

PROGETTO DI BASI DI DATI

COMPAGNIA ARMATRICE

SEBASTIANO BERTOLIN
A.A. 2015/2016
MATRICOLA:1102978

CARLO BENETAZZO
A.A. 2016/2017
MATRICOLA:1102979

Account Laboratorio: sbertoli
(bd2016.sh)

Indice

Abstract.....	3
Descrizione dei requisiti.....	3
Progettazione concettuale.....	6
Entità e Attributi.....	7
Relazioni.....	8
Ristrutturazione dello schema E-R.....	11
Glossario.....	12
Queries e Procedures.....	15
Queries.....	15
Procedures.....	19
Triggers.....	22
Functions.....	25

Progetto realizzato con:

S/O Ubuntu 16.04.1 LTS 64-bit - Linux version 4.4.0-59-generic

S/O-2 Windows 10 Home

mySQL Ver. 14.14 Distrib 5.7.16, for Linux (x86_64);

Editor di testo **Atom** Ver. 1.12.4;

Word processor **LibreOffice Writer** Ver. 5.1.4.2

Editor usato per gli schemi **Dia** Ver. 0.97.3

Abstract

“Foina Shipping Company” è una compagnia armatrice di navi mercantili che si occupa del trasporto di diverse materie prime per l’industria (prodotti petroliferi, chimici e minerari).

La flotta della società è composta da due diverse tipologie di navi: chimichiere (navi cisterne adibite al trasporto di prodotti petroliferi e prodotti dell’industria chimica/petrolchimica) e portarinfusa (navi dotate di grandi stive compartimentate per il trasporto di materiali grezzi provenienti dall’industria mineraria).

Lo scopo della base di dati è quello di tenere uno storico di posizione, equipaggio e merce trasportata di ogni nave, per poter compiere operazioni sui dati, quali calcolo di tempistiche di spostamento, cambi di equipaggio e allo stesso tempo essere a conoscenza dell’attuale stato e posizione della nave.

Descrizione dei requisiti

Il progetto si propone di realizzare una base di dati per gestire e raccogliere informazioni riguardanti l’amministrazione della compagnia, la quale è interessata a conoscere lo stato, la posizione delle sue navi e la gestione dell’equipaggio.

L’entità principale che si vuole definire è **Nave**, atta a rappresentare le navi della compagnia. Ogni nave è caratterizzata da:

- Numero IMO: codice identificativo unico per ogni nave. Viene assegnato al momento della costruzione;
- Nome;
- Classe: le navi sono, in genere, costruite in una serie di poche unità presso uno stesso cantiere navale. La classe identifica la serie di appartenenza e la tipologia di nave;
- Stato di bandiera: indica lo Stato che attribuisce la propria nazionalità sulla nave;
- Anno di costruzione;
- Stato corrente: specifica il più recente stato operativo della nave (es: ‘in navigazione’, ‘all’ancora’, ‘ormeggiata’, ecc);

- Ultima posizione: l'ultima posizione ricevuta conosciuta;
- Data e ora ultima posizione;
- Velocità attuale.

Ogni nave appartiene a una **Classe** che ne identifica la tipologia e le dimensioni. Le navi che appartengono a una stessa classe sono dette gemelle. Della classe si è interessati a definire:

- Nome: è dato dal nome della prima nave di una serie costruita da un progetto di base comune presso un determinato cantiere, il Nome è identificatore unico della classe;
- Cantiere di costruzione: nome e nazione del cantiere di costruzione;
- Tipo: indica la categoria della nave e le tipologie di prodotti che può trasportare;
- Livello di facilities richiesto;
- Dimensioni della nave: lunghezza, altezza e larghezza;
- Velocità di crociera: indica la velocità di operazione più economica della nave a pieno carico;
- Velocità massima;
- Equipaggio massimo: l'equipaggio massimo che la nave può tenere a bordo.

L'entità **Viaggio** specifica il trasporto effettuato da una nave con un determinato carico. Un nuovo viaggio inizia quando viene prelevato il carico (corrisponde alla fine del viaggio precedente).

Viaggio è caratterizzato da:

- Numero viaggio: un numero che va da 1 a N, dove N è il numero totale di viaggi effettuati da una specifica nave;
- Nave: è l'identificativo della nave che ha effettuato il viaggio. Assieme a Numero viaggio identifica univocamente il viaggio;
- Tipo di carico;
- Equipaggio: identifica l'equipaggio che ha effettuato il viaggio;
- Porto di partenza: il porto in cui è stato prelevato il carico;
- Porto di destinazione: porto dove dovrà essere consegnato il carico;
- Data di inizio viaggio;
- Data di fine viaggio.

Ogni viaggio può essere o meno intervallato da scali presso porti di passaggio.

Di ogni **Scalo** si desidera registrare:

- Numero viaggio;
- Nave: insieme a numero viaggio identifica univocamente ogni scalo;
- Data e ora arrivo;
- Data e ora partenza;
- Porto;
- Operazione: rifornimento di provviste o carburante, carico/scarico di merci, manutenzione.

Personale è la raccolta di tutti i dipendenti della compagnia.

