

ऑन लाइन पाठ्य सामग्री

1PGDCA3(B)
DATABASE USING MS-ACCESS
(Elective-I)

इकाई - एक

मनोज निवारिया

सहा. प्राध्यापक, कम्प्यूटर विज्ञान एवं अनुप्रयोग
माखनलाल चतुर्वेदी राष्ट्रीय पत्रकारिता एवं संचार विश्वविद्यालय, भोपाल



माखनलाल चतुर्वेदी राष्ट्रीय पत्रकारिता एवं संचार
विश्वविद्यालय

बी-38, विकास भवन, एम.पी. नगर, झोन - I, भोपाल





डेटाबेस यूजिंग एमएस-एक्सेस

माइक्रोसॉफ्ट एक्सेस

माइक्रोसॉफ्ट ने 13 नवंबर, 1992 को एक्सेस का पहला वर्जन 1.0 जारी किया और मई 1993 में एक्सेस 1.1 रिलीज किया। इसके बाद एम एस एक्सेस के कई वर्जन बाजार में आये सभी वर्जन की अपनी अलग अलग विशेषताये हैं।

एम एस एक्सेस एक Data Management System Software है। ये MS Office का ही एक Software है। इसकी मदद से खुद का Data Entry Program बनाया जा सकता है। इसमें किसी विशेष समूह या संस्था की अलग-अलग जानकारी को रखा जाता है, जैसे कि किसी कॉलेज के Students का Data, किसी कंपनी में काम करने वाले कर्मचारियों की जानकारी जैसे उनका नाम, पता, सैलरी, उम्र आदि।

Introduction to database -What is a Database ?

डेटाबेस का परिचय - डेटाबेस क्या होता है ?

डेटाबेस के बारे में जानने से पहले हमें डाटा क्या होता है यह पता होना चाहिए। डेटा का अर्थ है वैल्यू, जैसे 27, 2500, मोती। डेटा जिसका कुछ अर्थ हो उसे जानकारी कहते हैं। मीनिंगफुल डाटा / सार्थक डेटा का अर्थ है सूचना (इंफॉर्मेशन)। मतलब ऐसा डाटा जिसके साथ एट्रिब्यूट भी हो उसे सूचना (इंफॉर्मेशन) कहा जाता है।

डाटा को तथ्यों या आंकड़ों के रूप में परिभाषित किया जाता है या ऐसी जानकारी जो एक कंप्यूटर में संग्रहित या उपयोग की जा सकती है जैसे टेक्स्ट संख्याएं यहां चित्र या धोनी का डिजिटल रूप।

एक डेटाबेस डाटा का एक व्यवस्थित और संगठित संग्रह होता है। सामान्यतः इसे कंप्यूटर सिस्टम पर इलेक्ट्रॉनिक रूप से रखा (संग्रहीत) और एक्सेस किया जाता है। जहां डेटाबेस अधिक जटिल होते हैं, उन्हें अक्सर औपचारिक डिजाइन और मॉडलिंग तकनीकों का उपयोग करके विकसित किया जाता है।

डेटाबेस में किसी समूह या संस्था की जानकारी रखी जाती है।

डेटाबेस मैनेजमेंट सिस्टम (DBMS) : डेटाबेस मैनेजमेंट सिस्टम एक सॉफ्टवेयर होता है, जिसके अंदर डेटाबेस तैयार किए जाते हैं । डेटाबेस मैनेजमेंट सिस्टम में डाटा को व्यवस्थित रूप से स्टोर किया जाता है, रिप्लाइ किया जाता है और एक्सेस किया जाता है।

डेटाबेस मैनेजमेंट सिस्टम (DBMS) वह सॉफ्टवेयर है जो डेटा को रखने और उसका विश्लेषण करने के लिए अंतिम उपयोगकर्ताओं (end user), अनुप्रयोगों (applications) और डेटाबेस के साथ सहभागिता (interact) करता है।

डेटाबेस मैनेजमेंट सिस्टम (DBMS) एक एप्लीकेशन प्रोग्राम है जो बड़ी मात्रा में जानकारी के प्रवेश (data entry), डेटा के कुशल भंडारण (storage), प्रसंस्करण (processing), पुनर्प्राप्ति (retrieval) प्रदान करता है।

एक डेटाबेस एक डेटा संरचना (Data Structure) है जो संगठित जानकारी संग्रहीत करता है।

अधिकांश डेटाबेस में कई टेबल होते हैं, जिनमें से प्रत्येक में कई फ़ील्ड हो सकते हैं। उदाहरण के लिए, एक कंपनी डेटाबेस में उत्पादों, कर्मचारियों और वित्तीय रिकॉर्ड के लिए टेबलों (tables) को शामिल किया जा सकता है। इन टेबलों में से प्रत्येक में अलग-अलग फ़ील्ड होंगे, जो टेबलों में संग्रहीत जानकारी से संबंधित होंगे ।

प्रारंभिक डेटाबेस अपेक्षाकृत "फ्लैट" थे, जिसका अर्थ है कि वे एक स्प्रेडशीट की तरह सरल पंक्तियों (rows) और स्तंभों (columns) तक सीमित थे। हालांकि, आज के रिलेशनल डेटाबेस उपयोगकर्ताओं को, विभिन्न टेबलों में संग्रहीत डेटा के संबंधों के आधार पर, जानकारी तक पहुंचने, अपडेट करने और खोज करने की अनुमति देते हैं। रिलेशनल डेटाबेस कई डेटाबेस पर आधारित क्वेरिस (Queries) चला सकते हैं। जबकि शुरुआती डेटाबेस केवल पाठ (Text) या संख्यात्मक (Numeric) डेटा को संग्रहीत कर सकते थे । आधुनिक डेटाबेस उपयोगकर्ताओं को अन्य डेटा प्रकार जैसे ध्वनि क्लिप, चित्र और वीडियो को संग्रहीत करने की भी अनुमति देता है।

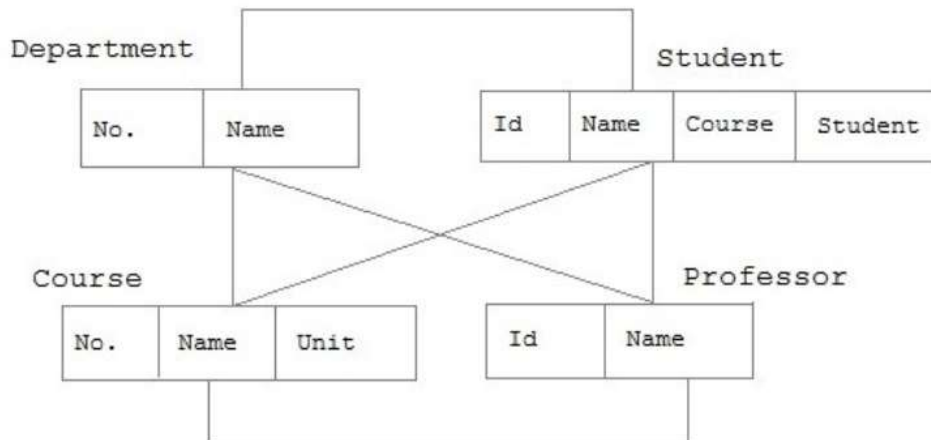
डेटाबेस मैनेजमेंट सिस्टम के कुछ उदाहरण : Dbase, Sybase, FoxPro, IMS, Microsoft Access, DB2, Oracle, SQL Server, MySQL etc.

