

TESTO

Il Ministero dell'Ambiente commissiona lo sviluppo di un nuovo Sistema Informativo per censire la fauna e la flora presenti nei vari parchi naturali di ciascuna regione. Le informazioni della fauna da raccogliere riguardano l'ordine di appartenenza (mammiferi, rettili, uccelli, ...), la specie (scoiattolo, volpe, orso, vipera, falco, aquila, ...), il numero di esemplari adulti e cuccioli (con distinzione maschio, femmina), lo stato di salute di ogni esemplare. La flora viene distinta in alberi (quercia, faggio, acacia, ...), arbusti (lavanda, rosmarino, rododendro, ...) e piante erbacee (fragola di bosco, margherita, primula, ...) ed è necessario catalogarle memorizzandone anche la stagione di fioritura e altre caratteristiche peculiari.

In ciascun parco, un responsabile (guardaparco) raccoglie le informazioni attraverso un software installato sul notebook in dotazione; il programma, dopo avere acquisito i dati dalle maschere di immissione, li memorizza in un database.

Ciascuna regione, dotata del nuovo Sistema Informativo, deve provvedere alla gestione della raccolta dati dei propri parchi ai fini di produrre e inviare al Ministero dell'Ambiente dei report periodici. Il candidato, formulate le opportune ipotesi aggiuntive, realizzi:

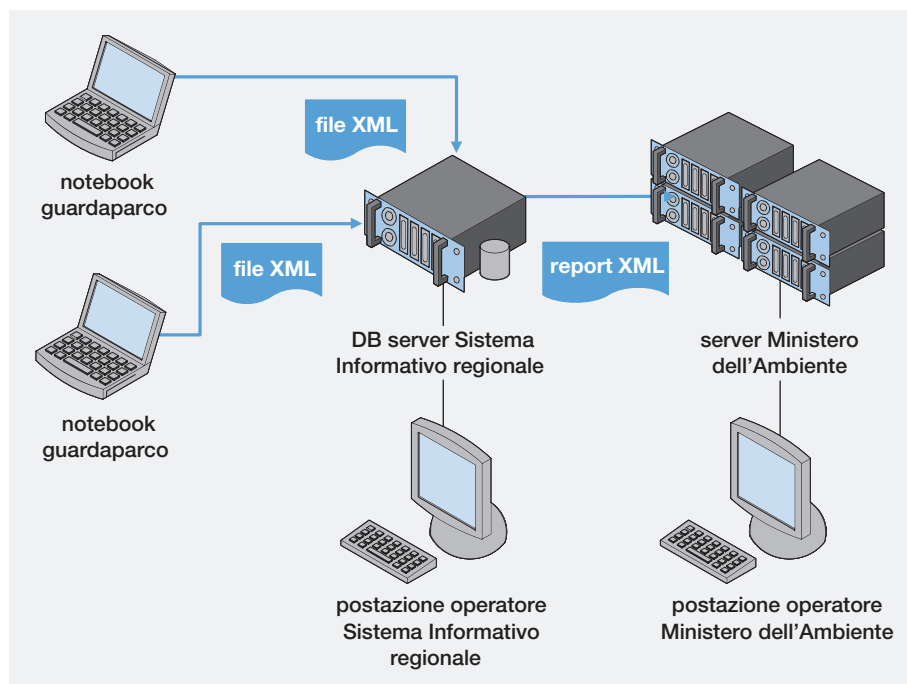
- un'analisi della realtà di riferimento, completa dello schema funzionale dell'architettura proposta, evidenziandone le principali componenti e la tecnologia utilizzata per la trasmissione delle informazioni;
- uno schema concettuale e uno schema logico del database;
- la definizione delle relazioni e le seguenti interrogazioni espresse in linguaggio SQL:
 - 1 visualizzare l'elenco di tutti gli esemplari di fauna, suddivisi per specie, presenti nei vari parchi;
 - 2 visualizzare tutti gli esemplari a rischio di estinzione;
 - 3 calcolare e visualizzare il numero di nascite di un certo esemplare nell'arco di un anno di monitoraggio;
 - 4 calcolare e visualizzare il numero totale di diverse specie di arbusti presenti nei vari parchi della regione;
 - 5 visualizzare quante specie diverse di pino sono presenti in ciascun parco;
 - 6 calcolare il numero medio dei cuccioli di ciascuna specie presenti in tutti i parchi della regione;
 - 7 visualizzare l'esemplare più anziano di ogni specie presente in un determinato parco.
- La codifica in un linguaggio di programmazione a scelta di un segmento significativo del progetto realizzato.

4.1 Analisi della realtà di riferimento e ipotesi aggiuntive

Ipotizzando la difficoltà di una connessione di rete permanente nelle aree di un parco naturale, si prevede che l'applicazione dei notebook in dotazione alle guardie del parco produca dei file in formato XML contenenti i dati relativi alla flora e alla fauna oggetto del censimento. Non richiedendo l'operazione di censimento un aggiornamento in tempo reale dei dati acquisiti, i file vengono successivamente trasmessi al Sistema Informativo regionale e costituiscono l'input del software di gestione del database che

viene mantenuto e gestito a questo livello. In modo analogo il software di gestione del database regionale esporta in un file in formato XML i report periodici da inviare al Ministero dell'Ambiente. Non si ritiene utile dettagliare le modalità di trasmissione dei file e dei report in formato XML: date le caratteristiche di periodicità e di non urgenza dei dati essa può anche consistere nel semplice invio mediante il sistema di posta elettronica certificata del Sistema Informativo, se presso il centro regionale e al Ministero dell'Ambiente sono presenti operatori incaricati di effettuare la gestione dei file ricevuti rispettivamente dai parchi e dai centri regionali³.

