Pre-requisites

12 December 2022 19:53

Lab Setup Requirement	Hardware CPU - Intel Core i3/i5/i7 processor, RAM - at least 8 GB HDD- 512 GB / 1 TB, OS - Windows 10 /8.1/11, MS Word/excel, PowerBI desktop (optional)

Pre-requisites
Software - 1. SQL Server 2016, 2017 or 2019 Enterprise/Developer edition, 2. SQL Server Management Studio/Azure Data Studio, 3. Visual Studio 2019/2022/VS code, (IDE) 4. A Valid Azure Subscription, Azure CLI, Storage explorer, Microsoft Azure Subscription, Git tools and GitHub account, Azure PowerShell, PowerShell ISE, Note: Linux environment VMs (Apache hadoop/big data) (Linux OS) Tool: Putty.exe Azure based Linux servers, Vmware player/ virtual box AWS Tools for VS code, GCP Tools for VS code.

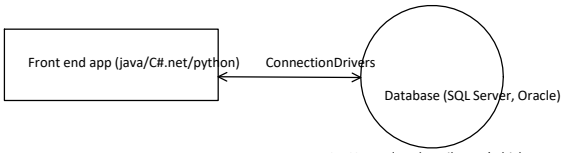
Azure Subscription (trial)
-- 12 months of free service + 30 days of 200 USD credit
-- enterprise subscription

Database Fundamentals and SQL Server BI

12 December 2022 19:54

Application metadata

MSSQL - 1433
MySQL - 3306



1. User related attributes (which users, port, MSSQL - 1433, encryption, protocol (TCP)
2. SQLConnection
3. ADOConnection (ADO.Net)

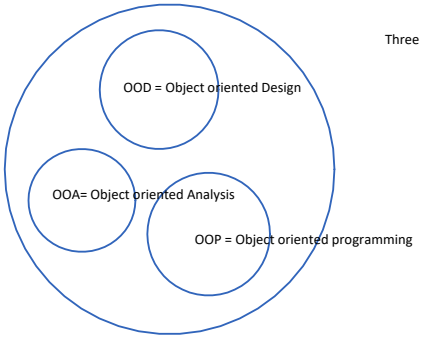
1960 (IBM) - developed the integrated Management system which is based on hierarchical database model.

1970 (IBM) - the relational database model was developed by E.F Codd.

1980 (IBM) - developed the Structured Query Language (SQL). It is declared as the standard language for the queries by ISO (International Standard Organization) and ANSI (American National Standards Institute).

OOPs principles

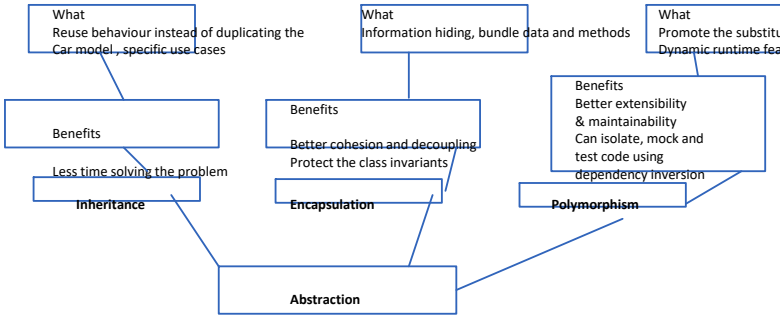
- Abstraction
- Encapsulation
- Inheritance
- Polymorphism



Three pillars of object design

Type Car = {
Types: 'small' | 'medium' | 'heavy'
Color: 'white' | 'black' | 'red'
Full_efficiency: 'efficient' | 'non-efficient'
Price: 'chaper' | 'moderate' | 'expensive'
}

Data Abstraction is the root principle of design.



1. Cohesion - represents a relationship within the module where a single class, has its well-defined purpose.

- It refers to what the class (a module) can do.

a) Low cohesion - low cohesion means a class which can perform a great variety of actions - broad, involves multiple operations.

Contracts - interface, abstract class, types

What
Data Modelling concepts using the contracts and concretions. (objects)

Benefits
Simply design API, simply data design, should
Separate how the data/ abstract can be designed
Declarative vs imperative

Staff Class
checkEmail()
sendEmail()
emailValidate()
PrintLetter()

b) High cohesion - means the class is to be focused on what it should be implementing. It includes only the methods related to the intension of the class.