डाटा मॉडल :

डेटाबेस मैनेजमेंट सिस्टम में डेटाबेस बनाए जाते हैं । डेटाबेस में डाटा के मध्य किस तरह का संबंध है, यह इस बात पर निर्भर करता है कि डेटाबेस किस डाटा मॉडल पर आधारित है । किसी डेटाबेस में उपस्थित डाटा के मध्य संबंधों को बताने के लिए और डेटा की प्रकृति बताने के लिए डाटा मॉडल का उपयोग किया जाता है । डेटाबेस मैनेजमेंट सिस्टम डाटा मॉडल पर आधारित होते हैं ।

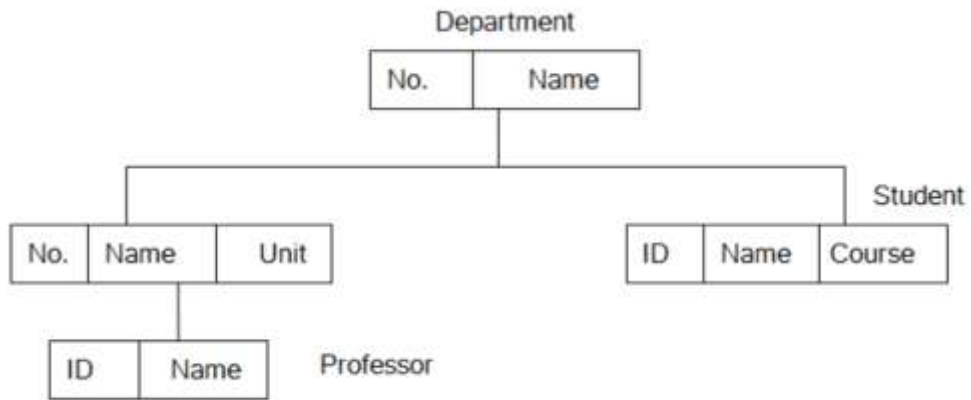
मुख्य रूप से तीन प्रकार के डाटा मॉडल होते हैं । अतः इन पर आधारित तीन प्रकार के डेटाबेस होते हैं जो निम्नलिखित हैं :

1. **नेटवर्क डेटाबेस (Network Database) :** नेटवर्क मॉडल पर आधारित डेटाबेस को नेटवर्क डेटाबेस कहते हैं इस प्रकार के डेटाबेस में डाटा को रिकार्ड (Record) के रूप में दर्शाया जाता है और डेटा के बीच संबंध लिंक (Link) के रूप में दर्शाया जाता है । इस डाटा मॉडल में डाटा को एक ग्राफ (Graph) के रूप में व्यवस्थित किया जाता है, जिसमें एक रिकॉर्ड से दूसरे रिकॉर्ड तक पहुंचने की कई रास्ते (Paths) हो सकते हैं ।

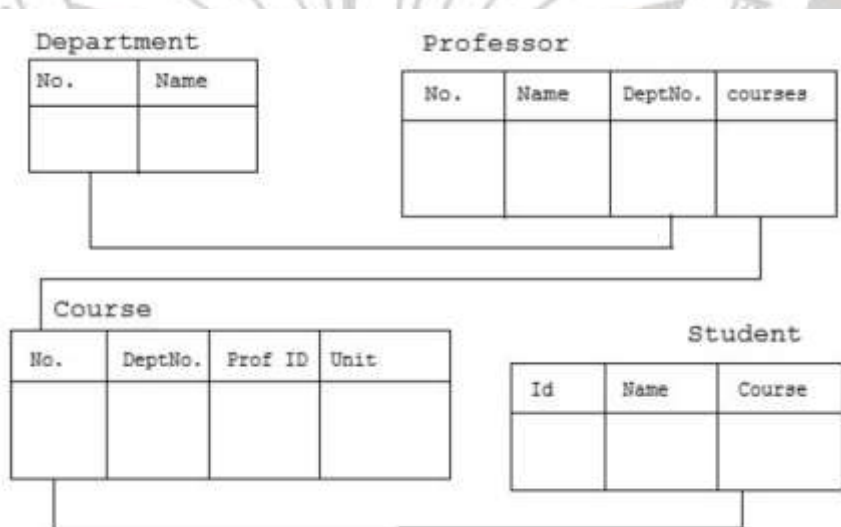


2. **हायरार्चिकल डेटाबेस (Hierarchical Database) :** हायरार्चिकल मॉडल पर आधारित डेटाबेस को हायरार्चिकल डेटाबेस कहते हैं । इस तरह के डेटाबेस में डाटा को ट्री स्ट्रक्चर (पेरेंट चाइल्ड) के रूप में व्यवस्थित किया जाता है और दर्शाया जाता है । ट्री स्ट्रक्चर (Tree Structure) में नोड्स (Nodes) लिंक के माध्यम से जुड़े हुए होते हैं । डाटा में पेरेंट चाइल्ड रिलेशनशिप (Parent-Child Relationship) होती है । इस

डाटा मॉडल में प्रत्येक चाइल्ड रिकॉर्ड का केवल एक पेरेंट होता है, लेकिन एक पेरेंट रिकॉर्ड की बहुत सारे चाइल्ड रिकॉर्ड हो सकते हैं ।



3. **रिलेशनल डेटाबेस (Relational Database)** : रिलेशनल मॉडल पर आधारित डेटाबेस को रिलेशनल डेटाबेस कहते हैं । इस डेटाबेस में डाटा टेबल्स (Tables) के रूप में संग्रहित होता है । टेबल में रो (Row) और कॉलम (Column) होते हैं । एक रो में एक रिकॉर्ड संग्रहित किया जाता है ।



Why use a Relational Database

हायरार्चिकल डेटाबेस में यह नुकसान है कि इसका स्ट्रक्चर सभी तरह के डेटाबेस में अप्लाई (apply) नहीं किया जा सकता यह डेटाबेस फ्लैक्सिबल (flexible) नहीं होता है ।

नेटवर्क डेटाबेस में यह नुकसान है कि इसमें रिकॉर्ड को अपडेट (update) और इंsert (insert) करना बहुत जटिल (complex) होता है।

रिलेशनल डेटाबेस का प्रयोग सबसे ज्यादा किया जाता है क्योंकि यह सबसे सरल है और आसानी से उपयोग किया जा सकता है । इस डेटाबेस में डाटा एक टेबल (Table) के अंदर स्टोर किया जाता है । टेबल में रो (Row) और कॉलम (Column) होते हैं । टेबल में एक रो एक एंटीटी (Entity) के रिकॉर्ड को दर्शाती है और कॉलम, एट्रीब्यूट्स (Attributes) / फील्ड्स (Fields) दर्शाते हैं । इस डेटाबेस में डाटा को इंsert (Insert), डिलीट (Delete) और अपडेट (update) करना बहुत आसान होता है । डाटा को इंsert , डिलीट और अपडेट करने के लिए एसक्यूएल क्वेरीज (SQL Queries) का उपयोग किया जाता है । रिलेशनल डेटाबेस में टेबल को रिलेशन भी कहते हैं इसीलिए इस डेटाबेस को रिलेशनल डेटाबेस कहा जाता है ।

कुछ प्रचलित रिलेशनल डीबीएमएस (Relational DBMS) के उदाहरण हैं : ओरेकल (Oracle), एसक्यूएल सर्वर (SQL Server), माय एसक्यूएल (MySQL), DB2 आदि ।

रिलेशनल डेटाबेस का सबसे मुख्य फायदा यह है कि इसमें डाटा टेबल के रूप में होता है जिसके कारण उपयोगकर्ता (User) इसे आसानी से समझ लेते हैं और एक्सेस (Access) कर लेते हैं । रिलेशनल मॉडल बहुत पावरफुल (Powerful) और सिंपल (Simple) है तथा बहुत ज्यादा फ्लैक्सिबल (Flexible) भी है ।

इसका यह नुकसान है कि ज्यादा डाटा होने पर यह जटिल बन जाता है और डाटा के मध्य की रिलेशनशिप भी जटिल और कठिन हो जाती है ।

रिलेशनल डेटाबेस मैनेजमेंट सिस्टम (RDBMS) : रिलेशनल डेटाबेस मैनेजमेंट सिस्टम एक ऐसा डेटाबेस मैनेजमेंट सिस्टम होता है, जो रिलेशनल डाटा मोडल पर आधारित है और जिसके अंदर रिलेशनल डेटाबेस तैयार किए जाते हैं ।

रिलेशनल डेटाबेस मैनेजमेंट सिस्टम के कुछ उदाहरण : Microsoft Access, Oracle, SQL Server, MySQL etc.