3. La soluzione più adatta è comunque quella di un'applicazione client/server che consenta ai guardaparco di caricare direttamente nel database regionale i risultati del censimento e, analogamente, agli operatori del Sistema Informativo regionale di inoltrare i report periodici al Ministero dell'Ambiente.



La soluzione proposta riguarda il database che verrà gestito da ogni singola regione ed è sviluppata tenendo conto delle seguenti ipotesi aggiuntive⁴:

- in relazione alle informazioni relative alla fauna si prevede di censire i singoli esemplari di alcune specie messe sotto controllo (questa soluzione è infatti l'unica che consente di formulare alcune delle query richieste dal testo);
- l'identificatore numerico del singolo esemplare di fauna viene generato dall'applicazione del notebook del guardaparco con modalità che non ne consentono la duplicazione⁵;
- il file XML prodotto dall'applicazione dei notebook dei guardaparco relativo alla flora ha la struttura mostrata dal seguente esempio:

```
<?xml version="1.0" ?>
<flora>
  <ID-parco>1</ID-parco>
  <parco>Parco Alpi Apuane</parco>
  <regione>Toscana</regione>
  <specie>
    <ID-specie>200</ID-specie>
    <denominazione>Quercia</denominazione>
```

4. Per semplicità non viene fornito lo schema XSD del formato dei vari file XML, ma solo un esempio per ciascuno di essi.

5. Per esempio parti del codice potrebbero riferire univocamente il parco e il guardaparco.

```

</specie>
<specie>
  <ID-specie>101</ID-specie>
  <denominazione>Rododendro</denominazione>
</specie>
<specie>
  <ID-specie>3</ID-specie>
  <denominazione>Campanula</denominazione>
</specie>
</flora>

```

- il file XML prodotto dall'applicazione dei notebook dei guardaparco relativo agli esemplari della fauna ha la struttura mostrata dal seguente esempio:

```

<?xml version="1.0" ?>
<fauna>
  <ID-parco>2</ID-parco>
  <parco>Parco S. Rossore, Migliarino, Massaciuccoli</parco>
  <regione>Toscana</regione>
  <esemplare>
    <ID-specie>6</ID-specie>
    <specie>Volpe</specie>
    <ID-esemplare>1234</ID-esemplare>
    <nome>Red</nome>
    <mese-nascita>03</mese-nascita>
    <anno-nascita>2011</anno-nascita>
    < Sesso>M</ Sesso>
    <salute>Buona</salute>
  </esemplare>
  <esemplare>
    <ID-specie>6</ID-specie>
    <specie>Volpe</specie>
    <ID-esemplare>1256</ID-esemplare>
    <nome>Toby</nome>
    <mese-nascita>04</mese-nascita>
    <anno-nascita>2010</anno-nascita>
    < Sesso>F</ Sesso>
    <salute>Ottima</salute>
  </esemplare>
  <esemplare>
    <ID-specie>3</ID-specie>
    <specie>Lupo</specie>
    <ID-esemplare>1278</ID-esemplare>
    <nome>Buck</nome>
    <mese-nascita>03</mese-nascita>
    <anno-nascita>2010</anno-nascita>
    < Sesso>M</ Sesso>
    <salute>Cattiva</salute>
  </esemplare>
</fauna>

