Date
Page

1.> But the update u may want to do is to modify takes

Create views ACGPA As (Select Any CGPA from student),
|| Update CGPA* = 1.1;
Updates on views is generally useless, there are keys

11/10/19 ~~XML ex~~ Lec-12

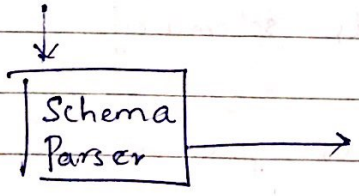
XML & XQuery → Semi-structured Data
XML : a complicated JSON
Size of each object won't be same

<Rak, abc@t.com 99-001 >

Metadata is needed to process this.

Html was popular.
So user defined tags were thought of.

XML Schema
(, ,)

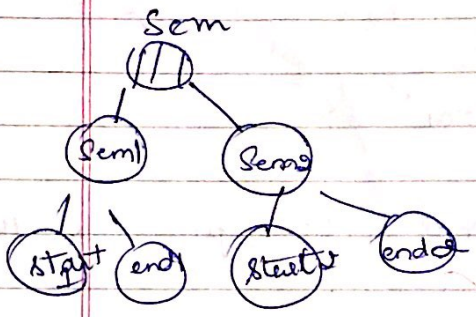


Interoperability : <price>1828</price>
↓ ↓
Ability to decipher and apply constraints. To decipher this a ~~price~~ is present.

1.8 MB is the avg XML document size if you order a prod from IBM.

→ facilitates user-driven for consumption

Characteristics of XML:
User defined tags
Models - Unstructured Data and Rlnal Database
Based on tree structures
Non leaf → Element node
Leaf node → Content containing Nodes



Notes:
Not exactly for storage
But for displaying dynamic web pages

Semistructured data: Needs schema info
[self-describing data
Html is unstructured, as you can't tell the data type
Schema information

XML Tree Data Model:
Elements and Attributes
↓
Tag's attributes

Complex Simple

XML Docs → Datacentric
→ Document centric : Text files
→ Hybrid XML : Mix

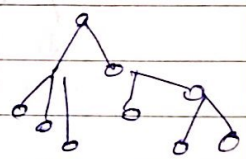
Structured Data ⟺ Schema. (Having a pre-defined)
Semi Structured ⟺ Schemaless

Well formed XML:
1) XML declaration
2) Single root
3) Valid: Have a schema too XML DTD file (Document Type Definition)

20/10/17 Lec-13

In Rel database:
Select
from <context>
where <candidate>

XML



XPath gives the context that lacks in XML

/bookstore/book : Select from root node

//lang → attributes

/bookstore/book[last()]
//As opposed to Relational Db, where no first & Last exist

/title[@lang] → Only those titles that have lang as attrib

//* → All elements

//title[@*]
Traverses the tree and looks for tags

//book[@*]/title All nodes having category & print titles attributes

//XML

//book | title | //book/price

FLOWR Queries

```
for $x in doc("books.xml") /bookstore/bo
where $x /price > 30
order by $x/title
return $x/title | $x/price
```

```
for $x in doc("books.xml")/bookstore/bo
  return if ($x/@category = "CHILDREN")
  then <child> {data($x/title)}. </child>
  else <adult> {data($x/title)} </adult>
```

FLOWR → return
↓ ↓ → where
for let → order by

Use | Need of XML

1> Semistructured Data
2> Kind of Query langs we encounter

24/10/17

Lec-14 Lucene.

