

TRAVEL TO THE MOON

DATI DI INTERESSE:

1. CROCIERE

1.1 CODICE (UNIVOCO)

1.2 DATA DI INIZIO

1.3 DATA DI FINE

1.4 NAVE UTILIZZATA (REQ. 2)

1.5 ITINERARIO (REQ. 3)

1.6 TIPOLOGIA (UNA DI MIELE O PER FAMIGLIE)

1.6.1 CROCIERE LUNA DI MIELE:

1.6.1.1 SOTTO-TIPOLOGIA (TRADIZIONALE (DEST: ROMANTICHE), DIVERTENTI) O ALTERNATIVE)

1.6.2 CROCIERE PER FAMIGLIE

1.6.2.1 ADATTE AI BAMBINI?

2. NAVI

2.1 NOME (UNIVOCO)

2.2 GRADO DI CONFORT (DA 3 A 5)

2.3 CAPIENZA MAX

3. ITINERARIO

3.1 NOME

3.2 TAPPE

3.2.1 DESTINAZIONE (REQ. 4)

3.2.2 QUANDO SARÀ RAGGIUNTA LA DESTINAZIONE (UNA COPPIA DI GIORNO DI CROCIERA, ORA)

4. DESTINAZIONI

4.1 NOME

4.2 ATTRAZIONI (REQ. 5) 4.3 CONTINENTE 4.4 ROMANTICA O DIVERTENTE

5. ATTRAZIONI

4.5 SE È GETTONATA* 4.6 SE È ESOTICA**

5.1 NOME

5.2 DESCRIZIONE

5.3 FASCIA ORARIA CONSIGLIATA

6. PRENOTAZIONI

6.1 CLIENTE (REQ. 7)

6.2 ISTANTE DI PRENOTAZIONE

6.3 CROCIERA (REQ. 1)

6.4 POSTI PRENOTATI

7. CLIENTI

7.1 CF (PER IDENTIFICAZIONE)

7.2 NOME

7.3 COGNOME

7.4 ETÀ

7.5 INDIRIZZO

* GETTONATA SIGNIFICA RAGGIUNTA DA ALMENO 10 CROCIERE ROMANTICHE O 15 ALTERNATIVE

** ESOTICA SIGNIFICA CHE SI TROVA IN UN CONTINENTE DIVERSO DALL'EUROPA

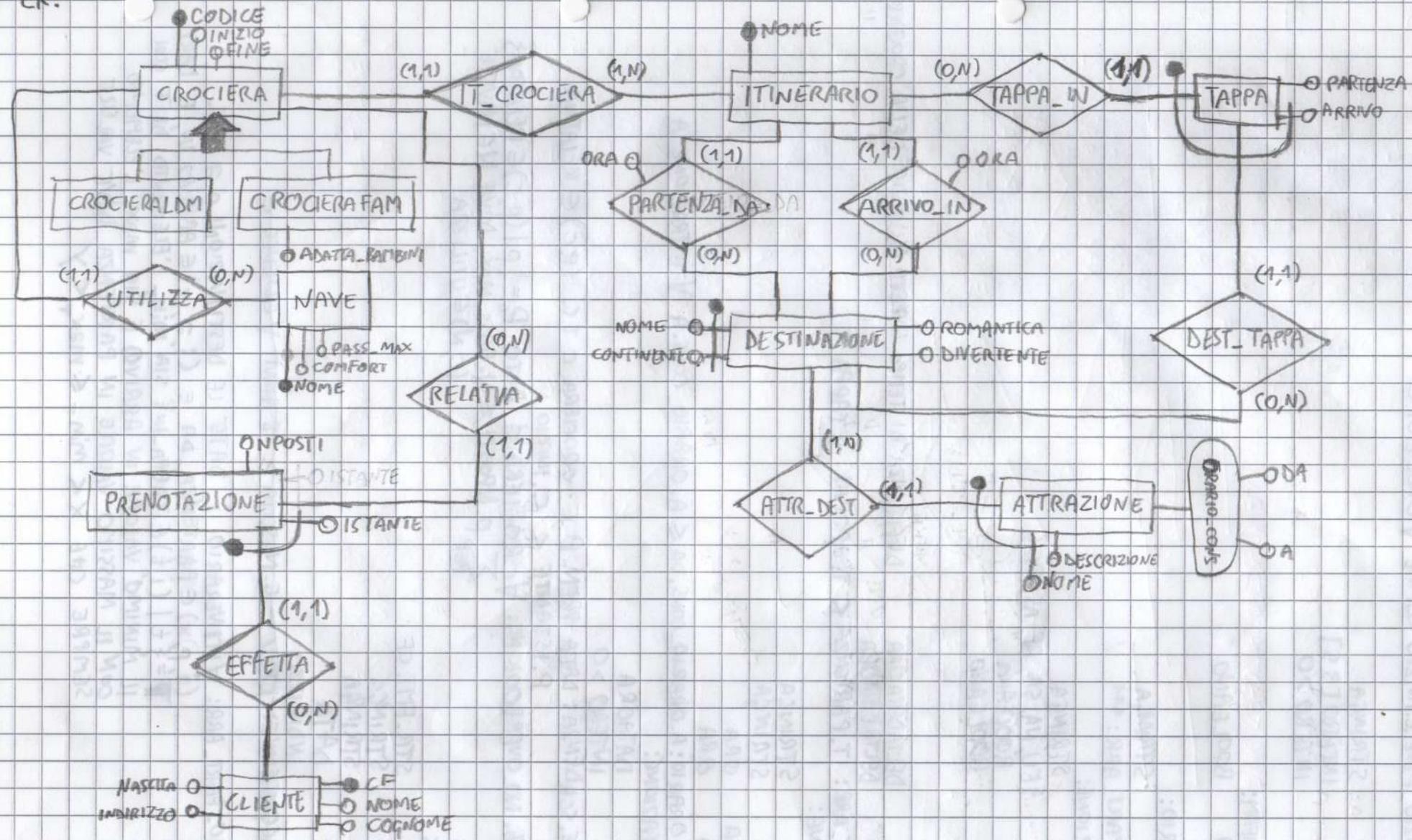
8. FUNZIONALITÀ RICHIESTE:

