बी.सी.एस.टी.

MCRPSV

कनक कम्प्यूटर एजुकेशन

मारवनलाल चतुर्वेदी विश्वविद्यालय से संबद्ध

Mob: 9589995353, 9926326401

website :- www.bcstkanak.in

DCA
1 वर्षीय

PGDCA
1 वर्षीय

कृष्णा टाकिज के पीछे, राजत भवन के सामने मुलताई

डाटाबेस यूजिंग एमएस-एक्सेस

Narendra Publication

Narendra Publication

PGDCA Semester -1

हिंदी संस्करण

Author : Narendra Tiwari

डाटाबेस यूजिंग एमएस-एक्सेस

New Syllabus 2018

DCA

MS Access

Unit 1

Introduction to database- what is database, why use a relational database, overview of database design-, data normalization(determining table, determining fields, determining relationships) integrity rules (primary/ foreign key, one to many, many to many, one two one) introduction to MS Access(objects, navigation)

Unit -1

डाटाबेस के मूल तत्व

डाटा :– डाटा ऐसे तथ्य है, जिसमें से कुछ निष्कर्ष निकलता है। किसी संस्था के कार्य मे डाटा एक बहुत आवश्यक अंग है।

फाइल :– फाइल एक समान रिकार्ड का संग्रह है। इसमे उपयोगकर्ता की आवश्यकता के अनुसार रिकार्ड डाल सकते है।

DBMS यह संबंधित डाटा का संग्रह है, तथा प्रोग्राम का समूह है, जिसमे डाटा डाला जाता है। डाटा का संग्रह यह साधारणतः डाटाबेस के नाम से जाना जाता है। जिसमे संस्था से संबंधित जानकारी होती है। DBMS का मुख्य उद्देश्य ऐसी प्रणाली बनाना है जिसमे डाटा से संबंधित जानकारी आसानी से तथा प्रभावशाली ढंग से संग्रहित कर आवश्यकतानुसार प्राप्त किया जा सके। डाटा बेस प्रणाली को सूचनाओं के बहुत बडे भाग को नियंत्रित करने के लिए बनाया जाता है। किसी कंपनी के व्यक्तिगत डाटाबेस मे कर्मचारियों की विस्तृत जानकारी होती है। जैसे कर्मचारी का नाम, परीचय क्रमांक, पद, उम्र, वेतन आदि। सूचनाओ की ऐसी व्यक्तिगत इकाई की जानकारी जैसे कर्मचारी का नाम, क्रमांक आदि को फील्ड कहते है। प्रत्येक फील्ड के अंदर character का समूह रहता है, जो वर्णमाला, संख्या तथा सिम्बाल हो सकते है, जिससे हमे अर्थपूर्ण जानकारी मिलती है। वास्तविक डाटा को रिकार्ड कहा जाता है। फील्ड जानकारी की मूल इकाई होती है। इन संबंधित फील्डों मे डाली गयी जानकारी के समूह से एक रिकार्ड बनता है। डाटाबेस संबंधित रिकार्डो का व्यवस्थित ढंग

से बनाया हुआ संग्रह है। कम्प्यूटर का प्रयोग कर जानकारीयों को बहुत अच्छे तरीके से संग्रहित एवं प्रयोग किया जा सकता है। कम्प्यूटर मे इस प्रकार के कार्य के लिए data base management system प्रोग्राम उपलब्ध है, इसे DBMS भी कहा जाता है। DBMS यह डाटा के व्यवस्थापन का योजनाबद्ध तरीका है, उदाहरण Foxpro, Oracle, Access

अच्छा DBMS प्रोग्राम उपयोगकार्ता को प्रभावशाली ढंग से डाटा बेस बनाने तथा सुधार करने की सुविधा प्रदान करता है। आप डाटाबेस का ढॉचा एक्सेस मे मेन्यु प्रणाली या कमांड के द्वारा बना सकते है, तथा उपलब्ध ढांचे का प्रयोग कर नया डाटाबेस बना सकते है। जिसमे फील्ड के नाम, प्रकार इत्यादि निश्चित कर सकते है।

DBMS का अवलोकन

1. एक साफ्टवेयर जो एक या अधिक उपयोगकर्ता को डाटा उपलब्ध कराता है, उसे DBMS कहते है।
2. इसमे संबंधित डाटा एवं प्रोग्राम का समूह होता है।
3. डाटा का समूह डाटाबेस कहलाता है, जो निश्चित डाटा के बारे मे सूचना रखता है।
4. DBMS का मुख्य उद्देश्य, आसानी से तथा प्रभावी ढंग से इच्छित डाटा को प्राप्त करने की सुविधा प्रदान करना है।
5. DBMS के चार मुख्य घटक होते है। 1. हार्डवेयर 2 सॉफ्टवेयर 3. डाटा 4. उपयोगकर्ता

DBMS के गुण

1. केंद्रिय नियंत्रण :– इसका केंद्रिय नियंत्रण डाटाबेस एडमिनिस्टेटर (DBA) द्वारा होता है। यह डाटा के बदलाव, सुधार एव संग्रहण के लिए जिम्मेदार होता है। यह अधिकृत व्यक्ति होता है, जो किसी उपयोगकर्ता को प्रणाली प्रयोग करने की अनुमति देता है।
2. पुनरावर्त्ती को कम करना :– यह सुविधा DBA द्वारा आपको मिलती है, जिसमे कि डाटा के पुनरावर्त्ती को कम किया जा सकता है।
3. डाटा की आत्मनिर्भरता :– DBMS मे प्रत्येक डाटा स्वतंत्र होता है, अर्थात किसी एक डाटा मे परीवर्तन से दूसरे डाटा पर कोई प्रभाव नही होता है। या दूसरे डाटा पर निश्चत प्रभाव दिया जा सकता है। यह दो प्रकार से होती है।
   a. डाटा की तार्किक स्वतंत्रता:– इसमे प्रोग्राम को परीवर्तन किए बिना भी डाटा मे बदलाव किया जा सकता।
   b. डाटा का भौतिकी स्वतंत्रता :– इसमे डाटा मे परिवर्तन करने पर डाटा की संपूर्ण संरचना या प्रोग्राम पर कोई प्रभाव नही पडता है।
4. डाटा का साझा करना :– डाटा बेस विभिन्न प्रयोगकर्ता या प्रोग्राम को डाटा का साझा करने की अनुमति प्रदान करता है। वर्तमान मे इंटरनेट नेटवर्क का प्रयोग एक ही डाटा को विकल्प जगहो पर स्थित व्यक्ति प्रयोग कर सकते है।
5. डाटा की सम्पूर्णता :– डाटा की संपूर्णता से यह अभिप्राय है, डाटा बेस मे उपस्थित डाटा एकदम विशुद्ध एवं स्थाई होना चाहिये। जब एक डाटाबेस जिसमे बहुत से प्रयोगकर्ता काम कर

रहे हो तब यह आवश्यक होता है, की कोई भी डाटा तथा उनके बीच का सम्बन्ध नष्ट नही होना चाहिए। DBA यह निश्चित करता है कि डाटा की संपूर्णता के लिए पर्याप्त साधन उपलब्ध है या नही।

6. विरोधाभास को समाप्त करना :– डाटा बेस यह DBA के नियंत्रण मे होता है। प्रयोगकर्ता तथा अनुप्रयोगों के बीच के विरोधाभास को कम करता है।

7. डाटा की सुरक्षाः– डाटा की सुरक्षा यह DBA करता है। डाटा किसी संस्था के लिए बहुत महत्वपूर्ण होता है। इसलिए इन डाटा का अंवाछित उपयोगकर्ता से सुरक्षा करना आवश्यक होता है। DBA प्रत्येक प्रयोगकर्ता को प्रवेश के लिए एक नाम एवं पासवर्ड देता है, जिसकी जानकारी सिर्फ उस उपयोगकर्ता को ही होती है।

DBMS की कमीयॉं (Disadvantages)

DBMS क्रियान्वन के समय होने वाले दोष निम्न प्रकार के है

1. केंद्रियकरण (centralizations) से संबंधित समस्या।
2. हार्डवेअर तथा सॉफ्टवेअर के खरीदी एवं विकास मे अधिक लागत लगना।
3. डाटा को पुनः उसी स्थिति मे लाना यह बहुत जटिल कार्य है। यदि एक से अधिक प्रयोगकर्ता एक डाटाबेस को प्रयोग कर रहे है, तब डाटा प्राप्त करने की गती कम हो सकती है।
4. डाटाबेस मे कुछ समस्या आती है, तब संपूर्ण डाटा खराब होने की संभावना बनी रहती है।
5. डाटा को पुनः उसी स्थिति मे लाना यह बहुत जटिल कार्य है।

DBMS के उपयोग

बैंक :–ग्राहक की जानकारी रखना, बैंक के खाते, लोन एंव बैंक के लेन देन कि जानकारी रखना।

यातायात सेवा :– आरक्षण एवं यातायात से संबंधित सारणी कि सूचनाओं के लिए। विमान सेवा मे डाटा बेस सर्वप्रथम भौगोलिक वितरण प्रणाली के रूप मे प्रयुक्त किया गया था। विश्व के दूसरे कोने मे इसकी सूचना केंद्रिय डाटा बेस तंत्र मे फोन लाइन द्वारा पहुँचती है।

संचार प्रणाली :– किए गये कॉल कि जानकारी, महीने का बिल, prepaid कार्ड के नियंत्रण एवं संचार तंत्र की सूचना रखने के लिए DBMS का प्रयोग होता है।

वित्त व्यवस्था :– व्यापार मे वित्तीय स्थिति का व्यवस्थापन जैसे खरीदी, ब्रिक्रि का रिकार्ड रखना। कंपनी की वित्तिय स्थिति जैसे जमापूंजी, माल का संग्रह आदि कि जानकारी देना।

बिक्रीः– ग्राहक, उत्पाद एंव खरीदी की सूचना रखने के लिए ।

निर्माण कार्य :– वस्तुओं कि बिक्री, प्रबंध समिती एवं कारखानों, गोदाम में उपस्थित समूहों का लेखाजोखा रखने के लिए प्रयुक्त होता है।

व्यक्तिगत उपयोग :– कर्मचारी, वेतन, कर, एवं लाभ आदि में प्रयोग होता है।

Views of database

डाटा बेस प्रणाली एक दूसरे से संबंधित फाइलों एवं प्रोग्राम के समूह का संग्रह है, जो उपयोगकर्ता को इन फाइलों मे प्रवेश करने एवं उनमे परिवर्तन करने कि अनुमति देता है। डाटा बेस का मुख्य उद्देश्य डाटा को अलग–अलग ढंग से उपयोगकर्ता के सामने प्रदर्शित करना है। सिस्टम हाइड (system hides) के द्वारा हम डाटा का संग्रहण एंव रखरखाव कैसे करें यह जानकारी मिलती है।

डाटा का संक्षेपीकरण (data abstraction) :– जो प्रणाली प्रयोग हो रही है, उसमे डाटा को कुशलता पुर्वक रखरखाव आवश्यक है। डाटाबेस के जरूरत के अनुसार डाटाबेस मे डाटा को प्रदर्शित करने के लिए डिजाइनर को जटिल डाटा संरचना का प्रयोग करना पडता है।

क्योकि सभी उपयोगकर्ता कम्प्यूटर एवं प्रोग्रामींग मे दक्ष नही होते है, अतः डिजाइनर यहा संक्षेप्तीकरण के बहुतसे स्तरों के माध्यम से इस जटिलता को उपयोगकर्ता से छिपाता है। जिससे उपयोगकर्ता तंत्र के साथ आसानी से जुड जाता है।

Physical level
Logical level
View level

Physical level:- संक्षेपीकरण का यह सबसे निचला स्तर है। जो डाटा के वास्तविक संग्रह के बारे मे जानकारी देता है। सबसे निचले स्तर पर डाटा की संरचना किस प्रकार होगी यह जानकारी देता है।

Logical Level:- यह स्तर, डाटा बेस मे डाटा कैसे संग्रहित होना चाहिये, तथा उनके बीच कैसा सम्बन्ध होना चाहिए इसका वर्णन करता है। अतः logical level वर्णित करता है कि संपूर्ण डाटाबेस मे सम्बन्धित संरचना कि प्रणाली क्या होगी, DBA जो कि यह निश्चित

करता है, डाटा बेस मे क्या रखना है, जिसका प्रयोग लॉजिकल लेवल मे होता है।

View level:- संक्षेपीकरण सबसे ऊपरी सतह संपूर्ण डाटा बेस के कुछ भाग का वर्णन करती है। यद्यपी logical level समान्य संरचना का प्रयोग करता है तब भी इसमे जटिलता बनी रहती है। क्योकि बहुत प्रकार कि सूचनांये एक बडे डाटा बेस मे संग्रहित रहती है। डाटा बेस प्रणाली के बहुत से उपयोगकर्ता को सभी सूचनाओं कि आवश्यकता नही होती है, उन्हे डाटा बेस के केवल एक भाग कि आवश्यकता होती है। View level उन उपयोगकार्ता को प्रणाली के साथ सरलता से समांजस्य बनाने मे सुविधा प्रदान करता है। प्रणाली एक ही डाटा बेस के लिए बहुतसे *view* देती है। view level मे उपयोगकर्ता ऐसे प्रोग्राम को रखता है, जिसमे डाटा के प्रकार की सभी जानकारीयाँ होती हैं। इस प्रकार view level मे डाटाबेस के बहुत से view परिभाषित रहते है। एवं डाटा बेस उपयोगकर्ता इन view को देखता है। view level सुरक्षा प्रदान करता है, जो उपयोगकर्ता को कुछ भाग मे पहुचंने से रोकता है। उदाहरण बैक का अकाउण्टंट डाटा बेस के केवल उसी भाग को देख सकता है, जिसमे ग्राहक के खातों कि सूचनांये है। वह कर्मचारी के वेतनमान से सम्बन्धित सूचनांओ को नही देख सकता।

Schema and Subschema

डाटा बेस मे निरंतर सूचनाओं को डालने तथा अनावश्यक डाटा मिटाने से डाटाबेस परिवर्तित होते रहता है। एक निश्चित समय मे डाटाबेस मे सूचनांओ के संग्रहण को डाटाबेस का Instance कहा जाता है। डाटा बेस की संपूर्ण संरचना को डाटा बेस schema कहते है। तथा schema मे परिवर्तन हो सकता है। डाटा बेस की

schema के तथ्यों को ऐसे प्रोग्राम जो प्रोग्रामिंग भाषा मे लिखे गये है उनसे समझ सकते है। एक डाटा बेस कि schema यह variable declaration के समान है। किसी दिए हुए समय मे प्रत्येक Variable का एक निश्चित मान होता है। डाटा बेस प्रणाली मे निम्न schema होती है।

1. Physical schema :- यह डाटाबेस के डिजाइन के भौतिक लेवल का वर्णन करती है।
2. Logical schema :- यह डाटाबेस के डिजाइन के लॉजीकल लेवल को वर्णित करती है।
3. sub-schema :- डाटा बेस मे बहुत सी रूपरेखा view level मे होती है। इन schema को कई बार sub-schema कहते है। जो डाटा बेस के अलग–अलग view को वर्णित करती है।

Concept of Data model

डाटा बेस की संरचना डाटा मॉडल है। डाटा, डाटा रिलेशन, डाटा semantics को वर्णित करने के लिए प्रयुक्त conceptual tools का समूह एक डाटा मॉडल है।

डाटा मॉडल के तथ्य को समझने के लिए हम दो डाटा मॉडल को इस खण्ड मे रेखांकित करेगे।

1. Entity relationship model (ER model)
2. Relational model

यह दोनों मॉडल logical level डाटा बेस को डिजाइन करने का रास्ता दिखाते है।

Entity Relationship Model (E-R Model)

यह मॉडल मूल आब्जेक्ट के समूह पर आधारित है, जिसे **entity** कहते है, एवं यह इन आब्जेक्ट के सम्बन्धों पर भी आधारित है।

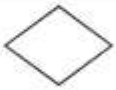
Entity एक आब्जेक्ट है, जो अन्य आब्जेक्ट से भिन्नता रखती है। उदाहरण प्रत्येक व्यक्ति एक **entity** है। **entity** को डाटा बेस मे एट्रीब्युट के समूह द्वारा वर्णित कर सकते है। उदाहरण एट्रीब्युट खाता क्रंमाक एवं कुल जमा, बैक के किसी एक खाते को वर्णित करते है। एवं वह **entity account** के एट्रीब्युट को बनाती है। इस प्रकार एट्रीब्युट, ग्राहक का नाम, पता, शहर यह **entity customer** को परिभाषित करती है। यह संभव है कि ऐसे दो ग्राहक हो, जिसके नाम, पता एवं शहर मे समानता हो। अतः दुविधा को कम करने के लिए प्रत्येक ग्राहक को **customer_id** एट्रीब्युट के अंदर एक अलग नंबर दिया जाता है।

Relationship बहुत सी **entities** का समूह है। उदाहरण एक **depositor relationship** मे ग्राहक एंव उसके अकाउण्ट जो उसके पास है सम्मिलित रहते है। सभी समान **entities** के समूह को "**entity set**" कहते है। उसी समान **relationship** के समूह को "**relationship set**" कहा जाता है। इसे समझने के लिए हम एक उदाहरण लेते है, एक बैंक का ग्राहक एक **entity** है, लेकिन ग्राहक के भी अलग अलग प्रकार हो सकते है, जैसे कुछ ग्राहक पुंजी जमा करते है, कुछ ग्राहक बैंक से कर्ज लेते है। यदि हम उन सभी ग्राहक जो बैंक से कर्ज ले रहे है, उन्हे एक साथ रखते है, तब उसे मे एक **entity set** कहते है। इन मे प्रत्येक ग्राहक का अलग अलग कर्ज की रकम, अवधी आदि हो सकती है, तथा वह सभी एक बैंक से जुडी है। यह एक समान प्रकार के कर्ज मे ब्याज के दर मे बदलाव होते है, तब उन सभी ग्राहकों के ब्याज के दर मे एक साथ बदलाव होता है,

इसे हम relationship set कह सकते है। यह relationship set बहुत सरल भी हो सकता है, तथा अधिक जटिल भी हो सकता है।

E-R DIGRAM

डाटाबेस की संपूर्ण तार्किक संरचना को ग्राफ के रूप मे दिखाने के लिए E-R diagram का प्रयोग करते है। जो कि निम्न घटको से मिलकर बना होता है।

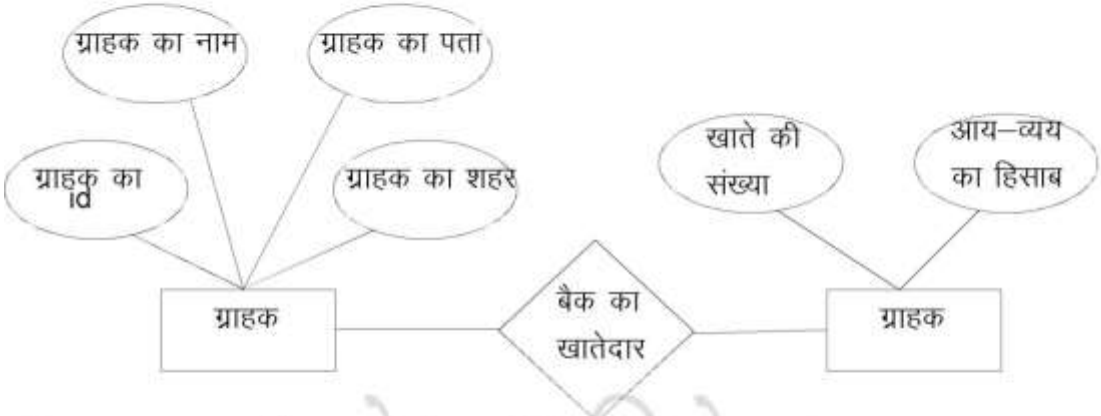
Component	Use	आकृति
Rectangle आयत	यह entity के समूह को दिखाता है।	
Ellipses अण्डाकार आकृति	यह attribute को दिखाता है।	
diamond समचर्तुभुज	यह entity के समूह मे संबंध को दिखाता है।	
line रेखा	यह attribute को entity, से जोडती है एवं entity sets को संबंधो से जोडती है।	

प्रत्येक घटक मे entity एवं उनके संबंधो का संक्षिप्त विवरण होता है जिसको दिखाया जाता है।

ER Diagram का उदाहरण

इसे समझने के लिए हम बैक के डाटाबेस प्रणाली को देखते है। जिसमे ग्राहक के खाते की जानकारी होती है।

उपरोक्त विवरण को नीचे दिखाया गया है। E-R diagram मे दो entity के समूह है। एक ग्राहक एवं दूसरा खाते जिसमे attribute भी है। attribute को ऊपर schema मे दिखाया गया है। रेखाचित्र मे ग्राहक एवं खाते के सम्बन्ध को depositor (खातेदार) द्वारा



दिखाया गया है। इसके अतिरिक्त Entity एंव relationship को डालने के लिए E-R model मे कुछ नियम होते है। जिसे डाटाबेस के द्वारा निश्चित किया जाता है। entity एवं रिलेशन के अतिरिक्त ER Model मे निश्चित घटक होते है, जो डाटा बेस के डाटा से संतुष्ट होते है। एक महत्वपूर्ण घटक Mapping cardinalities है, यह महत्वपूर्ण परीस्थिति है। जो उन entity को दर्शाता है, जिसमे दूसरी entity एक relationship द्वारा आकार जुड सकती है। उदा. प्रत्येक खाते के लिए केवल एक ग्राहक ही रहता है।

E-R model data base की संरचना करने मे बहुत अधिक प्रयोग मे आता है।

2) Relational model :- relational model डाटा एंव उनके बीच के सम्बन्धो को दिखाने के लिए table का प्रयोग करता है। प्रत्येक टेबल मे बहुत से कॉलम होते है। एवं प्रत्येक कॉलम का एक अलग

नाम होता है। नीचे दिया गया टेबल एक relational data database को दिखाता है।, इसमे तीन टेबल है। पहली टेबल मे बैंक के ग्राहको कि सपूंर्ण जानकारी है। दूसरी टेबल खाते को दिखाती है, एवं तीसरी टेबल मे कौन सा ग्राहक किस खाते को रखता है, यह दिखाया गया है।

टेबल **1** ग्राहक डाटा का टेबल

custome r_id	Customer _name	Customer_s treet	Customer _city
1	नेहा	ए. पी. रोड	औरंगाबाद
2	रानी	बर्डी	जालना
3	हर्शल	राम नगर	नागपुर
4	सीता	जवाहर नगर	मुबंइ
5	गीता	नेहरू रोड	दिल्ली
6	राजु	गांधी नगर	रायपुर
7	मोहन	जय नगर	भोपाल

टेबल **2** खाता डाटा बेस का टेबल

account number	*balance*
A-101	*500*
A-215	*700*
A-102	*400*
A-305	*350*
A-201	*900*
A-217	*750*
A-222	*700*

टेबल **3** जमाकर्ता के डाटा बेस कि टेबल

customer_id	*account_no*
1	*A-101*
2	*A-215*
3	*A-102*
4	*A-305*
5	*A-201*
6	*A-217*
7	*A-222*

पहली टेबल जो ग्राहक टेबल है, यह दिखाती है कि प्रत्येक ग्राहक को उनके customer_id द्वारा पहचाना जाता है। जैसे customer_id-4 का नाम सीता है, एवं वह जवाहर रोड मुंबइ मे रहती है।

दूसरी टेबल, एकांउट यह टेबल दिखती है कि एकांउट नंबर *A-101* मे 500 रूपये जमा है। एवं *A-201* मे 700 रूपये जमा है।

तीसरी टेबल मे कौन सा खाता नंबर किस ग्राहक का है, यह दिखाया गया है। उदाहरण A-101 यह अकाउण्ट नंबर ऐसे ग्राहक को दिखाता है, जिसका customer _id यह 1 है एवं उसका नाम नेहा है।

Relational model यह रिकार्ड पर आधारित मॉडल का एक उदाहरण है। प्रत्येक टेबल एक निश्चित प्रकार के रिकार्ड को रखता है। प्रत्येक रिकार्ड का प्रकार एक निश्चित संख्या मे फील्ड या एट्रिब्यूट को परीभाषित करता है। टेबल मे उपस्थित कॉलम रिकार्ड के एट्रिब्युट से संबंधित है।

रिलेशनल मॉडल सबसे अधिक प्रयोग मे आनेवाला डाटा मॉडल है। एवं वर्तमान मे डाटा बेस प्रणाली मे सबसे अधिक रिलेशनल मॉडल प्रयोग मे आते है।

Design Issues

entity set एवं relationship set के विचार निश्चित नही है। एवं कई तरीके से entity set एवं entities set के बीच के सम्बन्धो को परीभाषित करना संभव है। यहा कुछ मूल डिजाइन करने के लिए दी हुई है। यह कुछ मूलभूत डिजाइन के विचार ER डाटा बेस के रूपरेखा को डिजाइन करने के लिए दिए गए है।

A) एट्रीब्युट के प्रति entity set का प्रयोग :– Entity set employees जिसमे कि एट्रीब्युट कर्मचारी के नाम, फोन नंबर के बारे मे विचार करते है। यह आसानी से तर्क सम्मत है की टेलीफोन एक एन्टिटी है जिसके साथ एट्रीब्युट टेलीफोन नंबर एवं पता है। यदि हम इस नजरिये को लेते है तो कर्मचारी एन्टीटी सेट को पुनः इस प्रकार परीभाषित करना चाहिए

1) कर्मचारी entity set के साथ एट्रीब्युट कर्मचारी का नाम है।

2) टेलीफोन entity set के एट्रीब्युट टेलीफोन नंबर एवं पता है।

3) relationship set emp_telephone जो कर्मचारी एवं टेलीफोन के बीच साझेदारी को बताते है।

कर्मचारी की इन दो परीभाषाओ मे मुख्य अंतर क्या हो सकता है। टेलीफोन को एक एट्रीब्युट टेलीफोन नंबर की तरह व्यवहार करने से यह संकेत देता है, की प्रत्येक कर्मचारी का एक सुनिश्चित अलग टेलीफोन नंबर होता है। टेलीफोन को एक entity जैसे व्यवहार मे लाने से यह स्वीकृति देता है, कि एक कर्मचारी बहुत से टेलीफोन

नंबर को रख सकता है। यद्यपि हम इसके अलावा टेलीफोन नंबर को बहु मानों पर आधारित एट्रीब्युट जैसे भी परीभाषित कर सकते है।

टेलीफोन नंबर को एक entity के रूप मे व्यवहार मे लाने पर किसी स्थिति का एक अच्छा model बनता है, जहां कोइ भी अतिरिक्त सूचनाए जो टेलीफोन से संबंधित हो जैसे इसकी स्थिति या उसका प्रकार या टेलीफोन को कितने लोग साझा कर रहे है। इसको प्राप्त कर सकते है। यह इसका मुख्य अंतर है।

टेलीफोन को एट्रीब्युट के रूप मे व्यव्हार मे लाने कि अपेक्षा एक entity के रूप मे इसका व्यव्हार ज्यादा सामान्य है। एवं यह उपयुक्त भी है, जब सामान्य नियम लागू हो।

B) Relationship sets के प्रति entity set का प्रयोग :- यह बात कभी भी स्पष्ट नही हुई है, कि एक आब्जेक्ट को entity set द्वारा सही तरीके से दिखा सकते है, या एक relationship set द्वारा सही तरीके से दिखा सकते है। हम मानकर चलते है, कि बैंक लोन को एक entity कि तरह तैयार किया गया है। इसके अलावा एक विकल्प यह भी है। हम लोन को entity model कि तरह नही बल्कि ग्राहक एंव शाखाओ के मध्य relationship के द्वारा प्रतिरूप बनाये जिसमे लोन नंबर और अमांउण्ट एक वर्णित एट्रीब्युट कि तरह सम्मिलित है। प्रत्येक लोन को ग्राहक एव शाखाओ के बीच के relationship द्वारा प्रस्तुत किया जाता है। यदि प्रत्येक लोन को केवल एक ग्राहक द्वारा रखा गया है, एव जो केवल एक शाखा से जुडा हुआ है, तो हम उस डिजाइन से जहां एक लोन को relationship द्वारा दर्शाया गया है, इससे संतुष्ट हो जाते है। यद्यपि इस डिजाइन से हम उस स्थिति को जहा बहुत से ग्राहक एक हि लोन का

मिलाकर लेते है, उसे स्पष्ट नही कर सकते। इस स्थिति को संभालने के लिए हमे प्रत्येक धारको जिनने मिलकर एक लोन लिया है के relationship को अलग–अलग परिभाषित करना पडेगा। तब हमे प्रत्येक रिलेशनशिप मे विवरणात्मक एट्रीब्युट लोन नंबर और अमांउण्ट के लिए मानों को दोहराना पडेगा। इस प्रकार के relationship मे वर्णित एट्रीब्युट लोन नंबर एवं अमांउण्ट के लिए समान value होना चाहिये।

