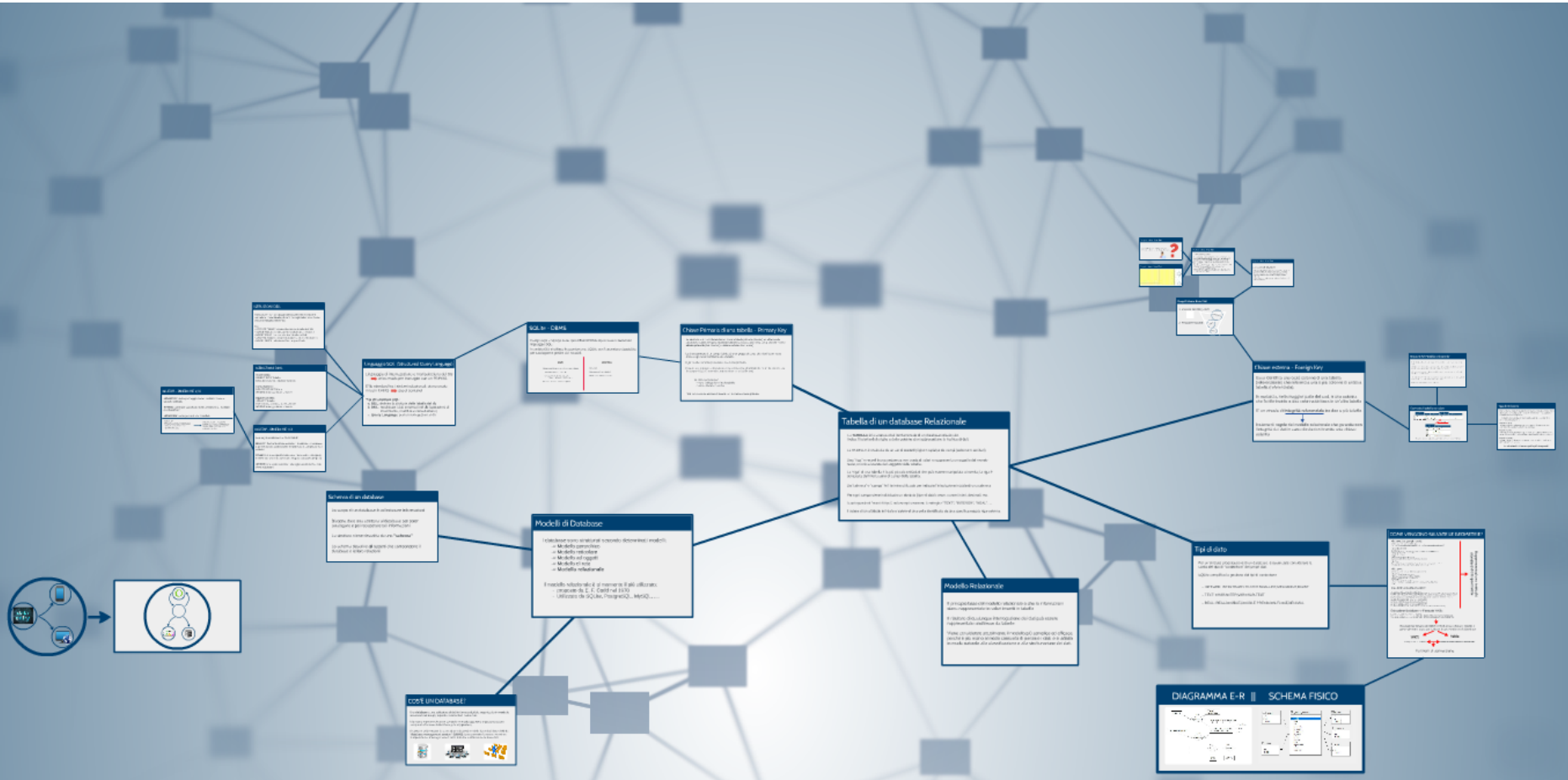


DATABASE RELAZIONALI

VENEZIA 16/02/2020

Paolo Tomè

COS'È UN DATABASE?