L'entità si modella in:

- **Matricola:** codice alfanumerico identificativo univoco di ogni marittimo. È il numero d'iscrizione alle matricole della capitaneria di porto;
- **Nome;**
- **Cognome;**
- **Data di nascita;**
- **Cittadinanza.**

L'entità Personale è suddivisa in sottocategorie, quali: Ufficiale, Tecnico, Marinaio e Cuoco.

Ufficiale è composto da Matricola, Titolo (indica il grado);

Tecnico si suddivide Matricola e Qualifica (può essere Operaio o Elettricista);

Marinaio ha come attributi Matricola e Grado (un marinaio può essere Nostromo, Marinaio scelto o Mozzo);

Infine Cuoco ha Matricola e Qualifica (Cuoco e Aiuto cuoco).

L'equipaggio di una nave è composto dai vari membri del personale.

Dell'**Equipaggio** interessa conoscere:

- La nave in cui il membro dell'equipaggio ha prestato servizio;
- Il membro dell'equipaggio: identificato mediante matricola;
- **Data imbarco:** insieme alla matricola del membro dell'equipaggio e alla nave forma un identificativo unico. La data imbarco corrisponde all'inizio del contratto di arruolamento;
- **Data sbarco.**

Le facilities sono delle infrastrutture portuali che offrono servizi alle navi che sostano nel porto. Ogni tipologia di nave necessita di determinate facilities per effettuare operazioni commerciali o anche solo per poter aver accesso al porto.

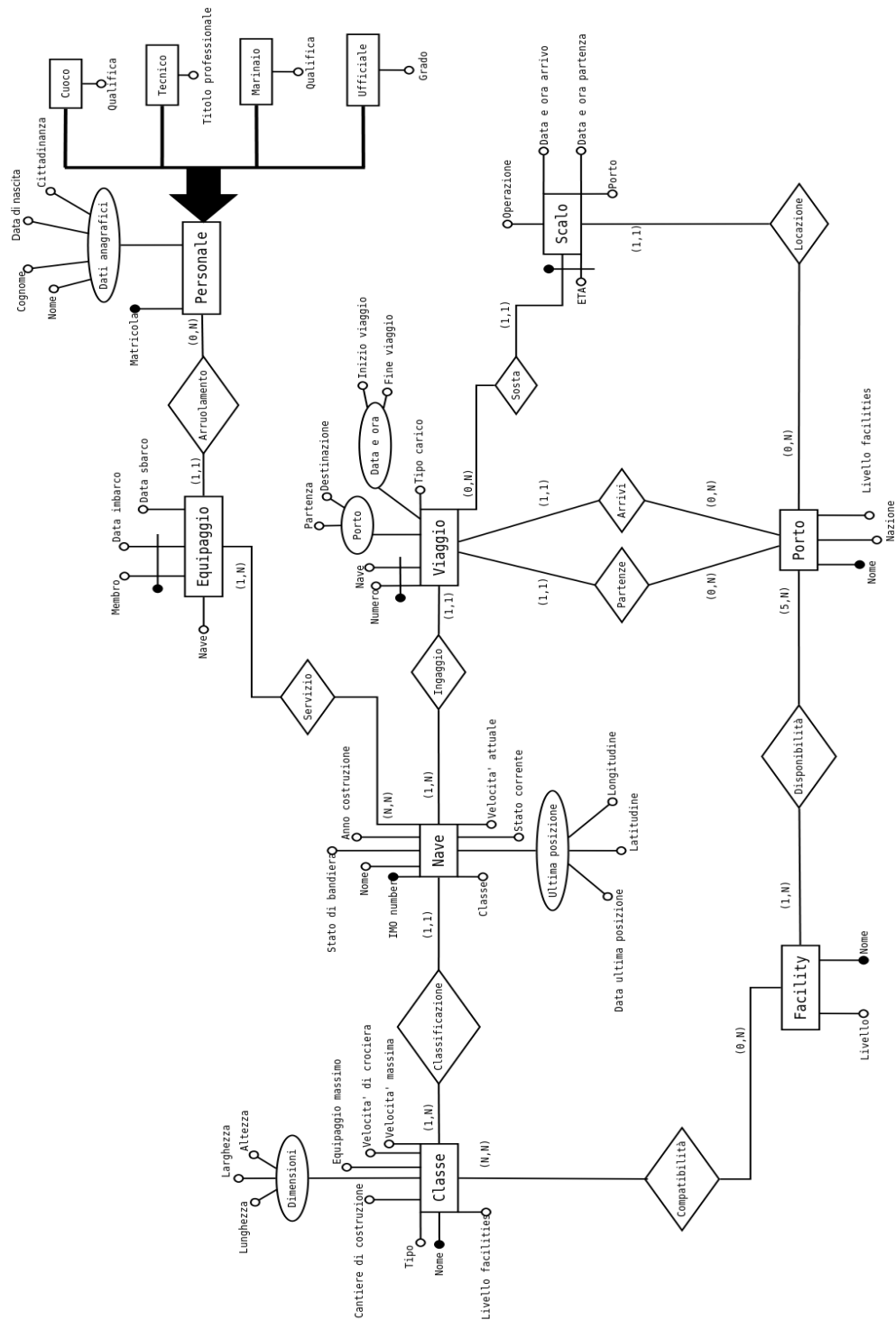
Ogni **Facility** è descritta in una tabella, che indica:

- **Nome:** indentificatore univoco;
- **Livello di appartenenza :** numero da 0 a 4 che rappresenta un insieme di servizi offerti dalla struttura portuale (vigili del fuoco, polizia, rimorchiatori, cantiere, strutture per carico/scarico di determinate merci) al quale appartiene la facility in questione.

