

DBMS की अवधारणायें

पाठ्यपुस्तक के प्रश्न

वस्तुनिष्ठ प्रश्न

प्रश्न 1. निम्न में से कौन-सा एक DBMS का एक लक्ष्य नहीं है?

- (अ) बड़ी जानकारी का प्रबंध
- (ब) कुशल पुनः प्राप्ति
- (स) Prevention concurrent access
- (द) डेटा की सुरक्षा

उत्तर: (स) Prevention concurrent access

प्रश्न 2. निम्न में से कौन सा एक वाणिज्यिक DBMS का एक उदाहरण है?

- (अ) Oracle
- (ब) IBM
- (स) Sybase
- (द) all

उत्तर: (द) all

प्रश्न 3. निम्न में से कौन सा डेटा abstraction के सरलतम स्तर है?

- (अ) भौतिक
- (ब) तार्किक
- (स) View
- (द) इनमें से कोई भी देखें

उत्तर: (स) View

प्रश्न 4. सामान्यीकरण का मतलब है

- (अ) Joining relations
- (ब) अपघटन के सम्बंध में शामिल
- (स) दोनों
- (द) इनमें से कोई नहीं

उत्तर: (ब) अपघटन के सम्बंध में शामिल

प्रश्न 5. ये normal form अधिक प्रतिबंधित है

- (अ) 1NF
- (ब) 2NF
- (स) BCNF
- (द) 3NF

उत्तर: (स) BCNF

अतिलघु उत्तरीय प्रश्न

प्रश्न 1. DBMS क्या है?

उत्तर- DBMS-एक डेटाबेस प्रबंधन सिस्टम (DBMS) interrelated डेटा का एक संग्रह और उन डेटा तक पहुँच प्राप्त करने के लिए प्रोग्राम्स का एक सेट होता है।

प्रश्न 2. एक रिकॉर्ड को परिभाषित करो।

उत्तर- एक रिकॉर्ड attribute के एक संग्रह का प्रतिनिधित्व करता है जो एक वास्तविक दुनिया की एंटिटी का वर्णन है।

प्रश्न 3. भिन्न डेटा रिडेडेंसी के नाम दें।

उत्तर- डेटा रिडेडेंसी के नाम निम्न Student table से देख सकते हैं

Student Table

Roll_no	Name	Age	Address	Phone	class	Subject
101	Harish	10	Ajmer	1234567891	5th	Hindi
101	Harish	10	Ajmer	1234567891	5th	maths
101	Harish	10	Udaipur	1234567891	5th	Sanskrit
120	Ronak	14	Udaipur	2222222222	8th	Hindi
120	Ronak	14	Udaipur	2222222222	8th	Sanskrit

उपरोक्त स्कीमा का विश्लेषण करके हम आसानी से पा सकते हैं कि यह डिजाइन एक अच्छा डेटाबेस डिजाइन नहीं है।

यह डिजाइन redundant है क्योंकि एक छात्र का name, age, address, class कई बार, दर्ज की गई है जिसमें प्रत्येक विषयों के लिए छात्र पढ़ रहा है। यह redundancy डिजाइन में गंभीर समस्याओं का कारण है। जो कि एक भण्डारण स्थान waste करता है और डेटाबेस में संभावित inconsistency शुरू करता है इसीलिए यह डेटाबेस डिजाइन खराब है।

प्रश्न 4. डेटाबेस स्कीमा को परिभषित करो।

उत्तर- एक स्कीमा (over all डिजाइन) डेटा का, डेटा मॉडल के संदर्भ में एक विवरण है। हम एक स्कीमा के बारे में कह सकते हैं कि डेटा तार्किक (logically) रूप में संरचित किया जा सकता है और शायद ही कभी उसमें परिवर्तन हो, एक स्कीमा इसका एक specification है।

प्रश्न 5. DBMS में इंडेक्सेस की भूमिका क्या है?

उत्तर- DBMS में इंडेक्सेस एक डेटा संरचना तकनीक है जो कुछ विशेषताओं पर आधारित डेटाबेस फाइलों के रिकॉर्ड को प्रभावी तरीके से पुनर्प्राप्ति करती है, जिस पर इंडिक्सिंग हो रही है। यह तीन प्रकार के होते हैं

- प्राइमरी इंडेक्स
- सेकेण्डरी इंडेक्स
- क्लस्टरिंग इंडेक्स

प्रश्न 6. एक केरी भाषा क्या है?

उत्तर- एक DBMS, DBMS से सवाल पूछने के लिए एक विशेष भाषा प्रदान करता है जिसे केरी भाषा कहा जाता है। पुनः डेटाबेस से प्राप्ति के लिए हम केरी भाषा जो DML का हिस्सा है, के द्वारा डेटाबेस से केरी की जाती है। केरी language और DML शब्द पर्याय हैं।

प्रश्न 7. Procedural और Non-procedural DML के बीच क्या अन्तर है?

उत्तर- Procedural DML: इसमें यह आवश्यकता है कि, उपयोगकर्ता द्वारा डेटा और उन डेटा को प्राप्त करने के लिए प्रक्रिया (procedure) को specified किया जाना चाहिए।

Non-procedural DML Non-procedural DML में केवल आवश्यक डेटा उपयोगकर्ता द्वारा specified किया जाता है उन डेटा के प्राप्त करने के लिए की प्रक्रिया को बिना specified करे। एक DBMS, DBMS से सवाल पूछने के लिए एक विशेष भाषा प्रदान करता है जिसे केरी language बुलाया जाता है।

प्रश्न 8. स्कीमा और Instances के बीच क्या अन्तर है?

उत्तर- स्कीमा-एव स्कीमा (over all डिजाइन) डेटा का, डेटा मॉडल के संदर्भ में एक विवरण है। हम एक स्कीमा के बारे में कह सकते हैं कि, डेटा तार्किक (logically) रूप से कैसे संरचित किया जा सकता है और शायद ही कभी उसमें परिवर्तन हो, एक स्कीमा इसका एक specification है। रिलेशनल मॉडल में स्कीमा इस तरह दिखता है:

Relation Name (field1 : type1,, fieldn : typen)
Students (Roll_no: int, name: char, age: integer, class : char)

Instance :

दूसरी ओर एक instance समय के किसी भी विशेष क्षण पर स्कीमा की सामग्री (content) का प्रतिनिधित्व करता है और जो तेजी से बदलता है, लेकिन हमेशा एक स्कीमा के अनुरूप होता है। हम स्कीमा और instances की तुलना एक प्रोग्रामिंग भाषा में type और ऑब्जेक्ट्स के type के साथ कर सकते हैं।

प्रश्न 9. एक डेटाबेस डिजाइन के क्या चरण हैं?

उत्तर- डेटाबेस डिजाइन चरण: एक DBMS application किसी भी उद्यम (enterprise) के लिए डिजाइन करने के लिए कुछ चरणों का पालन करना चाहिए।

आवश्यकताओं के विश्लेषण (analysis): इस चरण में एक enterprise की डेटा आवश्यकताओं की पहचान की जाती है।

वैचारिक डेटाबेस डिजाइन: ज्यादातर ई आर मॉडल का उपयोग कर किया जाता है। वैचारिक स्कीमा का निर्माण करने के लिए चुने हुए डेटा मॉडल की अवधारणा को, पहचान किये हुए डेटा पर लागू किया जाता है।

तार्किक (Logical) डेटाबेस डिजाइन: इस चरण में उच्च स्तर वैचारिक स्कीमा को डेटाबेस सिस्टम के उपयोग होने वाली कार्यान्वयन डेटा मॉडल पर मैप किया जाएगा, जैसे RDBMS के लिए यह संबंधपरक मॉडल है।

स्कीमा (Refinement) : स्कीमा को छोटे स्कीमा में परिष्कृत (refine) करने के लिए सामान्यीकरण (normalization) लागू किया जाता है।

भौतिक (physical) डेटाबेस डिजाइन: यह चरण डेटाबेस के भौतिक सुविधाएँ (physical features) जैसे आंतरिक भण्डारण संरचना, फाइल संगठन आदि निर्दिष्ट करता है।

प्रश्न 10. एक एंटीटी क्या है?

उत्तर- एंटीटी-एंटीटी असली दुनिया में एक "ऑब्जेक्ट" है जिसकी अन्य सभी वस्तुओं से अलग पहचान है। उदाहरण के लिए, एक CLAN, एक teacher, teacher का address, एक student, एक subject। एक एंटीटी का वर्णन attributes के एक सेट का उपयोग कर किया जा सकता है।

लघु उत्तरीय प्रश्न

प्रश्न 1. Atomicity से आप क्या समझते हैं?

उत्तर- Atomicity बताती है कि डेटाबेस संशोधनों को "सभी या कुछ नहीं" ("all or nothing") नियम का पालन करना चाहिए। प्रत्येक लेनदेन (transaction) को atomic कहा जाता है। यदि लेन-देन (transaction) का एक हिस्सा विफल रहता है, तो सम्पूर्ण लेनदेन विफल हो जाता है।

प्रश्न 2. तार्किक और भौतिक डाटा Independence के बीच क्या अन्तर है?