Una **database** è una collezione di dati in formato digitale, organizzata in modo da preservarli nel tempo, reperirli e condividerli facilmente.

I dati sono memorizzati su un computer in modo opportuno e possono essere recuperati attraverso delle interrogazioni (**queries**)

Il software utilizzato per la costruzione e la gestione della base dati viene definito **"database management system" (DBMS)**. Esso permette la memorizzazione, manipolazione, interrogazione di tutti i dati che costituiscono la base dati.

Schema di un database

Lo scopo di un database è collezionare informazioni

Bisogna dare una struttura al database per poter catalogare e poi recuperare tali informazioni

La struttura viene descritta da uno **"schema"**

Lo schema descrive gli oggetti che compongono il database e le loro relazioni

Modelli di Database

I database sono strutturati secondo determinati modelli:

-> Modello gerarchico
-> Modello reticolare
-> Modello ad oggetti
-> Modello di rete
-> **Modello relazionale**

Il modello relazionale è al momento il più utilizzato:

- proposto da E. F. Codd nel 1970
- Utilizzato da SQLite, PostgreSQL, MySQL, ...

Modello Relazionale

Il principio base del modello relazionale è che le informazioni siano rappresentate in valori inseriti in tabelle

Il risultato di qualunque interrogazione dei dati può essere rappresentato anch'esso da tabelle

Viene considerato attualmente il modello più semplice ed efficace, perché è più vicino al modo consueto di pensare i dati, e si adatta in modo naturale alla classificazione e alla strutturazione dei dati.

Tabella di un database Relazionale

La **TABELLA** è la struttura dati fondamentale di un database relazionale.
Indica l'insieme delle righe e delle colonne che rappresentano la matrice di dati.

La TABELLA è costituita da un set di **record** (righe o tuple) e da campi (colonne o attributi):

Una “riga” o record è una sequenza non vuota di valori e rappresenta un oggetto del mondo reale, un’unica istanza del soggetto della tabella.

La “riga” di una tabella è la più piccola unità dati che può essere manipolata o inserita; la riga è composta dall’intera serie di campi della tabella.

Un “colonna” o "campo" è il termine utilizzato per indicare l'intestazione iniziale di una colonna

Per ogni campo viene individuato un dominio (tipo di dato): testo, numeri interi, decimali, etc.

I corrispondenti "nomi di tipo", ad esempio saranno le stringhe "TEXT", "INTEGER", "REAL", ...

Il valore di un attributo è il dato o valore di una cella identificata da una specifica coppia riga-colonna

Chiave Primaria di una tabella - Primary Key

Le relazioni, o vincoli interelazionali, tra una tabella primaria (master) ed altre tabelle secondarie (slave) vengono realizzate attraverso l'associazione tra campi che definiremo **chiave primaria** (tab. master) o **chiave esterna** (tab. slave)

La chiave primaria è un campo (attributo) o un gruppo di campi che identifica in modo univoco ogni record all'interno della tabella.

Ogni tabella dovrebbe possedere una chiave primaria.

Quando non è possibile trovare un campo chiave tra gli attributi di una entità, si definisce un campo di tipo ID numerico autoincrementante (contatore)

La chiavi servono per:
-> fare "collegamenti" tra le tabelle
-> velocizzare le ricerche

N.B non possono esistere 2 record con la stessa chiave primaria

Chiave esterna - Foreign Key

Essa identifica una o più colonne di una tabella (referenziante) che referenzia una o più colonne di un'altra tabella (referenziata).

In sostanza, nella maggior parte dei casi, è una colonna che fa riferimento a una colonna univoca in un'altra tabella

E' un vincolo di **integrità referenziale** tra due o più tabelle.