L'entità **Porto** è descritta dai seguenti campi:

- **Nome Porto;**
- **Nazione:** insieme a nome Porto identifica univocamente il porto;
- **Livello di facilities offerto:** numero che va da 0 a 4.

(Per una migliore consultazione dello schema si consiglia la visione del file SchemaConcettuale.png contenuto nella cartella schemi)



Entità e Attributi

Nave:

- Numero IMO
- Nome
- Classe
- Stato di bandiera
- Anno di costruzione
- Stato corrente
- Longitudine ultima posizione
- Latitudine ultima posizione
- Data e ora ultima posizione
- Velocità attuale

Classe:

- Nome
- Cantiere
- Tipo
- Capacità di carico
- Livello di facilities richiesto
- Lunghezza
- Altezza
- Larghezza
- Velocità di crociera
- Velocità massima
- Equipaggio massimo

Viaggio:

- Numero viaggio
- Nave
- Tipo di carico
- Equipaggio
- Porto di partenza
- Porto di destinazione
- Data inizio viaggio
- Data fine viaggio
- ETA

Scalo:

- Numero viaggio
- Nave
- Data e ora arrivo
- ETA
- Data e ora partenza

- Porto
- Nazione
- Operazione

Personale:

- Matricola
- Nome
- Cognome
- Data di nascita
- Cittadinanza

Ufficiale:

- Matricola
- Grado

Tecnico:

- Matricola
- Qualifica

Marinaio:

- Matricola
- Grado

Cuoco:

- Matricola
- Qualifica

Equipaggio:

- Nave
- Membro equipaggio
- Data imbarco
- Data sbarco

Facility:

- Nome
- Livello di appartenenza

Porto:

- Nome porto
- Nazione
- Livello facilities

Relazioni

Classificazione (1 a N): Nave-Classe

- Ogni nave appartiene ad una classe
- A ogni classe può appartenere più di una nave

Ingaggio (1 a N): Nave-Viaggio

- Un viaggio è effettuato da una sola nave
- Una nave può effettuare molti viaggi

Servizio (N a N): Nave-Equipaggio

- Ogni membro dell'equipaggio può aver prestato servizio su più navi
- Ogni nave ha più membri dell'equipaggio

Arruolamento (1 a N): Equipaggio-Personale

- Un membro del personale può essere stato arruolato più volte
- Un membro dell'equipaggio è stato arruolato dal personale

Partenze (1 a N): Viaggio-Porto

- Ogni viaggio ha un porto di partenza
- Un porto può essere partenza di più viaggi

Arrivi (1 a N): Viaggio-Porto

- Ogni viaggio ha un porto di destinazione
- Un porto può essere destinazione di più viaggi