उत्तर- तार्किक (Logical) डेटा Independence

तार्किक डेटा, डेटाबेस के बारे में डेटा अर्थात् यह जानकारी संग्रहीत करता है कि डेटा कैसे प्रबंधित किया जाता है। उदाहरण के लिए एक टेबल (table) और उसकी सभी बाधाएँ (constraints), उस संबंध पर लागू की गई।

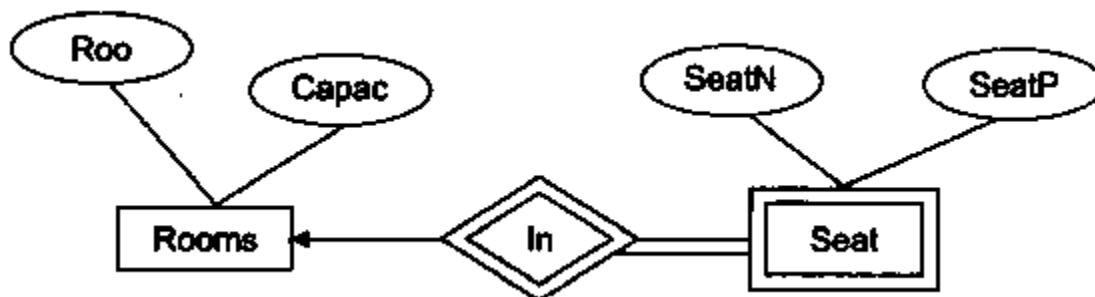
भौतिक (Physical) डेटा Independence

भौतिक डेटा independence, logical डेटा independence को प्रभावित किए बिना भौतिक डेटा को बदलने की शक्ति है।

प्रश्न 3. एक Weak एंटीटी क्या है? यह E-R आरेख में वर्णित कीजिए।

उत्तर- Weak एंटीटी: कभी-कभी, एक एंटीटी सेट E की key को पूरी तरह से अपनी attributes द्वारा गठित नहीं किया जा सकता, लेकिन अन्य एंटीटी सेट (एक या अधिक) की (keys) जिससे E many to many (या one to one) relationship सेट द्वारा लिंक है के द्वारा गठित किया जा सकता है, इसका मतलब है कि एक एंटीटी सेट को अद्वितीय (uniquely) रूप से इस एंटीटी से संबंधित सभी attributes द्वारा नहीं पहचाना जा सकता। ऐसे एंटीटी को Weak एंटीटी कहा जाता है।

उदाहरण: मान लीजिए स्कूल के प्रत्येक कमरे में प्रत्येक छात्र के लिए सीट नंबर है। सीटों की विशेषताएँ (attributes) SeatNo और Seat position हैं। ये विशेषताएँ एक सीट एंटीटी की विशिष्ट पहचान नहीं कर सकते हैं तो सीट एक कमजोर (weak) एंटीटी है।



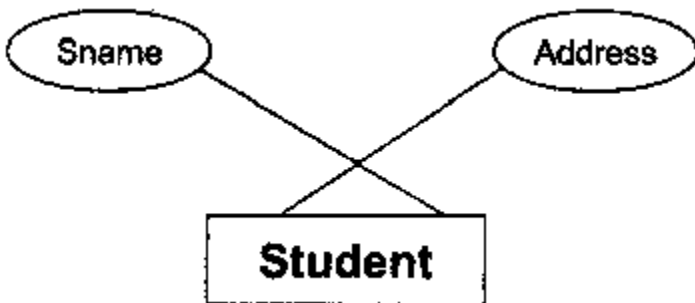
Weak एंटीटी को E-R आरेख में डबल डायमण्ड का उपयोग कर प्रतिनिधित्व किया जा सकता है। किसी

weak एंटीटी के कुछ attributes के किसी अन्य एंटीटी (identifying या owner एंटीटी) की primary key के साथ संयोजन पर विचार करके weak एंटीटी को पहचाना जा सकता है। संबंधित relationship सेट को identifying relationship कहा जाता है।

प्रश्न 4. प्राथमिक और समग्र कुंजी के बीच क्या अन्तर है?

उत्तर- प्राथमिक कुंजी (Primary Key) : किसी relational table की प्राथमिक कुंजी (primary key) टेबल के प्रत्येक रिकॉर्ड को विशिष्ट रूप (uniquely) से बतलाता है। साधारण प्राथमिक कुंजी केवल एक फील्ड (field) से मिलकर बनी होता है। किसी भी टेबल (table) में केवल एक ही प्राथमिक कुंजी हो सकती है।

समग्र कुंजी (Composite Key): दो या अधिक attributes का उपयोग एक key के रूप में कर सकते हैं। जैसे Name or Address अकेले छात्र पहचान नहीं कर सकते हैं लेकिन एक साथ वे एक छात्र पहचान कर सकते हैं।



प्रश्न 5. एक खराब डेटाबेस क्या है?

उत्तर- एक खराब डेटाबेस को समझने के लिए हम निम्नलिखित student table पर विचार करेंगे।

Student						
Roll no	Name	Age	Address	Phone	class	Subject
101	Harish	10	Ajmer	1234567891	5th	Hindi
101	Harish	10	Ajmer	1234567891	5th	maths
101	Harish	10	Udaipur	1234567891	5th	Sanskrit
120	Ronak	14	Udaipur	2222222222	8th	Hindi
120	Ronak	14	Udaipur	2222222222	8th	Sanskrit

उपरोक्त स्कीमा का विश्लेषण करके हम आसानी से पा सकते हैं कि यह डिजाइन एक अच्छा डेटाबेस डिजाइन नहीं है।

यह डिजाइन redundant है क्योंकि एक छात्र का name, age, address, class कई बार, दर्ज की गई है जिसमें प्रत्येक विषयों के लिए छात्र पढ़ रहा है। यह redundancy डिजाइन में गंभीर समस्याओं का कारण

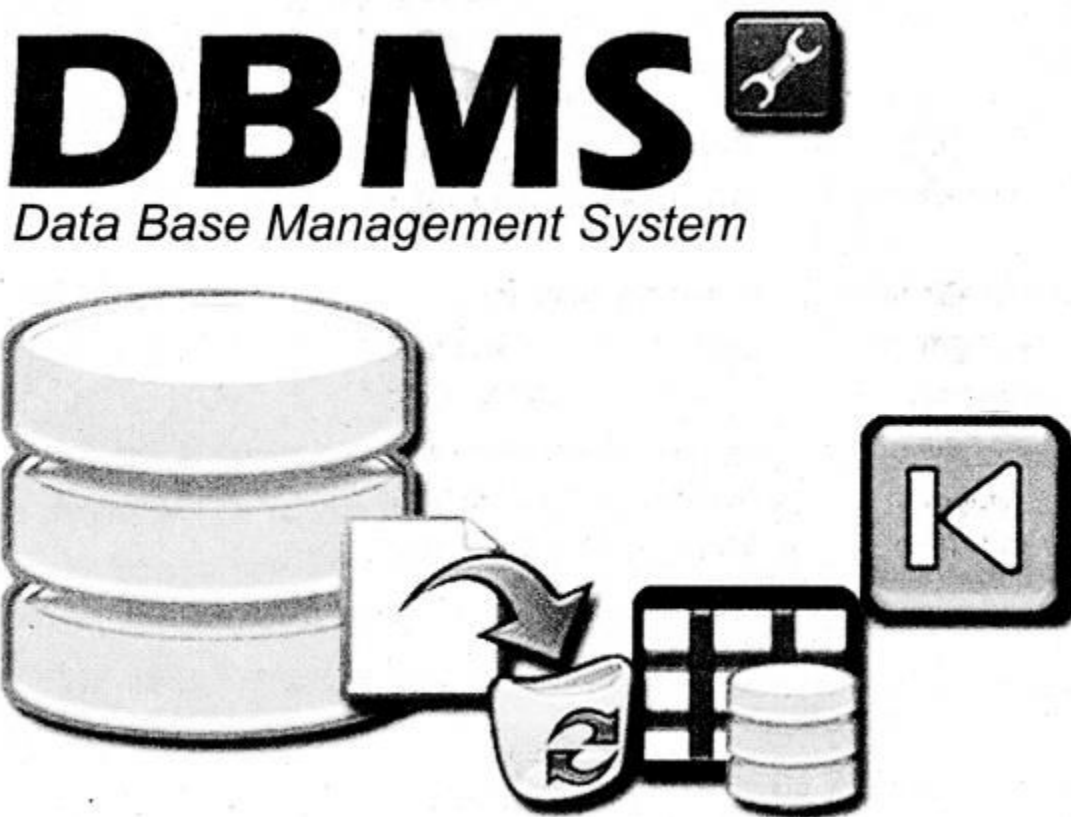
है। जो कि एक भण्डारण स्थान waste करता है और डेटाबेस में संभावित inconsistency शुरू करता है। इसीलिए यह डेटाबेस डिजाइन खराब है।

निबंधात्मक प्रश्न

प्रश्न 1. DBMS के विभिन्न घटक क्या हैं? उपयुक्त चित्र के साथ समझाओ।

उत्तर- DBMS के अनेक घटक होते हैं। जैसे:

(i) सॉफ्टवेयर (Software) : यह सम्पूर्ण डेटाबेस को नियंत्रित करने और प्रतिबंधित करने के लिए उपयोग किए जाने वाले कार्यक्रमों का सेट है। इसमें DBMS सॉफ्टवेयर, ऑपरेटिंग सिस्टम, नेटवर्क सॉफ्टवेयर का प्रयोग उपयोगकर्ताओं के डेटा को साझा करने के लिए उपयोग किए जाने वाले एप्लिकेशन प्रोग्राम शामिल हैं।



