DBMS/SQL

Lesson 01: Getting Started
with Database

Lesson Objectives

- To understand the following topics:
  - Database
  - Features of DBMS
  - DBMS models
  - Relational DBMS

1.1: Introduction to Database

What is Data?

- Data (plural of the word datum) is a factual information used as a basis for reasoning, discussion, or calculation
- Data may be numerical data which may be integers or floating point numbers, and non-numerical data such as characters, date etc.
- Data by itself normally doesn't have a meaning associated with it.
  - e.g:-   Jack

  - 01-jan-71
  - 15-jun-05
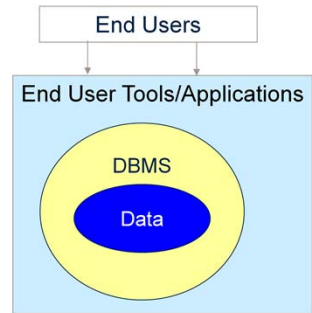 - 50000

1.1: Introduction to Database

What is Information?

- Related data is called as information
- Information will always have a meaning and context attached to the data element
- When we add meaning and context to the data it becomes information.
  - Employee name: Jack
  - Date of birth: 01-jan-71
  - Data of joining: 15-jun-05
  - Salary: 50000
  - Department number: 10

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

1.1: Introduction to Database
Defining Database, DBMS & Schema

- Database: It is a set of inter-related data
- DBMS: It is a software that manages the data
- Schema: It is a set of structures and relationships, which meet a specific need

End Users

End User Tools/Applications

DBMS

Data

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved   5

Introduction to Database:
A logically coherent collection of related data ("information") with inherent meaning, built for a certain application, and representing a subset of the "real-world". For eg: Customer database in bank, Employee Details
The software that manages the database is known as "Database Management System" or "DBMS". Hence DBMS can be described as "a computer-based record keeping system which consists of software for processing a collection of interrelated data". The general purpose of a DBMS is to provide for the definition, storage, and management of data in a centralized area that can be shared by many users
A set of structures and relationships that meet a specific need is called as a "schema".
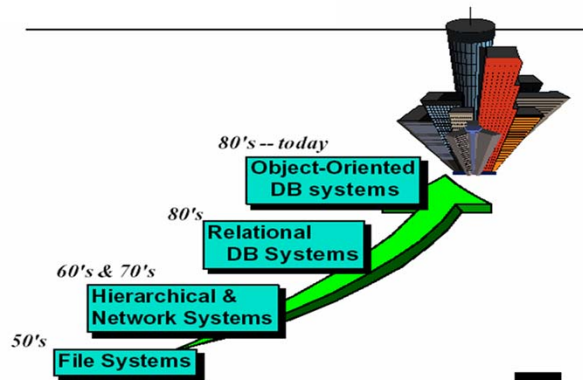
1.2: Features of DBMS
Characteristics of DBMS

- Given below are the characteristics of DBMS:
  - Control of Data Redundancy
    - Traditionally, same data is stored in a number of places
    - Gives rise to data redundancy and its disadvantages
    - DBMS helps in removing data redundancies by providing means of data- integration.
  - Sharing of Data
    - DBMS allows many applications to share the data.
  - Maintenance of Integrity
    - DBMS maintains the correctness, consistency, and interrelationship of data with respect to the application, which uses the data.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING   Copyright © Capgemini 2015. All Rights Reserved   7

Characteristics of DBMS:
Some of the characteristics of the DBMS are given below:
Control of Data Redundancy
    When the same data is stored in a number of files, it results in
    data redundancy. In such cases, if the data is changed at one
    place, the change has to be duplicated in each of the files.
    The main disadvantages of data redundancy are:
        Storage space is wasted.
        Processing time may be wasted as more data needs
        to be handled.
        Inconsistencies may creep in.
    DBMS helps in removing redundancies by providing means of
    integration.
Sharing of Data
    DBMS allows many applications to share the data.
Maintenance of Integrity
    Integrity of data refers to the correctness, consistency and
    interrelationship of data with respect to the application that
    uses the data. Some of the aspects of data integrity are:
        Many data items can only take a restricted set of
        values.