```

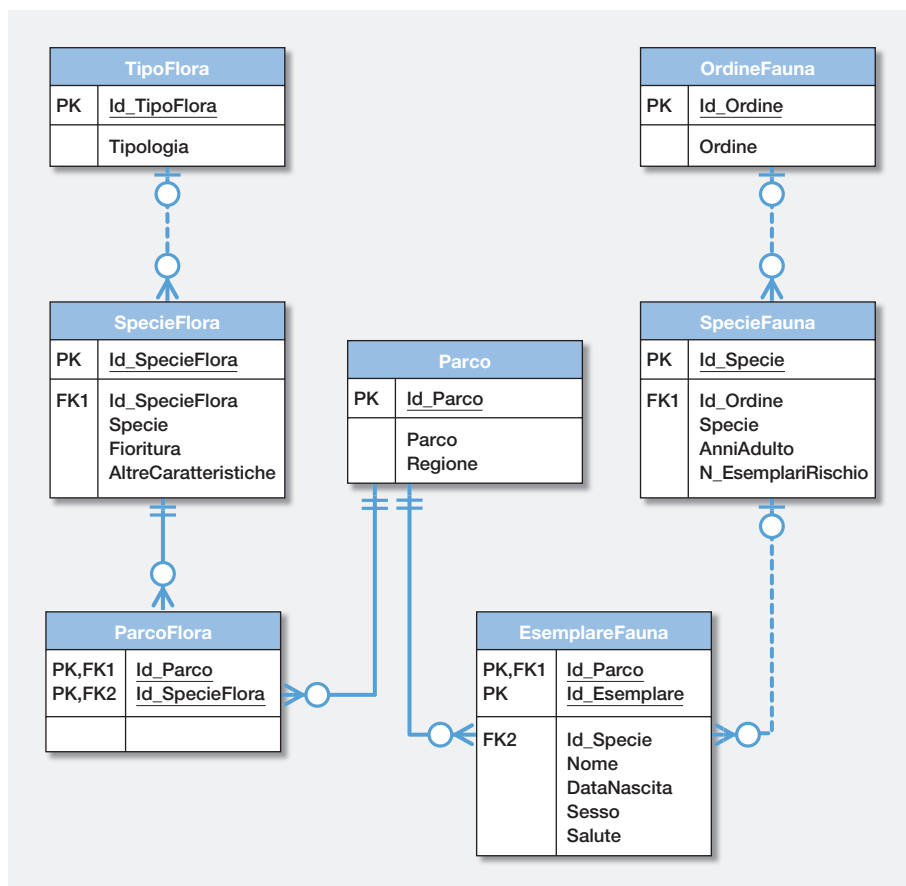
- il report periodico in formato XML esportato dal Sistema Informativo della regione per la trasmissione al Ministero dell'Ambiente è relativo ai soli esemplari della fauna e ha la struttura mostrata dal seguente esempio:

```

<?xml version="1.0" ?>
  <fauna>
    <esemplare>
      <ID-parco>2</ID-parco>
      <parco>Parco S. Rossore, Migliarino, Massaciuccoli</parco>
      <regione>Toscana</regione>
      <ID-specie>6</ID-specie>
      <specie>Volpe</specie>
      <ID-esemplare>1234</ID-esemplare>
      <nome>Red</nome>
      < mese-nascita>03</ mese-nascita>
      < anno-nascita>2011</ anno-nascita>
      < sesso>M</ sesso>
      < salute>Buona</ salute>
    </esemplare>
    <esemplare>
      <ID-parco>2</ID-parco>
      <parco>Parco S. Rossore, Migliarino, Massaciuccoli</parco>
      <regione>Toscana</regione>
      <ID-specie>6</ID-specie>
      <specie>Volpe</specie>
      <ID-esemplare>1256</ID-esemplare>
      <nome>Toby</nome>
      < mese-nascita>04</ mese-nascita>
      < anno-nascita>2010</ anno-nascita>
      < sesso>F</ sesso>
      < salute>Ottima</ salute>
    </esemplare>
    <esemplare>
      <ID-parco>1</ID-parco>
      <parco>Parco Alpi Apuane</parco>
      <regione>Toscana</regione>
      <ID-specie>3</ID-specie>
      <specie>Lupo</specie>
      <ID-esemplare>1278</ID-esemplare>
      <nome>Buck</nome>
      < mese-nascita>03</ mese-nascita>
      < anno-nascita>2010</ anno-nascita>
      < sesso>M</ sesso>
      < salute>Cattiva</ salute>
    </esemplare>
  </fauna>

```

4.2 Diagramma database



4.3 Tabelle database

EsemplareFauna

È la tabella in cui sono censiti gli i singoli esemplari delle varie specie di animali:

Campo	R	Tipo	Dim.	Descrizione
Id_Parco	PK	Intero		Identificativo parco
Id_Esemplare	PK	Carattere	50	Identificativo esemplare
Id_Specie	FK	Intero		Identificativo specie
Nome		Carattere	50	Nome dell'esemplare
DataNascita		Data		Data di nascita (essendo il monitoraggio mensile, il valore del giorno è sempre 1)
Sesso		Carattere	1	Sesso (M/F)
Salute		Carattere	10	Descrizione dello stato di salute (Buona, Cattiva, ...)

Chiave Primaria: Id_Parco, Id_Esemplare

Chiave Esterna: Id_Parco → Parco.Id_Parco

Chiave Esterna: Id_Specie → SpecieFauna.Id_Specie

La seguente è una possibile istanza della tabella **EsemplareFauna**:

Id_Parco	Id_Esemplare	Id_Specie	Nome	DataNascita	Sesso	Salute
1	1301	3	Olaf	2008/03/01	M	Ottima
1	1302	3	Gala	2010/04/01	F	Buona
1	1501	5	Yoghi	2006/04/01	M	Ottima
1	1502	5	Clarabella	2007/05/01	F	Ottima
1	1601	6	Renard	2009/04/01	M	Ottima
1	1602	6	Titù	2010/03/01	F	Ottima
1	1701	7	Corno Nero	2008/04/01	M	Ottima
2	2301	3	Zanna Bianca	2004/03/01	M	Buona
2	2302	3	Gamma	2008/03/01	F	Ottima
2	2501	5	Bubu	2004/03/01	M	Buona
2	2502	5	Yara	2003/04/01	F	Buona
2	2701	7	Big Horn	2007/05/01	M	Buona
2	2702	7	Cynthia	2007/05/01	F	Ottima
2	2703	7	Unico	2006/04/01	M	Cattiva

OrdineFauna

È la tabella in cui sono riportati vari ordini della fauna:

Campo	R	Tipo	Dim.	Descrizione
Id_Ordine	PK	Intero		Identificativo ordine fauna
Ordine		Carattere	50	(Mammiferi, Rettili, Uccelli, ...)

