1

डाटाबेस मेनेजमेंट सिस्टम
(Database Management System)

1.1 परिचय (Introduction)

डीबीएमएस एक ऐसा सिस्टम है, जो प्रयोगकर्ता (user) को डाटा access करने एवं डाटा को सूचना में बदलने में सहायता प्रदान करता है। यह सिस्टम उपयोगकर्ताओं को– डाटाबेस बनाने, अपडेट करने एवं सूचना प्राप्त करने की सहमति प्रदान करता है। डीबीएमएस समस्त सम्बन्धित रिकॉर्ड्स का समूह एवं प्रोग्राम है, जिसके द्वारा रिकॉर्ड access किए जाते हैं। डीबीएमएस कम्प्यूटर पर आधारित रिकॉर्ड व्यवस्था को पूरी तरह से संभालने (maintain) एवं access करने के लिये है।

डीबीएमएस का प्राथमिक लक्ष्य ऐसा वातावरण प्रदान करना है, जो कि सुविधाजनक एवं डाटाबेस सूचना को संग्रह करने एवं Retrieve करने के योग्य हो।

डाटाबेस बहुत बडे समूह की समस्त सूचना को व्यवस्थित करने के लिए डिजाईन की गई प्रणाली है। डाटाबेस प्रणाली की व्यवस्था द्वारा समस्त संरचनाओं की परिभाषा अपने साथ रखती है ताकि समस्त सूचनाओं को स्टोर या संग्रह किया जा सके। इसके अतिरिक्त डाटाबेस सिस्टम संग्रहित सूचना को सिस्टम क्रेश, बिना अनुमति के उपयोगकर्ताओं इत्यादि से सुरक्षा प्रदान करता है।

डीबीएमएस की विशेषता एवं उत्तरदायित्व

- डाटा की सुलभ उपलब्धता
  - डाटाबेस में लाखों रिकॉर्डस स्टोर होते हैं।
- नियन्त्रित एवं सुलभ डाटा की उपलब्धता
  - कौन डाटा देख सकता है यह निश्चित करना।
  - किस उपयोगकर्ता के लिए डाटा को पढ़ने/लिखने (Read/write) करने की अनुमति है।
- एक साथ एक ही समय डाटा को उपलब्ध कराना
  - इसमें एक साथ एक ही समय में एक से अधिक उपयोगकर्ता डाटाबेस पर कार्य कर सकते हैं।
  - जब आप रिकॉर्ड में कोई परिवर्तन कर रहे हो तो रिकॉर्ड को Lock व Save किया जा सकता है।
  - डाटाबेस के माध्यम से क्रॉस सेक्शन द्वारा किसी निश्चित कार्य को किया जाना ।

- ■ बाद के यूजर्स भी डाटा को अपनी आवश्यकतानुसार देख सकते हैं।

- डीबीएमएस डाटाबेस रिकवरी में भी सहायक है

डाटा के खराब अथवा फेल होने पर पुनः निर्माण की तकनीक से सुधारा जा सकता है, जैसे– disk crash, program error, power outage, fire आदि।

अवधारणा रूप से ऐसा होता है–

1. उपयोगकर्ता द्वारा डाटाबेस को access करने के लिए DML भाषा का प्रयोग किया जाता है।
2. डीबीएमएस आग्रह में यदि कोई अवरोध (Interrupt) की आवश्यकता होती है तो डीवीएमएस कार्य रोककर उस अवरोध की व्याख्या करता है।
3. डीबीएमएस बाहरी डिजाईन को चैक करता है, मैपिंग करता है और स्ट्रक्चर के डिजाईन को स्टोर करता है।
4. डीबीएमएस संग्रहित डाटाबेस में चाहे गये आवश्यक ऑपरेशन करता है।

1.2 डाटाबेस के लाभः

1. एक जैसे रिकॉर्ड को कम करना।
2. डाटा की एकात्मकता एवं गुणवत्ता को बनाये रखना।
3. डाटा की असंगति (Inconsistency) को हटाना।
4. डाटा को सुरक्षित रखने के लिये आवश्यक प्रतिबन्ध लगाना।
5. सॉफ्टवेयर डवलपमेन्ट का खर्च कम करना।

1.3 डाटाबेस के व्यू (View of Database):

डाटाबेस उपयोगकर्ताओं, प्रोग्रामरों और डाटाबेस प्रशासक (Administrator) के समक्ष कई तरह के view देता है।

- Internal Data Representation यह यूजर या Application Programmer द्वारा देखा नहीं जा सकता।
- अवधारणा (Conceptual) Schema या conceptual view ही एक मुख्य माध्यम है, जिसके द्वारा DBA (Data Base Administrator) डाटाबेस को बनाते है एवं देख–रेख करते है।
- डीबीएमएस द्वारा external schema के कई view, प्रोग्रामर व यूजर को उपलब्ध होते हैं जो कि Application की आवश्यकता के अनुसार होते हैं।

1.4 डाटा के व्यू (Views of Data):

(i) इन्टरनल/फिजिकल स्तर व्यू (Internal / Physical Level View)

यह डाटा के भौतिक रूप से संग्रहण (physically storage) की व्याख्या करता है व उनके आपसी relations को दर्शाता है और सम्पूर्ण डाटाबेस को छोटे और अपेक्षाकृत साधारण संरचना में परिभाषित किया जा सकता है।

(ii) अवधारणा/लॉजिकल लेवल व्यू (Conceptual / Logical Level View)

यह स्तर Database Administrators के द्वारा प्रयोग किया जाता है जो यह निर्णय लेता है

कि कौन–कौनसी सूचना डाटाबेस में रखनी है। प्रत्येक डाटाबेस का एक ही conceptual view होता है और इसके द्वारा यह जांच की जाती है कि डाटा consistency एवम् integrity नियमों का ध्यान रखे हुये है।

(iii) एक्सटरनल/व्यू लेवल (External / View Level)

यह सबसे ऊपरी स्तर होता है जिसमें डाटाबेस से सम्बन्धित उन भागों को शामिल किया जाता है जो कि उपयोगकर्ता से सम्बन्धित होते हैं। डीबीएमएस एक database के लिए अनेक views प्रदत्त कर सकता है।

1.5 डाटा मॉडल (Data Model):

(i) ऑब्जेक्ट आधारित लॉजिकल मॉडल (Object Based Logical Model)

यह डाटा को logical एवम view level पर व्याख्या करने हेतु प्रयोग किया जाता है। ये लचीले एवम् स्पष्ट structure capabilities उपलब्ध कराते हैं और data constraints को अलग से परिभाषित करते हैं।

- एन्टिटी रिलेशनशिप मॉडल (The Entity relationship model)
- ऑब्जक्ट ऑरियन्टेड मॉडल (The object Oriented model)
- सीमान्टिक डाटा मॉडल (The semantic data model)
- फंक्शनल डाटा मॉडल (The functional data model)

(ii) रिकॉर्ड आधारित लॉजिकल मॉडल (Record Based Logical Model)

Hierarchical Model, file पर आधारित System है और Network Model, Hierarchical Model का Superset है। Relational Model, Relation की परिकल्पना पर आधारित है। यह data को logical और view level पर दर्शाता है–

रिलेशनल मॉडल (Relational Model) : इस मॉडल में डाटा टेबलों में संग्रहित किया जाता है। टेबल पंक्ति एवं कॉलम का समूह होती है। पंक्ति को टपल कहते हैं एवं कॉलम को एट्रीब्यूट कहते हैं।

नेटवर्क मॉडल (Network Model): इस मॉडल में डाटा को एक से अधिक रिलेशनशिप द्वारा पेरेन्ट एवं चाइल्ड फॉर्म में संग्रहित किया जाता है। इसमें जल्दी एवं सीधा डाटा उपलब्ध हो जाता है क्योंकि डाटा को प्राप्त करने के एक से अधिक रास्ते (Path) होते हैं।

हायरारकीकल मॉडल (Hierarchical Model): इस मॉडल में डाटा को ट्री–स्ट्रक्चर में रखा जाता है। इसमें रिकॉर्ड नोड्स एवं ब्रान्चेज के रूप में संग्रहित किये जाते हैं।

(iii) फिजिकल डाटा मॉडल (Physical Data Model)

यह डाटा को lowest level पर दर्शाने हेतु प्रयोग होता है।

1.6 आरडीबीएमएस का परिचय (Introduction of RDBMS):

आरडीबीएमएस में डाटा टेबलों में संग्रहित होता है जो कि पंक्ति एवं कॉलम की बनी हुई होती हैं। पंक्ति को टपल द्वारा एवं कॉलम को एट्रीब्यूट द्वारा इंगित किया जाता है। डोमैन(Domain), कॉलम में संग्रहित होने वाले मान का वास्तविक मान है जो कि किसी कॉलम में दर्ज होता है।

1.6.1 आरडीबीएमएस की विशेषता (Properties of RDBMS):

(i) डाटा को टेबल की फोर्म में प्रदर्शित किया जाता है।

(ii) उपयोगकर्ता को यह समझना आवश्यक नहीं होता है कि डाटा कैसे संग्रहित किया जायेगा।

(iiii) सिस्टम टेबल, रखी जाने वाली सूचना की ऑनलाइन की जानकारी देता है।

(iv) यह नल (NULL) मान की परिकल्पना (Concept) को Support करता है।

(v) पंक्ति कहाँ स्थित है इसका आरडीबीएमएस के लिए महत्व नहीं है।

(vi) किसी भी पंक्ति में प्राईमरी की का मान अद्वितीय होना चाहिए।

(vii) कॉलम का नाम अद्वितीय होना चाहिए।

(viii) आरडीबीएमएस में टेबलों के मध्य कोई हार्डकोड रिलेशनशिप नहीं होती है।

(ix) डाटा जो पंक्ति एवं कॉलम में, टेबल में संग्रहित किया जाता है इसे रिलेशन कहते हैं।

(x) पंक्ति एवं कॉलम का इन्टरसैक्शन आरडीबीएमएस में एक मान देता है।

1.6.2 आरडीबीएमएस के घटक (Components of RDBMS):

डाटा (Data): डाटा, डाटाबेस में संग्रहित (stored) किया जाता है जिसमें अंकगणितीय (numerical data), अगणितीय (non-numerical data) जैसे कि chraracters (alphabetic and numeric characters), तिथि (date) और तर्क सम्बन्धी (logical data) प्रकार के होते हैं। ज्यादा उन्नत सिस्टम (advanced system) में ज्यादा कठिन data entities शामिल करते हैं, जो कि picture और image के रूप में होते हैं।