Overview of Database Design / डेटाबेस डिजाइन का अवलोकन :

डेटाबेस डिजाइन को इस प्रकार परिभाषित किया गया है :

अनुप्रयोगों (applications) के एक निर्धारित सेट के लिए एक संगठन (organisation) में उपयोगकर्ताओं की सूचना आवश्यकताओं (information needs) को समायोजित करने के लिए एक या अधिक डेटाबेस की तार्किक और भौतिक संरचना (logical and physical structure) डिजाइन करना ।

डेटाबेस डिजाइन एक डेटाबेस के विस्तृत डेटा मॉडल को प्रोड्यूस करने की प्रक्रिया है । डेटाबेस डिजाइन टर्म एक संपूर्ण डेटाबेस सिस्टम के डिजाइन के बहुत सारे भिन्न भागों का वर्णन करने के लिए प्रयोग की जाती है।

डेटाबेस डिजाइन के मुख्य स्टेप्स निम्नलिखित हैं :

1. Planning and Analysis
2. Conceptual design
3. Logical design
4. Physical design

1. Planning and Analysis (योजना और विश्लेषण) :

साइट अवलोकन (On Site Observation), साक्षात्कार (Interview), मैनुअल और प्रलेखन (Manual and Documentation), प्रश्नावली (Questionnaire) के आधार पर जानकारी प्राप्त करना और उसका विश्लेषण करना । जहाँ का भी डेटाबेस तैयार करना है वहाँ खड़े होकर अवलोकन (observe) करना अर्थात यह देखना कि वहाँ क्या क्या हो रहा है, किस तरह कार्य किया जा रहा है, कार्य को करने की क्या प्रक्रिया है, आदि । अवलोकन करने के बाद उस व्यक्ति का Interview लेना जिससे जानकारी प्राप्त करना है । आवश्यकताओं का संकलन साधारण: संगठन के यूजरों से पूछताछ करके किया जाता है । जानकारी को इकट्ठा करने का एक दूसरा तरीका प्रश्नावलियों का उपयोग है । अनुभव द्वारा प्रभावशाली प्रश्नवली बनाकर उन्हें लोगों में बाँट दिया जाता है और उनके उत्तर फॉर्म के रूप में एकत्र कर लिए जाते हैं ।

2. Conceptual design (वैचारिक डिजाइन या प्रारूप) :

डेटाबेस से सम्बंधित जानकारी को एकत्रित करने के बाद उस डाटा पार विचार किया जाता है कि जो जानकारी एकत्रित की गई है वह सही है या नहीं ।

एक बार डेटाबेस डिजाइनर उस डाटा के बारे में जानकारी प्राप्त कर लेता है, जो डेटाबेस में स्टोर होनी है, तब वह यह निर्धारित करता है कि कहां पर डाटा में निर्भरता (Dependency) है । कभी-कभी जब data बदल जाता है , तो वे अपने आप दूसरे डेटा को बदल सकते हैं , जो उचित नहीं होता है ।

डेटा मॉडल, वैचारिक डेटाबेस डिजाइन प्रक्रिया (Conceptual Database Design Process) का एक भाग है । दूसरा भाग कार्यात्मक मॉडल है (Functional Model) ।

डेटा मॉडल को प्लानिंग और एनालिसिस स्टेज से इनपुट मिलता है । डाटा मॉडल के दो आउटपुट होते हैं, पहला Entity-Relationship (ER Diagram) डायग्राम जो data-structure को चित्र के रूप में दिखाता है । ER Diagram की सहायता से एंड यूजर को डाटा मॉडल के विषय में बताना आसान हो जाता है । दूसरा आउटपुट है डाटा डिक्शनरी डॉक्यूमेंट । डाटा डिक्शनरी वह डॉक्यूमेंट है जो डाटा ऑब्जेक्ट, उनके मध्य संबंध और डेटाबेस के लिए आवश्यक नियमों की विस्तार से व्याख्या करता है ।

Conceptual design के टूल्स :-

- A - ER Diagram
- B - Relationship
- C - Normalization
- D - Anomaly check

3. Logical Design (लॉजिकल डिजाइन)

डेटाबेस का दूसरा चरण लॉजिकल डिजाइन होता है । डेटाबेस डिजाइन करते समय आवश्यक जानकारियों को इकट्ठा करने के बाद उस इंफॉर्मेशन को अलग-अलग बांट कर Data का लॉजिकल स्ट्रक्चर तैयार किया जाता है ।

लॉजिकल डिजाइन को डेटाबेस के विस्तृत वर्णन के बारे में सोच है जैसे डेटाबेस में क्या इंफॉर्मेशन स्टोर होगी , किस टाइप की इंफॉर्मेशन स्टोर होगी आदि ।

4. Physical Design (फिजिकल डिजाइन)

डेटाबेस का तीसरा चरण डेटाबेस डिजाइन को फिजिकली डिजाइन करना है अर्थात आवश्यक सूचनाओं को इकट्ठा करने उनके बारे में विस्तृत जानकारी निकालने के बाद उस डाटा को भौतिक रूप से प्रयोग में लाया जाता है ।

डेटाबेस की फिजिकल डिजाइन स्टोरेज मीडिया पर डेटाबेस के फिजिकल कॉन्फिगरेशन को specify करता है । फिजिकल डिजाइन डेटाबेस के सबसे नीचे के लेवल के डेटा का वर्णन करने के लिए प्रयोग किया जाता है । फिजिकल डिजाइन के अंतर्गत डेटाबेस के वास्तविक स्ट्रक्चर को डिजाइन किया जाता है ।

Data Normalization (Determining tables, Determining Fields, Determining Relationships)

नोर्मलाइजेशन क्या हैं ? (What is Normalization?) :

Normalization (मानकीकरण) डेटाबेस डिजाइन को सरल बनाता है । इसे डेटाबेस नोर्मलाइजेशन या डेटा नोर्मलाइजेशन के रूप में भी जाना जाता है । नोर्मलाइजेशन डेटाबेस डिजाइन का एक महत्वपूर्ण हिस्सा है , क्योंकि यह डेटाबेस की गति, सटीकता में सहायता करता है। नोर्मलाइजेशन एक ऐसी तकनीक है जिसका उपयोग रिलेशनल (relational) डेटाबेस को डिजाइन करने के लिए किया जाता है । नोर्मलाइजेशन दो चरणों में होती है जो डाटा से Repeating Groups को हटा कर टेबल के रूप में रखती है और उसके बाद रिलेशनल टेबल से एक जैसी यानी कि duplicate entries को हटाती है ।

नोर्मलाइजेशन का काम redundancy को कम करना होता है और redundancy को कम करने का अर्थ है एक information को एक ही बार स्टोर करना। एक ही Information को एक से ज्यादा बार स्टोर करने से स्टोरेज बढ़ता है। Normalized Relation से मतलब है कि जब भी

डेटाबेस में रिलेशनस को परिवर्तित (alter) किया जाए, तो इनफार्मेशन गुम नहीं होना चाहिए।

नार्मल फॉर्म में किसी डेटाबेस टेबल को बनाना नोर्मलाइजेशन कहलाता है । नोर्मलाइजेशन में किसी टेबल के प्रत्येक एट्रिब्यूट के बीच आपसी निर्भरता का विश्लेषण किया जाता है और फिर बड़े टेबल का प्रोजेक्शन लेकर उससे छोटे-छोटे टेबल बनाये जाते हैं । प्रोजेक्शन ऑपरेशन का अर्थ है किसी टेबल में उपस्थित एट्रिब्यूट में से कुछ एट्रिब्यूट को लेकर टेबल बनाना ।

डेटाबेस को नोर्मलाइजेशन करके, आप डेटा को टेबल और कॉलम में व्यवस्थित कर सकते हैं। आप सुनिश्चित करते हैं कि प्रत्येक टेबल में केवल संबंधित डेटा होता है। यदि डेटा सीधे संबंधित नहीं है, तो आप उस डेटा के लिए एक नई टेबल बनाते हैं।

उदाहरण के लिए, यदि आपके पास “ग्राहक” टेबल है, तो आप आमतौर पर उन उत्पादों के लिए एक अलग टेबल बना सकते हैं, जिन्हें वे ऑर्डर कर सकते हैं (आप इस टेबल को “Product” कह सकते हैं)। आप ग्राहकों के आदेशों के लिए एक और टेबल तैयार करेंगे (“ऑर्डर” टेबल कहा जाता है)। और यदि प्रत्येक ऑर्डर में कई आइटम हो सकते हैं, तो आप आमतौर पर प्रत्येक ऑर्डर आइटम को स्टोर करने के लिए एक और टेबल बनाते हैं (“ऑर्डरइटम” टेबल कहा जाता है)। इन सभी टेबलों को उनकी प्राथमिक कुंजी (Key) से जोड़ा जाएगा, जो आपको इन सभी टेबलों में संबंधित डेटा ढूँढने की अनुमति देता है (जैसे किसी दिए गए ग्राहक द्वारा सभी ऑर्डर) ।

नोर्मलाइजेशन के लाभ (Benefits of Normalization) :

