



Ministero dell'Istruzione, dell'Università e della Ricerca
M070 – ESAME DI STATO DI ISTITUTO TECNICO INDUSTRIALE

CORSO DI ORDINAMENTO

Indirizzo: INFORMATICA

Tema di: INFORMATICA GENERALE, APPLICAZIONI TECNICO SCIENTIFICHE
INFORMATICA

(Testo valevole per i corsi di ordinamento e per i corsi sperimentali del Progetto “Sirio”)

Una Società Aeroportuale commissiona lo sviluppo di un software per la gestione dei controlli passeggeri e merci nei vari punti di dogana presenti in un Aeroporto Internazionale.

Si richiede la memorizzazione, per ogni controllo effettuato, dei dati relativi a:

1. passeggero
cognome, nome, nazionalità, N. passaporto o carta d'identità, aeroporto di provenienza/destinazione, motivo del viaggio;
2. merce trasportata
categoria di appartenenza (generi alimentari, strumentazione elettronica, farmaci, abbigliamento, ...), descrizione, quantità dichiarata;
3. controllo effettuato
punto di controllo, identificativo dell'addetto al controllo, data e ora inizio controllo, ... , esito (nessuna segnalazione, merce respinta, fermo del passeggero, ...), dazio doganale (non dovuto / importo del dazio, se dovuto), note.

La procedura prevede che il funzionario incaricato provveda ad autenticarsi sul sistema e inserisca in tempo reale i dati relativi ad ogni nuovo controllo.

Il sistema dovrà registrare i dati provenienti dai vari punti di controllo e fornire le necessarie funzioni per la segnalazione ai funzionari incaricati, delle situazioni da gestire:

- ritiro della merce sequestrata da parte del personale addetto al controllo;
- apertura della pratica di fermo del passeggero con segnalazione all'organo preposto (Polizia Aeroportuale);
- analisi delle note di controllo e gestione delle eventuali contestazioni;
- chiusura di tutti i rapporti di controllo aperti dagli addetti secondo le loro competenze.

Il candidato realizzi:

- un'analisi della realtà di riferimento, completa delle ipotesi aggiuntive necessarie, a suo motivato giudizio, per la realizzazione di uno schema adeguato dell'architettura proposta;
- uno schema concettuale ed uno schema logico del data base;