Staff Class
salary
emailaddr
setSalary (newsalary)
getSalary()
setEmailAddr(newEmail)
getEmailAddr()

Dependency Inversion - forms the pillar of object oriented programming to define as couple the software modules loosely so that one form of objects/ class while changing, doesn't affect others.

Polymorphism - Compile time / Static polymorphism (method overloading, operator overloading)
Runtime / dynamic polymorphism (function overriding)

2. Coupling - refers to the principle of how related or

dependent two classes / modules are toward each other. For low coupled classes, changing something major in one class should not affect other.

e.g. microservices application design

Empaddr and emp classes are separate classes where change of a simple address of a emp entity / object does not affect to the emp class and its related methods.

High coupling scenario - it would make difficult to change or maintain the code, since classes are closely knit together , making a change could require an entire system revamp.

Best practice - good software design should always has **high cohesion and low coupling**.

Coupling definition

In OOD, the coupling refers to the degree of the direct knowledge that one element/ object has of another element/object. How often the changes in class A forces the changes in class B.

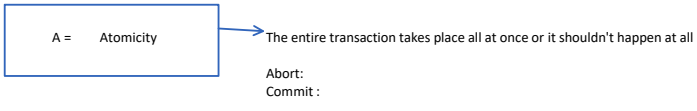
1. Tight coupling - means when two classes often change together, when class A cant be changed directly because it's going to make changes in class B.

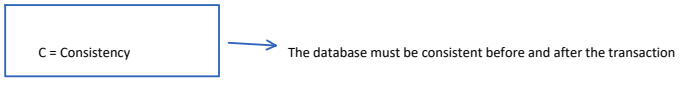
```
Class Subject {
Topic t = new Topic();        // subject class is tightly coupled the topic class
Public void letsRead()
{
 t.understand();
}
}


Class Topic {
 public void undertand()
{
  System.out.println("Tight coupling scenario");
}

}
```

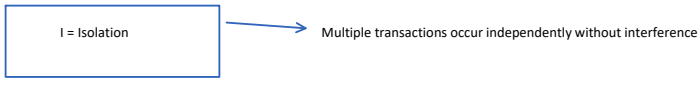
2. Loose coupling - when two classes are just aware of each other , like class A and class B. Also, class B is expose through its interface, then class A and class B are loosely coupled.

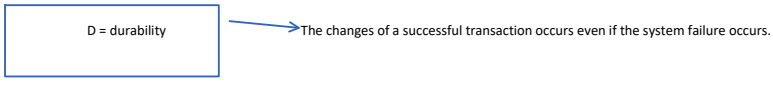
e.g. Microservices application

```
Public interface Topic
{
 void understand();
}

Class Topic1 implements Topic {
Public void undertand()
{
  System.out.println("This is loose coupling example");
}
}
Class Topic2  implements Topic {
Public void understand()
{
System.out.println("This is another loose coupling class");
}
}
Public class Subject{
Public static void main(String[] a)
{
 Topic a  = new Topic1();
t.understand();
}
}
```









Different Types of Databases

1. **Flat file database** -- kind of text database where each line of the plain text file holds only a single record. (e.g. MS Access, MS Excel)
2. **Hierarchical database** -- based on hierarchical data model, where the data is viewed as a collection of tables, data is designed into a tree like structure where each record consists of one parent record and many child record. (e.g. IBM DB2 - IBM Information Management System (IMS) , Windows Registry, XML data storage)
3. **Network model database** - can consists of multiple parent segments and this segments can be grouped together as levels but there's always exists a logical association between the segments belonging to any level.

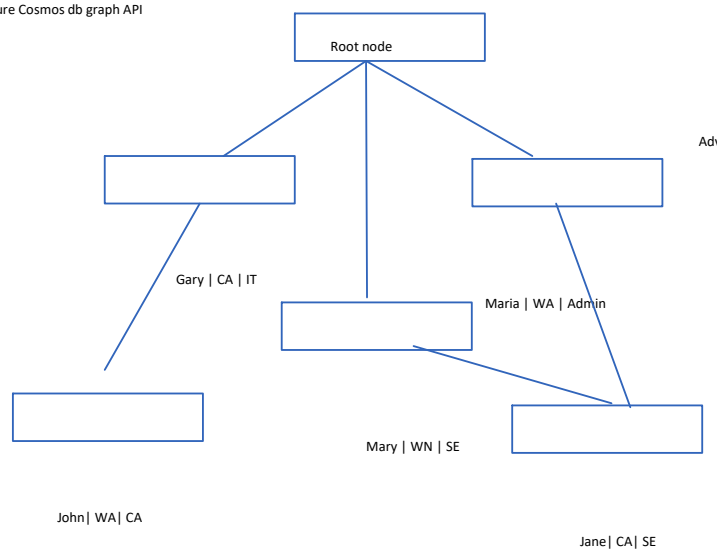
4. **Relational database** - consists of set of tables with columns and rows
5. **Object-oriented database** - information can be represented in the form of object-oriented programming. Inclined towards the objects e.g. multimedia records in a relational database can be definable data object.
   Mongo db is a object-oriented database.
6. **Distributed Database** - Consists of two or more files located in different sites/ location.

