

DBMS

Chapters

DBMS

- A dbms is a software designed to store, retrieve, define, and manage data in a DB.
- ପରିପରା ତଥକ୍ୟୁତ ମରା ବା ଅନାଫର କେବଳ ଏକ ପାଇଁ ଗମନ ହେଉ ପରିଯୋଜନ,
- କ୍ଷେତ୍ରବିଧି
 - ଶତ କ୍ଷତି- ବୈଷ୍ଣବ ଯୋଗିରେ,
 - ନାରୀଙ୍କ କ୍ଷତି- ପ୍ରକାଶକାରୀଙ୍କ ବିଭାଗ,
 - ବୃକ୍ଷବିଧି କୌଣସି ବିଭାଗ
- DBMS system are highly used to manage collection of data that are
 - Highly valuable
 - Relatively large.
 - Access by multiple user and application.

Applications:

- Banking finance → School, college, university
- Enterprise information → Web base service

Purpose of DBMS System

- Data redundancy and inconsistency
 - Redundancy means multiple copies of same data. Duplicate copies of same data is stored at many place. Ex: Student and teacher table
- Data inconsistency means that different file contain different information of a particular object.

- Difficulty in accessing data (difficult access DBMS and query or).
- Data isolation (multiple files and formats).
- Integrity problems:
 - Define constraint / limit define rules.
 - User can't violate constraint when insert / delete / update records.

Ex: Bank, → Acc open min 5000. → So, it's integrated.
- Security problems.

To solve all of these problems we use DBMS System.

DBMS System.

- A DBMS system is an interrelated data and set a program that allows users to access and modify data.

Data model.

- A collection of conceptual tool to described data, data relationships, data semantics, data constraints.

Data model.

→ Relational model.

→ ER - model

→ object based data model.

→ network model.

→ Hierarchical model.

Data abstraction.

- Hide complexity of data structure from user.
- Data representation level.
- That is called data abstraction.

Relational model

- **Structure** - Data store in tabular form.
- **Object** represent in tabular form.

level of abstraction

- **Physical level** → describe how a record is stored. (Schema)
- **Logical level** → 1st level is Data store
2nd Database 3rd logic 4th Data structure
5th Relation 6th row (Insert data).
- **View level** → Data type or hide part of it.
from user's fore security purpose.

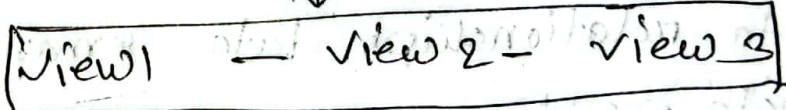
Architecture of DB system

Physical level

↓

logical level

↓



Instance and schemas

- **Instance** → Database at any point of time.
- **two types** Instance 1st logical 2nd physical
- addition, deletion, updation & instance change
- Searching instance change 2nd pt.

- **schema** → a database structure, schema is a structure that represent the logical storage of data in DB. → these are two type
 - **logical**
 - **physical**

- logical schema → the overall logical structure.
- physical schema → the overall physical structure.

□ DDL → (Data Definition Language).

- यह भूल देता है कि database को create, alter, drop करते हैं।
- यह Database को define करते हैं।
- यह Query language.

□ DML → (Data manipulation language).

- यह query language.
- DML two type.
 - Procedural → फ़िर से डिटेल्स बताते हैं।
→ यह access करने की formate → Ex:- Reln Alg
 - Declarative DML →
 - Select, Insert, delete, update operation perform करता है।

□ SQL → (Structured query language).

- यह non-procedural language.
- यह turning machine equivalent language है।

□ DB overall db system का किसी भी एक एक एक administrator

इसका नियन्त्रण करता है। जो उसकी सभी विकास कार्यों को करता है।

Chapter-3 - Introduction to SQL

SQL → allows the specification of information.

→ Domain type.

→ char → varchar(256) → int →
smallint → double → float(n).

Create table.

→ create table T.name

(A₁ D₁, A₂ D₂ --- A_n D_n,

Integrity constraint.

Some integrity relation.

→ Primary key (A₁ - A_n)

→ foreign key (A_m - A_n) reference R.

→ not null.

→ Auto-increment.

Update to table.

→ Insert into T.name values (val, val);

→ Delete ~~from~~ T.name where condition ?

→ Truncate table table name;

→ drop table;

→ ALTER.

→ add column.

→ ALTER table T.name Add col . f.name (DT)

→ drop column

→ ALTER table T.name drop column F;

→ Rename column.