        .

Characteristics of DBMS

- Support for Transaction Control and Recovery
  - DBMS ensures that updates physically take place after a logical Transaction is complete.
- Data Independence
  - In DBMS, the application programs are transparent to the physical organization and access techniques.
- Availability of Productivity Tools
  - Tools like query language, screen and report painter, and other 4GL tools are available.

Characteristics of DBMS (contd.):

Certain field values cannot be duplicated across records. Such restrictions, called primary key constraints, can be defined to the DBMS.
Data integrity, which defines the relationships between different files, is called referential integrity rule, which can also be specified to the DBMS

Support for Transaction Control and Recovery

Multiple changes to the database can be clubbed together as a single "logical transaction".
The DBMS ensures that the updates take place physically, only when the logical transaction is complete.

Data Independence

In conventional file based applications, programs need to know the "data organization" and "access technique" to be able to access the data.
This means that if you make any change in the manner the data is organized, then you have to make changes to the application programs that apply to the data.
In DBMS, the application programs are transparent to the "physical organization" and "access techniques".

Characteristics of DBMS

- Control over Security
  - DBMS provides tools with which the DBA can ensure security of the database.
- Hardware Independence
  - Most DBMS are available across hardware platforms and operating systems.

Characteristics of DBMS (contd.):
Availability of Productivity Tools
> Tools like query language, screen and report painter, and other 4GL tools are available.
> These tools can be utilized by the end-users to query, print reports, etc. SQL is one such language, which has emerged as standard.

Security
> DBMSes provide tools, which can be used by the DBA to ensure security of the database.

Hardware Independence
> Most DBMSes are available across hardware platforms and operating systems.
> Thus the application programs need not be changed or rewritten when the "hardware platform" or "operating system" is changed or upgraded.

Levels of Abstraction

- There are three levels of database abstraction:
  - Conceptual Level:
    - The overall integrated structural organization of the database.
  - Physical Level:
    - The information about how the database is actually stored in the disk.
  - View / External Level:
    - The user view of the database. It is different for different users based on application requirement.

1.3: The Data Models
What is a Data Model?

- The "Data model" defines the range of data structures supported and the availability of data handling languages.
  - It is a collection of conceptual tools to describe:
    - Data
    - Data relationships
    - Constraints
  - There are different data models:
    - Hierarchical Model
    - Network Model
    - Relational Model

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved   11

What is a Data Model?
A "Data model" is a conceptual representation of the data structures that are required by a database. The data structures include:
        the data objects
        the associations between data objects, and
        the rules which govern operations on the objects
As the name implies, the "Data model" focuses on the data that is required, and how it should be organized rather than the operations that will be performed on the data.
The DBMS MODELS
        The range of "data structures" that are supported, and the availability of data handling languages depend on the model of DBMS on which it is based. The models are:
                The hierarchical model
                The network model
                The relational model

Why is Data Modeling Important?

- Why is Data Modeling important?
    - The goal of the "data model" is to ensure that all the data objects required by the database are completely and accurately represented.
    - The "data model" uses easily understood notations and natural language. Hence, it can be reviewed and verified as correct by the end-users.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved   12

Why is Data Modeling Important?

The "data model" is also detailed enough to be used, by the database developers, as a "blueprint" for building the physical databases. The information contained in the "data model" will be used to define the relational tables, primary and foreign keys, stored procedures, and triggers.
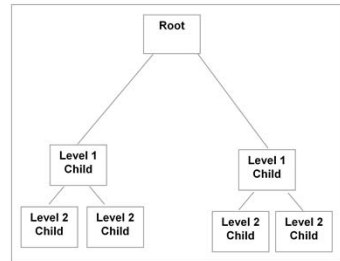
Poorly designed databases require more time in the long-term. Without careful planning you may create a database that:

> Omits data required to create critical reports.
> Produces results that are incorrect or inconsistent.
> Is unable to accommodate changes in the user requirements.