→ Unstructured

w_1

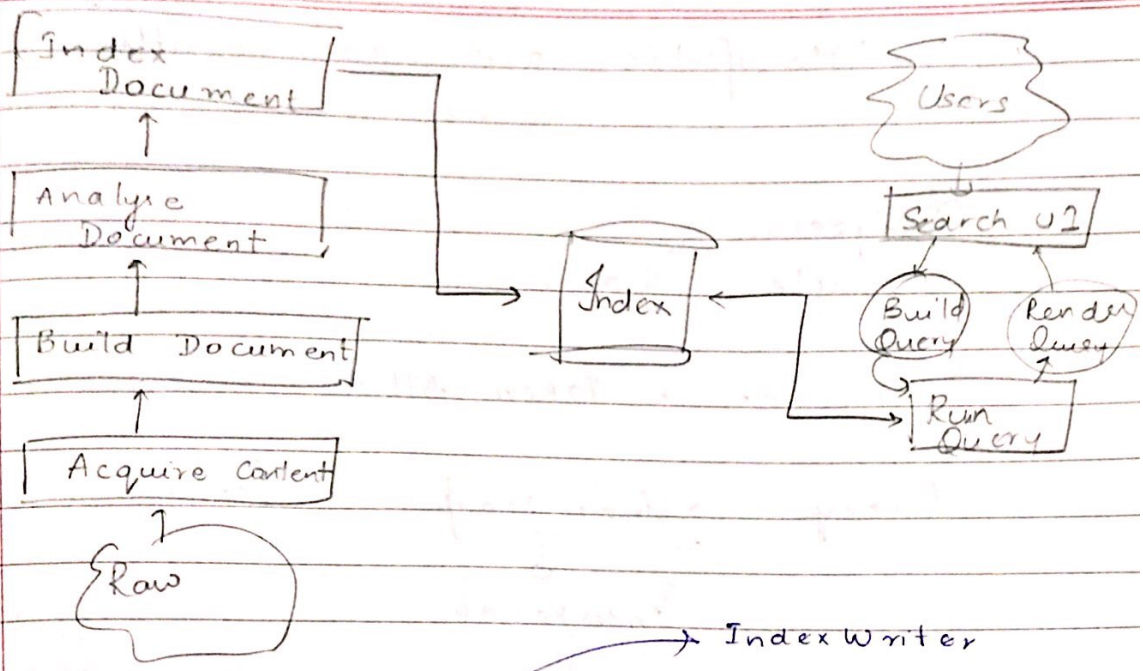
w_1 and w_2

[w_1 or w_2] and word 3

Integrated in many langs

Intro to Info Retrieval

Pandu ←



Core Indexing Classes → IndexWriter, Directory, Analyzer

Core Search classes → IndexSearcher (Exposes several methods of index), Query, QueryParser

A query class Subclasses contain types of queries

TopDocs, ScoreDocs

A Doc is an atomic unit of indexing and searching → Docs have fields

{ name : value }

Raw content needs to be translated into fields eg: Title, author, date, abstract, body.

Different docs have different fields

Search fields using <name>:<term>

eg: title:HP vCC

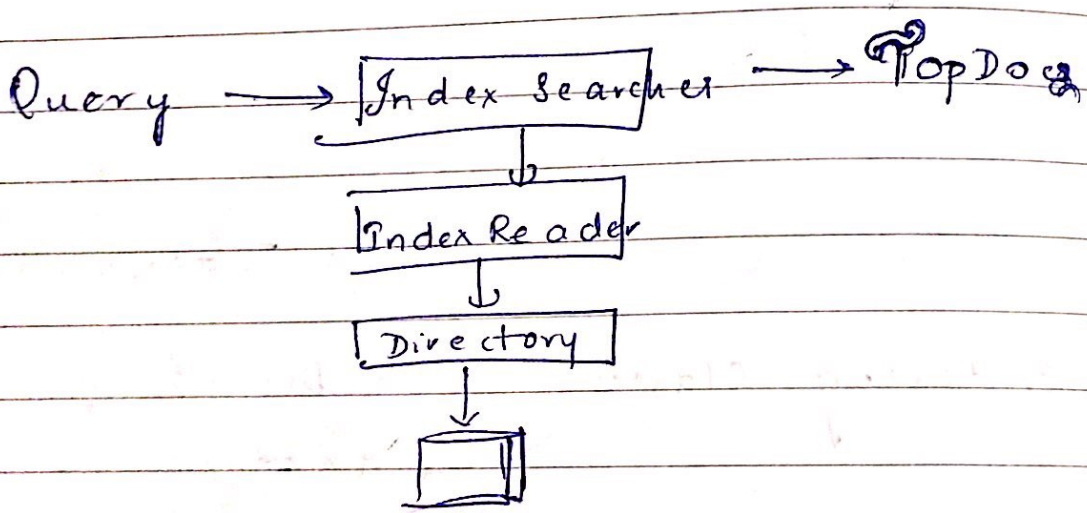
Now, even Multi-media is Queriable

Fields may be indexed/not, stored/not, term vector

multiple fields are also possible.

Analyzers:-
- simple
- Stop - -

Tokenizer & Token filter



27/10/17

Lec-15

NoSql RDBMS

. portable RDBMS.

.

Docs Keys/Values Graph Traversal
Graphs Text Search
Map/Reduce

4 Types of NoSql:
Key/Value
Column
Graph
Document

Wide Column Store:
Basic Data model:
Key, Value, Timestamp

Acid vs Base

NeO4J

Lucene - Notes:
A core is a running instance of a Lucene index that has all Solr config files required
See examples/films for more info

XML - Notes:
XQuery ⊇ XPath

31/10/17

Lec-16

Rollno	Name	Hostel

If all stus living in a hostel leave, Hostel info is lost,

So

R.no	Name	Hostel No	Hostel ID

Avoid Redundancy.

Avoid Null Values by making ~~Tab~~ new relations

Consider

1>	Alba	Veena	Music
2>	B	Odissi	Dance
3>	R	Guitar	Music
4)	S	Foot Ball	Dance
5)	Alba	Odissi	Dance

Divide into 2 rlns

T_1: Topic | Hob

T_2: sid | Name | Top

$T_1 \bowtie T_2$ (Topic$_1$ = Topic$_2$) → we won't be able to retrieve the exact info.