e.g. SQL Server mirror databases,

7. **NoSQL databases** - non-relational db has support for unstructured, semi-structured data and it can include dynamic schema, flexible data model for faster data retrieval
   e.g. Mongo db, Cassandra, Couch db, Azure Cosmos db
8. **Graph database** - node - entity (rows in the table), attributes (relationship/columns)

e.g. Neo4j, Azure Cosmos db graph API

Hierarchical data model

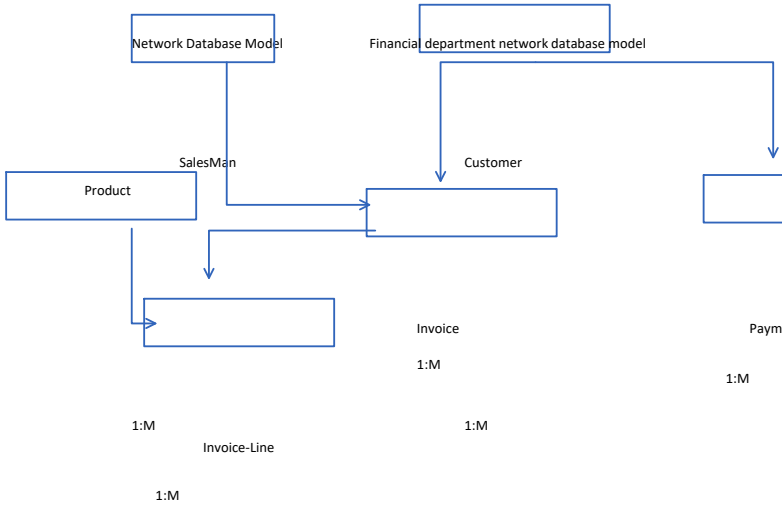


Advantages

- As the database is based on the hierarchical data models, the relationship between various layers are logically simple , it has a very simple hierarchical database structure.
- It has the data sharing facility since all data are held in a common database data and sharing of data becomes practically feasible
- It offers data security and integrity since it's based on parent-child relationship.

Disadvantages:

- Design is simple but implementation is complex
- This model also lacks flexibility as the changes to the new tables or segments often leads to very complex system management tasks.
- It has no standards as the implementation of the model does not provide any specific standard and limited to the relationship s which does not conform to 1:N format.



Advantages

- This model is simple and easy to design compared to hierarchical data model
- This model is capable of handling multiple types of data easily and we can access data easily, we can implement 1:1, 1:M, M:N relationships.

Disadvantages

- The schema of the network database mode is quite complex and records are maintained through pointers.
- The design of the network database mode is not user-friendly
- The model does not have any scope of automated query optimization

Data Dictionary or metadata in DBMS

A data dictionary is an integral part of a database. It holds the information about the database and the data which it stores named as metadata.

Characteristics of data dictionary

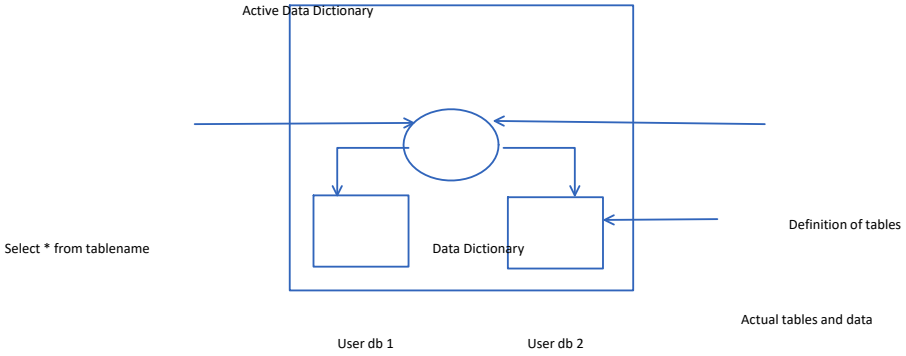
- A metadata is defined as the data about the data
- It's the self-describing in nature of databases
- It holds the information about each data elements in the database
- Such as names, types, ranges of values, access authorization, include the application details / program which uses the data
- Metadata is being used by programmers to develop the application, queries to manage and manipulate the data.