Sosta (1 a N): Viaggio-Scalo

- Ogni viaggio può avere uno o più scali
- Uno scalo è sosta di un viaggio

Locazione (1 a N): Scalo-Porto

- Uno scalo ha luogo in un solo porto
- Un porto può avere più scali

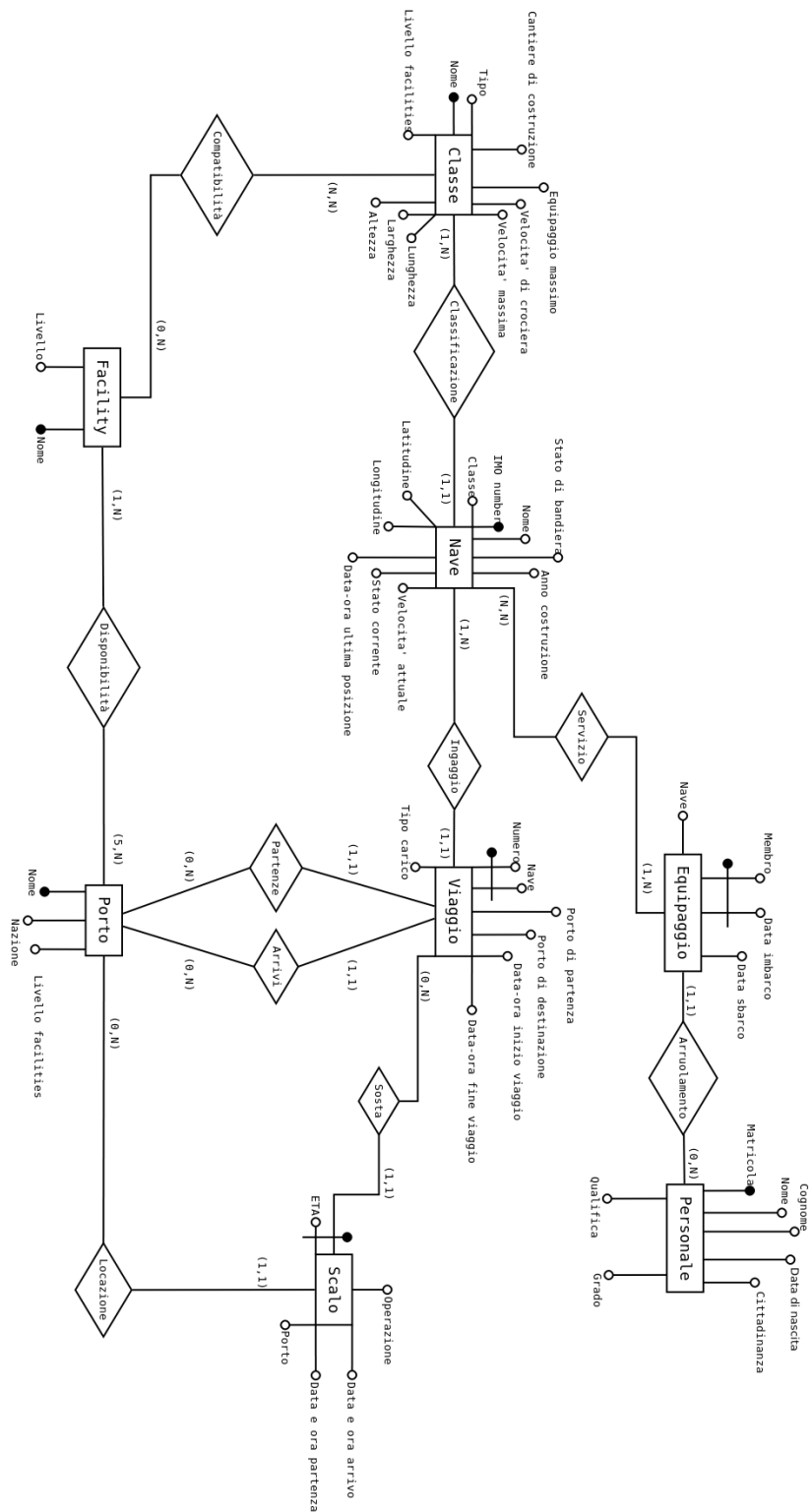
Disponibilità (N a N): Porto-Facility

- Un porto mette a disposizione almeno cinque facilities
- Più facilities possono appartenere a più porti

Compatibilità (N a N): Classe-Facility

- Una classe è compatibile con più facilities
- Ogni facility può essere compatibile con più classi

Ristrutturazione dello schema E-R



(Per una migliore consultazione dello schema si consiglia la visione del File Schema_E-R-Ristrutturato.png contenuto nella cartella schemi)

Eliminazione gerarchia

Le entità figlie 'Cuoco', 'Tecnico', 'Marinaio', 'Ufficiale' sono state accorpate nella entità genitore 'Personale'. Facendo collassare le figlie nel genitore l'entità Personale ha assunto due nuovi attributi:

- **Qualifica**: assume come valori di dominio i nomi delle entità figlie, ;
- **Grado**: specifica il livello professionale conseguito in una data qualifica. Gli attributi delle entità figlie ('Qualifica', 'Titolo professionale', 'Grado') sono stati unificati nell'attributo unico 'Grado';

Glossario

Nave:

- **IMO Number**: è una sequenza di sette numeri assegnata ad ogni nave al momento della costruzione. Viene utilizzato per identificare univocamente ogni tipo di natante superiore alle 100 tonnellate
- **Stato di Bandiera**: Stato che attribuisce la propria nazionalità ad una nave.
- **Latitudine**: coordinata geografica che indica la distanza dall'equatore in gradi (da 0 a 90). Viene rappresentata nel database come un numero di 8 cifre, di cui 6 decimali. Se positiva indica l'emisfero nord, se negativa quello sud.
- **Longitudine**: distanza dal meridiano di Greenwich (da 0 a 180 gradi). Rappresentata nel database con un numero di 9 cifre, di cui 6 decimali, positivamente verso est e negativamente verso ovest.
- **Classe**: Serie di navi aventi caratteristiche e progetto comuni.

Personale:

- **Qualifica**: ruolo ricoperto dal membro del personale.
- **Grado**: ulteriore specifica della qualifica che indica il livello professionale raggiunto in tale mansione.