डाटाबेस (Database): यह डाटा का समूह होता है जो कि टेबलों में संग्रहित होता है और यह पंक्ति एवं कॉलम की फॉरमेट में होता है। डाटा जो टेबल में संग्रहित होता है वह एक दूसरे से सम्बन्धित होता है।

स्टैन्डर्ड ऑपरेशन (Standard Operations): ये ऑपरेशन (operation), उपयोगकर्ता (user) को data के manipulation हेतु प्रारम्भिक संभावनाएं उपलब्ध कराते हैं। record को क्रम में व्यवस्थित (sorting) करना, हटाना (removing) और चयन (selecting) करना standard operations के उदाहरण हैं।

प्रोग्रामिंग टूल्स (Programming Tools): Command & query के अतिरिक्त conventional programming language में function calls (subroutine calls) की सहायता से application programms में सीधे तौर पर database accessible होना चाहिये।

फाइल स्ट्रक्चर (File Structure) : प्रत्येक डीबीएमएस में data को सुव्यवस्थित करने हेतु स्वयं का internal structure होता है। इसके लिए कुछ common data models सामान्य रूप से डीबीएमएस द्वारा प्रयोग किये जाते हैं।

डाटा डेफिनिशन लैंग्वेज (Data Definition Langague (DDL)) :

डीडीएल एक ऐसी भाषा है जो कि डाटाबेस के आब्जेक्ट का वर्णन करती है। इसके द्वारा फील्डनेम, डाटा टाइप, डाटाबेस संग्रहित करने का स्थान एवं अनुबन्धों के वर्णन के साथ साथ, डाटाबेस के ऑब्जेक्ट्स जैसे कि टेबल, व्यू, इन्डेक्सेज आदि को क्रियेट एवं ड्रॉप किया जाता है।

डाटा मेनिपुलेशन लैंग्वेज (Data Manipulation Language (DML)):

जिस भाषा के द्वारा data को manipulate किया जाए। उसे DML कहा जाता है, इसके द्वारा निम्न कार्य किये जाते है।

- डाटाबेस में संग्रहित सूचना का retrieval.
- डाटाबेस में नयी सूचनाओं का insertion.
- डाटाबेस में से सूचनाओं को delete करना
- डाटाबेस में संग्रहित सूचनाओं को modify करना।

DML एक ऐसी भाषा है, जिसे काम में लेकर user द्वारा data access व manipulate किया जा सकता है। यह दो तरीके की हैः–

1. Procedural DML : उपयोगकर्ता द्वारा यह दर्शाना आवश्यक है, "किस तरह का data , व कैसे" प्राप्त किया जा सकता है।

2. Nonprocedural DML: उपयोगकर्ता द्वारा यह दर्शाना आवश्यक है, "कैसा data चाहिए" बिना बताये कि उस data को कैसे प्राप्त करें।

Non-procedural DMLs, Procedural DMLs की तुलना में याद करने में ज्यादा सरल हैं, क्योंकि उपयोगकर्ता को specify नहीं करना पड़ता "कैसे डाटा प्राप्त किया जाए"। इस language से एक ऐसा code उत्पन्न होता है जो उतना सक्षम नहीं होता है जैसा कि procedural language द्वारा उत्पन्न होता है।

रिलेशन्स / टेबल्स (Relations / Tables):

Relational model में data, relation प्रारूप में संग्रहित किया जाता है। रिलेशन का औपचारिक नाम Tables है। Table में database इसकी contents को पहचानने (identify) के लिए एक unique name रखता है। प्रत्येक table में कॉलम व पंक्तियाँ (rows and columns) होती है।

टेबल (Table): Table एक प्राथमिक इकाई (primary unit) है, जो कि data का संग्रह database में करती है। जब कोई भी user database को access करता है, तब table का reference वांछित data पाने के लिए किया जाता है। Multiple tables में डाटा होने पर डाटा प्राप्त करने के लिये टेबल्स के मध्य रिलेशनशिप भी स्थापित किया जा सकता है। table के लिए database में physical storage वाछंनीय है।

टपल्स (Tuples): प्रत्येक relation/ tables बहुत सारे tuples से बनी होती है। इन्हें record भी कहा जाता है। यह पंक्तियां (rows) होती है, जिनसे table बनती है।

Domain: ये एक ऐसी values का समूह है, जिसके द्वारा एक या ज्यादा column, attribute / field अपनी सही values को प्राप्त कर सकते हैं। दूसरे शब्दों में यह एक फील्ड (field) का नाम है।

डाटाबेस स्कीमा (Database Schema): Schema साधारण तौर पर कई जुड़े हुए ऑब्जेक्ट्स का समूह है, जो database में होता है। Schema के अन्तर्गत जो भी ऑब्जेक्ट जुड़ा हुआ होता है, उनकी आपस में relationship भी होती है। Schema का सिर्फ एक ही owner होता है।

आर.डी.बी.एम.एस. के उपयोगकर्ताः– आर.डी.बी.एम.एस. के उपयोगकर्ताओं को मुख्यतः चार भागों में विभाजित किया गया है।

1. डाटाबेस डिजाइनरः– यह डाटाबेस की स्ट्रक्चर का डिजाइन करता है एवं यह सुनिश्चित करता है कि किस प्रकार का डाटा डाटाबेस में संग्रहित किया जाना है।
2. डाटाबेस एडमिनिस्ट्रेटरः– यह डीबीए कहलाता है तथा यह डाटाबेस के समस्त कार्यों को मॉनीटर

करता है एवं यह सुनिश्चित करता है कि यह अच्छी तरह कार्य करें।

3. एप्लीकेशन डवलपरः— इनके द्वारा सामान्यतः प्रोग्राम लिखे जाते हैं जो कि डाटाबेस को एक्सेस करने के लिए काम में आते हैं।

4. अन्तिम उपयोगकर्ता (End User):— इनके द्वारा डाटा एन्ट्री एवं मैनीपुलेशन का कार्य किया जाता हैं।

1.7 डाटा टाइप (Data Types):

Data type के माध्यम से यह निश्चित किया जाता है कि database के कॉलम्स में किस तरह का डाटा रखा जाना है। हालांकि कई तरह के data type उपलब्ध हैं, लेकिन ज्यादातर काम में लिए जाने वाले 3 मुख्य हैं, जो कि निम्न हैं–

- अल्फा न्यूमेरिक (Alphanumeric)
- न्यूमेरिक (Numeric)
- डेट एण्ड टाईम (Date and time)

Alphanumeric data का इस्तेमाल character, number, special character आदि का संग्रह करने के लिए होता है। alphanumeric type में अगर numeric value का संग्रह होता है तब उस value को character की तरह माना जाता है, न कि नम्बर की तरह। Numeric data types में सिर्फ numeric values का ही संग्रह किया जाता है।

Date and time, data types का इस्तेमाल date and time values का संग्रह करने के लिए किया जाता है जो कि सामान्यतया अलग अलग आरडीबीएमएस के लिये बदलती है।

प्राईमरी की, फोरेन की, कैन्डीडेट की

उदाहरणः— नीचे तीन टेबल दर्शायी गयी है जिनके द्वारा प्राइमरी की, फोरेन की एवं कैन्डीडेट की को समझाया गया है।

REGISTRATION DETAIL

REGISTRATION_NO	STUDENT_NAME	CLASS	DOB	PLACE
101	ALPHA	X	10/10/1965	JAIPUR
102	BETA	II	07/01/1995	DELHI
103	GAMA	III	01/01/1993	JAIPUR
104	ARUN	IV	01/10/1990	JAIPUR
105	VARUN	V	05/08/1990	DELHI

BOOK DETAIL REGISTER

BOOKNO	BOOKNAME	AUTHOR	COST	EDITION
1	SQL		500	II
2	DBMS	KORTH	485	I
3	MSACCESS	-	400	I
4	DBMS	-	300	II
5	VB	STERLING	200	III

BOOK ISSUE REGISTER

REGISTRATION_NO	BOOKNO	ISSUEDATE	DEPOSITDATE
101	1	01/01/2006	
102	3	31/12/2005	
103	5	25/12/2005	01/01/2006
104	5	02/01/2006	

प्राईमरी की– प्राईमरी की, प्रत्येक रिकोर्ड में एक या एक से अधिक फील्ड का समूह होता है जो कि प्रत्येक रिकोर्ड को अलग से प्रदर्शित करता है तथा उनकी वेल्यू से डाटा का रिट्रीवल व अपडेटिंग शीघ्र संभव हो जाता है। यह डाटा के मान की एक से अधिक प्रविष्टि को रोकती है। प्राईमरी की का मान नल नहीं होनी चाहिए। उपरोक्त वर्णित उदाहरण में Registration table में REGISTRATION_NO प्राइमरी की है जो कि अद्धितीय है। BOOK DETAIL REGISTER में BOOKNO प्राइमरी की है जो कि अद्धितीय है।

फोरेन की– फोरेन की, किन्हीं दो टेबल्स में संबंध स्थापित करने के लिए होती है। दोनों ही टेबल्स में कम से कम एक फील्ड कॉमन होना आवश्यक होता है। सामान्यतः एक टेबल में यह प्राईमरी की कहलाती है एवं दूसरी टेबल में यह फोरेन की कहलाती है। फोरेन की, की के मान बार बार (Repeat) हो सकते है। फोरन की का उपयोग सामान्यतः संदर्भित टेबल के लिए किया जाता है और यह डाटा डोमेन इन्टेग्रिटी सुनिश्चित करती है। उपरोक्त वर्णित उदाहरण में BOOK DETAIL REGISTER **में** REGISTRATION_NO **फोरेन की है जो कि रजिस्ट्रेशन टेबल में प्राइमरी की है।**

कैन्डीडेट की (Candidate Keys) : candidate key एक minimal super key है। Minimal का मतलब यह होता है कि यदि पूरे set में किसी एक attribute हटाया जाए तो फिर ये super key नहीं रहती। यदि एक attribute को जोड़ें तो यह Minimal नहीं रहती है। (पर फिर भी यह super key है)। 'Candidate' शब्द का मतलब साधारण तौर पर यह है कि attributes के set को primary key की तरह इस्तेमाल कर सकते हैं, क्योंकि हमें आवश्यकता से ज्यादा attribute की जरूरत नहीं होती। किसी भी रिलेशन में एक से अधिक कैन्डीडेट की हो सकती है।

1.8 डाटा इन्टेग्रिटी (Data Integrity):

परिचयः– system integrity का अर्थ है कि database में डाटा सही एकात्मक व सुसंगत होना चाहिए। विभिन्न तरीके की integrity निम्न लिखित है–