Such extra rows are called Spurious Rows

Joins should be done on KEYS Primary s/ foreign keys

Functional Dependencies

Consider $R_1(A_1, A_2, \ldots A_n)$

Let $X \subseteq (A_1, A_2, \ldots A_n)$

$Y \subseteq (A_1, A_2, \ldots A_n)$

(X, Y are set of attribs)

X determines Y if for any 2 rows of R,

$t_1[X] = t_2[X] \Rightarrow t_2[Y] = t_2[Y]$

It comes from the semantics of the application environment

eg:

SId	Name	Db	CNo	cName
1	A	3/8/1998	101	
2	B	9/8/1998	101	
1	A	5/8/1998	102	
2	B	9/8/1998	102	

SId → DOB

$\Rightarrow$ $SId[R_1] = SId[R_3] \Rightarrow DOB[R_1] = DOB[R_3]$

So, But DOB $\not\rightarrow$ SId

So Non- Commutative..

$X \supseteq Y \Rightarrow X \rightarrow Y$ (Reflexivity)

$X \rightarrow Y \Rightarrow XZ \rightarrow YZ$ (Augmentation)

$X \rightarrow Y, Y \rightarrow Z \Rightarrow X \rightarrow Z$ (Transitivity)

Proof $X \rightarrow Y$ but $XZ \not\rightarrow YZ$

$t_1[XZ] = t_2[XZ]$ but $t_1[YZ] \neq t_2[YZ]$

$t_1[X] = t_2[X]$

$t_1[Z] = t_2[Z]$ → ①

From $X \rightarrow Y \Rightarrow t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y]$ → ②

① & ② $\Rightarrow t_1[Y] = t_2[Y]$ and $t_2[Z] = t_1[Z]$

$\Rightarrow t_1[XYZ] = t_2[YZ]$

$X_1 \rightarrow Y_1, \ldots X_n \rightarrow Y_n$ are basis

then RAT can generate all ~~basis~~ other final dep

Sound →

complete →

Final dependencies → Violation can happen when rows are added

If CNo → CName ⇒ Then Violation can be checked but not Vice-Versa

If additional deps are generated, even then they satisfy

If R(A, C, D, E, H)

A → CD , E → AH

AE → CDE ⟨ AE → AH

AE → CDE ✗ AE → AH ⇒

AE → ACDEH

Additional rules:

⇒ {X → YZ} ⇒ X → Y and X → Z,

{X → Y, X → Z} ⇒ X → YZ

{X → Y, WY → Z} → WX → Z

F(A → C, AC → D, E → AD, E → H).

E → E.ADH

A ⊆ EADH

E^+ → EADHC

If all attributes are determined ⇒ A Key

$H^+ = H$

AE^+ → All attribs

$E^+ = AE^+$

Algo $X^+ = X$

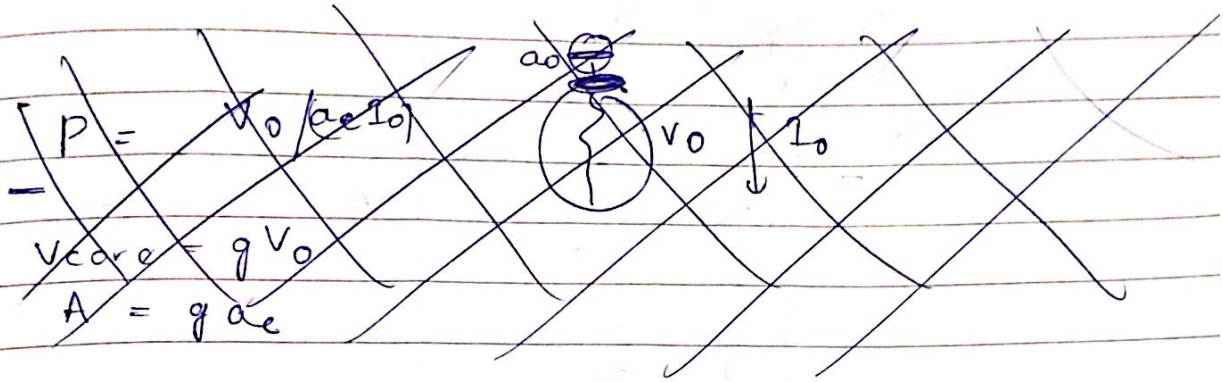
while change to X^+ do

for each new FD Y → Z in F:

if $Y \subset X^+ \Rightarrow X^+ = X^+ \cup Z$

$F^+ = G^+ \iff F^+ \subseteq G^+$ and $G^+ \subseteq F^+$

Even though F, G look different, they are the same



3/11/17

Lec - 17

$A \to C$, $AC \to D$, $E \to AD$, $E \to H$

$A \to CP$, $E \to AH$,

$AC \to D$

$A \to C$, $C \to D$, $E \to A$, $E \to H$ → Minimal Cover

$P(A_1, \ldots, A_n)$

Prime — Belong to any key

Non Prime — Dont Belong to any key

Full dependency

$X \to Y$ if no Z exists so that $Z \subset X$ and $Z \to Y$

Partial dependency

↳ Opposite

Transitive deps:-

1st Normal form:

All attributes are atomic.