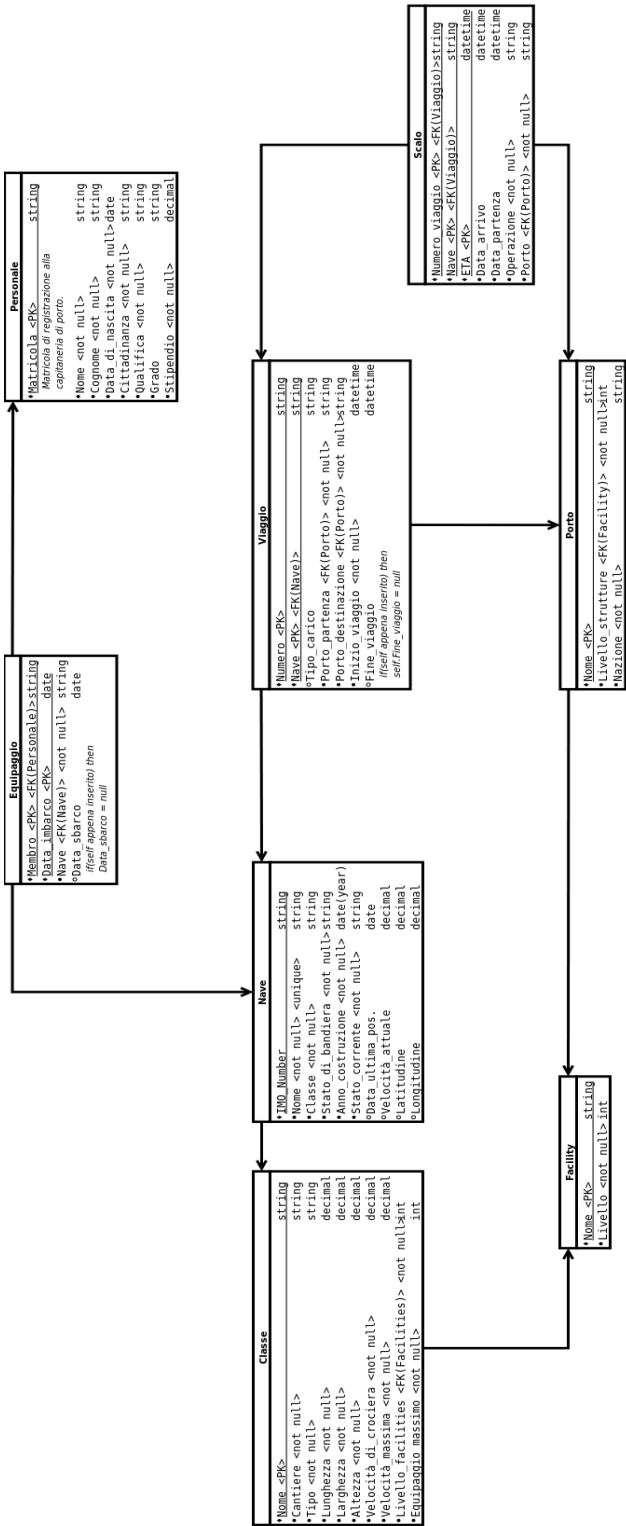
Facility: Struttura portuale che offre un determinato tipo di servizio (es. particolare struttura per carico/scarico di prodotti liquidi, vigili del fuoco).

Scalo:

- **ETA**: acronimo di "Estimated time of arrival" (tempo stimato di arrivo)

Schema Logico

(Per una migliore consultazione dello schema si consiglia la visione del file Schema_Logico.png contenuto nella cartella schemi)



Schema Relazionale

CLASSE (Nome, Cantiere, Tipo, Lunghezza, Larghezza, Altezza, Velocità_di_crociera, Velocità_max, Equipaggio_max)

CLASSIFICAZIONE (Classe, Nave, Tipo)

NAVE (IMO_number, Nome, Classe, Stato_di_bandiera, Anno_costruzione, Stato_corrente, Data_ultima_posizione, Velocità_attuale, Latitudine, Longitudine)

VIAGGIO (Numero, Nave, Tipo_carico, Porto_partenza, Porto_destinazione, Inizio_viaggio, Fine_viaggio)

EQUIPAGGIO (Membro, Data_imbarco, Nave, Data_sbarco)

ARRUOLAMENTO (Membro, Durata_contratto, Ruolo)

PERSONALE (Matricola, Nome, Cognome, Data_di_nascita, Cittadinanza, Qualifica, Grado, Data_qualifica, Stipendio)

ARRIVI (Viaggio, Nave, Porto, Data)

PARTENZE (Viaggio, Nave, Porto, Data)

SCALI (Numero_viaggio, Nave, ETA, Data_arrivo, Data_partenza, Operazione, Porto)

PORTO (Nome, Nazione, Livello_facilities)

FACILITY (Nome, Livello)

Queries e Procedures

Queries

1) Query che restituisce:

- l'identificativo della nave;
- matricola e cognome del membro del personale sbarcante;
- qualifica e grado dei membri del personale che dovranno sostituirsi;
- matricola e cognome del membro del personale imbarcante;
- ETA al prossimo scalo;
- il numero di viaggio della nave;
- l'operazione che verrà effettuata all'arrivo;
- il porto in cui avverrà lo scalo.

```
CREATE OR REPLACE VIEW Sostituzioni AS
SELECT
K.Nave,K.MatricolaS,K.NomeS,K.CognomeS,K.Qualifica,K.Grado,K.MatricolaI,K.NomeI,K.CognomeI,ETA,No_Viag,Operazione,Porto FROM(
SELECT P_sbar.Nave as Nave, P_sbar.Matricola as MatricolaS, P_sbar.Nome as NomeS,
P_sbar.Cognome as CognomeS, P_imb.Matricola AS MatricolaI, P_imb.Nome AS NomeI,
P_imb.Cognome AS CognomeI, P_imb.Qualifica, P_imb.Grado
FROM (SELECT *
      FROM Personale JOIN Equipaggio ON Matricola = Membro
      WHERE Data_sbarco = "2016-02-22" AND Nave = "IM05641147") as P_sbar
JOIN (SELECT *
      FROM Personale JOIN Equipaggio ON Matricola = Membro
      WHERE Data_imbarco = "2016-02-22" AND Nave = "IM05641147") as P_imb
WHERE P_sbar.Qualifica = P_imb.Qualifica
AND P_sbar.Grado = P_imb.Grado) AS K
JOIN (SELECT Nave, CAST(ETA AS DATE) AS ETA, Numero_viaggio AS No_Viag, Operazione,
Porto,          CAST(Data_arrivo AS DATE) AS Data_arr, CAST(Data_partenza AS DATE) AS
Data_part
      FROM Scalo) AS S
WHERE S.Nave = K.Nave
AND S.Data_arr = "2016-02-22";
```

```
mysql> select * from Sostituzioni;
```