Ministero dell'Istruzione, dell'Università e della Ricerca

M070 – ESAME DI STATO DI ISTITUTO TECNICO INDUSTRIALE

CORSO DI ORDINAMENTO

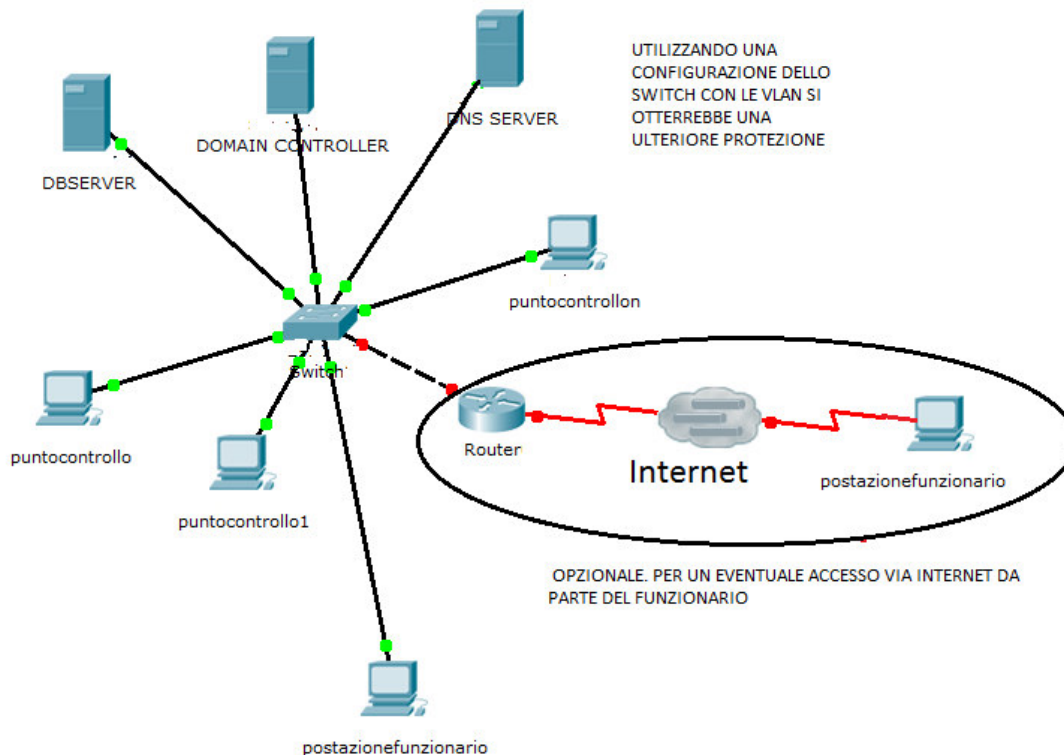
Indirizzo: INFORMATICA

**Tema di: INFORMATICA GENERALE, APPLICAZIONI TECNICO SCIENTIFICHE
INFORMATICA**

(Testo valevole per i corsi di ordinamento e per i corsi sperimentali del Progetto “Sirio”)

- la definizione delle relazioni e le seguenti interrogazioni espresse in linguaggio SQL:
 1. visualizzare i dati di tutti i passeggeri che sono stati controllati in ciascuno dei punti di dogana nell'arco della giornata;
 2. visualizzare per ciascun punto di controllo l'ammontare dei dazi doganali registrati;
 3. calcolare e visualizzare quante merci per ogni categoria sono state respinte dall'inizio dell'anno;
 4. calcolare e visualizzare quante contestazioni sono state registrate da ciascun addetto;
 5. calcolare la durata media dei controlli per ogni punto di controllo nell'arco della giornata;
 6. visualizzare l'elenco, in ordine alfabetico, raggruppato per nazionalità, dei passeggeri in stato di fermo, registrati dall'inizio dell'anno in tutti i punti di controllo;
 7. visualizzare gli addetti in servizio nella giornata, suddivisi per nome del funzionario incaricato.
- la codifica in un linguaggio di programmazione a scelta di un segmento significativo del progetto realizzato.

Il tema propone un classico problema di database condiviso mediante una rete locale. Infatti è logico ipotizzare, anche per evidenti motivi di sicurezza, che i vari punti di dogana siano collegati tra loro mediante una LAN dedicata in modo da ridurre al minimo la possibilità di intrusione da parte di estranei.



Ovviamente sarà a carico del responsabile della rete provvedere ad un controllo sistematico degli accessi e ad una revisione periodica delle password e delle user id dei singoli addetti e funzionari.

Ovviamente, nella creazione del database in Access bisogna prestare particolare attenzione ai permessi di accesso alle singole tabelle, maschere e query in maniera da assicurare il livello più alto di sicurezza.

Infine ipotizziamo che sia il funzionario mediante opportune procedure precostituite che vengono attivate manualmente a gestire la parte relativa all'apertura delle pratiche di fermo e all'inoltro dei dati alle autorità competenti.

Una semplice osservazione: La traccia non chiarisce fino in fondo il compito del funzionario e degli addetti alla dogana.

Sembra comunque alquanto improbabile che in una situazione reale l'addetto al controllo non possa interagire direttamente con il database per inserire personalmente i dati relativi al controllo effettuato, ma debba passare tutti i dati (in forma cartacea?) al funzionario che provvederà (difficilmente in tempo reale) a inserirli nel database. Si potrebbe quindi ipotizzare che l'addetto in prima battuta inserisca nel database i dati relativi al controllo effettuato e sia poi il funzionario a decidere l'azione da intraprendere.

