

डेटाबेस, सूचनाओं (या डेटा) का एक ऐसा व्यवस्थित संग्रह (Organised Collection) होता है, जिससे हम किसी भी सूचना को सरलता से प्राप्त कर सकते हैं। डेटाबेस व्यवस्थित इसलिए होता है, क्योंकि इसमें किसी भी डेटा या सूचना को एक निश्चित स्थान पर पहले से तय किए हुए रूप में रखा जाता है, ताकि कभी भी आवश्यकता पड़ने पर उसे आसानी से ढूँढकर देखा जा सके।

व्यवस्थित डेटाबेस में हमें निम्नलिखित कार्य की सुविधा होती है

- आवश्यक सूचना को निकालना (Retrieving)।
- सूचनाओं के अनुसार उचित कार्यवाही करना या निर्णय लेना।
- सूचनाओं को नई आवश्यकताओं के अनुसार फिर से व्यवस्थित करना।
- सूचनाओं के आधार पर रिपोर्ट आदि बनाना तथा नई सूचनाएँ निकालना।

एक डेटाबेस, नामों की सूची की एक फाइल के रूप में आसान भी हो सकता है और डेटा की बहुत-सी फाइलों के समूह के रूप में कठिन भी हो सकता है।

डेटाबेस के प्रकार (Types of Database)

डेटाबेस मुख्य रूप से तीन प्रकार का होता है, जोकि निम्नलिखित हैं।

- 1. नेटवर्क डेटाबेस (Network Database)** इस प्रकार के डेटाबेस में, डेटा रिकॉर्ड के समूह के रूप में तथा डेटा के बीच सम्बन्ध लिंक के माध्यम से दर्शाया जाता है।
- 2. हैरार्किकल डेटाबेस (Hierarchical database)** इस प्रकार

उदाहरण के लिए एक कक्षा का औसत स्कोर एक सूचना है, जोकि उस कक्षा के विद्यार्थियों के स्कोर से निकाला जा सकता है।

संक्षेप में, डेटा डेटाबेस में स्टोर मूल्यों को सन्दर्भित (Refer) करता है, जबकि सूचना उन मूल्यों से निकाले गए निष्कर्ष या अर्थ को सन्दर्भित करती है।

कम्प्यूटरीकृत डेटाबेस की आवश्यकता

(Need Of Computerised Database)

हाथ से बनाए गए डेटाबेस (हस्तचालित डेटाबेस) में बहुत-सी समस्याएँ होती हैं; जैसे कि

1. नया डेटा जोड़ने की समस्या,
2. डेटा को बदलने की समस्या,
3. डेटा को अपनी शर्तों के अनुसार प्राप्त करने की समस्या आदि।

इन सभी समस्याओं को दूर करने के लिए कम्प्यूटरीकृत डेटाबेस का निर्माण किया गया। इसमें सभी सूचनाएँ कम्प्यूटर पर रखी जाती हैं और कम्प्यूटर की सहायता से ही उनका रख-रखाव तथा प्रोसेसिंग की जाती है। कम्प्यूटर पर डेटाबेस बनाने के कई प्रोसेसिंग की जाती है। कम्प्यूटर पर डेटाबेस बनाने के कई कारण हैं, जो निम्नलिखित हैं

1. कम्प्यूटर पर बड़े आकार का डेटाबेस सरलता से बनाया जा सकता है, क्योंकि उसमें डेटा को संग्रहीत करने की क्षमता अधिक होती है।
2. कम्प्यूटर की कार्य करने की गति तेज होने के कारण कितने भी बड़े डेटाबेस में, डेटा को वृक्ष के रूप में नोड्स के माध्यम से व्यवस्थित किया जाता है। हैरारिकल डेटाबेस में नोड्स आपस में लिंक के माध्यम से जुड़ी होती हैं।
3. **रिलेशनल डेटाबेस (Relational Database)** रिलेशनल डेटाबेस को **संरचित डेटाबेस (Structured Database)** भी कहा जाता है, जिसमें डेटा को सारणियों (Tables) के रूप में संग्रहीत (Store) किया जाता है। इन डेटा सारणियों में स्तम्भ (Column), सारणी

में स्टोर होने वाले डेटा के प्रकार को तथा पंक्तियाँ (Rows) डेटा को दर्शाती हैं।

डेटा (Data)

डेटाबेस में से कोई भी इच्छित सूचना निकालना और डेटाबेस पर विभिन्न क्रियाएँ करना आदि कार्य बहुत कम समय में ही सम्पन्न हो जाते हैं। इतना ही नहीं तेज गति के कारण उस पर कोई लम्बी-चौड़ी रिपोर्ट निकालना और छापना मिनटों का कार्य होता है।