(ii) हार्डवेयर (Hardware) : इसमें कम्प्यूटर, आई/ओ डिवाइसेस, स्टोरेज डिवाइसेस आदि जैसे भौतिक इलेक्ट्रॉनिक उपकरणों का एक सेट होता है। यह कम्प्यूटर और वास्तविक दुनिया प्रणालियों के बीच

अंतरफलक प्रदान करता है।



(iii) डेटा (Data) : तथ्य, figures, आँकड़ों के संग्रह को डेटा (Data) कहते हैं।

(iv) प्रक्रिया (Procedure) : ये निर्देश और नियम हैं जो DBMS और डेटाबेस का उपयोग करने, डेटाबेस तैयार करने और प्रबंधित करने में सहायता करते हैं, जो दस्तावेज प्रक्रियाओं का उपयोग करते हैं।

(v) डेटाबेस एक्सेस लैंग्वेज (Database Access Language) : इसका उपयोग डेटा से डेटा तक पहुँचने और नए डेटा दर्ज करने, मौजूदा डेटा को अपडेट करने, या डेटाबेस से आवश्यक डेटा प्राप्त करने के लिए किया जाता है। उपयोगकर्ता डेटाबेस एक्सेस भाषा में उपयुक्त आदेशों का एक सेट लिखता है, इन्हें DBMS तक सबमिट करता है।

(vi) क्वेरी प्रोसेसर (Query Processor) : यह उपयोगकर्ता के प्रश्नों को निम्न स्तर के निर्देशों की एक श्रृंखला में परिवर्तित करता है। यह ऑनलाइन उपयोगकर्ता की क्वेरी को पढ़ता है और निष्पादन के लिए रन-टाइम डेटा मैनेजर को भेजे जाने के सक्षम स्वरूप में एक कुशल श्रृंखला के ऑपरेशनों में इसका अनुवाद करता है।

(vii) रन टाइम डेटाबेस मैनेजर (Run Time Database Manager) : यह DBMS का केन्द्रीय सॉफ्टवेयर घटक होता है जो उपयोगकर्ता द्वारा सबमिट किए गए एप्लिकेशन प्रोग्राम और क्वेरीज के साथ इंटरफेस करता है और रन टाइम पर डेटाबेस पहुँच को संभालता है।

(viii) डेटा मैनेजर (Data Manager) : डेटा मैनेजर, डेटाबेस में डेटा को संभालने के लिए जिम्मेदार होता है। यह सिस्टम को पुनर्घाप्ति प्रदान करता है। जिससे यह विफलता के बाद डेटा को पुनर्घाप्ति करने की अनुमति देता है।

(ix) डेटाबेस इंजन (Database Engine) : यह सबसे अधिक माँग वाली डेटा खपत अनुप्रयोगों की आवश्यकताओं को पूरा करने के लिए नियंत्रित अभिगम और तेजी से लेन-देन प्रसंस्करण प्रदान करता है।

(x) डेटा डिक्शनरी (Data Dictionary) : एक डेटा डिक्शनरी केवल पढ़ने के लिए तालिका और दृश्यों का एक सेट है, जिसमें यह सुनिश्चित करने के लिए उद्यम में प्रयुक्त डेटा के बारे में अलग-अलग सूचनाएं शामिल हैं जो डेटा के डेटाबेस प्रतिनिधित्व को एक मानक के रूप में परिभाषित करता है।

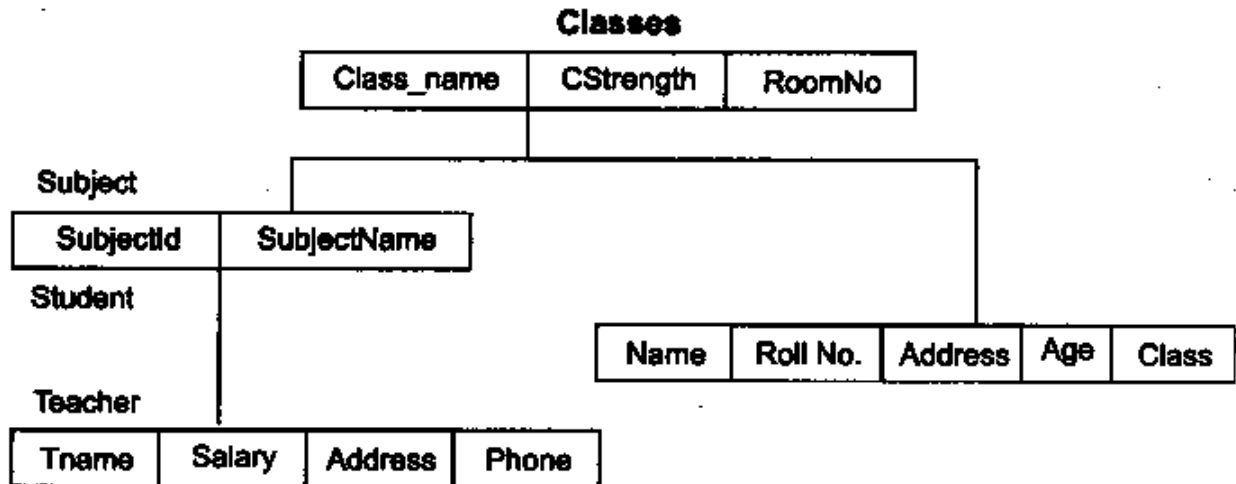
(xi) रिपोर्ट राइटर (Report Writer) : इसे रिपोर्ट जनरेटर के रूप में भी जाना जाता है। यह एक ऐसा प्रोग्राम है जो एक या अधिक फाइलों से जानकारी निकालता है और निर्दिष्ट प्रारूप में जानकारी प्रस्तुत करता है।

प्रश्न 2. एक डेटा मॉडल क्या है? श्रेणीबद्ध डेटा मॉडल समझाइये। यह नेटवर्क डेटा मॉडल से कैसे अलग है?

उत्तर- डेटा मॉडल : एक डेटा मॉडल डेटा, डेटा पर प्रतिबंध और डेटा के अर्थ का वर्णन करने के लिए अवधारणाओं का एक संग्रह है। Hierarchical, नेटवर्क, संबंधपरक और Object-oriented कुछ डेटा मॉडल हैं।

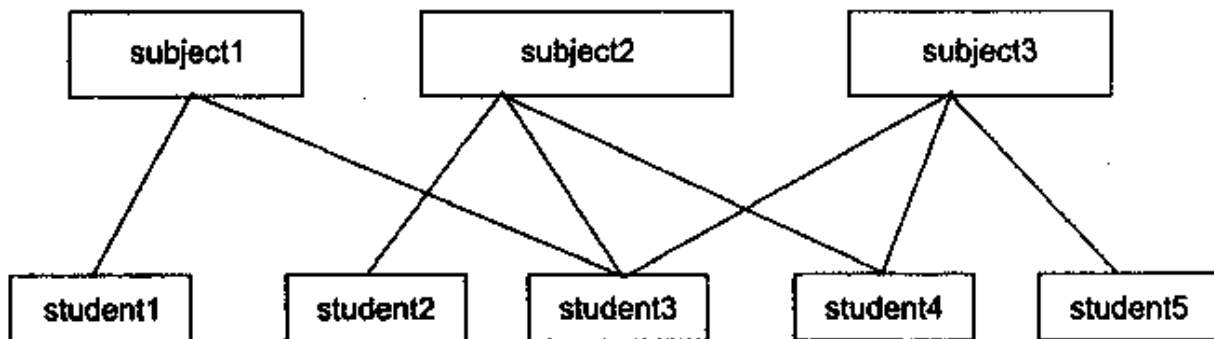
श्रेणीबद्ध (Hierarchical) डेटा मॉडल : हम पदानुक्रमित मॉडल पर आधारित पुराने सिस्टम देख सकते हैं। पहला पदानुक्रमित DBMS "IMS" था और यह 1968 में जारी किया गया था। पदानुक्रमित DBMS एक-से-अनेक (one-to-many relationships) संबंध के मॉडल लिए उपयोग किया जाता है जो डेटा को एक treelike संरचना में उपयोगकर्ताओं के लिए पेश किया जाता है। प्रत्येक रिकॉर्ड के भीतर, डेटातत्व, रिकॉर्ड्स के टुकड़ों में व्यवस्थित होते हैं जिन्हें खण्ड (segments) कहते हैं। उपयोगकर्ता के लिए, प्रत्येक रिकॉर्ड रूट (root) नामक एक toplevel खण्ड के साथ एक संगठनात्मक चार्ट की तरह लगता है। एक ऊपरी खण्ड तार्किक रूप से एक निचले खण्ड से एक पैरेंट-चाइल्ड संबंधों से जुड़ा है। एक पैरेंट खंड के एक से अधिक child हो सकते हैं, लेकिन एक बच्चे के केवल एक पैरेंट हो सकते हैं। नीचे चित्र से एक पदानुक्रमित संरचना का पता चलता है जो स्कूल प्रबंधन प्रणाली (School management system) के लिए इस्तेमाल किया जा सकता है।

