Database Management System

CS-502

Unit – 1 : DBMS concept and architecture introduction

Unit – 2 : Relational data models

Unit – 3 : Normalization

Unit – 4 : Transaction Processing Concepts

Unit – 5 : Study of RDBMS software through SQL/MySQL

What is a Database?

A database is a collection of related data which represents some aspect of the real world.

(Introduction to DBMS)

DBMS (Data Base Management System) एक Software होता हैं, जिसका उपयोग डेटाबेस को बनाने और संभालने के लिए किया जाता हैं। DBMS अपने उपयोगकर्ताओं (users), और प्रोग्रामरों को एक व्यवस्थित तरीके के साथ डाटा को बनाने, संभालने और update करने की सुविधा प्रधान करता हैं।

DBMS के उदाहरण – MySQL, Postgre SQL, Microsoft Access, Oracle etc.

What is DBMS?

Database Management System (DBMS) is a software for storing and retrieving user's data while considering appropriate security measures. It consists of a group of programs which manipulate the database.

DBMS allows users to create their own databases as per their requirement.

The term "DBMS" includes the user of the database and other application programs.

It provides an interface between the data and the software application.

Characteristics of Database

Management System

Provides security and removes redundancy

Self-describing nature of a database system

Insulation between programs and data abstraction

Support of multiple views of the data

Sharing of data and multiuser transaction processing

DBMS allows entities and relations among them to form tables.

It follows the ACID concept (Atomicity, Consistency, Isolation, and Durability).

DBMS supports multi-user environment that allows users to access and manipulate data in parallel.

Application of DBMS.

Banking sector, Airlines, UniversitiesUniversities etc.

Types of DBMS

Four Types of DBMS systems are:

1.Hierarchical database

  Use:-telecommunications and banking.

2.Network database.

 Use:-when querying results from multiple foreign-key tables associated with one primary-key table.

3.Relational database.

 Use:-The primary benefit of the relational database approach is the ability to create meaningful information by joining the tables.

4.Object-Oriented database.

 Use:-When relational database concepts like concurrency, transaction, and recovery are applied, the object-oriented database model is created. Object-oriented databases.

Advantages of DBMS

No redundant data: Redundancy removed by data normalization. No data duplication saves storage and improves access time.

Data Consistency and Integrity: As we discussed earlier the root cause of data inconsistency is data redundancy, since data normalization takes care of the data redundancy, data inconsistency also been taken care of as part of it.

Data Security: It is easier to apply access constraints in database systems so that only authorized user is able to access the data. Each user has a different set of access thus data is secured from the issues such as identity theft, data leaks and misuse of data.

Privacy: Limited access means privacy of data.

Easy access to data – Database systems manages data in such a way so that the data is easily accessible with fast response times.

Easy recovery: Since database systems keeps the backup of data, it is easier to do a full recovery of data in case of a failure.

Flexible: Database systems are more flexible than file processing systems.

Disadvantages of DBMS:

DBMS implementation cost is high compared to the file system.

Complexity: Database systems are complex to understand.

Performance: Database systems are generic, making them suitable for various applications. However this feature affect their performance for some applications.

Entity–relationship model (ER model)

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of

 E-R model are: entity set and relationship set.

Rectangle: Represents Entity sets.

Ellipses: Attributes

Diamonds: Relationship Set

Lines: They link attributes to Entity Sets and Entity sets to Relationship Set

Double Ellipses: Multivalued Attributes

Dashed Ellipses: Derived Attributes

Double Rectangles: Weak Entity Sets

Double

 Lines: Total participation of an entity in a relationship set

Difference Between DBMS and RDBMS

1.

DBMS फ़ाइल के रूप में डेटा को सेव करता है

----RDBMS डाटा को टेबल में सेव करता है

2.

एक बार मे केवल एक ही डेटा एलिमेंट्स को एक्सेस किया जाता है|

----   Multiple data elements can be एक साथ कई डेटा एलीमेंट को एक्सेस किया जा सकता है

3.

एक डेटा एलिमेंट का दूसरे डेटा एलिमेंट से कोई सम्बन्ध नही होता है|

----   इसमे डेटा टेबल के फॉर्म में सेव होतें हैं जो कि एक दूसरे से related होतें हैं|

4.

Normalization नहीं होता है|

---Normalization होता है

5.

DBMS डिस्ट्रिब्यूटेड डेटा को सपोर्ट नहीं करता है |

----RDBMS डिस्ट्रिब्यूटेड डेटा को सपोर्ट करता है

6.