C) Binary versus non-binary relationship sets : - डाटाबेस मे हम अनेक बार बायनरी relationship का प्रयोग करते है। कुछ relationship जो non binary है उन्हे binary relationship द्वारा अच्छे से प्रदर्शित किया जा सकता है। एक ternary relationship अभिभावक के बारे मे सोचते है। जो एक बच्चे का उसके माता या पिता से संबंध बताते है। इस प्रकार के relationship को दो बायनरी relationship माता एवं पिता के द्वारा भी दिखा सकते है। जिसमे बच्चे का उसके माता एवं पिता से अलग–अलग संबंधो को दिखाया जा सकता है। दो relationship माता एवं पिता का प्रयोग करने से यदि हमे उसके पिता के सम्बन्ध मे कोई जानकारी नही है तब हम सिर्फ बच्चे की माता के रिकार्ड को ही रख सकते है। तो एक null value कि आवश्यकता ternary relationship अभिभावक मे होती हैं यह हमेशा संभव नही है कि एक non binary relationship set को बहुत से व्यक्त binary relationship set द्वारा हटाया जाये।

Weak entity sets :- यदि एक entity set मे पर्याप्त एट्रीब्युट प्रायमरी key बनाने के लिए उपलब्ध ना हो, तो ऐसी entity set को एक Weak entity sets कहते है।

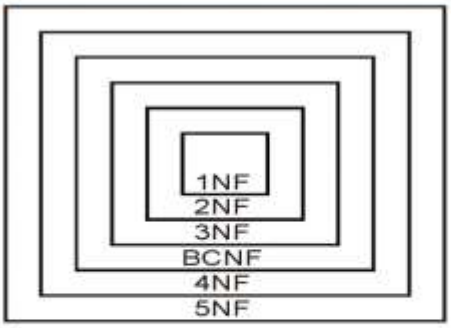
Strong entity sets:- entity set जिसमे एक primary key हो उसे strong entity set कहते है। उदाहरण वेतन को लेते है जिसमे तीन एट्रीब्युट payment-number, payment -date, payment-amount है। payment number जो क्रमवार संख्या मे हो 1 से शुरू होते हो एवं प्रत्येक लोन को अलग वितरित करते है। यद्यपि प्रत्येक payment entity का अलग अंत होता है,अलग–अलग लोन के लिए payment एक ही payment नंबर का साझा करती है। अतः इस entity set के पास कोई primary key नही होती है। यह एक weak entity set है। weak entity set को अर्थपूर्ण बनाने के लिए इसे अन्य entity set के साथ जोडना आवश्यक होता है। अन्य entity set को identifying or owner entity set कहते है।

Normalization

इसका मुख्य आधारए अतिरिक्त डाटा को कम करना है। अर्थात सूचनाओं को केवल एक बार संग्रहित किया जाये। बार–बार संग्रहित डाटा, संग्रहण के लिए अधिक जगह लेती है, एवं डाटा संग्रहण का कुल आकार बढा देती है ।

नॉरमेलाइजेशन के मुख्य बिंदु निम्न लिखित है।

1. अलग–अलग रो मे डाटा के मानों की अनावश्यक रूप से पुनरावर्ती नही होना चाहीए
2. रो के प्रत्येक एट्रीब्युट के लिए एक मान आवश्यक है।



3. एक रो से डाटा के हटाने या डालने से अन्य Row पर प्रभाव नही पडना चाहिए।

नॉरमेलाइजेशन की क्रिया मे विभिन्न फॉर्म सम्मिलित रहते है।

1. First normal form :- इस को 1NF से दर्शाते है। किसी दी गई रिलेशन को 1NF मे परीवर्तित करना यह नॉरमेलाइजेशन का बहुत आवश्यक पद है। प्रत्येक फॉर्म अपने पहले के फॉर्म से अधिक बेहतर रहता। 2NF यह 1NF से तथा 3NF यह 2NF से अधिक बेहतर होता है।

1NF मे प्राप्त डाटा को बिना परिवर्तन किए टेबल मे रख देते है।

order no.	order date	Item Code	Quantity	price
1234	12–07–2003	5678	52	10.50
1234	12–07–2003	5964	25	58.60
1234	12–07–2003	5682	100	10.20
1444	15–10–2003	6596	25	35.20
1444	15–10–2003	5964	65	58.60
1445	15–11–2003	5964	40	58.60

टेबल 1 पहला नार्मल फॉर्म

दूसरा नार्मल फॉर्म :– किसी रिलेशन को हम 2NF मे तब कह सकते है। जब यह 1NF मे हो एवं non-key एट्रीब्युट, key एट्रीब्युट पर फक्शनली आधारित हो। उदाहरण टेबल 1 को देखें यह रिलेशन

1NF मे आर्डर नंबर, आयटम कोड, एक key है। 2NF रिलेशन को टेबल 2 मे दिखाया गया है।

order no.	order date
1234	12–07–2003
1444	15–10–2003
1445	15–11–2003

order no.	item code	qty
1234	5678	52
1234	5664	25
1234	5682	100
1444	6596	25
1444	5964	65
1445	5964	40

item code	price unit
5678	10.50
5964	58.60
5682	10.20
6596	35.20

टेबल 2

तीसरा नार्मल फॉर्म :– यह उस जगह आवश्यक है जहाँ किसी रिलेशन के सभी एट्रीब्युट यह key एट्रीब्युट पर functionally निर्भर नही है। यदि दो non-key एट्रीब्युट functionally एक दूसरे पर आधारित है, तब यहां अनावश्यक रूप से डाटा पुनरावर्ती होती है। टेबल 3 पर विचार करें। यहॉ रोल नंबर keyह तथा अन्य एट्रीब्युट इस पर functionally निर्भर है।

रोल नंबर	नाम	विभाग	साल	छात्रावास का नाम
1745	नरेन्द्र	रसायन	1	निर्मल
4578	अजय	गणीत	1	निर्मल
4699	अभय	भौतिक	2	गोदावरी

4478	विजय	रसायन	2	गोदावरी
4523	कमल	गणीत	3	गंगा

टेबल नंबर 3

अतः यह टेबल 2NF मे है। टेबल मे यह दिखाई देता है कि कॉलेज के प्रथम वर्ष के छात्र "निर्मल" छात्रवास तथा द्वितीय वर्ष के छात्र "गोदावरी" छात्रवास मे, तृतीय वर्ष के छात्र "गंगा" छात्रवास मे रहते है। अतः यह कह सकते है कि छात्रवास का नाम दूसरे non key एट्रीब्युट साल पर निर्भर है। अतः 3NF मे यह टेबल निम्न रूप मे दिखाई देगें

रोल नंबर	नाम	विभाग	साल
1745	नरेन्द्र	रसायन	1
4578	अजय	गणीत	1
4699	अभय	भौतिक	2
4478	विजय	रसायन	2
4523	कमल	गणीत	3

टेबल नंबर 3

साल	छात्रावास का नाम
1	निर्मल
2	गोदावरी
3	गंगा

टेबल नंबर4

Boyce-code normal form (BCNF)

यदि एक मिश्रित key के एट्रीब्युट दूसरे मिश्रित key के एट्रीब्युट पर निर्भर होते है, तो इस प्रकार के नॉरमेलाइजेशन को BCNF कहा

जाता है। इसका उदाहरण रिलेशनशिप प्रध्यापक के द्वारा समझा जा सकता है।

1 प्रध्यापक एक से अधिक विभाग मे कार्य कर सकता है।

2 प्रत्येक विभाग कि समय सारणी निश्चित होती है।

3 प्रत्येक विभाग का एक विभागाध्यक्ष होता है।

इस रिलेशन को टेबल नबर 4 मे दिखाया गया है। इसमे दो संभव मिश्रित key Professor_code एवं Department या Professor_code एवं Head of Department है।

department एवं Head of Department मिश्रित key के भाग है।

प्रोफेसर कोड	विभाग	विभागाध्यक्ष	समय
1	रसायन	गुप्ता	50
1	भौतिक	वर्मा	50
2	गणीत	शर्मा	25
2	रसायन	गुप्ता	75
3	भौतिक	वर्मा	40

टेबल नंबर 5

टेबल नंबर 4 3NF मे है यद्यपी इस टेबल मे विभाग एवं विभागाध्यक्ष कि पुनरावर्ती हुई है। विभाग कि पुनरावर्ती को कम नही किया जा सकता क्योकि इससे सूचना कि हानि होगी। इसलिए हम इस टेबल को BCNF मे डाल कर विभागाध्यक्ष कि पुनरावर्ती को हटा सकते है।

प्रोफेसर कोड	विभाग	समय
1	रसायन	50
1	भौतिक	50

विभाग	विभागाध्यक्ष
रसायन	गुप्ता
भौतिक	वर्मा

2	गणीत	25
2	रसायन	75
3	भौतिक	40

टेबल नबर 6

गणीत	शर्मा

टेबल नंबर 5

चौथा और पाँचवा नार्मल फॉर्म :– जब कोइ रिलेशन के एट्रीब्युट बहु मानों पर आधारित हो तब **4NF** एवं **5NF** जरूरी होते है। एक ऐस2 उदाहरण लेते जिसमे एक दुकानदार बहुतसे कंपनीयों को बहुत सी प्रकार कि वस्तु बेचता है। तब इस कार्य मे निम्नलिखित पद आते है।

1)एक दुकानदार बहुते सी वस्तुओं को बेचने कि क्षमता रखता है।

2)एक कंपनी के बहुत सी वस्तुओं की आवश्यकता पडतीं है।

3)एक दुकानदार बहुत सी कंपनीयों को वस्तु बेच सकता है।

4)एक वस्तु को विभिन्न दुकानदार बेच सकते है।

इस उदाहरण को टेबल नंबर 5 मे इस रिलेशन व्दारा हल किया गया है।

दुकानदार का कोड	वस्तु का कोड	कंपनी का नाम
ABC1	बेअरींग001	इंडोरामा
ABC1	बेअरींग002	इंडोरामा
ABC1	बेंअरींग001	निको स्टिल
ABC1	बेअरींग002	निको स्टिल
PQR1	बेअरींग002	इंडोरामा
PQR1	मोटर001	इंडोरामा

टेबल नंबर 7

यह 3NF एवं BCNF के रूप मे है। चुकि इसमे पुनरावर्ती की समस्या खत्म नही हुई है। इसलिए इसे 4NF एवं 5NF मे डाला गया है।

दुकानदार का कोड	वस्तु का कोड
ABC1	बेअरींग001
ABC1	बेअरींग002
PQR1	बेअरींग002
PQR1	मोटर001

टेबल नं 8

दुकानदार का कोड	कंपनी का नाम
ABC1	इंडोरामा
ABC1	निको स्टिल
PQR1	इंडोरामा

टेबल नं9

इस रिलेशन मे अभी भी समस्या है दुकानदार abc1 मे सभी वस्तु बेचने कि क्षमता है लेकिन सभी कंपनीयों कि उसकी जरूरत नही है। अतः एक और फॉर्म जो टेबल 7 एवं 8 का योग है, कि आवश्यकता होगी इसको 5NF कहा जाता है।

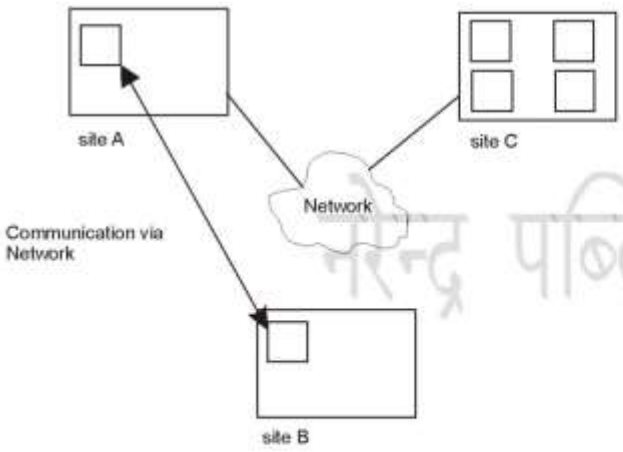
कंपनी का नाम	वस्तु का कोड
इंडोरामा	बेअरींग001
इंडोरामा	बेअरींग002
निको स्टिल	बेअरींग001
निको स्टिल	बेअरींग002

5NF एडीशनल रिलेशन है।

Distributed Database

डिस्ट्रीब्यूटेड डाटा बेस प्रणाली मे, डाटा बेस यह अलग–अलग कम्प्यूटर पर होता है। वह कम्प्यूटर विभिन्न माध्यमों जैसे कि तेज

गति के नेटवर्क , टेलीफोन लाइन आदि से डाटा का अदानप्रदान करते है। यह मुख्य मेमोरी या डिस्क का साझा करते है। यह कम्प्यूटर अलग–अलग आकार के हो सकते है। जो एक छोटे वर्कस्टेशन से लेकर बडे मेनफ्रेम कम्प्यूटर तक हो सकते है। इस प्रणाली मे प्रयोग होने वाले कम्प्यूटर को संख्या पर आधारित अलग–अलग नाम दिये जाते है, जैसे कि **sites** या **node** आदि। जो उसके काम पर आधारित होते है। हम ज्यादातर साइट (Site) इस शब्द का प्रयोग करते है।



डिस्ट्रीब्युटेड डाटा बेस और साझा नही करने वाले डाटा बेस मे मुख्य अंतर भौगोलिक पृथकता का होता है। डिस्ट्रीब्यूटेड डाटा बेस प्रणाली यह अलग–अलग नियंत्रित होते है तथा उनमे डाटा जुडने कि गति कम होती है। इसे मे मुख्य घटक स्थानीय और सार्वभौमिक ट्रान्जक्शन का होता है। स्थानीय ट्रान्जक्शन वह होता है जिसमे डाटा उन्ही साइट से लिया जाता है जहाँ से ट्रान्जक्शन की शुरूवात हुई है। जबकी ग्लोबल ट्रान्जक्शन मे हम विभिन्न साइटो से डाटा प्राप्त कर सकते है, इससे कोई अंतर नही आता की ट्रान्जक्शन कहा से चालू हुआ है। डिस्ट्रीब्यूटेड डाटा बेस प्रणाली के निम्न लाभ है

डाटा का अदान प्रदानः– इस प्रणाली मे एक साइट का प्रयोगकार्ता किसी दूसरे जगह के साइट के डाटा को देख या प्राप्त कर सकता

है। उदाहरण के लिए डिस्ट्रीब्यूटेड बैक प्रणाली मे हर शाखा मे डाटा संग्रहित होता है तथा वह दूसरे शाखा के साइट से प्राप्त किया जा सकता है। इससे किसी एक शाखा का ग्राहक दूसरे शाखा मे लेन देन कर सकता है।

स्वतंत्रताः– जो डाटा स्थानीय तौर पर संग्रहित किया गया है उस पर प्रत्येक साइट का नियंत्रण होता है। केंद्रियकृत प्रणाली मे डाटा बेस प्रबंधक सभी साइटों पर केंद्रिय नियत्रंण रख सकता है। डिस्ट्रीब्युटेड प्रणाली मे ग्लोबल डाटा बेस व्यवस्थापक संपूर्ण प्रणाली का जिम्मेदार होता है। इस जिम्मेदारी को स्थानिय डाटा बेस प्रंबधक मे बांटा जा सकता है। यह डिस्ट्रीब्यूटेड डाटा बेस प्रणाली के आकार पर निर्भर होती है। हर डाटा बेस प्रबंधक को कुछ सीमा तक स्वंतत्रता होती है। इस स्थानिय स्वतंत्रता की संभावना इस प्रणाली का लाभ है।

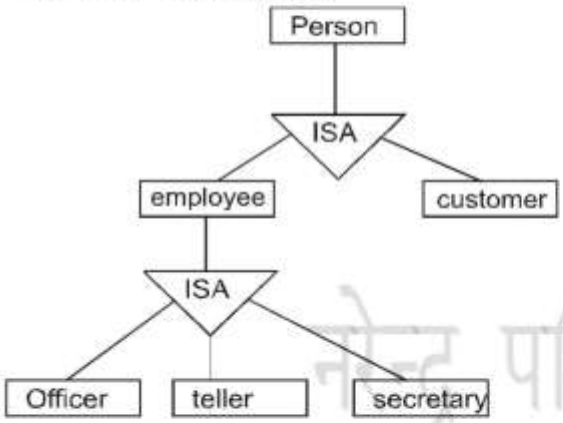
उपलब्धता :– यदि इस प्रणाली की कोई एक साइट मे समस्या आती है, तब भी उससे जुडी हुई बाकी साइटों पर क्रियाये चलती रहती है। विशेषतः जब डाटा घटक विविध साइटो पर दोहरायें जाते है, तब वांछित डाटा घटक दूसरे साइटों से प्राप्त किये जा सकते है। इसमे किसी एक साइट मे समस्या आने पर संपूर्ण प्रणाली ठप्प नही होती है।

किसी साइट कि समस्या को प्रणाली द्वारा ढूँढा जा सकता है। तथा उसके सुधार के लिए जरूरी कदम लिए जा सकते है। जैसे ही कोई साइट खराब होती है वैसे ही वह संपूर्ण प्रणाली की सेवा से हट जाती है। जब उस साइट को सुधारा जाता है, तब उसमे ऐसे कार्य रचना होनी चाहिये जो उस साइट का प्रणाली मे निर्विघ्न रूप से वापस ला सकें। यह उपयोग रीयल टाइम एप्लिकेशन कामो मे बहुत महत्वपूर्ण होता है।

Inheritance

एक आब्जेक्ट ओरीयेंटेड डाटा बेस कि रूपरेखा मे बहुत से क्लास कि आवश्यकता होती है। कई बार बहुत सी क्लास एक समान होती है। उदाहरण के लिए हमारे पास एक आब्जेक्ट ओरीयेंटेड डाटा बेस बैंक के अनुप्रयोग के लिए है। हम यह मानते है कि इस डाटा बेस कि दो क्लास bank_emp एवं bank_cust दोंनों समानता रखती है जैसे यह दोंनो क्लास एक समान वेरीएबल जैसे name, address आदि को रखती है। यद्यपी emp के लिए एक विशेष वेरीएबल (salary) एवं customer क्लास के लिए एक विशेष व्हेरिएबल (credit rating) होते है। यह वेरीएबल केवल इन्ही क्लास मे परीभाषित होते है। हम एक समान वेरीएबल को एक ही जगह पर परीभाषित कर सकते है, यह तभी कर सकते जब हम कर्मचारी या ग्राहक को एक क्लास मे साथ मे रखते है। हम यह कह सकते है कि कर्मचारी एक व्यक्ति की विशेषता बताता है। क्योकि व्यक्ति के समूह का कर्मचारी एक उपसमूह है। अतः प्रत्येक ग्राहक भी व्यक्ति को विशेषीकृत करता है। किसी E-R Model मे विशेषीकृत hierarchy का तथ्य क्लास hierarchy से समानता रखता है। कर्मचारी एवं ग्राहक को उस क्लास मे रख सकते है,जो की एक व्यक्ति के क्लास की विशेषता बताती है। वेरीएबल एवं पद्धती जो

चित्र a बैंक के hierarchy



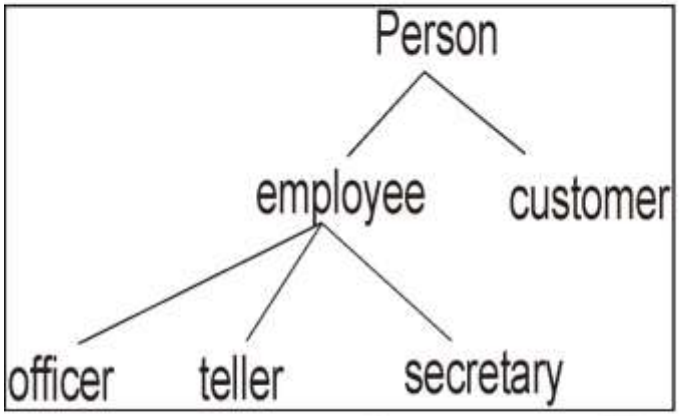
ग्राहक की विशेषता बताती है, उसे ग्राहक कि क्लास मे रखा जाता है। वेरीएबल एवं पद्धती जो कर्मचारी की विशेषता बताती है, उसे कर्मचारी कि क्लास मे रखा जाता है। वेरीएबल एवं पद्धती जो ग्राहक एवं कर्मचारी दोनो की विशेषता बताती है, उसे व्यक्ति (person)की क्लास मे रखा जाता है।

चित्र a एक स्पेसलायजेशन hierarchy को दिखाता है जिसमे हमारे बैक के उदाहरण मे कितने लोग कार्य कर रह यह प्रर्दशित हेाता है। चित्र b मे क्लास hierarchy को दिखाया गया है। हम कि वर्ड ISA का प्रयोग यह दिखाने मे करते है कि यह क्लास किसी अन्य क्लास कि विशेषता बताती है। क्लास का इस प्रकार का वर्गीकरण को सब क्लास कहते है । उदाहरण के लिए कर्मचारी, व्यक्ति की सब क्लास है। एवम teller कर्मचारी की सब क्लास है। इसके विपरीत हम यह कह सकते है कि कर्मचारी सुपर क्लास है teller की एवम व्यक्ति सुपर क्लास है कर्मचारी की। आब्जेक्ट जो एक ऑफिसर को दिखाते है वे ऑफिसर क्लास के सभी वेरीएबल को रखते है एवम इसके साथ साथ यह कर्मचारी एवम व्यक्ति के क्लास के सभी वेरीएबल को रखते है। इसका कारण यह है की प्रत्येक ऑफिसर एक कर्मचारी होता है एवम प्रत्येक कर्मचारी एक व्यक्ति होता है । अतः हम कह सकते है की प्रत्येक ऑफिसर एक व्यक्ति होता है।

वह गुण जिसमे किसी क्लास के आब्जेक्ट वेरीएबल को रखते है एवम यह वेरीएबल इसकी सुपर क्लास मे परीभाषित होते है इसे वेरीएबल का inheritance कहते है ।

सूचनाएं एवम पद्धती भी वेरीएबल के समान ही inherited होती है। inheritance का सबसे मुख्य लाभ oops मे यह है की इसमे प्रतिस्थापन संभव है ।

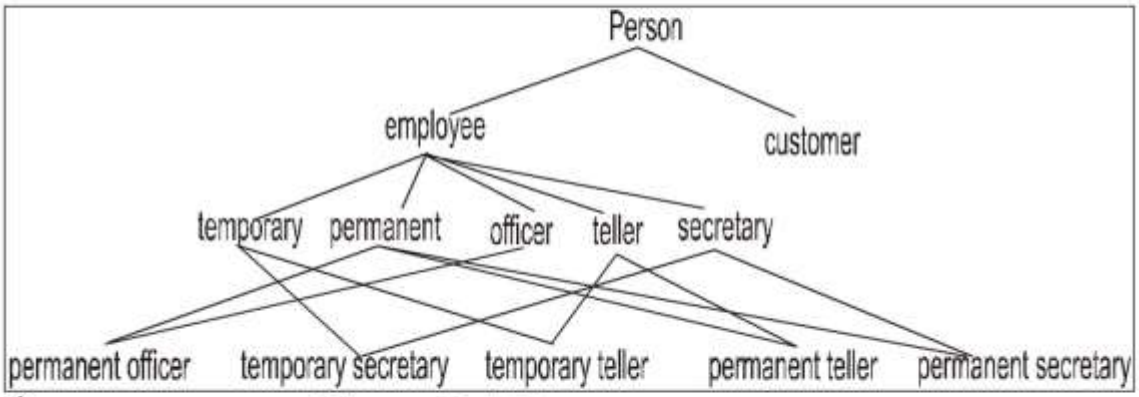
चित्र b जो चित्र a के अनुरूप class hierarchy है



क्लास A की कोई भी पद्धती सब क्लास B के कोई आब्जेक्ट से समान तरीके से जुडी हुई होती है। यह गुणधर्म code reuse को देता है क्योंकि सूचनाएं, पद्धती एवम फंक्शन यह क्लास B के आब्जेक्ट के लिए दुबारा नही लिखते ।

अधिकतर स्थिति मे क्लास का tree structure organization संतोष जनक होता है यह बहूत से अनुप्रयोग जो इसमे शामिल है उन्हे वर्णित करता है यद्यपि कुछ स्थिति है जिन्हे tree संरचना व्दारा अच्छे से प्रदर्शित नही किया जा सकता । किसी भी क्लास का कम से कम एक सुपर क्लास अवश्य होता है multiple inheritance यह क्लास को वेरीएबल एवम पद्धती को दूसरे और सुपर क्लास से inherit करने की अनुमती देते है क्लास एवम सब क्लास की रिलेशनशिप यह directed a cyclic graph (DAG) से दिखाते है और इसमे एक क्लास के एक से ज्यादा सुपर क्लास रहते है उदाहरण – मान लिजिए कोई कर्मचारी या तो स्थाई है या तो अस्थाई है इसके व्दारा हम कर्मचारी क्लास मे दो सब क्लास temporary(अस्थाइ) एवम permanent(स्थाई) बना सकते है । सब क्लास temporary मे termination date नाम का एक एट्रीब्युट रहता है जो यह दर्शता है की कर्मचारी की कार्य सीमा समाप्त हो चुकी है सब क्लास permanent मे एक पद्धती है जो

कंपनी पेन्शन प्लान के योगदान की गणना करती है एवम यह पद्धती temporary कर्मचारी पर लागू नही कर सकते क्योकि उन्हे पेन्शन नही दी जाती । ऊपर दिया गया वर्गीकरण कर्मचारी के job से स्वतंत्र है । क्योकि job कोई भी हो सकता है जैसे ऑफिसर, टेलर, सेक्टरी । ऐसे भी ऑफिसर हो सकते है जो permanent है एवम कुछ temporary है टेलर भी temporary या permanent हो सकते है। multiple inheritance का प्रयोग करके हम सिर्फ नये क्लास को तैयार कर सकते है जैसे की temporary teller जो की temporary एवम टेलर का सब क्लास है । और permanent टेलर जो की permanent एवम टेलर का सब क्लास है । ऐसे combination जिनकी वास्तविकता में आवश्यकता नही होती उनको तैयार नही करते है। उदाहरण – अगर सभी ऑफिसर permanent है तो temporary ऑफिसर क्लास तैयार करने की आवश्यकता नही है। नीचे दिये गये चित्र मे क्लास hierarchy के परिणाम को दिखाया गया है



बैक उदाहरण का Class DAG

MS Access

Unit 2

UNIT–II

Create a Table in MS Access -Data Types, Field Properties, Fields: names, types, properties--default values, format, caption, validation rules Data Entry, Add record, delete record and edit text, Sort, find/replace, filter/ select, rearrange columns, freeze columns. Edit a Tables- copy, delete, import, modify table structure, find, replace.