रूट खण्ड classes, जिसमें classes की बुनियादी जानकारी जैसे name, strength, और room number है। तुरन्त नीचे दो child segments: subjects (subject id और subject name डेटा युक्त), student (name, address, rollno, age और class डेटा युक्त) हैं। एक subject खण्ड के तुरन्त नीचे teacher (teacher name, salary, address, phone and results evaluations) child segment हैं :



यह पाया है कि बड़ी विरासत सिस्टम (large legacy systems) में जहाँ उच्च मात्रा लेनदेन प्रसंस्करण (high volume transaction processing) की आवश्यकता है श्रेणीबद्ध DBMS अभी भी इस्तेमाल किया जा सकता है। बैंकों, बीमा कंपनियों, और अन्य high volume उपयोगकर्ताओं को विश्वसनीय पदानुक्रम डेटाबेस का उपयोग कर रहा है।

नेटवर्क डेटा मॉडल: एक नेटवर्क DBMS डेटा को अनेक-से-अनेक संबंध (many-to-many relationships) के तार्किक रूप में दर्शाती है। एक नेटवर्क DBMS के लिए एक many-to-many relationships student-subject relationships है (नीचे दिए गए चित्र को देखें)। एक class में कई subjects और कई students हैं। एक student कई subjects लेत ., और कई students एक विषय रखते हैं। पदानुक्रम (Hierarchical) और नेटवर्क DBMS पुराना माना जाता है और अब नए डेटाबेस अनुप्रयोगों (applications) के निर्माण के लिए इन्हें इस्तेमाल नहीं कर रहे हैं।



Student-Subject relationship के लिए नेटवर्क मॉडल

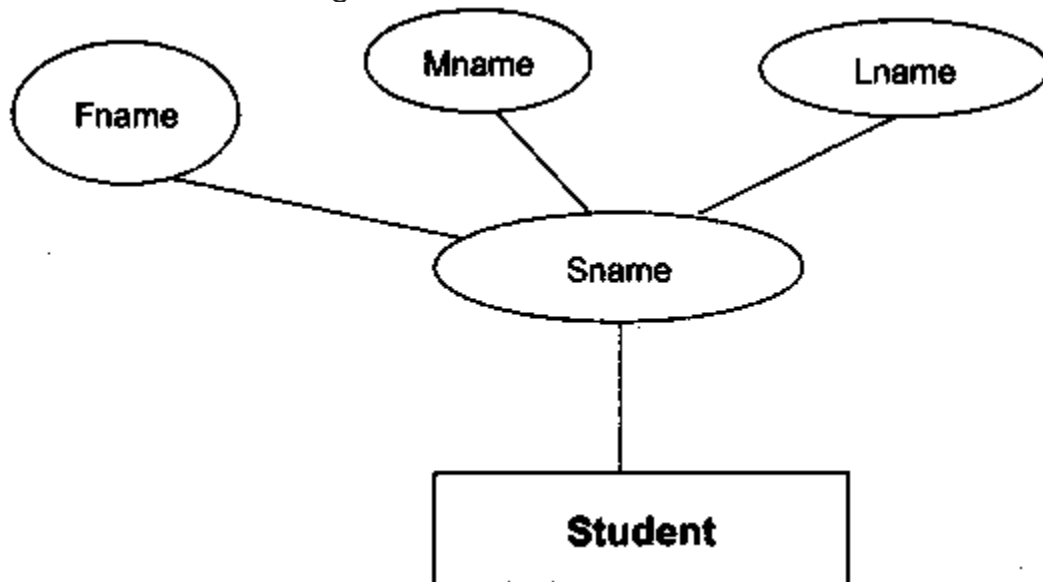
प्रश्न 3. E-R आरेख में विभिन्न प्रकार के attributes और relationship की व्याख्या को समझाए और उनके लिए ग्राफिकल प्रतिनिधित्व भी दें।

उत्तर- Attributes निम्न प्रकार के हो सकते हैं और E-R आरेख में अण्डाकार या दीर्घवृत्त के रूप में प्रतिनिधित्व करते हैं।

साधारण (Simple) attribute : इस प्रकार के attributes का एक एकल मान होता है, जैसे-student एंटीटी में Roll_no और age साधारण attributes हैं।।

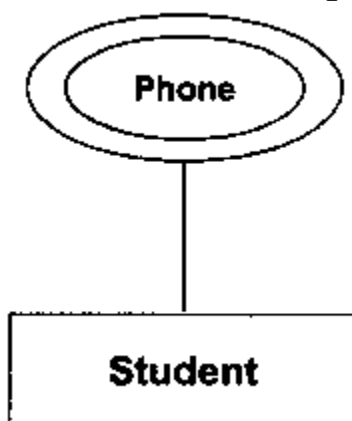
समग्र (Composite) attribute : इस प्रकार के attributes के कई घटक हो सकते हैं, जैसे-Sname attribute में first name, middle name, last name घटक शामिल हैं।

E-R आरेख में ग्राफिकल प्रस्तुति : student एंटीटी और इसका attribute Sname का उदाहरण।



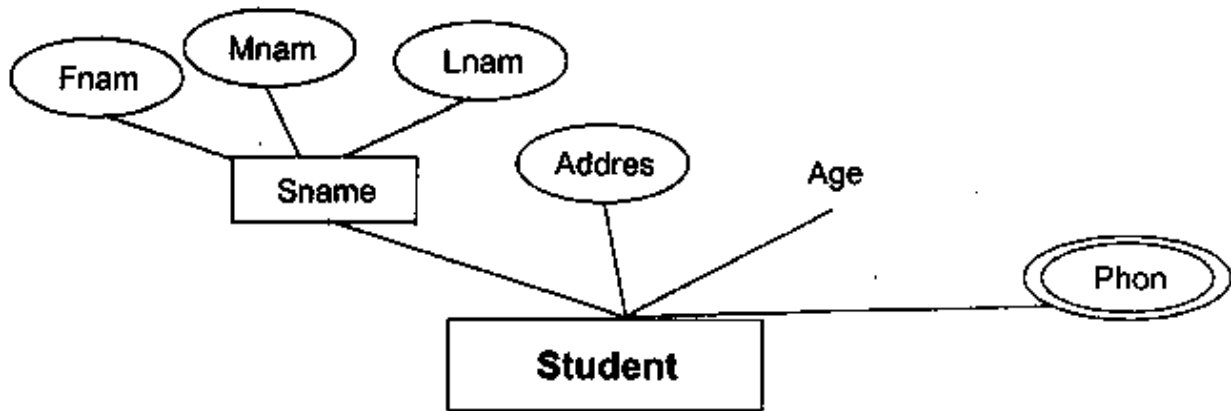
बहु-मान (Multi-valued) attribute : इस तरह की attributes के एक से अधिक मान होते हैं। जैसे-phone attributes के कई मोबाइल नंबर, लैंड लाइन नंबर, कार्यालय नंबर एक से अधिक मान हैं।

E-R आरेख में ग्राफिकल प्रस्तुति: student एंटीटी और इसका attribute Phone का उदाहरण।



Derived attribute: इस तरह के attributes का मान Multi-valued attributes से अभिकलन किया जा सकता है जैसे-उम्र, जन्म दिनांक और वर्तमान दिनांक के डेटा से परिकलित कर सकते हैं।

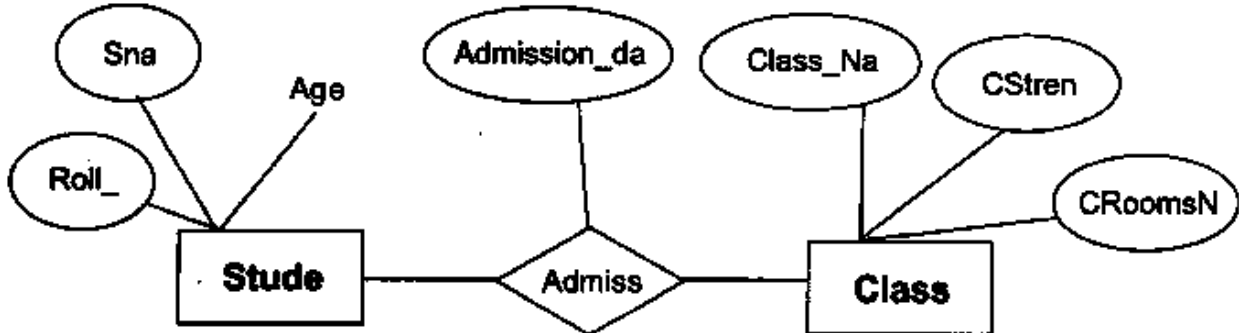
E-R आरेख में ग्राफिकल प्रस्तुति: Composite और Multi-valued attribute के साथ student एंटीटी और इसका attribute age का उदाहरण ।



Relationship: एक Relationship दो या दो से अधिक एंटीटीज के बीच एक संबंध है। relationship जिसमें कि दो एंटीटी सेट शामिल हैं बायनरी (या दो डिग्री) Relationship कहा जाता है।