3. इसमें हस्तचालित डेटाबेस की तुलना में बहुत कम खर्च आता है।

डेटाबेस के अवयव (Components of Database)

एक डेटाबेस विभिन्न प्रकार के अवयवों से मिलकर बना होता है। डेटाबेस का प्रत्येक अवयव आब्जेक्ट (Object) कहलाता है।

प्रत्येक डेटाबेस फाइल में आप अपने डेटा को विभिन्न **सारणियों (Tables)** में विभाजित कर सकते हैं; **फॉर्म (Form)** के माध्यम से सारणी के डेटा को देख किसी वस्तु, व्यक्ति या समूह के बारे में किसी तथ्य अथवा जानकारी को डेटा सकते हैं, नया डेटा जोड़ सकते हैं तथा अपडेट (Update) भी कर सकते हैं; (Data) कहा जाता है। किसी व्यक्ति का नाम, किसी वस्तु का वजन तथा मूल्य, **क्वैरी (Queries)** के माध्यम से आवश्यकतानुसार सारणियों में से डेटा को किसी कक्षा के विद्यार्थियों की उम्र आदि ये सभी डेटा के उदाहरण हैं। खोज सकते हैं। तथा पुनः प्राप्त कर सकते हैं और

रिपोर्ट (Report) के माध्यम

सूचना (Information)

से डेटा का विश्लेषण (Analyse) तथा डेटा को एक विशेष लेआउट में प्रिन्ट कर सकते हैं। (layout)

जब किसी डेटा को सार्थक तथा उपयोगी बनाने के लिए संसाधित,

संग्रहित किया जाता है, तो उसे हम सूचना कहते हैं।

व्यवस्थित, डेटाबेस के अवयवों का विस्तारपूर्वक वर्णन निम्नलिखित हैं

66

1. सारणी (Table)

वैसे तो डेटाबेस कई प्रकार के होते हैं, परन्तु सबसे अधिक प्रचलित और प्राकृतिक डेटाबेस रिलेशनल डेटाबेस हैं, जिसमें डेटा एक सारणी के रूप में संग्रहीत होता है। सारणी, स्तम्भ तथा पंक्तियों के कटाव से बने सैल (Cells) से मिलकर बनी होती है, यही सैल सारणियों में डेटा को स्टोर करने के लिए प्रयोग की जाती है।

इन सारणियों पर विभिन्न प्रकार के ऑपरेशन, जैसे कि डेटा को स्टोर करना, निस्पन्दन (Filtering) करना, पुनः प्राप्त करना, डेटा का सम्पादन करना आदि किए जा सकते हैं। मुख्य रूप से, सारणी फील्ड तथा रिकॉर्ड से मिलकर बनी होती है जिनका विवरण निम्नलिखित हैं

2. फील्ड (Field)

सारणी के प्रत्येक स्तम्भ को फील्ड कहते हैं, प्रत्येक फील्ड का एक

5. फॉर्म (Forms)

यद्यपि आप सारणी में डेटा को स्टोर कर सकते हैं। तथा सुधार भी सकते हैं, लेकिन सारणी में डेटा को स्टोर करना तथा सुधारना आसान नहीं होता है। इस समस्या को फॉर्म की सहायता से दूर कर सकते हैं।

फॉर्म आपकी स्क्रीन पर एक ऐसी विण्डो होती है, जिसकी सहायता से आप किसी सारणी में भरे गए डेटा को देख सकते हैं, सुधार सकते हैं और नया डेटा जोड़ भी सकते हैं, सामान्यतः फॉर्म एक समय पर एक

रिकॉर्ड को देखने तथा सुधारने के लिए प्रयोग किया जाता है।

6. रिपोर्ट (Reports)

सरल शब्दों में कोई रिपोर्ट एक ऐसा डायनासेट है, जिसे कागज पर छापा गया हो, आप किसी डायनासेट की सूचनाओं को किन्हीं आधारों पर समूहबद्ध कर सकते हैं।

निश्चित नाम होता है, जिसमें उसे पहचाना जाता है। प्रत्येक फील्ड का नाम उस फील्ड में स्टोर होने वाले डेटा के प्रकार को बताता है। उदाहरण के लिए विद्यार्थी का नाम, शहर, देश टेलीफोन नम्बर आदि फील्ड के नाम हो सकते हैं।

3. रिकॉर्ड (Record)

सारणी की प्रत्येक पंक्ति को रिकॉर्ड कहा जा सकता है। दूसरे शब्दों में, 'एक रिकॉर्ड एक एंटिटी' (जैसे कि वस्तु, व्यक्ति, आदि) से सम्बन्धित सभी फील्डों में उपस्थित डेटा का संग्रह होता है।