→ ALTER table T.name change column old-col
new col-name (DT)

II) Select clause.

- Select clause use ഫോർമേറ്റ്. ടബ്ലു 10 - നിന്നും കമ്പ്യൂട്ടറിൽ നിന്നും താഴെ പറയുന്നവയും പറയാം.
- SQL രത്ന അല്ഗൈറ്റി പ്രോജക്റ്റ് ഓപറേഷൻ 1, 2, 3
- SQL allows duplicate value. So ഇംഗ്ലീഷിൽ duplicate value എന്ന് avoid ചെയ്യുന്നതും ദിശയിൽ distinct വീഡിയോ ഉപയോഗിച്ചാണ്.
- Select (*) → ഒരു ദാഖല അടിസ്ഥാനം കൂടാൻ കൂടാൻ.
- Select clause മുൻ അളക്കൽ എന്ന് arithmetic expression അല്ലെങ്കിൽ (DIV, (+, -, %, *)).

Select ID, salary / 12 from emp.

III) Where clause.

- Where clause എന്ന് condition എന്ന് ഉപയോഗിച്ചു കൊണ്ട് use query.
- SQL → allows → logical operators. (And, or, not).
- Where clause Δ $<$, \leq , $>$, \geq , $=$, \neq use ഓഫോൺ എന്ന്.

IV) From clause.

- From clause use ഫോർമേറ്റ്. Table 10. Rethink Q. 10 നിൽ.

V) Rename operation.

- Attributes (or rename ഫോർമേറ്റ് "As" use query).
- Table (or rename ഫോർമേറ്റ് "Rename" old-T. to new) use ഫോർമേറ്റ് query.

String operations

- like use to pattern matching in string.
- Percent(%) the parent character matches any substring
- (-) the (-) character use to match a character.
- ? = find value that start with "a"
"ya" = u n ends with "a"
- "%or%" = u n have "or" in any position.
- "%o" = u n in 2nd position
- "a%o" = start with "a" ends with "o"

Ordering clause

- By default any filter field are ordered ASC.
- To sort increase, decrease, or we use, ASC and DSC.
- ORDER keyword to use for this operation.

Where clause predicates

- must and use WHERE clause.
- between → condition between range

Set operations

For table A & B record show

- union → For table A & B record show
- intersect → For table A & B common record show
- except all → For table A & B data show
except from 2nd table B

For table A & B null value show if condition is not use

is not → Null value show

II) Aggregate function

- $\text{Avg}(\text{Salary}) \rightarrow$ average value
 - ↳ formula $\rightarrow \frac{\text{Sum}(\text{Salary})}{\text{Count}(\text{Salary})}$
- $\text{min}(\text{Salary}) \rightarrow$ Salary field. \rightarrow minimum value returned.
- $\text{max}(\text{Salary}) \rightarrow$ Salary field. \rightarrow maximum value returned.
- $\text{sum}(\text{Salary}) \rightarrow$ Salary field. \rightarrow sum of all values.
- $\text{Count} \rightarrow$ rows count also.

Notes: 1) If no select clause in query then QV.

2) If select distinct use ignore duplicate value

III) Group by clause. \rightarrow group wise table \rightarrow second devide QV. then aggregate fn. execute QV.

IV) Set membership.

\Rightarrow find course offered in fall 2017 in and spring 2018

Select course-id from section

where semister = fall and year = 2017 and

course id in (Select course-id from section
where semister = spring and year = 2018)

\Rightarrow find course offered in fall 2017 but not in spring 2018.

Select distinct course-id from section

where semister = fall and year = 2017 and course id not in (Select course id from section where

\Rightarrow Name all instruction whose name is neither mozart nor Einstein.

Select distinct name from instructor where name not in (mozart, Einstein).

Q4 Some clause / all clause:

→ all / some keyword select always where / having clause as still use %,

→ ~~any~~ और जितनी भी।

→ minimum वर्गीय values / tuple returning.

→ Some use ~~any~~ जितना भी। Sub query use %.

Ex: find the name of instructor with salary greater

than that of some instrnor in Bio dept.

Select name from instructor

where salary > Some (Select salary from Instructor

where dept = Biology)

Sub query का उपयोग करके Bio dept में से वह salary जिसका salary अधिक है।

Output: 11120, 11300.

1120, 11300 - minimum रफ़ि

12300, salary जो 20% है।

all
Ex: find the name of all instructors whose salary is greater than the salary of instructor in Biology dept.