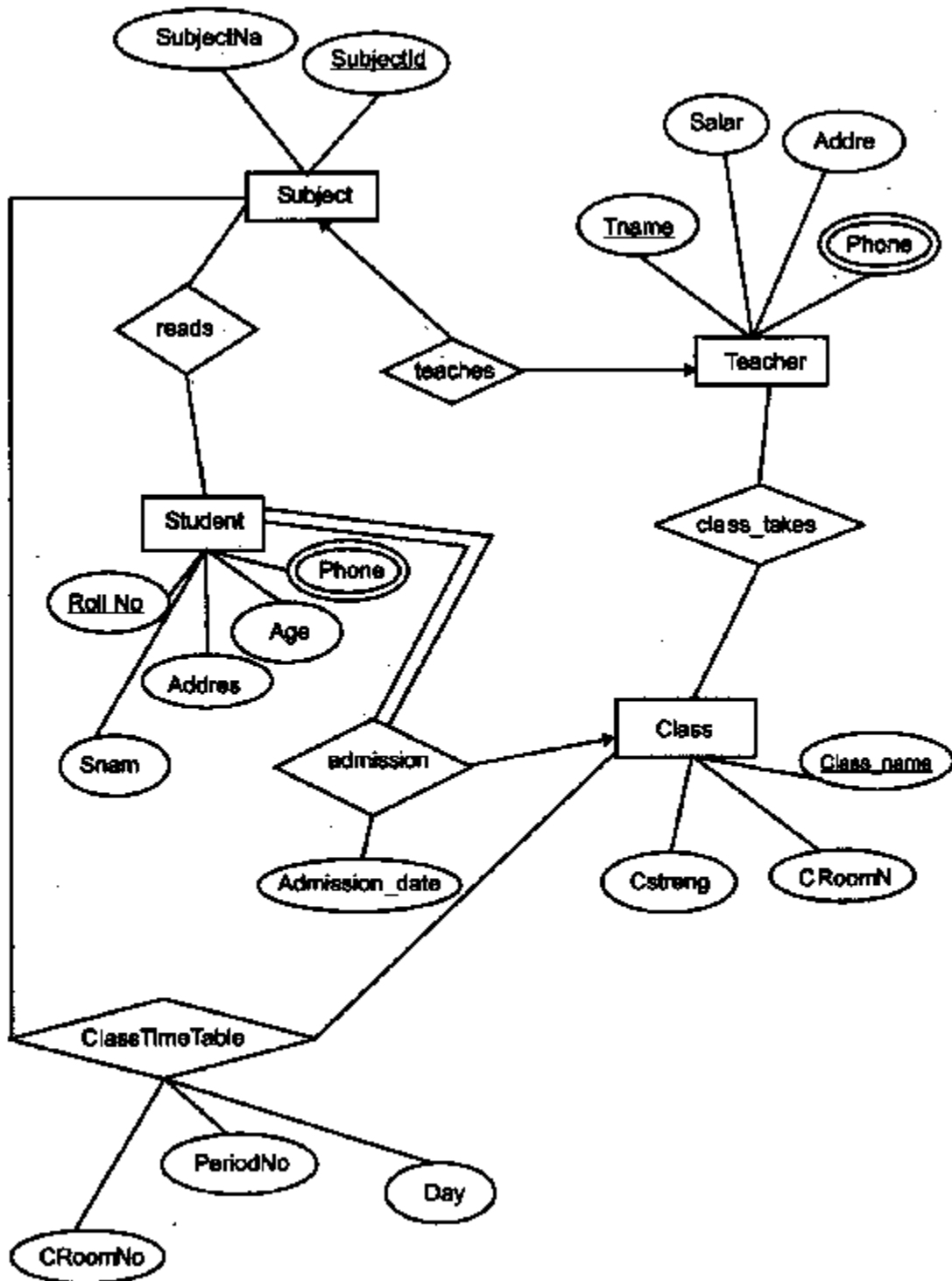
Relationship सेट: एक ही प्रकार के Relationship, student का एक सेट जैसे student का classes में admission यहाँ admission, student और classes एंटीटी सेट के बीच एक Relationship है। इसका E-R आरेख में एक diamond आकार के रूप में प्रतिनिधित्व किया जा सकता है।

E-R ओरेख में ग्राफिकल प्रस्तुति:



प्रश्न 4. एक स्कूल अलग-अलग classes 1st से 10th तक मिलकर बनी है इसके लिए E-R आरेख डिजाइन करो।

उत्तर- स्कूल डेटाबेस का पूरा E-R आरेख :



चित्र : 1 से 10 वीं के एक स्कूल का E-R आरेख

प्रश्न 5. Normalization क्या है? Normalization के विभिन्न रूपों को बताएं।

उत्तर- Normalization :

Normalization डेटाबेस में डेटा के आयोजन की एक प्रक्रिया है जो डेटारिडंडेंसी insertion विसंगति (anomaly), update विसंगति एवं deletion विसंगति को दूर करने के लिए काम करता है। इस प्रक्रिया में हम एक दिए गए रिलेशन स्कीमा की जाँच कुछ Normal forms के विरुद्ध करते हैं यह पता करने कि यह किसी normal form को संतुष्ट करता है या नहीं। यदि एक रिलेशन स्कीमा किसी normal form को संतुष्ट नहीं करता है, तो फिर हम इसे छोटे स्कीमा में विघटित करते हैं।

Normalization मुख्य रूप से दो उद्देश्य के लिए उपयोग किया जाता है,

- अनावश्यक (अनुपयोगी) डेटा को नष्ट करने के लिए।
- यह सुनिश्चित करने के लिए की डेटा dependencies का मतलब है अर्थात् डेटा संग्रहीत तार्किक है।

Normal forms: विभिन्न Normal forms हैं।

1. First Normal Form (1NF)
2. Second Normal Form (2NF)
3. Third Normal Form (3NF)
4. BCNF

First Normal Form (1NF):

यदि प्रत्येक फील्ड केवल Atomic मान रखता है (कोई सूचियाँ नहीं और न ही सेट) तो रिलेशन First Normal Form (1NF) में है।

उदाहरण: चित्र में नीचे दी गई Student table 1NF में नहीं है,

Student			
Roll_no	Name	Age	Subject
101	Harish	10	Hindi, Maths
120	Ronak	14	Maths

चित्र : Student table showing subjects of student not in 1NF

लेकिन नीचे चित्र में 1NF है।

Student			
Roll_no	Name	Age	Subject
101	Harish	10	Hindi
101	Harish	10	Maths
101	Harish	10	Sanskrit
120	Ronak	14	Hindi
120	Ronak	14	Maths

चित्र : Student table showing subjects of student in 1NF

Second normal form: Second normal form के अनुसार किसी भी स्तंभ की primary key पर partial dependency नहीं होनी चाहिए। इसका मतलब है कि किसी table के लिए जिसकी primary key है, table का हर non prime attribute, primary key attribute पर पूरी तरह functionally dependent होना चाहिए। यदि कोई भी स्तंभ केवल primary key के एक भाग पर निर्भर करता है, तो table Second normal form में विफल है।

StudentReads Subject (Roll_no, subjectId, Sname, address, SubjectName)

इस student और subject रिलेशन में primary key attribute Roll_no और subjectId है। नियम के अनुसार non key attributes sname और subjectName दोनों पर निर्भर होना चाहिए लेकिन हम पाते हैं कि Sname को Roll no द्वारा और, subjectName को SubjectId द्वारा स्वतंत्र रूप से पहचाना जा सकता है। यह partial dependency कहा जाता है, जिसकी Second normal form में अनुमति नहीं है।

हमने रिलेशन को दो रिलेशन में तोड़ दिया है। ताकि वहाँ कोई partial dependency मौजूद न हो।

Student (Roll_no, Sname, address)

Subject (SubjectId, Subject Name)

Third Normal Form (3NF)

कोई रिलेशन स्कीमा 3NF में है या नहीं यह चेक करने के लिए हम उस स्कीमा की प्रत्येक FD के लिए निम्न शर्तों को चेक करते हैं। अगर किसी FD के लिए निम्न शर्त फेल होती है तो वह स्कीमा 3NF में नहीं होगा।

शर्तें:

- अगर कोई FD ट्राइवल (trivial) है अर्थात् $\beta \rightarrow A$ में $(A \in \beta)$, है तो या
- अगर किसी FD के बाँयी तरफ के ऐट्रीब्यूट्स उस स्कीमा की key है या
- अगर किसी FD के दाँयी तरफ के ऐट्रीब्यूट्स रिलेशन स्कीमा की key का पार्ट है

BCNF

कोई रिलेशन स्कीमा 3NF में है या नहीं यह चेक करने के लिए हम उस स्कीमा की प्रत्येक FD के लिए निम्न शर्तों को चेक करते हैं। अगर किसी FD के लिए निम्न शर्त फेल होती है तो वह स्कीमा 3NF में नहीं होगा।

शर्तें:

- अगर कोई FD ट्राइवल (trivial) है अर्थात् $\beta \rightarrow A$ में ($A \in \beta$), है तो या
- अगर किसी FD के बाँयी तरफ के ऐट्रीब्यूट्स उस स्कीमा की key है या
- यदि कोई रिलेशन R, BCNF में है तो फिर यह 3NF में भी हो जाएगा। तात्पर्य BCNF implies 3NF but 3NF cannot implies BCNF.

अन्य महत्वपूर्ण प्रश्न

अतिलघु उत्तरीय प्रश्न

प्रश्न 1. फाइल सिस्टम क्या होता है?