उदाहरण के लिए, आपके मित्रों के नाम तथा टेलीफोन नम्बर वाले डेटाबेस की संरचना नीचे दिखाए गए चित्र की तरह हो सकती है।

Table : Friends	
Name	Telephone No.
Rakesh	0123
Ridhi	4567
Hradesh	7869
Hari	4123
Kamal	8445

आप देख सकते हैं कि दी गई सारणी Friends में दो फील्ड हैं- Name, Telephone_No. और पाँच रिकॉर्ड हैं।

उदाहरण के लिए (Hradesh, 7869) एक रिकॉर्ड है।

4. क्वैरी (Queries)

किसी सारणी या डेटाबेस से आवश्यकतानुसार डेटा को निकालने के लिए जो आदेश दिया जाता है, उसे क्वैरी कहा जाता है।

उदाहरण के लिए, आप अगर मेरठ शहर में रहने वाले मित्रों की सूची

डेटाबेस के अनुप्रयोगी क्षेत्र

(Application Areas of Database)

डेटाबेस का उपयोग विभिन्न क्षेत्रों में किया जाता है। जिनमें से कुछ क्षेत्र निम्नलिखित हैं

- **बैंकिंग** के क्षेत्र में ग्राहकों की पर्सनल सूचना, उनके खातों की सूचना, लोन (Loans) आदि की सूचना रखने के लिए।
- **विश्वविद्यालयों** में विद्यार्थियों की सूचना, उनके अंक, कोर्स रजिस्ट्रेशन की सूचना आदि रखने के लिए।
- **एयरलाइन** (Airline) में रिजर्वेशन (Reservation) तथा कार्यक्रम की सूचना आदि के लिए।
- **क्रेडिट कार्ड** के लेन-देन में (Credit-card Transaction) क्रेडिट कार्ड के द्वारा खरीदारी तथा मासिक लेन-देन की रिपोर्ट तैयार करने के लिए।
- **संचार** के क्षेत्र में कॉल (Call) की मासिक रिकॉर्ड रखने के लिए, मासिक बिल बनाने के लिए।
- **विक्रय** (Sale) के क्षेत्र में ग्राहकों, उत्पादों तथा खरीदारी की सूचना रखने के लिए।
- **वित्तीय** (Finance) क्षेत्र में बिक्री तथा खरीद के बारे में जानकारी संग्रहीत करने के लिए।

- **एच आर (Human Resource)** क क्षेत्र म कमचारया, उनक वतन, टैक्स आदि के बारे में जानकारी संग्रहीत करने के लिए।

डेटाबेस प्रबन्धन प्रणाली

निकालनी चाहें, तो इसे एक क्वैरी कहेंगे। क्वैरी आपकी **(Database Management System)**

आवश्यकतानुसार डेटा को निकालने के लिए आवश्यक फील्डों, रॉयों, सारणी का नाम आदि को दर्शाता है। किसी क्वैरी के उत्तर में जो सूचनाएँ के सॉफ्टवेयर की आवश्यकता होती है, जिसे डेटाबेस प्रबन्धन प्रणाली या रिकॉर्ड डेटाबेस से निकाले जाते हैं, उसे उस क्वैरी का डायनासेट (DBMS) कहा जाता है। मुख्यतः यह एक कम्प्यूटर आधारित रिकॉर्ड के रख-रखाव की प्रणाली है अर्थात् यह एक ऐसी प्रणाली है, जिसका उद्देश्य रिकॉर्ड (Dynaset) कहते हैं।

67

एवं सूचनाओं को सम्भाल कर रखना है। यह उपयोगकर्ता को एक ऐसा वातावरण प्रदान करती है, जिसके माध्यम से डेटा को संग्रहीत करना तथा पुनः प्राप्त करना बहुत ही सुविधाजनक हो जाता है। MySQL, INGRES, MS-ACCESS आदि इसके उदाहरण हैं।