Ans: Select name from Instructor where salary > all
(Select salary from Instructor where dept = Bio)

□ Exists clause / not exist clause (Correlated nested qu)

→ কোনো কোনো nested query / Sub query কোনো
→ nested query = outer query + inner query

→ q or and both { Exist = intersect } not in { use of
all / some. } Sub query add one

not exist = union { q } exist / not exists. কোনো

normally In / not in কোনো কোনো first inner / Sub query execute কোনো. Then outer query execute

কোনো exist / not exist কোনো - প্রথমে outer query কোনো value কোনো then কোনো value match কোনো.
inner query কোনো মান,

□ Subquery in from clause.

→ এখনের মানের ক্ষেত্রে Data find করতে হবেন ও কোনো
existing table কোনো find out কোনো আইন না. So কোনো
from clause কোনো কোনো Sub query কোনো কোনো
table কোনো Preprocessed কোনো কোনো কোনো updated table
কোনো কোনো কোনো প্রস্তুতি কোনো নাই.

{ কোনো কোনো from Clause কোনো Sub query কোনো কোনো table
কোনো কোনো required data easily find out কোনো কোনো

□ Complex query part কোনো কোনো Paint কোনো.

Modification of the database

- Rtm (or) tuples delete sql.
- Create new tuple insert, sql.
- Change rtm to value update sql.

Delete

- Delete from table where condition.

Insertion.

- Insert into table name values (value1, value2, ...)

Update.

- Update table-name set change where condition.

-----x-----
-----End-----

In / not in of where

Sub queries } gate

Group by

having

join

DBMS

Chap-2

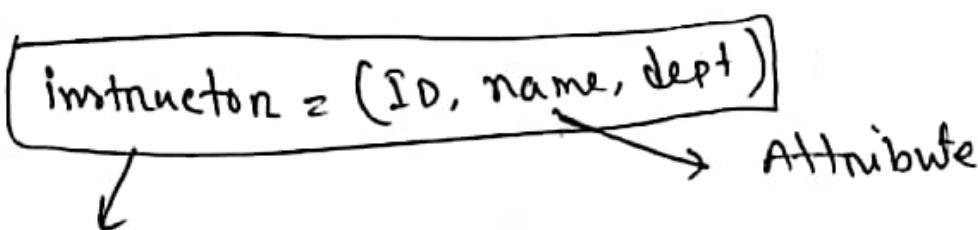
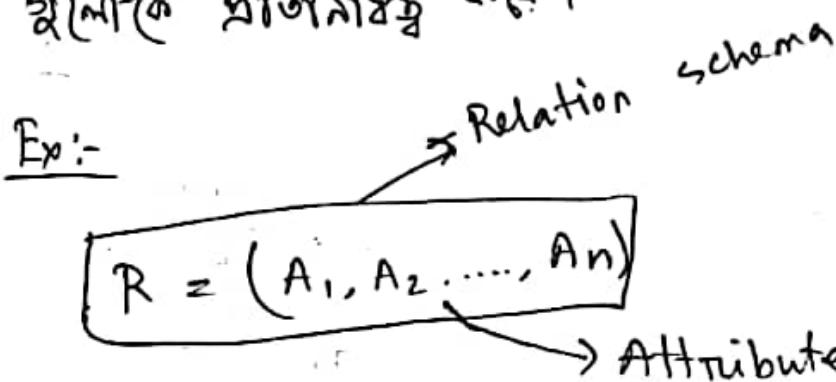
Relation Schema and Instance :

Attribute → A component of database refer একটি, একটা - table.

Relation Schema → A blueprint of database front

-এর সমূহ database এর data

বিশেষ প্রতিকর্ষ এবং।



Tuple :- tuple এর অন্তর্ভুক্ত record or row in table

↳ Tuples arbitrary order & store করা

Attributes

- Domain → attribute નું માત્ર તાંકી value કોણું allow કરી શકે છે એ,
- Null → નાચું કોણું નાચું special value કરી શકે નાચું member.
- attribute નું અને અધ્યાત્મ અને atomic હોય

Database Schema → Database નું logical structure ?

એથી Schema :

- physical - actual storage of data
- logical - all the logical constraints that need to be applied on the data stored.

Database Instance → Database ey નું (નિયાન અને) data અને info કુટુંબ અનુભવ (સંખ્યાની) instance.