Chiave Primaria: Id_Ordine

La seguente è una possibile istanza della tabella *OrdineFauna*:

Id_Ordine	Ordine
1	Mammiferi
2	Rettili
3	Uccelli
4	Pesci

Parco

È la tabella in cui sono elencati i parchi regionali:

Campo	R	Tipo	Dim.	Descrizione
Id_Parco	PK	Intero		Identificativo parco
Parco		Carattere	50	Denominazione del parco
Regione		Carattere	50	Regione di appartenenza

Chiave Primaria: Id_Parco

La seguente è una possibile istanza della tabella *Parco*:

Id_Parco	Parco	Regione
1	Parco Alpi Apuane	Toscana
2	Parco S. Rossore, Migliarino, Massaciuccoli	Toscana
3	Parco della Maremma	Toscana

ParcoFlora

È la tabella in cui sono censite le varie specie della flora di ogni parco:

Campo	R	Tipo	Dim.	Descrizione
Id_Parco	PK	Intero		Identificativo parco
Id_SpecieFlora	PK	Intero		Identificativo specie flora

Chiave Primaria: Id_Parco, Id_SpecieFlora

Chiave Esterna: Id_Parco → Parco.Id_Parco

Chiave Esterna: Id_SpecieFlora → SpecieFlora.Id_SpecieFlora

La seguente è una possibile istanza della tabella *ParcoFlora*:

Id_Parco	Id_SpecieFlora
1	1
1	2
1	3
1	100
1	101
1	102
1	200
1	203
1	205
1	207
2	1
2	2
2	3
2	100
2	101
2	200
2	206
2	207

SpecieFauna

È la tabella in cui sono riportate le varie specie di fauna e l'ordine cui appartengono:

Campo	R	Tipo	Dim.	Descrizione
Id_Specie	PK	Intero		Identificativo specie fauna
Id_Ordine	FK	Intero		Identificativo ordine fauna
Specie		Carattere	50	Nome della specie (Scoiattolo, Volpe, Orso, Falco, Aquila, ...)
AnniAdulto		Intero		Numero degli anni che impiega un esemplare della specie a divenire adulto
N_EsempariRischio		Intero		Numero minimo di esemplari sotto il quale la specie è a rischio di estinzione

Chiave Primaria: Id_Specie

Chiave Esterna: Id_Ordine → OrdineFauna.Id_Ordine

La seguente è una possibile istanza della tabella *SpecieFauna*:

Id_Specie	Id_Ordine	Specie	AnniAdulto	N_EsemplarilRischio
1	1	Scoiattolo	1	1000
2	1	Cinghiale	1	1500
3	1	Lupo	2	200
4	1	Lepre	1	2000
5	1	Orso	3	40
6	1	Volpe	2	300
7	1	Stambecco	3	200
8	1	Camoscio	3	100
9	1	Marmotta	2	1500
100	2	Vipera	1	1000
200	3	Falco	2	200
201	3	Aquila	3	20
202	3	Corvo	2	2000
300	4	Trota	1	1000
301	4	Cavedano	2	1500

SpecieFlora

È la tabella in cui sono riportate le varie specie di flora suddivise per tipologia:

Campo	R	Tipo	Dim.	Descrizione
Id_SpecieFlora	PK	Intero		Identificativo specie flora
Id_TipoFlora	FK	Intero		Identificativo tipologia flora
Specie		Carattere	20	Nome della specie
Fioritura		Carattere	10	Mese di fioritura (se pertinente)
AltreCaratteristiche		Carattere	250	Altre caratteristiche

Chiave Primaria: Id_SpecieFlora

Chiave Esterna: Id_TipoFlora → TipoFlora.Id_TipoFlora

La seguente è una possibile istanza della tabella *SpecieFlora*:

Id_SpecieFlora	Id_TipoFlora	Specie	Fioritura	AltreCaratteristiche
1	1	Fragola	Maggio	
2	1	Margherita	Marzo	
3	1	Campanula	Febbraio	
100	2	Lavanda	Aprile	
101	2	Rododendro	Aprile	
102	2	Mirto	Maggio	
103	2	Rosmarino	Marzo	
200	3	Quercia		
201	3	Leccio		
202	3	Olmo		
203	3	Ontano		
204	3	Platano		
205	3	Pino		
206	3	Abete		
207	3	Larice		
208	3	Pino d'Aleppo		
209	3	Cedro		

TipoFlora

È la tabella in cui sono elencate le varie tipologie di flora:

Campo	R	Tipo	Dim.	Descrizione
Id_TipoFlora	PK	Intero		Identificativo tipologia flora
Tipologia		Carattere	20	(Alberi, Arbusti, Erbacee, ...)