- एन्टिरी इन्टेग्रिटी (Entity Integrity)
- रैफ्रेन्सियल इन्टेग्रिटी (Referential Integrity)
- डोमैन इन्टेग्रिटी (Domain Integrity)
- उपयोगकर्ता द्वारा परिभाषित इन्टेग्रिटी (User Defined Integrity)

उदाहरण

PARTS RELATION

P#	Pname	P-colour	P-wt	P-city
P1	nut	Red	12	London
P2	bolt	Green	17	Paris
P3	screw	Blue	17	Rome
P4	screw	Red	14	London

SHIPMENTS RELATION

S#	Part#	qty
S1	P1	300
S1	P2	200
S1	P3	400
S2	P1	300
S2	P2	400
S3	P3	200

यहां पार्टस रिलेशन में P# और pname, candidate key हैं, P# और S# primary key के उदाहरण पार्टस एवं शिपमैन्ट रिलेशन में है, P#, Shipment relation में foreign key का उदाहरण है।

(i) एन्टिटी इन्टेग्रिटी (Entity Integrity): Entity Ingerity नियम, Referential Integrity नियम के साथ मिलकर, Relational Model पर लागू होते हैं।

Entity Integrity Rule कहता है कि कोई भी attribute जो कि primary key का भाग है, उसमें null values नहीं आ सकती, क्योंकि नल वैल्यू की पहचान नहीं कर सकते इसके पीछे मुख्य कारण यह है कि उसकी सूचनाओं को भी हम recrod नहीं कर सकते हैं। इस वास्तविक दुनिया में Entity की किसी भी माध्यम से पहचान हो सकती है। Relational Model में Primary Key के द्वारा ही पहचानने का कार्य किया जाता है। अगर किसी भी part relation के part number में null values को स्वीकृति दी जाती है, तो यह कहा जायेगा कि या तो उस भाग की पहचान नहीं है, या खो रही है, लेकिन परिभाषा के तौर पर किसी भी Entity की कोई न कोई पहचान जरूर होनी चाहिये।

```
CREATE TABLE Customer
(Customer_No        number (10)       NOT NULL,
Name                varchar2(20),
Address             varchar2(20),
Depot_no            number (10),
Credit_limit        number (10) );
```

(ii) रैफ्रेन्सियल इन्टेग्रिटी (Referential Integrity):

Referential Integrity के नियम हैं

Database में unmatched foreign key values नहीं होनी चाहिये।

दूसरे तरीके से एक foreign key F1, base relation R1 में जो कि match करनी चाहिए है primary key P2 के base relation , R2 में दोनों में से या तो

a) R2 के one tuple के लिए P2 की Value के बराबर होनी चाहिए।

b) पूर्ण रूप से निरर्थक (Null) (i.e. अगर F1, one attribute से ज्यादा का बना हुआ है तो वे सभी निरर्थक values है)।

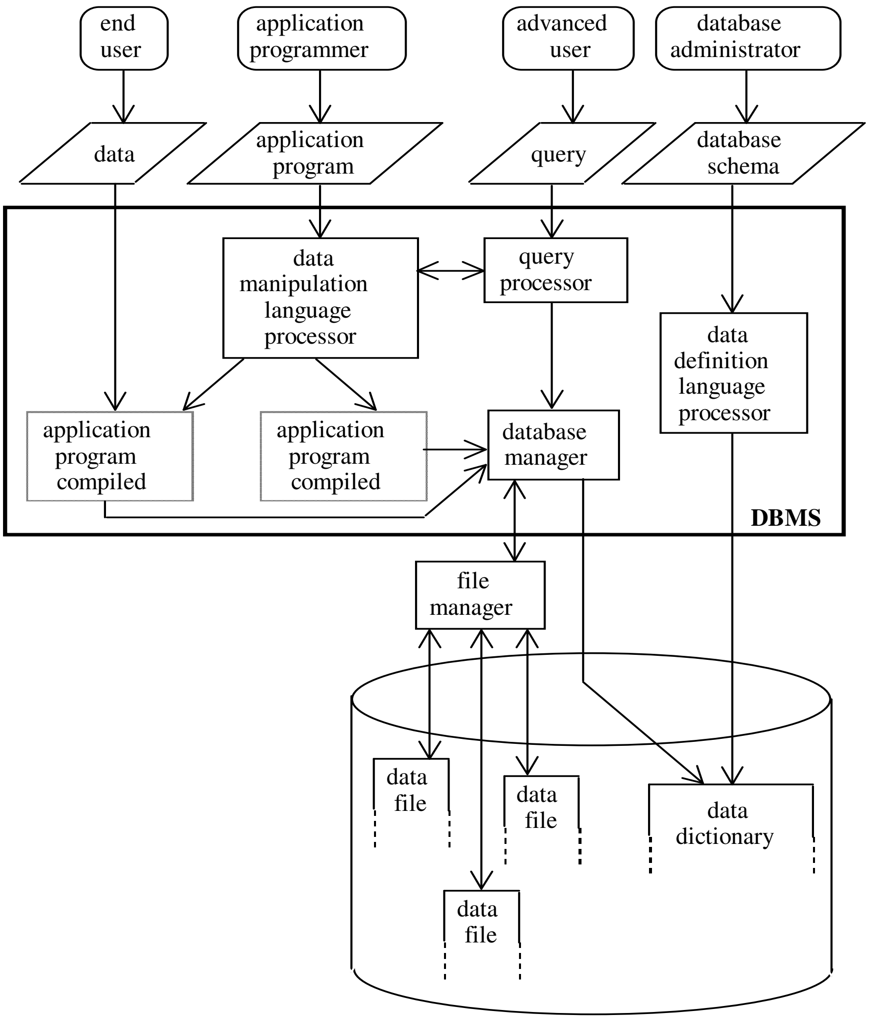
अगर ऊपर दी गयी parts एवं shipments की table देखेंगे तो part P2 जो कि part table में है उसको डिलीट करना असंभव है यदि P2 को shipment table में छोड दिया जावे तो।

(iii) डोमेन इन्टिग्रेटी (Domain Integrity):

- Domain integrity में आवश्यक है कि डाटा का मान कालम में दिये गये क्षेत्र के अन्दर ही होना चाहिये।
- उदाहरण: Domain integrity के द्वारा यह तय किया जाता है कि 'age' क्षेत्र में जो entry है वह integer है एवं " 0एवं 100" values के मध्य की है।
- Data types broad categories में क्षेत्र को सीमित कर देती है। (e.g. integers)
- Default एक ऐसी value की परिभाषा है जिसे column में स्वत् ही भरा जा सकता है, Rule ऐसे अपनाए जाने वाली values की परिभाषा है, जिन्हें भी column में डाला सकता है।
- Rules एवं defaults भी constraints की भांति ही है, लेकिन ANSI स्तर के नहीं है, इनके निरन्तर प्रयोग को बढ़ावा नहीं दिया जाता है।

(iv) उपयोगकर्ता द्वारा परिभाषित इन्टेग्रिटी (User Defined Integrity):इस प्रकार की इन्टेग्रिटी उपयोगकर्ता द्वारा केवल ट्रिगर, रूल्स या प्रोसिजर (जो कि प्रोग्राम होते हैं एवं डाटाबेस में संग्रहित किये जाते हैं) इनको लागू करके यह इन्टेग्रिटी सुनिश्चित की जाती है।

General Organization of डीबीएमएस



चित्र 1

महत्वपूर्ण बिन्दु

डीबीएमएस एक ऐसा सिस्टम है, जो प्रयोगकर्ता (user) को डाटा access करने एवं डाटा को सूचना में बदलने में सहायता प्रदान करता है। यह सिस्टम उपयोगकर्ताओं को– डाटा बनाने, अपडेट करने एवं सूचना प्राप्त करने की सहमति प्रदान करता है। डीबीएमएस समस्त सम्बन्धित रिकॉर्ड्स का समूह एवं प्रोग्राम है, जिसके द्वारा रिकॉर्ड access किए जाते हैं। डीबीएमएस कम्प्यूटर पर आधारित रिकॉर्ड व्यवस्था पूरी तरह से रिकॉर्ड को maintain (संभालने) एवं access करने के लिये है।

डीबीएमएस का प्राथमिक लक्ष्य ऐसा वातावरण प्रदान करना है, जो कि सुविधाजनक एवं डाटाबेस सूचना को संग्रह करने एवं Retrieve करने के योग्य हो।

डाटाबेस के लाभ–

1. एक जैसे रिकॉर्ड को कम करना।
2. डाटा की एकात्मकता एवं गुणवत्ता को बनाये रखना।
3. डाटा की असंगति (Inconsistency) को हटाना।
4. डाटा को सुरक्षित रखने के लिये आवश्यक प्रतिबन्ध लगाना।
5. सॉफ्टवेयर डवलपमेन्ट का खर्च कम करना।

डाटा के view-

- Internal / Physical Level View
- Conceptual / Logical Level View
- External / View Level

Data Model

- Object-Based Logical Model
- Record Based Logical Model
- Physical Data Model

आरडीबीएमएस का परिचय: आरडीबीएमएस में डाटा टेबलों में संग्रहित होता है जो कि पंक्ति एवं कॉलम की बनी हुई होती हैं। पंक्ति को टपल द्वारा एवं कॉलम को एट्रीब्यूट द्वारा इंगित किया जाता है। डोमैन(Domain), कॉलम में संग्रहित होने वाले मान का वास्तविक मान है जो कि किसी कॉलम में दर्ज होता है।

Data Definition Langague (DDL) : डीडीएल एक ऐसी भाषा है जो कि डाटाबेस में सारांश का वर्णन करती है। इसके द्वारा फील्डनेम, डाटा टाइप, डाटाबेस संग्रहित करने का स्थान एवं अनुबन्धों के वर्णन के साथ साथ, डाटाबेस के ऑब्जेक्ट्स जैसे कि टेबल, व्यू, इन्डेक्सेज आदि को क्रियेट एवं ड्रॉप किया जाता है।

Data Manipulation Language (DML): जिस भाषा के द्वारा data को manipulate किया जाए। उसे DML कहा जाता है, इसके द्वारा निम्न कार्य किये जाते है –

- डाटाबेस में संग्रहित सूचना का retrieval.

