

Object Oriented Database Model

By Afroz Ahmad, KBOPL, Mob. 9006658740

किसी भी सॉफ्टवेयर का विकास करने का एक इंजीनियरिंग दृष्टिकोण (Approach) विधि होता है। यह वस्तु (Object) और वर्ग (Group) का उपयोग करता है और इसमें वस्तुओं का एक वर्ग होता है जो आपस में एक दूसरे को संचार (Communicate) करता है और संदेशों (Message) का आदान प्रदान करता है ताकि लोग एक साधारण (common) वस्तु (Object) को प्राप्त कर सकें।

ऑब्जेक्ट ओरिएंटेड डेटाबेस एक ऐसा मॉडल है जो ऑब्जेक्ट ओरिएंटेड प्रोग्रामिंग के सिद्धांत पर काम करता है। इस डेटाबेस में हम जो संग्रह (Store) करते हैं या दर्शाते (Represent) हैं वह ऑब्जेक्ट या वर्ग (Class) के रूप में करते हैं। इसे हम हाइब्रिड एप्लीकेशन भी कहते हैं और इसमें जितने भी (Entities) होते हैं वह सभी वस्तु के रूप में काम करते हैं। जो भी चीज़ें हमारे पास मौजूद हैं या फिर जो डेटाबेस का हिस्सा हैं वह सभी दरअसल (Entities) हैं।

अगर किसी कॉलेज का डेटाबेस बनाया जा रहा है तो उसके अन्दर जो स्टूडेंट्स हैं वह इकाई (Entity) है और इसमें जितने भी इकाई हैं वह सभी वस्तु (Object) हैं।

इसके अन्दर दी चीज़ें बहुत महत्वपूर्ण होती हैं पहला वस्तु के गुण (properties of object) और दूसरा उसका विधि या कार्य (method or function)। ऑब्जेक्ट ओरिएंटेड डेटाबेस में हर वस्तु एक खास अद्वितीय वस्तु पहचानकर्ता (unique object identifier) से पहचान करते हैं यानि हम एक खास नंबर निर्धारित (Assign) करते हैं जिससे हम उस वस्तु (Object) की पहचान कर सकें।

अब हम उदहारण से इस बात को समझने की कोशिश करते हैं। नीचे एक स्टूडेंट का एक ऑब्जेक्ट बनाया गया है।

Students (Object)	
<ul style="list-style-type: none">• Name• Roll No.• Residential address• Date of birth	(Properties)
<ul style="list-style-type: none">• Written exam• Submitted assignment• Pay fees	(Behavior)

ऊपर बॉक्स में स्टूडेंट्स कि प्रॉपर्टीज है और नीचे उसके व्यवहार (Behavior) है।

हम जानते हैं कि किसी भी सॉफ्टवेयर को बनाने के लिए पूरी टीम होती है। इस टीम के काम में ऑब्जेक्ट ओरिएंटेड डेटाबेस एक महत्वपूर्ण भूमिका निभाता है। जब भी हम कोई सॉफ्टवेर बनाते हैं तो दुनिया की समस्याओं को समाधान करने की कोशिश करते हैं और विश्व की जो वस्तुयें होती हैं इसका सॉफ्टवेर के अन्दर मॉडल तैयार किया जाता है। मॉडल को तैयार करने के लिए हमलोग ऑब्जेक्ट की अवधारणा (Concept) का उपयोग करते हैं। इसमें दो चीज़ें होती हैं।

- State
- Behavior

उदहारण के लिए एक घोडा है जो कि एक ऑब्जेक्ट है उसकी समस्या को सॉफ्टवेर बनाकर समाधान करना चाहते हैं, तो

- Horse state: Name, Color, Breed, Age etc.

- Horse behavior: Sleeping, Running, Eating etc.

हमें उस सॉफ्टवेर के अन्दर घोड़े का वर्णन (State) और व्यवहार (Behavior) को परिभाषित और निर्धारित करना होगा ताकि वर्णन (State) और व्यवहार (Behavior) अच्छी तरह से प्रतिनिधित्व (Represent) हो जाये।

ऑब्जेक्ट ओरिएंटेड डेटाबेस मैनेजमेंट सिस्टम में सभी डेटा को ऑब्जेक्ट के रूप में रखा जाता है। यह डेटाबेस बनाने का दृष्टिकोण (Approach) है। उदहारण के लिए हमें एक बिल्डिंग बनानी है और हमें मालूम है कि बिल्डिंग कैसे बनती है मगर अलग अलग दृष्टिकोण (Approach) से बिल्डिंग को बना सकते हैं उदहारण के लिए किसी देश में लकड़ी से, जापानी तरीके से, इंडियन तरीके से यानि किस तरीके से हम घर बनाना चाहते हैं। जिस जगह जो ज़रूरत होगी उसी प्रकार से घर बनाया जाएगा।