Ex:-

Schema : instructor (ID, Name, Dept)

Instance : 004 | Akash | CSE

3

Key

→ Super Key — Super key ඇහි group of single or multiple keys සඳහා මත තුළු තැබූ නො පෙන්වනු ලබයි.

→ Primary Key — primary key ඇහි මත තුළු table නේ group of column — (විවෘත තුළු) table නේ group of column — (විවෘත තුළු) row නේ ප්‍රමාණය නො පෙන්වනු ලබයි.

→ Alternate Key — මුදල්සු table නේ group of column — (විවෘත තුළු) row නේ ප්‍රමාණය ප්‍රමාණය නො පෙන්වනු ලබයි. මුදල්සු table නේ මුදල්සු primary key නො පෙන්වනු ලබයි. Primary key නො පෙන්වනු ලබයි. මුදල්සු primary key නො පෙන්වනු ලබයි. මුදල්සු alternate key නො පෙන්වනු ලබයි.

→ Rules for defining Primary Key —

- 1. මුදල්සු row නේ ප්‍රමාණය primary key value නො පෙන්වනු ලබයි.

- 2. මුදල්සු row නේ යෙහෙම primary key value නො පෙන්වනු ලබයි.

⇒ primary key field null असावा आवडे ना.

→ Candidate key - याचे SQL नंदी एका set of attributes (एका) table नंदी tuples शैलीत uniquely identify ठेवो.

→ Properties of Candidate key -

- Unique value असावा
- null value असाव्या नाही.
- SQL multiple attributes असावा candidate key नाही.

→ Foreign key - याचे एका नोंदी table नंदी column गोंडावा एका नोंदी table नंदी primary key तरुणीत आवडे नाही.

Relational Algebra

Relational algebra इतर नोंदी procedural query language - (एका) Relation तरुणीत input फ्रेग्रेंट तरुणीत output फ्रेग्रेंट relation तरुणीत ठेवो.

6
Six basic operators:-

- Select (σ)
- Project (Π)
- Union (\cup)
- set difference (-)
- Cartesian Product (\times)
- Rename (ρ)

Select:-

- select operation - select tuples that satisfy a given predicate
- Notation: $\sigma_p (r)$
 - ↙ Select
 - ↘ selection predicate
- Comparisons symbol -
 - =
 - ≠
 - >
 - <
 - >=
 - <=
- Connectives -
 - And (\wedge)
 - OR (\vee)
 - NOT (\neg)

Project :-

- denoted by Π
- It is used to select desired column or attributes from a table or relation.
- Notation - $\Pi_{A_1, A_2, A_3, \dots, A_k} (r)$
 - ↓ project
 - ↓ Attributes
 - relation

Composition of Relational Operations:-

- Relation - The result of relational operation.

Cartesian Product

- denoted by \times
- Cartesian Product \times relation \times relation
to table to information combine

Join Operation

- Join \Rightarrow Cartesian product \Rightarrow combination \Rightarrow selection process \Rightarrow follow step
- Join operation \Rightarrow two \Rightarrow two \Rightarrow two different relation (table) \Rightarrow two \Rightarrow tuples \Rightarrow one \Rightarrow one \Rightarrow condition \Rightarrow two \Rightarrow two \Rightarrow two satisfied \Rightarrow two \Rightarrow two
- Ex: $\sigma_{\text{instructor.id} = \text{student.id}} (\text{instructor} \times \text{student})$

Union :-

- denoted by \cup
- two relation \Rightarrow combine two
- Notation: $\pi_r \cup \pi_s$
 - ↳ r, s \Rightarrow same number of attribute
 - ↳ r, s

Intersection:-

- two table \Rightarrow have same tuple \Rightarrow two
- two \Rightarrow two
- Notation: $\pi_r \cap \pi_s$
 - ↳ r, s same number of attribute
 - ↳ r, s

Set Difference :-

- If a tuple exists in one table & not in another table, then it is called tuple too.
- To find set difference we use '-'.
- Notation: $R - S$
- Two compatible relations to have same arity.
- relation exists for same arity.

Equivalent Queries:

একটি বিশেষ প্রশ্নের query এর অন্যান্য রূপ।
যদি ১০ টি query এর মধ্যে কোনো কোনো query অন্যান্য query এর সমাধান হয়।

Ex:-

Query - 1

$\sigma_{\text{dept} = \text{"Physics"} \wedge \text{salary} > 50,000} (\text{instructor})$

Query - 2

$\sigma_{\text{dept} = \text{"Physics"} \left(\sigma_{\text{salary} > 50,000} (\text{instructor}) \right)}$