- डाटाबेस में नयी सूचनाओं का insertion.
- डाटाबेस में से सूचनाओं को delete करना
- डाटाबेस में संग्रहित सूचनाओं को modify करना।

रिलेशन्स/टेबल्स (Relations / Tables): Relational model में data, relation प्रारूप में संग्रहित किया जाता है। रिलेशन का औपचारिक नाम Tables है। Table में database इसकी contents को पहचानने (identify) के लिए एक unique name रखता है। प्रत्येक table में कॉलम व पंक्तियाँ (rows and columns) होती है।

टेबल (Table): Table एक प्राथमिक इकाई (primary unit) है, जो कि data का संग्रह database में करती है। जब कोई भी user database को access करता है, तब table का reference वांछित data पाने के लिए किया जाता है। Multiple tables में डाटा होने पर डाटा प्राप्त करने के लिये टेबल्स के मध्य रिलेशनशिप भी स्थापित किया जा सकता है। table के लिए database में physical storage वाछंनीय है।

टपल्स (Tuples): प्रत्येक relation/ tables बहुत सारे tuples से बनी होती है। इन्हें record भी कहा जाता है। यह पंक्तियां (rows) होती है, जिनसे table बनती है।

Domain: ये एक ऐसी values का समूह है, जिसके द्वारा एक या ज्यादा column, attribute / field अपनी सही values को प्राप्त कर सकते है। दूसरे शब्दों में यह एक फील्ड (field) का नाम है।

डाटाबेस स्कीमा (Database Schema): Schema साधारण तौर पर कई जुड़े हुए ऑब्जेक्ट्स का समूह है, जो database में होता है। Schema के अन्तर्गत जो भी ऑब्जेक्ट जुड़ा हुआ होता है, उनकी आपस में relationship भी होती है। Schema का सिर्फ एक ही owner होता है।

आर.डी.बी.एम.एस. के उपयोगकर्ता:– आर.डी.बी.एम.एस. के उपयोगकर्ताओं को मुख्यतः चार भागों में विभाजित किया गया है।

1. डाटाबेस डिजाइनरः
2. डाटाबेस एडमिनिस्ट्रेटरः
3. एप्लीकेशन डवलपरः
4. एण्ड यूजरः

Data Types: Data type के माध्यम से यह निश्चित किया जाता है कि database के कालम में किस तरह का डाटा रखा जाना है। जिसे database column में संग्रह किया जाता है। हालांकि कई तरह के data type उपलब्ध हैं, लेकिन ज्यादातर काम में लिए जाने वाले 3 मुख्य हैं, जो कि निम्न हैं–

- Alphanumeric
- Numeric
- Date and time

प्राईमरी की– प्राईमरी की, प्रत्येक रिकोर्ड में एक या एक से अधिक फील्ड का समूह होता है जो कि प्रत्येक रिकोर्ड को अलग से प्रदर्शित करता है तथा उनकी वेल्यू से डाटा का रिट्रीवल व अपडेटिंग शीघ्र संभव हो जाता है। यह डाटा के मान की एक से अधिक प्रविष्टि को रोकती है। प्राईमरी की का मान नल नहीं होनी चाहिए।

फोरेन की– फोरेन की, किन्हीं दो टेबल्स में संबंध स्थापित करने के लिए दोनों ही टेबल्स में कम से कम एक फील्ड कॉमन होना आवश्यक होता है। सामान्यतः एक टेबल में यह प्राईमरी की कहलाती है एवं दूसरी टेबल में यह फोरेन की कहलाती है। फोरेन की, का मान बार बार (Repeat) हो सकता है। फोरन की का उपयोग सामान्यतः संदर्भित टेबल के लिए किया जाता है और यह डाटा डोमेन इन्टेग्रिटी सुनिश्चित करती है।

कैन्डीडेट की (Candidate Keys) : candidate key एक minimal super key है। Minimal का मतलब होता है, यदि पूरे set में एक attribute हटाया जाए तो फिर ये super key नहीं रहती। यदि एक attribute को जोड़ें तो यह Minimal नहीं रहती है। (पर फिर भी यह super key है)। 'Candidate' शब्द का मतलब साधारण तौर पर यह है कि attributes के set को primary key की तरह इस्तेमाल कर सकते हैं, क्योंकि हमें आवश्यकता से ज्यादा attribute की जरूरत नहीं होती। किसी भी रिलेशन में एक से अधिक कैन्डीडेट की हो सकती है।

DATA INTEGRITY: system integrity का अर्थ है कि database में डाटा का सही व सुसंगत होना। विभिन्न तरीके की integrity निम्न प्रकार है–

- Entity Integrity
- Referential Integrity
- Domain Integrity
- User defined Integrity

अभ्यासार्थ प्रश्न

वस्तुनिष्ठ प्रश्न

1 रिलेशनल मॉडल किस डाटा मॉडल के अन्तर्गत आता है?

(अ) ऑब्जेक्ट बेस्ड लॉजिकल मॉडल के अन्तर्गत

(ब) रिकोर्ड बेस्ड लॉजिकल मॉडल के अन्तर्गत

(स) दोनों में

(द) उपरोक्त में से कोई नहीं

2. डाटाबेस स्कीमा के कितने Owner होते हैं–

(अ) एक

(ब) दो

(स) तीन

(द) चार

अति लघुत्तरात्मक प्रश्न

1. डाटा के व्यू कितने प्रकार के होते हैं?
2. रिकोर्ड बेस लॉजिकल मॉडल क्या होता है?
3. इन्टरनल/फिजिकल लेवल डाटा का वर्णन करें।

4. रिलेशन क्या होता है?
5. टपल क्या होता है?

लघुत्तरात्मक प्रश्न

1. डी.डी.एल. की व्याख्या कीजिए।
2. डी.एम.एल. की व्याख्या कीजिए।
3. डाटाबेस के क्या–क्या लाभ हैं?
4. एट्रीव्यूट्स कितने प्रकार के होते हैं?
5. डी.बी.एम.एस. के क्या–क्या लाभ है?
6. डी.बी.एम.एस. का वर्णन कीजिए।

निबन्धात्मक प्रश्न

1. डाटाबेस के विभिन्न यूजरों का वर्णन कीजिए।
2. डाटाबेस के व्यू का वर्णन कीजिए।

उत्तरमाला

1. (ब) 2. (अ)

2

माईएसक्यूएल का परिचय
INTRODUCTION TO MySQL

2.1 एसक्यूएल का परिचयः

एस.क्यू.एल. एक नॉन प्रोसीजरल भाषा है। यह डाटाबेस पर कठिन क्वैरी देने की सुविधा प्रदान करती है। SQL में उपयोगकर्ता (यूजर) कों बताना होता है कि क्या डाटा चाहिए, बिना यह बताये कि डाटा कैसे प्राप्त किया जाये। एस.क्यू.एल. का पूरा नाम स्ट्रक्चर्ड क्वैरी लैंग्वेज है और यह डाटाबेस से कम्यूनिकेट करने के उपयोग में लायी जाती है। यह अमेरिकन नेशनल स्टैण्डर्ड इंस्टीट्यूट (ANSI) मानक कम्प्यूटर भाषा है जो कि रिलेशनल डाटाबेस मेनेजमेंट सिस्टम के लिये डाटा एक्सेस एवम् डाटा के बदलाव के उपयोग में लायी जाती है। एस.क्यू.एल. स्टेटमेन्ट्स का उपयोग डाटाबेस में डाटा के अपडेट या देखने (रिट्रीव) के लिये किया जाता है। यह रिलेशनल डाटाबेस मैनेजमेंट सिस्टम के साथ काम करती है। रिलेशनल डाटाबेस मेनेजमेंट सिस्टम में डाटा टेबल्स में एकत्रित (स्टोर) किया जाता है। डाटाबेस टेबल्स का एक संग्रहण है। टेबल में रिकॉर्ड होते है एवम् रिकॉर्ड में फील्ड होते हैं। टेबल के प्रत्येक रिकॉर्ड का स्ट्रक्चर समान होता है।

एस.क्यू.एल. डाटाबेस के विभिन्न प्रोग्रामों जैसे एम.एस. एक्सेस, इन्ग्रेस, डी.बी.2, इनफोर्मिक्स, एम.एस.एस.क्यू.एल. सर्वर, ऑरेकल, साईबेस एंव माईएसक्यूएल आदि के साथ प्रयोग की जाती हैं। मानक एस.क्यू.एल. कमाण्ड जैसे कि सलेक्ट, अपडेट, डिलीट, क्रियेट एवम् ड्रॉप आदि का प्रयोग सामान्यतः डाटाबेस की सभी आवश्यकताओं की पूर्ति के लिये किया जाता है।

एस.क्यू.एल. को सामान्यतः तीन उप भाषाओं में बांटा जा सकता है। डाटा डेफिनेशन लैग्वेज (DDL) में सामान्यतः क्रियेट एवम् ड्रॉप कमाण्ड होते हैं जो कि डाटाबेस एवम् डाटाबेस के ऑब्जेक्ट (टेबल्स, व्यू, इन्डेक्स) आदि बनाने के काम आते हैं। एक बार डाटा स्ट्रक्चर तय होने के पश्चात् यूजर डाटा मैनीपुलेशन लैंग्वेज (DML) का उपयोग करता है जिसमें कि सामान्यतः इन्सर्ट, रिट्रीव एवम् मोडीफाई कमाण्ड होते हैं जो कि डाटा को मॉडीफाई करने के उपयोग में लिये जाते हैं। डाटा कन्ट्रोल लैंग्वेज का उपयोग डाटाबेस के रख–रखाव के लिये होता है जिसमें कि ग्रान्ट एवम् रिवोक की स्वीकृति डाटाबेस ऑब्जेक्ट्स के लिये दी जाती है।

2.2 माईएसक्यूएल(MySQL) का परिचयः

माईएसक्यूएल एक रिलेशनल डाटाबेस मैनेजमेंट सिस्टम (आरडीबीएमएस) है। इसको मुक्त स्त्रोत अनुज्ञापत्र (Open Source license) के तहत जारी किया गया है। इसमें मानक स्तर की एसक्यूएल (स्ट्रक्चर्ड क्वैरी लैंग्वेज) का उपयोग किया जाता है जो कि डाटाबेस में डेटा डालने के लिए, देखने के लिए एंव प्रकिया के लिए उपयोग की जाती है। माईएसक्यूएल बहुत ही तेज, सुरक्षित, विश्वसनीय एंव लचीली डीबीएमएस है। जिसका उपयोग व्यवसाइयों एंव अन्य प्रतिष्ठानों द्वारा किया जा रहा है। यह एक बहुत ही प्रसिद्ध मुक्त स्त्रोत अनुज्ञापत्र के तहत जारी की गई डेटाबेस है क्योंकि यह मुफ्त में (बिना कोई मूल्य दिये) उपलब्ध है। एंव यह भिन्न भिन्न ऑपरेटिंग सिस्टम पर कार्य करती है। माईएसक्यूएल का उपयोग भिन्न भिन्न प्रकार के कार्यो (एप्लीकेशन्स) में लिया जाता है लेकिन इसका मुख्यतः उपयोग वेब एप्लीकेशन्स के लिए किया जाता है।