यह डेटा को navigational या hierarchical फॉर्म में सेव करता है|

    यह डेटा को सेव करने के लिए tabular फॉर्म का उपयोग करता है|

7.

यह छोटे डेटा को हैंडल करने के लिए यूज़ किया जाता है |

-----यह बहुत ज्यादा डेटा को हैंडल करने के लिए यूज़ किया है

8.

Data redundancy(अनावश्यक) इसमे एक आम बात है|

---   Keys और index में डेटा redundancy नहीं होती है

9.

यह छोटे आर्गेनाइजेशन

के में यूज़ होता है|

----यह बड़े आर्गेनाइजेशन में यूज़ किया जाता है जहाँ बहुत ज्यादा डेटा को हैंडल करना होता है

10.

single यूजर को सपोर्ट करता है|

----   यह मल्टीयूज़र को सपोर्ट करता है

11.

बड़े डेटा को fetch करने में बहुत समय लगता है|

-----इसमे डेटा fetching DBMS से तेज होती है

12.

डेटा manipulation के लिए low सिक्योरिटी लेवल होता है|

-----   RDBMS में मल्टिपल लेवल की सिक्योरिटी होती है

13.

Low software और hardware की आवश्यकता होती है|

---Higher software और hardware की आवश्यकता होती है

14.

Examples: XML, Microsoft Access, etc.

Examples: MySQL, PostgreSQL, SQL Server, Oracle, etc.

Relational model in DBMS

In relational model, the data and relationships are represented by collection of inter-related tables. Each table is a group of column and rows, where column represents attribute of an entity and rows represents records.

Sample relationship Model: Student table with 3 columns and four records.

Table: Student

Stu_Id	Stu_Name	Stu_Age
111	Ashish	23
123	Saurav	22
169	Lester	24
234	Lou	26

Constraints in DBMS

Constraints are the rules enforced on the data columns of a table. These are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the database.

Constraints could be either on a column level or a table level. The column level constraints are applied only to one column, whereas the table level constraints are applied to the whole table.

NOT NULL Constraint − Ensures that a column cannot have NULL value.

DEFAULT Constraint − Provides a default value for a column when none is specified.

UNIQUE Constraint − Ensures that all values in a column are different.

PRIMARY Key − Uniquely identifies each row/record in a database table.

FOREIGN Key − Uniquely identifies a row/record in any of the given database table.

CHECK Constraint – The CHECK constraint ensures that all the values in a column satisfies certain conditions.

INDEX – Used to create and retrieve data from the database very quickly.

RDBMS stands for relational database management system. A relational model can be represented as a table of rows and columns. A relational database has following major components:

1. Table

2. Record or Tuple

3. Field or Column name or Attribute

4. Domain

5. Instance

6. Schema

7. Keys

1. Table

A table is a collection of data represented in rows and columns. Each table has a name in database. For example, the following table "STUDENT" stores the information of students in database.

Ex:-Table: STUDENT

2. Record or Tuple

Each row of a table is known as record. It is also known as tuple. For example, the following row is a record that we have taken from the above table.

3. Field or Column name or Attribute

The above table "STUDENT" has four fields (or attributes): Student_Id, Student_Name, Student_Addr & Student_Age.

4. Domain

A domain is a set of permitted values for an attribute in table. For example, a domain of month-of-year can accept January, February,…December as values, a domain of dates can accept all possible valid dates etc. We specify domain of attribute while creating a table.

An attribute cannot accept values that are outside of their domains. For example, In the above table "STUDENT", the Student_Id field has integer domain so that field cannot accept values that are not integers for example, Student_Id cannot has values like, "First", 10.11 etc.

5. Key

it is used for identifying unique rows from table. It also establishes relationship among tables.

[07/01, 6:16 pm] Rahul dogney: Types of keys in DBMS

Primary key is a minimal set of attributes (columns) in a table that uniquely identifies tuples (rows) in that table.

Primary Key Example in DBMS

Lets take an example to understand the concept of primary key. In the following table, there are three attributes: Stu_ID, Stu_Name & Stu_Age. Out of these three attributes, one attribute or a set of more than one attributes can be a primary key.

Attribute Stu_Name alone cannot be a primary key as more than one students can have same name.

Attribute Stu_Age alone cannot be a primary key as more than one students can have same age.