Chiave Primaria: Id_TipoFlora

La seguente è una possibile istanza della tabella *TipoFlora*:

Id_TipoFlora	Tipologia
1	Erbacee
2	Arbusti
3	Alberi

4.4 DB-schema

```
CREATE TABLE OrdineFauna(  
  Id_Ordine INTEGER,  
  Ordine VARCHAR(50),  
  CONSTRAINT PrimaryKey PRIMARY KEY (Id_Ordine)  
);  
  
CREATE TABLE Parco(  
  Id_Parco INTEGER,  
  Parco VARCHAR(50),  
  Regione VARCHAR(50),  
  CONSTRAINT PrimaryKey PRIMARY KEY (Id_Parco)  
);  
  
CREATE TABLE SpecieFauna(  
  Id_Specie INTEGER,  
  Id_Ordine INTEGER,  
  Specie VARCHAR(50),  
  AnniAdulto INTEGER,  
  N_EsemplariRischio INTEGER,  
  CONSTRAINT OrdineFaunaSpecieFauna FOREIGN KEY (Id_Ordine)  
    REFERENCES OrdineFauna (Id_Ordine),  
  CONSTRAINT PrimaryKey PRIMARY KEY (Id_Specie)  
);  
  
CREATE TABLE TipoFlora(  
  Id_TipoFlora INTEGER,  
  Tipologia VARCHAR(20),  
  CONSTRAINT PrimaryKey PRIMARY KEY (Id_TipoFlora)  
);
```

```

CREATE TABLE EsempireFauna(
  Id_Parco INTEGER,
  Id_Esempire VARCHAR(50),
  Id_Specie INTEGER,
  Nome VARCHAR(50),
  DataNascita DATE,
  Sesso VARCHAR(1),
  Salute VARCHAR(10),
  CONSTRAINT ParcoEsempireFauna FOREIGN KEY (Id_Parco) REFERENCES Parco (Id_Parco),
  CONSTRAINT PrimaryKey PRIMARY KEY (Id_Parco, Id_Esempire),
  CONSTRAINT SpecieFaunaEsempireFauna FOREIGN KEY (Id_Specie)
                                     REFERENCES SpecieFauna (Id_Specie)
);

CREATE TABLE SpecieFlora(
  Id_SpecieFlora INTEGER,
  Id_TipoFlora INTEGER,
  Specie VARCHAR(20),
  Fioritura VARCHAR(10),
  AltreCaratteristiche VARCHAR(250),
  CONSTRAINT PrimaryKey PRIMARY KEY (Id_SpecieFlora),
  CONSTRAINT TipoFloraSpecieFlora FOREIGN KEY (Id_TipoFlora)
                                   REFERENCES TipoFlora (Id_TipoFlora)
);

CREATE TABLE ParcoFlora(
  Id_Parco INTEGER,
  Id_SpecieFlora INTEGER,
  CONSTRAINT PrimaryKey PRIMARY KEY (Id_Parco, Id_SpecieFlora),
  CONSTRAINT ParcoParcoFlora FOREIGN KEY (Id_Parco) REFERENCES Parco (Id_Parco),
  CONSTRAINT SpecieFloraParcoFlora FOREIGN KEY (Id_SpecieFlora)
                                    REFERENCES SpecieFlora (Id_SpecieFlora)
);

```

4.5 Query

1

```

SELECT EsempireFauna.Id_Esempire, SpecieFauna.Specie,
       Parco.Parco, EsempireFauna.Nome,
       EsempireFauna.DataNascita, EsempireFauna.Sesso,
       EsempireFauna.Salute
FROM Parco INNER JOIN ((OrdineFauna INNER JOIN SpecieFauna
ON OrdineFauna.Id_Ordine = SpecieFauna.Id_Ordine)
INNER JOIN EsempireFauna
ON SpecieFauna.Id_Specie = EsempireFauna.Id_Specie)
ON Parco.Id_Parco = EsempireFauna.Id_Parco
ORDER BY SpecieFauna.Id_Specie, Parco.Parco;

```

2

L'ambigua richiesta del testo è stata interpretata come «specie» (e non «esemplare») a rischio di estinzione:

```
SELECT Specie, N_EsemplariRischio, N_Esemplari
FROM SpecieFauna, (SELECT Id_Specie, COUNT(*) AS N_Esemplari
                   FROM EsemplareFauna
                   GROUP BY id_Specie) AS T
WHERE SpecieFauna.Id_Specie = T.Id_Specie AND N_Esemplari <=
      N_EsemplariRischio;
```

3

L'ambigua richiesta del testo è stata interpretata come numero di nascite di una certa «specie» (e non «esemplare») nell'arco di un anno solare di monitoraggio:

```
SELECT Specie, COUNT(*) AS N_Nascite
FROM SpecieFauna INNER JOIN EsemplareFauna
      ON SpecieFauna.Id_Specie = EsemplareFauna.Id_Specie
WHERE YEAR(DataNascita) = '...'
GROUP BY Specie;
```

4

```
SELECT SpecieFlora.Specie, COUNT(*) AS TipoArbusti
FROM (TipoFlora INNER JOIN SpecieFlora
      ON TipoFlora.Id_TipoFlora = SpecieFlora.Id_TipoFlora
      INNER JOIN ParcoFlora
      ON SpecieFlora.Id_SpecieFlora = ParcoFlora.Id_SpecieFlora
WHERE TipoFlora.Tipologia='Arbusti'
GROUP BY SpecieFlora.Specie;
```