2.3 माईएसक्यूएल(MySQL) की विशेषताऐंः

- यह मुक्त स्त्रोत अनुज्ञापत्र (Open Source license) के तहत जारी की गई डेटाबेस है एवं यह मुफ्त में (बिना कोई मूल्य दिये) इन हाउस उपयोग के लिए उपलब्ध है।
- यह बहुत ही तेज, सुरक्षित, विश्वसनीय एंव लचीली है।
- यह बहुत बडे डेटा पर बहुत तेज कार्य करती है।
- यह इन्डेक्स एंव बाईनरी ऑब्जेक्टस को समर्थन देती है।
- माईएसक्यूएल के द्वारा एसक्यूएल डाटा लैग्वेंज का उपयोग किया जाता है।
- माईएसक्यूएल का सोर्स कोड (Source Code) उपलब्ध होता है और इसको पुनः कम्पाइल (recompile) किया जा सकता है।
- यह C एंव C++ भाषा में लिखी गयी है।
- यह भिन्न भिन्न ऑपरेटिंग सिस्टमस (OS) को सपोर्ट (support) करती है।
- यह कई भाषाओं जैसे PHP, C, C++,JAVA,PERL etc.को सपोर्ट (support) करती है।

2.4 डाटा डेफिनेशन लैंग्वेज (डी.डी.एल) :

डाटा डेफिनेशन लैंग्वेज का उपयोग डाटाबेस एवम् डाटाबेस ऑब्जेक्ट्स जैसे कि टेबल्स, व्यू एवम् इण्डेक्स आदि को क्रियेट एवम् ड्रॉप करने के लिये किया जाता है। यह एसक्यूएल का भाग है जो डेटाबेस बनाने एवं हटाने के लिए अनुमति प्रदान करता हैं। इसमें इन्डेक्सेज को परिभाषित कर सकते हैं, टेबल्स के मध्य लिंक को स्पेसीफाई कर सकते हैं एवम् डाटाबेस टेबल्स पर प्रतिबन्ध लगा सकते हैं।

डाटा डेफिनेशन लैंग्वेज के प्रमुख कार्य निम्न हैः–

1. नई टेबल / डाटाबेस बनाना
2. टेबल / डाटाबेस में परिवर्तन करना
3. टेबल / डाटाबेस को ड्रॉप करना
4. इन्डेक्स बनाना
5. इन्डेक्स को ड्रॉप करना

2.5 माईएसक्यूएल द्वारा समर्थित डीडीएल कथन :

डाटाबेस बनाना

डाटाबेस बनाने का कमाण्ड निम्नानुसार है–

Create Database [IF NOT EXISTS] <db_name>;

उपरोक्त कमाण्ड के द्वारा ऊपर दिये गये नाम की डाटाबेस बनाई जा सकती है। यदि ऊपर दिये गये नाम के डाटाबेस पूर्व में ही बनी हुई है तो इस कमाण्ड द्वारा नई डाटाबेस इसी नाम से नहीं बनेगी। यदि **[IF NOT EXISTS]** ऑप्शन का उपयोग नहीं किया गया है एवं उपरौक्त नाम की डाटाबेस पूर्व में ही उपलब्ध है ऐसी स्थिति में यह त्रुटि संदेश देगा एवं डाटाबेस नहीं बनायेगा।

नई डाटाबेस बनानाः

school नाम से नई डाटाबेस बनाना–

create DATABASE IF NOT EXISTS school;

उपरोक्त कमाण्ड के द्वारा **school** नाम से नई डाटाबेस बनाई गई है। अब यदि सिस्टम पर उपलब्ध डाटाबेस देखना चाहते है तो निम्न कमाण्ड का उपयोग किया जाएगा।

show databases;

उपरोक्त कमाण्ड देने पर यह स्कूल डाटाबेस के साथ–साथ अन्य उपलब्ध डाटाबेस के नामों की सूची उपलब्ध करायेगा।

नई टेबल बनाने से पूर्व यह ठीक रहेगा कि डाटा के प्रकार की जानकारी की जायें।

2.6 डाटा के प्रकार–

टेबल बनाते समय फील्ड टाइप एंव फील्ड साइज की जानकारी होना बहुत ही महत्वपूर्ण है। माईएसक्यूएल में भिन्न–भिन्न प्रकार के डाटा टाइप का उपयोग किय जाता है। यह मुख्यतः तीन श्रेणियों में विभाजित किये जा सकते है।

1 String data types 2 Numeric data types 3 Date & Time data types

2.6.1 String Types:

CHAR(M) – यह एक निश्चित लम्बाई की स्ट्रिंग के लिए उपयोग किया जाता है। यह 1 से 255 करेक्टर तक संग्रहण (स्टोर) कर सकता है। **M** स्ट्रिंग की लम्बाई बताता है। एवं डिफाल्ट लम्बाई 1 है।

VARCHAR(M) – यह भी स्ट्रिंग डाटा टाइप के लिए उपयोग किया जाता है एवं यह परिवर्तनीय लम्बाई (1 से 255 करेक्टर) का संग्रहण कर सकता है। **VARCHAR , CHAR** की अधिक लचीली Form है। **M** स्ट्रिंग की लम्बाई बताता है।

BLOB or TEXT – यह भी स्ट्रिंग डाटा टाइप है। BLOB(Binary Large Objects) BLOB का उपयोग बहुत अधिक बाईनरी डाटा जैसे कि ईमेज या अन्य तरीके की फाइल का संग्रहण करने के लिए किया जाता है। यह अधिकतम 65535 करेक्टर तक संग्रहित कर सकता है।

LONGBLOB or LONGTEXT - यह भी स्ट्रिंग डाटा टाइप है तथा यह अधिकतम 4294967295 करेक्टर तक संग्रहित कर सकता है।

MEDIUMBLOB or MEDIUMTEXT - यह भी स्ट्रिंग डाटा टाइप है तथा यह अधिकतम 16777215 करेक्टर तक संग्रहित कर सकता है।

TINYBLOB or TINYTEXT - यह भी स्ट्रिंग डाटा टाइप है तथा यह अधिकतम 255 करेक्टर तक संग्रहित कर सकता है।

ENUM – यह भी स्ट्रिंग डाटा टाइप है। यह एक आईटमस की सूची बनाता है। जिसमें से किसी एक को आप के द्वारा चयन किया जाता है। यह नल भी हो सकता है। उदाहरण के तौर पर यदि आप किसी फील्ड में E या F या G में से चाहते है। तो ENUM (‘E’,’F’,’G’) फील्ड बनायें। यह या तो E या F या G में से चयन करेगा या नल मान लेगा।

SET -यह भी स्ट्रिंग डाटा टाइप है। यह निर्दिष्ट सूची में से एक या एक से अधिक मान को चयन करने की सुविधा देता है। जैसे कि transport SET (“jeep”, “scooter”) NOT NULL ; उपरोक्त transport फील्ड में “jeep” “scooter” “jeep,scooter “ मान आ सकता है। अधिकतम 64 मान निर्दिष्ट किये जा सकते है। किसी फील्ड में सेट डेटाटाईप से उपलब्ध सूची से एक से अधिक मान भी संग्रहित किया जा सकता है। जबकि ENUM DATA TYPE द्वारा केवल एक ही मान संग्रहित किया जा सकता है।

1.6.2 Numeric Data Types:

INT – एक पूर्णांक मान जो कि चिन्हित अथवा बिना चिन्ह का हो सकता है। चिन्हित की सीमाऐं –2147483648 से 2147483647 तक होती है। यदि बिना चिन्ह के है तो सीमाऐं 0 से 4294967295 तक होगी।

BIGINT - एक बडा पूर्णांक मान जो कि चिन्हित अथवा चिन्ह बिना हो सकता है। चिन्हित की सीमाऐं 9223372036854775808 सें 9223372036854775807 तक होती है। यदि बिना चिन्ह के है तो सीमाऐं 0 से 18446744073709551615 तक होगी। इसका आकार 8 बाईट का होता हैं।

MEDIUMINT - एक मध्यम आकार का पूर्णांक जो कि चिन्हित अथवा बिना चिन्ह के हो सकता है। यदि चिन्हित है तो सीमाऐं –8388608 से 8388607 तक होती है। यदि चिन्ह बिना है तो सीमाऐं 0 से 16777215 तक होती है। इसका आकार 3 बाईट का होता है।

SMALLINT - छोटे आकार का पूर्णांक चिन्ह अथवा बिना चिन्ह हो सकता है। यदि चिन्हित है तो सीमाऐं -32768 से 32767 तक होती है। यदि चिन्ह बिना है तो सीमाऐं 0 से 65535 तक होती है। इसका आकार 2 बाईट का होता है।

TINYINT - अति छोटे आकार का पूर्णांक चिन्ह अथवा बिना चिन्ह हो सकता है। यदि चिन्हित है तो सीमाऐं -128 से 127 तक होती है। यदि चिन्ह बिना है तो सीमाऐं 0 से 255 तक होती है। इसका आकार 1 बाईट का होता है।

FLOAT(M,D) - फ्लोटिंग प्वांइट संख्या बिना चिन्ह के नहीं हो सकती। यहां (M) संख्या की लम्बाई तथा (D) दशमलव के बाद के अंको की संख्या है। यदि परिभाषित नहीं किया गया है तो यह स्वतः 10,2 होती है, जहां दो दशमलव के बाद के अंको की संख्या हैं तथा 10 संख्या के कुल अंक है (दशमलव सहित) इसका आकार 4 बाईट होता है।

DOUBLE(M,D) - डबल प्रसिशन फ्लोटिंग प्वांइट संख्या भी बिना चिन्ह के नहीं हो सकती है। यहां (M) संख्या की लम्बाई तथा (D) दशमलव के बाद के अंको की संख्या है। यदि परिभाषित नहीं किया गया है तो यह स्वतः 16,4 होती है, जहां 4 दशमलव के बाद के अंको की संख्या हैं। 16 संख्या के कुल अंक है (दशमलव सहित) इसका आकार 8 बाईट होता है।

DECIMAL(M,D) – अनपैक्ड फ्लोटिंग प्वाइंट संख्या भी बिना चिन्ह के नहीं हो सकती है। प्रत्येक डेसीमल एक बाइट को प्रदर्शित करता है। इसका आकार 8 बाईट होता है। (M) तथा (D) का मान प्रदर्शित करना आवश्यक है। DECIMAL का समानार्थक NUMERIC है।