ऑब्जेक्ट ओरिएंटेड डेटाबेस मैनेजमेंट सिस्टम एक डेटाबेस मैनेजमेंट सिस्टम है जिसमें ऑब्जेक्ट ओरिएंटेड प्रोग्रामिंग में उपयोग किये जाने वाली ऑब्जेक्ट के रूप में सूचना का प्रतिनिधित्व किया जाता है और हर ऑब्जेक्ट का एक काम (Function) होता है। उदहारण के लिए मोबाइल एक ऑब्जेक्ट है जिसका काम किसी से बात करना, फोटो लेना आदि होता है और इसकी विशिष्टता (Specification) मोबाइल के स्क्रीन का अकार (Size), बैटरी की लाइफ, प्रोसेसर आदि है।

यह सिस्टम ऑब्जेक्ट के रूप में डेटा के मॉडलिंग और निर्माण का समर्थन करती है। इसमें ऑब्जेक्ट के वर्ग और उपवर्ग और उनकी ऑब्जेक्ट द्वारा वर्ग के गुण (Class property) और कार्यप्रणाली (Method) की विरासत के लिए किसी प्रकार का समर्थन शामिल है।

ऑब्जेक्ट ओरिएंटेड डेटाबेस मैनेजमेंट सिस्टम की विशेषताएँ या सिद्धांत क्या है।

Characteristics / Principles of Object Oriented Database

1. **Inheritance:** यह एक ऐसा तंत्र (Mechanism) है जिसके द्वारा एक वर्ग (Class) के ऑब्जेक्ट या डेटा से दूसरे या कई वर्ग को ऑब्जेक्ट या डेटा को साझा (Share) कर सकते हैं यानि एक वर्ग के डेटा की मदद से दूसरे वर्ग को बनाते हैं। उदहारण के लिए एक सुपर वर्ग है जिसमें जानवर (Animal) के सभी डेटा मौजूद हैं अब हम इससे दूसरे तीन वर्ग बनाते हैं। पहला वर्ग चिड़िया (Bird), दूसरा वर्ग स्तनीय जन्तु (Mammals) और तीसरा रेंगने वाला जन्तु (Reptiles) है। पहले सुपर वर्ग में सभी जानवर मौजूद थे अब चिड़िया (Bird) में सिर्फ चिड़िया के डेटा को सुपर वर्ग से लेकर बनायेंगे। इसी प्रकार से स्तनीय जन्तु (Mammals) और रेंगने वाला जन्तु (Reptiles) का वर्ग भी सुपर वर्ग से इससे सम्बंधित डेटा लेकर बनायेंगे। इसे ही हम इनहेरिटेंस (Inheritance) कहते हैं और इसे सुपर / पैरेंट / या बेस क्लास भी कहते है और इससे बनाये गए वर्ग को Child / Derived क्लास कहते हैं।

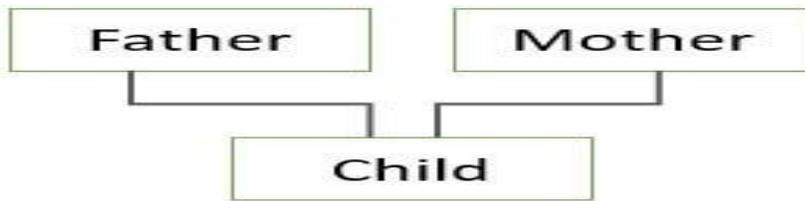
इनहेरिटेंस के प्रकार (Types of inheritance)

- **Single inheritance:** इसमें एक वर्ग से एक वर्ग एक वर्ग को बनाते हैं यानि A वर्ग से B वर्ग को बनाते हैं।



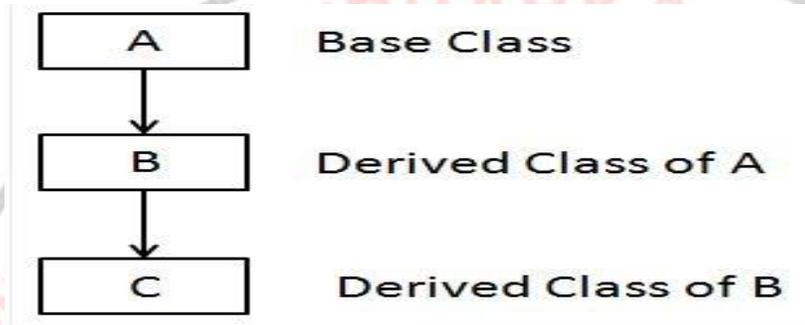
Example of Single inheritance

- **Multiple inheritance:** इसमें एक से अधिक वर्ग मिलाकर एक तीसरा वर्ग बनाते हैं।