IPOTESI AGGIUNTIVE

- IL FUNZIONARIO È UN ADDETTO CHE DIRIGE ALTRI ADDETTI E CHE PUÒ EFFETTUARE CONTROLLI IN PROPRIO
- UN PASSEGGERO PUÒ SUBIRE PIÙ CONTROLLI NEI VARI PUNTI DI SOSTA CHE MAGARI POSSONO TROVARE QUALCOSA CHE AI QUEGLI È STUCCO

VINCOLI ESPliciti

- V1: (Controllo.Stato = "APERTO") OR (Controllo.Stato = "CHIUSO")
- V2: (Controllo.Composizione = 'S') OR (Controllo.Composizione = 'N')
- V3: (Controllo.Esito = "NESSUNA COMPOSIZIONE") OR (Controllo.Esito = "MERCE RESINATA") OR (Controllo.Esito = "FERMO PASSEGGERO")
- V4: (Controllo.Pazio ≥ 0)
(NB. È ZERO NEGLI 0 NON SU STATO IMPOSSIBILE)
- V5: (Addetto.Ruolo = "FUNZIONARIO") OR (Addetto.Ruolo = "OPERATORE")
- V6: (Esegue.DataIn ≤ Esegue.DataFin)
- V7: (Genere.Descrizione G = "ALIMENTARI") OR (Genere.Descrizione G = "ELETTRICI") OR (Genere.Descrizione G = "FARMACI") OR (Genere.Descrizione G = "OGGETTI D'ARTE")
- V8: (Controllo.TipoDoc = "LA VAGLIA") OR (Controllo.TipoDoc = "CARTA IDENTITÀ")