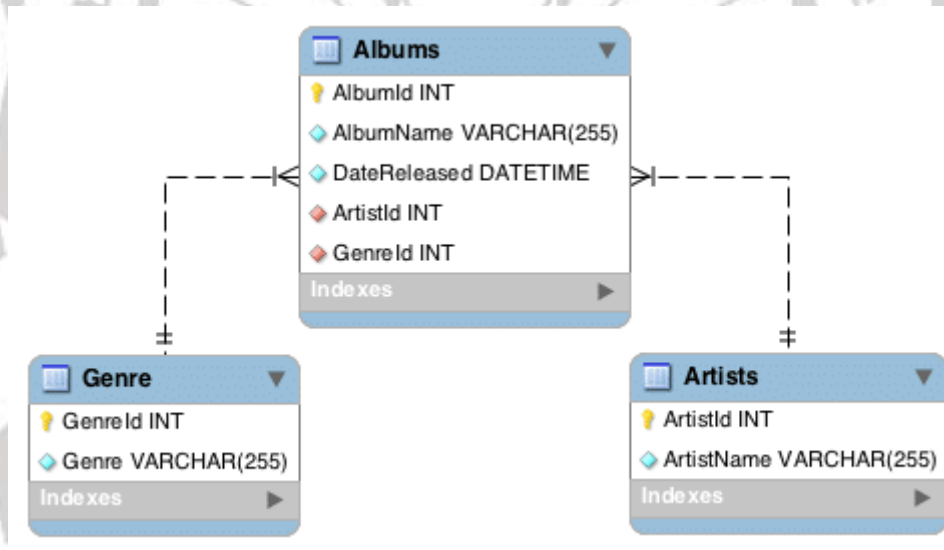
- यह डेटा रिडंडेंसी को कम करता है (डुप्लिकेट डेटा)।
- यह शून्य मूल्यों (null values) को कम करता है।
- यह अधिक कॉम्पैक्ट डेटाबेस में परिणाम देता है। (कम डेटा रिडंडेंसी / शून्य मानों के कारण)।
- यह डेटा संशोधन मुद्दों को कम करता है ।
- यह क्वेरी को सरल बनाता है।

- यह डेटाबेस संरचना को स्वच्छ और समझने में आसान बनाता है।
- आप मौजूदा डेटा को जरूरी रूप से प्रभावित किए बिना डेटाबेस का विस्तार कर सकते हैं।
- इसमें इंडेक्स खोजना, सॉर्ट करना और बनाना तेजी से हो सकता है, क्योंकि टेबल संकुचित होते हैं, और डेटा पंक्ति पर अधिक पंक्तियां फिट होती हैं।

एक नॉर्मलाइजेशन डेटाबेस का उदाहरण (Example of a Normalized Database) :

एक रिलेशनल डेटाबेस डिज़ाइन करते समय, आमतौर पर एक स्कीमा बनाने से पहले डेटा को नॉर्मलाइजेशन करता है। डेटाबेस स्कीमा संगठन और डेटाबेस की संरचना को निर्धारित करता है - मूल रूप से डेटा कैसे संग्रहीत किया जाएगा।

Normalization डेटाबेस स्कीमा का एक उदाहरण यहां दिया गया है:



यह स्कीमा डेटा को तीन अलग-अलग टेबलों में विभाजित करता है। प्रत्येक टेबल उस डेटा में काफी विशिष्ट है जो इसे स्टोर करती है - एल्बमों के लिए एक टेबल है, कलाकारों के लिए एक है, और दूसरा जो डेटा के लिए विशिष्ट डेटा रखता है। हालांकि, रिलेशनशिप मॉडल हमें इन

टेबलों के बीच रिलेशन बनाने की इजाजत देता है, हम यह पता लगा सकते हैं कि कौन से एल्बम किस कलाकार से संबंधित हैं, और किस शैली में वे संबंधित हैं।

Normalization के प्रकार (Types of Normalization) :

1. First Normal Form (1NF):-

First Normal Form को 1NF से भी दर्शाते हैं। एक Relational Table 1NF में होती है, जब कॉलम की सभी values Atomic (एकल मान) होती है। प्रत्येक कॉलम में प्रत्येक रो के लिए सिर्फ एक वैल्यू होनी चाहिए। रिलेशनल डेटाबेस में बनाया गया टेबल by default 1NF में होता है। हम इसे इस तरह से भी समझ सकते हैं कि कोई table 1NF में होती है यदि

- हर एक Cell में Atomic Value होनी चाहिए।
- कॉलम में entries/values का प्रकार एक जैसा ही होना चाहिए। एक कॉलम में सभी वैल्यूज का डोमेन (Domain) एक जैसा होना चाहिए।

2. Second Normal Form (2NF):-

एक टेबल या रिलेशन तब 2nd Normal Form में होता है जब वह 1st Normal Form की सभी जरूरतों को पूरी करता हो और सभी Non-Key Attributes पूरी तरह से Primary Key पर निर्भर हों।

3. Third Normal Form (3NF):-

कोई टेबल या रिलेशन तब 3rd Normal Form में होता है जब वह 2nd Normal Form की सभी जरूरतों को पूरी करता हो तथा उनमें Transitive Function Dependency नहीं होनी चाहिए।

4. बॉयस-कोड नार्मल फॉर्म / Boyce Codd Normal Form (BCNF):-

3NF, जो आज डेटाबेस में ज्ञात अधिकांश विसंगतियों को समाप्त करता है, वाणिज्यिक डेटाबेस और CASE टूल में Normalization का सबसे अच्छा मानक है। इसमें उपस्थित कुछ शेष

विसंगतियों को बॉयस-कोड नार्मल फॉर्म (BCNF) द्वारा समाप्त किया जा सकता है। BCNF को 3NF का एक मजबूत स्वरूप माना जाता है।

परिभाषा : एक टेबल R बॉयस-कोड नार्मल फॉर्म (BCNF) में है यदि प्रत्येक नॉन-ट्रिविअल FD, $X \rightarrow A$ के लिए, X एक सुपरकी है।

3NF की तुलना में BCNF, Normalization का एक मजबूत रूप है क्योंकि यह 3NF के लिए दूसरी कंडीशन को समाप्त कर देता है, जिसमें FD के दाईं ओर एक प्राइम एट्रिब्यूट (Prime Attribute) होने की अनुमति है। इस प्रकार, एक टेबल में FD के हर बाईं ओर एक सुपर की होना चाहिए। पिछली परिभाषाओं द्वारा हर टेबल जो BCNF में है वो 3NF, 2NF और 1NF में भी है।

Primary/Foreign Key, One-to-Many, Many-to-Many, One-to-One Relationship

Primary Key / प्राइमरी की :

Primary key एक टेबल में प्रत्येक रिकॉर्ड को यूनिक (अद्वितीय) रखने के लिए प्रयोग की जाती है। इस key के attribute कभी बदलना नहीं चाहिए जैसे एक व्यक्ति का एड्रेस (पता) प्राइमरी Key का हिस्सा नहीं होना चाहिए क्योंकि यह बदल सकता है। परंतु Employee code नहीं बदला जा सकता, जब तक की वह व्यक्ति उस संस्था को ना छोड़ दें। प्राइमरी की किसी भी टेबल में यूनिक की होती है जो पूरे रिकॉर्ड को रिप्रेजेंट करती है। Primary Key Column को Null नहीं रखा जा सकता है।

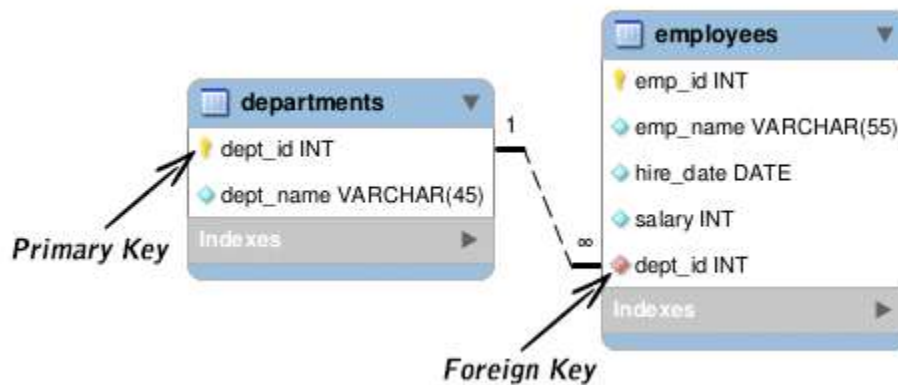
एक Table में केवल एक Primary Key हो सकती है। Primary Key में एक या अधिक फ़ील्ड होते हैं जो Table में संग्रहीत प्रत्येक रिकॉर्ड को विशिष्ट रूप से पहचानते हैं। अक्सर, एक Unique पहचान संख्या होती है, जैसे एक आईडी नंबर, एक सीरियल नंबर, या एक कोड, जो Primary Key के रूप में कार्य करता है। उदाहरण के लिए, आपके पास ग्राहक Table है जहां प्रत्येक ग्राहक के पास एक Unique ग्राहक आईडी नंबर होता है। ग्राहक आईडी फ़ील्ड, ग्राहक Table की Primary Key है। जब Primary Key में एक से अधिक फ़ील्ड होते हैं, तो यह आमतौर पर पूर्व-मौजूदा फ़ील्ड से बना होता है, जो एक साथ लेते हैं और Unique मान प्रदान करते हैं। उदाहरण के लिए, आप लोगों के बारे में एक Table के लिए, Primary Key के रूप