5

```
SELECT Parco.Parco, COUNT(*) AS NSpeciePino
FROM (TipoFlora INNER JOIN SpecieFlora
      ON TipoFlora.Id_TipoFlora = SpecieFlora.Id_TipoFlora)
      INNER JOIN (Parco INNER JOIN ParcoFlora
      ON Parco.Id_Parco = ParcoFlora.Id_Parco)
      ON SpecieFlora.Id_SpecieFlora = ParcoFlora.Id_SpecieFlora
WHERE TipoFlora.Tipologia='Alberi' AND SpecieFlora.Specie
      LIKE 'Pino%'
GROUP BY Parco.Parco;
```

6

Data l'ambiguità della richiesta si presentano due query: la prima conta il numero di cuccioli per ogni parco e per ogni specie, mentre la seconda calcola la media dei cuccioli per specie indipendentemente dal parco:

```

SELECT Parco.Parco, SpecieFauna.Specie, COUNT(*) AS Cuccioli
FROM SpecieFauna INNER JOIN (Parco INNER JOIN EsempolareFauna
    ON Parco.Id_Parco = EsempolareFauna.Id_Parco)
    ON SpecieFauna.Id_Specie = EsempolareFauna.Id_Specie
WHERE (YEAR(DataNascita) - YEAR(CURRENT_DATE())) < AnniAdulto
GROUP BY Parco.Parco, SpecieFauna.Specie;

SELECT Specie, AVG(Cuccioli) AS MediaCuccioli
FROM (SELECT Parco.Parco, SpecieFauna.Specie, COUNT(*) AS Cuccioli
    FROM SpecieFauna INNER JOIN (Parco INNER JOIN EsempolareFauna
        ON Parco.Id_Parco = EsempolareFauna.Id_Parco)
        ON SpecieFauna.Id_Specie = EsempolareFauna.Id_Specie
    WHERE (YEAR(DataNascita) - YEAR(CURRENT_DATE())) < AnniAdulto
    GROUP BY Parco.Parco, SpecieFauna.Specie) AS T
GROUP BY Specie;

```

7

```

SELECT *
FROM EsempolareFauna, (SELECT T.Id_Parco, Id_Specie, MAX(Anni) AS Eta
    FROM
        (SELECT Parco.Id_Parco, SpecieFauna.Id_Specie,
            Id_Esempolare,
            YEAR(CURRENT_DATE()) - YEAR(DataNascita) AS Anni
        FROM Parco INNER JOIN
            (SpecieFauna INNER JOIN EsempolareFauna
                ON SpecieFauna.Id_Specie = EsempolareFauna.Id_Specie)
            ON Parco.Id_Parco = EsempolareFauna.Id_Parco
        WHERE Parco.Parco = '...') AS T
        GROUP BY Id_Parco, Id_Specie) AS T1
WHERE EsempolareFauna.Id_Parco = T1.Id_parco
    AND EsempolareFauna.Id_Specie = T1.Id_Specie
    AND YEAR(CURRENT_DATE()) - YEAR(DataNascita) = Eta;

```

o, in alternativa:

```

SELECT *
FROM EsempolareFauna, (SELECT T.Id_Parco, Id_Specie, MIN(DataNascita) AS MinData
    FROM
        (SELECT Parco.Id_Parco, SpecieFauna.Id_Specie,
            Id_Esempolare, DataNascita
        FROM Parco INNER JOIN
            (SpecieFauna INNER JOIN EsempolareFauna
                ON SpecieFauna.Id_Specie = EsempolareFauna.Id_Specie)
            ON Parco.Id_Parco = EsempolareFauna.Id_Parco
        WHERE Parco.Parco = '...') AS T
        GROUP BY Id_Parco, Id_Specie) AS T1
WHERE EsempolareFauna.Id_Parco = T1.Id_parco
    AND EsempolareFauna.Id_Specie = T1.Id_Specie
    AND DataNascita = MinData;