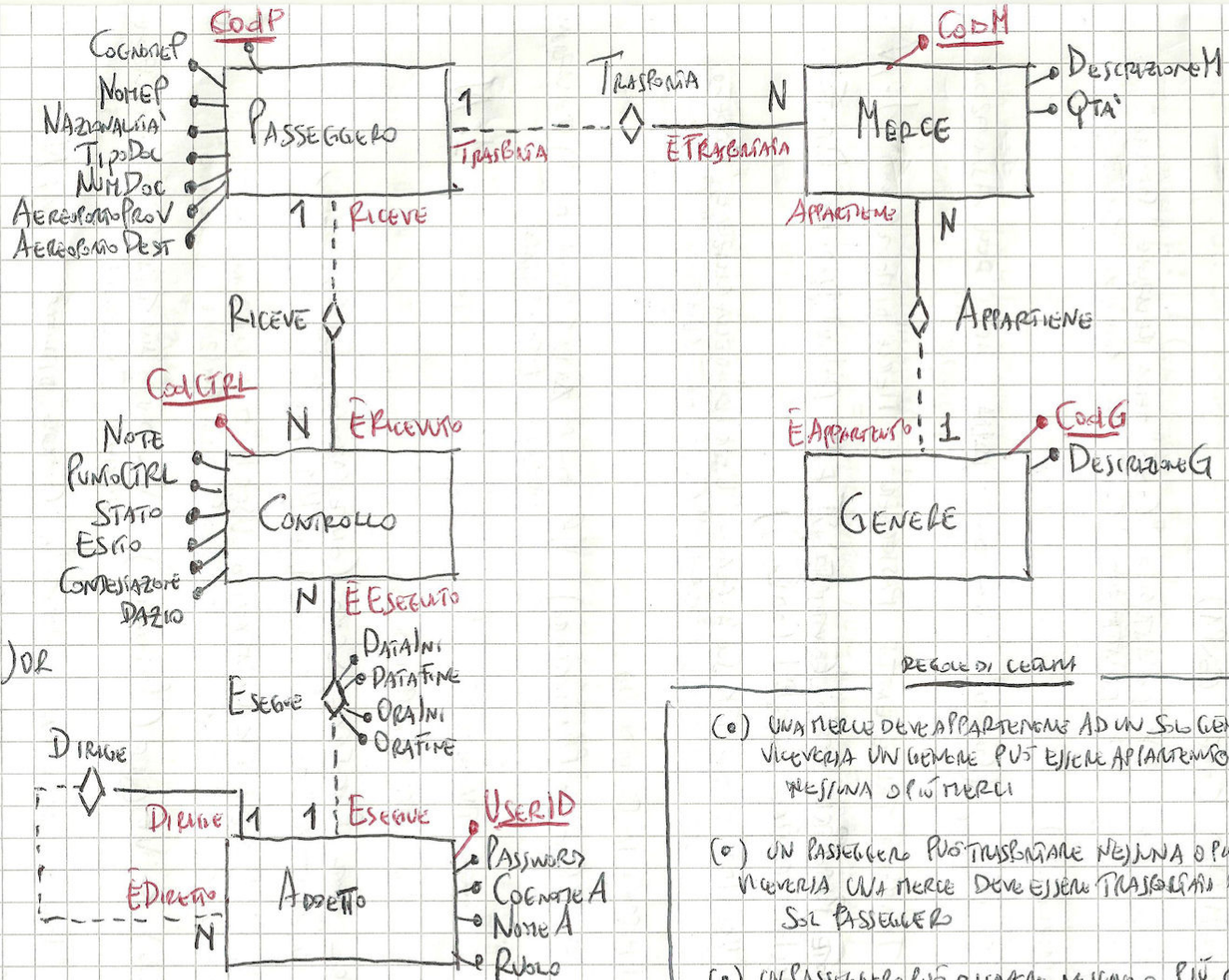


DIAGRAMMA ER
(Scheme concettuale)

REGOLE DI CENSA

- UNA MERCE DEVE APPARTENERE AD UN SOLO GENERE E VICEVERSA UN UOMO PUÒ ESSERE APPARTENUTO DA PIÙ MERCI
- UN PASSEGGERO PUÒ TRASFERIRE NESSUNA O PIÙ MERCI E VICEVERSA UNA MERCE DEVE ESSERE TRASFERITA DA UN SOLO PASSEGGERO
- UN PASSEGGERO PUÒ RICEVERE NESSUNO O PIÙ CONTROLLI E VICEVERSA UN CONTROLLO DEVE ESSERE RICEVUTO DA UN SOLO PASSEGGERO
- UN ADDETTO PUÒ ESEGUIRE NESSUNO O PIÙ CONTROLLI E VICEVERSA UN CONTROLLO DEVE ESSERE ESEGUITO DA UN SOLO ADDETTO
- UN ADDETTO DEVE DIRIGERE UNO O PIÙ ADDETTI (FUNZIONARIO) E VICEVERSA UN ADDETTO PUÒ ESSERE DIRIGITO DA NESSUNO O UN ADDETTO

(B)

SCHEMA LOGICO-RELAZIONALE

(a) MAPPING ASSOCIAZIONE "APPARTURE" TRA LE ENTITÀ "MERCE" E "GENERE" DI RIFERIMENTO N:1

(o) Merce (CodM, Descrizione M, PIA, CodG1, CodP1)

- con l'attributo "CodG1" che è FK sull'attributo "CodG" della relazione "Genere"
- con l'attributo "CodP1" che è FK sull'attributo "CodP" della relazione "Passaggio"

(o) Genere (CodG, Descrizione G)
$$VR_{CodG1}(Merce) \subseteq VR_{CodG}(Genere) \quad VR \text{ dovuto alla totalità dell'associazione diretta "Appartiene"}$$

(b) MAPPING ASSOCIAZIONE "TRASFERITA" TRA LE ENTITÀ "PASSEGGERO" E "MERCE" DI RIFERIMENTO 1:N

(o) PASSEGGERO (CodP, CognomeP, NomeP, Nazionalità, Tipo Doc, NumDoc, Arretrato Prov, Arretrato Dest.)