1.3: The Data Models
Hierarchical Model

▪ The Hierarchical model:
  ▪ In this model, data is represented by a simple tree-structure.
  ▪ Relationships between entities are represented as parent-child.
  ▪ Many-to-many relationships are not allowed.
  ▪ Parents and children are tied together by links called "pointers".

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved   13

Hierarchical Model- Example

- Example:
  - Consider a student course - marks database.
  - In the Hierarchical model a student can register for many courses and gets marks for each course.

- A parent can have many children
- A child cannot have more than one parent
- No child can exist without its parent

Scode	Sname
S1	A

Scode	Sname
S2	B

Ccode	Cname	Marks
C1	Physics	65
C2	Chemistry	78
C3	Maths	83
C4	Biology	85

Ccode	Cname	Marks
C3	Maths	83
C4	Biology	85

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved   14

Example of a Hierarchical model:
Consider a student course - marks database. In the Hierarchical model a student can register for many courses, and get marks for each course.
The student record is called as "root". It has got a course - marks record that is called as "child record".
In general:
        A parent can have many children.
        A child cannot have more than one parent.
        No child can exist without its parent.

Hierarchical Model - Possibilities

- Possibilities in a Hierarchical model:
  - INSERT
    - Insertion of Dummy student is required to introduce a new course.
  - DELETE
    - Deleting a student - the only one to take the course deletes course information.
  - UPDATE
    - To change the course name of one course, the whole database has to be searched. This may result in data inconsistency.

Possibilities in a Hierarchical model:

In the Hierarchical model, following possibilities exist:

INSERT

Since no child record can exist without it's parent, it is not possible to insert the new course details without introducing a dummy student record.

DELETE

If a course is selected by only one student, then deleting that student will automatically delete all information about the course.

UPDATE

To change the course name of one course, the whole database has to be searched. This may result in data inconsistency.

1.3: The Data Models
Network Model

- The Network model:
  - The Network model solves the problem of data redundancy by representing relationships in terms of "sets" rather than "hierarchy".
  - A record occurrence may have any number of immediate superiors.
  - The Network model supports many-to-many relationships.
  - There is no restriction on number of parents.
  - A record type can have a number of parent and child record types.
  - It is more complex than the Hierarchical model because of links.
  - It is a superset of the Hierarchical model.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved   16

Network Model - Example

- In the example of student course – marks, "student record" and "course record" is linked together through "marks record".
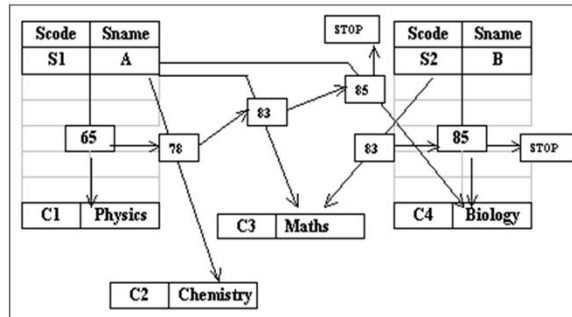
Example of Network model:
In the Network model, the "student record" and "course record" is linked together through a "marks record".
There are no restrictions on number of parents.
A record type can have any number of "parent" and "child" record types.
The Network model is more complex than the Hierarchical model because of it's links.
The Network model can represent any structure that is designed in the Hierarchical model. Hence, it is a superset of the Hierarchical model.

Network Model - Possibilities

- Possibilities in a Network model:
  - INSERT
    - Inserting a "course record" or "student record" poses no problems. They can exist without any connectors till a student takes the course.
  - DELETE
    - Deleting any record automatically adjusts the chain.
  - UPDATE
    - Update can be done only to a particular child record.

