**PROVA D’ESAME 2013**

**SCHEMA LOGICO**

passeggeri (idPasseggero, nome, nazionalita, numDoc, aereoportoProv, aereoportoDest, motivo)

merciTrasportate (idMerce, categoria, descrizione, qt, idP\*, idC\*)

addetti (idAdd, nome, cognome)

controlliEffettuati (idC, puntoControllo, idAdd\*, idFunz\*, idP\*, dataControllo, oraControllo, esito, dazioDoganale, note)

funzionari (idFunz, nome, cognome, username, psw)

* Se è N a N: si crea una nuova entità, le chiavi esterne vanno verso le altre due entità collegate
* Se è 1 a N: no nuova entità e si mette la chiave esterna verso l’entità dalla parte a cui appartiene l’1
* Se è 1 a 1: si uniscono in una sola entità

**CREATE**

CREATE TABLE "passeggeri" (

"idPasseggero" INTEGER NOT NULL,

"nome" VARCHAR(25) NOT NULL,

"nazionalita" CHAR(3) NOT NULL,

"numDoc" CHAR(10) NOT NULL,

"aereoportoProv" VARCHAR(30) NOT NULL,

"aereoportoDest" VARCHAR(30) NOT NULL,

"motivo" VARCHAR(30) NOT NULL,

PRIMARY KEY("idPasseggero" AUTOINCREMENT)

);

CREATE TABLE "merciTrasportate" (

"idM" INTEGER NOT NULL,

"categoria" VARCHAR NOT NULL CHECK(categoria IN ('generi alimentari', 'strumentazione elettronica', 'farmaci', 'abbigliamento')),

"descrizione" TEXT NOT NULL,

"qt" INTEGER NOT NULL,

PRIMARY KEY("idM" AUTOINCREMENT)

FOREIGN KEY("idP") REFERENCES "passeggeri"("idPasseggero") ON DELETE RESTRICT ON UPDATE CASCADE

FOREIGN KEY("idC") REFERENCES "controlliEffettuati"("idC") ON DELETE RESTRICT ON UPDATE CASCADE

);

CREATE TABLE "controlliEffettuati" (

"idC" INTEGER NOT NULL,

"puntoControllo" VARCHAR(20) NOT NULL,

"idAdd" INTEGER NOT NULL,

"idFunz" INTEGER NOT NULL,

"dataControllo" DATE NOT NULL,

"oraControllo" TIME NOT NULL,

"esito" VARCHAR(30) NOT NULL CHECK("esito" IN ('nessuna segnalazione', 'merce respinta', 'fermo del passeggero')),

"dazioDoganale" VARCHAR(30) NOT NULL CHECK("dazioDoganale" IN ('non dovuto', 'importo del dazio', 'se dovuto')),

"note" TEXT NOT NULL,

PRIMARY KEY("idC" AUTOINCREMENT),

FOREIGN KEY("idAdd") REFERENCES "addetti"("idAdd") ON DELETE RESTRICT ON UPDATE CASCADE,

FOREIGN KEY("idFunz") REFERENCES "funzionari"("idFunz") ON DELETE RESTRICT ON UPDATE CASCADE

FOREIGN KEY("idP") REFERENCES "passeggeri"("idPasseggero") ON DELETE RESTRICT ON UPDATE CASCADE

);

CREATE TABLE "funzionari" (

"idFunz" INTEGER NOT NULL,

"nome" VARCHAR(25) NOT NULL,

"cognome" VARCHAR(30) NOT NULL,

"username" VARCHAR(20) NOT NULL,

"psw" VARCHAR(20) NOT NULL,

PRIMARY KEY("idFunz" AUTOINCREMENT)

);

CREATE TABLE "addetti" (

"idAdd" INTEGER NOT NULL,

"nome" VARCHAR(25) NOT NULL,

"cognome" VARCHAR(30) NOT NULL,

PRIMARY KEY("idAdd" AUTOINCREMENT)

);

CREATE TABLE "controlliMerci" (

"idC" INTEGER NOT NULL,

"idM" INTEGER NOT NULL,

PRIMARY KEY("idC","idM"),

FOREIGN KEY("idC") REFERENCES "controlliEffettuati"("idC") ON DELETE RESTRICT ON UPDATE CASCADE,

FOREIGN KEY("idM") REFERENCES "merciTrasportate"("idM") ON DELETE RESTRICT ON UPDATE CASCADE

);