8.1 DATO UN CLIENTE CHE VUOLE PRENOTARE UN CERTO NUMERO DI POSTI PER UNA CROCIERA C, IL PERSONALE DELL'UFFICIO PRENOTAZIONI DEVE POTER EFFETTUARE LA RELATIVA PRENOTAZIONE, CHE DEVE ESSERE RIFIUTATA NEL CASO IN CUI I POSTI DISPONIBILI SIANO INSUFFICIENTI IN QUESTO ISTANTE

8.2 DATO UN INSIEME DI CLIENTI, L'UFFICIO MARKETING DEVE POTER CALCOLARE L'ETÀ MEDIA DI QUELLI CHE HANNO PRENOTATO ALMENO UNA CROCIERA CHE PREVEDE UNA DESTINAZIONE ESOTICA^{**}

8.3 DATO UN INSIEME DI DESTINAZIONI, L'UFFICIO MARKETING DEVE POTER CALCOLARE LA PERCENTUALE DI QUELLE GETTONATE*

ER:



DIZIONARIO DEI DATI:

ATRIBUTO

TIPO

NOTE

ENTITÀ CROCIERA:

CODICE	STRINGA
INIZIO	DATAORA
FINE	DATAORA

V. CROCIERA. INIZIO <= FINE \wedge CROCIERA C

ENTITÀ NAVE:

NOME	STRINGA
COMFORT	INTERO [3,5]
PASS_MAX	INTERO > 0

ENTITÀ:

ENTITÀ CROCIERAFAM:

ADATTA_BAMBINI	BOOLEANO
----------------	----------

ENTITÀ ITINERARIO:

NOME	STRINGA
V. ITINERARIO. PART_ARR: ***	V

ENTITÀ DESTINAZIONE:

NOME	STRINGA
CONTINENTE	{EU, NA, SA, AF, AS, OC}
ROMANTICA	BOOLEANO
DIVERTENTE	BOOLEANO

ENTITÀ TAPPA:

PARTENZA	DELTA DATAORA	DIFFERENZA DI TEMPO RISPETTO ALL'INIZIO DELLA CROCIERA
ARRIVO	DELTA DATAORA	" " "
V. TAPPA. PART_ARR: T. PARTENZA < T. ARRIVO	V. TAPPA T	" " "

ENTITÀ ATTRAZIONE:

NOME	STRINGA
DESCRIZIONE	STRINGA
ORARIO_CONS_DA	ORA
" , A	ORA

V. ATTRAZIONE_ORARIO: A. ORARIO_CONS_DA <= A. ORARIO_CONS_A \wedge ATTRAZIONE A

ENTITÀ PRENOTAZIONE:

ISTANTE	DATAORA
NPOSTI	INTERO > 0

V. PRENOTAZIONE_SCADENZA: DATA PREN. P E CROCIERA C T.C. (P,C) E RELATIVA
P. ISTANTE < C. INIZIO

V. PRENOTAZIONE_NO_OVERBOOKING: \forall CROCIERA C DATO $P_C = \{p \mid (p, c) \text{ E RELATIVA}\}$
 $\sum_{p \in P_C} p.NPOSTI \leq N.PASS.MAX.$, DOVE NE NAVE E
 (C, N) E UTILIZZA

ENTITÀ CLIENTE:

CF	STR.FMT.CF
NOME	STRINGA
COGNOME	STRINGA
NASCITA	DATA
INDIRIZZO	INDIRIZZO

VINCOLI ESTERNI

V. CROCIERA_INIZIO_FINE: $\forall i, f, c \ [\text{CROCIERA}(c) \wedge (\text{INIZIO}(c, i) \wedge \text{FINE}(c, f))] \rightarrow i < f$

V. ITINERARIO_PART_ARR: $\forall i, p, x, y, \min_T, \max_T \ [\text{ITINERARIO}(i) \wedge \text{PARTENZA-DA}(i, p) \wedge \text{ARRIVO-IN}(i, z) \wedge \text{ORA}(i, p, x) \wedge \text{ORA}(i, z, y) \wedge \min_T = \min(\{t \mid \exists T \text{ TAPPA-IN}(i, t)\} \wedge \text{PARTENZA}(i, T, t)) \wedge \max_T = \max(\{t \mid \exists T \text{ TAPPA-IN}(i, t)\} \wedge \text{ARRIVO}(i, T, t))] \rightarrow x < \min_T \leq \max_T < y$

V. TAPPA_PART_ARR: $\forall t, p, z \ [\text{TAPPA}(t) \wedge \text{PARTENZA}(t, p) \wedge \text{ARRIVO}(t, z)] \rightarrow z < p \leq z$

V. ATTRAZIONE_ORARIO: $\forall z, c, c_{DA}, c_A \ [\text{ATTRAZIONE}(z) \wedge \text{ORARIO-CONS}(z, c) \wedge \text{DA}(c, c_{DA}) \wedge A(c, c_{DA})] \rightarrow c_{DA} \leq c_A$

V. PRENOTAZIONE_NO_OVERBOOKING: $\forall P, c, m, n \ [P = \sum_{k \in \exists P(\text{POSTI}(P, k) \wedge \text{RELATIVA}(P, c))} k \wedge \text{CROCIERA}(c) \wedge \text{UTILIZZA}(c, n) \wedge \text{MAX-POSTI}(n, m)] \rightarrow P \leq m$

V. CLIENTE_MAGGIORENNE: $\forall c, m \ [\text{CLIENTE}(c) \wedge \text{NASCITA}(n)] \rightarrow \text{OGGI} - n \geq 18 \text{ ANNI}$

TIPI DI DATI PERSONALIZZATI:

STRINGA FORMATO CF: UNA STRINGA DI 16 CARATTERI CHE RAPPRESENTI
UN CF VALIDO

INDIRIZZO: RECORD{

VIA_PZA: STRINGA
CIVICO: INTERO > 0, (0,1)
CITTÀ: STRINGA
CAP: STRINGA NUMERICA DIM. 5

}

DIZIONARIO DEI DATI (CONT.):

ATTRIBUTO	TIPO	NOTE
-----------	------	------

RELAZIONE PARTENZA_DA:

ORA	DELTADATAORA	DIFF. DI TEMPO RISPETTO ALL'INIZIO DELLA CROCIERA
-----	--------------	---

RELAZIONE ARRIVO_IN:

ORA	DELTADATAORA						
-----	--------------	--	--	--	--	--	--

UML USE-CASE:

CLIENTI

GESTIONE CLIENTI

GESTIONE PRENOTAZIONI

UFF. PREN.