Unit-2

MS-Access

जानकरियों को व्यवस्थित रखने के लिए डाटाबेस प्रप्रणाली का उपयोग किया जाता हैं। सही तरीके से डाटाबेस डिजाइन करने से इच्छित जानकारी, सटिक तरीके से एवं तेजी से प्राप्त कि जा सकती है, उसके विपरीत यदि डाटाबेस का व्यवस्थापन सही तरीके से नही किया है, तब वांछित जानकारी प्राप्त करने मे कठनाईया आ सकती है। एमएस एक्सेस यह डाटाबेस को Row और Column के रूप मे संग्रहित करता है। एमएस एक्सेस मे विविध प्रकारों के डाटा को व्यवस्थित तरीके से संग्रहित किया जा सकता है, जैसे टेक्स्ट, अंक, पिक्चर, इमेज, ध्वनी आदि। प्रत्येक प्रकार के डाटा के अलग फील्ड प्रकार निश्चित कि जाती है। एक्सेस मे सभी प्रकार के डाटा पर कार्य करने के लिए उसे डाटाबेस फाइल मे संग्रहित किया जाता है, एक डाटाबेस फाइल मे टेबल, रिपोर्ट, फॉर्म आदि संग्रहित किये जाते है। सभी प्रकार के रिपोर्ट, क्यूरी, फॉर्म आदि टेबल के आधार पर बनाये जाते है। एक्सेस मे RDBMS(relational database management system) की सुविधा है, अर्थात आप एक डाटाबेस का दूसरे डाटाबेस के साथ साझा कर सकते है। एक डाटाबेस मे दूसरे डाटाबेस का डाटा का प्रयोग

कर सकते है। लेकिन डाटा साझा करने के लिए कुछ नियम है।

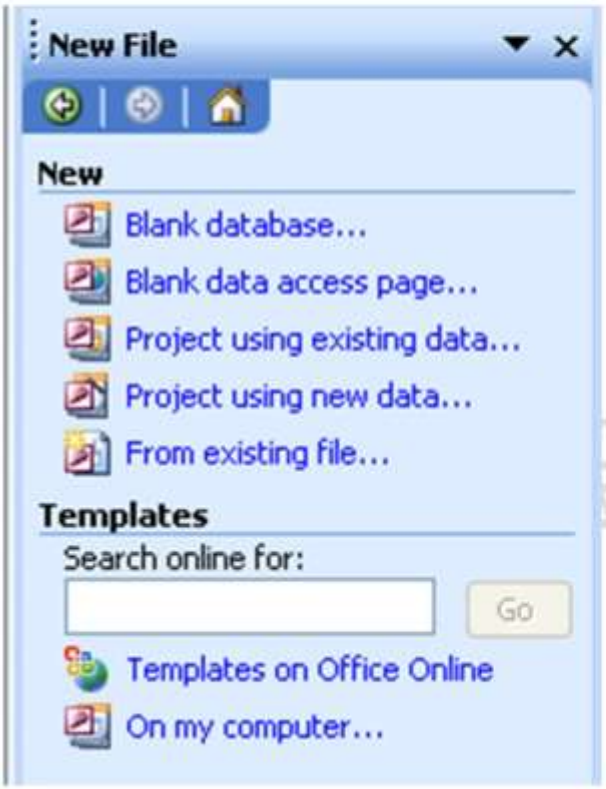
एम एस एक्सेस यह एम एस ऑफिस पैकेज का एक अनुप्रयोग (application) है। इस अनुप्रयोग मे डाटा बेस बनाना, उसका प्रबंधन करना, डाटाबेस मे संग्रहित इच्छित जानकारी प्राप्त करना आदि कार्य के लिए विभिन्न टूल दिये है। यह एक RDBMS प्रणाली है, जिसमे डाटाबेस बना कर दूसरे अप्लिकेशन से जोड सकते है। इसमे डाटाबेस को एक से अधिक टेबल से प्राप्त कर इच्छित जानकारी दर्शाने के सुविधा है। इसमे डाटाबेस फाइल बनाने के साथ ही Form, Report, Quarries, आदि बनाने की सुविधा है। इसमे बहुतसे मुख्य कार्य के लिए Wizard दिये है, जिससे इच्छित काम तेजी से एवं सरलता से हो सकता है। इसमे बनाये गये डाटा को विभिन्न प्रोग्रामिंग भाषाओं मे प्रयोग किया जा सकता है। Visual Basic से एमएस एक्सेस के टेबल को जोड कर उच्चस्तरीय सॉफ्टवेयर का निर्माण किया जा सकता है। वर्तमान मे बहुतसे सॉफ्टवेयर मे एमएस एक्सेस के डाटाबेस फाइल का उपयोग हो रहा है। वर्तमान मे डाटाबेस प्रबंधन के लिए इस अनुप्रयोग का अधिक प्रयोग हो रहा है।

एम एस एक्सेस यह एक RDBMS (Relation database management system) है, यद्यपी इसमे DBMS के गुण भी है। इसके RDBMS होने के निम्न कारण है

1. एम एक्सेस मे आप एक फाइल के अंदर विभिन्न टेबल का निमार्ण कर सकते है, तथा उनके बीच मे संबंध स्थापित कर पाते है।
2. एक फाइल के अंदर ना सिर्फ टेबल का निमार्ण किया जा सकता है, अपितु अन्य ऑब्जेक्ट जैसे Form, report, query का भी प्रयोग किया जा सकता है।
3. जैसे ही किसी टेबल मे डाटा मे बदलाव होता है, वैसे ही उससे जुडे रिपोर्ट, क्युरी आदि मे भी बदलाव हो जाता है।
4. एमएस एक्सेस मे डाटाबेस का प्रयोग अन्य अप्लिकेशन जैसे Visual Basic आदि मे किया जा सकता है।
5. अन्य प्रकार के डाटा को एमएस एक्सेस के डाटाबेस से जोडा जा सकता है।
6. एमएस एक्सेस मे टेबल के बीच विभिन्न प्रकार के रिलेशनशिप का विकास किया जा सकता है।
7. एमएस एक्सेस के विभिन्न टूल का प्रयोग कर एक संपूर्ण अप्लिकेशन का निमार्ण किया जा सकता है।
8. एम एस एक्सेस मे अन्य फारमेट के डाटाबेस मे import या export किया जा सकता है। इसमे आप dbase, foxpro SQL sever, oracle आदि फारमेट के डाटा को प्राप्त कर सकते है।

9. एमएस एक्सेस एक टेबल के बदलाव का असर उससे संबधित टेबल पर पडता है।

एमएस एक्सेस चालू करने के लिए



- Start बटन क्लिक करें
- Program बटन क्लिक करें
- Ms-office विकल्प क्लिक करे।
- Ms-access विकल्प क्लिक करें

एमएस एक्सेस की विंडो खुल जाती है। जिसमे दायें ओर वर्तमान मे बनाये टेबल की सूची दिखाई देती है। तथा नीचे की ओर "Create new file" विकल्प दर्शाया जाता है। इच्छित फाइल खोलने के लिए उस फाइल पर क्लिक करें अथवा नई फाइल बनाने के लिए create New file विकल्प को क्लिक करें।

यदि नई डाटाबेस फाइल बनाने का विकल्प चुना है, तब दायें तरफ के Pane मे अन्य विकल्प दिखाई देते है। इसमे नई

फाइल बनाने के लिए Blank Database का उपयोग किया जाता है।

Blank Database :- यदि इस विकल्प को चुनते है, तब आपको संपूर्ण डाटाबेस डिजाइन करना होगा। आप कार्य के अनुसार फील्ड, उनकी डाटा टाइप आदि सेट कर सकते है। साधारणतः कार्य के अनुसार डाटाबेस डिजाइन करने के लिए इस विकल्प का चुनाव किया जाता है।

इसे क्लिक करने के बाद File new Database का डायलॉग बॉक्स दिखाई देता है। जिसमे उस डाटाबेस फाइल का नाम टाइप करे। एक डाटाबेस फाइल के अंदर बहुतसी प्रकार की फाइल रहती है, जैसे प्रत्येक फॉर्म के लिए अलग फाइल बनती है, प्रत्येक टेबल की अलग फाइल होती है। जो फाइल एक्सेस मे डाटाबेस के रूप मे संग्रहित कि जाती है, उसका extension यह .mdb होता है।

फाइल का नाम टाइप करने के बाद "Create" बटन को क्लिक करे।

अब एक्सेस की मुख्य स्क्रिन खुल जाती है। इस विंडो के निम्न हिस्से होते है।

Title bar

इसमे आपके द्वारा किया जा रहा कार्य प्रदर्शित होता रहता है, आप वर्तमान मे जिस पर कार्य कर रहे है, वह टायटल बार मे

दिखते रहता है। इससे हमें यह पता चलता है कि कौन ऑब्जेक्ट पर कार्य कर रहे है उसका नाम क्या है।

The minimize and maximize button :- यह आपके स्क्रिन पर टायटल बार मे प्रदर्शित होता रहता है। यह अन्य विन्डो प्रोग्रामों की भांति हि कार्य करती है। यह **Application** के मेन्यू बार मे भी प्रदर्शित होता रहता है। जिसकी सहायता से आप विन्डो को बंद कर सकते है, या असक्रिय कर सकते है। **minimize** बटन आपके **application** को **task bar** मे स्टोर करता है। और **maximize** वापस इसे खोल कर आपके सामने स्क्रिन पर ले आता है।

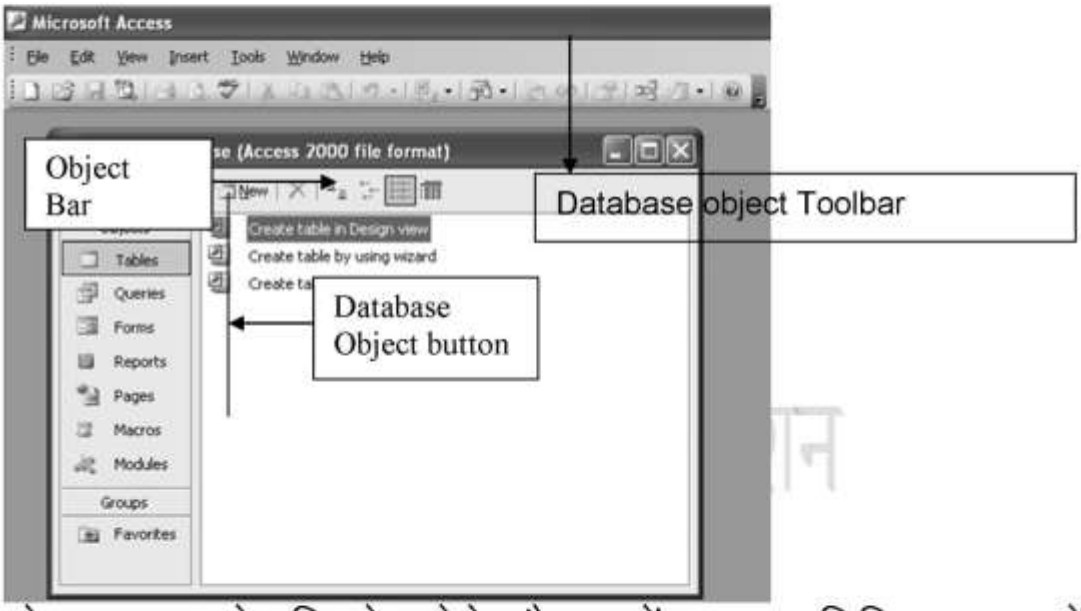
The close button :- यह भी टाइटेल बार मे होता है।इससे आप स्क्रिन को बंद कर सकते है।

Menu bar :- एक्सेस मे भी विंडो के समान पांरपरीक मेन्यू प्रणाली होती है, इसमे **File, Edit, View, Insert, tool, window** एवं **help** मेन्यू है। प्रत्येक मेन्यू मे अलग–अलग कार्य के विभिन्न विकल्प है। यद्यपी एक्सेस मे मेन्यू प्रणाली का उपयोग कम होता है।

Toolbar :- मेन्यू बार के नीचे कुछ छोटे टूल दर्शाये जाते है, उस संपूर्ण बार को टूल बार कहा जाता है। टूल बार मे अलग–अलग कार्य के लिए अलग–अलग आयकॉन दर्शाये जाते है। आयकॉन पर एक क्लिक से उसे संबंधित कार्य चालू हो जाते है।

डाटाबेस विंडो

एक्सेस मे कार्य करने के लिए यह मुख्य विंडो है। उस डाटाबेस से संबंधित सभी घटक जैसे Table, Form, Report, Page आदि इस विंडो मे दर्शाये जाते है। इस विंडो



के मूलतः दो हिस्से होते है, बायें तरफ विभिन्न घटकों के आयकॉन दर्शाये गये है। इच्छित कार्य करने के लिए अपेक्षित आयकॉन का क्लिक किया जाता है, उदाहरण के लिए नया टेबल बनाना है, तब Table आयकॉन को क्लिक किया जाता है। यदि नया फॉर्म बनाना है, या पुराने फॉर्म मे बदलाव करना है, तब Form आयकॉन को क्लिक किया जाता है।

इस विंडो को दायें तरफ चुने हुए घटक की फाइलों की सूची दर्शाई जाती हैं, तथा नया टेबल, फॉर्म रिपोर्ट आदि बनाने के विकल्प दर्शाये जाते है।

एक्सेस मे निम्न मुख्य घटक है।

Tables

टेबल, डाटाबेस फाइल की यह मूल इकाई है। एक डाटाबेस मे डाटा को एक व्यवस्थित रूप मे संग्रहित किया जाता है। एक्सेस मे डाटा को Table मे व्यवस्थित रूप मे संग्रहित किया जाता है। एक डाटाबेस फाइल मे एक से अधिक टेबल हो सकते है। यह टेबल पृथक भी कार्य कर सकते है, तथा एक दूसरे से जुड कर भी कार्य कर सकते है। सभी प्रकार के कार्य जैसे Form, Report आदि सभी, टेबल पर आधारित होते है। डाटाबेस मे सबसे पहले टेबल ही बनाया जाता है। एक डाटाबेस मे बनाये गये प्रत्येक टेबल को अलग–अलग नाम दिया जाता है, तथा सभी टेबल की संरचना अलग–अलग होती है।

Queries

डाटाबेस फाइल मे बनाये गये अलग–अलग टेबल मे से इच्छित रिकार्ड प्राप्त कर, उन्हे कार्य के अनुसार प्रयोग किया जाता है। इच्छित डाटा को प्राप्त करने, उनका विश्लेषण करने या डाटा को बदलने के लिए queries का उपयोग होता है। जहाँ पर डाटा बहुत अधिक मात्रा मे है, वह पर query बहुत महत्वपूर्ण हो जाती है। आप क्यूरी का उपयोग फॉर्म, रिपोर्ट आदि बनाने के लिए कर सकते है। डाटाबेस विन्डो मे बने हुए सभी

queries को दिखाता है। एक डाटाबेस फाइल मे एक से अधिक क्यूरी बनाई जा सकती है। प्रत्येक क्यूरी फाइल अलग–अलग नाम मे संग्रहित की जाती है।

Forms

फॉर्म का मुख्य उपयोग टेबल या क्यूरी मे डाटा एन्ट्री करने के लिए होता है। इसे प्रयोगकर्ता के अनुसार डिजाइन किया जाता है। एक फॉर्म एक या एक से अधिक टेबल से जुडा हो सकता है। जो डाटा फॉर्म के द्वारा डाला जाता है, वह उससे जुडे हुए टेबल मे सही जगह पहुंच जाता है। आप एक फॉर्म के अंदर दूसरा फॉर्म भी बना सकते है। एक्सेस मे एक डाटाबेस फाइल मे बहुत से फॉर्म हो सकते है, प्रत्येक फॉर्म अलग–अलग नाम से संग्रहित किया जाता है। एक्सेस मे फॉर्म डिजाइन के लिए बहुतसे कंन्ट्रोल दिये है, जिनके सहायता से, जटिल एवं बडे फॉर्म भी डिजाइन किये जा सकते है।

Reports

रिपोर्ट का उपयोग विभिन्न टेबल से इच्छित रिकार्ड प्राप्त कर स्क्रिन पर दर्शाने या प्रिन्ट करने के लिए होता है। डाटाबेस अप्लिकेशन का यह एक महत्वपूर्ण हिस्सा है। प्रयोगकर्ता (user) के आवश्यकता के अनुसार टेबल से रिकार्ड को प्राप्त कर, समझने योग्य फारमेट मे दर्शाने का कार्य रिपोर्ट करता है। रिपोर्ट विभिन्न टेबलों से जुडा होता है, तथा जानकारी भी उससे जुडे टेबल से ही प्राप्त करता है। एक डाटाबेस फाइल

मे बहुत से रिपोर्ट हो सकती है। प्रत्येक रिपोर्ट अलग नाम से संग्रहित की जाती है।

Data Access Page

इस घटक का मुख्य प्रयोग बेव पेज या इंटरनेट पर डाटा पेज बनाने के लिए होता है। यह एक प्रकार का टेबल से जुडा वेब पेज है, जो इंटरनेट या इंन्टानेट के माध्यम से देखा जा सकता है। इस पेज मे विभिन्न data source से डाटा भेजा जाता है। इसका प्रयोग इंटरनेट पर एक फॉर्म के तरह भी किया जा सकता है, जिसमे डाटा डालना, देखना एवं उसमे बदलाव करना आदि कार्य किये जा सकते है। इस प्रकार के घटक का उपयोग Ms-Access के बगैर भी कर सकते है। इसे html फारमेट मे सेव किया जाता है।

Macros

मायक्रो का मुख्य कार्य बांर–बार प्रयोग होने वाले कार्य को एक साथ संग्रहित करने एवं उन सभी कार्य को एक क्लिक द्वारा पूर्ण करना है। मायक्रो एक action का समूह है, जो क्रमवार तरीके से कार्य कर एक से अधिक task को पुरा करता है। इसमे संग्रहित किये action अलग–अलग कार्य के लिए हो सकते है, लेकिन सब मिल कर कोई बडा कार्य पूर्ण करते है।

Modules

माडयूल यह किसी कार्य को पूर्ण करने के लिए दिये गये निदेर्शो का समूह है, जो उसमे दि गई कंडिशन के अनुसार

कार्य करते है। इसे Visual Basic भाषा के फारमेट मे बनाया जाता है। जटिल डाटाबेस मैनेंजमेंट प्रणाली मे इस घटक का बहुत महत्व है। इसमे वह कार्य किये जाते है, जो एक्सेस के फॉर्म, रिपोर्ट या मायक्रो से नही किये जा सकते है। माडयूल, एक्सेस से डाटाबेस प्राप्त कर विजुअल बेसीक के प्रोग्रामिंग क्षमताओं का प्रयोग कर इच्छित कार्य पूर्ण करता है।

एमएस एक्सेस मे डाटाबेस बनाना

Table

टेबल मे विभिन्न प्रकार के डाटा को संग्रहित किया जा सकता है। उदाहरण के लिए हम एक फोल्डर मे विभिन्न फाइलों को संग्रहित करते है। साधारणतः एक फोल्डर मे उससे संबंधित फाइलों को संग्रहित करते है, इसी तरह एक टेबल मे उससे संबंधित डाटा रखा जाता है। एक्सेस मे टेबल की डिजाइन यह एक महत्वपूर्ण हिस्सा है। यदि टेबल सही तरीके से डिजाइन किया गया है, तब अन्य कार्य मे दिक्कतें कम होती है। किसी भी टेबल का डिजाइन बनाने से पहले उसका लेआउट कागज पर उतार लें। छोटे डाटाबेस बनाने के लिए बहुत अधिक योजना कि आवश्यक्ता नही होती है, लेकिन बडे एवं जटिल डाटाबेस डिजाइन के लिए विभिन्न तरीको से सोच विचार कर टेबल का अंतिम रूप दिया जाता है। एमएस–एक्सेस टेबल एक डाटाशीट के समान दिखाई देता है, उसमे सबसे पहले या ऊपर के रो मे फील्ड के नाम दर्शाये जाते है, तथा उसके नीचे डाटा दर्शाया जाता है।

नाम	पता	शहर	उम्र	पद	वेतन
अविनाश	राम नगर	नागपुर	34	मैनेजर	20000
अनंत	गोल बजार चौक	अमरावती	20	सुपरवाइजर	8500
संजय	दुर्गा मंदिर के पास	भोपाल	32	एकाउण्टेंट	14000
राजन	विजय नगर बस स्टाप	पुना	23	सुपरवाइजर	8500

रिकार्ड

प्रत्येक कॉलम मे अलग रिकार्ड होते है। प्रत्येक रिकार्ड मे उसके फील्ड के अनुसार डाटा डाला जाता है। एक डाटाबेस फाइल मे निश्चित संख्या के ही फील्ड होते है, लेकिन रिकार्ड की संख्या कार्य के अनुसार कम या अधिक की जा सकती है। डाटाबेस मे रिकार्ड डालने के लिए विभिन्न तरीके है, आप सीधे डाटाबेस फाइल खोल कर उसमे रिकार्ड जोड सकते है, या फॉर्म का प्रयोग कर डाटा जोड सकते है। एक बार डाटा डालने के बाद आप उसे इच्छित तरीके से प्राप्त कर सकते है। वांछित डाटा को स्क्रिन पर या पेपर मे प्रिन्ट प्राप्त किया जा सकता है।

डाटाबेस डिजाइन करना

जब भी कोई डाटाबेस बनाना है, तब सर्वप्रथम उसकी आउटलाइन पेपर पर बना ले। जैसे उसमे कौन से प्रकार का डाटा डालना है, उसके फील्ड को क्या नाम देना है, उनका

प्रकार क्या होना चाहिये आदि। इससे काम व्यवस्थित तथा आसान हो जाता है। यदि सरल प्रोग्राम बना रहे है, तो साधारणतः एक ही डाटाबेस फाइल बना कर काम चल सकता है, लेकिन जटिल एप्लिकेशन के लिए, काम के अनुसार एक से ज्यादा डाटाबेस फाइल बना कर काम कर सकते है। यह काम के अनुसार तय कर सकते है, की एक डाटाबेस मे काम करना है या एक काम के लिए एक से ज्यादा डाटाबेस फाइल बनाना है। जैसे हम Computer Institute का उदाहरण लेते है। इसमे हमें निम्नलिखित डाटा डालना है।

विद्यार्थी का नाम

पता

फोन नंबर

रोल नंबर

कोर्स

फीस

दाखिले की तारीख

भरी गई किस्त

बैलेंस

रिमार्क

साधारणतः डाटाबेस फाईल मे एक फील्ड ऐसी होनी चाहिये, जिसमे प्रत्येक रिकार्ड का डाटा अलग हो। उदाहरणार्थ ऊपर

दिये गये डाटाबेस मे दो विद्यार्थियों के नाम एक समान हो सकते है। लेकिन रोल नंबर प्रत्येक विद्यार्थी का अलग–अलग होता है। इस फील्ड को Key Field (की फील्ड) कहते है, इस Key फील्ड से हम इन्डेक्स, सर्च कर सकते है। फील्ड के नाम तय करने के बाद आपको उसका प्रकार निश्चित करना पडता है। अर्थात उसमे किस प्रकार का डाटा डालना है। एक्सेस मे विभिन्न प्रकार के डाटा टाईप होते है, Character ,Numeric, Float, Date, Logic आदि। फील्ड मे किस प्रकार के घटक डालना है, तथा इसमे किस प्रकार कि गणनाएँ करना है, उसके अनुसार डाटा टाईप को तय किया जाता है।

नया टेबल बनाना

प्रत्येक डाटा बेस फाइल के अंदर विभिन्न फील्ड डाले जाते है, प्रत्येक फील्ड को अलग नाम दिया जाता है। फील्ड के नाम सावधानीपुर्वक बनाये। किसी भी फील्ड का नाम उसमे डाले जाने वाले रिकार्ड के अनुसार होना चाहिए। फील्ड का नाम बहुत बडा नही होना चाहिए। यद्यपी आप फील्ड का नाम टेबल बनाने के बाद भी बदल सकते है।

जैसे–जैसे किसी डाटाबेस फाइल का आकार बढते जाता है, या अन्य डाटाबेस फाइल से जुडते जाता है। या बहुतसी रिपोर्ट, क्यूरी, फॉर्म आदि उस पर आधारित होते है, तब उसके अंदर के फील्ड का नाम, प्रकार आदि बदलना मुश्किल होते जाता है। यदि आप टेबल मे किसी फील्ड का नाम बदलते है, तब आप उस पर आधारित फॉर्म, रिपोर्ट आदि सभी जगह उस

फील्ड का नाम बदलना पडता है। इसी तरह टेबल का नाम भी कार्य के अनुसार दिया जाता है। विभिन्न रिपोर्ट, क्यूरी के लिए फील्ड को कुछ तुकडो मे बांटा जाता है, उदाहरण के लिए address यह फील्ड रखने के बजाय उसे दो या तीन हिस्सो मे रखा जाता है उदाहरण के लिए Address1, Area, City आदि।

डाटाबेस बनाने के पहले निम्न बातों का ध्यान रखना आवश्यक है

1) साधारणतः कैरेक्टर प्रकार का डाटा टाईप प्रयोग किया जाता है। इसमे उस प्रकार की फील्ड भी आती है, जिस मे डाटा संख्या मे होती है, लेकिन उसकी गणीतीय गणनाएँ नही करना पडता है। उदाहरण फोन नं. पिन कोड नं. आदि

2) फील्ड को हमेशा छोटे तुकड़ों मे विभाजित कर डाले। उदाहरण Address देने के जगह उसे Area, Street Number, City इस तरह से बनाई जाती है। इससे डाटा को क्रमबद्व करना, खोजना आसान होता है।

3) फील्ड का नाम उसमे डालने वाले डाटा के अनुसार होना चाहिये। फील्ड का नाम बहुत बडा नही होना चाहिये।

4) एक्सेस मे डाटा यह Fix Width का होता है, अर्थात यदि आपने "Name" नाम की फील्ड की चौडाई (Width) "20" की रखी है, और यदि किसी रिकार्ड मे सिर्फ 10

कैरेक्टर ही डालते है। फिर भी एक्सेस उसके लिए 20 कैरेक्टर की ही मेमोरी खर्च करता है। इसलिए फील्ड कि चौडाई तय करते समय इस बात का ध्यान देना आवश्यक है, कि उस फील्ड मे अधिकतम कितने कैरेक्टर का डाटा आयेगा। उदाहरण के लिए हमने Fees नाम की फील्ड बनाई है, तथा हमे यह पता है, कि किसी भी कोर्स की फीस यह 10000 से ज्यादा नही है, तब इसकी चौडाई 6 से ज्यादा नही रखना चाहिए। यद्यपि एक्सेस मे बाद में फील्ड की चौडाई बदलने कि सुविधा है, इसलिए फील्ड कि चौडाई बहुत ज्यादा नही रखनी चाहिए।

5) जब एक काम के लिए एक से अधिक डाटाबेस फाइल बना रहे है, तब सभी डाटाबेस फाइल मे एक फील्ड Common होना चाहिए। इस फील्ड का नाम, फील्ड का प्रकार एवं चौडाई एक समान होनी चाहिए। साधारणतः इस प्रकार फील्ड को Numeric प्रकार मे रखा जाता है। लेकिन इसका प्रकार काम के अनुसार निश्चित किया जाना चाहीए। इस प्रकार के फील्ड को Key फील्ड भी कहा जाता है। साधारणतः इसी फील्ड पर Index किया जाता है।

एक्सेस मे नया टेबल बनाने के निम्न पद् हैं

- File मेन्यू क्लिक करें

- new विकल्प क्लिक करें, आपको दायें तरफ एक पैनल दिखाई देगा
- उसमे blank database क्लिक करें, आपको new database file का डायलॉग बॉक्स दिखाई देगा, उसमे डाटाबेस का इच्छित नाम टाइप करे।
- Create बटन को क्लिक करे।

आपको एक अन्य डायलॉग बॉक्स दिखाई देगा, जिसमे निम्न विकल्प होते है, उनकी सहायता से आप डाटाबेस डिजाइन कर सकते है।

1. Create Table in design view
2. Create table by using wizard
3. Create table by entering data

इसमे विजार्ड द्वारा इच्छित विषय या कार्य के लिए सरलता से टेबल बनाया जा सकता है, लेकिन अधिक जटिल एवं बडे टेबल बनाने के लिए design view का उपयोग किया जाता है।

Create Table in design view

main : Table

	Field Name	Data Type	
🔑	bookid	AutoNumber	start form 1
	nameofbook	Text	
	subject	Text	
	language	Text	
	author	Text	author name
	isbnno	Text	isbn no
	dofpub	Date/Time	
	price	Number	

Design view पर क्लिक करने से

नयी विंडो खुल जाती है, जिसमे रो और कॉलम कि संरचना दिखाई देती है। इसमे तीन कॉलम होते है

1 Field name

इस बॉक्स मे फील्ड का नाम टाइप किया जाता है। एक रो मे एक ही फील्ड का नाम टाइप किया जाता है। एक्सेस मे फील्ड देने के लिए निम्न नियम है

a. फील्ड का नाम कम से कम एक कैरेक्टर का होना चाहिए
b. फील्ड के नाम मे अधिकतम 64 कैरेक्टर ही होने चाहिए।
c. फील्ड के नाम मे कैरेक्टर, अंक एवं कुछ स्पेशल कैरेक्टर का प्रयोग किया जा सकते हैं।
d. फील्ड के नाम मे (. , !, [ ] ‘) आदि का प्रयोग नही किया जा सकता हैं।
e. फील्ड के नाम मे 0 से 31 तक के ascii कैरेक्टर का प्रयोग नही किया जा सकता है।
f. फील्ड का नाम खाली स्पेस के साथ नही चालू होनी चाहिए, यद्यपी आप कैरेक्टर के बीच मे स्पेस दे सकते है।
g. आप फील्ड का नाम upper case, lower case आदि मे दे सकते है, एक्सेस मे फील्ड के नाम case sensitive नही होते है। अर्थात कैपीटल या स्मॉल केस को एक समान ही मान कर चलता है। उदाहरण के लिए

studname और StudName यह दोनों एक ही माना जाता है।

यद्यपी फील्ड के नाम देने के बाद आप उसे आवश्यक्तानुसार बाद मे भी बदल सकते है, लेकिन यदि वह फील्ड अन्य आब्जेक्ट जैसे फॉर्म, क्यूरी आदि मे प्रयोग हो रहा है, तब उन सभी जगह आपको बदलाव करना पडता है। इसलिए फील्ड के नाम पहले से ही एक पेज पर बना कर फिर उसे एक्सेस मे डालना चाहिए।

2. Data Type

प्रत्येक फील्ड का एक डाटा टाइप सेट किया जाता है। डाटा टाइप यह उसमे डाले जाने वाले डाटा पर निर्भर होता है। उदाहरण के लिए हमे छात्र का नाम किसी फील्ड मे डालना है, तब उसका डाटा टाइप यह Text रखते है। प्रत्येक डाटा टाइप की अलग–अलग विशेषताए होती है। फील्ड का नाम डालने के बाद इस कॉलम पर जाते है, आपको एक drop down सूची दिखाई देती है, उनमे से इच्छित डाटा टाइप सिलेक्ट करें। एक्सेस मे निम्न प्रकार के डाटा टाइप उपलब्ध है।

Text Data Type :- जिन फील्ड मे कैरेक्टर का उपयोग हो रहा है, या जिन फील्ड के डाटा मे गणीतीय गणनाएँ नही करना है, उनमे इस डाटा टाइप का प्रयोग किया जाता है। उदाहरण के लिए studname इस फील्ड मे छात्रो के नाम डाले जाने है, इसलिए इसमे Text डाटा टाइप का उपयोग