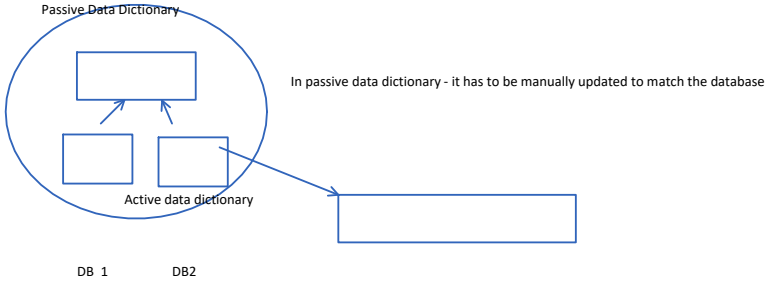
Types of Data Dictionary

1. Active Data Dictionary -- self updating

a) managed automatically by the data management software
b) It's always consistent with the current structure of the database (e.g. RDBMS)

2. Passive Data Dictionary - mainly for documentation purpose
   - Managed by user of the system and is modified manually by the user. (e.g. Dataedo by IBM IMS)





Data considered in DBMS for data Dictionary

Schema
- Tables
-Columns
  - Constraints
  - Foreign keys
  - Indexes
  - Sequences

Programs in SQL
- Views
- Stored Procedures
- User defined Functions (UDFs)
- Triggers

Storage
- Size of tables and indexes stored inside the db
- Number of rows in table

Benefits of data dictionary

1. Improves the data quality
2. Spot the data anomalies
3. Implement transparency and collaboration
4. Get access to the good data
5. Involve regulatory compliance
6. Enables fast and accurate data analysis

OLTP - Online Transaction Processing (SQL server database engine / MSSQL, mysql, oracle)
OLAP - Online Analytical Processing (SQL server datawarehouse)

Employee Table

Fields	Columns/Attributes Emp ID	Columns/Attributes Emp Name	Columns/Attributes Emp Hire date
Rows1 (records/ tuples)	1001	John	10/12/2021
Row 2	1002	Nash	11/12/2021

Foreign Key

Primary Key - Emp_id which uniquely defines the each record in the employee table & help to distinctly identify each employee.

Employee_Address table

Fields	Attributes (emp_id)	Emp_primary_address	Emp_secondary_address
Row1	1001		
Row2	1002		

Primary Key = emp_id

Primary Key definition

A primary key is a column or a set of columns in a table whose values uniquely identifies a row in the table. A relational database is designed to enforce the uniqueness of primary keys by allowing only one row with a given primary key value in a table.

Foreign Key definition

A foreign Key (FK) is a column or combination of columns which is used to establish and enforce a relationship between the data in two tables.

- A Foreign key is a column whose value corresponds to the values of the primary key in another table
- A foreign key is a column or a set of column in table whose values corresponds to the values of the primary key in another table.

- In order to add a row with a given foreign key value, there must exist a row in the related table with the same primary key value.

- Emp_id the foreign key column of employee_address table whose value corresponds to values of the primary key (emp_id) of employee table

SQL Data Warehouse

12 December 2022 19:54

Cloud Fundamentals

12 December 2022 19:54

Azure Fundamentals

12 December 2022 19:55

Basics of PowerShell Scripting

12 December 2022 19:55

Intro to Big Data

12 December 2022 19:55

Apache Hadoop Overview

12 December 2022 19:55

Apache Hadoop (Deep Dive)

12 December 2022 19:56

Azure Data Factory

12 December 2022 19:56

Azure Data Lake Gen2

12 December 2022 19:56

Azure SQL Database

12 December 2022 19:57

Azure Blob Storage

12 December 2022 19:57

Azure Analysis Services

12 December 2022 19:57

Azure Synapse Analytics

12 December 2022 19:57

Case Study

12 December 2022 19:57

Apache Spark

12 December 2022 19:58

Python Programming

12 December 2022 19:58

Azure Databricks

12 December 2022 19:58

Overview of AWS

12 December 2022 19:58

Overview of GCP

12 December 2022 19:58

Case Study

12 December 2022 19:58

L1 Preparation

12 December 2022 19:59