उत्तर- फाइलों के एक सेट को स्टोर, पुनः प्राप्त और update करने के लिए एक Abstraction को फाइल सिस्टम कहा जाता है।

प्रश्न 2. डेटा redundancy से आप क्या समझते हैं?

उत्तर- किसी डेटा का बार-बार दोहराव (repetition) ही डेटा redundancy कहलाता है। किसी डेटाबेस में डेटा redundancy का अर्थ है कि कुछ डेटा फील्ड बार-बार repeat हो रहे हैं।

प्रश्न 3. बिट क्या है?

उत्तर- एक बिट डेटा प्रतिनिधित्व की सबसे छोटी इकाई है। एक बिट का मान 0 या 1 हो सकता है।

प्रश्न 4. फील्ड को परिभाषित कीजिए।

उत्तर- एक फील्ड characters का एक समूहीकरण होता है, एक डेटा फील्ड, एक एंटीटी (object, person, place or event) के attributes (एक विशेषता या गुणवत्ता) का प्रतिनिधित्व करता है।

प्रश्न 5. फाइल को परिभाषित कीजिए।

उत्तर- फाइल संबंधित अभिलेखों का एक समूह है। एक फाइल में एक प्राथमिक कुंजी वह फील्ड है जिसका मान एक डेटा फाइल में एक रिकॉर्ड की पहचान करता है।

प्रश्न 6. निम्नलिखित पर संक्षिप्त टिप्पणी लिखिए

(i) डेटाबेस.

- (ii) प्रबंधन प्रणाली
- (iii) DBMS

उत्तर- (i) डेटाबेस: Interrelated डेटा या डेटा फाइलें या संबंध (संबंधपरक डेटाबेस) का संग्रह है। डेटा का यह संग्रह interrelated है तो यह एक संगठन की एक प्रासंगिक जानकारी हो सकती है और जानकारी के इस संग्रह तक application प्रोग्राम के एक सेट का उपयोग करके पहुँचा जा सकता है।

(ii) प्रबंधन प्रणाली (Management system): एक प्रबंधन प्रणाली नियमों का एक सेट और वे प्रक्रियाएँ हैं जो डेटाबेस को बनाने में, हेरफेर करने में और व्यवस्थित करने के लिए हमारी मदद करता है। यह डेटाबेस में डेटा आइटम्स को हटाने, संशोधित करने और जोड़ने के लिए भी हमारी मदद करता है।

(iii) DBMS : एक डेटाबेस प्रबंधन सिस्टम (DBMS) interrelated डेटा का एक संग्रह और उन डेटा तक पहुँच प्राप्त करने के लिए प्रोग्राम्स का एक सेट है।

प्रश्न 7. खण्ड (Segments) क्या होते हैं?

उत्तर- प्रत्येक रिकॉर्ड के भीतर, डेटा तत्व, रिकॉर्ड्स के टुकड़ों में व्यवस्थित होते हैं, जिन्हें खण्ड (Segments) कहते हैं।

प्रश्न 8. Key क्या है?

उत्तर- एक Key न्यूनतम attributes का सेट है जो एक सेट में एक एंटिटी की विशिष्ट रूप से पहचान करता है।

प्रश्न 9. एंटिटी सेट क्या है?

उत्तर- एंटिटी सेट समान ऑब्जेक्ट (उसी प्रकार) के एंटिटीज का एक संग्रह है। एक एंटिटी सेट में, एक एंटिटी का एक ही प्रकार की दूसरे एंटिटी से अन्तर करने के लिए attributes के मान का उपयोग किया जाता है।

प्रश्न 10. बहु-मान (Multi-valued) attribute क्या होते हैं?

उत्तर- बहु-मान (Multi-valued) attribute : इस तरह की attributes के एक से अधिक मान होते हैं जैसे phone attributes के कई मोबाइल नंबर, लैंड-लाइन नंबर, कार्यालय नंबर एक से अधिक मान हैं।

प्रश्न 11. Relationship से आप क्या समझते हैं?

उत्तर- एक Relationship दो या दो से अधिक एंटीटीज के बीच एक संबंध है। Relationship जिसमें कि दो एंटीटी सेट शामिल हैं बायनरी (या दो डिग्री) Relationship कहा जाता है।

प्रश्न 12. Candidate key की परिभाषा बताइये।

उत्तर- Attributes का एक न्यूनतम सेट जो अद्वितीय रूप से एक एंटीटी की पहचान करता है। Candidate key कहा जाता है।

लघु उत्तरीय प्रश्न

प्रश्न 1. फाइल सिस्टम की हानियाँ बताइए।

उत्तर- फाइल सिस्टम की प्रमुख हानियाँ निम्नलिखित हैं

- डेटा redundancy : एक ही सूचना कई फाइलों में उपलब्ध है। जैसे छात्र पता विभिन्न प्रयोजनों के लिए अलग-अलग फाइलों में उपलब्ध है।
- डेटा Access difficulty : इसमें जब नया अनुरोध (request) आता है तब नए प्रोग्राम की आवश्यकता होती है क्योंकि, हर समय नया प्रोग्राम नये अनुरोध को पूर्ण करने के लिए उपलब्ध नहीं होता है इसलिए नये आने वाले अनुरोध को पूर्ण करने के लिए नए प्रोग्राम लिखने की आवश्यकता होती है।
- डेटा isolated है: डेटा अलग-अलग फाइलों में, अलग-अलग स्वरूप में है।
- Multiple उपयोगकर्ता, एक ही डेटा एक साथ उपयोग नहीं कर सकते क्योंकि समानांतर अनुरोध के लिए पर्यवेक्षण में कठिनाई होती है।
- डाटाबेस में सुरक्षा को लागू करने में कठिनाईयाँ आती है।
- अखण्डता संबंधी समस्याएँ (Integrity issues): डेटाबेस पर बाधाओं को सुनिश्चित किया जाना चाहिए।

प्रश्न 2. फाइल सिस्टम के प्रमुख लाभ बताइए।

उत्तर- फाइल सिस्टम के लाभ:

- एकल-अनुप्रयोग (single application) के साथ आसान डिजाइन।
- एक एकल अनुप्रयोग आधारित उन्नत (optimized) संगठन।
- प्रदर्शन में कुशल।

प्रश्न 3. DBMS का प्रमुख लक्ष्य बताइए।

उत्तर- DBMS का लक्ष्य: किसी भी सिस्टम डिजाइन के निम्न लक्ष्य होते हैं

- मुख्य लक्ष्य किसी भी DBMS सिस्टम को बड़ी निकायों की जानकारी का प्रबंध करने के लिए है।
- डेटाबेस में जानकारी संग्रहीत करने के लिए सुविधाजनक तरीका प्रदान करते हैं।
- डेटाबेस में कुशलता से जानकारी की पुनः प्राप्ति।
- डेटाबेस में संग्रहीत जानकारी की सुरक्षा।
- कई उपयोगकर्ताओं द्वारा जानकारी के simultaneous access के दौरान विसंगतियों से बचना।

प्रश्न 4. DBMS के प्रमुख लाभ लिखो।

उत्तर- पारंपरिक फाइल सिस्टम पर DBMS के कई फायदे हैं। ये फायदे DBMS को कई अनुप्रयोगों (applications) में और अधिक उपयोगी बनाते हैं। DBMS के निम्नलिखित लाभ हैं

1. डेटा redundancy (duplicacy) निकालना : एक ही सूचना कई स्थानों में संग्रहीत है, तो संग्रहण स्थान और प्रयास बर्बाद होगा। इस redundancy की समस्या को DBMS ने संभाला है।
2. डेटा sharing : कम्प्यूटरीकृत DBMS में, कई उपयोगकर्ता एक ही डेटाबेस को साझा कर सकते हैं।
3. डेटा Integrity : हम डेटा integrity को डेटा integrity constraints specification द्वारा बनाए रख सकते हैं, जोकि डेटाबेस में किस प्रकार का डेटा दर्ज हो सकता है और manipulate हो सकता है के बारे नियम और प्रतिबंध हैं। यह डेटाबेस की विश्वसनीयता को बढ़ाता है क्योंकि यह गारंटी देता है कि समय के किसी भी बिन्दु पर डेटाबेस में मौजूद डेटा गलत नहीं हो सकता है।
4. डाटा Independence: application प्रोग्राम जितना संभव हो डेटा प्रतिनिधित्व और भण्डारण (storage) के विवरण से independent होना चाहिए। application कोड को ऐसे विवरण से insulate करने के लिए DBMS डेटा का कोई abstract view प्रदान कर सकते हैं।
5. कुशल डेटा पहुँच (Efficient data access): एक DBMS कुशलता से डेटा को पुनर्माप्त करने के लिए और डेटा संग्रहीत करने के लिए कई किस्म की परिष्कृत (sophisticated) तकनीक का इस्तेमाल कर सकते हैं। अगर डेटा बाह्य भण्डारण उपकरणों पर है, तो यह सुविधा विशेष रूप से महत्वपूर्ण हैं।
6. डेटा अखण्डता (Integrity) और सुरक्षा : डेटा का abstract view और integrity constraints उपलब्ध कराने के द्वारा सुरक्षा को सुनिश्चित किया जा सकता है। DBMS डेटा का abstract view प्रदान करता है ताकि सभी प्रकार के उपयोगकर्ताओं को सभी प्रकार की जानकारी देखने की जरूरत नहीं हो। उपयोगकर्ता द्वारा केवल डेटाबेस के विशेष भाग को देखा जा सकता है।
7. Reduced application development time: DBMS स्पष्ट रूप से कई महत्वपूर्ण फंक्शन का समर्थन करता है जो DBMS में संग्रहीत डेटा तक पहुँचने के लिए काम आने वाली ऐसी कई applications के लिए common हैं। यह डेटा के उच्च स्तरीय इंटरफेस के साथ, applications के त्वरित विकास में सुविधा के लिए है।
8. Recovery in DBMS: Transaction की विफलता के दौरान डेटाबेस उसकी मूल स्थिति में पुनर्स्थापित हो जाएगा।