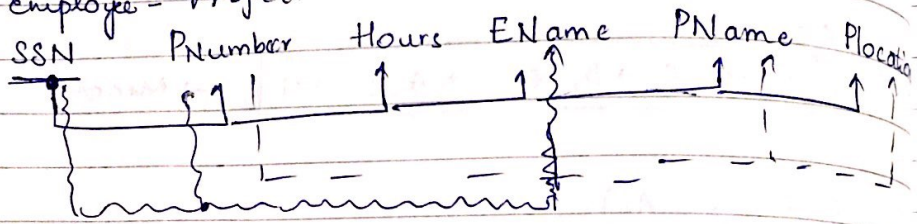
Student Name:	Rollno	Contact No
Akshay	8	181, 125, 161
Amba	2.1	201, 222332

R.no	Contact No
8	181
8	125
8	161
2	261
1	211

Both are Keys

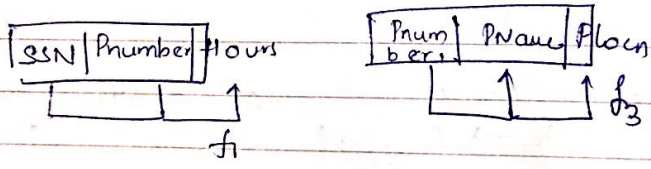
2nd Normal Form = 1st Normal form + All non prime attribute is fully finally dependent on Primary Key

Employee - Project



f_1 : SSN, PNO → PName, Ploc

f_3 : Pnumber → Pname, Ploc



3rd Normal form:

2NF + for all non-trivial deps in F+ are in the form

X → A with either

i) X is a super key

ii) A is a prime Attrib

Boyce - Codd Normal Form:-

3NF + for all non-trivial dependencies in F+ are of the form X → A with X is a Super Key

Lossless Join: No spurious tuples are formed

If D $(R_1 \cap R_2) \rightarrow (R_1 - R_2)$ is in F^+ or
$(R_1 \cap R_2) \rightarrow (R_2 - R_1)$ is in F^+

Tut

R
/ \
R_1 R_2

$R_1 \bowtie R_2 = R$

$F_1 \cup F_2 = F$

Trivial: Final Deps
Sid → Sid
Non-Trivial
Sid → Cid

If $\Rightarrow X \rightarrow Y$ ($Y \in X$) (Like Trivial)

Transitive $x \quad y \quad y \quad z$
$x \rightarrow y, y \rightarrow z \Rightarrow x \rightarrow z$

Augment $x \rightarrow y$
$xz \rightarrow yz$

$A \rightarrow B$, $B \rightarrow C$, $C \rightarrow D$

$A^+ = ABCD$

A → B

R(ABCDEF)

$AB \rightarrow C$, $C \rightarrow D$, $D \rightarrow E$, $E \rightarrow F$, $F \rightarrow A$

$C \rightarrow CDEFA$

$AB \rightarrow C$, $C \rightarrow CDEFA$

$AB \rightarrow ABCDEF$ → A candidate Key

(AB is a superkey)

$F \rightarrow A$, $AB \rightarrow CDEFAB$

$FB \rightarrow$ "

$EB \rightarrow$ "

A B C

A → B B → C

A → AB, B → BC

→ Atleast one common

→ and should be a Primary Key

Normal forms:

1) Partial deps:-

AB is a Key ∧ B → C

↑ Non Prime

2) Transitive deps:-

NP → NP

3)

=

AB → C, C → D, B → E, E → F

AB is a candidate key

Violates 2nd normal form

=

AB → CD , D → AE, E → F

AB → CD AB

AB → CD AB (AE)

AB → CD AB AEF

AB → ABCDEF

BCNF:

PK → PK

AB → CD

AB → ABCD

C → B

AC → ABCD

A B C D F E G H

PK →

$A \to BD$

$B \to C$

$E \to FG$

$AE \to H$

$A \to BCD$

$AE \to ABCDFGH$

Violates 2NF

- R_1 (A B C D) Only A
- R_2 (E, F, G) Only ~~B~~
- R_3 (A, E, G) Both ~~AE~~

Restaurant → Pizza Varities

Rest → : Del Area