2.6.3 Date and Time data types :

DATE - DATE का फोरमेट YYYY-MM-DD है तथा इसकी सीमाऐं 1000-01-01 से 9999-12-31 तक है। इसका आकार 3 बाइट का है। उदाहरणार्थ November 28th, 1986 फील्ड में 1986-11-28 के रूप में संग्रहित होगा।

DATETIME - इसमें दिनांक तथा समय का संयोजन YYYY-MM-DD HH:MM:SS फोरमेट में होता है। जिसकी सीमाऐं 1000-01-01 00:00:00 से 9999-12-31 23:59:59 तक होती है। इसका आकार 8 बाइट का होता है। उदाहरणार्थ 30 अक्टूबर, 1986 के दोपहर बाद के समय 5:45 को 1986-10-30 17:45:00 के रूप में संग्रहित होगा।

TIMESTAMP - इसका फोरमेट पूर्व DATETIME की तरह ही होता है परन्तु – (hyphen) का उपयोग नहीं होता। उदाहरणार्थ 30 दिसम्बर 1986 के दोपहर बाद के 3:30 बजे को 19861230153000 (YYYYMMDDHHMMSS) रूप में संग्रहित होगा। इसका आकार 4 बाइट है।

TIME - यह समय को HH:MM:SS फोरमेट में संग्रहित करता है। इसका आकार 3 बाइट है।

YEAR(M) - यह वर्ष को 2 अंक या 4 अंक के फोरमेट में संग्रहित करता है। यदि दो अंक में प्रदर्शित किया जाता है तो YEAR 1980 से 2079 के लिए (80 से 79) होगा। यदि 4 अंको में प्रदर्शित किया जाता है तो 1980 से 2079 होगा। स्वतः(default) ही इसका आकार 4 अंको में तथा इसका मैमोरी वर्ड साइज 1 बाइट का होता है।

2.7 प्रतिबन्ध (Constraints):

Constraint specifications द्वारा किसी भी टेबल में अतिरिक्त शर्ते/प्रतिबन्ध/चैक लगाये जा सकते है और इनको डी.बी.एम.एस. स्वतः ही लागू करता है।

किसी टेबल पर निम्न प्रतिबन्ध (constraints) लगाये जा सकते हैं –

प्राईमरी की (Primary key) प्रतिबन्ध– प्राईमरी की, टेबल के एक या एक से अधिक फील्ड का समूह होता है जो कि प्रत्येक रिकॉर्ड को टेबल में अलग से प्रर्दशित करता है तथा उनकी वेल्यू से डाटा का रिट्रीवल व अपडेटिंग शीघ्र संभव हो जाता है। प्राईमरी की एक अद्वितीय मान होता है जिसके द्वारा टेबल के प्रत्येक रिकॉर्ड को प्रथक से ज्ञात किया जाता है। इस प्रतिबन्ध से यह सुनिश्चित किया जाता है कि प्रत्येक रिकॉर्ड में प्राईमरी की का मान अलग–अलग है। यदि प्राईमरी की का मान एक समान दिया जाता है तो टेबल में डाटा स्वीकार नहीं किया जायेगा। इसमें नल वैल्यू को दर्ज नहीं किया जा सकता है।

उदाहरण : Alter table titles

ADD Constraint pk_titleid PRIMARY KEY (title_id);

इसमें pk_titleid प्रतिबन्ध प्राईमरी की title_id कॉलम पर title टेबल में लगाया गया है।

यूनिक (UNIQUE) प्रतिबन्ध— यूनिक द्वारा यह दर्शाया जाता है कि प्रत्येक पंक्ति के लिये कॉलम विशेष का मान अद्वितीय है। टेबल की प्रत्येक पंक्ति में इस कॉलम की वेल्यू अलग–अलग होती है। Unique constraint द्वारा यह सुनिश्चित किया जाता है कि टेबल की प्रत्येक पंक्ति में उस कॉलम की अलग–अलग वेल्यू है। यह इन्डेक्स स्वतः ही क्रियेट हो जाता है। यह प्रतिबन्ध उस कॉलम पर भी लगाया जा सकता है जो कि नल वैल्यू लेता हो। यह प्रतिबन्ध एन्टिटी इंटेग्रिटी को सुनिश्चित करता है।

उदाहरणार्थः Alter table stores

ADD Constraint uk_storeid UNIQUE store_id;

उपरोक्त उदाहरण द्वारा प्रतिबन्ध uk_storeid कॉलम store_id पर लगाया गया हैं।

उदाहरणार्थः CREATE TABLE job

```
(
job_id smallint PRIMARY KEY,
min_amt int NOT NULL
CHECK (min_amt>=1000),
```

Referential Integrity प्रतिबन्ध– किन्ही दो टेबलों में संबंध स्थापित करने के लिए दोनों ही टेबलों में कम से कम एक फील्ड कॉमन होना आवश्यक होता है। सामान्यतः एक टेबल में यह प्राईमरी की कहलाती है एवं दूसरी टेबल में यह फोरेन की कहलाती है। इस प्रतिबन्ध द्वारा दूसरी टेबल में रखे जाने वाले डाटा को नियन्त्रित करना होता है। तात्पर्य यह है कि दूसरी टेबल में वही डाटा आना चाहिए जो कि प्रथम टेबल की प्राईमरी की के डाटा से मिलता हो। जिन कॉलम्स पर फोरेन की प्रतिबन्ध लगाया जाता है उन पर इंडेक्स क्रिएट नहीं किया जा सकता। यह प्रतिबन्ध रैफरेन्सीयल इंटेग्रिटी को सुनिश्चित करता है।

उदाहरणार्थः Alter table titleauthor

```
ADD Constraint fk_titleid REFERENCES titles.title_id;
```

उपरोक्त उदाहरण में फोरेन की प्रतिबन्ध fk_titleid कॉलम title_id जो कि titles table में प्राईमरी की भी है, पर लगाया गया है। MySQL में टेबलो मे सबंध स्थापित करने के लिए रेफरेसिंग टेबल में फोरेन की स्थापित की जाती है हम child table के कॉलम अथवा कॉलम्स में फोरेन की स्थापित कर सकते है जो कि रेफरेन्स्ड parent टेबल के कॉलम अथवा कॉलम्स को रेफर करता है। ये फोरेन की child तथा parent टेबल में संगतता (consistency) रखती है।

Syntax

```
REFERENCES referenced-table(column)
… or as a separate constraint ..
FOREIGN KEY(referencing-table-columns)
REFERENCES referenced-table(columns);
```

उदाहरणार्थः

Creating the person - phone foreign key constraint

```
ALTER TABLE phone ADD
    CONSTRAINT FK_phone_person_num
   FOREIGN KEY (person_num)
  REFERENCES person(num);
```

चैक (CHECK) प्रतिबन्ध — यह यूजर द्वारा टेबल पर लगाया गया प्रतिबन्ध हैं इसका उपयोग domain integrity को लागू करने के लिए, किसी कॉलम की वेल्यू को नियंत्रित (limiting) करने के लिए किया जाता है।

उदाहरणार्थः

```
CREATE TABLE job
(
job_id smallint PRIMARY KEY,
min_amt int NOT NULL
CHECK (min_amt>=1000),
        max_amt int NULL
);
```

उपरोक्त उदाहरण में प्रतिबन्ध CHECK कॉलम min_amt पर लगाया गया है जिसमें डाटा दर्ज करते समय यह सुनिश्चित किया जायेगा कि min_amt कॉलम का मान 1000 के बराबर या अधिक हो।

नॉट नल (NOT NULL Constraint) प्रतिबन्ध— इसके द्वारा यह देखा जाता है कि किसी कॉलम विशेष में नल वेल्यू नहीं हो। यह यूजर को उस कॉलम में मान डालने के लिए बाध्य करता है।

Constraints को कभी भी टेबल बनाने के पश्चात् और टेबल बनाते समय add/disable/enable/drop किया जा सकता है।

उदाहरणार्थः CREATE TABLE job

```
(
job_id smallint PRIMARY KEY,
min_amt int NOT NULL
max_amt int
);
```

उपरोक्त उदाहरण में प्रतिबन्ध NOT NULL कॉलम min_amt पर लगाया गया है जिसमें डाटा दर्ज करते समय यह सुनिश्चित किया जायेगा कि min_amt कॉलम का मान NULL नहीं होना चाहिए।

2.8 टेबल बनाना

डाटाबेस बनाने के पश्चात डाटाबेस में टेबल का निर्माण करना है जो डाटा को संग्रहित करके रख सके। टेबल बनाने के कमाण्डः

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS]
<table_name>[(create_definition,...)] [table_options] [select_statement]
```

table-name में नयी टेबल का नाम होता है। यदि उक्त नाम की टेबल पहले से उपलब्ध है तो उसनाम से टेबल क्रियेट नहीं होगी। [IF NOT EXISTS] ऑप्शन काम में नहीं लिया जा रहा है और दिये नाम की टेबल पहले से है तो कम्प्यूटर Error देगा तथा टेबल क्रियेट नहीं होगी। TEMPORARY ऑप्शन काम में लिया जा रहा है तो जो टेबल बनेगी वह डेटाबेस के वर्तमान कनेक्शन तक ही उपलब्ध रहेगी जैसे ही कनेक्शन समाप्त होगा टेबल भी समाप्त हो जायेगी।

column-name किसी भी टेबल में अद्वितीय होता है। data-type द्वारा उस कॉलम में किस प्रकार का मान रखा जाना है बताया जाता है। टेबल का निर्माण किसी अन्य दूसरी टेबल से भी किया जा सकता है यदि हम दूसरी टेबल के कॉलम्स को काम में ले रहे हैं।

- NOT NULL/NULL का उपयोग फील्ड में डेटा होना आवश्यक है अथवा नहीं, इसे प्रदर्शित करता है।
- DEFAULT option का उपयोग स्वतः सिद्ध मान को निर्धारित करने के लिए किया जाता है।
- Integer डेटा टाईप वाले कॉलम में AUTO-INCREMENT attribute का उपयोग किया जा सकता है जो इनपुट NULL होने पर अपने–आप ही कॉलम (फील्ड़) के मान को एक से बढ़ा देता है।
- PRIMARY KEY attribute उस कॉलम (फील्ड़) के लिए उपयोग होता है जिसे हम डेटा को शीघ्र एक्सेस करने के लिए Index के रूप में लेना चाहते है।

CREATE TABLE command उपयोग करने से पहले use db_name command का उपयोग करते है।

उदाहरणार्थः