```

La seguente classe Java *Import* consente – mediante il metodo *parseDocument* – di caricare nel database un file XML relativo al censimento degli esemplari della fauna; nel caso che l'identificativo di esemplare non sia presente nel database viene creato un nuovo record, se è invece già presente viene aggiornato il record esistente. Il *parsing* del file XML avviene con tecnica SAX, mentre per l'accesso al DBMS si utilizzano le API JDBC; a questo scopo si è ipotizzato un server My-SQL in esecuzione sullo stesso computer dell'applicazione e che l'accesso al database «Parco» venga effettuato come utente «root» privo di password.

```
import java.io.*;
import java.sql.*;
import javax.xml.parsers.*;
import org.xml.sax.*;
import org.xml.sax.helpers.*;

public class Import extends DefaultHandler {
    private String text;
    private int ID_parco;
    private int ID_specie;
    private int ID_esemplare;
    private String nome;
    private int mese_nascita;
    private int anno_nascita;
    private String sesso;
    private String salute;
    private int inserted; // numero di nuovi record inseriti
    private int updated; // numero di record esistenti aggiornati
    private Connection con;
    final String URL = "jdbc:mysql://localhost:3306";
    final String database = "Parco";
    final String user = "root";
    final String password = "";

    public Import() throws SQLException {
        // connessione al server DB
        con = DriverManager.getConnection(URL+"/"+database, user, password);
    }

    public void parseDocument(String filename) throws SAXException,
        ParserConfigurationException,
        IOException {
        SAXParserFactory factory = SAXParserFactory.newInstance();
        SAXParser parser = factory.newSAXParser();
    }
}
```

```

    parser.parse(filename, this);
}

// metodi di gestione degli eventi SAX
public void startDocument() {
    inserted = 0;
    updated = 0;
}
public void endDocument() {
}
public void startElement(String uriNamespace, String localNamespace,
    String qualifiedName, Attributes attributes)
    throws SAXException {
    text = "";
}
public void characters(char[] characters, int start, int length)
    throws SAXException {
    text = new String(characters, start, length);
}
public void endElement(String uriNamespace, String localNamespace,
    String qualifiedName) throws SAXException {
    Statement stat;
    ResultSet result;
    String data_nascita;
    String query;
    String command;

    if (qualifiedName.equalsIgnoreCase("ID-parco")) {
        ID_parco = Integer.parseInt(text);
    }
    else if (qualifiedName.equalsIgnoreCase("ID-specie")) {
        ID_specie = Integer.parseInt(text);
    }
    else if (qualifiedName.equalsIgnoreCase("ID-esemplare")) {
        ID_esemplare = Integer.parseInt(text);
    }
    else if (qualifiedName.equalsIgnoreCase("nome")) {
        nome = text;
    }
    else if (qualifiedName.equalsIgnoreCase("mese-nascita")) {
        mese_nascita = Integer.parseInt(text);
    }
    else if (qualifiedName.equalsIgnoreCase("anno-nascita")) {
        anno_nascita = Integer.parseInt(text);
    }
    else if (qualifiedName.equalsIgnoreCase("sesso")) {
        sesso = text;

```



```

}
else if (qualifiedName.equalsIgnoreCase("salute")) {
    salute = text;
}
else if (qualifiedName.equalsIgnoreCase("esemplare")) {
    try {
        stat = con.createStatement();
        query = "SELECT COUNT(*) AS numero FROM EsempireFauna WHERE
                ID_esemplare = " + ID_esemplare + " AND ID_parco = " +
                ID_parco + ";";
        result = stat.executeQuery(query);
        result.next();
        if (result.getInt("numero") > 0) {
            // esemplare esistente (aggiornamento record)
            data_nascita = anno_nascita + "-" + mese_nascita + "-1";
            command = "UPDATE EsempireFauna SET Id_specie=" + ID_specie +
                    ", Nome='" + nome + "', DataNascita='" + data_nascita +
                    "', Sesso='" + sesso + "', Salute='" + salute +
                    "' WHERE ID_esemplare = " + ID_esemplare +
                    " AND ID_parco = " + ID_parco + ";";
            if (stat.executeUpdate(command) != 0)
                updated++;
            result.close();
        }
        else {
            // nuovo esemplare (inserimento record)
            data_nascita = anno_nascita + "-" + mese_nascita + "-1";
            command = "INSERT INTO EsempireFauna(ID_parco,
                    ID_esemplare, ID_specie, nome, DataNascita,
                    Sesso, Salute) VALUES (" + ID_parco + ", " +
                    ID_esemplare + ", " + ID_specie + ", '" + nome +
                    "', '" + data_nascita + "', '" + sesso + "', '" +
                    salute + "')";
            if (stat.executeUpdate(command) != 0)
                inserted++;
        }
        stat.close();
    }
    catch (SQLException exception) {
    }
}
else if (qualifiedName.equalsIgnoreCase("fauna")) {
    System.out.println(inserted + " nuovi esemplari inseriti e " +
        updated + " esemplari aggiornati.");
}
}

```



```

// metodi di gestione degli errori SAX
public void warning(SAXParseException exception) {
    System.out.println(exception.getMessage());
}
public void error(SAXParseException exception) {
    System.out.println(exception.getMessage());
}
public void fatalError(SAXParseException exception) {
    System.out.println(exception.getMessage());
}

public static void main(String[] args) {
    try {
        Import importer = new Import();
        importer.parseDocument(args[0]);
    }
    catch (SAXException | ParserConfigurationException | IOException |
           SQLException exception) {
        System.out.println("Errore!");
    }
}
}

```

La seguente classe Java *Export* permette – mediante il metodo *printToFile* – di esportare dal database in un file XML il report periodico da inviare al Ministero dell’Ambiente relativo agli esemplari della fauna dei parchi. La creazione del file XML avviene a partire dalla costruzione dell’albero DOM, mentre per l’accesso al DBMS si utilizzano le API JDBC; a questo scopo si è ipotizzato un server My-SQL in esecuzione sullo stesso computer dell’applicazione e che l’accesso al database «Parco» venga effettuato come utente «root» privo di password.

```

import java.io.*;
import java.sql.*;
import javax.xml.parsers.*;
import javax.xml.transform.*;
import javax.xml.transform.dom.*;
import javax.xml.transform.stream.*;
import org.w3c.dom.*;

public class Export {
    Document document;
    private Connection con;
    final String URL = "jdbc:mysql://localhost:3306";
    final String database = "Parco";