Merce : Relazione già mappata in precedenza

$$VR_{CodP1}(Merce) \subseteq VR_{CodP}(Passeggero) \quad VR \text{ dovuto alla totalità dell'associazione inversa "È trasferita"}$$

(c) MAPPING ASSOCIAZIONE "RICEVE" TRA LE ENTITÀ "PASSEGGERO" E "CONTRO" DI RIFERIMENTO 1:N

PASSEGGERO : Relazione già mappata in precedenza

(o) Controllo (CodCtrl, Note, Puntoctrl, Stato, Contrassegno, Esito, Dazio, DataIn, DataFine, Opalm, Opafine, UserID1, CodP2)

- con l'attributo "UserID1" che è FK sull'attributo "UserID" della relazione "Addetto"
- con l'attributo "CodP2" che è FK sull'attributo "CodP" della relazione "Passeggero"

$$VR_{CodP2}(Controllo) \subseteq VR_{CodP}(Passeggero) \quad VR \text{ dovuto alla totalità dell'associazione inversa "È ricevuto"}$$

(d) MAPPING ASSOCIAZIONE "ESegue" TRA LE ENTITÀ "ADDETTO" E "CONTROLLO" DI RIFERIMENTO 1:N

(o) Addetto (UserID, Password, Cognome A, Nome A, Ruolo, UserID2)

- con l'attributo "UserID2" che è FK sull'attributo "UserID" della stessa relazione "Addetto"

Controllo : Relazione già mappata in precedenza

$$VR_{UserID1}(Controllo) \subseteq VR_{UserID}(Addetto) \quad VR \text{ dovuto alla totalità dell'associazione inversa "È eseguito"}$$

(e) MAPPING ASSOCIAZIONE RICORSIVA "DIRETTA" SULL'ENTITÀ "ADDETTO" DI RIFERIMENTO 1:N

ADDETTO : Relazione già mappata in precedenza

$$VR_{UserID1}(Addetto) \subseteq VR_{UserID2}(Addetto) \quad VR \text{ dovuto alla totalità dell'associazione diretta "Dirige"}$$

MAPPING RELAZIONE DEI VINCOLI ESPLICITI

V1 (Controllo): (Stato = "APERTO") OR (Stato = "CHIUSO")

V1 ESPL. → VINCOLO INTRINSECAMENTE O INTERNO SU SINGOLA N-PLA SUL DOMINIO DI UN ATTRIBUTO

V2 (Controllo): (Gestazione = 'S') OR (Gestazione = 'N')

V2 ESPL. → VINCOLO INTRINSECAMENTE O INTERNO SU SINGOLA N-PLA SUL DOMINIO DI UN ATTRIBUTO

V3 (Controllo): (Esito = "NESSUNA GESTAZIONE") OR (Esito = "MERCE ACQUISTA") OR (Esito = "FERMO PASSEGGERO")

V3 ESPL. → VINCOLO INTRINSECAMENTE O INTERNO SU SINGOLA N-PLA SUL DOMINIO DI UN ATTRIBUTO

V4 (Controllo): (Pazio ≥ 0)

V4 ESPL. → VINCOLO INTRINSECAMENTE O INTERNO SU SINGOLA N-PLA SUL DOMINIO DI UN ATTRIBUTO

V5 (Addizio): (Ruolo = "FUNZIONARIO") OR (Ruolo = "OPERATORE")

V5 ESPL. → VINCOLO INTRINSECAMENTE O INTERNO SU SINGOLA N-PLA SUL DOMINIO DI UN ATTRIBUTO

V6 (Controllo): (DataIn ≤ DataFine)

V6 ESPL. → VINCOLO INTRINSECAMENTE O INTERNO SU SINGOLA N-PLA SUL DOMINIO DI PIÙ ATTRIBUTI

V7 (Genere): (DescrizioneG = "AUTOMOBILI") OR (DescrizioneG = "ELETTROMOBILI") OR (DescrizioneG = "FARMACI") OR (DescrizioneG = "OGGETTI D'ARTE")

V7 ESPL. → VINCOLO INTRINSECAMENTE O INTERNO SU SINGOLA N-PLA SUL DOMINIO DI UN ATTRIBUTO

V8 (Passaggio): (TipoDoc = "PASSAGGIO") OR (TipoDoc = "CARICA IDENTITÀ")

V8 ESPL. → VINCOLO INTRINSECAMENTE O INTERNO SU SINGOLA N-PLA SUL DOMINIO DI UN ATTRIBUTO



RELAZIONI IN SQL

CREATE DATABASE Dogana;