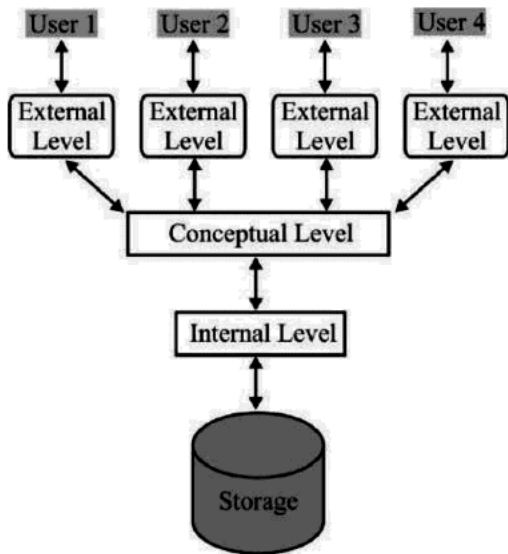
DBMS की मुख्य विशेषताएँ निम्नलिखित हैं

- डेटाबेस का निर्माण करना
- नए डेटा को शामिल करना या जोड़ना।
- वर्तमान डेटा को सम्पादित करना।
- डेटा को अस्थायी एवं स्थायी रूप से मिटाना।
- सूचना पद्धति को ढूँढना एवं प्राप्त करना।
- डेटा को क्रमबद्ध रूप से व्यवस्थित करना।
- आकर्षक एवं अर्थपूर्ण रिपोर्ट्स को डिजाइन करना एवं प्रिन्ट करना।

DBMS की संरचना (Architecture of DBMS)

DBMS की संरचना तीन स्तरों से मिलकर बनी होती है, जिनका विवरण निम्नलिखित हैं

1. आन्तरिक स्तर (Internal Level) इस स्तर में, डेटाबेस के भौतिक संग्रहण की संरचना का वर्णन करता है। यह वर्णित करता है कि वास्तव में डेटा डेटाबेस में कैसे संग्रहीत और व्यवस्थित होता है। वह यह भी निर्धारित करता है कि कौन-सी इण्डेक्सेस मौजूद हैं, स्टोर किए गए रिकॉर्ड किस क्रम में हैं आदि। इसे **भौतिक स्तर (Physical Level)** भी कहा जाता है।



DBMS की संरचना

2. विचार सम्बन्धी स्तर (Conceptual Level) इस स्तर में, पूर्ण डेटाबेस की संरचना होती है। यह स्तरों के मध्य जानकारी के रूपान्तरण की प्रक्रिया होती है। यह डेटाबेस में संग्रहीत डेटा के प्रकार को तथा डेटा के बीच सम्बन्ध को वर्णित करता है। इसे **तर्कसिद्ध स्तर (Logical Level)** भी कहा जाता है।

3. बाहरी स्तर (External Level) इस स्तर में डेटा व्यक्तिगत

उपयोगकर्ता द्वारा उपयोग में लाया जाता है। यह डेटाबेस के उस भाग का वर्णन करता है जो उपयोगकर्ता के लिए उपयोगी होता है। यह उपयोगकर्ताओं को उनकी आवश्यकतानुसार डेटा को एक्सेस (Access) करने की अनुमति इस प्रकार करता है, ताकि एक ही डेटा एक ही समय पर कई उपयोगकर्ताओं (Users) द्वारा प्रयोग किया जा सके। यह स्तर डेटाबेस की सूची को उपयोगकर्ता से छिपाता है। यह स्तर अलग-अलग उपयोगकर्ता के लिए अलग-अलग होता है। इसे **दर्शनीय स्तर (View Level)** भी कहा जाता है।

DBMS के लाभ (Advantages of DBMS)

DBMS के कई लाभ हैं जो निम्नलिखित हैं

- **डेटा के दोहराव में कमी (Reduction in Data Repetition)** अच्छी तरह व्यवस्थित किए गए डेटाबेस में सामान्यतः डेटा का कोई दोहराव नहीं होता। समस्त डेटा को एक जगह रखे जाने के कारण हर सूचना को केवल एक बार स्टोर किया जाता है।
- **डेटा की स्थिरता (Data Consistency)** डेटा के एक ही स्थान पर केन्द्रित होने के कारण डेटा की स्थिरता बनी रहती है, क्योंकि उसमें एक ही सूचना के दो मानों की सम्भावना समाप्त हो जाती है। डेटा अस्थिर तब होता है जब डेटा दो जगह रखा गया हो और केवल एक जगह सुधारा गया हो।
- **डेटा की साझेदारी (Data Sharing)** डेटा की साझेदारी करके एक समय पर कई प्रोग्राम डेटा का प्रयोग कर सकते हैं। जिससे प्रोग्रामों को अपना डेटाबेस तैयार करने की आवश्यकता नहीं होती और बहुत-सा समय और परिश्रम बच जाता है।
- **डेटा की सुरक्षा (Security of Data)** डेटाबेस प्रबन्धन प्रणाली (DBMS) डेटा को निषिद्ध उपयोगकर्ताओं तथा अवैध परिवर्तन से बचाता

है। यह केवल अधिकृत उपयोगकर्ताओं को डेटा का प्रयोग करने की अनुमति प्रदान करता है।

- **डेटा की सम्पूर्णता (Data Integrity)** डेटा की सम्पूर्णता, डेटा की समग्र पूर्णता (Overall Completeness), सटीकता (Accuracy) तथा निरन्तरता (Consistency) को सन्दर्भित करती है।