Possibilities in a Network model:
In Network model, following possibilities exist:
INSERT
> Inserting a course record or student record poses no problems, as they can exist without any connectors till a student takes the course.

DELETE
> Deleting any record automatically adjusts the chain.

UPDATE
> Update can be done only to a particular child record.

1.3: The Data Models
Relational Model

- **The Relational model:**
  - The Relational model developed out of the work done by Dr. E. F. Codd at IBM in the late 1960s. He was looking for ways to solve the problems with the existing models.
  - At the core of the Relational model is the concept of a "table" (also called a "relation"), which stores all data.
  - Each "table" is made up of:
    - "records" (i.e. horizontal rows that are also known as "tuples"), and
    - "fields" (i.e. vertical columns that are also known as "attributes")

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved   19

Relational Model

- The Relational model:
  - Examples of RDBMS:
    - Oracle
    - Informix
    - Sybase
  - Because of lack of linkages, the Relational model is easier to understand and implement.

Student Table

Scode	Sname
S1	A
S2	B

Course Table

Ccode	Cname
C1	Physics
C2	Chemistry
C3	Maths
C4	Biology

Marks Table

Ccode	Scode	Marks
C1	S1	65
C2	S1	78
C3	S1	83
C4	S1	85
C3	S2	83
C4	S2	85

Relational Model - Possibilities

- Possibilities in a Relational model:
  - INSERT
    - Inserting a "course record" or "student record" poses no problems because tables are separate.
  - DELETE
    - Deleting any record affects only a particular table.
  - UPDATE
    - Update can be done only to a particular table.

1.4: Relational DBMS
Relational Tables

- Examples of Relational tables:

"column" or "attribute"

Dept table

Deptno	Dname	Loc
10	Accounting	New York
20	Research	Dallas
30	Sales	Chicago
40	Operations	Boston

"row" or "tuple"

Emp table

Empno	Empname	Job	Mgr	Deptno
7369	Smith	Clerk	7902	20
7499	Allen	Salesman	7839	30
7566	Jones	Manager	7839	20
7839	King	President		10
7902	Ford	Analyst	7566	20

Relational DBMS (RDBMS):
The Relational model presents an orderly, predictable, and intuitive approach for:

Organizing data,
Manipulating data, and
Viewing data

RDBMS Terminology
Relational data consists of relations.

A relation (or relational table) is a "two dimensional" table with special properties.

A relational table consists of:

a set of named columns, and
an arbitrary number of rows

The columns are called as "attributes" or "fields". The rows are called as "tuples" or "records".

Each "attribute" is associated with a "domain".

A "domain" is a set of values that may appear in one or more columns.

Relational Tables - Properties

- Properties of Relational Data Entities:
  - Tables must satisfy the following properties to be classified as relational:
    - Entries of attributes should be single-valued.
    - Entries of attributes should be of the same kind.
    - No two rows should be identical.
    - The order of attributes is unimportant.
    - The order of rows is unimportant.
    - Every column can be uniquely identified.

Properties of Relational Data Entities:
Relational tables have six properties, which must be satisfied for any
table to be classified as Relational. These are :

1.4: Relational DBMS
Data Integrity

- "Data Integrity" is the assurance that data is consistent, correct, and accessible throughout the database.
- Some of the important types of integrities are:
  - Entity Integrity:
    - It ensures that no "records" are duplicated, and that no "attributes" that make up the primary key are NULL.
    - It is one of the properties that is necessary to ensure the consistency of the database.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING   Copyright © Capgemini 2015. All Rights Reserved   24

Data Integrity:
Data Integrity refers to the wholeness and soundness of the database.
Some of the most important integrities are given below.
Domain Constraints
        A "domain" is a set of values that are permitted to appear in
        one or more columns. Once a "domain" is specified and a
        "column" is associated with the "domain", then the "column"
        can take only those values that are permitted by the "domain".
Primary Key and Entity Integrity
        "Primary key" is a "column" or "set of columns" in a table
        which uniquely identifies a "row" in a table.
                No two rows of the table can have the same values
                for the Primary key.
                Entity integrity is maintained by ensuring that none of
                the columns that make up the Primary key can take
                "NULL" (unknown) values.