```



```

final String user = "root";
final String password = "";

public Export() throws ParserConfigurationException, SQLException {
    // connessione al server DB
    con = DriverManager.getConnection(URL+"/"+database, user, password);
    // creazione dell'albero DOM del documento XML
    createDOMTree();
}

private void createDOMTree() throws ParserConfigurationException, SQLException {
    Statement stat;
    ResultSet result;
    int numero;
    int ID_parco;
    String parco;
    String regione;
    int ID_specie;
    String specie;
    int ID_esemplare;
    String nome;
    Date data_nascita;
    int mese_nascita;
    int anno_nascita;
    String sesso;
    String salute;
    Element element;
    DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
    DocumentBuilder builder = factory.newDocumentBuilder();

    // creazione elemento root albero DOM
    document = builder.newDocument();
    Element root = document.createElement("fauna");
    document.appendChild(root);

    // interrogazione del database
    stat = con.createStatement();
    String query = "SELECT Parco.ID_parco, Parco, Regione, ID_esemplare,
                    SpecieFauna.ID_specie, Nome, DataNascita, Sesso, Salute,
                    Specie FROM Parco, EsempireFauna, SpecieFauna
                    WHERE Parco.ID_parco = EsempireFauna.ID_parco AND
                    EsempireFauna.ID_specie = SpecieFauna.ID_specie;";
    result = stat.executeQuery(query);
    numero = 0;
    while (result.next()) { // iterazione risultati interrogazione DB
        ID_parco = result.getInt("ID_parco");

```

```

parco = result.getString("Parco");
regione = result.getString("Regione");
ID_esemplare = result.getInt("ID_esemplare");
ID_specie = result.getInt("ID_specie");
nome = result.getString("Nome");
data_nascita = result.getDate("DataNascita");
mese_nascita = data_nascita.getMonth() + 1;
anno_nascita = data_nascita.getYear() + 1900;
sesso = result.getString("Sesso");
salute = result.getString("Salute");
specie = result.getString("Specie");
// creazione elemento XML "esemplare"
element = createEsemplareElement(ID_parco, parco, regione, ID_esemplare,
                                ID_specie, specie, nome, mese_nascita,
                                anno_nascita, sesso, salute);

root.appendChild(element);
numero++;
}
System.out.println(numero + " esemplari esportati.");
result.close();
stat.close();
}

```

```

private Element createEsemplareElement(int ID_parco, String parco,
                                       String regione, int ID_esemplare,
                                       int ID_specie, String specie, String nome,
                                       int mese_nascita, int anno_nascita,
                                       String sesso, String salute) {

Text text;
Element esemplare = document.createElement("esemplare");

Element _ID_parco = document.createElement("ID-parco");
text = document.createTextNode(Integer.toString(ID_parco));
_ID_parco.appendChild(text);
esemplare.appendChild(_ID_parco);
Element _parco = document.createElement("parco");
text = document.createTextNode(parco);
_parco.appendChild(text);
esemplare.appendChild(_parco);
Element _regione = document.createElement("regione");
text = document.createTextNode(regione);
_regione.appendChild(text);
esemplare.appendChild(_regione);
Element _ID_specie = document.createElement("ID-specie");
text = document.createTextNode(Integer.toString(ID_specie));
_ID_specie.appendChild(text);

```

```

    esemplare.appendChild(_ID_specie);
    Element _specie = document.createElement("specie");
    text = document.createTextNode(specie);
    _specie.appendChild(text);
    esemplare.appendChild(_specie);
    Element _ID_esemplare = document.createElement("ID-esemplare");
    text = document.createTextNode(Integer.toString(ID_esemplare));
    _ID_esemplare.appendChild(text);
    esemplare.appendChild(_ID_esemplare);
    Element _nome = document.createElement("nome");
    text = document.createTextNode(nome);
    _nome.appendChild(text);
    esemplare.appendChild(_nome);
    Element _mese_nascita = document.createElement("mese_nascita");
    text = document.createTextNode(Integer.toString(mese_nascita));
    _mese_nascita.appendChild(text);
    esemplare.appendChild(_mese_nascita);
    Element _anno_nascita = document.createElement("anno_nascita");
    text = document.createTextNode(Integer.toString(anno_nascita));
    _anno_nascita.appendChild(text);
    esemplare.appendChild(_anno_nascita);
    Element _sesso = document.createElement("sesso");
    text = document.createTextNode(sesso);
    _sesso.appendChild(text);
    esemplare.appendChild(_sesso);
    Element _salute = document.createElement("salute");
    text = document.createTextNode(salute);
    _salute.appendChild(text);
    esemplare.appendChild(_salute);

    return esemplare;
}

private void printToFile(String XMLfile) throws TransformerException {
    TransformerFactory factory;
    factory = TransformerFactory.newInstance();
    Transformer transformer = factory.newTransformer();
    DOMSource source = new DOMSource(document);
    StreamResult result = new StreamResult(new File(XMLfile));

    transformer.transform(source, result);
}

public static void main(String args[]) {
    Export exporter;

```



```
try {
    exporter = new Export();
    exporter.printToFile(args[0]);
}
catch (ParserConfigurationException | TransformerException | SQLException
    exception) {
    System.out.println("Errore!");
}
}
```