प्रश्न 5. DBMS के विभिन्न Applications बताइए। DBMS के कुछ उदाहरण भी दीजिए।

उत्तर- Applications of DBMS: लगभग सभी क्षेत्र में, DBMS की applications हैं। इनमें से कुछ हैं बैंकिंग:

- बैंकिंग क्षेत्र के सभी लेन-देन।
- एयरलाइन: reservation, schedules, availability
- विश्वविद्यालय: पंजीकरण, ग्रेड।
- बिक्री: ग्राहकों, उत्पादों, खरीद।
- निर्माण: उत्पादन, माल, आदेश, आपूर्ति श्रृंखला।
- मानव संसाधन: कर्मचारी अभिलेखों, वेतन, कर की कटौती।

DBMS का उदाहरण : DBMS जो वर्तमान में उपयोग में हैं।

वाणिज्यिक DBMS:

कम्पनी – उत्पाद

Oracle – 8i, 9i, 10g

IBM – DB2, यूनिवर्सल सर्वर

Microsoft – Access, SQL सर्वर

Sybase – Adaptive सर्वर

Informix – डायनेमिक सर्वर

प्रश्न 6. संबंधपरक (रिलेशन) डेटा मॉडल के विषय में बताइए।

उत्तर- संबंधपरक (रिलेशनल) डेटा मॉडल:

यह एक रिकॉर्ड आधारित डेटा मॉडल है। यह डेटा और डेटा के बीच संबंध का प्रतिनिधित्व करने के लिए रिलेशन (या table) का एक संग्रह का उपयोग करता है। हर रिलेशन attributes (या स्तंभ) की एक सूची रखता है जो अद्वितीय (unique) नाम रखते हैं। प्रत्येक attributes का एक डोमेन (या प्रकार) होता है। प्रत्येक टपल संबंध की प्रत्येक attribute के लिए एक मूल्य (value) रखता है। यहाँ डुप्लिकेट टपलस की अनुमति नहीं होती है। यह, ज्यादातर वर्तमान डेटाबेस सिस्टम के द्वारा प्रयुक्त डेटा मॉडल है। नीचे दिए गए table, student table का एक उदाहरण है जो student विवरण दिखाता है।

उदाहरण

Student table				
Roll no	Name	Age	Address	class
101	Harish	10	Ajmer	5th
105	Kailash	20	Kota	10th
109	Manish	18	Ahmadabad	9th
120	Ronak	14	Udaipur	8th
135	Shanker	13	Jaipur	7th

चित्र: Student डेटाबेस के लिए संबंधपरक डेटा मॉडल

प्रश्न 7. एक डेटाबेस को डिजाइन करने के लिए किन चरणों का पालन करना चाहिए?

अथवा

डेटाबेस डिजाइन चरण के विषय में बताइए।

उत्तर- एक डेटाबेस को डिजाइन करने के लिए निम्न चरणों का पालन करना चाहिए

आवश्यकताओं के विश्लेषण (Analysis): इस चरण में एक enterprise की डेटा आवश्यकताओं की पहचान की जाती है।

वैचारिक डेटाबेस डिजाइन: ज्यादातर ईआर मॉडल का उपयोग कर किया जाता है। वैचारिक स्कीमा का निर्माण करने के लिए चुने हुए डेटा मॉडल की अवधारणा को, पहचान किये हुए डेटा पर लागू किया जाता है।

तार्किक (Logical) डेटाबेस डिजाइन : इस चरण में उच्च स्तर वैचारिक स्कीमा को डेटाबेस सिस्टम के उपयोग होने वाली कार्यान्वयन डेटा मॉडल पर मैप किया जाएगा, जैसे RDBMS के लिए यह संबंधपरक मॉडल है।

स्कीमा Refinement : स्कीमा को छोटे स्कीमा में परिष्कृत (refine) करने के लिए सामान्यीकरण (normalization) लागू किया जाता है।

भौतिक (Physical) डेटाबेस डिजाइन : यह चरण डेटाबेस के भौतिक सुविधाएँ (physical features) जैसे आंतरिक भंडारण संरचना, फाइल संगठन आदि निर्दिष्ट करता है।

प्रश्न 8. एक खराब डेटाबेस डिजाइन डेटाबेस पर कारवाई के दौरान आने वाली problems (Anomalies) के कारण बताइए।

उत्तर- एक खराब डेटाबेस डिजाइन डेटाबेस पर कारवाई के दौरान भी कई problems (Anomalies) का कारण है।

Update विसंगतियाँ : यदि एक प्रतिलिपि Update होती है तब सब दोहराये डेटा की भी Update की जरूरत होती है। उदाहरण के लिए हम एक विशेष छात्र का पता Update करने के लिए हम उस छात्र के सभी tuples Update करते हैं।

Insertion विसंगतियाँ : जब तक असंबंधित जानकारी संग्रहीत है तब तक कुछ डेटा संग्रहीत नहीं हो सकता है। किसी टपल को सम्मिलित (Insert) करने के लिए फोन पता करने की आवश्यकता है यह एक

रिक्त (null) मान के साथ तय हो सकता है लेकिन null समस्याओं के कारण हैं या हैंडल करने में मुश्किल है।

Deletion विसंगतियाँ: कुछ अन्य, असंबंधित जानकारी खोने के बिना कुछ जानकारी को हटाना संभव नहीं हो सकता है। अगर हम सभी tuples एक दिए गए (class, Roll no) के लिए हटाएँ तो हम उस एसोसिएशन को खो सकते हैं। यदि हम डिजाइनों में redundancy अच्छा डेटाबेस को डिजाइन करना चाहते हैं, तो हम Dependencies, decomposition और normal forms के उपयोग करते हैं।

निबंधात्मक प्रश्न

प्रश्न 1. डेटाबेस अब्स्ट्रक्शन क्या होता है? DBMS में विभिन्न levels के विषय में बताइए।

अथवा

DBMS में abstraction levels के विषय में बताइए।

उत्तर- डेटाबेस अब्स्ट्रक्शन (Database Abstraction)

डेटाबेस सिस्टम उपयोगकर्ताओं को केवल उनकी आवश्यकतानुसार डेटा उपलब्ध करवाता है और मैमोरी में कैसे डेटा संग्रहीत और प्रबंधित होता है, की जानकारी छुपाता है। यही प्रक्रिया डेटाबेस अब्स्ट्रक्शन कहलाती है।

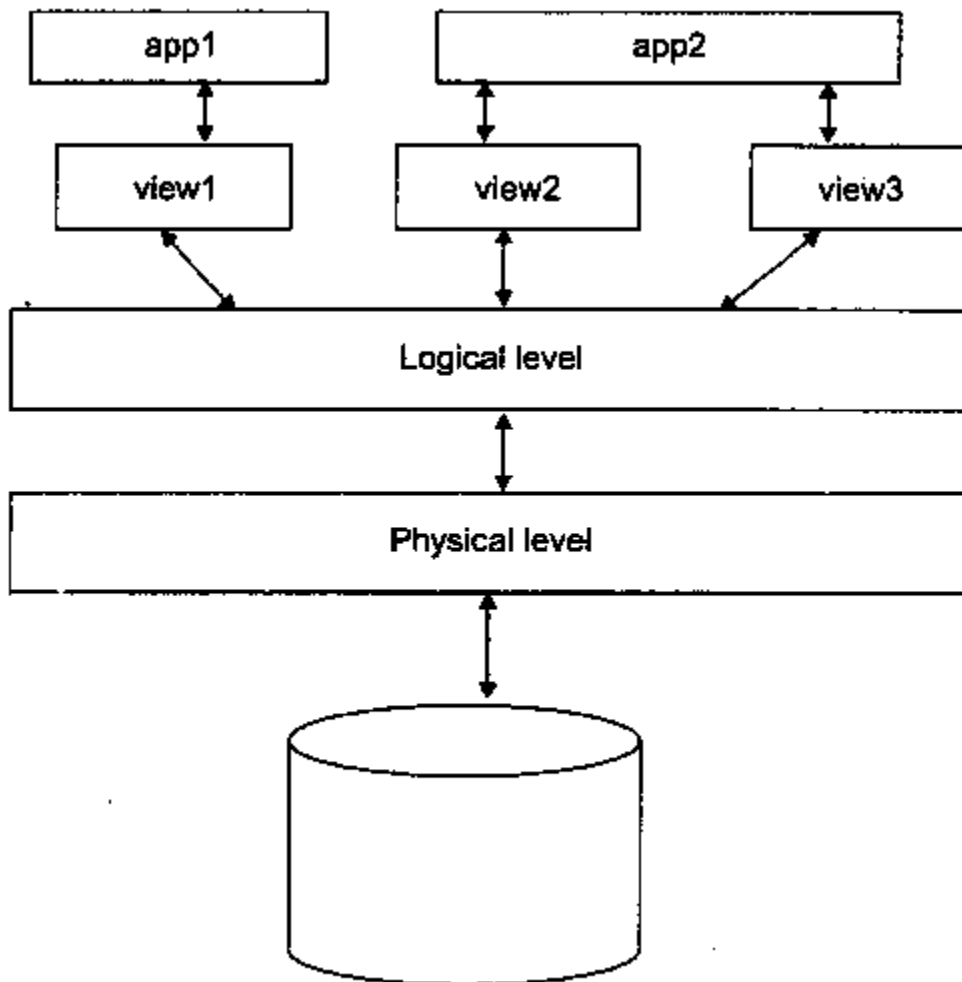
DBMS में abstraction levels: किसी भी DBMS का मुख्य लक्ष्य डेटाबेस के साथ उपयोगकर्ता की interaction को सरल रूप में बनाने के लिए है किसी भी तरह के उपयोगकर्ता (naive, programmers, sophisticated आदि) किसी भी तरह आसानी से और कुशलता से डेटाबेस से जानकारी प्राप्त कर सकते हैं। डेटा कैसे संग्रहीत और बनाए रखा (maintained) है जैसे कुछ विवरण को छुपाने के लिए डेटा का abstract view मदद करता है।