OPERAZIONI CROCIERE

UFF. MARKETING

STATISTICHE CLIENTI

STATISTICHE DESTINAZIONI

SPECIFICHE USE-CASE GESTIONE CLIENTI

REGISTRA_CLIENTE (CF: CF, NOME: STRINGA, COGNOME: STRINGA, NASCITA: DATA, INDIRIZZO: INDIRIZZO) : CLIENTE

PRECONDIZIONI: $\exists c \text{ CLIENTE}(c) \wedge CF(c, CF)$

POSTCONDIZIONI:

MODIFICA DEL LIVELLO ESTENSIONALE DEI DATI:

NUOVI ELEMENTI DEL DOMINIO DI INTERPRETAZIONE: c

ELEMENTI DEL DOMINIO DI INTERPRETAZIONE CHE NON ESISTONO PIÙ: NESSUNO

NUOVE ENNUPLE DI PREDICATI:

CLIENTE(c)

CF(c, CF)

NOME(c, NOME)

COGNOME(c, COGNOME)

NASCITA(c, NASCITA)

INDIRIZZO(c, INDIRIZZO)

ENNUPLE DI PREDICATI CHE NON ESISTONO PIÙ: NESSUNA

VALORE DI RITORNO: RESULT = c

SPECIFICHE USE-CASE GESTIONE PRENOTAZIONI

PRENOTA(CL: CLIENTE, CR: CROCIERA, NP: INTERO > 0) : PRENOTAZIONE

PRECONDIZIONI: $\exists n, MAX, p \text{ OCC NAVE}(n) \wedge UTILIZZA(CR, n) \wedge PASS_MAX(n, MAX) \wedge$

$$P = \sum_{(p, n)} \text{PRENOTAZIONE}(p) \wedge \text{NPOTI}(p, n) \wedge \text{RELATIVA}(p, CR) \wedge$$

$$\text{OCC} = \sum_{(n)} n \wedge \text{MAX} - \text{OCC} \geq \text{NP}$$

POSTCONDIZIONI:

MODIFICA DEL LIVELLO ESTENSIONALE DEI DATI:

NUOVI ELEMENTI DEL DOMINIO DI INTERPRETAZIONE: p

ELEMENTI DEL DOMINIO DI INTERPRETAZIONE CHE NON ESISTONO PIÙ: NESSUNO

NUOVE ENNUPLE DI PREDICATI:

PRENOTAZIONE(p)

NPOTI(p, NP)

INSTANTE(p, ADESSO)

EFFETTUA(CL, p)

RELATIVA(p, CR)

ENNUPLE DI PREDICATI CHE NON ESISTONO PIÙ: NESSUNA

VALORE DI RITORNO: RESULT = p

ANNULLA_PRENOTAZIONE (p: PRENOTAZIONE)

PRECONDIZIONI: $\exists c, d \text{ CROCIERA}(c) \wedge \text{RELATIVA}(p, c) \wedge \text{INIZIO}(c, d) \wedge d > \text{ADESSO}$

POSTCONDIZIONI:

MODIFICA DEL LIVELLO ESTENSIONALE DEI DATI:

NUOVI ELEMENTI DEL DOMINIO DI INTERPRETAZIONE: NESSUNO

ELEMENTI DEL DOMINIO DI INTERPRETAZIONE CHE NON ESISTONO PIÙ: p

NUOVE ENNUPLE DI PREDICATI:

$\neg \text{PRENOTAZIONE}(p)$

$\neg \text{NPOTI}(p, d)$

ENNUPLE DI PREDICATI CHE NON ESISTONO PIÙ:

PRENOTAZIONE(p)

VALORE DI RITORNO: NESSUNO

SPECIFICHE USE-CASE OPERAZIONI_CROCIERE

LDM_TRADIZIONALE (c: CROCIERALDM) : BOLEANO

PRECONDIZIONI: NESSUNA

POSTCONDIZIONI:

MODIFICA AL LIVELLO ESTENSIONALE DEI DATI: NESSUNA

VALORE DI RITORNO: SIA i T.C. ITINERARIO(i) \wedge IT_CROCIERA(i, c)

$$D_s = \{d \mid \text{DESTINAZIONE}(d) \wedge (\text{PARTENZA_DA}(i, d) \vee \\ \text{ARRIVO_IN}(i, d) \vee (\exists t \text{ TAPPA}(t) \wedge \\ \text{DEST_TAPPA}(d, t) \wedge \text{TAPPA_IN}(i, t)))\}$$

$$D = \{d \in D_s \mid \text{DIVERTENTE}(d, \text{VERO})\}$$

$$R = \{d \in D_s \mid \text{ROMANTICA}(d, \text{VERO})\}$$

$$\text{RESULT} = |R| \geq |D|$$

CROCIERA_ESOTICA (c: CROCIERA) : BOLEANO

PRECONDIZIONI: NESSUNA

POSTCONDIZIONI:

MODIFICA AL LIVELLO ESTENSIONALE DEI DATI: NESSUNA

VALORE DI RITORNO: SIA i T.C. ITINERARIO(i) \wedge IT_CROCIERA(i, c)

$$D_s = \{d \mid \text{DESTINAZIONE}(d) \wedge (\text{PARTENZA_DA}(i, d) \vee \text{ARRIVO_IN}(i, d) \vee \\ (\exists t \text{ TAPPA}(t) \wedge \text{DEST_TAPPA}(d, t) \wedge \\ \text{TAPPA_IN}(i, t)))\}$$

$$\text{RESULT} = |D_s| > 0$$

SPECIFICHE USE-CASE STATISTICHE_CLIENTI

ETA_MEDIA_ESOTICA (C: CLIENTE(1, N)) : REALE ≥ 18

PRECONDIZIONI: $\exists c, p, cr \in C \wedge \text{PRENOTAZIONE}(p) \wedge \text{EFFETTUA}(c, p) \wedge \\ \text{RELATIVA}(p, cr) \wedge \\ \text{"OPERAZIONI_CROCIERE".CROCIERA_ESOTICA}(cr) = \text{VERO}$