CREATE TABLE Genere

```
(
  CodG CHAR(10) NOT NULL,
  DescrizioneG CHAR(255) NOT NULL,
  PRIMARY KEY (CodG),
  CHECK DescrizioneG VALUE IN ("AUTOMOBILI", "ELETTROMOBILI", "FARMACI", "OGGETTI D'ARTE") // V7
);
```

CREATE TABLE Passeggero

```
(
  CodP CHAR(10) NOT NULL,
  CognomeP CHAR(30) NOT NULL,
  NomeP CHAR(30) NOT NULL,
  Nazionalità CHAR(30) NOT NULL,
  TipoDoc CHAR(20) NOT NULL,
  NumDoc CHAR(20) NOT NULL,
  IndirizzoProv CHAR(255) NOT NULL,
  IndirizzoDest CHAR(255) NOT NULL,
  PRIMARY KEY (CodP),
  CHECK TipoDoc VALUE IN ("PASSAGGIO", "CARICA IDENTITÀ") // V8
);
```


CREATE TABLE Addetto

```
(
  UserID CHAR(10) NOT NULL,
  Password CHAR(10) NOT NULL,
  CognomeA CHAR(30) NOT NULL,
  NomeA CHAR(30) NOT NULL,
  Ruolo CHAR(20) NOT NULL,
  UserID2 CHAR(10),
  PRIMARY KEY (UserID),
  FOREIGN KEY UserID2 REFERENCES Addetto (UserID)
  ON UPDATE CASCADE
  ON DELETE SET NULL,
  CHECK Ruolo VALUE IN ("FUNZIONARIO", "OPERATORE") // V5
);
```

// N.B. OPZIONALE PERCHÉ U.A. INV. "E' DUE" NON E' TOTALE

CREATE TABLE Merce

```
(
  CodM CHAR(10) NOT NULL,
  DescrizioneM CHAR(255) NOT NULL,
  Qta DECIMAL(6,2) NOT NULL,
  CodG1 CHAR(10) NOT NULL,
  CodP1 CHAR(10) NOT NULL,
  PRIMARY KEY (CodM),
  FOREIGN KEY CodG1 REFERENCES Generale (CodG)
  ON UPDATE CASCADE
  ON DELETE CASCADE,
  FOREIGN KEY CodP1 REFERENCES Passerella (CodP)
  ON UPDATE CASCADE
  ON DELETE CASCADE
);
```

// N.B. ASS. DUE "ACQUANTO" E' TOTALE
// N.B. ASS. INVERIA "ETRASSAMENTO" E' TOTALE

CREATE TABLE Comando

```
(
  CodCTRL CHAR(10) NOT NULL,
  Note CHAR(255) NOT NULL,
  Stato CHAR(10) NOT NULL,
  Confezionamento CHAR(1) NOT NULL,
  Esito CHAR(30) NOT NULL,
  Pazio DECIMAL(7,2) NOT NULL,
  DataINI DATE NOT NULL,
  DataFINE DATE,
  DataINI DATE NOT NULL,
  DataFINE TIME,
  UserID1 CHAR(10) NOT NULL,
  CodP2 CHAR(10) NOT NULL,
  PRIMARY KEY (CodCTRL),
  FOREIGN KEY UserID1 REFERENCES Addetto (UserID)
  ON UPDATE CASCADE
  ON DELETE CASCADE,
  FOREIGN KEY CodP2 REFERENCES Passerella (CodP)
  ON UPDATE CASCADE
  ON DELETE CASCADE,
  CHECK Stato VALUE IN ("APERTO", "CHIUSO"), // V1
  CHECK Confezionamento VALUE IN ("S", "N"), // V2
  CHECK Esito VALUE IN ("NESSUNA COMESSIONE", "MERCE RESPINSA", "FERMO PASSERELLA") // V3
  CHECK (PAZIO >= 0), // V4
  CHECK (DATAINI < DATAFINE) // V6
);
```