Insieme di regole del modello relazionale che garantiscono l'integrità dei dati in caso di relazioni tramite una chiave esterna

Esempio di tabelle correlate

Campo Numerico Chiave Primaria	Campo Data	Campo Testo	Chiave Esterna	Campo Numerico	Campo Testo
id_scheda_generale	data_verifica	codice_segnalazione	id_specie	numero	habitat_100metri
4	17/02/10 11.29	CR-Canis_lupus_170210	13	1	strada larga, nuova e diritta che
6	21/03/10 15.46	CR-Rattus_rattus_21031(	1	1	prato - giardino con ristagno di acqua
7	29/03/10 23.41	FD-Coypus-01	96	1	Area di risorgiva nel Piave con

Record di dati 94.413 da 94.413

RELAZIONE (1:n)

Campo Numerico Chiave Primaria	Campo Testo
id_specie	specie
5	Cervo Cervus elaphus
6	Cinghiale Sus scrofa
7	Daino Dama dama
8	Muflone Ovis orientalis musimon
9	Stambecco Capra ibex
10	Sciacallo dorato Canis aureus
13	Canis lupus
14	Canis lupus familiaris

Record di dati 1 da 41 *

Relazione creata per evitare ridondanze di dati, errori di inserimento dei dati, ...

Tipo di Relazioni

Una relazione dunque indica una interazione tra entità ed è caratterizzata da un grado che indica quante istanze dell'entità di arrivo si associano all'istanza dell'entità di partenza

Il grado può essere **a uno** oppure **a molti** e pertanto le relazioni tra due entità si classificano in:

Relazione (1:1)
Ad ogni elemento del primo insieme E1 corrisponde uno ed un solo elemento del secondo insieme E2, e viceversa

Relazione (1:N)
Ad un elemento di E1 possono corrispondere più elementi di E2 mentre ad ogni elemento di E2 deve corrispondere uno ed un solo elmento di E1

Relazione (N:M)
Ad ogni elemento dell'insieme E1 possono corrispondere più elementi dell'insieme E2 e viceversa.

Le relazioni 1:N sono quelle più frequenti

Requisiti del Modello relazionale

1. Tutte le righe della tabella contengono lo stesso numero di colonne, corrispondente agli attributi

2. I valori assunti da un campo appartengono al dominio dei valori possibili per quel campo, e quindi sono valori omogenei tra loro, cioè sono dello stesso tipo

3. In una tabella, ogni riga è diversa da tutte le altre, cioè non ci possono essere due righe con gli stessi valori dei campi