में अंतिम नाम, पहला नाम, और जन्म तिथि के combination(संयोजन) का उपयोग कर सकते हैं।

Foreign Key / फॉरेन की :

फॉरेन की का प्रयोग एक टेबल की, दूसरी टेबल के साथ रिलेशनशिप को स्थापित करने के लिए किया जाता है अर्थात एक टेबल की लिंक दूसरी टेबल के साथ बनाने के लिए किया जाता है । जैसे हम College के नाम से एक डेटाबेस को Create करना चाहते हैं । इस डेटाबेस में स्टूडेंट (student) तथा कोर्स (course) के नाम से 2 टेबल बनाते हैं ।

एक Table में एक या अधिक Foreign Key भी हो सकती है। एक Foreign Key में वे मान होते हैं जो किसी अन्य Table की Primary Key में मानों के अनुरूप होते हैं। उदाहरण के लिए, आपके पास employees Table है जिसमें प्रत्येक employee का एक डिपार्टमेंट आईडी नंबर होता है, जो departments Table में रिकॉर्ड के अनुरूप होता है। डिपार्टमेंट आईडी फ़ील्ड ,employee Table की एक Foreign Key है ।



Difference between Primary key and Foreign key :

प्राइमरी और फॉरेन की के बीच अंतर :

Primary Key	Foreign Key
प्राइमरी की एक कॉलम या कॉलम का सेट होता है जो एक टेबल में एक रो को यूनिक बनाता है ।	फॉरेन की एक कॉलम या कॉलम का सेट होता है जो दूसरी टेबल के एक प्राइमरी की या एक कैंडिडेट की को रेफर करता है ।
एक टेबल में सिंगल (सिर्फ एक) प्राइमरी की हो सकती है ।	एक टेबल में एक से अधिक फॉरेन की हो सकती हैं।
प्राइमरी की यूनिक होती है । प्राइमरी की नल नहीं हो सकती ।	फॉरेन की नल हो सकती हैं ।
प्राइमरी की Entity (एंटिटी) इंटीग्रिटी से सम्बंधित है ।	फॉरेन की referential (रेफरेंसीअल) इंटीग्रिटी से सम्बंधित हैं ।
प्राइमरी की को पैरेंट की (Parent Key) के नाम से भी जाना जाता है ।	फॉरेन की को चाइल्ड की (Child Key) के नाम से भी जाना जाता है ।

Connectivity and Cardinality of a Relationship :

एक Relationship की कनेक्टिविटी Relationship में संबद्ध entity के instances (उदाहरणों) के Mapping मानचित्रण का वर्णन करती है। कनेक्टिविटी की वैल्यू "एक" या "अनेक" होती हैं।

एक Binary Relationship की कार्डिनैलिटी (Cardinality) प्रत्येक दो entities के लिए संबंधित occurrences की वास्तविक संख्या है। Relationship के लिए निम्न प्रकार की कनेक्टिविटी हैं:

1. एक-से-एक (One-to-One Relationship)
2. एक से अनेक (One-to-Many Relationship)

3. अनेक -से-अनेक (Many-to-Many Relationship)

One-to-One Relationship : One-to-One (1:1) Relationship तब होता है जब एक Entity Set A का सिर्फ एक उदाहरण (instance) , एक Entity Set B के एक उदाहरण (instance) के साथ जुड़ा हुआ है । उदाहरण:

कंपनी में अधिकारियों को प्रत्येक को अपना कार्यालय सौंपा गया है। प्रत्येक अधिकारी के लिए एक अद्वितीय कार्यालय मौजूद है और प्रत्येक कार्यालय के लिए एक अद्वितीय अधिकारी मौजूद है ।

One-to-Many Relationship : One-to-Many (1: N) Relationship तब होता है जब Entity Set A के एक उदाहरण (instance) के लिए, Entity Set B के शून्य, एक या कई उदाहरण (instances) हैं । लेकिन Entity Set B के एक उदाहरण (instance) के लिए, Entity Set A का केवल एक उदाहरण (instance) है ।

उदाहरण :

एक विभाग में कई कर्मचारी होते हैं।

प्रत्येक कर्मचारी का एक विभाग होता है।

Many-to-Many (M:N) Relationship : Many-to-Many (M:N) Relationship तब होता है जब Entity Set A के एक उदाहरण (instance) के लिए, Entity Set B के शून्य, एक या अनेक उदाहरण (instance) हो । और Entity Set B के एक उदाहरण (instance) के लिए, Entity Set A के शून्य, एक या अनेक उदाहरण (instance) हो ।

उदाहरण :

कर्मचारियों को एक ही समय में दो से अधिक परियोजनाओं को सौंपा जा सकता है।

परियोजनाओं को कम से कम तीन कर्मचारियों को सौंपा जाना चाहिए।

एमएस एक्सेस 2007 का परिचय (Introduction to MS Access 2007) :

डाटाबेस डेवलपमेंट के लिए MS Access 2007 एक महत्वपूर्ण और शक्तिशाली प्लेटफार्म है । MS Access 2007 माइक्रोसॉफ्ट कंपनी द्वारा तैयार किया गया एक डेटाबेस एप्लीकेशन सॉफ्टवेयर है जिसके माध्यम से किसी भी कंपनी, संस्था आदि के बारे में हम ज्यादा से ज्यादा सूचनाओं को इकठ्ठा कर सकते हैं । तथा बाद में कोई भी विशिष्ट जानकारी जो की उस संस्था से सम्बंधित हो, उसे प्राप्त कर सकते हैं । MS Access को हम “collection of database “ भी कह सकते हैं ।

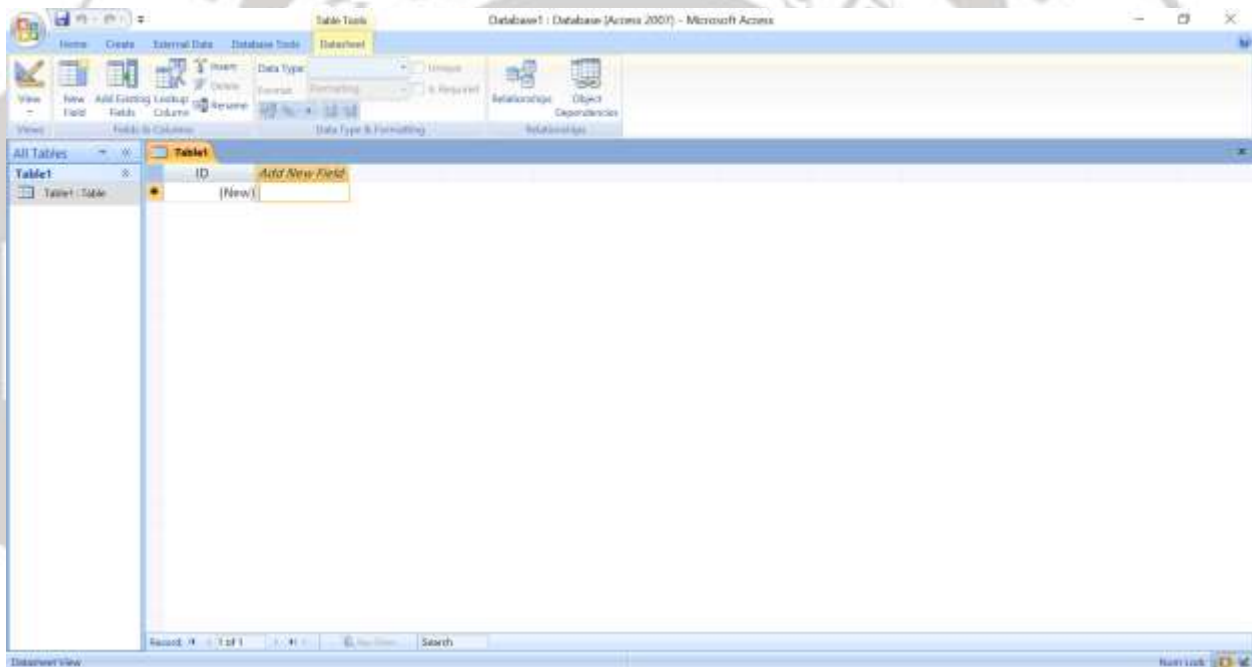
एक्सेस एक डाटाबेस मैनेजमेंट सिस्टम (डीबीएमएस) है जो एक ग्राफिकल यूजर इंटरफेस और सॉफ्टवेयर डेवलपमेंट टूल्स के साथ रिलेशनल माइक्रोसॉफ्ट जेट डाटाबेस इंजन को जोड़ती है।