// N.B. ASS. INV. "E' ELEVAZIONE" E' TOTALE
// N.B. ASS. INV. "E' RICARICA" E' TOTALE

D Query SQL

Q1

Q1 SQL = SELECT DISTINCT PuntCtrl, CognomeP, NomeP, Nazionalità
FROM Passeggero, Controllo
WHERE (CodP = CodP2) AND (DataIn = [DataX]);
- eff

NB. "Nell'arco della giornata" è stato inserito "Nell'arco di una giornata inserita dall'utente"

Q1 SQL = $\prod_A \left(\sigma_P \left(\begin{matrix} \text{Passeggero} & \times & \text{Controllo} \\ \text{CodP} = \text{CodP2} \end{matrix} \right) \right)$ — efficiente

A = { PuntCtrl, CognomeP, NomeP, Nazionalità }

P = { DataIn = [DataX] }

Q1 SQL = $\prod_A \left(\begin{matrix} \text{Passeggero} & \times & \sigma_P(\text{Controllo}) \\ \text{CodP} = \text{CodP2} \end{matrix} \right)$
+ eff

Q1 SQL = SELECT DISTINCT PuntCtrl, CognomeP, NomeP, Nazionalità
FROM Passeggero, (SELECT *
FROM Controllo
WHERE (DataIn = [DataX]))
WHERE (CodP = CodP2);

Q2

Q2 SQL = SELECT PuntCtrl, SUM(Pazio) AS TotalePazio
FROM Controllo
GROUP BY (PuntCtrl);

Q3

Q3 SQL = SELECT DescrizioneG AS Genere, COUNT(*) AS TotaleRisposte
FROM Merce, Genere, Passeggero, Controllo
WHERE (CodG = CodG) AND (CodP1 = CodP) AND (CodP = CodP2)
AND (Esito = "Merce Respinta")
GROUP BY (DescrizioneG);

Q4

Q4 SQL = SELECT UserID, COUNT(*) AS TotaleCompletamenti
FROM Appello, Controllo
WHERE (UserID = UserID1) AND (Completamento = 'S')
GROUP BY (UserID);

Q5

Q5 SQL = SELECT PuntCtrl, AVG(DataFine - DataIn) AS MediaGiorni
FROM Controllo
WHERE (DataIn = [DataX])
GROUP BY (PuntCtrl);

// NB è calcolata in giorni
// per cui 0 non significa ore
// Non la durata ⇒ ore

Q6

Q6 SQL = SELECT NAZAWA, Cognito, Nome
FROM PASSEGGIERS, CONTINUES
WHERE (CognP = CognP2) AND (YEAR(DATIM) = [AnnoX])
GROUP BY NAZAWA, CognitoP, NomeP
ORDER BY CognitoP, NomeP

Q7

Q7 SQL = SELECT DISTINCT t1.Cognito AS CognitoTune,
t1.Nome AS NomeTune, t2.Cognito AS CognitoAddio,
t2.Nome AS NomeAddio
FROM Addio AS t1, Addio AS t2 CONTINUES
WHERE (t1.UserID = t2.UserID2) AND
(t2.UserID = CONTINUES.UserID1) AND (DATIM = [DATAX]);

// NB IN QUESTA QUERY GLI ADDIO CHE IN PRIMA CONDIZIONE [DATAX] HANNO
PER IL SUO Cognito A SINISTRA.

E) CODIFICA DI UN LINGUAGGIO DI PROGRAMMAZIONE PER IL WEB A SEGUITO DI UN SEGMENTO SIGNIFICATIVO DEL PROGETTO REALIZZATO

ESISTONO MOLTE OPZIONI DISPONIBILI PER LO SVILUPPO DELLA COMPONENTE LATO SERVER.

UNA SOLUZIONE POSSIBILE UTILIZZANDO IL LINGUAGGIO DI PROGRAMMAZIONE **PHP** CHE CONSENTE DI ARRICCHIRE LE PAGINE WEB DI CODICE SCRIPT CHE SIANO ESEGUITO DIRETTAMENTE SUL SERVER.