Data Integrity

- Foreign Key and Referential Integrity
  - The Referential Integrity rule: If a Foreign key in table A refers to the Primary key in table B, then every value of the Foreign key in table A must be null or must be available in table B.
- Unique Constraint:
  - It is a single field or combination of fields that uniquely defines a tuple or row.
  - It ensures that every value in the specified key is unique.
  - A table can have any number of unique constraints, with at most one unique constraint defined as a Primary key.
  - A unique constraint can contain NULL value.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Data Integrity:
Data Integrity refers to the wholeness and soundness of the database.
Some of the most important integrities are given below.
Domain Constraints
      A "domain" is a set of values that are permitted to appear in one or more columns. Once a "domain" is specified and a "column" is associated with the "domain", then the "column" can take only those values that are permitted by the "domain".
Primary Key and Entity Integrity
      "Primary key" is a "column" or "set of columns" in a table which uniquely identifies a "row" in a table.
            No two rows of the table can have the same values for the Primary key.
            Entity integrity is maintained by ensuring that none of the columns that make up the Primary key can take "NULL" (unknown) values.

Data Integrity (contd.):

Update Cascade referential Integrity

> This means that if a Primary key value is updated, then all
> Foreign key values dependent on it will be updated to the new
> value of Primary key. That is to say, if we change the deptno
> 10 to 50, then all the employees in deptno 10 will be shifted,
> as well, to deptno 50 (column deptno will be automatically
> updated).

Column Constraints

> These are the constraints, which specify restrictions on the
> values that can be taken by a column. These restrictions may
> be defined with or without other values in the same row.

Data Integrity

- Column Constraint:
  - It specifies restrictions on the values that can be taken by a column.

DEPT table

Deptno	Dname	Loc
10	Accounting	New York
20	Research	Dallas

EMP table

Empno	Empname	Job	Mgr	Deptno
7369	Smith	Clerk	7902	20
7499	Allen	Salesman	7839	30

Database Administrator

- A Database Administrator (DBA) is the database architect.
  - DBA is responsible for the design and implementation of new databases, and:
    - centrally manages the database.
    - decides on the type of data, internal structures, and their relationships
    - ensures the security of the database
    - controls access to the data through user codes and passwords
    - can restrict the views or operations that the users can perform on the database

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved   28

Data Integrity (contd.):

Update Cascade referential Integrity

> This means that if a Primary key value is updated, then all Foreign key values dependent on it will be updated to the new value of Primary key. That is to say, if we change the deptno 10 to 50, then all the employees in deptno 10 will be shifted, as well, to deptno 50 (column deptno will be automatically updated).

Column Constraints

> These are the constraints, which specify restrictions on the values that can be taken by a column. These restrictions may be defined with or without other values in the same row.

Summary

- In this lesson, you have learnt:
  - What is a Database?
  - Characteristics of DBMS
  - The Data models, including:
    - the Hierarchical model
    - the Network model
    - the Relational model
  - Relational DBMS (RDBMS)
  - What is Data Integrity?

Summary

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Review Question

- Question 1: A DBA ___.
  - Option 1: ensures the security of the application server.
  - Option 2: controls access to the data through the user codes and passwords.
  - Option 3: manages users.

- Question 2: The Physical Level is ___.
  - Option 1: the overall structural organization of the d/b.
  - Option 2: the information about how the database is actually stored in the disk.
  - Option 3: the user view of the database.

Review Question

- Question 3: There are different data models such as
___.

- Question 4: In Network model, each table is made up "tuples" and "fields".
  - True / False

- Question 5: A table can have any number of "Unique constraints".
  - True / False

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved   31