- यह एक्सेस जेट डेटाबेस इंजन के आधार पर डेटा को अपने प्रारूप में स्टोर करता है।
- रिलेशनल डेटाबेस की तरह, माइक्रोसॉफ्ट एक्सेस आपको संबंधित जानकारी को आसानी से लिंक करने की अनुमति देता है ।
- यह अन्य प्रोग्राम और डेटाबेस में स्टोर डेटा को सीधे इम्पोर्ट या लिंक भी कर सकता है।
- जैसा कि इसके नाम से तात्पर्य है, एक्सेस कई लोकप्रिय पीसी डेटाबेस प्रोग्रामों सहित, डेस्कटॉप पर, सर्वर पर, मिनीकंप्यूटर पर या मेनफ्रेम पर, और संग्रहीत डेटा के साथ कई लोकप्रिय पीसी डेटाबेस प्रोग्राम समेत अन्य स्रोतों से डेटा के साथ सीधे काम कर सकता है।
- आप डेटा प्रोसेसिंग फ़ाइलों, स्प्रेडशीट्स या डेटाबेस फ़ाइलों से डेटा को डेटा निर्यात और आयात कर सकते हैं।
- एक्सेस SQL सर्वर, ओरेकल और डीबी 2 समेत ओपन डाटाबेस कनेक्टिविटी (ओडीबीसी) मानक का समर्थन करने वाले सबसे लोकप्रिय डेटाबेस के साथ काम कर सकता है।
- सॉफ्टवेयर डेवलपर्स एप्लिकेशन सॉफ्टवेयर विकसित करने के लिए माइक्रोसॉफ्ट एक्सेस का उपयोग कर सकते हैं।

एमएस एक्सेस वातावरण के साथ काम करना / (Working with MS Access Environment):

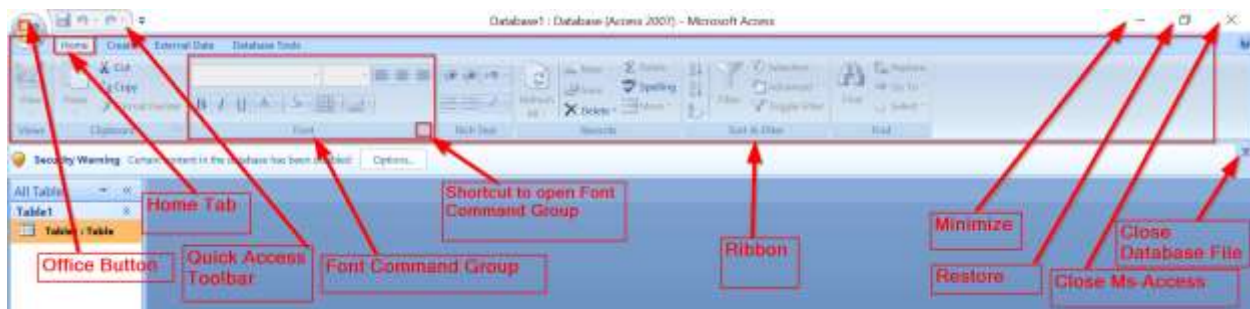
एमएस एक्सेस का ग्राफिकल यूजर इंटरफ़ेस / Graphical User Interface of Ms-Access:

एक्सेस 2007, 2010 और 2013 का का GUI (इंटरफ़ेस) काफी कुछ मिलता जुलता है । यह एक्सेस 2003 से काफी अलग है . हम यहाँ एम.एस.एक्सेस 2007 के इंटरफ़ेस पर चर्चा करेंगे . यह रिबन और क्विक एक्सेस टूलबार जैसी सुविधाओं का उपयोग करता है- जहां आपको बैकस्टेज व्यू में सामान्य कार्यों को करने के लिए कमांड मिलेंगे।



रिबन (The Ribbon) :

MS Access 2007 पारंपरिक मेनू के बजाय एक टैबड रिबन सिस्टम का उपयोग करता है। रिबन में कई टैब होते हैं, प्रत्येक में कमांड के कई समूह होते हैं। एक्सेस में सबसे आम कार्यों को करने के लिए आप इन टैब का उपयोग कर सकते हैं ।



रिबन को कैसे मिनीमाइज और मैक्सिमाइज करें / How to minimize and maximize the Ribbon :

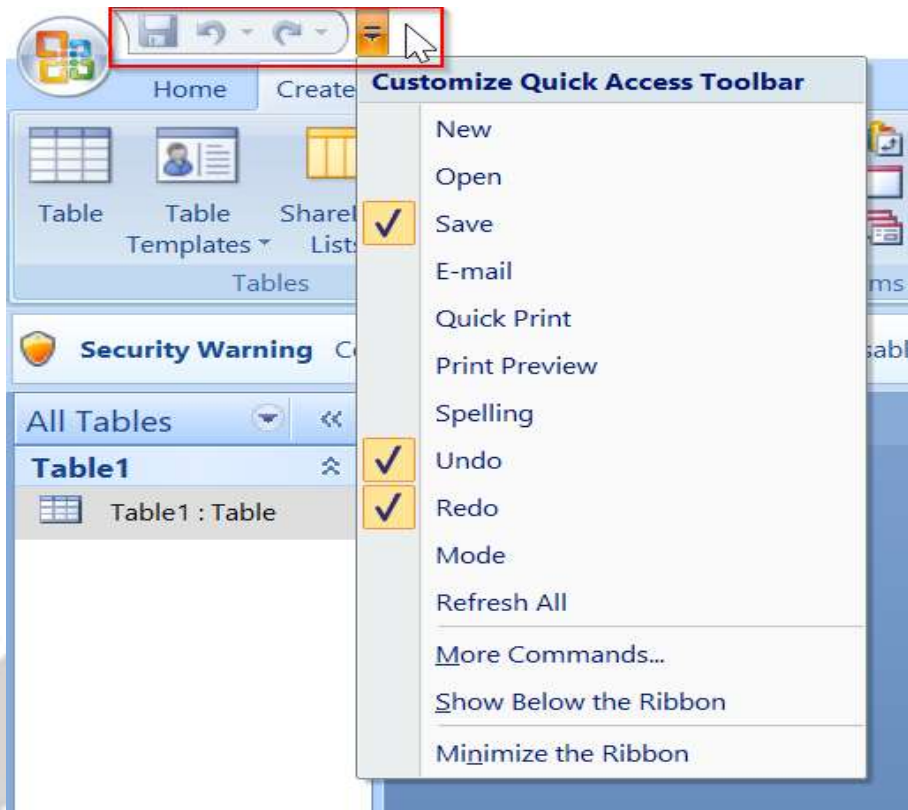
रिबन को आपके वर्तमान कार्य का जवाब देने के लिए डिज़ाइन किया गया है; हालांकि, यदि आप पाते हैं कि यह बहुत अधिक स्क्रीन स्पेस लेता है तो आप रिबन को Minimize कर सकते हैं।

इसे Minimize करने के लिए रिबन के निचले-दाएं कोने में तीर पर क्लिक करें। या किसी भी टैब पर डबल क्लिक करें। रिबन Minimize हो जाएगा। रिबन को फिर से दिखने के लिए एक टैब पर क्लिक करें। उपयोग में नहीं होने पर यह फिर से गायब हो जाएगा।

रिबन को Maximize करने के लिए, एक टैब पर क्लिक करें, फिर निचले-दाएं कोने में पिन आइकन पर क्लिक करें। या किसी भी टैब पर डबल क्लिक करें। रिबन हर समय दिखाई देगा।