INOLTRE PHP CONSENTE DI IMPLEMENTARE UN MOTORE DI SCRIPTING LATO SERVER MOLTO EFFICIENTE E MULTIPATAFORMA CON UN BUON SUPPORTO DELLA CONNESSIONE VERSO DATABASE DIVERSI (ES. DBASE, ORACLE, MySQL ETC.). ATTRAVERSO L'USO DI COMPONENTI STANDARD.

NEGLI SPECIFICI IPOTESI DI INTERFACCIA CON UN DBMS **MySQL** ANCHE IL MULTIPATAFORMA E PIU' SEMPLICE DA UTILIZZARE.

OVVIAMENTE PER POTER UTILIZZARE PHP ALL'INTERNO DI PAGINE WEB E' NECESSARIO AVER INSTALLATO SUL PROPRIO SISTEMA UN WEBSERVER COME AD ESEMPIO **APACHE**.

A COMPLETARE L'ARCHITETTURA SOFTWARE POSSIBILE CONNETTERE POSSONO ESSERE SCELTI DI SISTEMI OPERATIVI OPEN SOURCE COME **UNIX** O **LINUX** NELLA VERSIONE SERVER E CLIENT.

UTILIZZANDO INOLTRE IL LINGUAGGIO **SQL** IN MODALITA' EMBEDDED (OSSIA OGGI IN PHP) PER LA CREAZIONE E PER L'INTERAZIONE DELLE TABELLE DEL NOSTRO DATABASE

Riassumendo

PER POTER ESEGUIRE UNA QUERY SUL DB MySQL UTILIZZANDO IL LINGUAGGIO PHP OCCORRE EFFETTUARE I SEGUENTI PASSI:

- (1) VALORIZZAZIONE DEI PARAMETRI FRAMMENTARI CIOE' L'IP ADDRESS DEL HOST CHE OSPITA I DATI (**\$db-host**) L'UTENTE CHE EFFETTERA' LA QUERY (**\$db-user**) LA SUA PASSWORD EVENTUALMENTE CREATA (**\$db-pass**) ED IL NOME DEL DB SUL QUALE SI INTENDE EFFETTUARE LA QUERY (**\$db-name**)
- (2) CONNESSIONE AL DB : ATTRAVERSO L'USO DELLA FUNZIONE **mysql_connect()**
- (3) SCELTA DEL DB SU CUI OPERARE : COME ALL'USO DELLA FUNZ. **mysql_select_db()**
- (4) PREPARAZIONE SECONDO LA SINTASSI DEL SQL DELLA STRUTTURA DI QUERY (**\$query**)
- (5) ESECUZIONE QUERY SUL DB : ATTRAVERSO L'USO DELLA FUNZ. **mysql_query()**
- (6) RECUPERO DELLE RIGHE (NUMERO) CHE SODDISFANO LA QUERY ATTRAVERSO L'USO DELLA FUNZIONE **mysql_num_rows()**
- (7) GESTIONE DEL RECUPERO ORDEM DEL INFORMATION RECEVUTO
- (8) LIBERAZIONE MEMORIA PER LA QUERY : USO DELLA FUNZ. **mysql_free_result()**
- (9) LIBERAZIONE MEMORIA PER CONNESSIONE : USO DELLA FUNZ. **mysql_close()**

N.B. L'USO DELL'ARRAY GLOBALE **\$_POST[]** PRELIEVO DAL PHP SOSTITUENDO LA ACQUISIZIONE DEI PARAMETRI COME IL CODICE DELLA PAGINA IN CARICATA, ATTRAVERSO L'USO DI UN **FORM HTML** CON CAMPI DI TESTO DI NOME "COLONNINA".
L'USO DELLA FUNZIONE **die()** IN AGGIUNTA CON LA FUNZIONE **mysql_error()** PERMETTE, IN CASO DI ERRORE RILEVATO DI POTER COSTRUIRE UNA PAGINA SEGNALEZIONE DI ERRORE CONCORDATA AL NUMERO DI ERRORE MySQL RICEVUTO.