किया जाता है। इसके अतिरिक्त ऐसे कुछ फील्ड हो सकते है, जिनमे अंक डाले जाते है, लेकिन उनका कभी भी गणतीय गणनाओं के लिये प्रयोग नही होना है, जैसे फोन नंबर, पिन कोड नंबर आदि इनमें भी इसी डाटा टाइप का उपयोग किया जाता है। इस प्रकार के डाटा टाइप अधिक सरल होते है, तथा इनमे सभी कैरेक्टर, अंक, स्पेशल कैरेक्टर आदि डाटा डाला जा सकता है। किसी भी टेक्स्ट डाटा टाइप के फील्ड मे अधिकतम 256 कैरेक्टर का ही डाटा डाला जाता है। इस रो और कॉलम की संरचना के नीचे फील्ड प्रापर्टी का डायलॉग बॉक्स दिखाई देता है। उसमे Field size मे आप फील्ड मे डाले जाने वाले अधिकतम कैरेक्टर की संख्या निश्चित कर सकते है। यह संख्या डाटा के अनुसार सेट करें। उदाहरण के लिए आपको छात्रो के नाम के लिए फील्ड बनाना है, तब उसे आप 50 रख सकते है, क्योकि किसी छात्र का नाम 50 से अधिक कैरेक्टर के होने कि संभावना बहुत की कम है। यदि फील्ड का आकार अधिक रखा है, तब अनावश्यक रूप से मेमोरी प्रयोग होती है। किसी भी फील्ड का आकार बहुत अधिक या बहुत कम नही रखना चाहिए। यद्यपी आप किसी फील्ड का आकार बाद मे बदल सकते है। लेकिन यदि आपने किसी फील्ड का आकार डाटा डालने के बाद कम किया है, तब किसी रिकार्ड मे जहाँ फील्ड का संपूर्ण आकार तक डाटा डाला है, वहॉ पर कुछ डाटा मिट सकते है।

Memo Data type :- मेमो फील्ड का प्रयोग जब किसी फील्ड मे अधिक मात्रा मे टेक्स्ट डाटा डालना है, तब होता है। उदाहरण के लिए हमने बैंक के ग्राहकों की डाटाबेस फाइल बनाई है। उस फाइल मे प्रत्येक ग्राहक का नाम, एकांउट नंबर, फोन नंबर आदि डालने के लिए फील्ड है, इन सभी प्रकार के फील्ड के लिए अलग–अलग **Field Type** है। लेकिन हमे प्रत्येक ग्राहक के बारे मे कुछ महत्वपूर्ण सूचना भी रखना है, जो एक पेज या उससे अधिक भी हो सकती है। उस दशा मे **Memo** फील्ड का प्रयोग किया जाता है। एक मेमो फील्ड के रिकार्ड मे अधिकतम 65,536 कैरेक्टर डाले जा सकते है। इस प्रकार के डाटा प्रकार मे **field size** डालने की आवश्यकता नही होती है।

Number Data type :- जहाँ पर अंक डालना है, तथा उन अंको की गणीतीय गणनाएँ करने के आवश्यक्ता हो सकती है, उस प्रकार के फील्ड मे इस प्रकार का डाटा टाइप डाला जाता है। उदाहरण के लिए हमे **phy, chem., math** इन विषयों के मार्क डालना है, तब इन फील्ड मे इस प्रकार का डाटा टाइप रखा जाता है। **field size** अधिकतम अंक से कुछ ज्यादा रखी जाती है। उदाहरण के लिये हमे पता है, की विषयो के अधिकतम मार्क 100 है, इस स्थिती मे हम उस फील्ड का आकार **byte** रखते है, अर्थात उसमे अधिकतम 255 तक संख्या डाली जा सकती है। इस प्रकार के फील्ड निम्न प्रकार के

डाटा टाइप है, उसे आप field size विकल्प मे सेट कर सकते है।

प्रकार	range	decimal संख्या	storage size
byte	0 से 255	नही	1 byte
integer	-32768 से 32767	नही	2 byte
long integer	-2147483648 से 2147483647	नही	4 byte
double	$-1.797x\ 10^{308}$ से $1.797x\ 10^{308}$	15	8 byte
single	$-3.4\ x\ 10^{38}$ से $3.4\ x\ 10^{38}$	7	4 byte
Replication ID	-	.	16 byte
Decimal	1- 28 precision	15	8 byte

डाटाबेस प्रोग्राम का उपयोग बहुत लंबे समय तक होता है, इसलिए भविष्य कि संभावनांओ को देखते हुए फील्ड का आकार निश्चित करे। बडे मान के आकार के लिए double रखा जाता है, तब डाटाबेस मे गणीतीय गणनाओं की गति कम हो सकती है, तथा मेमोरी की अधिक आवश्यकता होती है।

Date /Time data type :- इस प्रकार के डाटा टाइप का उपयोग तारीख एवं समय के मान संग्रहित करने के लिए होता है। इसमे डाले गये मानों की गणीतीय गणनाएँ कि जा सकती है। यदि किसी फील्ड मे तारीख डाली है, तब उसमे दिन, माह या साल की गणनाएँ कि जा सकती है। आप दो तारीख के बीच के दिन आदि कि गणना कर सकते है। आप तारीख के अनुसार किसी डाटा को क्रमबद्ध कर सकते है।

Currency :- यह एक प्रकार की **number** प्रकार का डाटा टाइप है, लेकिन इस गणीतिय गणनाओ के समय **round off** नही किया जाता है। इसमे डेसिमल संख्या के बाद 15 अंक रह सकते है। इस प्रकार के डाटा टाइप की गणना अधिक तेजी से होती है।

Auto Number :- यह भी एक **number** प्रकार का डाटा टाइप है, जब किसी फील्ड मे इस प्रकार का डाटा टाइप डाला जाता है, तब अगले रिकार्ड मे एक्सेस स्वयं ही रिकार्ड का मान एक निश्चित अंतराल मे बढा देता है। लेकिन एक टेबल मे एक ही फील्ड को इस प्रकार का डाटा टाइप दिया जा सकता है। इसमे **long integer** के समान अधिकतम सीमा होती है, तथा प्रत्येक रिकार्ड **4 byte** मेमोरी का प्रयोग करता है। लेकिन यह सिर्फ घनात्मक संख्या ही प्रदर्शित एवं संग्रहित करता है।

Yes/No :- इस प्रकार के डाटा टाइप मे सिर्फ दो ही मान हो सकते है। इसमे **yes** के लिए "1" और **no** के "0" संग्रहित किया जाता है। इसे "True/False", "Yes/No" आदि मान

डाल सकते है। इस प्रकार के फील्ड के लिए 1 byte का प्रयोग किया जाता है।

OLE object :- इस प्रकार के डाटा टाइप के उपयोग से फील्ड मे अन्य प्रोग्राम के आब्जेक्ट को संग्रहित किया जा सकता है। इसमे एमएस वर्ड की फाइल, एमएस एक्सल कि वर्कशीट, विडीयो आदि भी संग्रहित किये जा सकते है। इस प्रकार के फील्ड मे वास्तवीक आब्जेक्ट को नही दर्शाया जाता है, अपितु उसका आयकॉन दर्शाया जाता हैं। इस प्रकार के फील्ड को index नही किया जा सकता है।

सभी फील्ड डालने एवं उनके डाटा टाइप निश्चित करने के बाद उस विंडो को बंद कर दे। एमएस एक्सेस आपको टेबल को सेव करने का आदेश मांगता है।

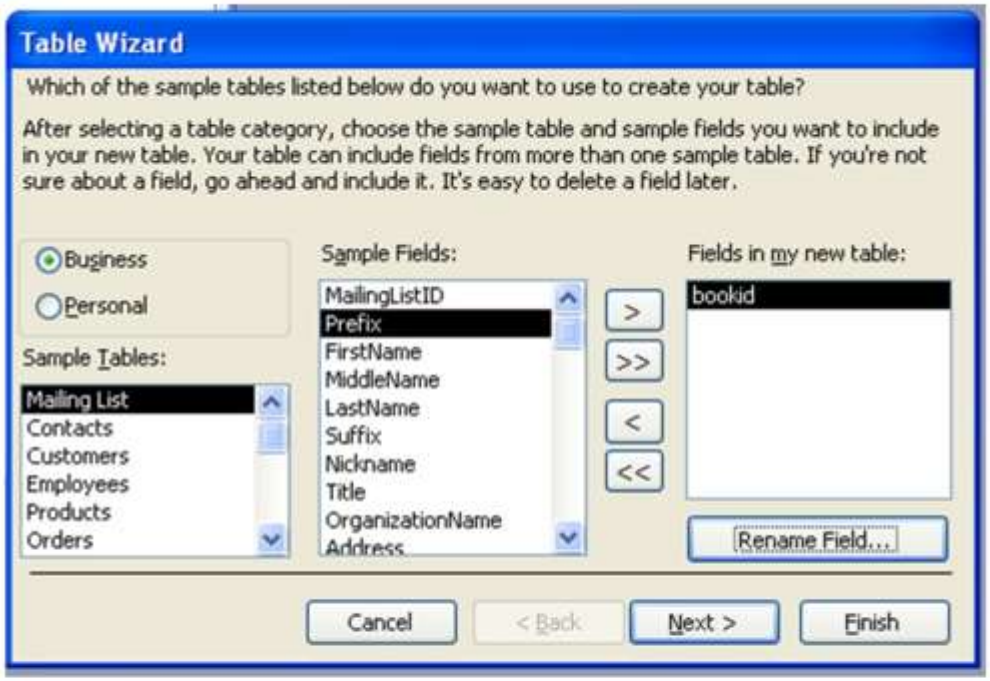
- "yes" बटन को क्लिक करे।
- आपको save as का डायलॉग बॉक्स दिखाई देगा। उसमे टेबल का उचित नाम टाइप करे।
- Ok बटन को क्लिक करे।

यदि प्रायमरी key सेट नही की है, तब एमएस एक्सेस उसकी चेतावनी संदेश दर्शाता है। यदि आप बगैर Primary Key के टेबल बनाना चाहते है, तब "Yes" बटन क्लिक करे। यदि "No" विकल्प का चुनाव करते है, तब वह वापस टेबल डिजाइन के डायलॉग बॉक्स मे जाता है, वहॉ पर नई फील्ड बना कर उस मे प्रायमरी key बनाए। अब डाटाबेस का मुख्य

डायलॉग बॉक्स दिखाई देता है। उसमे आपकी बनाई डाटाबेस टेबल का नाम दिखाई देता है।

Create table by using wizard

इस विकल्प का उपयोग सरल टेबल बनाने के लिए होता है। इसमे एमएस एक्सेस आपको टेबल बनाने मे सहायता प्रदान करता है। इसे क्लिक करने विजार्ड का पहला डायलॉग बॉक्स दिखाई देता है। इसमे किस प्रकार का डाटाबेस बनाना है, तथा उसमे कौन कौन सी फील्ड रखना है आदि निश्चित किया जाता है। इसमे तीन लिस्ट बॉक्स होते है।



पहले बॉक्स मे किस कार्य के लिए डाटाबेस फाइल बनाना है, वह सिलेक्ट करे। दूसरे बॉक्स उस विषय से संबंधित फील्ड की सूची दिखाई देती है। उस सूची मे इच्छित फील्ड को सिलेक्ट करें ">" बटन को

करे। सिलेक्ट कि गई फील्ड तिसरे बॉक्स मे आ जाती है। यदि सभी फील्ड को सिलेक्ट करना है, तब ">>" बटन क्लिक करे। किसी फील्ड का नाम बदलने के लिए "Rename field" बटन क्लिक करे। आपको rename field का डायलॉग बॉक्स दिखाई देगा, उसमे इच्छित नाम टाइप करें, तथा "OK" बटन क्लिक करें।

"Next" बटन को क्लिक करे। आपको विजार्ड का दूसरा डायलॉग बॉक्स दिखाई देगा। उसमे प्रायमरी key सेट करने का विकल्प दर्शाया जाता हैं। यदि आप प्रायमरी की सेट करना चाहते है, तब दूसरे बटन को सिलेक्ट करे। यदि पहला विकल्प सिलेक्ट किया है, तब एमएसएक्सेस स्वयं ही प्रायमरी की सेट कर लेता है। इस बॉक्स मे टेबल का नाम सेट करे। एक्सेस

स्वयं की ओर से एक नाम देता है, यदि नया नाम देना है, तब "what do you want to name

your table" बॉक्स मे नया नाम टाइप करें। फिर से "Next" बटन को क्लिक करे।

आपको विजार्ड का तीसरा डायलॉग बॉक्स दिखाई देगा। इसमे नया बनने वाला टेबल अन्य किसी टेबल के साथ जोडना है, या नही इस बारे मे जानकारी देना पडता है। यदि किसी अन्य टेबल के साथ सम्बन्ध जोडना है, तब "Relationship" बटन को क्लिक करे। उसे क्लिक करने पर "relationship" का डायलॉग बॉक्स दिखाई देता है। यदि उस डाटाबेस मे और भी टेबल है, तब उसके साथ किस प्रकार की Relationship सेट करना है, उसके लिए तीन विकल्प दिये है।

The table aren't related :- यदि नये टेबल को अलग ही प्रयोग करना है, पुराने या दूसरे टेबल के साथ जोडना नही है, तब इस विकल्प का प्रयोग करें।

"one record in "table1" table will match many record in "table2" table: - पहले या पुराने टेबल का एक रिकार्ड नये टेबल के विभिन्न रिकार्ड के जोडना है, तब इस विकल्प का प्रयोग किया जाते है। इसे one to many relationship कहा जाता है।

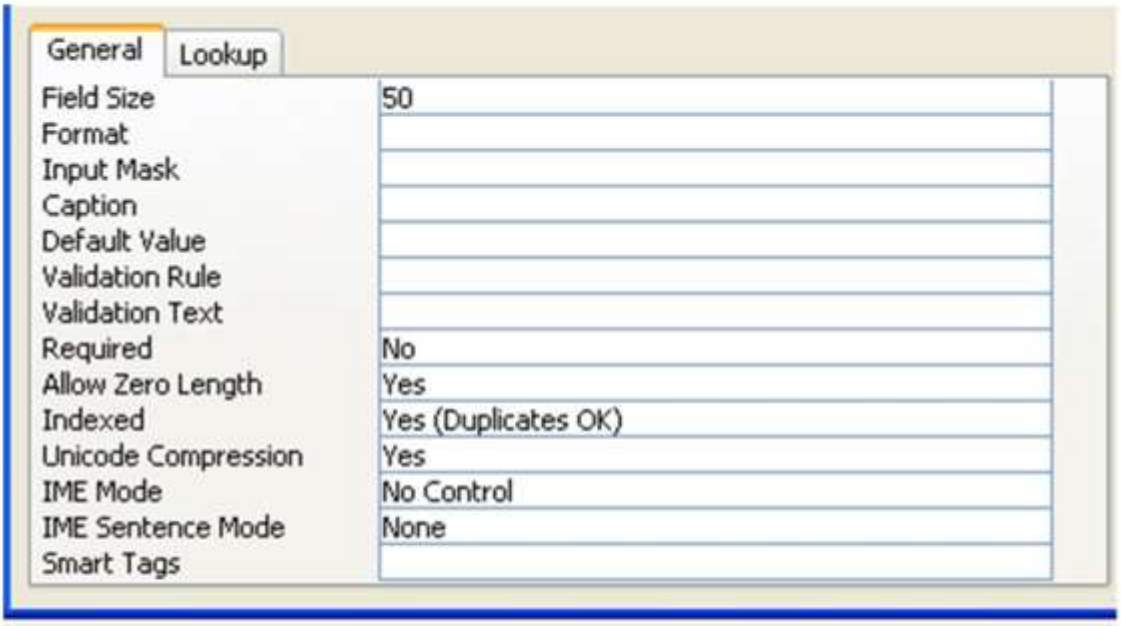
One record in "table2" will match many record in "table1" table :- यदि नये टेबल का एक रिकार्ड पुराने टेबल के विभिन्न रिकार्ड के साथ जोडना है, तब इस विकल्प का उपयोग किया जाता है।

- उसमे दूसरे टेबल को सिलेक्ट करे। "Next" बटन को क्लिक करे।

विजार्ड को अंतिम डायलॉग बॉक्स दिखाई देता है। जिसमे तीन विकल्प होते है। यदि आपको इसके अतिरिक्त भी टेबल मे बदलाव करना है, तब पहला विकल्प का चुनाव करे। यदि आपको विजार्ड द्वारा बनाये टेबल को ही अंतिम रूप देना है, तब दूसरा विकल्प का चुनाव करें, तथा सीधे रो और कॉलम की संरचना मे डाटा डालने का कार्य कर सकते है। यदि आपको विजार्ड द्वारा बनाये गये फॉर्म के लेआउट मे डाटा डालना है, तब तिसरे विकल्प का चुनाव करे।

- Finish बटन को क्लिक करे।

फील्ड प्रापर्टी सेट करना



डाटाबेस प्रोग्राम की सफलता डाटाबेस फाइल के डिजाइन मे होती है। डाटाबेस फाइल बनाने मे फील्ड की प्रापर्टी एक महत्वपूर्ण भाग है। फील्ड की विभिन्न प्रापर्टी से फील्ड के डाटा का फारमेट निश्चित किया जाता है। एक्सेस मे फील्ड मे डाटा डालने के लिए विभिन्न प्रापर्टी दि है। जब आप कोई टेबल की डिजाइन बनाते है, तब फील्ड बनाते समय टेबल डिजाइन के नीचे फील्ड प्रापर्टी का डायलॉग बॉक्स दिखाई देता है। आपके चुने हुए डाटा टाइप के अनुसार प्रापर्टी बॉक्स मे विभिन्न प्रापर्टी दर्शाई जाती है। प्रत्येक डाटा टाइप की अलग–अलग प्रापर्टी होती है। उदाहरण के लिए decimal places यह प्रापर्टी numeric फील्ड मे होती है, text डाटा टाइप मे यह प्रापर्टी

नही होती है। लेकिन कुछ प्रापर्टी सभी प्रकार के डाटा टाइप मे सेट कि जा सकती है। कुछ प्रापर्टी मे मान आप कार्य के अनुसार टाइप कर सकते है, तथा कुछ प्रापर्टी के मान उसमे दिये गये मानों मे से ही सिलेक्ट करना पडता है। निम्न कुछ महत्वपूर्ण प्रापर्टी अधिक प्रयोग होती है

Field size :- बनाई हुई फील्ड का अधिकतम आकार सेट करने के लिए इस प्रापर्टी का उपयोग होता है। उदाहरण के लिए आप ने एक फील्ड बनाई है, जिसमे **text** डाटा टाइप सेट किया है, यदि इस प्रापर्टी मे 50 सेट करते है। तब उस फील्ड के रिकार्ड मे अधिकतम 50 कैरेक्टर टाइप कर सकते है।

new Value :- यह प्रापर्टी **auto-number** डाटा टाइप के लिए प्रयोग होती है। इस विकल्प मे प्रत्येक रिकार्ड के नंबर किस तरह से बढना चाहिये, यह सेट कर सकते है।

Format :- फील्ड मे डाटा डालने के बाद, वह किस फॉरमेट मे दर्शाना है, यह इस विकल्प से सेट किया जा सकता है। उदाहरण के लिए आपने कोई **date/time** डाटा टाइप सेट किया हैं। डाटा डालने के बाद तारीख कौन से फारमेट मे दर्शाना यह इस विकल्प से सेट कर सकते है। इस विकल्प मे विभिन्न फारमेट की सूची दर्शाइ जाती है, उनमे से इच्छित फारमेट सिलेक्ट करे।

Input Mask : यदि कोई डाटा किसी एक निश्चित श्रंखला मे डालना है, उदाहरण के लिए हमे फोन नंबर मे **std** कोड के

बाद "-" डालना है, तथा उसके बाद फोन नंबर डालना है, जैसे 0712-2705674 तब इस प्रापर्टी का उपयोग किया जाता है। इसका उपयोग साधारणतः फोन नंबर, कोड नंबर, पिनकोड नंबर आदि डाटा डालने के लिए होता है। यद्यपी यह डालना आवश्यक नही है, लेकिन इस सेट करने से रिकार्ड टाइप करने मे आसानी होती है।

Decimal Places :- किसी numeric फील्ड मे दशांस (decimal) के बाद कितने बिंदु आना है, वह इस प्रापर्टी से सेट किया जाता है।

Caption :इच्छित फील्ड को एक और अन्य नाम देने के लिए इस प्रापर्टी का उपयोग किया जाता है। caption का उपयोग फॉर्म, क्यूरी आदि मे किया जाता है। कुछ फील्ड का नाम बहुत कठिन हो जाता है, जैसे Empcode_mkt यह नाम फॉर्म, रिपोर्ट मे बार बार टाइप करना मुश्किल हो जाता है, तथा ऐसे नाम को याद रखना भी मुश्किल होता है, तब caption का प्रयोग कर हम इस नाम का आसान कर सकते है। caption मे डाले गये नाम का प्रयोग फील्ड के नाम के जगह कर सकते है।

Default value :- किसी फील्ड मे डाटा डालने के पहल एक्सेस द्वारा एक मान दर्शाया जाता है, उस मान को default value कहा जाता हैं। इस default value को आप इस प्रापर्टी द्वारा सेट कर सकते है। यदि इसमे कुछ मे नही डाला है, तब उस डाटा डालते समय फील्ड के समान खाली बॉक्स

दर्शाया जाता हैं। numeric प्रकार के फील्ड मे "0" यह एक्सेस के तरफ से default value रहती है। आप इसे कार्य के अनुसार बदल सकते है। उदाहरण के लिए आपको किसी date प्रकार के फील्ड मे वर्तमान दिन की तारीख यह default value चाहिए, तब वह कार्य इस प्रापर्टी मे data() फंक्शन द्वारा डाल सकते है। जब भी उस टेबल मे डाटा डालेंग तब उस फील्ड मे उस दिन की तारीख स्वयं ही आ जायेगी। यद्यपी फील्ड का मान आप कभी भी बदल सकते है।

Required :- टेबल मे डाटा एन्ट्री करते समय यदि आप चाहते है, वांछित फील्ड को खाली रखना है, तब इस प्रापर्टी से यह कार्य किया जा सकता है।

Index :- यदि आपको किसी फील्ड पर टेबल को क्रमबद्ध करना है, तब इस प्रापर्टी से आप यह कार्य कर सकते है।

उपरोक्त प्रापर्टी के अतिरिक्त भी बहुत सी प्रापर्टी सेट कि जा सकती है।

डाटा फारमेट करना

डाटा को फारमेट करने के लिए format प्रापर्टी का उपयोग होता है। प्रत्येक डाटा टाइप के लिए अलग–अलग फारमेट दिये है। उदाहरण के लिए data डाटा टाइप के लिए विभिन्न तारीख के फारमेट दिये है, Numeric डाटा टाइप के लिए अलग–अलगनंबर के फारमेट दिये है।

उदाहरण के लिए "!" (Exclamation mark) से टेक्स्ट प्रकार के डाटा सेल के दाये तरफ अलाइन होता है। numeric प्रकार डाटा टाइप के निम्न फारमेट है

format	डाला गया अंक	एक्सेस मे दर्शाया हुआ अंक	प्रापर्टी बॉक्स मे सेट किया फारमेट
General	123456.123	123456.1	######.#
Currency	123456.123	$123,456.12	$###,###.##
Fixed	123456.123	123456.12	######.##
Standard	123456.123	123,456.12	###,###.##
percent	.12	12%	###.##%
Scientific	123456.123	1.22E+05	###E+00

numeric डाटा टाइप के कुछ अन्य फारमेट

. (dot):- डेसीमल बिंदु कहॉ पर रखना है, यह निश्चित किया जाता है।

, (comma):- इसे thousand separators कहा जाता है, इसे बडी संख्या को आसानी से पढने के लिए प्रयोग होता है।

Date/Time डाटा टाइप के फारमेट

एक्सेस मे तारीख विभिन्न प्रकार से दर्शाया जा सकता है। इसमे तारीख के लिए निम्न फारमेट उपलब्ध है।

General date :- इस फारमेट मे सिर्फ तारीख दर्शाई जाती है, समय नही दर्शाया जाता है।

Long date :- इस फारमेट मे दिन का नाम, माह, दिनांक एवं साल दर्शाया जाता है । उदाहरण Friday, March 5, 2013

Medium date :- इस फारमेट मे दिन का नाम नही दर्शाया जाता है। उदाहरण 5-Apr-2013

Short date :- इस फारमेट मे माह का नाम नही दर्शाया जाता है, अपितु उसे भी अंक मे दर्शाया जाता है। उदाहरण 4-5-2013

Long Time :- इस फारमेट मे समय मे सेकंड, मिनट, एवं घंटा एवं am या pm दर्शाया जाता है। उदाहरण 1:10:34 PM

Medium time :- इसमे सेकंद नही दर्शाये जाते है। उदाहरण 1:10 PM

Short time :- इस फारमेट मे am या pm नही दर्शाया जाता है। उदाहरण 1:10:34

तारीख के फारमेट बदलने के लिए निम्न परिमाणों का प्रयोग किया जाता है।

/ (forward slash) :- तारीख मे "-" की जगह "/" दर्शाया जाता है।

dd :- इस विकल्प से तारीख को दो डिजिट मे दर्शाया जाता है, उदाहरण की लिए 4 के जगह 04 दर्शाया जाता है।

ddd :- दिन का नाम तीन कैरेक्टर मे दर्शाया जाता है। जैसे Mon, Thu, Wed आदि

dddd:- दिन का नाम पूर्ण तरीके से दर्शाया जाता है। उदाहरण Monday, Tuesday, Wednesday आदि

ww :- साल का कौन सा सप्ताह है, यह दर्शाया जाता है। यह 0 से 53 तक होता है। उदाहरण के लिए यदि तारीख 4 Jan 2012 है तब 1 दर्शायेगा, क्योकि यह साल का पहला सप्ताह है।

mmm :- माह के पहले तीन कैरेक्टर दर्शाये जाते है, जैसे Jan, Feb, Mar आदि

mmmm:- माह के सभी कैरेक्टर दर्शाये जाते है, जैसे January, February आदि

फारमेट के कुछ उदाहरण

format	डाला गया डाटा	फारमेट किया डाटा
>	Narendra Publication	NARENDRA PUBLICATION
Currency	12345.67	$12,345.67
“Pin No.”, 000000	440009	Pin No. 440009

mmm yy	04/05/2013	Apr 13
Long Date	04/05/2013	Thursday, April 5, 2013

टेबल मे रिकार्ड जोडना

एक बार टेबल को सेव करने के बाद आप उसमे इच्छित रिकार्ड डाल सकते है। किसी टेबल मे रिकार्ड डालने के लिए

1. टेबल के नाम पर दो बार क्लिक करे।
2. ऊपर की ओर **open** विकल्प को क्लिक करे।

आपको रो और कॉलम की संरचना दिखाई देगी उसमे ऊपर की रो पर फील्ड के नाम दर्शाए जाते है। प्रत्येक रो मे एक रिकार्ड संग्रहित होता है। एक सेल से दूसरे सेल मे जाने के लिए **Tab** बटन का प्रयोग करें।

टाइप किया डाटा सेव करने करने के लिए

- File मेन्यू के **save** विकल्प को क्लिक करे।

रिकार्ड को सिलेक्ट करना

इच्छित रिकार्ड को सिलेक्ट करने के दो तरीके है

1. उस रिकार्ड के सबसे दाये तरफ के रो सिलेक्टर पर क्लिक करे। या

2. उस रिकार्ड पर कही भी क्लिक करें, edit मेन्यू के Select row विकल्प को क्लिक करे।

एक से अधिक रिकार्ड सिलेक्ट करने के लिए shift बटन का प्रयोग करे।

Column को freeze करना

यदि टेबल मे बहुत सी फील्ड है, तब एक स्क्रिन पर सभी फील्ड नही दिखती है। यदि आप चाहते है, कोई महत्वपूर्ण फील्ड हमेशा स्क्रिन पर रहे, उसके लिए उस फील्ड को freeze किया जाता हैं इससे वह फील्ड स्क्रिन पर ही रहती है, बाकी स्क्रिन दायें या बायें होते रहती है। कोइ फील्ड को freeze करने के लिए

- इच्छित कॉलम के हेडिंग रो को सिलेक्ट करें, संपूर्ण कॉलम सिलेक्ट हो जाता है।
- माऊस को right click करें, उस मे freeze column विकल्प को क्लिक करे।

Saving

अब आप अपना डाटाबेस बनाते है, उसी समय बनाने से पहले यह आपको अपन डाटाबेस का नाम बताना होता है। टेबल आदि को बनाने के बाद आपको क्रास पर क्लिक करना होता है। जब आप क्लिक करते है, तो यह आपके टेबल आदि को किस नाम से सेव करना है। आप उसका नाम देकर आप अपने रिकार्ड को सेव कर सकते है।

टेबल Copy करना

- जिस टेबल की संरचना कॉपी करना है, उस पर right click करें।
- उसमे निचे की ओर copy विकल्प को क्लिक करे।
- जिस डाटाबेस मे पेस्ट करना है, उसे खोलें
- फिर से right click लिए करे, तथा paste विकल्प को क्लिक करें। नये टेबल का नाम टाइप करें ,तथा "Structure only" विकल्प का चुनाव करें। तथा Ok बटन को क्लिक करें।

Delete a table

- एमएस एक्सेस में टेबल को हटाने के निम्न चरण है
- इच्छित टेबल को right Click करें।
- एक मेनू दिखाई देता है, उसमें Delete विकल्प कों क्लिक करे, एमएस एक्सेस आपको एक चेतावनी संदेश देता है, यदि आपको फिर भी टेबल डिलीट करना है "Yes" बटन को क्लिक करें।