The Quick Access toolbar

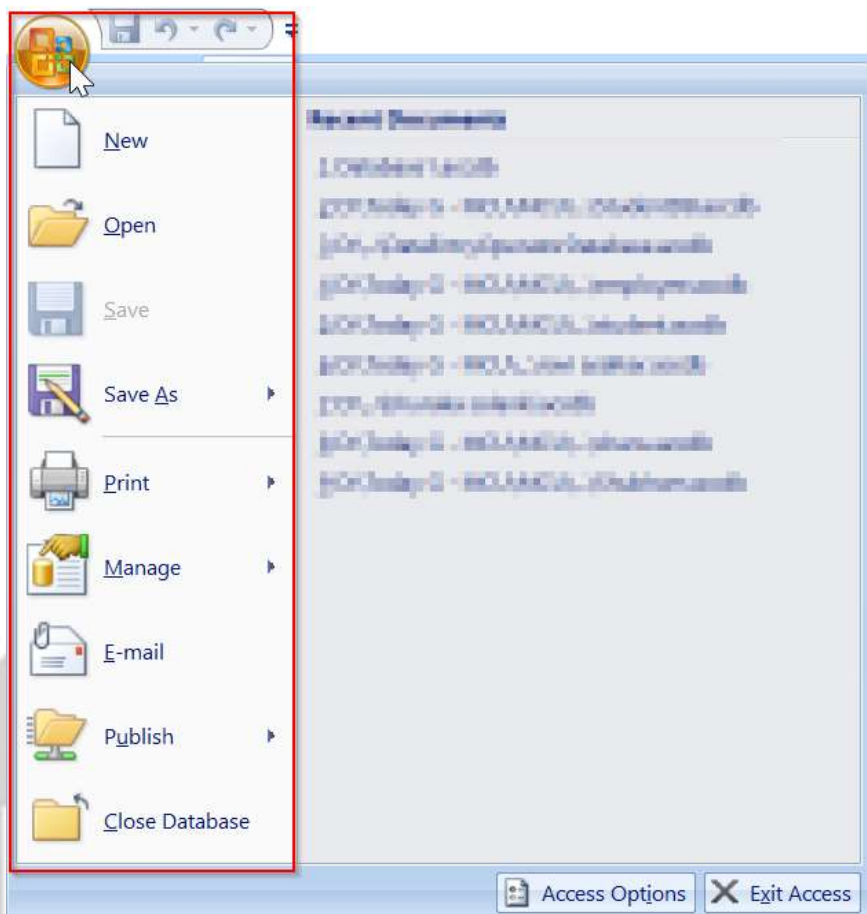
रिबन के ऊपर स्थित क्विक एक्सेस टूलबार, आपको सामान्य कमांड तक पहुंचने देता है इससे कोई फर्क नहीं पड़ता कि आप किस टैब पर हैं। डिफॉल्ट रूप से, यह Save, Undo और Repeat command दिखाता है। यदि आप चाहें, तो आप अतिरिक्त कमांड जोड़कर इसे कस्टमाइज कर सकते हैं।



Backstage view

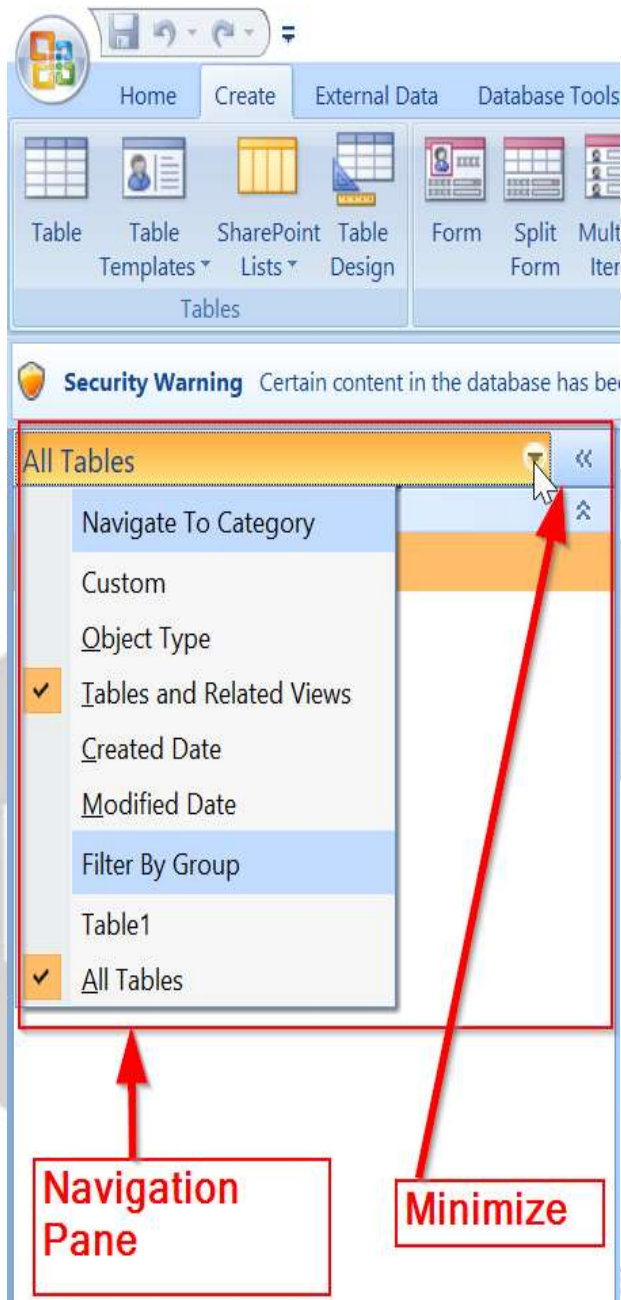
बैकस्टेज व्यू आपको अपने डेटाबेस को Save, Open और Print करने के लिए विभिन्न विकल्प देता है।

ऑफिस बटन पर क्लिक करें। Backstage view दिखाई देगा।



The Navigation Pane :

नेविगेशन पेन (फलक) एक list है जिसमें आपके डेटाबेस में प्रत्येक ऑब्जेक्ट होता है। आसान देखने के लिए, वस्तुओं को प्रकार के अनुसार समूह में व्यवस्थित किया जाता है। आप नेविगेशन पेन का उपयोग करके वस्तुओं को Open, Rename और Delete कर सकते हैं।



How to minimize and maximize the Navigation pane:

नेविगेशन पेन को मिनीमाइज कैसे करें :

नेविगेशन पेन (फलक) को आपके सभी ऑब्जेक्ट्स को प्रबंधित करने में आपकी सहायता के लिए डिज़ाइन किया गया है; हालांकि, अगर आपको लगता है कि यह आपकी स्क्रीन स्पेस को बहुत अधिक लेता है तो आप इसे Minimize (मिनीमाइज) कर सकते हैं।

नेविगेशन पेन को Minimize करने के लिए, ऊपरी-दाएं कोने में डबल तीर पर क्लिक करें।

नेविगेशन पेन को Minimize किया जाएगा। इसे Maximize (मैक्सिमाईज) करने के लिए दोबारा तीर पर फिर से क्लिक करें।



Introduction to MS Access Objects / एम.एस.एक्सेस ऑब्जेक्ट्स का परिचय :

Components of database management system / डेटाबेस मैनेजमेंट सिस्टम के घटक :

रिलेशनल डेटाबेस मैनेजमेंट सिस्टम के निम्नलिखित कंपोनेंट्स होते हैं जैसे टेबल, क्वेरी , फॉर्म , रिपोर्ट आदि । एम.एस-एक्सेस में इन्हें एक्सेस ऑब्जेक्ट कहा जाता है ।

1. **टेबल (Table) :** इस डेटाबेस में डाटा, टेबल्स (Tables) के रूप में संग्रहित और प्रदर्शित होता है। टेबल में रो (Row) और कॉलम (Column) होते हैं। एक रो में एक रिकॉर्ड संग्रहित किया जाता है। टेबल में एक रो एक एंटीटी (Entity) के रिकॉर्ड को दर्शाती है और कॉलम, एट्रीब्यूट्स (Attributes) / फील्ड्स (Fields) दर्शाते हैं। रिलेशनल डेटाबेस में टेबल को रिलेशन भी कहते हैं। एक रो और एक कॉलम जहां पर मिलते हैं उस बॉक्स को सेल (Cell) कहते हैं। सेल के अंदर डाटा प्रदर्शित किया जाता है। डेटाबेस का एक टेबल एक एंटीटी सेट (Entity Set) को प्रदर्शित करता है। टेबल की एक रो एक एंटीटी (Entity) या एक रिकॉर्ड को प्रदर्शित करती है। रिकॉर्ड को टपल भी कहा जाता है। टेबल का एक कॉलम, एक एट्रीब्यूट या एक फील्ड को प्रदर्शित करता है। एक डेटाबेस में कई टेबल हो सकते हैं। उदाहरण, किसी कॉलेज के सभी छात्रों की जानकारी का टेबल बनाना जिसमें छात्रों के रोल नंबर, नाम, कक्षा, मोबाइल नंबर, पूरा पता आदि हो।

Student				
rollno	sname	class	mobile	address
1	Raju Kumar Gu	DCA	9827000001	Khandwa
2	ShyamKumar	PGDCA	9827000002	Indore
3	Rahim Kumar	DCA	9827000002	Indore
4	Kishore Gupta	DCA	9827000003	Bhopal
5	Ramu	PGDCA	9827000004	Khandwa
6	Raj	PGDCA	9827000001	Khandwa
7	Ramu	DCA	9827000001	Khandwa
8	Raj	PGDCA	9827000004	Khandwa
9	Ravi Gupta	DCA	9827000003	Bhopal
10	Suresh	PGDCA	9827000003	Bhopal