```
use school;
 Database changed
 CREATE table student(
     Rollno INT NOT NULL PRIMARY KEY,
     Studentname VARCHAR(30),fname VARCHAR(30), dob datetime,
      Class CHAR(10), sect CHAR(5));
```

उपरोक्त command द्वारा school Database में table student क्रियेट हो रही है। जिसके कॉलम (फील्ड)

ROLLNO, STUDENTNAME, FNAME, DOB, CLASS and SECT हैं।

यहां ROLLNO कॉलम Primary Key के लिए काम आ रहा है।

अन्य उदाहरणः

अब अन्य टेबल Marks निम्न उदाहरण द्वारा की गई है। जिसमें ROLLNO, MARKS तथा SUBJECT फील्ड है

```
CREATE TABLE MARKS
(ROLLNO INT,
```

```
MARKS INT,
SUBJECT CHAR(30));
```

ये देखने के लिए की टेबल क्रियेट हुई अथवा नहीं **SHOW TABLES** command काम आती है। ये कमाण्ड डेटाबेस में उपस्थित टेबलों की लिस्ट को प्रदर्शित करती है।

```
show tables;
```

उपरोक्त कमाण्ड से निम्न परिणाम प्राप्त होगा।

student

marks

2.9 डेटाबेस में बदलाव (Altering the Database)

डेटाबेस में अद्यतन के लिए निम्न कमाण्ड का उपयोग करते है।

ALTER {DATABASE} [db_name] alter_specification [alter_specification]......

ALTER DATABASE कमाण्ड द्वारा हम डेटाबेस के लक्षणों (characteristics) में बदलाव कर सकते है। ये लक्षण db.opt फाइल में स्टोर होते है। निम्न उदाहरण द्वारा डेटाबेस की एक फाइल Sample1 के आकार को बढा रहे है।