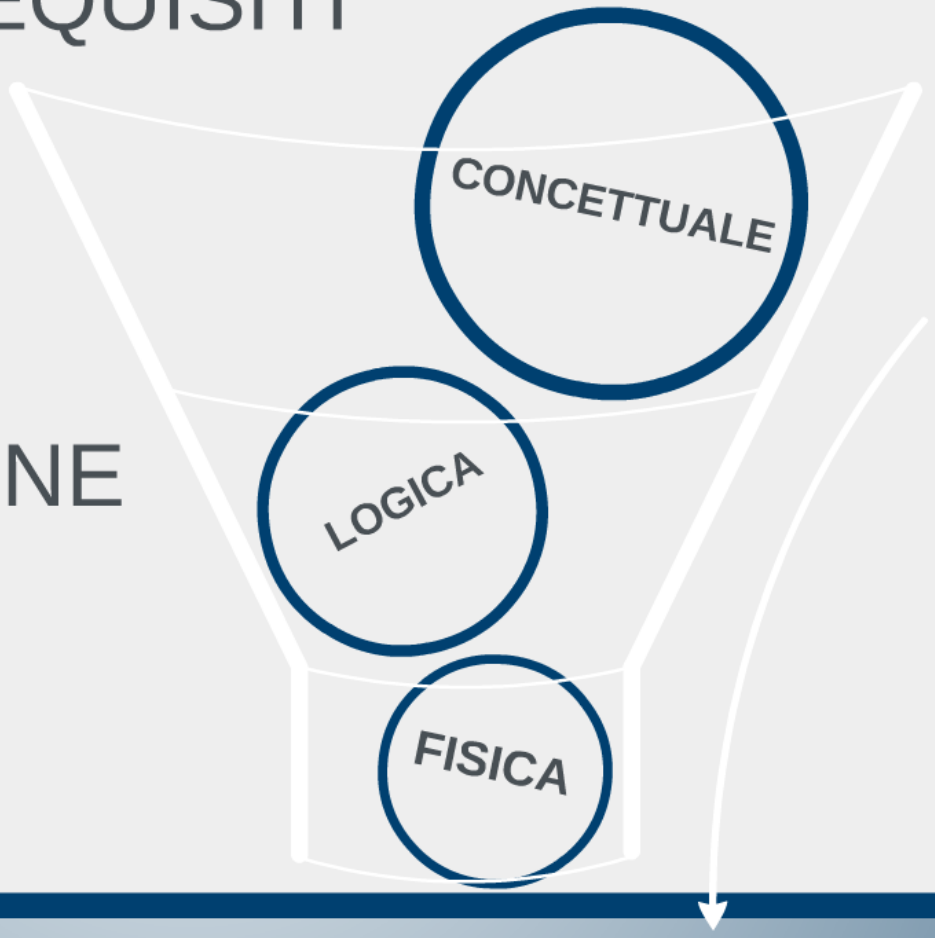
4. Le righe compaiono nella tabella secondo un ordine non prefissato, cioè non è rilevante il criterio con il quale le righe sono sistemate nella tabella

5. Nessuna chiave primaria può avere valore nullo

Progettazione Base Dati

1. ANALISI DEI REQUISITI

2. PROGETTAZIONE



Progettazione Base Dati

1. ANALISI DEI REQUISITI

Consiste nella individuazione e nello studio delle proprietà e delle funzionalità che il sistema informativo dovrà avere.
Questa fase richiede una interazione con gli utenti del sistema e produce una descrizione completa dei dati coinvolti.
Vengono inoltre stabiliti i requisiti software e hardware del sistema informativo.