2. **क्वेरी (Query) :** किसी टेबल या डेटाबेस से जरूरत के अनुसार डेटा निकालने के निर्देश (Command) को क्वेरी कहते हैं। डेटाबेस से पूछा गया प्रश्न क्वेरी कहलाता है। किसी क्वेरी के उत्तर में जो रिकॉर्ड डेटाबेस से निकाला जाता है उसे उस क्वेरी का डेटासेट या रिजल्टसेट (DataSet or ResultSet) कहते हैं। उदाहरण, इंदौर में रहने वाले सभी छात्रों के नाम और पिता के नाम की सूची निकालने की क्वेरी।

IndoreQueryWithJoin4

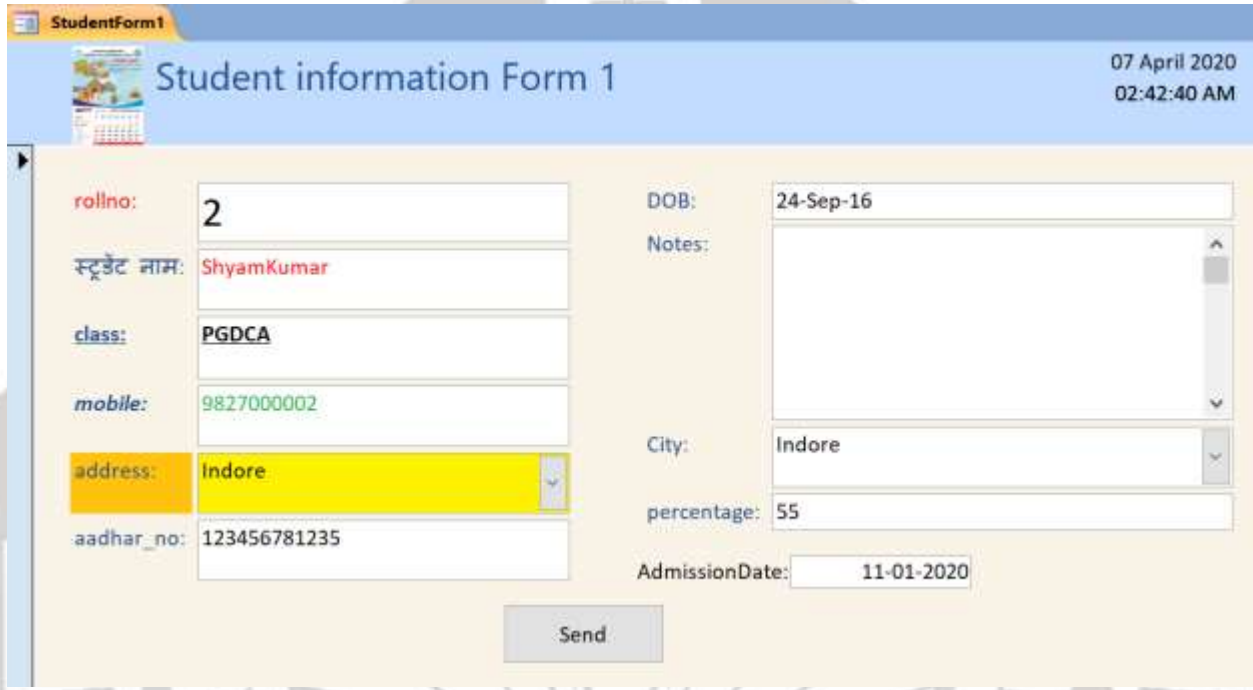
```

    erDiagram
        Student ||--}| StudentFamily : "1 to many"
        Student {
            string rollno PK
            string sname
            string class
            string mobile
            string address
            string aadhar_no
            string DOB
            string Notes
            string City
            string percentage
            string Gender
            string Email Id
            string AdmissionDate
        }
        StudentFamily {
            string rollno FK
            string fname
            string mname
            string bname
            string sisname
        }
  
```

Field:	sname	fname	address	
Table:	Student	StudentFamily	Student	
Sort:				
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteria:			"indore"	
or:				

3. **फॉर्म** : यद्यपि आप टेबल्स में डेटा दर्ज और संशोधित कर सकते, लेकिन टेबल्स में डेटा को दर्ज करना तथा संशोधन करना आसान नहीं होता है क्योंकि टेबल में हजारों रिकॉर्ड हो सकते हैं। इस समस्या को दूर करने के लिए, फॉर्म प्रस्तुत किए जाते हैं। टेबल्स की तरह फार्म में भी डेटा दर्ज किया जाता है। फार्म में दर्ज किया गया डेटा अंततः टेबल में जाकर ही संग्रहित होता है। फार्म में डेटा दर्ज, संशोधित और प्रदर्शित भी किया जाता है। फॉर्म को

आकर्षक भी बनाया जा सकता है और इसमें डाटा वेलिडेशन भी लगाया जा सकता है ताकि टेबल में गलत डाटा संग्रहित न हो पाए ।



The screenshot shows a web form titled "Student information Form 1" with a timestamp of "07 April 2020 02:42:40 AM". The form contains the following fields:

rollno:	2	DOB:	24-Sep-16
स्टूडेंट नाम:	ShyamKumar	Notes:	
class:	PGDCA	City:	Indore
mobile:	9827000002	percentage:	55
address:	Indore	AdmissionDate:	11-01-2020
aadhar_no:	123456781235		

A "Send" button is located at the bottom center of the form.

4. **रिपोर्ट्स** : क्वेरी द्वारा डेटाबेस या टेबल से लाए गये रिकॉर्ड को जब आप कागज पर प्रिंट करना चाहते हैं, तो उसे रिपोर्ट कहते हैं। जरूरत के अनुसार एक ही रिपोर्ट में कई टेबलों का डाटा रखा और प्रिंट किया जा सकता है । प्रिंट से पहले रिपोर्ट को आवश्यकता अनुसार फॉर्मेट भी किया जा सकता है और फ़ील्ड्स को व्यवस्थित भी किया जा सकता है । एक रिपोर्ट में एक या एक से ज्यादा पेज हो सकते हैं । डेटाबेस में संग्रहित विभिन्न प्रकार के डाटा का विश्लेषण करने के लिए भी रिपोर्ट का उपयोग किया जाता है ।

StudentForm2

Student

07 April 2020
02:43:37 AM

sname	Gender	rollno	DOB	class	mobile	address	aadhar_no	City
Ramu		7	14-Feb-19	DCA	9827000001	Khandwa	1234567812359	Khandwa
Raj	m	8	22-Aug-19	PGDCA	9827000004	Khandwa	1234567812359	Khandwa
Ravi Gupta	m	9	11-Sep-19	DCA	9827000003	Bhopal	1234567812357	Bhopal
Suresh	f	10	11-Sep-19	PGDCA	9827000003	Bhopal	1234567812357	Bhopal
Ramesh	f	11	11-Sep-19	DCA	9827000003	Bhopal	1234567812357	Bhopal
Uma	f	12	11-Sep-19	PGDCA	9827000003	Bhopal	1234567812357	Bhopal
Sudhir	M	13	24-Sep-16	PGDCA	9827000002	Indore	123456781235	Indore
Rani	M	14	24-Sep-16	DCA	9827000002	Indore	123456781235	Indore
Rajesh	m	15	22-Aug-19	PGDCA	9827000004	Khandwa	1234567812359	Khandwa
Raju Kumar Gupta	m	1	12-Jun-18	DCA	9827000001	Khandwa	123456781234	Khandwa

5. **मैक्रो (Macro)** : मैक्रो विभिन्न प्रकार के कार्यों को स्वचलित बनाने की सुविधा प्रदान करते हैं। आप मैक्रो का उपयोग किसी फॉर्म, रिपोर्ट या कंट्रोल में फंक्शनलिटी (Functionality) जोड़ने के लिए कर सकते हैं।

6. **मोड्यूल (Module)** : मैक्रो के समान ही मोड्यूल, विभिन्न प्रकार के कार्यों को स्वचलित बनाने की सुविधा प्रदान करते हैं। आप इनका उपयोग किसी फॉर्म, रिपोर्ट या कंट्रोल में फंक्शनलिटी (Functionality) जोड़ने के लिए कर सकते हैं। मैक्रो को बनाने के लिए मैक्रो एक्शन की सूची में से चुनाव करना होता है। जबकि मोड्यूल, विजुअल बेसिक फॉर एप्लीकेशन (Visual Basic for Applications) में लिखे जाते हैं।