MS Access

Unit 3

working with query, setting up relationships- define relationship, add a relationship, set a rule for differential integrity, change the joint type, delete a relationship, save relationship, where is and filter- difference between queries and filter, filter using multiple fields AND, OR, advanced filter queries with one table, find record with select query, find duplicate record with query, find unmatched record with query, run the query, same and change query

Unit -3

Primary Key

एक्सेस मे बनी डाटाबेस फाइल मे primary key का बहुत महत्व है। टेबल को उस key के आधार पर दूसरे टेबल के साथा जोडा जा सकता है। प्रत्येक टेबल मे एक primary key सेट की जाती है। primary key ऐसे फील्ड पर लगाई जाती है, जिसका प्रत्येक रिकार्ड मे अलग–अलग मान हो। उदाहरण के लिए हमने बैंक के ग्राहकों का डाटाबेस फाइल बनानी है, तब एकांउट नंबर को प्रायमरी key बना सकते है। किसी दो ग्राहक के नाम एक समान हो सकते है, या एक ग्राहक के एक से अधिक प्रकार के एकाउंट हो सकते है, लेकिन प्रत्येक ग्राहक का अलग एकांउट नंबर होता है। यदि नई डाटाबेस प्रणाली बना रहे है, तब हम एक ऐसी फील्ड बनाते है, जिसमे प्रत्येक फील्ड मे अलग–अलग मान के रिकार्ड हो। यदि आपने किसी टेबल मे ऐसी फील्ड नही बनाई है, तब एक्सेस स्वयं ही primary key फील्ड बना लेता है। साधारणतः primary key का डाटा टाइप numeric रखा जाता है, एवं इसके लिए Autonumber डाटा टाइप अधिक उचित है।

जिस फील्ड पर primary key लगाई जाती है, वह हमेशा index रहती है। इसका उपयोग क्यूरी, रिपोर्ट मे इच्छित डाटा अधिक तेजी से प्राप्त करने के लिए होता है। जब टेबल मे नये रिकार्ड जोडते है, तब एक्सेस primary key फील्ड मे उसी के समान रिकार्ड को खोजता है, यदि उसे duplicate रिकार्ड प्राप्त होते है,

तब उस डाटा को लेने से मना करता है। उदाहरण के लिए आपने बैंक के टेबल मे एक समान दो एकांउट नंबर डालने का प्रयास किया है, तब एक्सेस चेतावनी संदेश देता है, तथा उस डाटा लेने से इन्कार करता है। साधारणतः एक्सेस टेबल को primary key के आधार पर प्रदर्शित करता है। कोई टेबल की डिजाइन बनाने से पहले कौन से फील्ड पर primary key लगाना है, वह निश्चित करना आवश्यक है। प्रायमरी निश्चित करने से पहले निम्न बिंदुओ का ध्यान रखना आवश्यक है।

1. प्रत्येक रिकार्ड मे उस फील्ड मे अलग मान होना चाहिए।
2. प्रायमरी की मे null मान नही होना चाहिए, अर्थात उसमे कुछ मान होना आवश्यक है।
3. जब भी को नया रिकार्ड बनाते है, तब या तो एक्सेस द्वारा उसमे डाटा डालना चाहिए या प्रयोगकर्ता द्वारा डाटा डाला जाना चाहिए।
4. प्रायमरी की का डाटा बहुत बडा नही होना चाहीए, इससे गलती होने की संभावना अधिक होती है।
5. एक बार प्रामयरी की मे डाटा डालने के बाद उसमे बदलाव करने की आवश्यकता नही होनी चाहिए।

Primary key बनाना

एमएस एक्सेस मे प्रायमरी की बनाने के निम्न तरीके है,

1. जिस फील्ड को प्रायमरी की लगाना है, उसे सिलेक्ट करें, ऊपर की ओर Primary key बटन को क्लिक करे। यह बटन एक चाबी के आकार की होती है।

2. जिस फील्ड पर प्रायमरी की लगाना है, उसका right क्लिक करें, तथा उसमे primary key विकल्प को क्लिक करे।
3. टेबल को प्रायमरी की बनाये बगैर save किया है, तब एक्सेस स्वयं ही एक primary key बना लेगा।

जिस फील्ड को प्रायमरी कि लगाया जाता है, उसके आगे चाबी के आकार का आयकान दर्शाया जाता है।

Foreign key

यह टेबल की फिल्ड होती है, जिसमे दूसरे टेबल के मान दर्शाये जाते है, इसका प्रयोग दो टेबल के बीच संबंध स्थापित करने के लिए होता है। जिस टेबल मे foreign key फिल्ड होती है, उसे referenceing टेबल कहा जाता है, तथा जिसमे candidate key होती है, वह referenced टेबल कहा जाता है। इस key का मुख्य उद्देश्य reference टेबल के अपेक्षित रो या रिकार्ड को पहचानना है। साधारणतः forign key का मान candidate key के समान होना चाहिए। निम्न उदाहरण मे दो टेबल है। पहला टेबल "Cource code" मे C_code फिल्ड यह Candidate key है, तथा दूसरे टेबल "student database" मे "C_code" यह Foreign key है।

Cource code	
C_code	cource
DCA1	diploma sem 1
DCA2	Dipoma sem2
PGDCA1	post Graduate

Studetent Database		
roll no	Name	C_code
3	राजेश	DCA2
5	संजय	PGDCA1
7	नरेन्द्र	BCA1

	diplom sem1			
PGDCA2	Post Graduate diploma sem2	11	आकाश	BCA1
BCA1	Bachole sem 1	17	विशाल	DCA1

डाटाबेस डिजाइन मे foreign key की भुमिका बहुत महत्वपूर्ण है। दो टेबल के बीच मे जब relationship होती है, तब सही रिकार्ड प्राप्त करने मे foreign key उपयोगी होती है।

Candidate key

एक entity set मे दो entities के लिए दो एट्रीब्युट के मान समान नही होना चाहिए। key हमें एट्रीब्युट के एक set को पहचानने कि स्वीकृति देती है, जो प्रत्येक entity को एक दूसरे से पृथक करने के लिए पर्याप्त है। keys relationship को अलग–अलग पहचानने मे भी मदद करती है। अतः इस कारण हम relationship को एक दूसरे से पृथक कर सकते है।

entity set ग्राहक का customer-id एट्रीब्युट customer entity को अन्य से पृथक करने के लिये पर्याप्त है। अतः customer-id एक सुपर key है। इसीप्रकार Customer_name एवं customer_id का योग entity set customer के लिए एक supper key है। super key बाह्य एट्रीब्युट को रखता है। यदि K एक superkey है, तो S_o उसका super set है। हम उन supper key के प्रति लगाव रखते है, जिनके लिए कोई भी paper subset, superkey ना हो। यह न्यूनतम superkey को candidate key कहते है। केवल एक एट्रीब्युट ही candidate key हो सकता है। हम प्रायमरी key का प्रयोग candidate key

को अंकित करने के लिए करते है। हमें ऐसे primary key चुनना चाहिये जिसमे कि इसके एट्रीब्युट मे कभी भी परीवर्तन ना हो या बहुत कम परीवर्तन हो।

Table relationship

Ms-access मे RDBMS (Relational data base management system) कि सुविधा है, अर्थात आप एक टेबल का डाटा दूसरे टेबल के साथ साझा कर सकते है। छोटे एवं सरल डाटा के लिए हम एक टेबल मे इच्छित फील्ड संरचना बना कर उसमे रिकार्ड डाल कर कार्य कर सकते है। लेकिन बडे एवं पेचिदा डाटा के लिए अलग–अलगटेबल बनाये जाते है। एक प्रोजेक्ट मे साधारणतः सभी डाटा को एक डाटाबेस फाइल मे नही रखा जाता, अपितु कुछ अलग–अलगफाइल बना कर रखा जाता है। एक बडी डाटा–बेस फाइल मे काम करना उसी तरह है, जैसे ऑफिस के कागज एक ही फाइल मे डाल दिए है। और छोटे डाटा बेस फाइल मे काम करना उसी तरह है, जैसे प्रत्येक काम के लिए अलग फाइल बनाई है, तथा उससे सम्बन्धित कागज उस फाइल मे डाले है। छोटे डाटा बेस फाइल बनाने के निम्न कारण

1) डाटा बेस फाइल का डाटा बेस Structure बहुत बडा हो जाती है, इस कारण उसे संभालने मे मुश्किल हो सकती है।
2) एक डाटा बेस फाइल मे सभी डाटा डाला जायें तब वह फाइल बडी हो सकती है, जिससे उसका Backup लेना मुश्किल हो सकता है।

3) बडे Structure के डाटा बेस फाइल पर इंडेक्स करने मे अधिक वक्त लगता है।
4) छोटे फाइल को खोलने एवं काम करने मे कम समय लगता है।
5) जिस समय जिस डाटा कि आवश्यकता होती है, सिर्फ वही डाटा खुलता है, जिससे काम जल्दी एवं आसान हो जाता है।
6) यदि एक फाइल मे बडा डाटा है, और उसमे कुछ समस्या आती है, तब सभी डाटा खराब हो सकता है, लेकिन यदि छोटी फाइल बनाई है, एवं किसी एक फाइल मे समस्या आती है, तब उसी फाइल का डाटा खराब होने की संभावना होती है, अन्य फाइल का डाटा सुरक्षित रहता है।

RDBMS के निम्न लाभ है

1. इस प्रणाली के प्रयोग से बडे एवं बहुत बडे डाटा को आसानी से संभाला जा सकता है।
2. इच्छित डाटा को खोजना आसान हो जाता है।
3. बडा डाटाबेस पर कार्य करने के बावजुद प्रत्येक टेबल की संरचना सरल होती है।
4. एक से अधिक टेबल का प्रयोग किया जाता है, जिससे एक से अधिक प्रयोग कर्ता प्रणाली का प्रयोग कर सकते है।
5. डाटाबेस administrator के पास संपूर्ण कंन्ट्रोल रहता है, जिससे वह वांछित प्रयोगकर्ता को सिर्फ आवश्यक डाटा ही प्रदान करता हैं। इससे डाटा की सूरक्षा अधिक बेहतर होती है।

6.इस प्रणाली मे client –server तकनीक का भी प्रयोग किया जा सकता है, जिससे नेटवर्क पर आधारित अप्लिकेशन मे इस प्रकार के डाटाबेस तकनीक का प्रयोग किया जा सकता है।
7. बडे डाटाबेस प्रणाली मे इस तकनीक के प्रयोग से डाटा तेजी से प्राप्त किया जा सकता हैं। क्योकी प्रयोगकर्ता को जो डाटा चाहिए, सिर्फ उस पर ही कार्य किया जाता है, अन्य डाटा पर प्रक्रिया नही कि जाती है।
8. अलग अलग प्रकार के डाटा के अलग अलग टेबल का प्रयोग किया जाता है। डाटाबेस का रखरखाव अधिक आसान हो गया है।

उदाहरण के लिए हम एक बडे कंपनी के डाटाबेस को लेते है, जिसके अलग–अलगशहरों मे ऑफिस है, तथा उसमे बहुत से कर्मचारी काम करते है। इस प्रकार के डाटाबेस को एक टेबल मे मे संभालना मुश्किल हो सकता है। इसमे लिए हम तीन फाइल बनाते है।

Empdata :- इस फाइल मे प्रत्येक कर्मचारी का व्यक्तिगत डाटा जैसे कर्मचारी कोड, नाम, पता, फोन नंबर, योग्यता, अनुभव आदि डाटा रखा जाता है।

Empacc : इस डाटाबेस फाइल मे कर्मचारी की वेतन से जुडी जानकारी रखी है, जैसे कर्मचारी कोड, पद, मूल वेतन, विभिन्न भत्ते, कर्ज, आदि का डाटा रखा है।

Empprof:- इस डाटाबेस मे कर्मचारीयो के कार्य से जुडी जानकारी रखी है, जैसे कर्मचारी कोड, विभाग ,उपस्थिती, कार्यशैली इत्यादी

एक दुसरे से जुडे सभी टेबल मे एक फील्ड एक समान होना आवश्यक है। हमने कर्मचारी कोड को तीनो डाटाबेस फाइल मे रखा है। कर्मचारी कोड फील्ड से हि हम एक कर्मचारी की विभिन्न जानकारीयॉ अलग–अलगफाइल से प्राप्त कर सकते है।

अलग–अलग टेबल बनाने से डाटा कि पुनरावृर्ती कम हो जाती है। एक और उदाहरण लेते है, जिसमे विभिन्न ग्राहक विभिन्न उत्पाद की खरीदी का आर्डर देते है, यदि हम एक ही टेबल का उपयोग करते है, तब हमे प्रत्येक नये आर्डर का रिकार्ड डालना पडेगा, जिसमे खरीदने वाले का नाम, उत्पाद का नाम आदि प्रत्येक बार डालना पडेगा। लेकिन यदि यही काम हम अलग टेबल बना कर करते है, तक हमे सिर्फ नये आर्डर कि तारीख एवं उत्पाद की मात्रा डालना पडेगा, ग्राहक का नाम एवं उत्पाद की सूची हम दूसरे टेबल से प्राप्त कर सकते है। एक से अधिक टेबल का प्रयोग करने से डाटा को अच्छे तरीके से प्रबंधित किया जा सकता है।

जब दो या अधिक टेबल के बीच रिलेशन सेट करना है, तब उन सभी टेबल मे एक फील्ड common होना आवश्यक है। वह कॉमन फील्ड साधारणतः primary key की फील्ड होती है। इस फील्ड मे प्रत्येक रिकार्ड मे अलग मान होना आवश्यक है। उदाहरण के लिए empcode प्रत्येक टेबल यह फील्ड रहेगी, तथा वह प्रत्येक टेबल मे प्रायमरी की होगी।

एक्सेस मे टेबल के बीच तीन प्रकार की रिलेशनशिप हो सकती है।

One-to-one
One to many
Many to many

One to one

इस प्रकार के रिलेशनशिप मे एक टेबल कि फील्ड दूसरे टेबल के सिर्फ एक फील्ड से जुड सकती है । साधारणतः इस प्रकार की रिलेशनशिप RDBMS मे नही प्रयोग होती है। इसका प्रयोग जब एक टेबल मे फील्ड की संख्या 255 से अधिक हो रही है, तब किया जाता है। यद्यपी इसकी संभावना बहुत कम होती है, कि किसी टेबल मे इतने अधिक फील्ड का प्रयोग हो रहा हो।

One to many

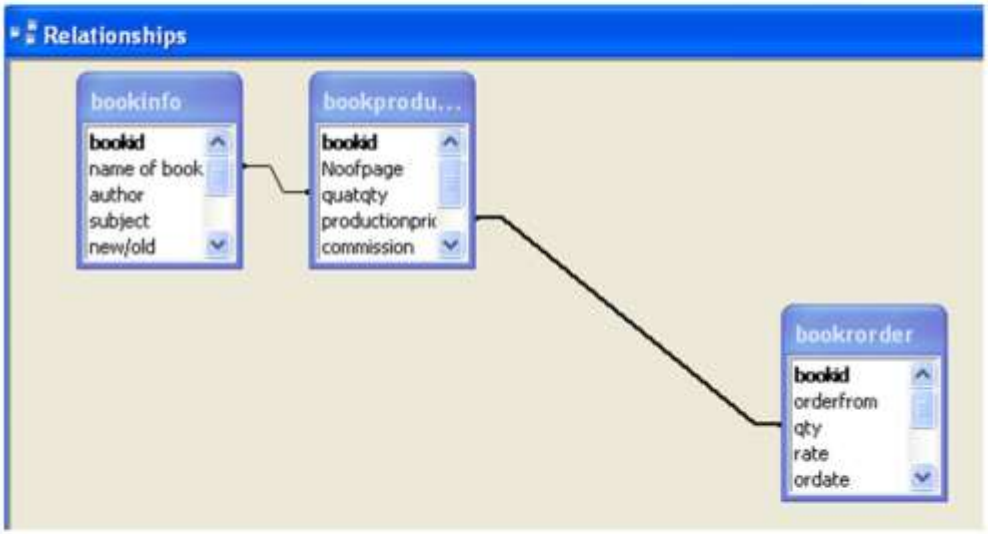
इस प्रकार कि संरचना बहुत आम है, उदाहरण के लिए किसी क्लास मे 25 छात्र है, तथा उनका एक शिक्षक है। इसमे एक शिक्षक का सम्बन्ध सभी 25 छात्रों से है। वैसे ही दो टेबल के बीच मे इस प्रकार का सम्बन्ध सेट कर सकते है। इस प्रकार की रिलेशन मे पहले टेबल के एक से अधिक रिकार्ड दूसरे टेबल के एक रिकार्ड के साथ जुड सकते है। उदाहरण के लिए एक कंपनी तीन अलग–अलग उत्पाद "A, B और C का व्यापार करती है, इन्हे विभिन्न ग्राहको को बेचा जाता है। यदि हमे यह पता लगाना है, कि उत्पाद A कौन कौन से व्यक्ति को बेचा है, तब उसे एस प्रकार की रिलेशनशिप की सहायता से देखा जा सकता है। लेकिन दूसरे टेबल का एक रिकार्ड पहले टेबल के एक ही रिकार्ड से जुडा होता है। एम एस एक्सेस मे टेबल के बीच मे इस प्रकार

की रिलेशनशिप सेट करना आसान है। इसे Parent-child रिलेशनशिप भी कहा जाता है।

Many to Many

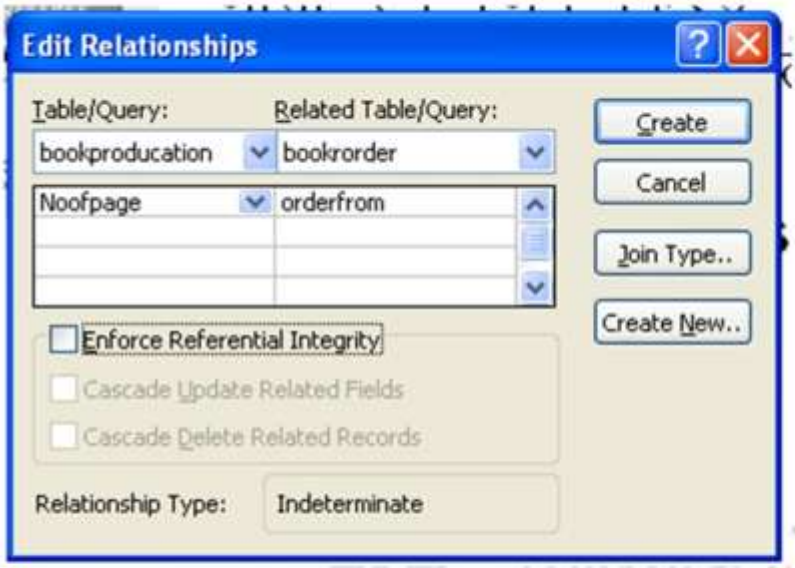
इस प्रकार के रिलेशनशिप मे एक टेबल के विभिन्न रिकार्ड दूसरे टेबल के विभिन्न रिकार्ड के साथ जुड सकते है। इस प्रकार के रिलेशनशिप को समझने के लिए हम बैंक एवं ग्राहक का उदाहरण लेते है, एक बैंक मे बहुत से ग्राहक होते है, तथा एक ग्राहक बहुत से बैंक मे खाते हो सकते है। यद्यपी इस प्रकार के रिलेशनशिप के टेबल संभालना मुश्किल होता है, लेकिन एमएस एक्सेस यह कार्य किया जा सकता हैं। इस प्रकार के रिलेशनशिप का उपयोग कम ही किया जाता है, या बहुत जटिल डाटाबेस मे किया जाता है।

Relationship बनाना



दो या अधिक टेबल के बीच मे रिलेशन बनाने के लिए निम्न पदो का प्रयोग करें।

मुख्य टूल बार मे Relationship बटन को क्लिक करे। relationship कि विंडो खुल जाती है।



इसमे उस डाटाबेस मे बनाये गये, सभी टेबल दर्शाये जाते है। सभी टेबल की फील्ड की संरचना दर्शाई जाती है। जिन फील्ड के बीच मे सम्बन्ध बनाना है, उस फील्ड को वांछित टेबल मे से क्लिक कर दूसरे टेबल पर इच्छित फील्ड पर drag करे।

आपको edit relationship डायलॉग बॉक्स दिखाई देगा। इसमे ऊपर की ओर टेबल के नाम दर्शाये जाते है, तथा उसके नीचे फील्ड के नाम दर्शाये जाते है। आप फील्ड के नाम कार्य के अनुसार सेट कर सकते है। इस रिलेशनशिप को बनाना है, तब " Create " बटन को क्लिक करे। create new बटन का उपयोग नई रिलेशनशिप बनाने के लिए होता है, इसमे विकल्प से अन्य टेबल की फील्ड का उपयोग किया जा सकता है।

join relationship का डायलॉग बॉक्स दिखाई देगा, इसमे तीन विकल्प है

1. पहले विकल्प को चुनने से सिर्फ वो ही रिकार्ड रिलेशनशिप मे आते है, जो दोनों टेबल मे एक समान है
2. पहले टेबल के सभी रिकार्ड, तथा दूसरे टेबल के वो ही रिकार्ड जो सम्बन्धों को पुरा करते है, उन्हे ही प्राप्त किया जा सकता है।
3. दूसरे टेबल के सभी रिकार्ड, तथा पहले टेबल के वो ही रिकार्ड जो सम्बन्धों को पुरा करते है, उन्हे ही प्राप्त किया जा सकता है।

टेबल को Index करना

डाटा बेस फाइल का इंडेक्स यह किताब मे दिये गये इंडेक्स के समान ही कार्य करता है। यदि हमे किसी एक अध्याय के बारे पढना हो तो हम प्रथम किताब के इंडेक्स से उस अध्याय तथा उसके पेज नंबर देखते है, फिर उस पेज नंबर पर जाकर उस अध्याय कि जानकारी लेते है। यदि हम एक ग्रंथालय का उदाहरण लेते है, जिसमे एक लेखक की सभी किताबो के कार्ड एक जगह रखे होते है। यदि इसमें उस लेखक की कोई किताब देखना है तो हम उस कार्ड मे से उसे ढुँढतें है। उन कार्डो के खोजने के लिए वह कार्ड वर्णमाल के अनुसार क्रम मे रखे होते है। डाटा बेस प्रणाली भी यही काम करती है। उदाहरण के लिए हमे एक अकांउट नंबर के खाते का विवरण देखना है, तब डाटा बेस इंडेक्स को देख कर यह निश्चित करते है, कि कौनसा डिस्क का ब्लॉक उस रिकार्ड का प्रतिनिधित्व करता है। तथा उस ब्लॉक को डिस्क पर से लेकर आती है।

बहुत बडे डाटाबेस में लाखों, करोडो रिकार्ड रहते है, उस मे रिकार्डो की क्रमवार सूची रखना बहुत मुश्किल काम होता है। इसके इंडेक्स भी बहुत बडे होते है। फिर भी इंडेक्स के द्वारा रिकार्डो को रखने से वांछित रिकार्ड कम समय मे खोजे जा सकते है।

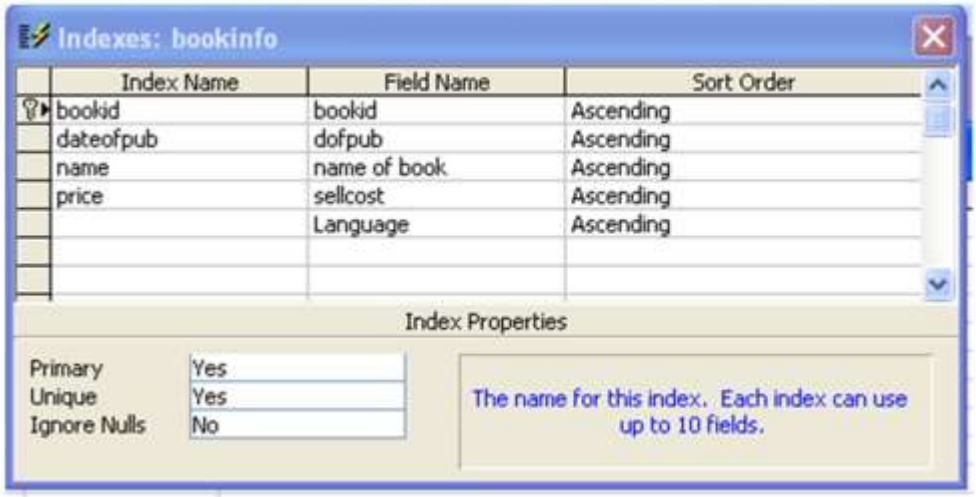
साधारणतः जिस क्रम मे कम्प्यूटर ऑपरेटर को डाटा मिलता है, उसी क्रम मे वह कम्प्यूटर मे एन्ट्री करता है। डाटा प्राय अव्यवस्थित क्रम मे टेबल मे डाला जाता है। उदाहरण के लिए किसी स्कुल मे छात्र के एडमीशन की एन्ट्री दिन के अंत मे या सप्ताह के अंत मे की जाती है। टेबल मे नया डाटा अंतिम रिकार्ड के बाद मे insert होता है। टेबल मे जैसे नया डाटा डाला जाता है, वैसे हि अनावश्यक डाटा हटाया भी जाता है। इन सभी प्रक्रिया मे डाटा टेबल मे अव्यस्थित हो जाता है। इंडेक्स के माध्यम से डाटा एक निश्चित क्रम मे लाया जाता है। यदि टेबल मे कम फील्ड है, तथा रिकार्ड भी कम है, तब बिना index किये टेबल से काम चलाया जा सकता है। लेकिन जहाँ टेबल मे अधिक फील्ड , बहुत रिकार्ड है, एवं टेबल दूसरे टेबल से जुडा है वहॉ टेबल को इंडेक्स करना आवश्यक हो जाता है। इंडेक्स को एक फील्ड या एक से अधिक फील्ड मे लगाया जा सकता है।

इंडेक्स का उपयोग क्यूरी, फिल्टर, एवं रिपोर्ट मे होता है। यदि बडी डाटाबेस फाइल है, एव उसमे index नही किया, ऐसी स्थिती मे एक्सेस किसी एक प्रकार के रिकार्ड को खोजने के लिए सभी रिकाड्ड को खंगालना पडेगा। लेकिन यदि फाइल इंडेक्स कि है, तब वह निश्चित रिकार्ड का समूह सीधे एक जगह से प्राप्त कर

आपको दर्शा सकता है। उदाहरण के लिए टेबल मे city फील्ड है, उसमे ग्राहक के शहर के बार मे जानकारी डाली है। आपको "bhopal" के ग्राहकों की सूचि प्राप्त करना है। यदि टेबल index नही है, तब पहले से अंत तक सभी रिकार्ड देखेगा एवं उनमे से जिन रिकार्ड के city फील्ड मे bhopal है, उन्हे दर्शाएगा। यदि टेबल index किया है, तब वह city फील्ड मे जहाँ से bhopal मान मिलता है, वहा से जहाँ खत्म होता है, वहा तक ही खोज कर रिकार्ड को दर्शाएगा।

यद्यपी वांछित रिकार्ड खोजने की गति छोटे डाटाबेस फाइल मे समझ मे नही आती है, लेकिन जहाँ बहुत अधिक रिकार्ड है, तब पर index किये टेबल और बैगेर index टेबल मे डाटा प्राप्त गति मे अंतर पता चलता है।

एक्सेस मे आप एक से अधिक फील्ड पर इंडेक्स लगा सकते है, इस प्रकार के इंडेक्स को compound index कहा जाता है। किसी टेबल के फील्ड पर इंडेक्स लगाने के लिए निम्न पदो का प्रयोग करें



- टेबल को design view मे खोले
- View मेन्यू को क्लिक करे।
- Index विकल्प को क्लिक करे। आपको इंडेक्स का डायलॉग बॉक्स दिखाई देगा। इस डायलॉग बॉक्स मे तीन कॉलम होते है।

Index name :- एक्सेस मे प्रत्येक इंडेक्स को एक नाम दिया जाता है। इस नाम का उपयोग क्यरी, रिपोर्ट मे किया जाता है। इंडेक्स का नाम उसके काम के अनुसार होना चाहीए।

field name :- जिस फील्ड के रिकार्ड को क्रमबद्ध करना है, उस फील्ड का नाम डाले। आप एक इंडेक्स मे एक से अधिक फील्ड का नाम डाल सकते है। लेकिन किसी भी इंडेक्स मे अधिकतम 10 फील्ड हि डाल सकते है।

sort order :- इस विकल्प मे सूची कैसे क्रम मे चाहिए वह सेट कर सकते है। इसमे ascending और descending दो विकल्प होते है।

