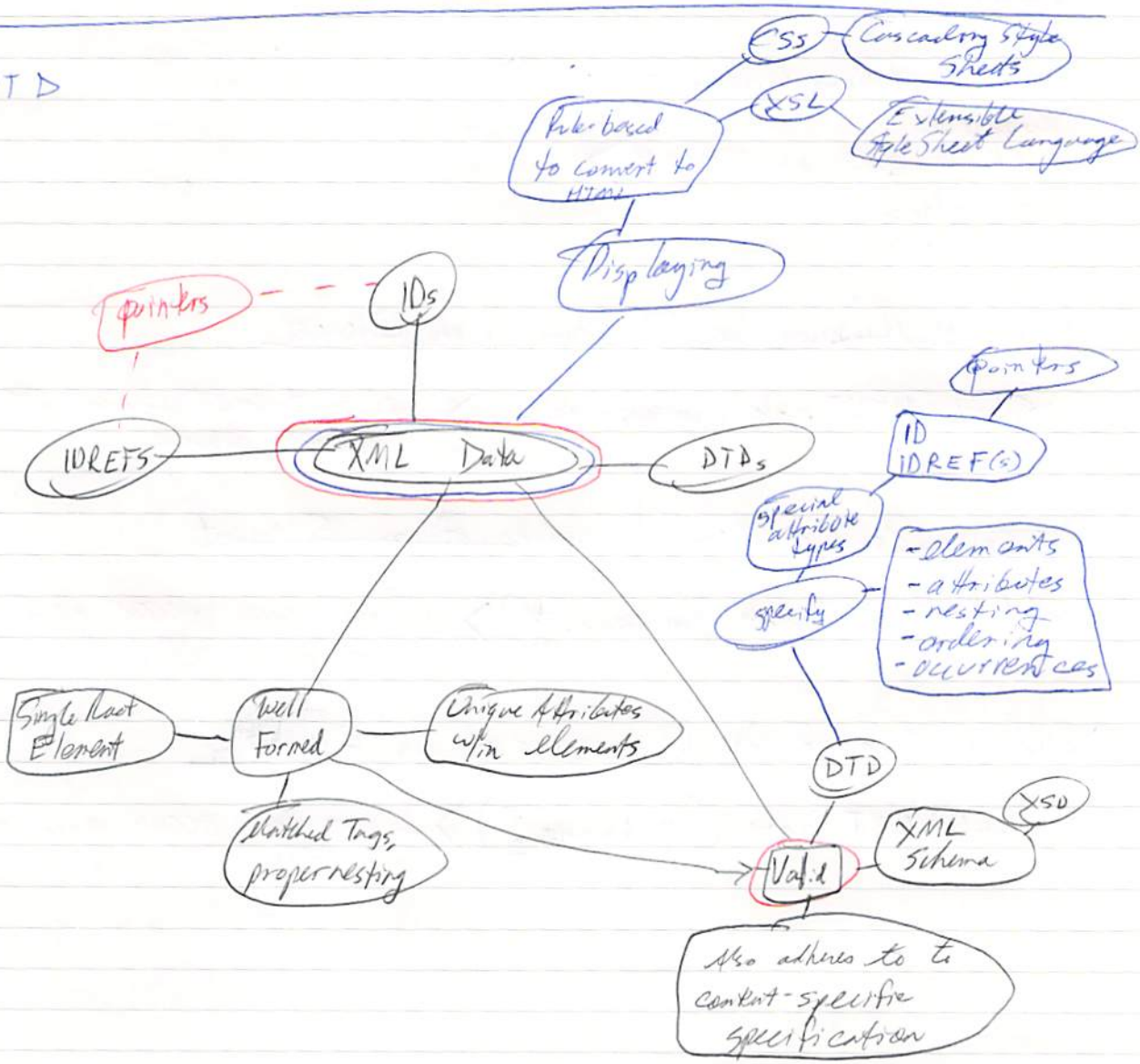
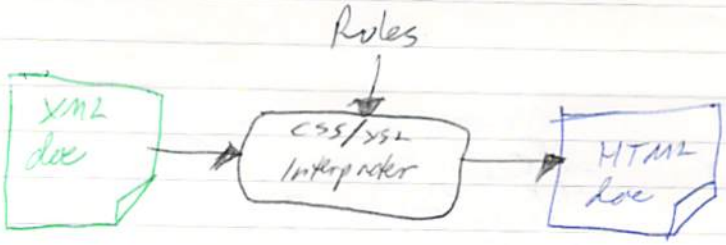
Question 3:

Find all pizzerias that serve only pizzas eaten by people over 30

TBC

DTD





W3schools.com XML tutorial

<?xml version="1.0" encoding="ISO-8859-1"?>

↑ XML Version — ↑ encoding used

XML elements must have a closing tag
XML tags are case sensitive
XML elements must be properly nested
XML documents must have a root element
XML attributes must be quoted ①
Use entity references for special chars < = <
White Space is preserved in XML
XML stores new line as LF

① Use attributes for meta data and not much more

Displaying Your XML files with CSS

<?xml-stylesheet type="text/css" href="cd_catalog.css"?>
↳ link to CSS file to use
W3C recommends using XLST to format XML
- eXtensible Stylesheet Language Transformations
- more sophisticated than CSS

Declaring Attributes

```
<!ATTLIST payment type CDATA "check" >
```

ex: `<payment type="check" />`

Attribute type possibilities:

Type:	Description
CDATA	The value is character data
(en1|en2|..)	from an enumerated list
ID	unique id
IDREF	id of another element
IDREFS	list of other ids
NMTOKEN	valid XML token
NMTOKENS	valid XML tokens
ENTITY	entity
ENTITIES	list of entities
NOTATION	name of an notation
xml:	predefined XML value

10.16.2011

W3Schools Intro to DTD

~~The~~ defines legal building blocks of an ~~XML~~ document.

Defines the document structure with a list of legal elements and attributes.

<!DOCTYPE root-element [element-declarations]>

```
<?xml version="1.0"?>
<!DOCTYPE note [                    defines root element
<!ELEMENT note (to, from, heading, body)>   defines child elements of note element
<!ELEMENT to (#PCDATA)>             defines element to be of type #PCDATA
<!ELEMENT from (#PCDATA)>           "
<!ELEMENT heading (#PCDATA)>        "
<!ELEMENT body (#PCDATA)>           "
]>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend</body>
</note>
```

<!DOCTYPE root-element SYSTEM "filename">

DTD defined in external file

Building Blocks of XML Data

From a DTD's perspective, all XML docs are made up of the following building blocks

- Elements
- Attributes
- Entities
- PCDATA
- CDATA

parsed character data → text between start and end tag character data. text will not be parsed Tags inside the text will Not be treated as markup and entities will not be expanded.

cont'd

Cont'd →

Declaring Elements

<!ELEMENT element-name category>

or

<!ELEMENT element-name (element-content)>

Empty Elements

<!Element element-name EMPTY>

example:
<!ELEMENT br EMPTY>

XML element:

Declaring ~~Minimum~~ One Occurrence of an Element

<!ELEMENT note (message+)> element must occur at least once

Declaring Zero or More Occurrences of an Element

<!ELEMENT note (message*)> element can occur zero or more times

Declaring Zero or One Occurrences of an Element

<!ELEMENT note (message?)> element can occur zero or one time