Super Key in DBMS: A super key is a set of one or more attributes (columns), which can uniquely identify a row in a table. Often DBMS beginners get confused between super key and candidate key, so we will also discuss candidate key and its relation with super key in this article.

Table: Employee

{Emp_SSN}

{Emp_Number}

{Emp_SSN, Emp_Number}

{Emp_SSN, Emp_Name}

{Emp_SSN, Emp_Number, Emp_Name}

{Emp_Number, Emp_Name}

Candidate Keys: As I mentioned in the beginning, a candidate key is a minimal super key with no redundant attributes. The following two set of super keys are chosen from the above sets as there are no redundant attributes in these sets.

{Emp_SSN}

{Emp_Number}

How candidate key is different from super key?

Answer is simple –

Candidate keys are selected from the set of super keys, the only thing we take care while selecting candidate key is: It should not have any redundant attribute. That's the reason they are also termed as minimal super key.

Foreign keys:- Foreign keys are the columns of a table that points to the primary key of another table. They act as a cross-reference between tables.

For example:

In the below example the Stu_Id column in Course_enrollment table is a foreign key as it points to the primary key of the Student table.

Course_enrollment table:

Course_Id	Stu_Id
C01	101
C02	102
C03	101
C05	102
C06	103
C07	102

Student table:

Stu_Id	Stu_Name	Stu_Age
101	Chaitanya	22
102	Arya	26
103	Bran	25
104	Jon	21

Composite key: A key that has more than one attributes is known as composite key. It is also known as compound key.

[07/01, 6:23 pm] Rahul dogney: Database normalization: In DBMS, normalization is the process of organizing (organizing) data. Generally it is two step process which is as follows: -

First step: - In the first step, it eliminates redundant data (data that is stored more than once) from the relational table.

Second step: - In the second step, it ensures that only the data related to it is stored in the table. The main objective of normalization is to create a group of relational tables that do not contain redundant data and which can be continuously and correctly modified.

[07/01, 6:27 pm] Rahul dogney: Types of Normal Forms:- DBMS में normal forms पाँच प्रकार के होते हैं। लेकिन हम यहां पर तीन प्रकार के नार्मल फॉर्स के बारें में discuss करेंगे जिनको कि E.F CODD ने प्रस्तावित किया है।

1. First normal form(1NF):- एक टेबल तब first normal form में होता है जब वह repeating groups(बार-बार एक ही डेटा) को contain नही करता।

वह रिलेशन या टेबल जो repeating groups को contain किये रहती है वह रिलेशन या टेबल un normalized कहलाता है।

2. Second normal form(2NF):- एक टेबल या

रिलेशन तब 2nd normal form में होता है जब वह 1st normal form की सभी requirements को पूरी करता हों और सभी non key attributes पूरी तरह से primary key पर निर्भर हों।

3. Third normal form(3NF):- कोई टेबल या रिलेशन तब 3rd normal form में होता है जब वह 2nd normal form की सभी जरूरतों को पूरी करता हो तथा उनमें transitive function dependency नही होनी चाहिए।

[07/01, 6:36 pm] Rahul dogney: Fourth normal form(4NF):-

एक relation या table तब 4NF में होती है जब वह निम्नलिखित condition को satisfy करते है: 'एक relation या टेबल तब 4NF में होती है यदि वह 3 normal form(3NF) में हो तथा उसके पास कोई multivalued dependencies ना हो।"

Multivalued dependency क्या होती है?

"Multivalued dependency तब होती है जब एक table में एक से ज्यादा independent(स्वतंत्र) multivalued attributes old gi" multivalued dependency at ->> चिन्ह से प्रदर्शित किया जाता है।

उदाहरण के लिए:- कोई मोबाइल कंपनी प्रत्येक model के दो color(white व grey) के मोबाइल बनाती है।

Mob_model Manuf_year Color

C_07   2013   White

C_07   2013   Grey

C_11   2014   White

C_11   2014   Grey

अब यहां पर manuf_year और color एक दूसरे से independent है तथा वे mobile_model पर dependent है। तो हम इस प्रकार की dependencies को निम्न प्रकार से प्रदर्शित करते है।mobile_model->>manuf_year

mobile_model->>color

Fifth normal form(5NF):-

एक relation या table तब 5NF में होती है जब वह निम्नलिखित condition को satisfy करती है: "एक टेबल या रिलेशन तब 5NF में होती है जब वह 4NF में हो तथा table में कोई non-loss decomposition ना हो।"