Progettazione Base Dati

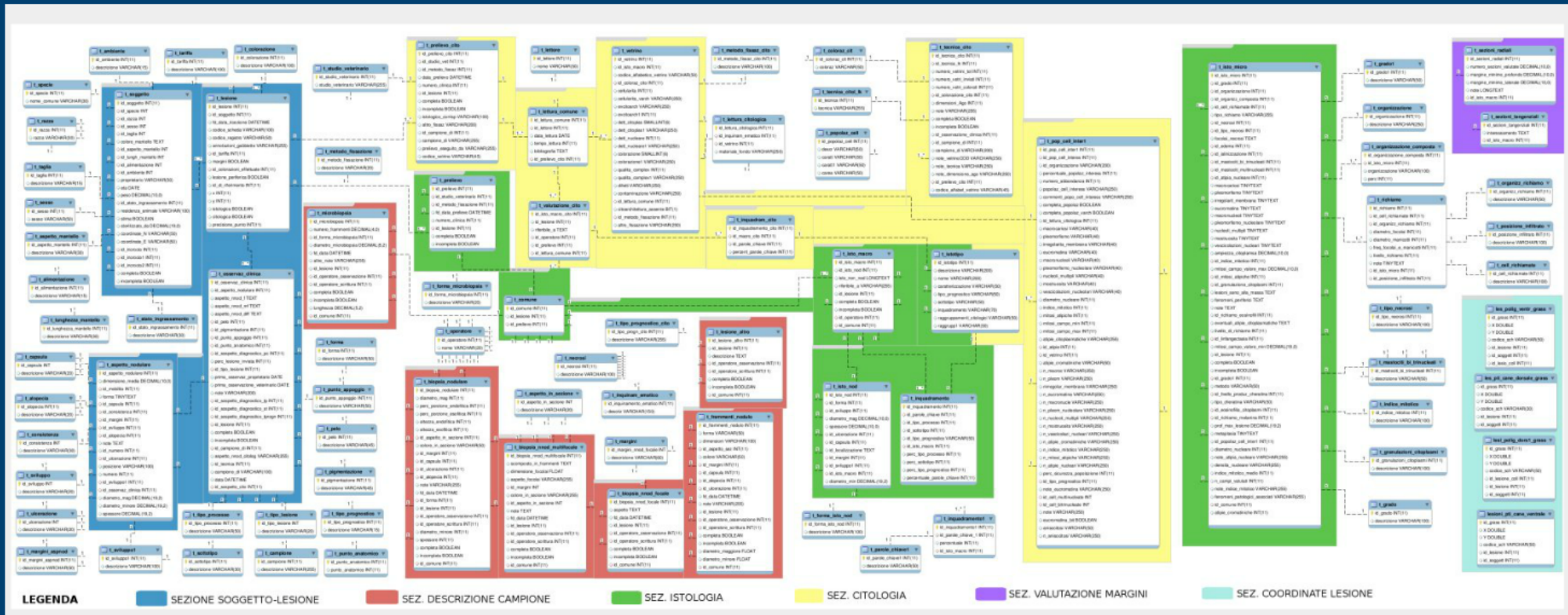
2. PROGETTAZIONE

Le fasi di progettazione partono dalla descrizione completa della realtà di interesse che si traduce nella realizzazione di un modello concettuale (**MODELLO E-R**) che descrive le ENTITÀ (classi di oggetti aventi proprietà comuni) e le loro RELAZIONI.
In seguito lo schema concettuale viene tradotto nel modello di rappresentazione dei dati adottato dal DBMS scelto

-> creazione tabelle e relazioni

Infine lo schema logico viene integrato con l'organizzazione dei file, a cura del **DBMS**

Progettazione Base Dati



Progettazione Base Dati

Come realizzare una semplice base dati per l'inserimento delle rilevazioni di segni di presenza

SQLite - DBMS

PostgreSQL e MySQL sono i più diffusi RDBMS Open Source basati sul linguaggio SQL.

In ambito GIS si utilizza frequentemente SQLite con l'estensione SpatiaLite, per catalogare e gestire dati spaziali.

PRO

Nessuna installazione o configurazione -> libreria

Il db è strutturato in un unico file

Il db è accessibile da molti software:
- QGIS, LibreoOffice, DB Browser

Facile da imparare, utilizzare e gestire

CONTRO

Db < 2GB

Nessuna gestione degli utenti

Non fornisce accesso alla rete

eSQL e MySQL sono i più diffusi RDBMS Open Source basati su
gio SQL.

to GIS si utilizza frequentemente SQLite con l'estensione Spatia
alogare e gestire dati spaziali.

PRO

Nessuna installazione o configurazione -> libreria

Il db è strutturato in un unico file

Il db è accessibile da molti software:
- QGIS, LibreoOffice, DB Browser

Facile da imparare, utilizzare e gestire

CONTRO

Db < 2GB

Nessuna gestione degli utenti

Non fornisce accesso alla rete

Linguaggio SQL (Structured Query Language)

Linguaggio di interrogazione e manipolazione del DB
➡ unico modo per interagire con un RDBMS

E' lo standard tra i sistemi relazionali: viene usato in tutti i DBMS ➡ set di comandi

Tipi di istruzioni SQL:

a. **DDL**: definire la struttura delle tabelle del db
b. **DML**: modificare i dati contenuti nel db (operazioni di inserimento, modifica e cancellazione)
c. **Query Language**: porre interrogazioni al db

ISTRUZIONI DDL

Il Data Definition Language (DDL) permette di creare e cancellare intere tabelle, di definire degli indici, specificare vincoli e integrità referenziali