यह एक डेटा रिकॉर्ड के दो अपडेट्स (Updates) के बीच परिवर्तन के अभाव को दर्शाता है। यह दर्शाता है कि डेटाबेस में स्टोर डेटा बिल्कुल सही है और नवीनतम है।

DBMS की सीमाएँ (Limitation of DBMS)

DBMS के कई लाभ हैं, लेकिन साथ ही इसकी कुछ सीमाएँ भी हैं जो निम्नलिखित हैं

- **हार्डवेयर और सॉफ्टवेयर की लागत (Cost of Hardware and Software)** सॉफ्टवेयर को चलाने के लिए डेटा को तीव्र गति से प्रोसेस

68

करने वाले प्रोसेसर (Processor) और अधिक क्षमता वाली मेमोरी (Memory) की आवश्यकता होती है, जिनकी लागत अधिक होती हैं।

- **कठिनता (Complexity)** एक डेटाबेस प्रबन्धन प्रणाली (DBMS) के अच्छे कार्य करने की क्षमता की पूर्व-कल्पना करना उस DBMS सॉफ्टवेयर को कठिन बना देती है। डेटाबेस प्रबन्धन प्रणाली को समझने की विफलता एक संगठन Organisation के लिए गम्भीर परिणामों का कारण बन सकती है।

- **कर्मचारियों के प्रशिक्षण की लागत (Cost of Staff Training)** अधिकतर DBMS सॉफ्टवेयर अत्यन्त जटिल होते हैं, इसलिए उपयोगकर्ताओं को डेटाबेस का प्रयोग करने के लिए एक प्रशिक्षण देने

की आवश्यकता होती है। इस प्रकार, DBMS सॉफ्टवेयर चलाने के लिए संगठन को कर्मचारियों के प्रशिक्षण के लिए एक बड़ी राशि का भुगतान करना पड़ता है।

- **टेक्निकल स्टाफ की नियुक्ति (Appointing Technical Staff)**
एक संगठन में डेटाबेस के लिए प्रशिक्षित टेक्निकल पर्सन (Trained Technical Staff) जैसे कि डेटाबेस व्यवस्थापक (Database Administrator), एप्लीकेशन प्रोग्रामर (Application Programmers) आदि की आवश्यकता होती है, जिसके लिए संगठन को इन व्यक्तियों को एक अच्छे वेतन का भुगतान करना पड़ता है जिससे प्रणाली की लागत बढ़ जाती है।
- **डेटाबेस की विफलता (Database Failure)** अधिकांश संगठन में सभी डेटा एक ही डेटाबेस में एकीकृत होता है। यदि पावर बन्द हो जाने के कारण डेटाबेस विफल हो जाता है या डेटाबेस स्टोरेज डिवाइस पर ही विफल (Fail) हो जाता है। तो हमारा सभी मूल्यवान (Valuable) डेटा लुप्त (Loss) हो सकता है या हमारी पूरी प्रणाली बन्द हो सकती है।

रिलेशनल डेटाबेस (Relational Database)

रिलेशनल डेटाबेस में डेटा को द्वि-आयामी सारणियों (2-Dimensional Tables) के रूप में संग्रहीत किया जाता है। इन सारणियों को रिलेशन (Relation) भी कहा जाता है। रिलेशन डेटाबेस के रख-रखाव के लिए रिलेशनल डेटाबेस प्रबन्धन प्रणाली (Relational Database Management System-RDBMS) की आवश्यकता होती है। RDBMS, DBMS का ही एक प्रकार है। रिलेशनल डेटाबेस की मुख्य विशेषता यह है कि एक एकल डेटाबेस में एक से अधिक सारणियों को संग्रहीत किया जा सकता है। और ये सारणियाँ आपस में सम्बन्धित होती हैं।

सम्बन्धित पदावली (Related Terminology)

रिलेशनल डेटाबेस की कुछ सम्बन्धित पदावली निम्नलिखित हैं

- 1. रिलेशन (Relation)** रिलेशन के अन्तर्गत एक टेबल (Table) तैयार की जाती है जो एक सिक्वेन्शियल फाइल को निरूपित करती है, जिसमें टेबल की पंक्तियाँ (Rows) फाइल के रिकॉर्ड को इंगित करती हैं एवं स्तम्भ (Column) रिकॉर्ड के फील्ड को दर्शाता है। ये टेबल्स रिलेशन ही होते हैं। रिलेशन को उच्च स्तरीय फाइल्स के रूप में समझा जाता है, जिसमें
 - प्रत्येक रिलेशन में एक ही तरह के रिकॉर्ड होते हैं।
 - किसी दिए गए रिलेशन में प्रत्येक रिकॉर्ड के फील्डों की संख्या समान होती है।
 - प्रत्येक रिकॉर्ड का एक अलग पहचानने वाला (Identifier) होता है।
 - रिलेशन के अन्दर रिकॉर्ड किसी विशेष क्रम में व्यवस्थित होते हैं।
 इसके लिए निम्नलिखित उदाहरण पर विचार कीजिए