Nave	MatricolaS	CognomeS	Qualifica	Grado	MatricolaI	CognomeI	ETA	No_Viag	Operazione	Porto
IM05641147	CTT3587	Vasile	Tecnico	Operaio	PA72541	Costa	2016-02-22	39	Ispezione/Personale	Port of Siracusa
IM05641147	PA58954	DeDomenico	Marinaio	Mozzo	LT42001	Pisani	2016-02-22	39	Ispezione/Personale	Port of Siracusa

```
2 rows in set (0,00 sec)
```

2) Query che trova le navi al momento impiegate in operazioni portuali e restituisce:

- la nave(IMO_number);
- il tipo di carico trasportato dalla nave;
- il comandante(Matricola, Nome, Cognome) imbarcato sulla nave;
- il primo ufficiale di coperta(Matricola, Nome, Cognome) imbarcato sulla nave.

```
CREATE OR REPLACE VIEW OperazioniPortualiCorrenti AS
SELECT n.IMO_number, n.Nome AS Nome_nave, Numero AS Numero_viaggio, Tipo_carico,
Porto_destinazione AS Porto, Matricola_cpt,
      Cognome_cpt, Nome_cpt, Matricola_fmt, Cognome_fmt, Nome_fmt
FROM Nave n,
      (SELECT cpt.Nave, cpt.Matricola AS Matricola_cpt, cpt.Cognome AS Cognome_cpt,
cpt.Nome AS Nome_cpt, fmt.Matricola AS Matricola_fmt, fmt.Cognome AS Cognome_fmt,
fmt.Nome AS Nome_fmt
FROM (SELECT * FROM Equipaggio JOIN Personale ON Membro = Matricola) AS cpt,
      (SELECT * FROM Equipaggio JOIN Personale ON Membro = Matricola) AS fmt
WHERE cpt.Grado = "Comandante"
      AND fmt.Grado = "Primo di coperta"
      AND cpt.Data_sbarco IS NULL
      AND fmt.Data_sbarco IS NULL
      AND cpt.Nave = (SELECT IMO_number FROM Nave WHERE Stato_corrente LIKE "%port
operations%")
      AND fmt.Nave = cpt.Nave) AS x,
      Viaggio v
WHERE n.IMO_number = x.Nave
      AND v.Numero = (SELECT Numero
FROM Viaggi
WHERE Nave = (SELECT IMO_number FROM Nave WHERE Stato_corrente LIKE
"%port operations%")
ORDER BY Numero DESC LIMIT 1);
```

```
mysql> select * from OperazioniPortualiCorrenti;
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| IMO_number | Tipo_carico | Porto          | Matricola_cpt | Cognome_cpt | Nome_cpt | Matricola_fmt | Cognome_fmt | Nome_fmt |
+-----+-----+-----+-----+-----+-----+-----+-----+
| IM05641234 | Grain       | Port of Melbourne | PA68107       | Rizzo       | Silvio  | SR35417       | Vella       | Giovanni |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0,00 sec)
```


3) La procedura numeroEquipaggio prende in ingresso la data di arrivo in un porto scalo e la nave che lo ha effettuato e restituisce una tabella con:

- l'identificativo della nave;
- numero del viaggio della nave;
- operazione compiuta nello scalo;
- il nome del porto;
- il numero di membri dell'equipaggio al momento a bordo;
- l'equipaggio massimo che può trasportare la nave.

```
DROP PROCEDURE IF EXISTS numeroEquipaggio;
DELIMITER //
CREATE PROCEDURE numeroEquipaggio(data DATE, nav CHAR(10))
BEGIN
SELECT Nave, Numero_viaggio, Operazione, Porto, Numero_imbarcati, Equipaggio_max
FROM (SELECT Equipaggio_max, IMO_number
      FROM Classe JOIN Nave ON Classe.Nome = Nave.Classe) AS eqMax
JOIN Scalo ON eqMax.IMO_number = Nave
JOIN (SELECT COUNT(*) AS Numero_imbarcati
      FROM Equipaggio JOIN (SELECT Nave AS NaveS, CAST(Data_arrivo AS DATE) AS
Data_arrivo
      FROM Scalo
WHERE CAST(Data_arrivo AS DATE) = data AND Nave = nav) AS scl
      WHERE Nave = NaveS
      AND Data_imbarco <= Data_arrivo
      AND Data_sbarco >= Data_arrivo
      OR Nave = NaveS
      AND Data_imbarco <= Data_arrivo
      AND Data_sbarco IS NULL) AS nImb
WHERE CAST(Data_arrivo AS DATE) = data
AND Nave = nav;
END //
DELIMITER ;
```

```
mysql>
mysql> call numeroEquipaggio('2016-02-24','IM05641147');
```