POSTCONDIZIONI:

MODIFICA AL LIVELLO ESTENSIONALE DEI DATI: NESSUNA

VALORE DI RITORNO: SIA $C_E = \{(c, cr, e) \mid c \in C \wedge \text{ANASCIA}(c, e) \wedge \text{CROCIERA}(cr) \wedge \\ (\exists p \text{ PRENOTAZIONE}(p) \wedge \text{EFFETTUA}(c, p) \wedge \\ \text{RELATIVA}(p, cr)) \wedge \\ \text{"OPERAZIONI_CROCIERE".CROCIERA_ESOTICA}(cr) = \text{VERO}\}$

$$\text{RESULT} = \sum (c, cr, e)$$

$$\text{RESULT} = \underline{\underline{(c, cr, e) \in C_E}}$$

$$|C_E|$$

SPECIFICHE USE-CASE STATISTICHE - DESTINAZIONI

DEST_GETTONATA(d: DESTINAZIONE): BOOLEANO

PRECONDIZIONI: NESSUNA

POSTCONDIZIONI:

MODIFICA AL LIVELLO ESTENSIONALE DEI DATI: NESSUNA

VALORE DI RITORNO: SIA $C = \{(c, i) | \text{CROCIERA}(c) \wedge \text{IT_CROCIERA}(c, i) \wedge (\text{PARTENZA_DA}(i, d) \vee \text{ARRIVO_IN}(i, d) \vee (\exists t \text{ TAPPA}(t) \wedge \text{TAPPA_IN}(i, t) \wedge \text{DEST_TAPPA}(d, t)))\}$

SIA $T = \{(c, i) \in C | \text{"OPERAZIONI_CROCIERE".LDM_TRADIZIONALE}(c) = \text{VERO}\}$

SIA $A = C \setminus T$.

RESULT = $\begin{cases} \text{VERO SE } |\text{IT}| \geq 10 \vee |\text{AI}| \geq 15 \\ \text{FALSO ALTRIMENTI} \end{cases}$

PERC_GETTONATE(D: DESTINAZIONE(1, N)): REALE [0, 100]

PRECONDIZIONI: NESSUNA

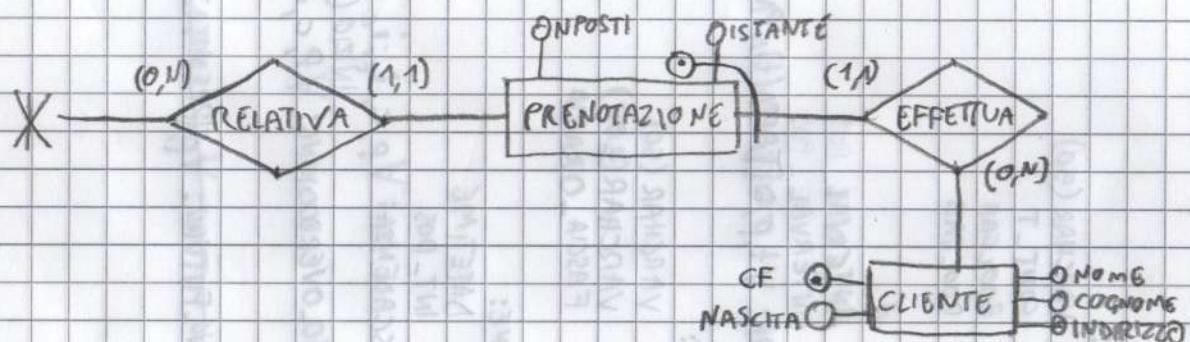
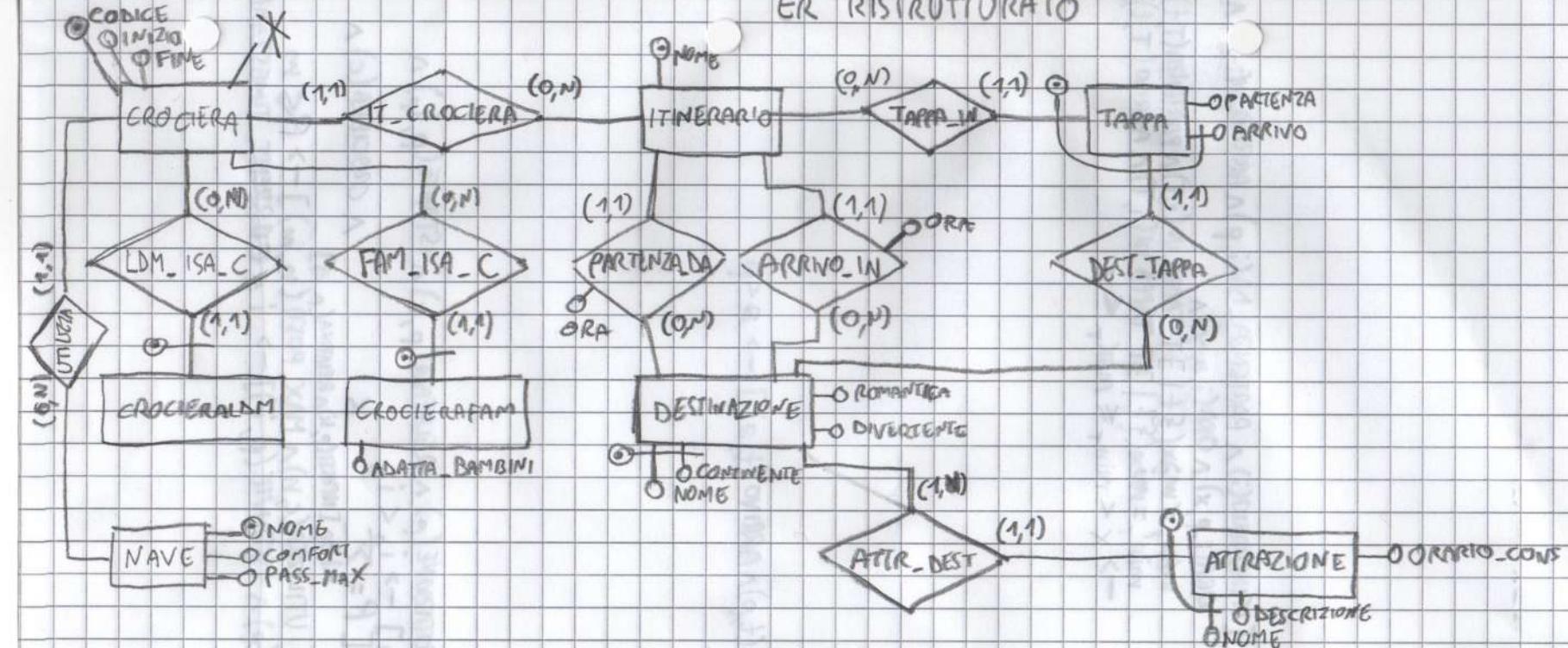
POSTCONDIZIONI:

MODIFICA AL LIVELLO ESTENSIONALE DEI DATI: NESSUNA

VALORE DI RITORNO: SIA $D_g = \{d \in D | \text{"STATISTICHE_DESTINAZIONI".DEST_GETTONATA}(d) = \text{VERO}\}$

RESULT = $\frac{|D_g|}{|D|} \cdot 100$

ER RISTRUTTURATO



DIZIONARIO DEI DATI (DBMS: POSTGRESQL):

ENTITÀ CROCIERA:

CODICE	VARCHAR(20)
INIZIO	DATETIME
FINE	DATETIME

V. CROCIERA. INIZIO_FINE: $\forall c, i, f \text{ CROCIERA}(c) \wedge \text{INIZIO}(c, i) \wedge \text{FINE}(c, f) \rightarrow i < f$

V. CROCIERA. ISA: $\forall c \text{ CROCIERA}(c) \rightarrow \exists c' \text{ LDM_ISA_C}(c, c') \vee \text{FAM_ISA_C}(c, c')$

V. CROCIERA. DISG: $\forall c \text{ CROCIERA}(c) \rightarrow \exists l \text{ LDM_ISA_C}(c, l) \rightarrow \neg \exists f \text{ FAM_ISA_C}(c, f)$

ENTITÀ NAVE:

NOME	VARCHAR(30)
COMFORT	COMFORT_T
PASS_MAX	INT_POS

ENTITÀ CROCIERAFAM:

ADATTA_BAMBINI: BOOLEAN

ENTITÀ ITINERARIO:

NOME	VARCHAR(50)
VALORE	REAL

V. ITINERARIO.PART_ARR: $\forall i, p, z, x, y, \min_t, \max_t [\text{ITINERARIO}(i) \wedge \text{PARTENZA_DA}(i, p) \wedge \text{ARRIVO_IN}(i, z) \wedge \text{ORA}(i, p, x) \wedge \text{ORA}(i, z, y) \wedge \min_t \neq \max_t \exists t \text{ TAPPA_IN}(i, t) \wedge \text{PARTENZA}(t, t)] \wedge \min_t = \min(\{t | \exists t \text{ TAPPA_IN}(i, t) \wedge \text{ARRIVO}(t, t)\}) \wedge \max_t = \max(\{t | \exists t \text{ TAPPA_IN}(i, t) \wedge \text{ARRIVO}(t, t)\}) \rightarrow x < \min_t \leq \max_t < y$

ENTITÀ DESTINAZIONE:

NOME	VARCHAR(50)
CONTINENTE	CONT_T
ROMANTICA	BOOLEAN
DIVERTENTE	BOOLEAN

ENTITÀ TAPPA:

PARTENZA	INTERVAL_POS
ARRIVO	INTERVAL_POS

V. TAPPA.PART_ARR: $\forall t, p, z [\text{TAPPA}(t) \wedge \text{PARTENZA}(t, p) \wedge \text{ARRIVO}(t, z)] \rightarrow z < p$

ENTITÀ ATTRAZIONE:

NOME	VARCHAR(50)
DESCRIZIONE	VARCHAR(300)
ORARIO_CONS	FASCIA_ORARIA

ENTITÀ PRENOTAZIONE:

ISTANTE	DATETIME
NPOSTI	INT_POS

V. PRENOTAZIONE_SCADENZA: $\forall p, c, i_p, i_c [\text{PRENOTAZIONE}(p) \wedge \text{RELATIVA}(p, c) \wedge \text{ISTANTE}(p, i_p) \wedge \text{INIZIO}(c, i_c)] \rightarrow i_p < i_c$

V. PRENOTAZIONE_NO_OVERBOOKING: $\forall p, c, m, n [p = \sum_{k=1}^n k \wedge \text{CROCIERA}(c) \wedge \text{UTILIZZA}(c, n) \wedge \text{MAX_POSTI}(n, m)] \rightarrow p \leq m$

ENTITÀ CLIENTE:

CF	CF-T
NOME	VARCHAR(30)
COGNOME	VARCHAR(30)
NASCITA	DATE
INDIRIZZO	INDR-T

V. CLIENTE, MAGGIORENNE: $\forall c, n [CLIENTE(c) \wedge NASCITA(n)] \rightarrow n + INTERVAL[18\text{ years}] \leq CURRENT_DATE$

RELATIONSHIP PARTENZA_DA:

ORA	INTERVAL
-----	----------

RELATIONSHIP ARRIVO_IN:

ORA	INTERVAL
-----	----------

TIPI PERSONALIZZATI:

COMFORT-T: INTEGER CHECK VALUE BETWEEN 3 AND 5

INT_POS: INTEGER CHECK VALUE > 0

CONT-T: ENUM ('EU', 'NA', 'SA', 'AF', 'AS', 'OC')

FASCIA_ORARIA: TYPE (DA TIME NOT NULL A TIME NOT NULL, CHECK DA ≤ A)

CF-T: CHAR(16) CHECK VALUE ^"["[a-z][a-z][a-z][a-z][a-z][a-z][a-z][a-z][a-z][a-z][a-z][a-z][a-z][a-z][a-z][a-z]"^"

INDR-T: TYPE (

VIA, PZA VARCHAR(50) NOT NULL,

CIVICO INT_POS

L_CIVICO CHAR(1),

CITTA VARCHAR(20) NOT NULL

)