नीचे के ओर तीन विकल्प होते है

1. Primary :- यदि इस विकल्प को "yes" सेट करते है, तब एक्सेस उस फील्ड को प्रायमरी की मान कर कार्य करता है। लेकिन प्रायमरी की के रिकार्ड मे कोई डाटा दोहराया नही जाता है,। तथा कोई रिकार्ड मे null value नही रहती है, अर्थात प्रत्येक रिकार्ड मे कुछ डाटा रहता है। इसलिए यदि किसी फील्ड के इंडेक्स यह मान लागू करते समय इन बातो का ध्यान रखना आवश्यक है। इंडेक्स बनाते समय इस विकल्प को default मान "No" रहता है।
2. Unique :- यदि इस विकल्प को "yes" सेट करते है, तब टेबल मे उस फील्ड मे सभी रिकार्ड अलग–अलग होने चाहिए। यदि दो या अधिक फील्ड को मिल कर इंडेक्स बनाया है, तब उन सभी फील्ड के रिकार्ड का combination यह अन्य रिकार्ड से अलग होना चाहिए। इस विकल्प को साधारणतः नंबर फील्ड आदि मे प्रयोग किया जाता है।
3. Ignore null :- यदि इस विकल्प को "yes" सेट किया है, यदि फील्ड मे कोई null value है, अर्थात रिकार्ड मे कुछ भी डाटा नही डाला गया है, उस रिकार्ड को इंडेक्स मे नही जोडा जाता है। अर्थात जिन रिकार्ड मे कुछ मान डाला गया है, उनके मान के अनुसार डाटा क्रमबद्ध किया जाता है। यदि इस विकल्प को "no" करते है, तब null value के रिकार्ड को इंडेक्स मे जोडा जाता हैं।

➲ उपरोक्त सभी विकल्प पूर्ण करने के बाद इस डायलॉग बॉक्स को बंद करे।

Filters

फिल्टर का उपयोग टेबल मे से इच्छित जानकारी को प्राप्त करने के लिए होता है। उदाहरण के लिए टेबल मे छात्रों को डाटा डाला है, हमे सिर्फ "DCA" कोर्स के ही छात्रों का डाटा चाहिए, उस स्थिती मे हम फिल्टर का उपयोग कर सकते है। किसी भी डाटाबेस प्रोग्राम फिल्टर का उपयोग बहुत महत्वपूर्ण है, इसके सहायता से डाटा तेजी से प्राप्त किया जाता है।

एक्सेस मे डाटा फिल्टर करने की सुविधा है। इसमे चार प्रकार से डाटा को फिल्टर किया जा सकता है।

Filter by form

जब प्रयोगकर्ता इच्छित जानकारी प्राप्त करने के लिए डाटाशीट मे आवश्यक फिल्ड को सिलेक्ट कर, इच्छित कंडीशन डालता है। उदाहरण के लिए हमे "DCA" के छात्रो को फिल्टर करना है, तब "Course" फिल्ड को क्लिक कर, उसमे "DCA" मान सेट करते है। इसे लागू करने के लिए टुल बार मे "Filter by form" आयकॉन को क्लिक करे।

Filter by selection

इस प्रकार के filter मे जो मान के आधार पर डाटा देखना है, उसे रिकार्ड पर पाईंटर रख, टुलबार filter by selection आयकॉन को क्लिक करते है। उदाहरण के लिए हमे, "DCA" कोर्स के छात्रों का डाटा देखना है, तब जिस रिकार्ड मे वह "dca" है, उस पर पाईंटर रख इस विकल्प को करें। उस कोर्स के छात्रों की सूची दर्शाई जाती है।

Queries

किसी भी डाटाबेस फाइल मे उस संरचना के अनुकूल डाटा डाला जाता है। किसी भी व्यावसायिक कार्य मे प्रयोग होने वाले डाटाबेस फाइल मे बहुत बडा डाटा संग्रहित किया जाता है। डाटाबेस फाइल उन डाटा को क्रमवार तरीके से संग्रहित करते जाती है। डाटाबेस फाइल का मुख्य उद्देश्य वांछित डाटा कम से कम समय मे प्रयोगकर्ता को दर्शाना है। जो डाटा हम टेबल या डाटाबेस फाइल मे संग्रहित करते है, उसकी आवश्यकता हमे दूसरे दिन, एक माह बाद या कुछ वर्षो बाद भी हो सकती है। सिर्फ टेबल के प्रयोग से यह कार्य मुश्किल हो सकता है। लेकिन इच्छित डाटा प्राप्त करने के लिए क्यूरी (query) बहुत अच्छा विकल्प है। जहाँ पर एक से अधिक टेबल का उपयोग हो रहा है, वहाँ क्यूरी का महत्व और अधिक बढ जाता है। क्यूरी के सहायता से प्रयोगकर्ता इच्छित डाटा को तेजी से प्राप्त कर सकता है। एक्सेसे मे क्यूरी से प्राप्त डाटा को Form, report आदि मे भी प्रयोग कर सकते है। दूसरे शब्दो मे "क्यूरी डाटा को जानकारी मे परीवर्तित करती है।"

जो डाटा टेबल मे संग्रहित किया जाता है, वह एक्सेस मे दिये गये विकल्पों के अनुसार होता है। टेबल की संरचना, एक्सेस से अनुकुल होती है, ना ही प्रयोगकर्ता के अनुकुल। डाटाबेस डिजाइन करते समय एक्सेस की क्षमताओं का ध्यान रखा जाता है। जहाँ एक से अधिक टेबल बनाये जाते है, वहाँ एक वस्तु या कार्य की जानकारी अलग–अलग टेबल मे होती है। यदि प्रयोगकर्ता को इच्छित जानकारी प्राप्त करना है, तब उसे प्रत्येक टेबल मे से

वांछित डाटा प्राप्त करना पडेगा। इस कार्य मे अधिक समय लग सकता है। इसलिए क्यूरी का उपयोग किया जाता है, जिसमे इच्छित जानकारी पहले से ही किसी जगह संग्रहित होती है। यह जानकारी विभिन्न टेबल से प्राप्त कि जाती है।

एक्सेस मे क्यूरी भी टेबल, फॉर्म या रिपोर्ट के समान ही डाटाबेस फाइल का एक ऑब्जेक्ट है। आप डाटाबेस फाइल मे एक से अधिक क्यूरी बना सकते है। जब आप क्यूरी बनाते है, वह उपलब्ध टेबल के फील्ड पर ही आधारित होती है, तथा क्यूरी मे प्राप्त उत्तर उन टेबल मे उपलब्ध रिकार्ड को दर्शाती है।

क्यूरी का उपयोग निम्न कामो मे होता है।

1. बडे डाटाबेस मे जहाँ बहुतसे टेबल का प्रयोग हो रहा है, वहॉ पर इच्छित डाटा मे बदलाव करना
2. बडे डाटाबेस मे अनावश्यक डाटा को हटाना
3. एक समान बडे डाटा टेबल मे जोडना
4. डाटाबेस फाइल मे संग्रहित डाटा के आधार पर इच्छित गणना करना ।

जब कोई क्यूरी का उपयोग करता है, तब एक्सेस विभिन्न टेबल से रिकार्ड प्राप्त कर उपयोगकर्ता को देता है, यह रिकार्ड अलग टेबल से अलग फील्ड से लिये जाते है, इस प्रकार के रिकार्ड के समूह को "Recordset" कहा जाता है। एक्सेस recordset को अलग टेबल मे संग्रहित कर देता है। जो बदलाव मूल टेबल मे हो रहे है, वह सभी बदलाव क्यूरी मे होते है। तथा जो बदलाव क्यूरी मे किये जाते है, वह सभी बदलाव उनसे जुडे टेबल मे भी होते जाते है।

क्यूरी के प्रकार

एक्सेस मे निम्न प्रकार के क्यूरी का उपयोग होता है।

Select :- यह सबसे अधिक प्रयोग होने वाली क्यूरी है। इस प्रकार के क्यूरी मे एक से अधिक टेबल से डाटा प्राप्त किया जाता हैं, तथा उन प्राप्त डाटा से recordset बनाया जाता है। इस प्रकार के क्यूरी से प्राप्त डाटा का उपयोग रिपोर्ट आदि मे किया जाता है।

Total :- यह एक विशेष प्रकार की क्यूरी है, इस प्रकार के क्यूरी जो रिकार्ड प्राप्त हुए है, उनका जोड (sum), औसत (average) गणना (count) आदि संग्रहित कि जाती हैं।

Action :- इस प्रकार की क्यूरी नया टेबल बनाने या पुराने टेबल मे डाटा मे बदलाव करने की सुविधा देती है।

Crosstab :- इस प्रकार क्यूरी डाटा को टेबल के रूप मे दर्शाती है। डाटा को वर्कशीट के संरचना मे दर्शाया जाता है, जिसमे प्रत्येक रो एक रिकार्ड को दर्शाती है, तथा प्रत्येक कॉलम एक फील्ड को दर्शाता है।

Top (n):- इस प्रकार के क्यूरी मे जो रिकार्ड की मांग की है, उनका संपूर्ण रिकार्ड के साथ प्रतिशत दर्शाती है।

क्यूरी की क्षमताएं

क्यूरी मे कार्य के अनुसार डाटा दर्शाया जाता है, यह प्रयोगकर्ता पर निर्भर होता है, कि उसे कौनसे फील्ड एवं किस प्रकार के रिकार्ड देखना है। क्यूरी का उपयोग यह कार्य के अनुसार बदलते

रहता है, इस रिकार्ड की संख्या बदलते जाती है। क्यूरी मे मांग के अनुसार डाटा बदलते रहता है।

इच्छित डाटा के मांग के निम्न प्रक्रिया कि जाती है।

Choose Tables :- डाटा को एक टेबल या एक से अधिक टेबल से प्राप्त किया जा सकता है। यदि एक से अधिक टेबल से डाटा प्राप्त करना है, तब एक्सेस उन सभी टेबल से डाटा प्राप्त करता है।

Choose Field :- जो रिकार्ड चाहिए उनके फील्ड को चुना जाता है। उदाहरण के लिए आपको छात्रों का नाम, दाखीले की तारीख एवं बकाया फीस की जानकारी प्राप्त करना है, तब हमे उनसे संबंधित फील्ड को सिलेक्ट करना पडता है।

Choose record:- किसी भी क्यूरी का यह सबसे महत्वुपर्ण हिस्सा होता है, जिसमे किस प्रकार का डाटा प्राप्त करना है, वह बताया जाता है। उदाहरण के लिए हमे सिर्फ DCA कोर्स के छात्रों का डाटा देखना है।

Sort Record :- क्यूरी मे एक क्रम मे डाटा को दर्शाया जाता है, इसके लिए इच्छित फील्ड को sort किया जाता है।

Query बनाना

क्यूरी ये टेबल पर आधारित होती है, क्यूरी बनाने से पहले टेबल को बनाना आवश्यक है। नई क्यूरी बनाने के लिए

- डाटाबेस विंडो के pane मे queries ऑयकॉन क्लिक करे।

- बाये ओर दो आयकॉन दिखाई देते है "Create Query by using wizard" एवं "Create Query in design view"

Create Query by using wizard

क्यूरी को विजार्ड की सहायता से बनाने के लिए "Create query by using wizard" आयकॉन को क्लिक करें।

विजार्ड की पहली विंडो दिखाई देती है। जिसमे ऊपर की ओर "tables/Queries" लिस्ट बॉक्स मे से इच्छित टेबल सिलेक्ट करें। उपलब्ध सभी टेबल की सूची दर्शाई जाती है। उनमे इच्छित टेबलों मे से इच्छित फील्ड सिलेक्ट करे। पहले एक टेबल की फील्ड का चुनाव कर, ">" से क्यूरी मे जोडें फिर दूसरे टेबल के फील्ड इस तरह इच्छित फील्ड का चुनाव करें।

सभी फील्ड चुनने के बाद "Next" बटन को क्लिक करे।

विजार्ड की दूसरी विंडो दिखाई देती है। इस विंडो मे क्यूरी मे सभी रिकार्ड को रखना है, या सिर्फ उनकी "Summary" रखना है, उसे सेट करे।

- Next बटन को क्लिक करे।
- क्यूरी का नाम निश्चित करे। तथा finish बटन को क्लिक करे।

Create Query in design view

विजार्ड के द्वारा सरल क्यूरी बनाई जा सकती है, लेकिन अधिक बडी या जटिल क्यूरी बनाने के लिए इस विकल्प का उपयोग किया जाता है। इसके लिए

- insert मेन्यू को क्लिक करें, query विकल्प को क्लिक करें
- New query का डायलॉग बॉक्स दिखाई देगा। उसमे design view को क्लिक करे।
- Ok बटन को क्लिक करे।

या

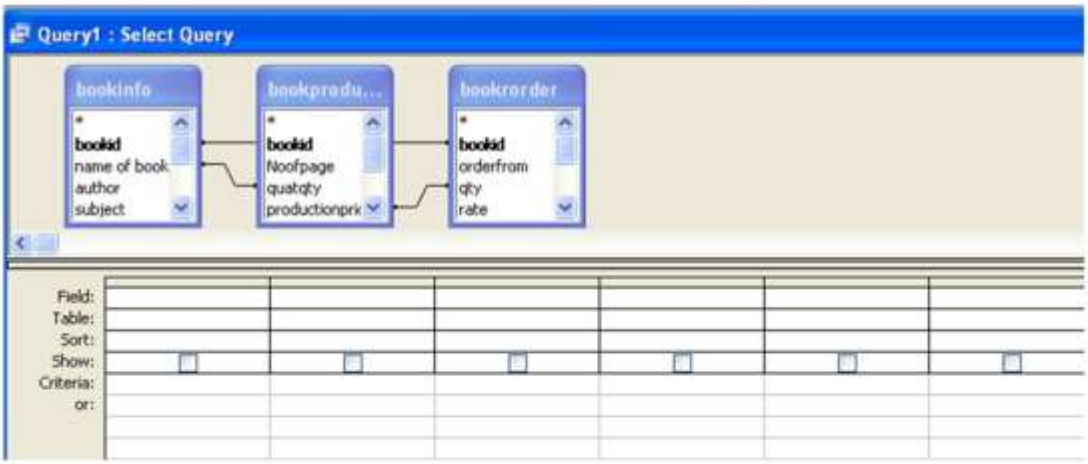
- मुख्य विंडो मे "Create query in design view" को दो बार क्लिक करे।
- उसमे से "Table" टेब को क्लिक करे।
- वर्तमान डाटाबेस फाइल मे उपलब्ध टेबल की सूची दिखाई देती है।
- जिन टेबल से डाटा लेना है, उनके नाम को क्लिक करे।
- "Add" बटन को क्लिक करे।

➲ अब उस टेबल की संरचना ऊपर एक विंडो मे दिखाई देती है।

➲ इसी तरह से अन्य टेबलों को जोडते जाए।

➲ इच्छित टेबलों का चुनाव होने के बाद "Close" बटन को क्लिक करे।

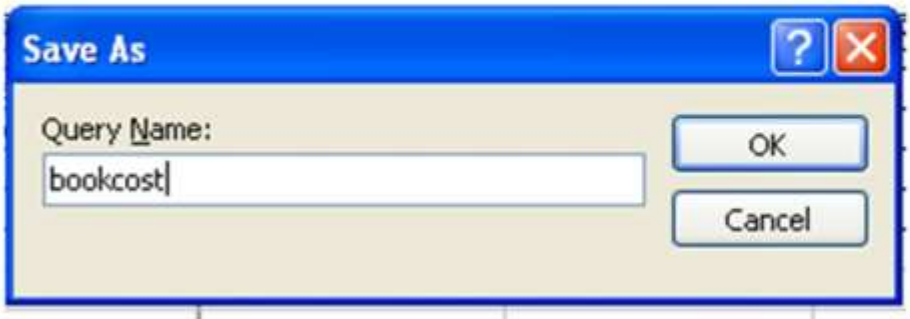
अब आपको निम्न प्रकार की संरचना दिखाई देगी।



इसमे ऊपर की ओर टेबल की संरचना, तथा उनके बीच का रिलेशन दर्शाया जाता है। किसी टेबल को हटाने के लिए उस टेबल पर Right click करें। एक shortcut मेन्यू दिखाई देता है, उस मे remove table विकल्प को क्लिक करें।

नीचे की ओर फील्ड, टेबल, sort, show आदि विकल्प दर्शाये जाते है। जिस रिकार्ड को क्यूरी मे शामिल करते है, उसके टेबल का नाम, फील्ड का नाम, उसे sort करना है या नही, तथा उसे प्रदर्शित करना है या नही यह नीचे के हिस्से मे सेट किया जाता है।

- जिस फील्ड को क्यूरी मे जोडना है, उस पर double click करें



- वह फील्ड पहले कॉलम मे दर्शाई जाती है।या फील्ड को drag कर इच्छित कॉलम मे रखे।
- सभी फील्ड को रखने एवं इच्छित विकल्प सेट करने के बाद क्यूरी को "Close" बटन से बंद करें
- एक्सेस क्यूरी को Save करने के बारे मे पूछता है। क्यूरी को इच्छित नाम दे। भविष्य मे इच्छित रिकार्ड प्राप्त करने के लिए क्यूरी का प्रयोग अधिक होता है, इसलिए क्यूरी का नाम सावधानीपुर्वक देना चाहीए। नाम देने के "Ok" बटन को क्लिक करे।

Cross tab query wizard

इस प्रकार के क्यूरी मे, क्यूरी के डाटा से प्राप्त परीणामों को summarize किया जाता है। इसमे एक फील्ड को बायें तरफ तथा अन्य फील्ड को ऊपर की ओर दर्शाया जाता है। उदाहरण के लिए हमे किसी एक कोर्स के छात्रों के नाम, दाखिले की तारीख, बकाया फीस की जानकारी लेना है, तब कोर्स का नाम बायें तरफ तथा अन्य जानकारीयाँ प्रत्येक रो मे दर्शाई जाती है।

डाटा विश्लेषण के लिए इस प्रकार की क्यूरी का उपयोग किया जाता है। इस प्रकार की क्यूरी बनाने के लिए

- insert मेन्यू को क्लिक करे।
- Query विकल्प को क्लिक करे।
- new Query का डायलॉग बॉक्स दिखाई देगा।
- उसमें से cross tab query विकल्प को क्लिक करे। तथा Ok बटन क्लिक करें।

विजार्ड का पहला डायलॉग बॉक्स दिखाई देगा। जिसमे ऊपर की ओर टेबल का नाम सिलेक्ट करे। next बटन क्लिक करे।

विजार्ड का दूसरा डायलॉग बॉक्स दिखाई देगा। जिसमे दायें ओर फंक्शन को सेट किया जा सकता है। उस टेबल की इच्छित फील्ड को सिलेक्ट करे। जो फील्ड सिलेक्ट करते है, उसे दायें

ओर की कॉलम मे दर्शाया जाता है। दायें ओर कॉलम मे अधिकतम तीन फील्ड का प्रयोग किया जा सकता है।

- Next बटन क्लिक करे।
- विजार्ड की अगली विंडो दर्शाई जाती है। जिसमे रो की फील्ड को सिलेक्ट कर सकते है। इसमे इच्छित फील्ड को सिलेक्ट करे। next बटन को क्लिक करें।
- विजार्ड की अगली विंडो दिखाई देगी जिसमे कौन से फील्ड पर गणना करना है, तथा कौन सी गणना करना है, वह सिलेक्ट करें। निम्न उदाहरण मे हमने
- name को बायें कॉलम को रखा है, बैलेंस को ऊपरी रो मे रखा है, तथा fee का औसत मुख्य क्षेत्र मे रखा है।
- next बटन को क्लिक करें।
- विजार्ड की अंतिम विंडो दिखाई देती है। जिसमे क्यूरी का नाम सेट करें।
- finish बटन को क्लिक करे।

क्यूरी के लिए SQL स्टेटमेंट

join any two table :- इस दो टेबल के डाटा को जोडने के लिए "Join" इस स्टेटमेंट का उपयोग किया जाता है। "join" में यदि संपूर्ण टेबल एक साथ जोडने के लिए "full outer join" इस का प्रयोग किया जाता है। उदाहरण के लिए हम tcustomer id तथा order id यह दो टेबल के स्टेटमेंट मे दिये फिल्ड जुड जाते है।

```
SELECT Customers.CustomerName,
Orders.OrderID
FROM Customers
FULL OUTER JOIN Orders ON Customers.Cust
omerID=Orders.CustomerID
ORDER BY Customers.CustomerName;
```

Display all fields of Table : इस कार्य के लिए "Select" स्टेटमेंट का उपयोग किया जाता हैं। जिस फिल्ड को सिलेक्ट करना है, उस फिल्ड का नाम टाइप करे। इसका सिन्टेक्स निम्न प्रकार से है

```
select col1, col...
from table
```


लिए यदि टेबल की सभी फिल्ड सिलेक्ट करना है, तब निम्न सिन्टेक्स का प्रयोग किया जाता है।

```
Select * from table
```

लिए उदाहरण

```
SELECT * FROM Customers;
```

find duplicate records :- तब डुप्लिकेट रिकार्ड खोजने के लिए "Count" तब फंक्शन का उपयोग किया जाता हैं, उदाहरण, निम्न स्टेटमेंट से डुप्लिकेट रिकार्ड को खोजा जा सकता है

```
SELECT username, email, COUNT(*)
FROM users
GROUP BY username, email
HAVING COUNT(*) > 1
```

MS Access

Unit 4

Working with forms, introduction to form types of basic forms, columnar

and tabular, Data sheet, main/ subform, add header and footer, add fields to form, add text to form, use label option button, checkbox, combobox, listbox, form wizard, create template

Unit -4

Forms

एक्सेस मे टेबल मे डाटा डालने के लिए बहुतसे विकल्प है, लेकिन फॉर्म विकल्प से डाटा सही तरीके से, आसानी से एवं तेजी से डाला जा सकता है। फॉर्म विकल्प मे ना सिर्फ डाटा डाला जा सकता है, अपितु रिकार्ड मे बदलाव कर सकते है, अनावश्यक रिकार्ड हटाये जा सकते है, तथा प्रिन्ट किये जा सकते है। फॉर्म के माध्यम से डाला गया डाटा, उससे संबंधित डाटाबेस मे जुड जाता है। फॉर्म मे कार्य करने मे हमे बहुतसी सुविधाएं प्राप्त होती हैं।

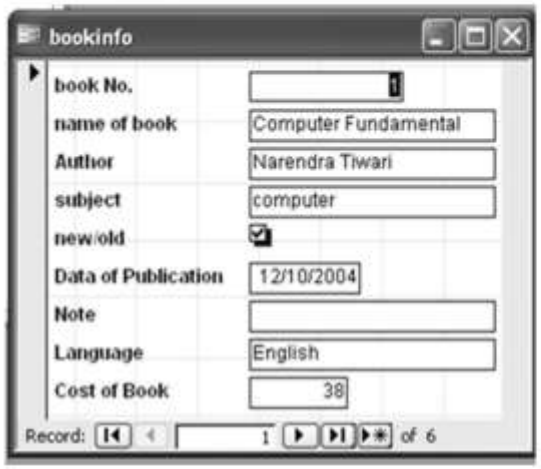
एक्सेस मे फॉर्म बनाने के लिए बहुतसे control दिये है, जिनकी सहायता से फॉर्म को कार्य के अनुसार सेट किया जा सकता हैं। एक्सेस मे प्रत्येक फॉर्म के लिए एक अलग विंडो मे खुलती है, तथा प्रत्येक फॉर्म एक अलग इकाई के रूप मे कार्य करती है। लेकिन भिन्न–भिन्न फॉर्म को एक दूसरे के साथ जोडा जा सकता है। एक्सेस मे फॉर्म बनाने के लिए

दाये ओर की मुख्य रिबन मे Form बटन को क्लिक करें

बायी ओर दो विकल्प दिखाये देते है। एक्सेस मे आप अपने फॉर्म को दो प्रकार से तैयार कर सकते है। जिसमे आपको डिजाइन व विजार्ड मिलता है। जिससे फॉर्म को तैयार कर सकते है। डिजाइन के द्वारा अपने फॉर्म को डिजाइन कर, उसमे सभी चीजों के बारे मे आप स्वयं चुनते है, जबकी विजार्ड के द्वारा आप को केवल उसके फील्ड चुनने होते है। उसकी सेटिंग पहले से ही सेट होती है।

Form के प्रकार

Columnar



इस प्रकार के लेआउट मे सभी फील्ड एक के नीचे दूसरी इस संरचना मे दर्शाई जाती है। इसमे प्रत्येक फील्ड के सामने उसमे रिकार्ड डालने की जगह दी होती है। इस प्रकार के फॉर्म एक या एक से अधिक स्क्रिन मे भी प्रदर्शित होते है। इसमे नीचे की ओर रिकार्ड पाईंटर बदलने के लिए कन्ट्रोल दर्शाया जाता है। जो फील्ड आपने सिलेक्ट कि है, वह सभी उसी क्रम मे दर्शाई जाती हैं इस प्रकार का लेआउट सरल डाटा एन्ट्री फॉर्म के लिए प्रयोग होता है। लेकिन जहाँ कोई दो फील्ड के मान की गणना करना है, यह किसी कंडीशन के आधार पर एन्ट्री करना है, वहॉ इस प्रकार की संरचना का प्रयोग कम किया जाता है। इस प्रकार के फॉर्म मे विकल्प कंन्ट्रोल का प्रयोग करते हुए एक professional फॉर्म डिजाइन किया जा सकता है। इस प्रकार के फॉर्म लेआउट मे प्रयोगकर्ता को डाटा डालने मे आसानी होती है।

Tabular

rollno	Name	course	Durtion	Fess	Paid Amt	Balance	Date of Birth
1	Anand Rahate	DCA	6	12000	10000	2000	
2	Vishal Lahare	PGDCA	12	15000	12000	3000	
3	Anita Varma	DCA	6	12000	5000	7000	
4	Shaila Rane	DCA	6	12000	7000	5000	
5	Anaji Gupta	DCA	6	12000	3000	9000	
6	Ayra Kumar	PGDCA	12	15000	5000	10000	
(Number)				0	0	0	

इस प्रकार की संरचना मे फील्ड की सूची आडे दिशा मे दर्शाई जाती है, तथा रिकार्ड को उन फील्ड के नीचे टाइप किया जाता सकता है। इस प्रकार की संरचना जहाँ एक समान डाटा की एन्ट्री करना है, उस स्थिती मे किया जाता है। जैसे छात्रों के मार्क की एन्ट्री करना है, या कर्मचारीयों के वेतन की एन्ट्री करना आदि है। जहाँ गणना करना है, वहा इस प्रकार के संरचना का प्रयोग किया जाता है। इस प्रकार की संरचना एक स्क्रिन पर एक से अधिक रिकार्ड देखे जा सकते है, तथा उनमे एन्ट्री की जाती है। इस संरचना मे आसानी से एक रिकार्ड से दूसरे रिकार्ड पर कर्सर ले जाया जा सकता है, तथा इच्छित बदलाव किये जा सकते है। यद्यपी यदि एक फील्ड मे बहुत अधिक रिकार्ड है, तब इस संरचना से कुछ मुश्किलें आ सकती है। इस प्रकार का फॉर्म सूचिबद्ध डाटा डालने एवं प्रिन्ट करने के लिए अधिक प्रयोग होता है।

Datasheet

bookid	name of book	author	subject	newVold	dofpub	remark	Language	sellcost
1	Computer Fundamental	Narendra Tiwari	computer	☑	12/10/2004		English	38
2	ms-office	narendra tiwari	computer	☐	12/10/2004		English	58
3	Desk top publication	Narendra Tiwari	computer	☑	12/10/2004		english	60
4	tally	Narendra Tiwari	computer	☑	18/12/2004		English	44
5	Marketing Management ti	Sandhya Agrawak	Managament	☐	22/12/205		English	0
6	dcaa	narendra tiwari	computerd	☑			hindi	145
(AutoNumber)				☐				0

इस प्रकार के संरचना मे रो और कॉलम होते है। जिसमे प्रत्येक रो एक रिकार्ड को दर्शाता है, तथा प्रत्येक कॉलम एक फील्ड को दर्शाता हैं। जहाँ अधिक डाटा डालना है, उस स्थिती मे इस संरचना का

प्रयोग किया जाता है। लेकिन महत्वपूर्ण या जटिल प्रकार के डाटा के लिए इस संरचना का प्रयोग नही किया जाता है। इस प्रकार की संरचना टेबल मे डाटा एन्ट्री मे भी प्रयोग की जाती है। साधारणतः फॉर्म को इस संरचना का प्रयोग नही किया जाता है। इस प्रकार के फॉर्म को प्रिन्ट करना कुछ मुश्किल हो सकता है।