Relation : Part				
P#	P Name	Colour	Weight	City
P1	Nut	Red	12	London
P2	Bolt	Green	15	Paris
P3	Screw	Blue	18	Rome
P4	Screw	Red	14	London
P5	Carn	Blue	19	Paris

- 2. ट्यूपल (Tuple)** रिलेशन में प्रत्येक रिकॉर्ड को ट्यूपल कहा जाता है। उदाहरण के लिए, दिए गए रिलेशन Parts में पाँच ट्यूपल हैं। उनमें से एक ट्यूपल (P2, Bolt, Green, 15, Paris) है जो एक Part के विषय में एक विशेष सूचना है।
- 3. एट्रिब्यूट (Attribute)** रिलेशन के सन्दर्भ में प्रत्येक कॉलम

(फील्ड) को एट्रिब्यूट कहते हैं। उदाहरण के लिए, दिए गए रिलेशन Parts में पाँच एट्रिब्यूट्स (P#, P Name, Colour, Weight, City) हैं। जिनमें से प्रत्येक कॉलम एक Part के विषय सूचना प्रदान करता है।

4. **डोमेन (Domain)** रिलेशन के सन्दर्भ में डोमेन मानों का एक समूह होता है जिससे किसी कॉलम में दिए गए वास्तविक मानों को व्युत्पन्न किया जा सकता है।

उदाहरण के लिए, हम निम्न रिलेशन पर विचार कर सकते हैं।

Relation: S			
S#	S Name	Status	City
S1	Amar	30	Paris
S2	Mohan	20	New Delhi
S3	Ram	10	London

Relation: P			
P#	P Name	Status	Quality
P1	Nut	12	A
P2	Bolt	15	B
P3	Screw	25	C

Relation: SP		
P#	S #	Quantity
P1	S1	300
P2	S2	400
P3	S1	200
P2	S1	300
P1	S2	200

यहाँ SP टेबल के P# कॉलम में जो मान दिए गए हैं उन्हें P टेबल से व्युत्पन्न किया गया है एवं SP टेबल के S# कॉलम में जो मान दिए गए हैं उन्हें S टेबल से व्युत्पन्न किया गया है। अतः यहाँ टेबल P एवं टेबल S एक डोमेन के रूप में हैं, जिनसे P# एवं S# मानों को व्युत्पन्न कर एक SP टेबल तैयार किया गया है।

5. कार्डिनैलिटी (Cardinality) रिलेशन के सन्दर्भ में ट्यूपल (रिकॉर्ड्स) की कुल संख्या को कार्डिनैलिटी कहते हैं। अतः ऊपर वर्णित उदाहरण के लिए रिलेशन P की कार्डिनैलिटी 3, S की 3 एवं SP की 5 है।

6. डिग्री (Degree) रिलेशन के सन्दर्भ में एट्रिब्यूट (फील्ड या कॉलम) की कुल संख्या को रिलेशन की डिग्री कहते हैं। अतः ऊपर दिए गए उदाहरण में रिलेशन P की डिग्री 4, S की 4 एवं SP की 3 है।

की-फील्ड (Key-Field)

सामान्यतः किसी डेटाबेस के हर रिकॉर्ड को उसकी रिकॉर्ड संख्या द्वारा पहचाना जाता है, लेकिन सभी रिकॉर्डों की रिकॉर्ड संख्या को याद रखना सम्भव नहीं है। इसलिए किसी रिकॉर्ड को पहचानने के लिए हम उसके एक फील्ड को मुख्य फील्ड या की-फील्ड मान लेते हैं।

की (Key) कई प्रकार की होती है, जो निम्नलिखित हैं

(i) प्राइमरी की (Primary Key)

‘प्राइमरी की’ किसी रिलेशन का एक एट्रिब्यूट होता है, जिसमें विभिन्न मान होते हैं और जिनका प्रयोग उस रिलेशन के ट्यूपल को निर्धारित करने में किया जाता है। प्राइमरी की के रूप में चुना गया फील्ड NULL वैल्यू स्वीकार नहीं कर सकता।

उदाहरण के लिए, रिलेशन P के लिए एट्रिब्यूट P# की यह विशेषता है

कि प्रत्येक ट्यूपल में एक विशिष्ट P# मान होता है एवं यह मान उस ट्यूपल को उस रिलेशन के अन्य ट्यूपल से पृथक् करता है। इस स्थिति में रिलेशन P के लिए P# को एक प्राइमरी की कहा जाता है।