Es.:
- CREATE TABLE : creare una nuova tabella nel DB
- ALTER TABLE : modificare la struttura di una tabella
- DROP TABLE : cancellare una tabella dal DB
- CREATE INDEX : creare un indice su una certa tabella
- DROP INDEX : eliminare l'indice specificato

ISTRUZIONI DML

Inserimento:
INSERT INTO Tabella
VALUES (valore1, valore2, valore3)

Cancellazione:
DELETE FROM Tabella
WHERE nome_colonna = valore

Aggiornamento:
UPDATE Tabella
SET nome_colonna = nuovo_valore
WHERE nome_colonna = valore

QUERY - RICERCHE 1/4

La query è suddivisa in 6 CLAUSOLE:

SELECT: Permette di fare estrazioni di dati da un database a partire da una o più tabelle; è il più ricco e complesso tra i comandi

FROM: Definisce quali tabelle sono interessate nella query ed eventualmente in che modo vengono concatenate (join)

WHERE: una o piu' condizioni che ogni record del risultato deve soddisfare

QUERY - RICERCHE 2/4

GROUP BY: serve per raggruppare i risultati in base a qualche attributo

HAVING: serve per specificare delle condizioni sul risultato di GROUP BY

ORDER BY: serve per ordinare il risultato

```
SELECT *
FROM t_scheda_monitoraggio
WHERE id_specie =  13
ORDER BY data;
```

Seleziona tutti i campi dalla tabella t_scheda_monitoraggio aventi come specie il lupo, ordinati per data

Tipi di dato

Per un'efficace progettazione di un database, è essenziale considerare la scelta del tipo di "contenitore" dei propri dati.

SQLite semplifica la gestione dei tipi di contenitore

-- INTEGER: INT,INTEGER,TINYINT,SMALLINT,MEDIUMINT,BIGINT

-- TEXT: CHARACTER,VARCHAR,TEXT

-- REAL: REAL,DOUBLE,DOUBLE PRECISION,FLOAT,DECIMAL

COME VENGONO SALVATE LE GEOMETRIE?

File .KML (es. google earth):

```
<?xml version="1.0" encoding="UTF-8" ?>
<kml xmlns="http://earth.google.com/kml/2.2" xmlns:gx="http://www.google.com/kml/ext/2.2">
..............
<name>Waypoints</name>
<Placemark>
<name><![CDATA[av - 2 capreolus adulti - POI 18/nov/2014 17:38:45]]></name>
<description><![CDATA[]]></description>
<styleUrl>#1</styleUrl>
<Point>
<altitudeMode>absolute</altitudeMode>
<coordinates>11.7765120,45.8023760,0.00</coordinates>
</Point>
</Placemark>
<Polygon><coordinates>12.762782203464761,45.803690182879663 12.761656354859483,45.803183912535168 12.761472219800646,45.803785846418748
12.762756476631342,45.804322268977391 12.762870475594648,45.80425694556847б 12.762916356934033,45.803960109128042
12.762885554316781,45.803766701096642 12.762782203464761,45.803690182879663</coordinates></Polygon>
```