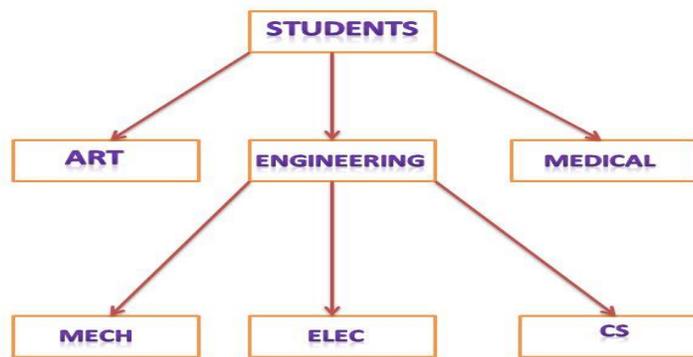
Multiple Inheritance

➤ **Multilevel inheritance:** इसमें एक वर्ग से दूसरा वर्ग और दूसरा वर्ग से तीसरा वर्ग बनाया जाता है।



Example of Multilevel inheritance

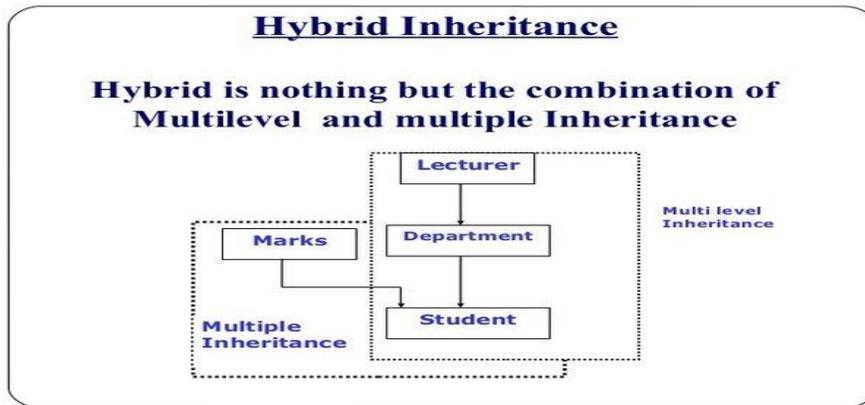
➤ **Hierarchical inheritance:** इसमें जो भी वर्ग पैरेंट है वह पैरेंट ही रहेगा और उसी पैरेंट वर्ग से ही सभी वर्ग बनाये जाते हैं।



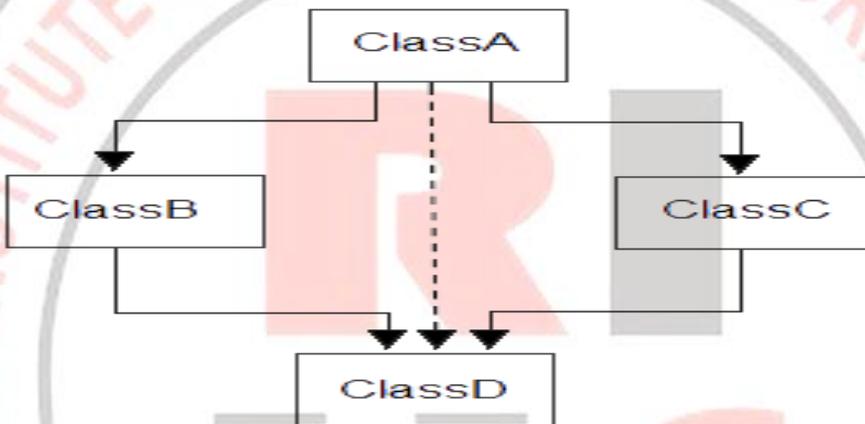
Hierarchical classification of students

Example of Hierarchical inheritance

➤ **Hybrid inheritance:** इसमें सभी इनहेरिटेंस एक दूसरे से जुड़े रहते हैं। इसमें सभी इनहेरिटेंस (Single, Multiple, Hierarchical) आवश्यकता पड़ने पर मिलाये जाते हैं।