| **Attributo** | **Tipo e dimensione** | **Dominio o vincoli** | **Chiave Primaria Candidata Secondaria** | **Obbligatorio** | **Semplice**  **Composto** | **Multiplo** |
| --- | --- | --- | --- | --- | --- | --- |
| PASSEGGERI |  |  |  |  |  |  |
| idP | INT | AI | PK | X | S | NO |
| nome | VARCHAR(20) |  | candidata | X | S | NO |
| nazionalita | CHAR(3) | Check ita, ecc |  | X | S | NO |
| numDoc | VARCHAR(10) | Check id o altro doc |  | X | S | NO |
| aereoportoProv | VARCHAR(25) |  |  | X | S | NO |
| aereoportoDest | CHAR(5) | REGEX (1num,4lettere) | SEC | X | C | NO |
| motivo | VARCHAR(35) |  |  | X | S | NO |
| MERCITRASPORTATE |  |  |  |  |  |  |
| idMerce | INT | AI | PK | X | S | NO |
| puntoControllo | VARCHAR(25) | CHECK IN | sec | X | S | NO |
| descrizione | TEXT |  |  | X | S | NO |
| qt | INT |  |  | X | S | NO |
| CONTROLLIEFFETTUATI |  |  |  |  |  |  |
| idC | INT | AI | PK | X | S | NO |
| puntoControllo | VARCHAR(30) |  |  | X | S | NO |
| idAdd | INT |  | FK | X | S | NO |
| dataControllo | DATE | CHECK <dataAttuale |  | X | S | NO |
| oraControllo | TIME |  |  | X | S | NO |
| esito | VARCHAR(30) | CHECK IN |  | X | S |  |
| dazioDoganale | VARCHAR(30) | CHECK IN |  | X | S |  |
| note | TEXT |  |  | X | S |  |
| FUNZIONARI |  |  |  |  |  |  |
| idFunz | INT | AI | PK | X | S | NO |
| nome | VARCHAR(25) |  |  | X | S | NO |
| cognome | VARCHAR(30) |  |  | X | S | NO |
| username | VARCHAR(25) | U |  | X | S | NO |
| psw | VARCHAR(12) | REGEX(1maiusc, 1 carattere speciale, numeri, minimo 8 tot) |  | X | S | NO |
| ADDETTI |  |  |  |  |  |  |
| idAdd | INT | AI | PK | X | S | NO |
| nome |  |  |  | X | S | NO |
| cognome | VARCHAR(30) |  |  | X | S | NO |
| CONTROLLIADDETTI |  |  |  |  |  |  |
| idC | INT |  | PK, FK | X | S | NO |
| idAdd | INT |  | PK, FK | X | S | NO |
| CONTROLLIFUNZIONARI |  |  |  |  |  |  |
| idC | INT |  | PK, FK | X | S | NO |
| idFunz | INT |  | PK, FK | X | S | NO |
| CONTROLLIPASSEGGERI |  |  |  |  |  |  |
| idC | INT |  | PK, FK | X | S | NO |
| idP | INT |  | PK, FK | X | S | NO |
| CONTROLLIMERCI |  |  |  |  |  |  |
| idC | INT |  | PK, FK | X | S | NO |
| idM | INT |  | PK, FK | X | S | NO |

QUERY

-- 1. Visualizzare i dati di tutti i passeggeri controllati in ciascun punto di dogana nella giornata

SELECT p.\*, c.puntoControllo

FROM passeggeri p

JOIN controlliPasseggeri cp ON p.idPasseggero = cp.idP

JOIN controlliEffettuati c ON cp.idC = c.idC

WHERE c.dataControllo = DATE('now');

-- 2. Ammontare dei dazi doganali registrati per ciascun punto di controllo

SELECT puntoControllo, SUM(CASE WHEN dazioDoganale != 'non dovuto' THEN CAST(dazioDoganale AS INTEGER) ELSE 0 END) AS totaleDazi

FROM controlliEffettuati

GROUP BY puntoControllo;

-- 3. Numero di merci respinte per categoria dall'inizio dell'anno

SELECT m.categoria, COUNT(\*) AS merciRespinte

FROM merciTrasportate m

JOIN controlliMerci cm ON m.idM = cm.idM

JOIN controlliEffettuati c ON cm.idC = c.idC

WHERE c.esito = 'merce respinta' AND strftime('%Y', c.dataControllo) = strftime('%Y', 'now')

GROUP BY m.categoria;

-- 4. Numero di contestazioni registrate da ciascun addetto

SELECT a.idAdd, a.nome, a.cognome, COUNT(\*) AS numeroContestazioni

FROM addetti a

JOIN controlliAddetti ca ON a.idAdd = ca.idAdd

JOIN controlliEffettuati c ON ca.idC = c.idC

WHERE c.note != ''

GROUP BY a.idAdd;

-- 5. Durata media dei controlli per ogni punto di controllo nella giornata

SELECT puntoControllo, AVG(julianday(oraControllo) - julianday(oraControllo)) AS durataMedia

FROM controlliEffettuati

WHERE dataControllo = DATE('now')

GROUP BY puntoControllo;

-- 6. Elenco dei passeggeri in stato di fermo, ordinato alfabeticamente e raggruppato per nazionalità

SELECT nazionalita, nome, cognome

FROM passeggeri p

JOIN controlliPasseggeri cp ON p.idPasseggero = cp.idP

JOIN controlliEffettuati c ON cp.idC = c.idC

WHERE c.esito = 'fermo del passeggero' AND strftime('%Y', c.dataControllo) = strftime('%Y', 'now')

ORDER BY nazionalita, nome, cognome;

-- 7. Addetti in servizio nella giornata, suddivisi per funzionario incaricato

SELECT f.nome AS NomeFunzionario, f.cognome AS CognomeFunzionario, a.nome AS NomeAddetto, a.cognome AS CognomeAddetto

FROM funzionari f

JOIN controlliFunzionari cf ON f.idFunz = cf.idFunz

JOIN controlliEffettuati c ON cf.idC = c.idC

JOIN controlliAddetti ca ON c.idC = ca.idC

JOIN addetti a ON ca.idAdd = a.idAdd

WHERE c.dataControllo = DATE('now')

ORDER BY f.cognome, f.nome;