SCHEMA RELAZIONALE

CROCIERA (CODICE: VARCHAR(20), INIZIO: DATETIME, FINE: DATETIME, ITINERARIO: VARCHAR(50),
NAVE: VARCHAR(30))
VINCOLO FOREIGN KEY: ITINERARIO REFERENCES ITINERARIO(NOME)
" " " : NAVE REFERENCES NAVE(NOME)
" " ENNUPLA: INIZIO < FINE

CROCIERA_LDM (CODICE: VARCHAR(20))
VINCOLO FOREIGN KEY: CODICE REFERENCES CROCIERA(CODICE)

CROCIERA_FAM (CODICE: VARCHAR(20), ADULTI_BAMBINI: BOOLEAN)
VINCOLO FOREIGN KEY: CODICE REFERENCES CROCIERA(CODICE)

NAVE (NOME: VARCHAR(30), COMFORT: COMFORT_T, PASS_MAX: INT_POS)

ITINERARIO (NOME: VARCHAR(50))
VINCOLO FOREIGN KEY: NOME REFERENCES PARTENZA_DA(IT)
" " " " : IT REFERENCES PARTENZA_DA(IT)
" " " " : ARRIVO_IN(IT)

PARTENZA_DA (IT: VARCHAR(50), ORA: INTERVAL, DST_C: CONT_T, DST_N: VARCHAR(50))
VINCOLO FOREIGN KEY: IT REFERENCES ITINERARIO(NOME)
" " " : DST_C REFERENCES DESTINAZIONE(CONTINENTE)
" " " : DST_N REFERENCES DESTINAZIONE(NOME)

ARRIVO_IN (IT: VARCHAR(50), ORA: INTERVAL, DST_C: CONT_T, DST_N: VARCHAR(50))
VINCOLO FOREIGN KEY: IT REFERENCES ITINERARIO(NOME)
" " " : DST_C REFERENCES DESTINAZIONE(CONTINENTE)
" " " : DST_N REFERENCES DESTINAZIONE(NOME)

DESTINAZIONE (CONTINENTE: CONT_T, NOME: VARCHAR(50), ROMANTICA: BOOLEAN, DIVERTENTE: BOOLEAN)
VINCOLO INCLUSIONE: (CONTINENTE, NOME) ⊆ ATTRAZIONE(DST_C, DST_N)

ATTRAZIONE (DST_C: CONT_T, DST_N: VARCHAR(50), NOME: VARCHAR(50), DESCRIZIONE: VARCHAR(300),
ORARIO_CONS: FASCIA_ORARIA)
VINCOLO FOREIGN KEY: DST_C REFERENCES DESTINAZIONE(CONTINENTE)
" " " : DST_N REFERENCES DESTINAZIONE(NOME)

TAPPA (IT: VARCHAR(50), DST_C: CONT_T, DST_N: VARCHAR(50), ARRIVO: INTERVAL,
PARTENZA: INTERVAL)
VINCOLO FOREIGN KEY: IT REFERENCES ITINERARIO(NOME)
" " " : DST_C REFERENCES DESTINAZIONE(CONTINENTE)
" " " : DST_N REFERENCES DESTINAZIONE(NOME)
" " ENNUPLA: ARRIVO < PARTENZA

PRENOTAZIONE(CLIENTE: CF_T, ISTANTE: DATETIME, NPOSTI: INT_POS, CROCIERA: VARCHAR(20))
VINCOLO FOREIGN KEY: CLIENTE REFERENCES CLIENTE(CF)
" " " : CROCIERA REFERENCES CROCIERA(CODICE)
" " ENNUPLA: Istante ≤ CURRENT_TIMESTAMP

CLIENTE(CF: CF_T, NOME: VARCHAR(30), COGNOME: VARCHAR(10), NASCITA: DATE,
INDIRIZZO: INDIR_T)
VINCOLO ENNUPLA: NASCITA + INTERVAL '18 years' ≤ CURRENT_DATE

CROCIERA(CODICE: VARCHAR(20), INIZIO: DATE, FINE: DATE, ITINERARIO: VARCHAR(50))
VINCOLO FOREIGN KEY: ITINERARIO REFERENCES ITINERARIO(NOME)

PROGETTAZIONE DEI VINCOLI ESTERNI: V.FOREIGN KEY: NAVE REFERENCES NAVE(NOME)

TRIGGER PER V.CROCIERA.ISA: VINCOLO FOREIGN KEY: NAVE REFERENCES CROCIERA(NOME)

- OPERAZIONI:

CROCIERA - INSERIMENTO O MODIFICA IN CROCIERA ISM: BOOLEAN)

VIA - MODIFICA O CANCELLAZIONE IN CROCIERALDM (CODICE)

- " " " IN CROCIERAFAM

NAVE - ISTANTE DI INVOCAZIONE: PRIMA DELL'OPERAZIONE

- FUNZIONE:

SIA ERROR = FALSE)

SE L'OPERAZIONE INSERISCE O MODIFICA IN CROCIERA:

ERROR := NOT EXISTS((SELECT CODICE FROM CROCIERALDM WHERE CODICE = NEW.CODICE)
UNION (SELECT CODICE FROM CROCIERAFAM WHERE CODICE = NEW.CODICE));

SE L'OPERAZIONE MODIFICA O CANCELLA IN CROCIERALDM O CROCIERAFAM:

ERROR := EXIST (SELECT 1 FROM CROCIERA WHERE CODICE = OLD.CODICE);

SG L'OPERAZIONE MODIFICA IN CROCIERALDM O CROCIERAFAM;

AND (OLD.CODICE <> NEW.CODICE);

SE ERROR = TRUE BLOCCA L'OPERAZIONE

ALTRIMENTI CONSENTILA

TRIGGER PER V.CROCIERA.DISG:

- OPERAZIONI:

- INSERIMENTO O MODIFICA IN CROCIERALDM

- " " " IN CROCIERAFAM

- ISTANTE DI INVOCAZIONE: PRIMA DELL'OPERAZIONE

- FUNZIONE:

SIA ERROR = FALSE

SE L'OPERAZIONE INSERISCE O MODIFICA IN CROCIERALDM:

ERROR := EXIST (SELECT 1 FROM CROCIERAFAM WHERE CODICE = NEW.CODICE);

SG L'OPERAZIONE INSERISCE O MODIFICA IN CROCIERAFAM:

ERROR := EXIST (SELECT 1 FROM CROCIERALDM WHERE CODICE = NEW.CODICE);

SE ERROR = TRUE BLOCCA L'OPERAZIONE

ALTRIMENTI CONSENTILA

TRIGGER PER V. ITINERARIO. PART_ARR:

- OPERAZIONI:

- INSERIMENTO O MODIFICA IN TAPPA - INSERIMENTO O MODIFICA IN ITINERARIO

- II II II IN PARTENZA_DA

- II II II IN ARRIVO_IN

- ISTANTE DI INVOCAZIONE: PRIMA DELL'OPERAZIONE

- FUNZIONE:

SIA ERROR = FALSE

SE L'OPERAZIONE INSERISCE O MODIFICA IN TAPPA:

PART, ARR := SELECT p.ORA, z.ORA FROM PARTENZA_DA p, ARRIVO_IN z
WHERE p.IT = z.IT AND z.IT = NEW.IT;

ERROR := (NEW.ARRIVO < PART) OR (NEW.PARTENZA > ARR);

SE L'OPERAZIONE INSERISCE O MODIFICA IN PARTENZA_DA:

MINT := SELECT MIN(ARRIVO) FROM TAPPA WHERE IT = NEW.IT;;

ERROR := NEW.ORA > MINT

SE L'OPERAZIONE INSERISCE O MODIFICA IN ARRIVO_IN:

MAXT := SELECT MAX(PARTENZA) FROM TAPPA WHERE IT = NEW.IT;;

ERROR := NEW.ORA < MAXT

SE L'OPERAZIONE INSERISCE O MODIFICA IN ITINERARIO:

PART, ARR := SELECT p.ORA, z.ORA FROM PARTENZA_DA p, ARRIVO_IN z
WHERE p.IT = z.IT AND z.IT = NEW.NOME;

MINT, MAXT := SELECT MIN(ARRIVO), MAX(PARTENZA) FROM TAPPA

WHERE IT = NEW.NOME.

ERROR := NOT (PART < MINT < MAXT < ARR);

SE ERROR = TRUE BLOCCA L'OPERAZIONE

ALTRIMENTI CONSENTILA

TRIGGER PER V. PRENOTAZIONE. SCADENZA:

- OPERAZIONI:

- INSERIMENTO O MODIFICA IN PRENOTAZIONE

- ISTANTE DI INVOCAZIONE: PRIMA DELL'OPERAZIONE

- FUNZIONE:

SIA ERROR = FALSE

C := SELECT INIZIO FROM CROCIERA WHERE CODICE = NEW.CROCIERA;

ERROR := NEW.ISTANTE >= C.

SE ERROR = TRUE BLOCCA L'OPERAZIONE

ALTRIMENTI CONSENTILA

TRIGGER PER V. PRENOTAZIONE. NO_OVERBOOKING:

- OPERAZIONI:

- INSERIMENTO O MODIFICA IN PRENOTAZIONE

- MODIFICA IN CROCIERA

- MODIFICA IN NAVE

- ISTANTE DI INVOCAZIONE: PRIMA DELL'OPERAZIONE

- FUNZIONE:

SIA ERROR = FALSE

SE L'OPERAZIONE INSERISCE O MODIFICA IN PRENOTAZIONE:

N := SELECT SUM(p.NPOSTI) FROM PRENOTAZIONE p WHERE p.CROCIERA = NEW.CROCIERA;

C := SELECT n.MAX_POSTI FROM NAVE n, CROCIERA c WHERE c.NAVE = n.NOME

AND c.CODICE = NEW.CROCIERA;

ERROR := (C - N) < NEW.NPOSTI;

SE L'OPERAZIONE MODIFICA IN CROCIERA:

N := SELECT SUM(p.NPOSTI) FROM PRENOTAZIONE p WHERE p.CROCIERA = NEW.CODICE;

C := SELECT n.MAX_POSTI FROM NAVE n WHERE n.NOME = NEW.NOME;

ERROR := (C - N) < 0;

.

:

:

SE L'OPERAZIONE MODIFICA IN NAVE:

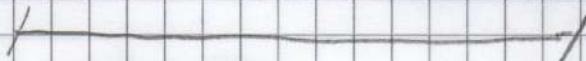
N := SELECT c.CODICE AS CR, SUM(p.NPOSTI) AS POSTI FROM PRENOTAZIONE p,
CROCIERA c WHERE c.NAVE = NEW.NOME AND p.CROCIERA = c.CODICE;
GROUP BY c.CODICE;

PER OGNI (CR POSTI) IN N:

ERROR := ERROR OR ((C-POSTI) < 0)

SE ERROR = TRUE BLOCCA L'OPERAZIONE

ALTRIMENTI CONSENTILA



SPECIFICHE REALIZZATIVE DEGLI USE-CASE

USE-CASE GESTIONE_CLIENTI

REGISTRA_CLIENTE(CF: CF_T, NOME: VARCHAR(30), COGNOME: VARCHAR(30), NASC: DATE,
IND: INDIR_T) : CF_T

INS ← INSERT INTO CLIENTE VALUES (:CF, :NOME, :COGNOME, :NASC, :IND);

IF INS RAPPR. ERR. THEN INOLTRALO;

ELSE RETURN CF;

USE-CASE GESTIONE_PRENOTAZIONI

IST ← PRENOTA(CL: CF_T, CR: VARCHAR(20), NP: INT_POS) : (CF_T, DATETIME)
CURRENT
TIMESTAMP
INSE ← INSERT INTO PRENOTAZIONE VALUES (:CF, :IST, :T, :NP, :CR);
IF INSE RAPPR. ERR. THEN INOLTRALO;
ELSE RETURN (CF, IST);

ANNULLA_PREN(CLP: CF_T, IP: DATETIME)

DEL ← DELETE FROM PRENOTAZIONE WHERE (CLIENTE, ISTANTE) = (:CLP, :IP);