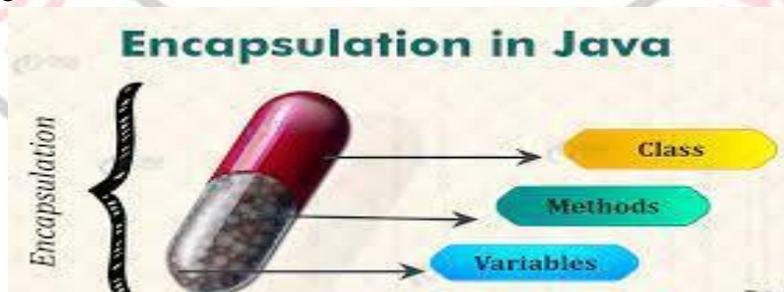


- **Multipath inheritance:** इससे पहले ऊपर के सभी इनहेरिटेंस में हम देखते हैं कि एक रास्ता (Single path) होता है मगर इसमें चारों ओर से रास्ता होता है।



Example of multipath inheritance

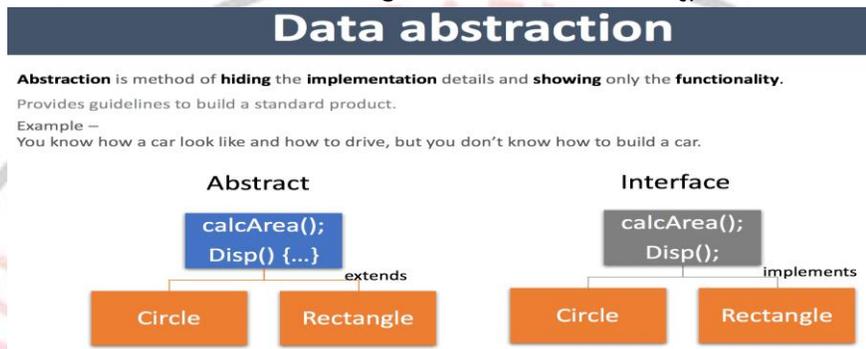
- Encapsulation:** ऑब्जेक्ट ओरिएंटेड डेटाबेस के चार सिद्धांतों में से दूसरा सिद्धांत एन्कैप्सुलेशन है। जैसा कि हम जानते हैं कि एक ऑब्जेक्ट के अन्दर दो चीज़ें होती हैं पहला ऑब्जेक्ट जिसके अन्दर डेटा होता है और दूसरा पद्धति (Method), प्रक्रिया (Process), काम (Function) होता है।
एक यूनिट के अन्दर एक डेटा या उससे सम्बंधित जितने भी पद्धति होती है उन्हें एक यूनिट के अन्दर बाँधना ही एन्कैप्सुलेशन कहलाता है।



Example of encapsulation

- Abstraction:** यह ऑब्जेक्ट ओरिएंटेड डेटाबेस मैनेजमेंट सिस्टम का महत्वपूर्ण अवधारणा (key concept) है। इसका असल उद्देश किसी भी वस्तु जिसमें अनावश्यक (unnecessary) विवरण है या जो जटिलता (Complexity) है उसे यूजर से छुपा कर रखना ही सारांशीकरण (Abstraction) कहलाता है।

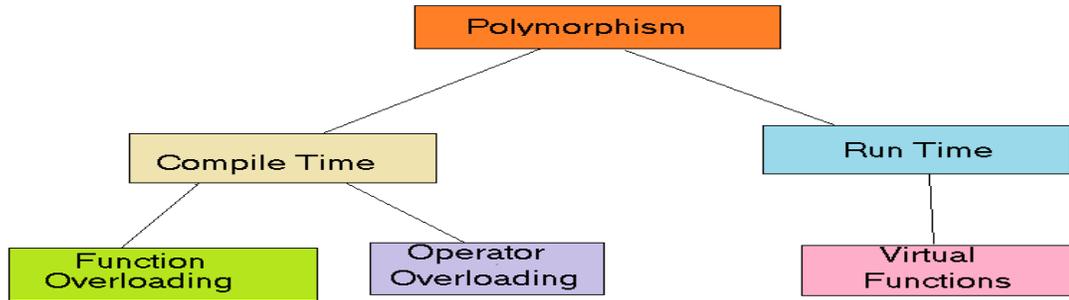
उदहारण के लिए हमारे पास एक मोबाइल होता है और उसे हम बहुत आसानी से उपयोग कर लेते हैं। हमें इसके अन्दर की चीज़ें जानने की ज़रूरत नहीं है इस प्रकार मोबाइल का इंटरफ़ेस सारांशीकरण (Abstraction) उपलब्ध कराता है यानि किसी वस्तु की आन्तरिक जटिलता (Internal complexity) या अनावश्यक विवरण (unnecessary details) को यूजर से छुपा कर रखा जाता है। दूसरा उदहारण इस प्रकार है कि एक कॉफ़ी मशीन के अन्दर की चीज़ों को जानने की ज़रूरत नहीं है बल्कि हमें उसे उपयोग करने के लिए चीनी, दूध और कॉफ़ी को मिलकर मशीन चला देना है। मशीन के अन्दर की जटिलता का सारकरण (Abstraction) कर दिया गया है। इसी प्रकार ऑब्जेक्ट ओरिएंटेड डेटाबेस मैनेजमेंट सिस्टम में सारकरण (Abstraction) के द्वारा उस सभी चीज़ों को छुपा दिया जाता है जिसकी यूजर को आवश्यकता नहीं है।