File .GPX:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<gpx xmlns="......">
<name><![CDATA[Waypoints]]></name>
<time>2015-10-24T22:37:37Z</time></metadata>
<wpt lat="45.8023760" lon="11.7765120">
<ele>0.00</ele>
<time>2014-11-18T16:39:10Z</time>
<name><![CDATA[av - 2 capreolus adulti - POI 18/nov/2014 17:38:45]]></name>
</wpt>
```

File .SHP -> FORMATO WKT:

wkt_geom id_scheda_ tipo_event tipo_even0 specie
POINT(12.37470900280418462 45.03362822499269669) 7779 Cattura con trappole Cattura con trappole Roditori
POINT(12.37571751339294224 45.0368808014505646) 7780 Cattura con trappole Cattura con trappole Roditori

POLYGON((1792417.31110000051558018 5079144.51409999933093786,1792332.48090000124648213
5079084.14419999998062849,1792315.02110000094398856
5079150.34380000084638596,1792412.00110000092536211
5079214.64379999786615372,1792421.20120000024326146
5079207.80399999674409628,1792426.32119999988935888
5079174.99400000367313623,1792424.94090000120922923
5079153.39379999879747629,1792417.31110000051558018 5079144.51409999933093786))

Rappresentazione testuale standard delle geometrie

Estrazione Database -> Formato WKB:

id_scheda_ tipo_event tipo_even0 specie **the_geom**
7779 Cattura con trappole Cattura con trappole Roditori, **0106000020E6100000010000000103000000100....**
7780 Cattura con trappole Cattura con trappole Roditori, **40FAD42B40050000C08A72245400B000040FAD4**...........

Equivalente binario del WKT, il Well-Known Binary (WKB) è generalmente usato per salvare le geometrie in un database

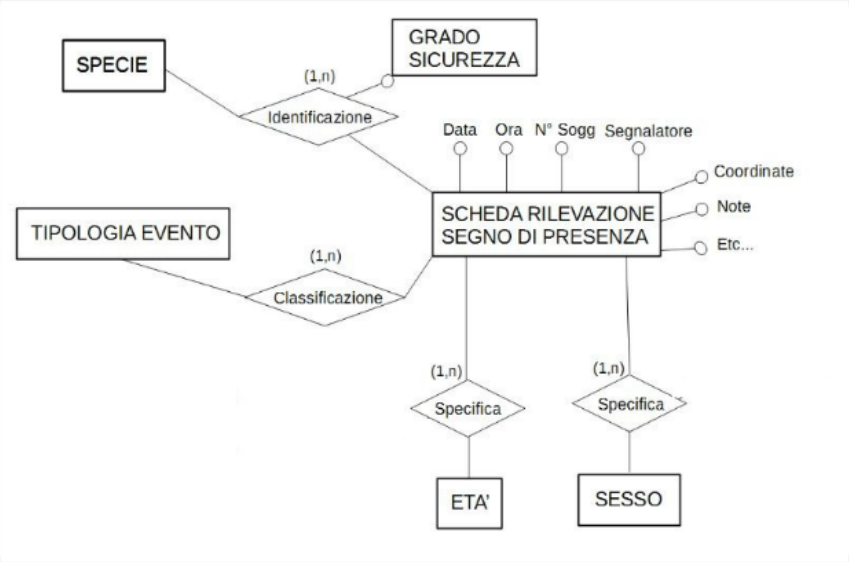
WKT: POINT(45.685811, 12.290501)

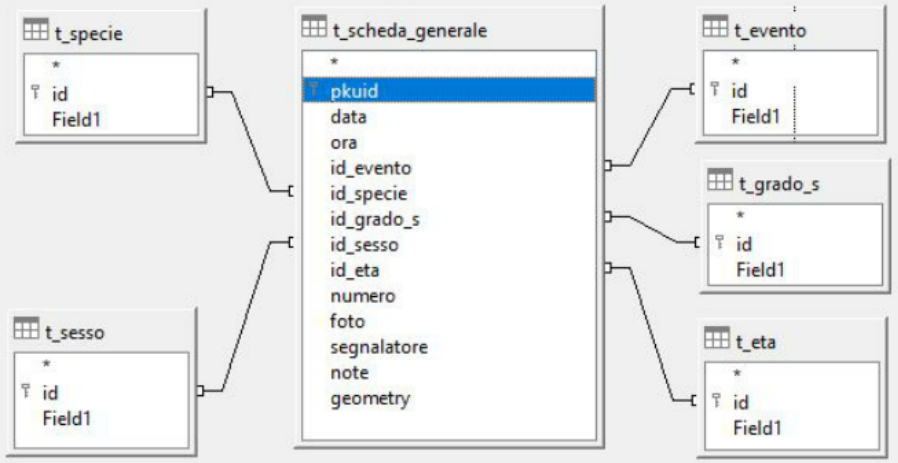
WKB: 000000000140000000000000004010000000000000

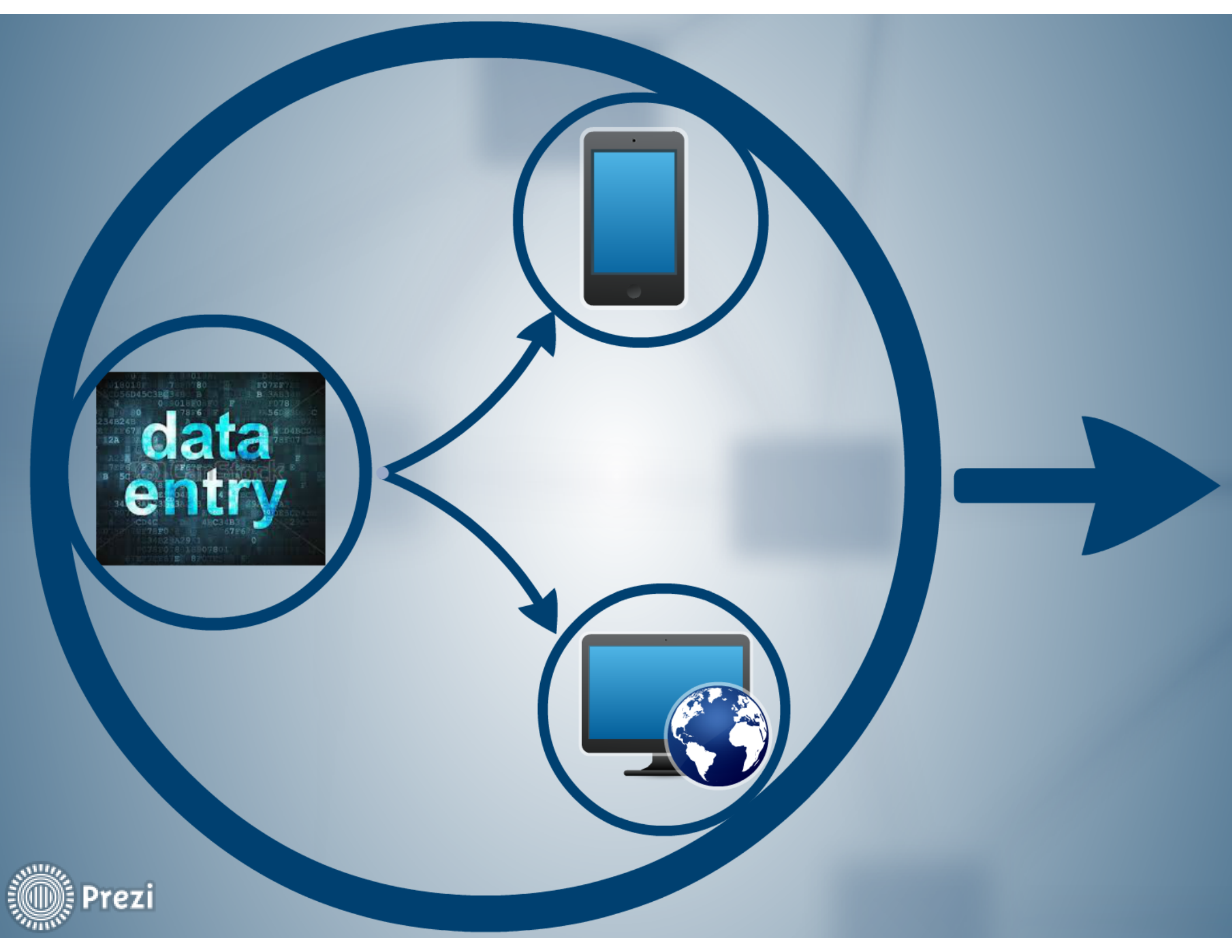
Funzioni di conversione

Prezi

DIAGRAMMA E-R || SCHEMA FISICO






data
entry