(ii) कैंडिडेट की (Candidate Key)

कभी-कभी ऐसे रिलेशन भी उत्पन्न होते हैं जिनमें एक से अधिक एट्रिब्यूट का समूह होता है जिसमें विशेष निर्धारक गुण होते हैं एवं इसके लिए एक से अधिक की (Key) का निर्माण करना आवश्यक हो जाता है। वह की जिस पर विशिष्ट निर्धारक गुण प्रयुक्त किए जाते हैं, कैंडिडेट की कहलाती है। किसी एक रिलेशन में एक या एक से अधिक कैंडिडेट की हो सकती है। दिए गए उदाहरण में रिलेशन S में एट्रिब्यूट S# एवं होता है एवं एक से अधिक कैंडिडेट की होते हैं। इस स्थिति में, किसी विशिष्ट गुण को निर्धारित करने के लिए जब एक से अधिक की (Key) के समूहों का उपयोग किया जाता है तब प्रथम की को प्राइमरी की एवं दूसरी की को आल्टरनेट की कहा जाता है। आल्टरनेट की को सेकण्डरी की भी कहा जाता है। ऊपर दिए गए उदाहरण में, S# एक प्राइमरी की है एवं SNAME एक आल्टरनेट की है।

(iv) फॉरेन की (Foreign Key)

किसी रिलेशनल डेटाबेस में, फॉरेन की एक या एक से अधिक फील्डों का समूह होता है जो दो सारणियों के डेटा के बीच लिंक (Link) प्रदान करता है। किसी सारणी की फॉरेन की फील्ड के लिए वैल्यू उसी सारणी के प्राइमरी की फील्ड या अन्य किसी सारणी के प्राइमरी-की फील्ड की वैल्यू से व्युत्पन्न (Derived) की जाती है। इस प्रकार फॉरेन की दो सारणियों के बीच सम्बन्ध स्थापित करती है। किसी सारणी में एक से अधिक फॉरेन की हो सकती हैं जो उस सारणी का अलग-अलग सारणियों से सम्बन्ध स्थापित करती है।

(v) यूनीक की (Unique Key)

किसी सारणी में यूनीक की एक या एक से अधिक फील्डों का समूह होती है जिनका उपयोग उस सारणी में प्रत्येक ट्यूपल को निर्धारित करने के लिए किया जाता है। एक सारणी में एक से अधिक यूनीक की हो सकती हैं। यूनीक की के लिए चुना गया फील्ड NULL वैल्यू स्वीकार कर सकता है।

डेटाबेस की भाषाएँ (Database Languages)

सैद्धान्तिक रूप से किसी दिए गए डेटा की उपभाषाएँ दो भाषाओं के समूह होती हैं

1. **डेटा डेफिनिशन लैंग्वेज (Data Definition Language-DDL)**
यह भाषा डेटाबेस ऑब्जेक्ट्स (Database objects) की विशेषताओं को परिभाषित करती है, इसका उपयोग डेटा स्ट्रक्चर, सारणी व्यू आदि को परिभाषित करने हेतु होता है।
2. **डेटा मैनिपुलेशन लैंग्वेज (Data Manipulation Language)** यह भाषा DDL के द्वारा परिभाषित ऑब्जेक्ट्स को मैनिपुलेट करती है या प्रोसेस करती है। इसका प्रयोग डेटा को जोड़ने मिटाने (Deletion), सुधारने (Modification) सारणी से सूचना को पुनः प्राप्त (Retrieve) करने के लिए होता है।

एंटिटी-रिलेशनशिप मॉडल (Entity-Relationship Model)

SNAME की यह विशेषता है- प्रत्येक ट्यूपल में एक विशिष्ट S# एवं Model)

SNAME मान है। जिसका उपयोग उस ट्यूपल को उस रिलेशन में एंटिटी-रिलेशनशिप मॉडल (E-R Mode) का प्रयोग डेटाबेस के सन्दर्भ में विद्यमान बाकी सभी ट्यूपल से पृथक् से करने में किया जाता है। एंटिटीज (Entities) तथा उनके बीच के सम्बन्ध को ग्राफिकल (Graphical)

(iii) आल्टरनेट की (Alternate Key)

रूप में प्रदर्शित करने के लिए किया जाता है। इसे

एंटीटी-रिलेशनशिप आल्टरनेट की वह होती है जो प्राइमरी की नहीं होती। इसकी उपयोगिता **डायग्राम (E-R Diagram)** भी कहा जाता है। E-R मॉडल से सम्बन्धित पदों

उस रिलेशन के लिए होती है जिसमें एक से अधिक एट्रिब्यूटों का समूह का विवरण निम्नलिखित है