Justified

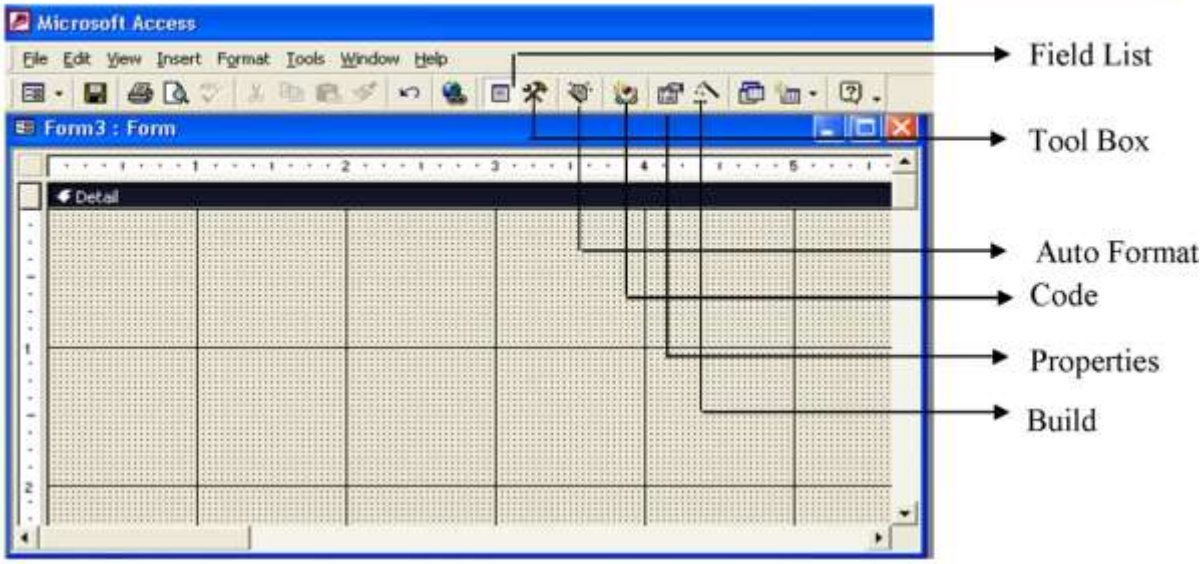
इस संरचना मे जहाँ एक फील्ड की **width** खत्म होती है, उसके बाद दूसरी फील्ड चालू होती है। तथा फॉर्म की चौडाई खत्म होने पर फील्ड नीचे की ओर दर्शाई जाती है। यद्यपी फॉर्म डिजाइन की यह एक आदर्श स्थिती है। बडे फॉर्म पर जहाँ अधिक फील्ड है, उस स्थिती मे इस संरचना का प्रयोग किया जाता है।

Create form in design view

फॉर्म को कार्य के अनुसार डिजाइन करने के लिए निम्न प्रकार से फॉर्म डिजाइन विकल्प को चुना जाता है।

➲ मुख्य विंडो मे "Create form in design view" ऑयकॉन को **double click** करे। या

➲ **Insert** मेन को क्लिक करे।



- Form विकल्प को क्लिक करे।
- New form का डायलॉग बॉक्स दिखाई देता है।
- उसमे ऊपर की ओर फॉर्म का प्रकार सेट करें
- नीचे की ओर टेबल या क्यूरी का नाम सिलेक्ट करे।

इस मे हम अपनी आवश्यक्ताओं के अनुसार फील्ड का चुनाव कर सकते है, तथा विभिन्न control का उपयोग कर सकते है। फॉर्म डिजाइन करने के लिए "create form in design view" आयकान को क्लिक करे। एक नई विंडो खुल जाती है, जिसमे एक कन्ट्रोल बाक्स एवं खाली चौकोर स्क्रिन दिखाई देती है। तथा नीचे की और टेबल की विंडो दिखाई देती है, जिसमे उस टेबल की फील्ड सूची होती है।

फॉर्म डिजाइन करते समय सिर्फ फील्ड की जानकारी डाटाबेस फाइल से ली जाती है, तथा अन्य विकल्प control box से लिये जाते है।

Control के प्रकार

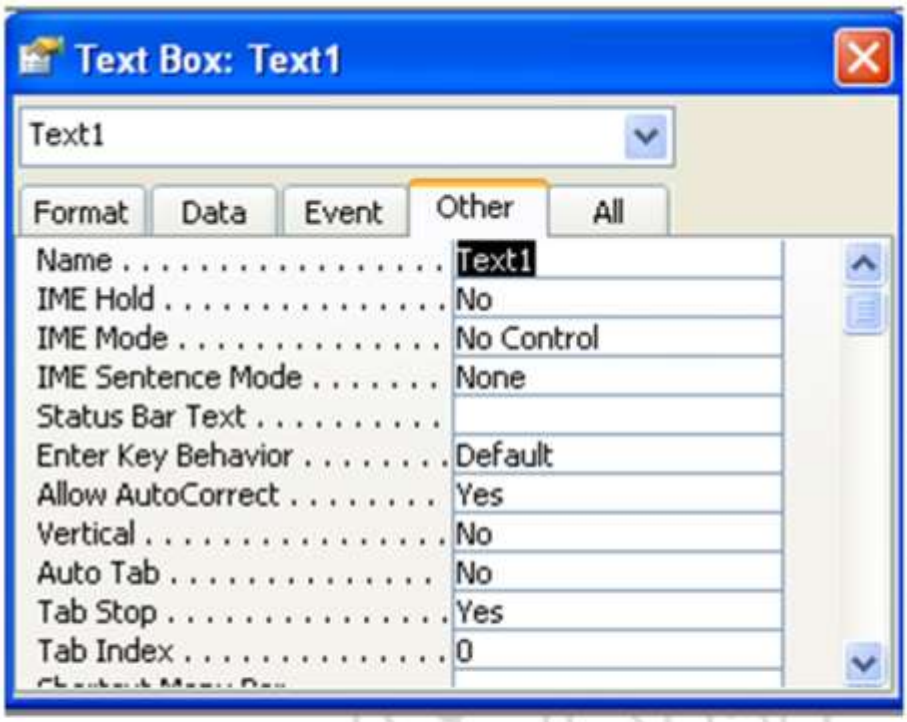
Bound control :- इस प्रकार के **control** यह डाटाबेस फाइल से जुडे होते है, तथा इसमे डाला गया डाटा इच्छित रिकार्ड मे बदलाव करता है। जब आप इस प्रकार के **control** मे कोई मान डालते है, तब एक्सेस उसे चालू डाटाबेस के फील्ड को **update** कर देता है। अधिकांश **control** इसी प्रकार के होते है।

Unbound control :- इस प्रकार के **control** यह डाटाबेस फाइल से नही जुडे होते है। इसमे जो मान डालते है, उस से डाटाबेस फाइल मे कोई बदलाव नही होता है। उदाहरण के लिये जो डाटा **label** कन्ट्रोल मे डाला है, उसका डाटाबेस फाइल से कोई सम्बन्ध नही होता है, यह सिर्फ जानकारी के लिए स्क्रिन पर दर्शाने के लिए प्रयोग करते है। एक्सेस मे **label, rectangular** आदि कंन्ट्रोल इस प्रकार मे आते है। इस प्रकार के कंन्ट्रोल फॉर्म मे हि संग्रहित रहते है।

Calculate Control :- इस प्रकार के कंन्ट्रोल मे कोई गणीतिय या तार्किक सूत्र **(expression)** दिये होते है। इस प्रकार के कंन्ट्रोल भी **unbound** प्रकार के होते है, क्योकि इसके डाटा से फील्ड मे कोई बदलाव नही होता हैं। यद्यपी कुछ गणनाओं मे डाटाबेस फाइल के रिकार्ड का उपयोग किया जा सकता है। उदाहरण **balanc=total_fee+Paid_fee** इस सुत्र से वह उत्तर दर्शाता है, लेकिन उसे **balanc** फील्ड को **update** नही करता है।

कंन्ट्रोल को फॉर्म पर डालना

फॉर्म विंडो खुलने के बाद



➲इच्छित कंन्ट्रोल को क्लिक करे।

➲उसे उठा कर (drag) फॉर्म पर इच्छित जगह रखे। जो भी कंन्ट्रोल आप फॉर्म पर रखते है, वह unbound प्रकार के होते है, उसे आपको डाटाबेस फाइल से जोडना पडता है। किसी भी कंन्ट्रोल की प्रापर्टी निश्चित करने के लिए ऊपर की ओर "properties" बटन को क्लिक करे। आपको उस कंन्ट्रोल का प्रापर्टी डायलॉग बॉक्स दिखाई देगा। इसमे ऊपर की ओर चार टैब होती है। तथा बायं ओर प्रापर्टी का नाम दर्शाया जाता है, तथा दाये ओर आप उस प्रापर्टी का मान निश्चित कर सकते है। ऊपर टैब के प्रयोग से प्रापर्टी को वर्गीकृत किया गया है।

Format :- इस टेब मे डाटा को या कंन्ट्रोल को किस तरह से प्रदर्शित करना है, उसकी प्रापर्टी की सूची रहती है।

Data :- कंन्ट्रोल मे किस प्रकार के डाटा का प्रयोग करना है, आदि की प्रापर्टी की सूची दर्शाइ जाती है।

Event :- किसी कंन्ट्रोल पर क्या प्रक्रिया करना है, या माऊस क्लिक होने पर क्या कार्य होना चाहिए आदि की प्रापर्टी की सूची इस श्रेणी मे दर्शाई जाती है।

other :- उपरोक्स सभी प्रकार से अगल यदि कोई प्रापर्टी है, तब उसे इस टैब के अतर्गत दर्शाया जाता है।

All:- कंन्ट्रोल की सभी प्रापर्टी की सूची दर्शाई जाती है।

फॉर्म मे डाटाबेस फाइल के फील्ड को डालना

फॉर्म टेबल से जुडा होता है, फॉर्म मे इच्छित फील्ड को आसानी से जोडा जा सकता है। इच्छित फील्ड को फॉर्म मे जोडने के लिए, फील्ड सूची मे उस फील्ड को क्लिक कर, फॉर्म पर drag करे। उस वह फील्ड एक टेक्स बॉक्स के रूप मे फॉर्म पर आ जाती है। उसके सामने उस फील्ड का नाम दर्शाया जाता है। फील्ड के नाम को क्लिक करें, उसे बदला जा सकता है।

drag and drop पद्धती से फील्ड को टेक्स्ट बॉक्स के रूप मे डाला जाता है।

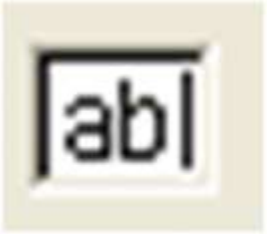
फॉर्म के कंन्ट्रोल

एक्सेस मे निम्न प्रकार के control है

1. Text box
2. Label
3. Command
4. Toggle button
5. List box
6. Option box
7. Check box

सभी प्रकार के control को सेट करने के लिए properties विकल्प दिया है।

Text box control



इस कंट्रोल मे प्रयोगकर्ता इच्छित टेक्स्ट इस बॉक्स मे टाइप कर सकता है। इस कंट्रोल से एक छोटा बॉक्स बनता है, जिसमे ना सिर्फ टेक्स्ट को टाइप किया जा सकता है, अपितु उसे कॉपी, डिलीट, पेस्ट आदि करने की भी सुविधा होती है। यदि बडा टेक्स्ट डालना है, तब उसे scroll विकल्प भी दिया जा सकता है। इस टेक्स्ट बॉक्स मे डाला गया टेक्स्ट यह साधारण टेक्स्ट, पासवर्ड, नंबर कुछ भी हो सकता है। यह एक लाइन या एक से अधिक लाइन का भी हो सकता है। लेकिन इसमे ग्राफिक नही डाल सकते है। इस कंट्रोल निम्न प्रापर्टी है।

i. **Name** :- एक्सेस मे प्रत्येक control को एक नाम दिया जाता है, इस विकल्प से उस कन्ट्रोल का नाम सेट किया जाता है। साधारणतः इस control को उस फील्ड का नाम दिया जाता है, लेकिन आप कार्य के अनुसार अलग नाम दे सकते है।

ii. **Decimal Places** :- यदि इसमे number प्रकार का डाटा डाला जाना है, तब उसमे कितनी decimal संख्या डाली जा सकती है, यह इस प्रापर्टी मे सेट किया जाता है। यदि हमे कोई रूपये के मूल्य को डालना है, तब इसे "2" रखते है।

iii. **display When** :- यदि किसी टेक्स्ट बॉक्स को सिर्फ प्रिन्ट मे दर्शाना है, या किसी टेक्स्ट बॉक्स को सिर्फ स्क्रिन पर दर्शाना है, प्रिन्ट नही करना है, तब इस प्रापर्टी का उपयोग किया जाता है। इसमे "Always" मान से टेक्स्ट बॉक्स के

घटक स्क्रिन एवं प्रिन्ट दोनों मे दर्शाये जाते है। यदि "screen only" मान सेट किया है, तब वह टेक्स्ट बॉक्स स्क्रिन पर दर्शाया जाता है, प्रिन्ट मे नही आता है।

iv. Alignment :- इस प्रापर्टी से टाइप किया गया टेक्स्ट, बॉक्स के अंदर किस प्रकार से अलाइन हो, यह निश्चित किया जाता है। इसके निम्न मान है

0 – left alignment 1-right alignment 2-center alignment

v. Special Effect :- टेक्स्ट बॉक्स किस प्रकार से दिखे यह इस प्रापर्टी से निश्चित किया जाता है।

vi. Fore colour :- टेक्स्ट बॉक्स के घटको के रंग को निश्चित करने के लिए इस प्रापर्टी का उपयोग किया जाता है।

vii. Default Value :- इस प्रापर्टी से टेक्स्ट बॉक्स मे पहले से हि एक मान सेट हो जाता है। यद्यपी प्रयोगकर्ता उस मान को बदल सकता है। जो मान किसी फील्ड मे बारंबार प्रयोग हो रहा है, उसे इस प्रापर्टी मे सेट किया जाता है। उदाहरण के लिए हमे duration फील्ड मे डाटा डालना है, तथा हमे पता है, कि आधिकांश कोर्स को duration "12 month" है, तब इस मान को default value प्रापर्टी मे सेट करते है।

viii. Validation Rule :- यह बहुत महत्वपूर्ण प्रापर्टी है, इस प्रापर्टी से टेक्स्ट मे अंवाछित डाटा डालने से रोका जा सकता है। टेक्स्ट बॉक्स मे जो इनपुट नही होने चाहिए वैसे

कंडिशन मान इस प्रापर्टी मे सेट किये जा सकते है। उदाहरण के लिए हमने छात्रों के मार्क का फॉर्म बनाया है, तथा इसमे प्रत्येक छात्र के मार्क की एन्ट्री होने वाली है। यदि गलती से 0 (शुन्य) से कम या 100 से अधिक मार्क डाला जाता है, तब वह एक गंभीर गलती हो सकती है। उपरोक्त उदाहरण के हम ">0 and <100" टाइप करते है, अर्थात उस टेक्स्ट बॉक्स मे कोई 0 से कम एवं 100 से अधिक मान डालता है, तब एक्सेस उस संग्रहित नही करता है, तथा एक चेतावनी संदेश दर्शाता है।

ix. **Validation Text** :- यदि प्रयोगकर्ता ने **validation rule** का उल्लघंन किया है, एवं गलत डाटा डाला है। तब उसे कौन सा संदेश दर्शाना है, यह इस प्रापर्टी मे सेट कर सकते है।

x. **Locked**:- इस प्रापर्टी से प्रयोगकर्ता टेक्स्ट बॉक्स के अंदर टेक्स्ट बदल सकता है या नही यह निश्चित किया जाता है। यदि इसे "**true**" रखते है, तब प्रयोगकर्ता उस टेक्स्ट बॉक्स के अंदर के घटकों को नही बदल सकता या उसके अंदर कोई नया टेक्स्ट नही डाल सकता है। यदि इसे **false** रखते है, तब उसमे टेक्स्ट डाला जा सकता है।

xi. **Scroll bar** :- यदि टेक्स्ट बॉक्स मे स्क्रोल बार लगाना है, तब इस प्रापर्टी का प्रयोग किया जाता है।

xii. **Password char** :- इस प्रापर्टी से टेक्स्ट बॉक्स मे डाला गया कैरेक्टर आपके दिए हुए कैरेक्टर मे परीवर्तित हो जाता है।

Option button

यह भी कंट्रोल चेक बॉक्स के समान है, लेकिन एक ग्रुप मे एक से ज्यादा विकल्प को चालू नही किया जा सकता है। अर्थात एक ग्रुप मे सिर्फ ही एक बटन चालू रहती है, और बाकी सभी बंद रहती है। साधारणतः इसे एक फ्रेम के अंदर रखा जाता है। value इस कि मुख्य प्रापर्टी है।

Image

इस कंट्रोल का प्रयोग फॉर्म पर विभिन्न आब्जेक्ट, बिटमैप डालने के लिए होता है।

आप फॉर्म पर design time एव run time दोनों समय इमेज डाल सकते है। design time मे पिक्चर के प्रापर्टी विंडो मे इमेज फाइल कि सेटींग कर सकते है। या rum time मे loadpicture() फंक्शन से इमेज डाल सकते है।

```
private sub form_load()
image1.ficture=loadpicture("C:\windows\arra.bmp")
```

यदि इसकी stretch का मान true रखा जाता है, तब प्रयोगकर्ता डाली गई इमेज का आकार बदल सकता है।

Form Control Properties

फॉर्म के विभिन्न परिमाण निश्चित करने के लिए, प्रापर्टी विंडो बहुत से विकल्प दिये। फॉर्म की प्रापर्टी सेट करने के लिए फॉर्म के खाली हिस्से मे क्लिक करे। फॉर्म प्रापर्टी के निम्न विकल्प है

Filter :- यदि फॉर्म चुने हुए रिकार्ड ही प्रदर्शित करना है, तब इस प्रापर्टी का उपयोग किया जाता है।

Data entry :- यदि इस प्रापर्टी को "yes" करते है, तब bound कंन्ट्रोल मे नया डाटा जा सकता है, यदि "no" रखते है, तब उस प्रकार के कंन्ट्रोल मे डाटा नही डाला जा सकता है।

Scroll bar :- फॉर्म मे scroll बार दर्शाया है, या नही दर्शाना है। तथा कौन से scroll बार को दर्शाना है, इस प्रकार के कार्य इस प्रापर्टी मे सेट किये जा सकता है। इसमे चार विकल्प है neither, Vertical only, horizontal only एवं both.

navigation button :- फॉर्म के नीचे की ओर रिकार्ड नेवीगेशन बटन रखना या अदृश्य करने के लिए इस प्रापर्टी का उपयोग होता है।

Auto size :- यदि इस प्रापर्टी मे ""yes" मान रखते है, तब एक्सेस फॉर्म के घटकों के अनुसार उस आकार स्वयं ही सेट कर लेता है।

Picture :- फॉर्म के background मे कौन सा चित्र डालना है, उसे इस प्रापर्टी मे सेट कर सकते है। इच्छित चित्र डालने के लिए उस फाइल को सिलेक्ट करे। ,

Menu Bar :- फॉर्म पर डिजाइन किये मेन्यू बार दर्शाने के लिए इस प्रापर्टी का उपयोग किया जाता है।

Movable :- यदि इस प्रापर्टी को "yes" सेट करते है तब प्रयोगकर्ता फॉर्म विंडो को स्क्रिन मे इच्छित जगह रख सकता है।

Sub-form

एक फॉर्म के अंदर दूसरे फॉर्म का प्रयोग किया जा सकता है। जो फॉर्म दूसरे फॉर्म के अंदर कार्य करता है, उसे sub-form कहते है। मुख्य फॉर्म को main form कहा जाता है, एक मुख्य फॉर्म के अंदर

एक या एक से अधिक फॉर्म हो सकते है। जब किसी टेबल मे **one to many relationship** का प्रयोग किया गया है, उस स्थिती मे **sub-form** अधिक कारगर होता है। उदाहरण के लिए आपने दो टेबल बनाये है, जिसमे एक टेबल मे उत्पाद की विभिन्न श्रेणीयॉ है, जैसे **TV, Fridge, Washing Machine** आदि उस टेबल को हम **"category"** नाम देते है, तथा दूसरे टेबल मे हमने प्रत्येक क्षेणी के वास्तविक उत्पाद को रखा है, जैसे **21inch lcd, 29inch led, 365 lit fridge** आदि तथा इस टेबल को **"product"** नाम दिया है।

इन दोनों टेबल मे **one to many relationship** है, अर्थात **TV** श्रेणी मे दूसरे टेबल मे विभिन्न उत्पाद जुडे है। ऐसे स्थिती मे हम दूसरे टेबल के लिए **sub form** बना सकते है। मुख्य फॉर्म एवं **sub form** एक दूसरे से जुडे होते है, तथा जो रिकार्ड पहले टेबल मे है, उससे संबंधित रिकार्ड दूसरे टेबल से दर्शाया जाता है। जब फॉर्म पर पहले टेबल के एक श्रेणी को दर्शाता है, तब **sub form** दूसरे टेबल मे उसी उत्पाद को दर्शाता है।

आप **sub-form** को **datasheet view, form view, Pivot table view** आदि मे डिजाइन कर सकते है। लेकिन यदि मुख्य फॉर्म **pivot table** या **pivot chart** मे डिजाइन किया है, तब उसमे **sub form** नही दिखाई देता है।

जब आप **one to many relationship** का प्रयोग कर मुख्य फॉर्म एवं उप फॉर्म बनाते है, तब इस **relationship** की **one side** यह मुख्य फॉर्म प्रयोग करता है, तथा **many side** यह उप फॉर्म प्रयोग करता है। जो भी उप फॉर्म आप मुख्य फॉर्म मे बनाते है, उनके बीच

रिलेशनशिप की व्याख्या करना आवश्यक है। दोनों टेबल मे प्रायमरी की एक समान होनी चाहिए।

आप मुख्य फॉर्म के अंदर एक से अधिक sub-form बना सकते है, तथा sub form के अंदर और भी sub form बना सकते है। लेकिन एक के अंदर दूसरे sub-form की संरचना सात sub-form से अधिक नही होनी चाहिए।

जब आप किसी मुख्य फॉर्म एव उप फॉर्म मे रिकार्ड डालते है, तब एक्सेस उप फॉर्म मे एन्ट्री करते समय मुख्य फॉर्म के टेबल के रिकार्ड को स्वयं ही सेव करता है।

Sub form बनाना

जब कोई मुख्य फॉर्म उसके उप फॉर्म से जुडा है, तब उसे टेबल को जाचँ लेना चाहिए। तथा दोनों टेबल के बीच relationship सेट कर लेना चाहिए। लेकिन जब मुख्य फॉर्म pivot table या pivot chart के रूप मे होता है, तब उसके अंदर उप फॉर्म नही बना सकते है। उपफार्म बनाने के लिए निम्न पदो का प्रयोग करें

- मुख्य विंडो मे form आयकॉन को क्लिक करे। फॉर्म का डायलॉग बॉक्स खुल जाता है।
- उसे "form wizard" पर दो बार क्लिक करें। विजार्ड मे पहले टेबल को सिलेक्ट करें, उसमे मुख्य फील्ड जो दूसरे टेबल से जुडी है, उसे भी insert करे। इसी डायलॉग बॉक्स मे दूसरे टेबल को भी सिलेक्ट करे। जिन फील्ड को जोडना है, उन्हे क्लिक कर insert करें।

यदि आपने फॉर्म बनाने के पहले दोनों टेबल के बीच मे relationship बना ली है, तब विजार्ड आपसे कौन से टेबल से डाटा

देखना है। विजार्ड डायलॉग बॉक्स मे "form with sub-forms" विकल्प को सिलेक्ट करे।

विजार्ड मे अन्य विकल्पों को सेट करें, तथा Finish बटन को क्लिक करे।

फॉर्म मे Header एवं footer जोडना

फॉर्म हेडर या फुटर मे वह जानकारी दर्शाई जाती है, जो प्रत्येक फॉर्म मे एक समान होती है। उदाहरण के लिए जो फॉर्म मे डाटा एन्ट्री कर रहे है, उसका शिर्षक, तारीख आदि। फॉर्म मे header या footer जोडने के लिए

- View मेन्यू क्लिक करे।
- Form header /Footer विकल्प को क्लिक रे।
- फॉर्म पर detail के साथ, ऊपर की ओर form header एवं नीचे के ओर form footer हिस्से दिखाई देते है।
- Header हिस्से मे टेक्स्ट डालने के लिए लेबल कंन्ट्रोल का प्रयोग किया जाता है। साधारणतः इन हिस्सों टेबल के रिकार्ड का प्रयोग नही किया जाता है।

प्रापर्टी बॉक्स मे हेडर एव फुटर की अलग–अलग प्रापर्टी निश्चित की जा सकती है। फॉर्म डिजाइन करते समय इन तीनों हिस्सो के बीच मे एक पैनल दिखाई देती है। उस पैनल को ऊपर या नीचे कर हेडर, डिटेल या फुटर का आकार कम या अधिक किया जाता है। यह पैनल फॉर्म मे डाटा डालते समय नही दिखता है।

Template

एमएस एक्सेस में निम्न विधि से Template का निर्माण किया जा सकता है

जिस कार्य के लिए टेम्प्लेट बनाना है, उसके अनुसार टेबल,फार्म आदि को डिजाइन करें। एमएस एक्सेस में आप अलग अलग ऑब्जेक्ट के लिए अलग अलग टेम्प्लेट बना सकते है। उदाहरण के लिए आप फॉर्म के विभिन्न टेम्प्लेट बना सकते है, तथा कार्य के अनुसार उन्हे प्रयोग कर सकते है।

जिस आब्जेक्ट का टेम्प्लेट बनाना है, उसे पूर्णतः डिजाइन करें, सभी प्रकार की फारमेटिंग, validation आदि सेट करें। सभी कार्य पूर्ण होने के बाद

- file मेनु को क्लिक करें।
- Save As विकल्प को क्लिक करें।
- Template विकल्प को क्लिक करें।
- उसमें "Save database as" Pane में "Save as" विकल्प को क्लिक करें।
- उसके बाद टेम्प्लेट की प्रापर्टि डालने कि विंडो दिखाई देती है, उसमे टेम्प्लेट से संबंधित जानकारी टाइप करें।
- Ok बटन को क्लिक करें।

MS Access

Unit 5

Working with report, introduction to reports, types of basic reports, single column, tabula report, group/ total, single table report, multi table report, preview report, print report, creating report and labels, wizard

Unit -5

रिपोर्ट

डाटाबेस मैनेजमेंट प्रणाली मे रिपोर्ट महत्वपूर्ण हिस्सा होता है। जो डाटा डाटाबेस फाइल मे डाला है, तथा प्रोग्राम से प्रोसेस किया है, उसे रिपोर्ट द्वारा प्रदर्शित किया जाता है। रिपोर्ट का मुख्य काम प्रयोगकर्ता के आवश्यक्ता के अनुसार स्क्रिन या प्रिन्ट पर आवश्यक डाटा, समझने योग्य प्रारूप मे दर्शाना है। रिपोर्ट मे एक टेबल या एक से अधिक टेबल का डाटा दर्शाया जा सकता है। रिपोर्ट डाटाबेस फील्ड मे संग्रहित रिकार्ड के अतिरीक्त अन्य जानकारीयॉ जैसे रिकार्ड का योग (total), रिकार्ड की गणना, कंपनी का नाम, रिपार्ट प्रिन्ट करने के तारीख एवं समय आदि भी दर्शाया जाता है। रिपोर्ट को साधारणतः ऐसे प्रारूप मे सेट किया जाता है, कि वह सभी प्रिन्टर पर भी प्रिन्ट कि जा सके। रिपोर्ट टेक्स्ट डाटा के अलावा ग्राफिक्स, ग्राफ आदि का भी प्रयोग किया जा सकता है, लेकिन रिपोर्ट मूलतः डाटाबेस फाइल पर ही आधारित होती है।

रिपोर्ट मे आप एक समान डाटा को एक साथ ग्रुप कर सकते है, श्रेणीयों के आधार पर वर्गीकृत कर सकते है। डाटा को आवश्यक क्रम मे सेट कर सकते है। रिपोर्ट मे आप टैक्स्ट, नंबर, डेट, मेमो आदि सभी प्रकार के फील्ड का प्रयोग कर सकते है।

एक्सेस मे रिपोर्ट बनाने एवं उसे फारमेट करने के लिए बहुत से विकल्प उपलब्ध है। आप एक रिपोर्ट मे इच्छित फील्ड का प्रयोग कर सकते है, इच्छित कंडीशन के आधार पर रिकार्ड को चुन सकते है। आप रिकार्ड का आकार, फारमेट, पेज पर स्थिती आदि निश्चित कर सकते है। एक्सेस मे एक से अधिक टेबल से डाटा प्राप्त कर रिपोर्ट बना सकते है। एक्सेस रिपोर्ट के कुछ पुर्वनिर्धारित फारमेट है, जिसके प्रयोग से बहुत कम समय मे रिपोर्ट की डिजाइन बनाई जा सकती है। रिपोर्ट मे विभिन्न expression, Sql statement का प्रयोग कर सकते है। एक्सेस मे रिपार्ट बनाने के लिए बहुत से control दिये है, जिनकी सहायता से, जटिल रिपोर्ट बनाई जा सकती है।

एक बार रिपोर्ट की डिजाइन बनाने के बाद, एक्सेस स्वंय ही जोडे गये टेबल से इच्छित रिकार्ड प्राप्त कर लेता है। जो बदलाव डाटा मे किये जाते है, उनके अनुसार ही रिपोर्ट दर्शाई जाती है।

रिपोर्ट बनाने की प्रक्रिया

रिपोर्ट मे इच्छित डाटा को समझने योग्य फारमेट मे बनाया जाता है। रिपोर्ट का उपयोग संस्था के अंदर एवं बाहरी व्यक्ति भी प्रयोग करते है। बहुत सी रिपोर्ट की कानुनी वैधता भी होती है, जैसे bill, invoice, notice आदि। इसलिये रिपोर्ट की डिजाइन सावधानी से की जानी चाहिए। रिपोर्ट की डिजाइन बनाने से पहले उसकी योजना बनाना आवश्यक है। रिपोर्ट की डिजाइन करते समय निम्न बिंदुंओ का ध्यान रखना आवश्यक है।