Nave	Numero_viaggio	Operazione	Porto	Numero_imbarcati	Equipaggio_max
IM05641147	39	Manutenzione	Port of Palermo	13	28

1 row in set (0,00 sec)

/* Il Comandante presente a bordo durante l'ultimo scalo con Manutenzione */

```
CREATE OR REPLACE VIEW ComUltimaManutenzione AS
SELECT
P.Nave,Matricola,Nome,Cognome,Grado,Data_imbarco,Data_sbarco,Viaggio,Data_arrivo,Porto
FROM (SELECT Nave,Matricola,Nome,Cognome,Grado,Data_imbarco,Data_sbarco
      FROM Equipaggio JOIN Personale ON Membro = Matricola) AS P
JOIN (SELECT Nave,Numero_viaggio AS Viaggio,CAST(Data_arrivo AS DATE)AS
Data_arrivo,Porto
      FROM Scalo
      WHERE Operazione LIKE "%Manutenzione%"
      ORDER BY Data_arrivo DESC LIMIT 1) AS Sc
WHERE P.Nave = Sc.Nave
      AND Data_imbarco < Data_arrivo
      AND Data_sbarco > Data_arrivo
      AND Grado = "Comandante"
      OR P.Nave = Sc.Nave
      AND Data_imbarco < Data_arrivo
      AND Data_sbarco IS NULL
      AND Grado = "Comandante";
```

Procedures

1) /* Procedura per trovare tutte le facilities di un determinato porto */

```
DROP PROCEDURE IF EXISTS 'portfacilities';
```

```
DELIMITER //
```

```
CREATE PROCEDURE portfacilities (nomeporto VARCHAR(50))  
BEGIN  
  SELECT p.Nome, f.Nome, f.Livello  
  FROM Porto p, Facility f  
  WHERE p.Nome = nomeporto AND f.Livello <= p.Livello_facilities  
  ORDER BY f.Livello;  
END//
```

```
DELIMITER ;
```

2) /* Procedura per trovare il membro del personale attualmente imbarcato da più tempo */

```
DROP PROCEDURE IF EXISTS membroDaSbarcare;
DELIMITER //
CREATE PROCEDURE membroDaSbarcare(qual VARCHAR(20), gr VARCHAR(30))
BEGIN
    SELECT Matricola, Nome, Cognome
    FROM (SELECT * FROM Personale JOIN Equipaggio ON Matricola = Membro) AS x
    WHERE x.Matricola NOT IN (SELECT Membro FROM Equipaggio WHERE Data_sbarco IS
NULL)
    AND Qualifica = qual
    AND Grado = gr
    AND Data_sbarco = (SELECT MIN(Data_sbarco)
    FROM (SELECT * FROM Personale JOIN Equipaggio ON Matricola = Membro) AS x
    WHERE x.Matricola NOT IN (SELECT Membro FROM Equipaggio WHERE Data_sbarco IS
NULL)
    AND Qualifica = qual
    AND Grado = gr);
END//
DELIMITER ;
```

3) /* Procedura per trovare l'equipaggio in una data data di una data nave */

```
DROP PROCEDURE IF EXISTS trovaEquipaggio;
DELIMITER //
CREATE PROCEDURE trovaEquipaggio(data DATE, nav CHAR(10))
BEGIN
SELECT *
FROM Equipaggio JOIN Personale ON Membro = Matricola
WHERE Nave = nav
    AND Data_imbarco <= data
    AND Data_sbarco >= data
    OR Nave = nav
    AND Data_imbarco <= data
    AND Data_sbarco IS NULL;
END//
DELIMITER ;
```

Triggers

1) /*Trigger: AFTER; prima dell'iserimento di un nuovo viaggio controlla che i porti di partenza e destinazione possano ospitare la nave*/

```
DROP TRIGGER IF EXISTS comp_viaggi;
DELIMITER //
CREATE TRIGGER comp_viaggi AFTER INSERT ON Viaggi
FOR EACH ROW
BEGIN
    DECLARE comp VARCHAR(13);
    DECLARE valid INT(1);
    SELECT compatibilita(NEW.Nave,NEW.Porto_partenza) INTO comp;
    IF comp = "compatibile" THEN
        SELECT compatibilita(NEW.Nave,NEW.Porto_destinazione) INTO comp;
        IF comp = "compatibile" THEN SET valid = 1;
        ELSE SET valid = 0;
        END IF;
    ELSE SET valid = 0;
    END IF;
    IF valid = 0 THEN
        SIGNAL SQLSTATE VALUE '45000'
        SET MESSAGE_TEXT = "INCOMPATIBILITA' PORTI Partenza/Destinazione con la classe";
    END IF;
END//
DELIMITER ;
```

2) /*Trigger: BEFORE su tabella Viaggi; Controlla e in caso corregge che il numero di viaggio inserito sia in successione con quelli relativi alla nave*/

```
DROP TRIGGER IF EXISTS num_viaggio;
DELIMITER //
CREATE TRIGGER num_viaggio BEFORE INSERT ON Viaggi
FOR EACH ROW
BEGIN
    DECLARE num INT(10);
    SELECT MAX(Numero) INTO num FROM Viaggio WHERE Nave = NEW.Nave AND Numero != NEW.Numero;
    IF NEW.Numero > num+1 THEN
        SET NEW.Numero = num+1;
        SIGNAL SQLSTATE '01000'
        SET MESSAGE_TEXT = "ATTENZIONE: Numero viaggio e' stato corretto", MYSQL_ERRNO =
        '1000';
    END IF;
END//
DELIMITER ;
```