70

1. एंटीटी (Entity)

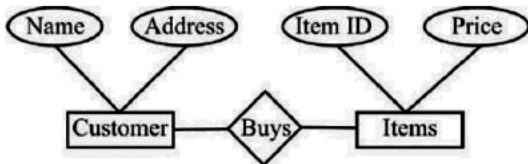
यह वास्तविक दुनिया की वस्तुओं को दर्शाती है। यह उन सभी वस्तुओं को सम्मिलित करती है जिनके बारे में डेटा एकत्रित किया जाना है, एंटीटी-रिलेशनशिप डायग्राम में इसे आयताकार बॉक्स के द्वारा दर्शाया जाता है। उदाहरण के लिए, Customer buys items, यहाँ पर Customer और items एंटीटी हैं।

2. एट्रिब्यूट्स (Attributes)

यह एक एंटीटी की विशेषताओं और गुणों का वर्णन करता है। सारणी में एट्रिब्यूट्स को फील्डों द्वारा दर्शाया जाता है। E-R डायग्राम में एट्रिब्यूट्स को दीर्घ वृत्ताकार बॉक्स में दर्शाया जाता है। उदाहरण के लिए, ItemID और Price एंटीटी ITEM के एट्रिब्यूट्स हो सकते हैं।

3. रिलेशनशिप (Relationship)

यह एंटीटीज के मध्य परस्पर सम्बन्धों को दर्शाता है। यह E-R डायग्राम में डायमण्ड की आकृति वाले बॉक्स के द्वारा दर्शाया जाता है। उदाहरण के लिए,



दिए गए चित्र में, Customer और Items एंटीटी है। यहाँ पर Name तथा Address, Customer के और ItemId तथा Price, Items के एट्रिब्यूट्स है, और Buys, customer तथा Items के बीच रिलेशनशिप को दर्शा रहा है।

एंटीटी सेट (Entity Set)

एक ही प्रकार की विशेषताओं या गुणों वाली एंटीटीज के सेट को एंटीटी सेट कहते हैं। उदाहरण के लिए, Students डेटाबेस में उपस्थित सभी Students एंटीटीज का एक एंटीटी सेट है।

एंटीटी सेट दो प्रकार के होते हैं

1. **स्ट्रॉन्ग एंटीटी सेट (Strong Entity Set)** ऐसा एंटीटी सेट, जिसमें प्राइमरी की फील्ड होता है, स्ट्रॉन्ग एंटीटी सेट कहलाता है।
2. **वीक एंटीटी सेट (Weak Entity Set)** ऐसा एंटीटी सेट, जिसमें प्राइमरी की बनाने के लिए पर्याप्त नहीं होते हैं, वीक एंटीटी सेट कहलाता है।

इन्हें भी जानें

✂ **डेटाबेस व्यवस्थापक (Database Administrator)** यह एक कम्प्यूटर में प्रशिक्षित व्यक्ति होता है जो डेटाबेस के उपयोग को अधिकृत करने के लिए, समन्वित करने के लिए और इसके उपयोग को मॉनीटर

(Monitor) करने के लिए जिम्मेदार होता है। यह सॉफ्टवेयर तथा हार्डवेयर को प्राप्त करने के लिए भी जिम्मेदार होता है। इसे DBA भी कहा जाता है।

स्कीमा (Schema) यह डेटाबेस की एक लॉजिकल संरचना (Logical Structure) है।

इन्स्टैंस (Instance) एक विशेष समय पर डेटाबेस में संग्रहीत जानकारी का संग्रह डेटाबेस का इन्स्टैंस कहलाता है।

डेटा माइनिंग (Data Mining) डेटा माइनिंग अलग-अलग दृष्टिकोण (Perspective) से डेटा का विश्लेषण करने और उपयोगी जानकारी में बदलने की प्रक्रिया है। कभी-कभी इसे Data or Knowledge Discovery भी कहा जाता है।

डॉ. ई एफ कॉड (Dr. E. F. Codd) ने वर्ष 1970 में रिलेशनल डेटाबेस प्रबन्धन प्रणाली के लिए 12 नियम प्रस्तुत किए थे।

डेटा मॉडल (Data Model) यह डेटा, डेटा के बीच सम्बन्ध, बाध्यताएँ (Constraints) सीमेण्टिक्स (Semantics) आदि का वर्णन करने के लिए वैचारिक उपकरणों (Conceptual Tools) का एक समूह होता है। सामान्यतः यह तीन प्रकार का होता है।

1. रिलेशनल डेटा मॉडल (Relational Data Model)
2. नेटवर्क डेटा मॉडल (Network Data Model)
3. हैरार्किकल डेटा मॉडल (Hierarchical Data Model)