- रिपोर्ट कौन से कार्य के लिए प्रयोग होने वाली है।
- रिपोर्ट की कौन सी जानकारी महत्वपूर्ण एवं आवश्यक है।

- रिपोर्ट मे कौन सी जानकारी आवश्यक नही है।
- रिपोर्ट सिर्फ स्क्रिन पर दर्शाना है, या पेज पर प्रिन्ट करना है।
- रिपोर्ट किस प्रकार के प्रिन्टर पर प्रिन्ट होनी है।
- रिपोर्ट इंटरनेट मे प्रयोग होनी है, या नही होनी है।

उपरोक्त बिंदुओं को निश्चित करने के बाद, निम्न योजनाएं बनाई जाती है।

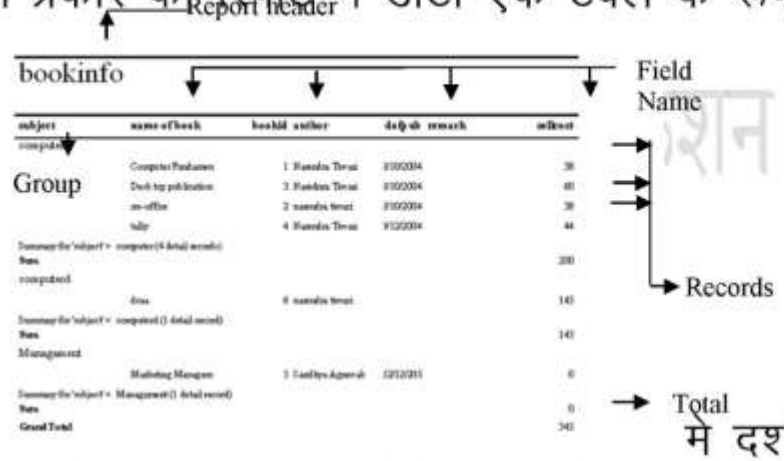
I. रिपोर्ट मे कौन–कौन से टेबल का प्रयोग करना है।
II. रिपोर्ट मे कौन–कौन सी फील्ड का उपयोग करना है।
III. यदि ऑब्जेक्ट रखना है, तब कौन सा ऑब्जेक्ट रखना है, तथा ऑब्जेक्ट को पेज पर कहॉ रखना है।
IV. कौन से expression का प्रयोग करना है।
V. कौन से फील्ड पर expression का उपयोग करना है।
VI. कौन सी जानकारियॉ महत्वपूर्ण है, तथा उसे किस तरह उभार कर दर्शाना है।
VII. रिपार्ट मे कौन से लेआउट का प्रयोग करना है। रिपोर्ट का लेआउट इस तरह से सेट किया जाता है, कि पढने वाले व्यक्ति को समझने मे आसानी हो। कोई भी डाटा स्क्रिन या पेज के बाहर ना जाये। सभी प्रकार के sum, count आदि उन अंकों के सीध मे हो। यदि रिपोर्ट पेज पर प्रिन्ट करना है, तब चारो ओर उचित मार्जिन छोडी जानी चाहिए। यदि एक से अधिक पेज का रिपोर्ट है, तब प्रत्येक पेज को उनके क्रम के अनुसार पेज नंबर आना चाहिए।

Report के प्रकार

वांछित डाटा स्क्रिन पर या प्रिन्ट पर अलग–अलग तरीके से दर्शाया जा सकता है। डाटा की स्क्रिन पर स्थिती तथा उनके क्रम के अनुसार रिपोर्ट को वर्गीकृत किया जा सकता है। किसी रिपोर्ट मे सिर्फ टेक्स्ट प्रकार का डाटा से काम हो सकता है, किसी रिपोर्ट डाटा के साथ ग्राफ या ऑब्जेक्ट की भी आवश्यकता हो सकती है। अलग–अलग स्थिती के अनुसार रिपोर्ट का लेआउट अलग–अलग होता है। एक्सेस मे रिपोर्ट के निम्न प्रकार के लेआउट है,

Tabular Report

इस प्रकार की रिपोर्ट मे डाटा एक टेबल के रूप

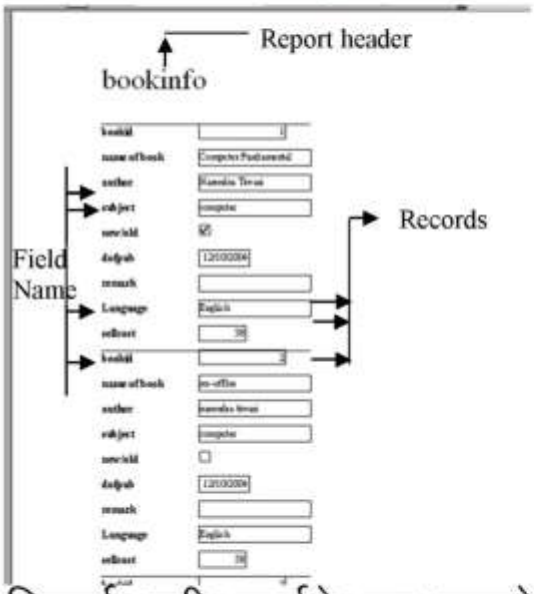


मे दर्शाया जाता है। यह सबसे अधिक प्रयोग होने वाला लेआउट है। इसमे प्रत्येक डाटा की हेडिंग सबसे ऊपरी रो मे दशाई जाती है, तथा एक डाटा एक लाइन मे दर्शाया जाता है। डाटा को यदि क्रमबद्ध(sort) किया है, तब क्रम के पहले डाटा ऊपर की ओर दर्शाये जाते है, तथा अंत के डाटा पेज के नीचे की ओर दर्शाये जाते है। यदि डाटा को ग्रुप किया है, तब प्रत्येक ग्रुप के अनुसार

डाटा एक के नीचे दूसरा इस प्रकार से दर्शाये जाते है। सभी प्रकार के योग या गणनायें नीचे की ओर वांछित कॉलम मे दर्शाई जाती है। इसमे एक प्रकार का डाटा एक के नीचे दूसरा इस प्रकार से आता है। यदि कोई रिपोर्ट एक पेज से अधिक की है, तब प्रत्येक पेज मे किसी कॉलम की sub-total दर्शाई जाती है, तथा अंत मे संपूर्ण योग दर्शाई जाती है। कुछ रिपोर्ट मे एक पेज मे एक से अधिक डाटा समूह का प्रयोग एक से अधिक कॉलम मे किया जाता है, उदाहरण के टेलीफोन डायरेक्टरी मे दो या तीन कॉलम मे डाटा दर्शाया जाता हैं इस प्रकार के रिपोर्ट

मे सभी प्रकार का डाटा जैसे टेक्स्ट, नंबर, मेमो आदि को दर्शाया जा सकता है। इस प्रकार के रिपोर्ट का फारमेट निम्न प्रकार से दिखता है।

Columnar Report



इस प्रकार के रिपोर्ट मे एक रिकाड के फील्ड एक से नीचे दूसरे इस प्रकार से दशार्या जाता है। इस प्रकार का लेआउट साधारणतः डाटा एन्ट्री के लिये प्रयोग होता है, लेकिन रिपोर्ट यह लेआउट डाटा को देखने के लिए प्रयोग होता है। इस प्रकार के रिपोर्ट form रिपोर्ट भी कहा जाता है। इसमे एक पेज मे बहुत अधिक रिकार्ड नही दर्शाये जा सकते है। फील्ड की संख्या के अनुसार

एक पेज मे दो, तीन या चार रिकार्ड दर्शाये जाते है। इस प्रकार के लेआउट मे दायें ओर फील्ड का नाम एवं बायें ओर रिकार्ड दर्शाया जाता है। इसमे प्रत्येक रिकार्ड मे फील्ड के नाम को दर्शाया जाता है। बडे डाटा के लिये इस प्रकार का रिपोर्ट का उपयोग कम ही किया जाता है। यदि किसी रिकार्ड ऑब्जेक्ट या पिक्चर डाटा है, तब इस प्रकार रिपोर्ट दर्शाया जाता है। उदाहरण के लिए हमे कंपनी सभी उत्पाद उनके फोटो के साथ रिपोर्ट दर्शाना है, या छात्रों का डाटा उनके फोटो के साथ दर्शाना है आदि। इस मे सभी प्रकार के योग या गणनायें पेज के अंत मे या रिपोर्ट के अंत मे दर्शाये जाते है। किसी रिपोर्ट मे एक से अधिक कॉलम मे भी डाटा को दर्शाया जा सकता है। इस प्रकार की रिपोर्ट स्क्रिन पर दर्शाने के उपयुक्त है, लेकिन प्रिन्ट करने मे उपयुक्त नही है। इस प्रकार रिपोर्ट का लेआउट निम्न प्रकार से है

Create Report in design view

रिपोर्ट को विजार्ड की सहायता से बहुत आसानी से बनाया जा सकता है, लेकिन उसमे बहुत कम विकल्प होते है। लेकिन कार्य के अनुसार रिपोर्ट बनाने के लिए design view का उपयोग किया जाता है। इस प्रकार से रिपोर्ट डिजाइन करने के लिए

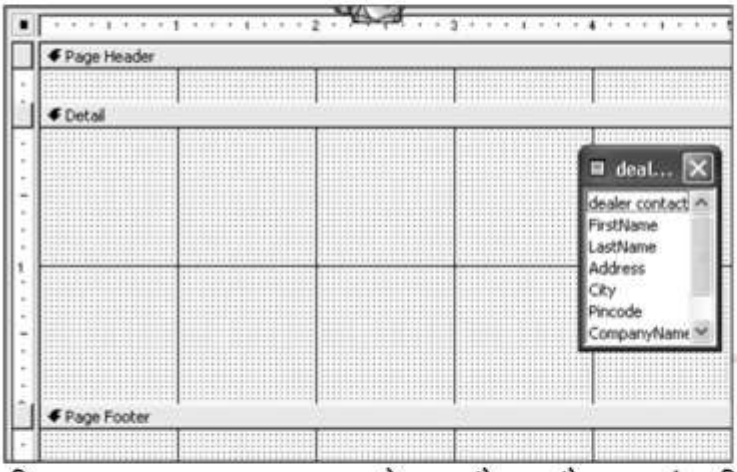
- report विंडो मे "Create report in design view" ऑयकान को double click करें।

या

- insert मेन्यू को क्लिक करे।
- report विकल्प को क्लिक करे।

- new report का डायलॉग बॉक्स दिखाई देता है, उसमे design view को क्लिक करें, तथा नीचे की ओर टेबल या क्यूरी का नाम सिलेक्ट करे।

रिपोर्ट डिजाइन की मुख्य विंडो दिखाई देती है, तथा टेबल या क्यूरी की संरचना दिखाई देती है।



रिपोर्ट के मे निम्न हिस्से होते है।

Report Header :- जो रिपोर्ट मे सबसे ऊपर होता है तथा सभी पेज मे एक समान होता है। साधारणतः इसमे एक ही प्रकार का डाटा होता है। जैस कंपनी का नाम, पता, रिपोर्ट का टाइटल आदि यह सूचना इस भाग मे डाली जाती है। कुछ रिपोर्ट मे हेडर भाग को अलग पेज मे भी प्रिन्ट किया जाता है। ऐसे स्थिती टेक्स्ट के साथ ऑब्जेक्ट भी डाल सकते है।

Page Header :- यह भी रिपोर्ट के ऊपरी भाग मे होता है, लेकिन यह प्रत्येक पेज मे दोहराया जाता है। जैसे तारीख, कॉलम कि हेडिंग, आदि

Group header :- यदि रिपोर्ट मे ग्रुप बनाया है, उस ग्रुप के अन्य रिकार्ड के पहले ग्रुप का नाम दर्शाया जाता है। साधारणतः ग्रुप को अन्य डाटा से बायें ओर रखा जाता है।

Detail :- इस भाग मे वास्तविक डाटा होता है। जैसे नाम, पता, रकम आदि

Group Footer :- एक ग्रुप का डाटा खत्म होने पर यह हिस्सा दर्शाया जाता है, जिसमे उस ग्रुप की summary दर्शाई जाती है।

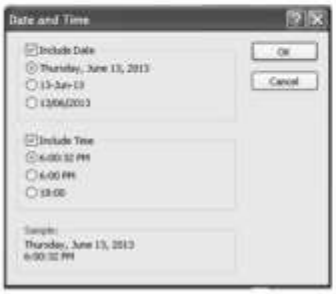
Page Footer :- यह रिपोर्ट का निचला भाग होता है, जो प्रत्येक पेज मे प्रिन्ट होता है। जैसे सभी अंको का जोड, पेज नंबर आदि

साधारणत ऊपर के हिस्से जैसे report header, page header मे लेबल कंन्ट्रोल का उपयोग किया जाता है, तथा इच्छित टेक्स्ट टाइप किया जाता है। उदाहरण के लिए हमे "Narendra Publication" किताबो की रिपोर्ट बनाना है, तब हम ऊपर की ओर लेबल कंन्ट्रोल सिलेक्ट कर, रिर्पोट के विंडो पर क्लिक करते है।, उसमे कंपनी का नाम, पता, फोन नंबर आदि टाइप करते है।

टेक्स्ट डालने के लिए लेबल कंन्ट्रोल का उपयोग किया जाता है। यह कन्ट्रोल टुलबार मे होता है। स्क्रिन पर टूलबॉक्स दर्शाने के लिए ऊपर टुलबार मे toolbox ऑयकान को क्लिक करे। आपको स्क्रिन पर टूलबॉक्स दिखाई देगा, उसमे स` "label" कंन्ट्रोल को क्लिक कर, माऊस को हेडर भाग मे क्लिक करे। आपको एक बॉक्स दिखाई देगा, उसमे इच्छित टेक्स्ट टाइप करे। उस टेक्स्ट का फारमेट सेट करने के लिए टुलबार के properties टूल को क्लिक करे।

आपको प्रापर्टी का डायलॉग बॉक्स दिखाई देगा

उसमे font name एवं font size बॉक्स मे टेक्स्ट का आकार एवं टाइप सिलेक्ट करे। इसी तरह अन्य टेक्स्ट को भी डाल कर फारमेट करे।



हेडर मे वर्तमान तारीख या रिपोर्ट प्रिन्ट करने की तारीख डालने के लिए

- insert मेन्यू मे क्लिक करे। Data and time विकल्प को क्लिक करें।
- date and time का डायलॉग बॉक्स दिखाई देगा
- उसमे से इच्छित फारमेट सिलेक्ट करें, यदि समय को भी रिपोर्ट मे दर्शाना है, तब "include time" बॉक्स चालू करे।
- इच्छित विकल्प चुनने के बाद "ok" बटन क्लिक करे।

फॉर्म मे तारीख की फील्ड दिखाई देती है। उसे इच्छित जगह पर drag करें।

हेडर भाग की विभिन्न प्रापर्टी सेट की जा सकती है। हेडर की प्रापर्टी सेट करने के लिए हेडर बार को क्लिक करे। हेडर का बार गहरे रंग मे दर्शाया जाता है। इसकी निम्न महत्वपूर्ण प्रापर्टी है।

Visible :- हेडर को रिपोर्ट के साथ दर्शाने के लिए "yes" प्रापर्टी सेट करे। यदि इसे "no" रखते है, रिपोर्ट मे यह हिस्सा अदृश्य हो जाता है।

Back color :- रिपोर्ट मे हेडर भाग का रंग सेट करने के लिए इस प्रापर्टी का उपयोग होता है। इस प्रापर्टी को क्लिक करने पर

colour pallet दिखाई देता है, उसमें से इच्छित रंग सिलेक्ट करे।

Detail: - इस हिस्से मे मुख्यतः फील्ड को जोडा जाता है।

हेडर भाग को बडा या छोटा करने के लिए उसे माऊस से ऊपर या नीचे drag करे।

रिर्पोट मे फील्ड जोडना

एक्सेस मे रिपोर्ट मे किसी फील्ड को जोडना बहुत आसान है, इच्छित फील्ड को क्लिक कर उसे रिपोर्ट मे drag करे। आप एक साथ एक से अधिक फील्ड को रिपोर्ट मे जोड सकते है। जब कोई फील्ड रिपोर्ट मे डालते है, तब उस फील्ड का नाम, एवं फील्ड दोनों रिपोर्ट मे आ जाते है, दोनों अलग–अलग ऑब्जेक्ट है। आप उन दोनों कि प्रापर्टी

अलग–अलग सेट कर सकते है, उन्हे इच्छित जगह पर move कर सकते है।

माऊस की सहायता से किसी फील्ड का आकार, स्थिती सेट कर सकते है। किसी भी फील्ड को सेट करते समय उसका डाटा दूसरे डाटा मे ना जायें इस की सावधानी ली जानी चाहिए। एक

से अधिक फील्ड को सिलेक्ट करने के लिए माऊसे को सबसे नीचे के ऑब्जेक्ट के नीचे की ओर से ऊपर के ओर माउस को drag करें। एक्सेस मे फील्ड की स्थिती सेट करने के लिए format मेन्यू मे निम्न विकल्प है।

Conditional formatting :- डाटा के मान के अनुसार फारमेटिंग सेट करने के लिए इस विकल्प का उपयोग होता है। इसे फारमेट का उपयोग number एवं date प्रकार के फील्ड मे होता है। इस विकल्प को क्लिक करने के बाद conditional formatting का डायलॉग बॉक्स दिखाई देता है। उसमे ऊपर को वर्तमान फारमेटिंग दर्शाई जाती जै। नीचे की ओर इच्छित फारमेटिंग के लिए विकल्प दर्शाये गये है। उदाहरण के यदि हमे रिपोर्ट मे जिन छात्रो पर 2000 से अधिक फीस बकाया है, वह bold फारमेटिंग मे दिखनी चाहिये। इसके लिए पहले बॉक्स "Field value is" विकल्प चुनते है। दूसरे बॉक्स मे "greater than" विकल्प चुनते है। तीसरे बॉक्स मे मान टाइप करते है। नीचे की ओर फारमेटिंग के विभिन्न विकल्प है, उनमे से इच्छित विकल्प सेट करे। यदि एक से अधिक कंडीशन पर फारमेटिंग करना है, तब "Add" बटन और कंडिशन जोड सकते है।

Align :- इस विकल्प से सभी फील्ड बॉक्स की alignment सेट कर सकते है। इस विकल्प को प्रयोग से डाटा एक सीध मे आता है।

size :- फील्ड के बॉक्स का आकार इस विकल्प मे सेट किया जा सकता है। यद्यपी इससे मूल डाटा का आकार नही बदलता है।

Horizontal spacing: - दो फील्ड की बीच के दूरीयॉ इस विकल्प से सेट की जा सकती है।

Vertical spacing: - दो रिकार्ड की बीच की दूरीयॉ इस विकल्प से सेट की जाती है।

Group :- इस विकल्प मे एक से अधिक फील्ड को एक ऑब्जेक्ट रूप मे बना सकते है, इससे उन्हे फारमेट करना, **move** करना आसान हो जाता है।

Page footer

इस हिस्से मे पेज या रिपोर्ट के अंत का डाटा सेट किया जाता है। जैसे रिकार्ड का योग, या रिपोर्ट मे डाटा की संख्या, पेज नंबर आदि।

रिपोर्ट मे पेज नंबर डालने के लिए

- Insert मेन्यू को क्लिक करे।
- Page number विकल्प को क्लिक करे।
- page number का डायलॉग बॉक्स दिखाई देता है।
- यदि सिर्फ पेज नंबर डालना है, तब "Page N" विकल्प चुनें
- यदि रिपोर्ट के संपूर्ण पेज की संख्या एवं प्रत्येक पेज का नंबर डालना है, तब "Page N of M" विकल्प सिलेक्ट करे।

- पेज नंबर की स्थिती तथा alignment निश्चत करे। यदि पहले पेज पर पेज नंबर नही चाहिए तब "Show number on first page" विकल्प को बंद करे।

किसी फील्ड का योग नीचे दर्शान के लिए sum फंक्शन का प्रयोग किया जाता है। उदाहरण के लिए हमे रिपोर्ट मे प्रयोग होने वाले सभी छात्रों के balance का योग दर्शाना है, तब page footer हिस्से मे निम्न स्टेटमेंट टाइप करे।

```
=sum([balance])
```

सभी प्रकार के प्रापर्टी एव फारमेट सेट करने के बाद रिपोर्ट को बंद करे। एक्सेस आपको रिपोर्ट सेव करने के बारे मे पूछता है। रिपोर्ट को इच्छित नाम से सेव करे।

Create Report by using multiple table

रिपोर्ट का मुख्य कार्य, इच्छित डाटा का प्रदर्शित करना है। बहुत बार इच्छित डाटा एक से अधिक टेबल से हि प्राप्त किया जा सकता है। मध्यम या बडे डाटाबेस मे एक से अधिक टेबल का प्रयोग किया जाता है, तथा प्रत्येक टेबल मे अलग–अलग फील्ड होती है। अधिकांश रिपोर्ट मे अलग–अलग टेबल मे की फील्ड की आवश्यकता होती है। ऐसी स्थिती मे हमे क्यूरी का ही उपयोग करना पडता है। किसी रिपोर्ट मे टेबल की जगह क्यूरी से डाटा प्राप्त किया जाता है। क्यूरी मे हम एक से अधिक टेबल का प्रयोग कर डाटा चुनते है। क्यूरी का प्रयोग कर रिपोर्ट बनाने के लिए

- Insert मेन्यू को क्लिक करें, report विकल्प को क्लिक करे।
- new report का डायलॉग बॉक्स दिखाई देता है, उसमे इच्छित क्यरी सिलेक्ट करे।
- क्यूरी से बनाई जाने वाले रिपोर्ट के अन्य सभी विकल्प टेबल पर आधारित रिपोर्ट डिजाइन के समान ही है।

- रिपोर्ट का पेज लेआउट सेट करना
- एक्सेस रिपोर्ट को default पेज मे सेट कर लेता है, लेकिन कार्य के अनुसार पेज को सेट करने के लिए निम्न विकल्प है।
- file मेन्यू को क्लिक करे।
- Page setup विकल्प को क्लिक करे।
- page setup का डायलॉग बॉक्स दिखाई देता है। उसमे तीन टेब है

Margins :- इस विकल्प मे पेज की चारो ओर की मार्जिन सेट की जा सकती है। इसमे print data only विकल्प है, यदि इस चालू करते है, तब वह ग्राफिक्स ऑब्जेक्ट जैसे लाइन, बॉक्स आदि प्रिन्ट नही करता है, सिर्फ टेबल का डाटा प्रिन्ट करता है।

Page :- इस टेब मे पेज का आकार, दिशा आदि सेट किया जा सकता है।

column :- इस विकल्प से डाटा एक पेज मे कितने कॉलम मे आना चाहिए तथा प्रत्येक कॉलम की चौडाइ आदि सेट कि जा सकती है।

सभी विकल्प सेट होने के बाद ok बटन क्लिक करे।

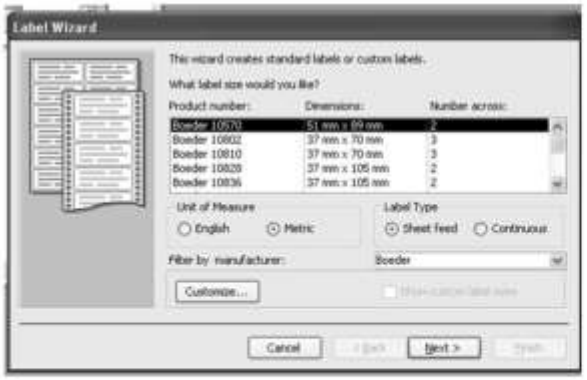
Report प्रिन्ट करना

रिपोर्ट, फॉर्म या टेबल प्रिन्ट करने के लिए

जो घटक प्रिन्ट करना है, उसे खोलें

- File मेन्यू क्लिक करे।
- Print विकल्प क्लिक करे।
- print डायलॉग बॉक्स खुल जाता है।
- उसमे कितने पेज तथा प्रत्येक पेज की कॉपीयॉ सेट करे।
- ok बटन को क्लिक करे।

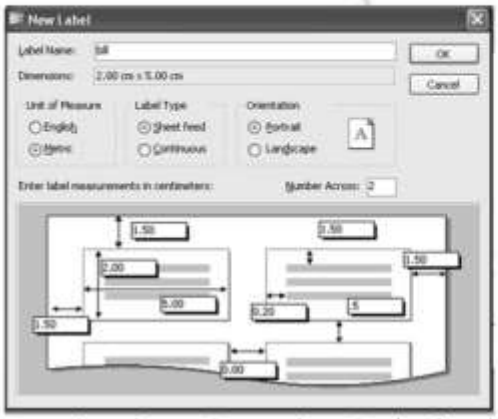
Mailing Label



टेबल के डाटा का प्रयोग कर, इच्छित व्यक्तियों के नाम एवं पते के लेबल बनाये जा सकते है। बहुतसी कंपनीयों मे ग्राहक, डिलर आदि का डाटा संग्रहित किया जाता है। संग्रहित व्यक्तियों से पत्र व्यवहार करने

के लिए mailing label बहुत उपयोगी है। इससे एक साथ इच्छित व्यक्तियो के नाम, पता फोन नंबर आदि पेज पर प्रिन्ट किये जा सकते है। कुछ क्षेत्रो मे प्रत्येक माह या निश्चित अंतराल मे प्रत्येक ग्राहक को बील भेजा जाता है, जैसे मोबाइल फोन बील, बीजली का बीज आदि इस स्थिती मे mailing label प्रिन्ट कर उन्हे लिफाफे पर चिपकाया जाता है। mailing label बनाने के लिए निम्न पद्ो का प्रयोग किया जाता है।

- insert मेन्यू को क्लिक करे।
- Report विकल्प को क्लिक करे।
- new report का डायलॉग बाक्स खुल जाता है। उसमे



label wizard विकल्प को सिलेक्ट करे, निचे की ओर टेबल का नाम सिलेक्ट करे

Print Preview and Layout Preview

प्रिन्ट प्रिव्यु के द्वारा आप अपने रिकार्ड को प्रिन्टेबल फॉरमेट मे देख सकते है। जिससे रिपोर्ट के बारे मे पता चलता है, कि आपके प्रिन्ट कैसा आयेगा साथ ही आप डाटा के एक साथ अनेक पेज को एक बार मे ही देख सकते है। इसके विपरीत लेआउट से हमे अपने पेज को quick view मिलता है। लेकिन यह आपको केवल सेम्पल डाटा के साथ प्रिव्यु दिखाता है। साथ ी यह आपके द्वारा दिये गये criteria को ignore कर देता है।

Creating Macros

मायक्रो को बनाने के लिए प्रथम new पर क्लिक करना होगा। ऐसा करने पर सामने एक स्क्रिन आ जायेगी जिस पर एक तरफ action एंव दूसरी तरफ comment विकल्प दर्शाया जाता है। जिसमे आपको action वाले भाग मे आप जिस कार्य को करना चाह रहे है उस कार्य को प्रथम लाइन मे एवं दूसरी लाइन मे उसका mode बता सकते है। एवं name मे जाकर उस कार्य का नाम देकर मायक्रो बना सकते है। इस पर क्लिक करने से हमारा मायक्रो रन हो जाएगा एवं जो एक्शन हमने दिया है वह परफॉर्म हो जाएगा ।

Specifying Macro Arguments

इसमे object के अनुसार विकल्प मिलते है, जिनका प्रयोग हम अपने action के लिए करते है। जैसे कि टेबल के लिए action argument मे आपको table name, data mode , view मिलता है। जिसमे आप अपने टेबल का नाम, डाटा मोड यानि डिजाइन मोड, एडिट मोड, एवं पायवोट मोड मिलता है। जिसमे द्वारा आप अपने मायक्रो के कार्य पद्धती को चुन सकते है।

Running Macro

जब आप अपना मायक्रो बना चुके हो तो उसे चलाने के लिए उस पर दो बार क्लिक करें या मायक्रो विन्डो पर बने run पर क्लिक कर दे इसमे आपका macro रन हो जायेगा। या आप चाहे तो इसे अपने फॉर्म आदि म लिंक करके भी रन करा सकते है।

BCST

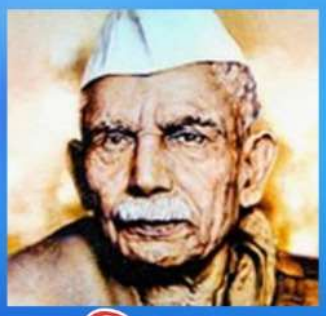
कनक कम्प्यूटर एजुकेशन

माखनलाल चतुर्वेदी वि.वि. से संबध्द

DCA / PGDCA

माखनलाल चतुर्वेदी विश्वविद्यालय भारत का एक प्रमुख विश्वविद्यालय है जो की पत्रकारिता तथा कम्प्यूटर शिक्षा के क्षेत्र में कार्य करता है । इसकी स्थापना मध्यप्रदेश शासन के व्दारा 1990 में की गयी ।

हमारी संस्था का उद्देश्य बेहतर तकनीकी ज्ञान रखने वाले विद्यार्थियों को तैयार करना है, जिससे की वे अपने भविष्य को अच्छा बनाने के साथ साथ सकारात्मक सोच रखते हुए देश के विकास मे भी भरपूर सहयोग कर सके ।



एक कदम उत्कृष्टता की ओर....

श्री फ्लेक्स मुलताई