY(type) REFERENCES NonComplianceList(code) ON UPDATE CASCADE ON DELETE
SET NULL,
   FOREIGN KEY(lot) REFERENCES Lot(shippingCode) ON UPDATE CASCADE ON DELETE SET
NULL,
   FOREIGN KEY(processOrigin) REFERENCES Process(code) ON UPDATE CASCADE ON DELETE
SET NULL,
   FOREIGN KEY(repEmployee) REFERENCES Employee(fiscalCode) ON UPDATE CASCADE ON
DELETE SET NULL
) ENGINE = INNODB;
```

```
CREATE TABLE Complaint (
   vatNum               char(11),
   shippingCode         char(10),
   nonComplianceCode    int,
   description          varchar(128),
   answer               varchar(254),
   closed               bit NOT NULL DEFAULT 0,
   PRIMARY KEY (vatNum, nonComplianceCode),
   FOREIGN KEY(vatNum) REFERENCES Customer(vatNum) ON UPDATE CASCADE ON DELETE
CASCADE,
   FOREIGN KEY(shippingCode) REFERENCES Lot(shippingCode) ON UPDATE CASCADE ON
DELETE SET NULL,
   FOREIGN KEY(nonComplianceCode) REFERENCES NonCompliance(code) ON UPDATE CASCADE
ON DELETE CASCADE
) ENGINE = INNODB;

CREATE TABLE NonComplianceAnalysis (
   nonComplianceCode    int,
   manager              char(16),
   employee             char(16),
   expirationDate       date NOT NULL,
   PRIMARY KEY(nonComplianceCode, manager),
   FOREIGN KEY(nonComplianceCode) REFERENCES NonCompliance(code) ON UPDATE CASCADE,
   FOREIGN KEY(manager) REFERENCES Employee(fiscalCode) ON UPDATE CASCADE,
   FOREIGN KEY(employee) REFERENCES Employee(fiscalCode) ON UPDATE CASCADE ON
DELETE SET NULL
) ENGINE = INNODB;

CREATE TABLE NonComplianceCheck (
   nonComplianceCode    int,
   manager              char(16),
   employee             char(16),
   expirationDate       date NOT NULL,
   PRIMARY KEY(nonComplianceCode, manager),
   FOREIGN KEY(nonComplianceCode) REFERENCES NonCompliance(code) ON UPDATE CASCADE,
```

```
   FOREIGN KEY(manager) REFERENCES Employee(fiscalCode) ON UPDATE CASCADE,
   FOREIGN KEY(employee) REFERENCES Employee(fiscalCode) ON UPDATE CASCADE ON
DELETE SET NULL
) ENGINE = INNODB;

CREATE TABLE NonComplianceResult (
   nonComplianceCode    int,
   correctiveActionCode int,
   responsibility       char(11),
   result               varchar(128) NOT NULL,
   cost                 double DEFAULT 0,
   comment              varchar(128),
   PRIMARY KEY(nonComplianceCode),
   FOREIGN KEY(nonComplianceCode) REFERENCES NonCompliance(code) ON UPDATE CASCADE,
   FOREIGN KEY(correctiveActionCode) REFERENCES CorrectiveAction(code) ON UPDATE
CASCADE ON DELETE SET NULL,
   FOREIGN KEY(responsibility) REFERENCES Company(vatNum) ON UPDATE CASCADE ON
DELETE SET NULL
) ENGINE = INNODB;
```

www.ipdigital.com
info@ipdigital.com
(+39) 049 000 0000

Queries

List of non-compliances

Open

```
SELECT DISTINCT NC.*
FROM NonCompliance AS NC
LEFT JOIN NonComplianceAnalysis AS NCA ON NC.code = NCA.nonComplianceCode
WHERE NCA.employee IS NULL;
```

In the intermediate phase

```
SELECT DISTINCT NC.*
FROM NonCompliance AS NC
JOIN NonComplianceAnalysis AS NCA ON NC.code = NCA.nonComplianceCode
LEFT JOIN NonComplianceResult AS NCR ON NC.code = NCR.nonComplianceCode
WHERE NCR.nonComplianceCode IS NULL;
```

Closed

```
SELECT NC.*
FROM NonCompliance AS NC
JOIN NonComplianceResult AS NCR ON NC.code = NCR.nonComplianceCode;
```

Most common non-compliances

A single non compliance can occur in more than one process

```
SELECT NCL.*, processOrigin AS processCode, COUNT(NCL.code) AS NCcount
FROM NonCompliance AS NC
JOIN NonComplianceList AS NCL ON NC.type = NCL.code
GROUP BY NC.type, NC.processOrigin
ORDER BY COUNT(NCL.code) DESC;
```