4. **Polymorphysm:** यह भी ऑब्जेक्ट ओरिएंटेड डेटाबेस मैनेजमेंट सिस्टम या मॉडल का एक सिद्धांत या विशेषता है। यह दो ग्रीक शब्द से मिल कर बना है। पहला पोलि (Poly) जिसका अर्थ है बहुत दूसरा मोर्फिस्म (Morphysm) जिसका अर्थ है रूप या बहुत सारे रूप। पोलिमोर्फिस्म एक ऐसी अवधारणा (Concept) या सिद्धांत (Principle) है जिसमें एक ही वस्तु या ऑब्जेक्ट कई तरीके से प्रदर्शित (Represent) किये जा सकते हैं। वास्तविक जीवन में भी हम देखते हैं कि विभिन्न प्रकार की चीज़ों का कोई समूह किसी एक निर्देश (Instruction) के प्रभाव से विभिन्न प्रकार की प्रतिक्रिया (Reaction) करता है। किसी एक निर्देश का विभिन्न लोगों पर विभिन्न प्रभाव होता है और सभी लोग उस एक निर्देश को अपनी सुविधा के अनुसार समझ कर उसके प्रति अपना प्रतिक्रिया देते हैं। यह प्रतिक्रिया पोलिमोर्फिस्म कहलाती है और डेटाबेस को ज़रूरत के अनुसार प्रदर्शित (Represent) करती है।

अब हम इसे उदहारण से समझने की कोशिश करते हैं कि अगर कोई विश्वविद्यालय यह संदेश देना चाहता है कि एडमिशन के लिए रजिस्ट्रेशन शुरू हो चुका है तो एक संदेश (Message) अपे वेबसाइट या समाचारपत्र में देता है अब हर विषय का छात्र अपने अपने विषय के लिए विभाग में फार्म जमा करता है। हर विषय के छात्र के लिए अलग अलग संदेश नहीं देना पड़ता है बल्कि इस संदेश का विभिन्न विषयों के साथ विश्वविद्यालय में एडमिशन लेने वाले विभिन्न छात्रों पर अलग अलग असर होता है। जो छात्र साइंस में एडमिशन लेना चाहते हैं वह साइंस के फॉर्म को भरता है, जो कंप्यूटर में एडमिशन लेना चाहता है वह कंप्यूटर के लिए फॉर्म भरता है यानि विश्वविद्यालय के एक संदेश को विभिन्न विषय के छात्र विभिन्न प्रकार से प्रतिक्रिया करते हैं।

इसलिए हम कह सकते हैं कि एक क्रिया में कई प्रकार की प्रतिक्रिया होना या एक ही बात का अलग अलग स्थितियों में अलग अलग रूप लेना ही पोलिमोर्फिस्म (Polymorphism) कहलाता है।



Example of Polymorphism

ऑब्जेक्ट ओरिएंटेड डेटाबेस मैनेजमेंट सिस्टम के लाभ (Advantages of OODBMS)

- यह जटिल डेटा के बड़े संग्रह को संभाल सकता है।
- इसमें डिजाईन करना और विकास करना आसान है और इसके कोड को हर जगह कम किया जा सकता है। यह सिस्टम निश्चित स्तर पर डेटाबेस डिजाईन और संचालन (Operation) को आसान बना सकती है।
- काम (Performance) में बड़ी मात्र में सुधार किया जा सकता है। जटिल डेटा के साथ बहुत ही अच्छा प्रदर्शन करता है।

Afroz Ahmad

Khuda Bakhsh Oriental Public Library, Patna

Mob. 9006658740