IF DEL RAPPR. ERR. THEN INOLTRALO;

USE-CASE OPERAZIONI_CROCIERE

LDM_TRADIZIONALE(C: VARCHAR(20)) : BOOLEAN

OK ← SELECT EXIST(SELECT 1 FROM CROCIERA_LDM WHERE CODICE = :C);

IF OK RAPPR. ERR. THEN INOLTRALO;

ELSE IF OK = FALSE THEN

INOLTRA ERR. 'CROCIERA NON TROVATA'; // SENZA QUESTO CONTROLLO LA FUNZIONE

POTEBBE ESSERE RICHIAMATA SU CROCIERE NON - LDM

ELSE

D ← 0, R ← 0; /* SELECT NOME FROM CROCIERA WHERE NOME */

P_R, P_D ← SELECT d.ROMANTICA, d.DIVERTENTE FROM DESTINAZIONE d, PARTEZIALE p
WHERE (d.NOME, d.CONTINENTE) = (p.DST_N, p.DST_C) AND p.IT = :i;

IF P_R, P_D RAPPR. ERR. THEN INOLTRALO;

IF P_R = TRUE THEN R ← R+1;

IF P_D = TRUE THEN D ← D+1;

A_R, A_D ← SELECT d.ROMANTICA, d.DIVERTENTE FROM DESTINAZIONE d, ARRIVO_IN_2
WHERE (d.NOME = d.CONTINENTE) = (2.DST_N, 2.DST_C) AND 2.IT = :i;

IF A_R, A_D RAPPR. ERR. THEN INOLTRALO;

IF A_R = TRUE THEN R ← R+1;

IF A_D = TRUE THEN D ← D+1;

T_R ← SELECT d.NOME, d.CONTINENTE FROM DESTINAZIONE d, TAPPA t
WHERE (t.IT, t.DST_C, t.DST_N) = (:i, d.CONTINENTE, d.NOME) AND d.ROMANTICA

IF T_D RAPPR. ERR. THEN INOLTRALO;
 $T_D \leftarrow$ SELECT COUNT(d.NOME d.CONTINENTE) FROM DESTINAZIONE d, TAPPA t,
 WHERE (d.NOME, d.CONTINENTE) = (t.DST_N, t.DST_C) AND
 AND t.IT = :i AND d.DIVERTENTE;
 IF T_D RAPPR. ERR. THEN INOLTRALO;
 $R \leftarrow R + T_D$; $D \leftarrow D + T_D$;
 RETURN ($R \geq D$);

 * $i \leftarrow$ SELECT ITINERARIO FROM CROCIERA WHERE CODICE = :c;
 IF i RAPPR. ERR. THEN INOLTRALO

CROCIERA_ESOTICA(C: VARCHAR(20)) : BOOLEAN
 $RES \leftarrow$ SELECT EXIST (SELECT 1 FROM DESTINAZIONE d, TAPPA t, ITINERARIO i, CROCIERA c
 WHERE d.CONTINENTE <> 'EU' AND c.CODICE = :c
 AND (d.NOME d.CONTINENTE i.NOME) = (t.DST_N, t.DST_C, t.IT)
 AND c.ITINERARIO = i.NOME);

 $RES_{pa} \leftarrow$ SELECT EXIST (SELECT 1 FROM DESTINAZIONE d₁, DESTINAZIONE d₂, PARTENZA_IN p₁,
 ARRIVO_IN p₂, ITINERARIO i, CROCIERA c
 WHERE c.CODICE = :c AND (d₁.CONTINENTE <> 'EU' OR
 d₂.CONTINENTE <> 'EU') AND
 (d₁.NOME d₁.CONTINENTE i.NOME) = (p₁.DST_N p₁.DST_C, p₁.IT)
 AND (d₂.NOME d₂.CONTINENTE i.NOME) = (p₂.DST_N p₂.DST_C, p₂.IT)
 AND c.ITINERARIO = i.NOME);

 IF RES₁ RAPPR. ERR. THEN INOLTRALO;
 IF RES_{pa} " " " " " "
 RETURN RES₁ OR RES_{pa};

USE-CASE STATISTICHE CLIENTI

ETA_MEDIA_ESOTICA(C: CF_T(1..N)) : REAL
 $C_E \leftarrow$ SELECT AVG(CURRENT_DATE - c.NASCITA) FROM CLIENTE c, PRENOTAZIONE p,
 CROCIERA cr WHERE c.CF IN :C AND (p.CLIENTE, p.CROCIERA) = (c.CF, cr.CODICE)
 AND OPERAZIONI_CROCIERE.CROCIERA_ESOTICA(ch.CODICE);
 IF C_E RAPPR. ERR. THEN INOLTRALO;
 ELSE RETURN C_E;

USE-CASE STATISTICHE_DESTINAZIONI

DEST_GETTONATA(n: VARCHAR(50), c: CONT_T) : BOOLEAN
 $C \leftarrow$ SELECT c.CODICE FROM CROCIERA c, ITINERARIO i, PARTENZA_IN p, ARRIVO_IN t, TAPPA t
 WHERE c.ITINERARIO = i.NOME AND ((p.DST_N, p.DST_C, i.IT) = (:n, :c, i.NOME)
 OR (p.DST_N, p.DST_C, p.IT) = (:n, :c, i.NOME)
 OR (t.DST_N, t.DST_C, t.IT) = (:n, :c, i.NOME));
 IF C RAPPR. ERR. THEN INOLTRALO;
 ELSE
 $T \leftarrow$ SELECT CODICE FROM CROCIERALDM WHERE CODICE IN :C AND
 OPERAZIONI_CROCIERE.LDM_TRADIZIONALE(CODICE) = TRUE;
 IF T RAPPR. ERR. THEN INOLTRALO;
 ELSE
 $A \leftarrow C \setminus T$;
 RETURN (|T| ≥ 10 OR |A| ≥ 15);

PERC_GETTONATE(D:(VARCHAR(50),CONT_T)(1,N)):REAL

~~n <= 0~~

FOREACH (n,c) IN D DO

IF STATISTICHE_DESTINAZIONI.DEST_GETTONATA(n,c)=TRUE THEN

n ← n + 1;

DONE

RETURN $\frac{n}{|D|} * 100$;