Physical level: भौतिक स्कीमा निर्दिष्ट (specifies) करता है कि कैसे रिलेशन वास्तव में द्वितीयक संग्रह डिवाइस में संग्रहीत की जाती हैं। यह रिलेशन की गति बढ़ाने के लिए इस्तेमाल सहायक डेटा संरचनाएँ (इंडेक्स) को भी निर्दिष्ट करता है।

Logical (वैचारिक) level: वैचारिक स्कीमा डेटाबेस में संग्रहीत डेटा और उन डेटा के बीच संबंध का वर्णन करता है, वैचारिक स्कीमा पूरे डेटाबेस के तार्किक संरचना को परिभाषित करता है। उदाहरण के लिए, एक रिलेशनल डेटाबेस में यह डेटाबेस में संग्रहीत सभी संबंधों का वर्णन करता है।

View level : यह वैचारिक स्कीमा का एक शोधन (refinement) है। यह व्यक्तिगत उपयोगकर्ताओं या उपयोगकर्ताओं के समूह के लिए अधिकृत पहुँच और अनुकूलित की अनुमति देता है। हर डेटाबेस एक वैचारिक और एक भौतिक स्कीमा रखते हैं, लेकिन यह view level पर कई स्कीमा रख सकता है। एक

view (external स्कीमा) conceptual एक रिलेशन है, लेकिन इसके रिकॉर्ड डेटाबेस में संग्रहीत नहीं हैं। इसके बजाय, वे अन्य संबंधों से अभिकलन (computed) हैं। चित्र abstraction के इन स्तरों के बीच संबंध दिखाता है।




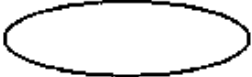
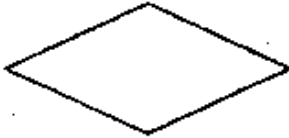
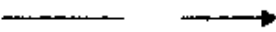


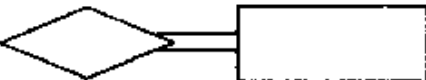
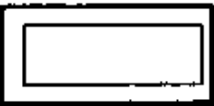
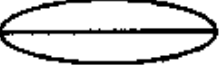

चित्र : डेटा abstraction levels

प्रश्न 2. E-R आरेख को परिभाषित कीजिए। इसके मुख्य घटक प्रतीक सहित बताइए।

अथवा

E-R आरेख के विभिन्न प्रतीक व उनके उद्देश्यों का वर्णन कीजिए। .

उत्तर- E-R आरेख: यह मूल रूप से एक डेटाबेस की सम्पूर्ण तार्किक संरचना का चित्रमय प्रतिनिधित्व है। इस आरेख में मुख्य घटक निम्नानुसार हैं

प्रतीक (symbol) का नाम	प्रतीक	उद्देश्य
आयत		एंटिटी के सेट का प्रतिनिधित्व करता है
दीर्घवृत्त		attributes का प्रतिनिधित्व करता है
Diamonds		relationships का प्रतिनिधित्व करता है
Lines		एंटिटी के सेट और एंटिटी के attributes और एंटिटी के सेट और relationship सेट के बीच संबंध का प्रतिनिधित्व करता है
डबल दीर्घवृत्त		Multivalued attributes
Dashed दीर्घवृत्त		derived attributes का प्रतिनिधित्व करता है
डबल लाइन		Total भागीदारी
डबल आयत		कमजोर एंटिटी सेट का प्रतिनिधित्व करता है
लाइन के साथ एक दीर्घवृत्त		Primary key
Double diamond		Identifying relationship कमजोर एंटिटी के लिए

एंटिटी: असली दुनिया में एक "ऑब्जेक्ट" है जिसकी अन्य सभी वस्तुओं से अलग पहचान है।

उदाहरण के लिए, एक class, एक teacher, teacher का address, एक student, एक subject। एक एंटिटी का वर्णन attributes के एक सेट का उपयोग कर किया जा सकता है। प्रत्येक attribute संभव मानों का एक डोमेन रखता है।

प्रश्न 3. मैपिंग कार्डिनलिटी बाधा कितने प्रकार की होती है? वर्णन कीजिए।

उत्तर- मैपिंग कार्डिनलिटी बाधा: बायनरी (या दो डिग्री) Relationship R के लिए मैपिंग कार्डिनलिटी बाधा दो एंटिटी सेट e1 और e2 के बीच निम्न प्रकार की हो सकती है।

Many to one: e1 में प्रत्येक एंटीटी e2 में 0 या 1 एंटीटी से संबंधित है, लेकिन e2 में प्रत्येक एंटीटी e1 में प्रत्येक एंटीटी e1 में 0 या और अधिक से संबंधित है।

उदाहरण: एक student एक ही समय में एक class में हो सकता है लेकिन एक class में कई छात्र हो सकते हैं। E-R आरेख में ग्राफिकल प्रस्तुति: रेखाओं का उपयोग कर।



Many to many : e1 में प्रत्येक एंटीटी e2 में 0 या अधिक एंटीटी से संबंधित है और इसके विपरीत।
उदाहरण : कई teachers एक class को पढ़ा सकते हैं या कई classes एक ही teacher द्वारा पढ़ा सकते हैं।



One to one : e1 में प्रत्येक एंटीटी e2 में 0 या 1 एंटीटी से संबंधित है और इसके विपरीत।
उदाहरण : एक teacher केवल एक subject जैसे Hindi, English, Maths सिखा सकते हैं। यह मान है कि स्कूल में प्रत्येक विषय के लिए विशेष शिक्षक निर्धारित है।



One to many : e2 में एक एंटीटी, e1 में at most one एंटीटी के साथ संबंधित है।
उदाहरण: विशेष मामले में विषय विशेष शिक्षक स्कूल में उपलब्ध नहीं है, तो एक teacher भी कई विषय सिखा सकता है।



प्रश्न 4. Functional dependencies को उदाहरण सहित समझाइए।

उत्तर- Functional dependencies: एक कार्यात्मक निर्भरता (Functional dependencies या FD) IC का एक प्रकार है जो key की अवधारणा को generalizes करता है। मान लो R एक रिलेशन स्कीमा है, X और Y रिलेशन R के nonempty attributes का सेट है तो R के instance के लिए हम कहते हैं कि FD (X कार्यात्मक निर्धारित करता है Y) संतुष्ट है: अगर

$$\forall t_1, t_2 \in r, t_1.X = t_2.X \Rightarrow t_1.Y = t_2.Y$$

$X \rightarrow Y$ इसका मतलब है कि जब भी R में दो tuples X में सभी attributes पर सहमत हैं, तो Y में भी सभी attributes पर सहमत होना होगा।

एक कार्यात्मक निर्भरता का उदाहरण:

कार्यात्मक निर्भरता $AB \rightarrow C$ के लिए निम्न instance संतुष्ट हैं

A	B	C	D
a1	b1	c1	d1
a1	b1	c1	d2
a1	b2	c2	d1
a2	b1	c3	d1

एक primary key एक FD की विशेष स्थिति है: अगर $X \rightarrow Y$ रखती है (जहाँ Y सभी विशेषताओं (attributes) का सेट है, तो X एक superkey है।

FDs रिलेशन की किसी भी instance के लिए रखना चाहिए। यदि FDs का एक सेट दिया है तो हम आमतौर पर अतिरिक्त FDs भी पा सकते हैं।

उदाहरण : यदि एक key दी है, तो हम हमेशा एक superkey पा सकते हैं।

FDs के उपयोग द्वारा keys को पुनर्परिभाषित करना :

K, attributes का एक सेट है जो रिलेशन R के लिए एक key है। यदि $K \rightarrow$ (अन्य) सभी attributes R के, अर्थात् K एक " superkey key है।

उपरोक्त शर्त को K का कोई उचित सब सेट संतुष्ट नहीं करता है, अर्थात् K कम से कम है।