3) /*Controllo di pertinenza tra il grado e la qualifica dei nuovi aggiunti alla tabella Personale*/

```
DROP TRIGGER IF EXISTS controlloQualificagrado;
DELIMITER //
CREATE TRIGGER controlloQualificagrado AFTER INSERT ON Personale
FOR EACH ROW
BEGIN

DECLARE msg VARCHAR(128);
SET msg = 'Inserito valore per Grado non coerente con Qualifica.';

IF NEW.Qualifica = 'Cuoco' and NEW.Grado != 'Cuoco' or NEW.Qualifica = 'Cuoco' and
NEW.Grado != 'Aiuto Cuoco'
THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = msg;

ELSEIF NEW.Qualifica = 'Marinaio' and NEW.Grado != 'Mozzo' or NEW.Qualifica = 'Marinaio'
and NEW.Grado != 'Marinaio Scelto'
THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = msg;

ELSEIF NEW.Qualifica = 'Tecnico' and NEW.Grado != 'Operaio' or NEW.Qualifica = 'Tecnico'
and NEW.Grado != 'Elettricista'
THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = msg;

ELSEIF NEW.Qualifica = 'Ufficiale' and NEW.Grado != 'Secondo di Macchina'
or NEW.Qualifica = 'Ufficiale' and NEW.Grado != 'Secondo di coperta'
or NEW.Qualifica = 'Ufficiale' and NEW.Grado != 'Comandante'
or NEW.Qualifica = 'Ufficiale' and NEW.Grado != 'Capo Macchinista'
or NEW.Qualifica = 'Ufficiale' and NEW.Grado != 'Primo di coperta'
or NEW.Qualifica = 'Ufficiale' and NEW.Grado != 'Primo di Macchina'
or NEW.Qualifica = 'Ufficiale' and NEW.Grado != 'Allievo di Macchina'
or NEW.Qualifica = 'Ufficiale' and NEW.Grado != 'Allievo di Coperta'
THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = msg;

END IF;

END//
DELIMITER ;
```


Functions

1) /* Funzione per trovare la nave più vecchia che ha effettuato il più alto numero di viaggi in un determinato periodo di tempo */

```
DROP FUNCTION IF EXISTS maxViaggi;
DELIMITER //
CREATE FUNCTION      maxViaggi(inizio DATE, fine DATE)
RETURNS CHAR(10)
BEGIN
DECLARE num INT;
DECLARE nav CHAR(10);
SELECT Nave INTO nav
FROM (SELECT Nave, count(*) cont FROM Viaggio WHERE Inizio_viaggio >= inizio AND
Fine_viaggio <= fine GROUP BY Nave) AS cv
JOIN Nave ON (cv.Nave = IMO_number)
WHERE cont = (SELECT max(cont)
FROM (SELECT Nave, count(*) cont
      FROM Viaggio WHERE Inizio_viaggio >= inizio AND Fine_viaggio <= fine
      GROUP BY Nave) AS cv)
ORDER BY Data_costruzione
LIMIT 1;
RETURN nav;
END//
DELIMITER ;
```

2) /* Funzione che ritorna una stringa "compatibile" o "incompatibile" a seconda se una nave può o meno entrare in un certo porto */

```
DELIMITER //
CREATE FUNCTION compatibilita(nave CHAR(10),port VARCHAR(50))
RETURNS VARCHAR(13)
BEGIN
DECLARE comp VARCHAR(13);
DECLARE class VARCHAR(50);
DECLARE lvlFaci INT(1);
DECLARE lvlPort INT(1);
SELECT Classe INTO class FROM Nave WHERE IMO_number = nave;
SELECT Livello_facilities INTO lvlFaci FROM Classe WHERE Nome = class;
SELECT Livello_facilities INTO lvlPort FROM Porto WHERE Nome = port;
IF lvlFaci <= lvlPort THEN SET comp = "compatibile";
ELSE SET comp = "incompatibile";
END IF;
RETURN comp;
END//
DELIMITER ;
```

3) /* Funzione che ritorna il numero di giorni di imbarco di un membro del personale*/

```
DROP FUNCTION IF EXISTS anzianitaMembro;
DELIMITER //
CREATE FUNCTION anzianitaMembro(matricola CHAR(7))
RETURNS INT

BEGIN
DECLARE sommaDate INT;
DECLARE differenzaDate INT;
SET differenzaDate = 0;
    SELECT SUM(DATEDIFF(Data_sbarco,Data_imbarco)) INTO sommaDate
    FROM Equipaggio
    WHERE Membro = matricola
        AND Data_sbarco IS NOT NULL;

    SELECT DATEDIFF(CURDATE(),Data_imbarco) INTO differenzaDate FROM Equipaggio WHERE
Membro = matricola AND Data_sbarco IS NULL;
    SET sommaDate = sommaDate + differenzaDate;

RETURN sommaDate;
END//
DELIMITER ;
```