```
ALTER DATABASE school
modify file (NAME=sample1, SIZE=10MB);
```

2.10 टेबल में बदलाव (Altering the Table):

टेबल में बदलाव का सामान्य Syntax निम्न प्रकार से है

ALTER TABLE tbl_name alter_specification [, alter_specification] ...

ALTER TABLE कमाण्ड द्वारा हम टेबल के स्ट्रक्चर में बदलाव कर सकते है।

इस कमाण्ड द्वारा हम टेबल के कॉलम्स को जोडना, बदलना तथा हटा सकते है, index को क्रियेट अथवा हटा सकते है, कॉलम का अथवा टेबल का नाम बदल सकते है। निम्न उदाहरणों द्वारा हम टेबल के विभिन्न बदलावों को समझ सकते है।

टेबल का नाम बदलने के लिए

ALTER TABLE old_name RENAME TO new_name;

SHOW tables;

उक्त कमाण्ड का परिणाम **student** तथा **marks** होगा।

ALTER TABLE student RENAME to student_data;

SHOW tables;

उक्त कमाण्ड का परिणाम **student_data** तथा **marks** होगा।

निम्न उदाहरण टेबल में नया कॉलम जोडने को प्रदर्शित करता है

ALTER TABLE tbl_name ADD column_name column_type;

अधिक कॉलम जोडने के लिए

ALTER TABLE tbl_name ADD (column_1 column_type, column_2 column_type);

DESCRIBE student;

Field	Type	Null	Key	Default	Extra
Rollno	Int (10) insigned	NO	PRI		
Studentname	Varchar (30)	YES			
Fname	Varchar (30)	YES			
dob	datetime	YES			
Class	CHAR (10)	YES			

ALTER TABLE student
-> ADD Age numeric (3);
DESCRIBE student;

Field	Type	Null	Key	Default	Extra
Rollno	Int (10) insigned	NO	PRI		
Studentname	Varchar (30)	YES			
Fname	Varchar (30)	YES			
dob	datetime	YES			
Class	CHAR (10)	YES			
Sect	CHAR (5)	YES			

Field	Type	Null	Key	Default	Extra
Rollno	Int (10) insigned	NO	PRI		
Studentname	Varchar (30)	YES			
Fname	Varchar (22)	YES			
dob	datetime	YES			
Class	CHAR (10)	YES			
Sect	CHAR (5)	YES			
Age	Decimal (3,0)	YES			

निम्न उदाहरण टेबल से कॉलम हटाने (DROP) को प्रदर्शित करता है।

ALTER TABLE tbl_name DROP COLUMN column_name;

DESCRIBE student;

Field	Type	Null	Key	Default	Extra
Rollno	Int (10) insigned	NO	PRI		
Studentname	Varchar (30)	YES			
Fname	Varchar (22)	YES			
dob	datetime	YES			
Class	CHAR (10)	YES			
Sect	CHAR (5)	YES			
Age	Decimal (3,0)	YES			

ALTER TABLE student **DROP COLUMN Age;**

DESCRIBE student;

Field	Type	Null	Key	Default	Extra
Rollno	Int (10) insigned	NO	PRI		
Studentname	Varchar (30)	YES			
Fname	Varchar (22)	YES			
dob	datetime	YES			
Class	CHAR (10)	YES			
Sect	CHAR (5)	YES			

2.11 डेटाबेस को हटाना (Dropping the Database):

DROP DATABASE SCHOOL:

इस कमाण्ड का प्रयोग सावधानी से करना चाहिए क्योंकि यह सम्पूर्ण डेटाबेस को ही हटा देती है। केवल कुछ रिकार्डस् को हटाने के लिए DML की DELETE command का प्रयोग होता है।

डेटाबेस को **DROP** करने का syntax निम्न प्रकार हैः

DROP {DATABASE | SCHEMA} [IF EXISTS] db_name;

उक्त कमाण्ड डेटाबेस की टेबल समेत डेटाबेस को समाप्त कर देता है। यदि " IF EXISTS" option काम में नहीं लिया जा रहा है तथा डेटाबेस उपलब्ध नहीं है तो Error message प्रदर्शित होगा।

माना कि हमने emp1 नाम का डेटाबेस क्रियेट किया हुआ है।

SHOW DATABASE;

उक्त कमाण्ड कुल डेटाबेस की लिस्ट emp1 के साथ प्रदर्शित करता है।

DROP DATABASE IF NOT EXISTS EMP1;

SHOW DATABASE;

उक्त कमाण्ड emp1 के अलावा उपलब्ध डेटाबेस की लिस्ट प्रदर्शित करेगी, क्योंकि emp1 डेटाबेस हट गया है।

2.12 टेबल को हटाना (Dropping the Table):

Data Definition Language की निर्णायक कमाण्ड है DROP। ये कमाण्ड डेटाबेस ऑब्जेक्ट्स को हटाने के काम आती है।

टेबल Drop करने की कमाण्ड का Syntax निम्न है।

DROP [TEMPORARY] TABLE [IF EXISTS] tbl_name;

यह कमाण्ड डेटाबेस से एक या एक से अधिक टेबल को हटाने के काम आती है। यह कमाण्ड टेबल के डेटा तथा परिभाषा दोनों को ही हटा देती है। यदि " IF EXISTS" option काम में नहीं लिया जा रहा है तथा टेबल उपलब्ध नहीं है तो Error message प्रदर्शित होगा। यदि TEMPORARY keyword काम में लिया जा रहा है तो कमाण्ड अस्थायी टेबल (temporary tables) को हटायेगा। यह कमाण्ड चल रहे कार्य सम्पादन को समाप्त नहीं करता है तथा यह अभिगम अधिकार (access right) को भी जांच नहीं करता है क्योंकि temporary table केवल client पर ही दर्शनीय है इसलिए अभिगम अधिकार की जांच आवश्यक नहीं है। यदि TEMPORARY keyword काम में नहीं लिया

जा रहा है तो DROP TABLE command स्वतः ही वर्तमान में सक्रिय अभिगम (transaction) को Commit कर देता है।

Example:

DROP TABLE MARKS;
DROP TABLE STUDENT;

उपरोक्त कमाण्ड से marks तथा student टेबल हटा दिये गये है।

2.13 टेबल का नाम बदलना (Rename the Table): टेबल का नाम बदलने का सामान्य Syntax निम्नलिखित हैः

RENAME TABLE tbl_name TO new_tbl_name, tbl_name2 TO new_tbl_name2 ...

इस कमाण्ड द्वारा हम एक या एक से अधिक टेबल के नाम बदल सकते है। नाम बदलने का कार्य स्वतः ही इस प्रकार होता है कि नाम बदलते समय कोई अन्य thread टेबल को एक्सेस नहीं कर सकता।

उदाहरणार्थः

SHOW TABLES;

उक्त कमाण्ड डेटाबेस में उपस्थित कुल टेबल की लिस्ट को प्रदर्शित करता है।

माना emptemp नामक टेबल को emp_temp1 नाम से परिवर्तित करना है तब निम्न कमाण्ड का उपयोग होगा।

RENAME TABLE EMPTEMP TO EMP_TEMP1;

यह कमाण्ड emptemp का नाम emp_temp1 नाम से परिवर्तित कर देगा।

2.14 Index:

हमें बहुत बडे डेटा को बिना समय नष्ट किये एक्सेस करना होता है। Index द्वारा टेबल से डेटा को शीघ्रता से एक्सेस करना संभव है। किसी टेबल में Index पंक्तियो को अतिशीघ्रता तथा निपुणता से एक्सेस करने के लिए क्रियेट किये जाते है। किसी भी टेबल में एक या एक से अधिक कॉलम पर Index क्रियेट करना संभव है तथा प्रत्येक Index को नाम देना आवश्यक है। उपयोगकर्ता (Users) Indexes को नहीं देख सकते हैं, वे केवल क्वैरीज की गति तेज करने के लिए इनका उपयोग करते हैं

Unique Index:

ये कमाण्ड **Unique Index** क्रियेट करने के काम आती है। यहां **Unique Index** का मतलब है कि किन्ही दो पंक्तियो के **Index** का मान समान नहीं होता।

CREATE UNIQUE INDEX index_name
ON table_name (column_name);

"column_name"उस कॉलम का नाम है जिस पर **Index** क्रियेट किया गया है।

Simple Index:

ये कमाण्ड टेबल में **Simple Index** क्रियेट करने के काम में आता है। यंहा UNIQUE keyword हटाया गया है अतः Index के मान की पुनर्रावृति (duplicate values) संभव है।

CREATE INDEX index_name
ON table_name (column_name);

Index को हटानाः

Index को हटाने के लिए DROP INDEX कमाण्ड काम में ली जाती है।

DROP INDEX <indexname> [ON <tablename>];

Index को क्रियेट करने के लिए आवश्यक Alter कमाण्ड निम्न प्रकार से हैः

- **ALTER TABLE tbl_name ADD INDEX index_name (column_list):** यह सामान्य Index है जिसमें कोई भी मान एक से अधिक बार आ सकता है।
- **ALTER TABLE tbl_name ADD UNIQUE index_name (column_list):** यह कमाण्ड ऐसा Index क्रियेट करती है जिसमें Index के मान अलग–अलग (unique) आवश्यक है (परंतु NULL मान एक से अधिक बार आ सकते हैं)
- **ALTER TABLE tbl_name ADD PRIMARY KEY (column_list)** : यह कमाण्ड PRIMARY KEY क्रियेट करती है। जिसका तात्पर्य है KEY में अलग–अलग (unique) मान होगें तथा कोई भी मान NULL नहीं होने चाहिए।
- **ALTER TABLE tbl_name ADD FULLTEXT index_name (column_list):** यह एक विशेष FULLTEXT index क्रियेट करता है। जो कि text-searching में काम आता है।

महत्वपूर्ण बिन्दु

एसक्यूएलः एक नॉन प्रोसीजरल भाषा है। यह डाटाबेस पर कठिन क्वैरी देने की सुविधा प्रदान करती है। SQL में उपयोगकर्ता (यूजर) कों बताना होता है कि क्या डाटा चाहिए, बिना यह बताये कि डाटा

मानक कम्प्यूटर भाषा है जो कि रिलेशनल डाटाबेस मेनेजमेंट सिस्टम के लिये डाटा एक्सेस एवम् डाटा के बदलाव के उपयोग में लायी जाती है। एस.क्यू.एल. स्टेटमेन्ट्स का उपयोग डाटाबेस में डाटा के अपडेट या देखने (रिट्रीव) के लिये किया जाता है। यह रिलेशनल डाटाबेस मैनेजमेंट सिस्टम के साथ काम करती है। रिलेशनल डाटाबेस मेनेजमेंट सिस्टम में डाटा टेबल्स में एकत्रित (स्टोर) किया जाता है। डाटाबेस टेबल्स का एक संग्रहण है। टेबल में रिकॉर्ड होते है एवम् रिकॉर्ड में फील्ड होते हैं। टेबल के प्रत्येक रिकॉर्ड का स्ट्रक्चर समान होता है।

एस.क्यू.एल. डाटाबेस के विभिन्न प्रोग्रामों जैसे एम.एस. एक्सेस, इन्ग्रेस, डी.बी.2, इनफोर्मिक्स, एम. एस.एस.क्यू.एल. सर्वर, ऑरेकल, साईबेस एंव माईएसक्यूएल आदि के साथ प्रयोग की जाती हैं। मानक एस.क्यू.एल. कमाण्ड जैसे कि सलेक्ट, अपडेट, डिलीट, क्रियेट एवम् ड्रॉप आदि का प्रयोग सामान्यतः डाटाबेस की सभी आवश्यकताओं की पूर्ति के लिये किया जाता है।

एस.क्यू.एल. को सामान्यतः तीन उप भाषाओं में बांटा जा सकता है। डाटा डेफिनेशन लैंग्वेज (DDL) में सामान्यतः क्रियेट एवम् ड्रॉप कमाण्ड होते हैं जो कि डाटाबेस एवम् डाटाबेस के ऑब्जेक्ट (टेबल्स, व्यू, इन्डेक्स) आदि बनाने के काम आते हैं। एक बार डाटा स्ट्रक्चर तय होने के पश्चात् यूजर डाटा मैनीपुलेशन लैंग्वेज (DML) का उपयोग करता है जिसमें कि सामान्यतः इन्सर्ट, रिट्रीव एवम् मोडीफाई कमाण्ड होते हैं जो कि डाटा को मॉडीफाई करने के उपयोग में लिये जाते हैं। डाटा कन्ट्रोल लैंग्वेज का उपयोग डाटाबेस के रख–रखाव के लिये होता है जिसमें कि ग्रान्ट एवम् रिवोक की स्वीकृति डाटाबेस ऑब्जेक्ट्स के लिये दी जाती है।

माईएसक्यूएलः एक रिलेशनल डाटाबेस मैनेजमेंट सिस्टम (आरडीबीएमएस) है। इसको मुक्त स्त्रोत अनुज्ञापत्र (Open Source license) के तहत जारी किया गया है। इसमें मानक स्तर की एसक्यूएल (स्ट्रक्चर्ड क्वैरी लैंग्वेज) का उपयोग किया जाता है जो कि डाटाबेस में डेटा डालने के लिए, देखने के लिए एंव प्रकिया के लिए उपयोग की जाती है। माईएसक्यूएल बहुत ही तेज, सुरक्षित, विश्वसनीय एंव लचीली डीबीएमएस है। जिसका उपयोग व्यवसाइयों एंव अन्य प्रति ठानों द्वारा किया जा रहा है। यह एक बहुत ही प्रसिद्ध मुक्त स्त्रोत अनुज्ञापत्र के तहत जारी की गई डेटाबेस है क्योंकि यह मुफ्त में (बिना कोई मूल्य दिये) उपलब्ध है। एंव यह भिन्न भिन्न ऑपरेटिंग सिस्टम पर कार्य करती है। माईएसक्यूएल का उपयोग भिन्न भिन्न प्रकार के कार्यो (एप्लीकेशन्स) में लिया जाता है लेकिन इसका मुख्यतः उपयोग वेब एप्लीकेशन्स के लिए किया जाता है।

डाटा के प्रकार-

Character (String)

- CHAR(M)
- VARCHAR(M)
- BLOB OR TEXT
- TINYBLOB OR TINYTEXT
- MEDIUMBLOB OR MEDIUMTEXT
- LONGBLOB OR LONGTEXT
- ENUM
- SET

Numeric

- TINYINT
- SMALLINT
- INT
- BIGINT
- MEDIUMINT
- FLOAT
- DECIMAL
- DOUBLE

Datetime

- DATE
- DATETIME
- TIMESTAMP
- TIME
- YEAR

Constraint specifications द्वारा किसी भी टेबल में अतिरिक्त शर्ते/प्रतिबन्ध/चैक लगाये जा सकते है और इनको डी.बी.एम.एस. स्वतः ही लागू करता है।

- Primary key
- UNIQUE
- REFERENCE
- CHECK
- NOT NULL

अभ्यासार्थ प्रश्न

वस्तुनिष्ठ प्रश्न

1 डी.डी.एल. के द्वारा आप क्या कर सकते हैं?

(अ) डाटा को एन्टर एवम् एडिट (ब) डाटाबेस ऑब्जेक्ट को क्रियेट व ड्रॉप

(स) उपरोक्त दोनों कार्य कर सकते है (द) उपरोक्त में से कोई नहीं

2 एस.क्यू.एल. किस प्रकार की भाषा है?

(अ) नॉन प्रोसीजरल लैग्वेज (ब) प्रोसीजरल लैग्वेज

(स) मशीन लैंग्वेज (द) उपरोक्त में से कोई नहीं

3. एस.क्यू.एल. को कितनी उप–भाषाओं में बांटा गया है?

(अ) दो (ब) तीन

(स) आठ (द) पाँच

4. निम्न में से RDBMS कौनसा है?
   (अ) MySQL (ब) MS excel
   (स) DBASE (द) उपरोक्त में से कोई नहीं

अति लघुत्तरात्मक प्रश्न

1. MEDIUMINT डाटा टाईप का उपयोग क्या है?
2. VARCHAR डाटा टाईप का क्या उपयोग है?
3. डेट डेटा टाईप में TIMESTAMP का क्या उपयोग होता है?

लघुत्तरात्मक प्रश्न

1. कन्स्ट्रेन्ट क्या होते हैं?
2. आल्टर कमाण्ड का क्या उपयोग होता है?
3. ड्रॉप कमाण्ड का क्या उपयोग होता है?

निबन्धात्मक प्रश्न

1. MySQL के क्या लाभ है?
2. माई.एस.क्यू.एल. का वर्णन कीजिए।
3. MySQL के विभिन्न डाटा टाईप कौन–कौन से है?
4. एक टेबल बनाइए जिसमें प्राईमरी की के साथ–साथ एक अन्य फील्ड पर नॉट नल का प्रतिबन्ध दिया गया हो। उदाहरण सहित वर्णन करें।
5. निम्न की व्याख्या कीजिए।
   (i) यूनिक कान्स्ट्रेन्ट
   (ii) रेफरेन्सेज कान्स्ट्रेन्ट
   (ii) नॉट नल कान्स्ट्रेन्ट

उत्तरमाला

1 ब 2 अ 3 ब 4 अ

3
डाटा मैनीपुलेशन लैग्वेज
Data Manipulation Language

3.1 परिचय

डाटा मैनीपुलेशन लैग्वेज Data Manipulation Language (DML) द्वारा डाटाबेस की सूचना को रिट्रीव, जोड एवम् बदल सकते है। DML कमाण्ड्स का उपयोग उपयोगकर्ता द्वारा नियमित डाटाबेस के कार्य के दौरान किया जाता है। क्वैरी तथा अपडेट कमाण्डस SQL के DML का भाग है तथा MySQL द्वारा समर्थित है। सामान्य तौर पर काम आने वाले DML कमाण्डस निम्न है–

- **SELECT** - Database में संग्रहित सूचना का retrieval.
- **UPDATE** - Database में संग्रहित सूचनाओं को modify करना।
- **DELETE** - Database में से सूचनाओं को delete करना।
- **INSERT INTO** - Database में नयी सूचनाओं को डालना (insertion)

MySQL द्वारा समर्थित DML कथनः

3.1.1 INSERT

MySQL में INSERT कमाण्ड का उपयोग वर्तमान टेबल में रिकॉर्डस को जोडने के लिये किया जाता है। माना कि हम student table में किसी student का रिकॉर्ड जोडना चाहते है तो हमें निम्न प्रकार से कमाण्ड लिखना होगा–

```
INSERT INTO STUDENT
VALUES (10001,'VINOD','SHRI DL AGRAWAL','07/10/1965','MBA');
```

उपरोक्त कमाण्ड से स्टूडेन्ट टेबल के ROLLNO, STUDENTNAME, FNAME, DOB, CLASS फील्डों में क्रमशः 10001,'VINOD','SHRI DL AGRAWAL','07/10/1965','MBA' के मान इन्सर्ट हो जाते है।

इसी प्रकार अन्य रिकॉर्डस भी जोडे जा सकते हैं। अन्य रिकॉर्ड इन्सर्ट करते हुये निम्नलिखित टेबल बनायें।

ROLLNO	STUDENTNAME	FNAME	DOB	CLASS
10001	VINOD	SHRI DL AGRAWAL	07/10/1965	MBA
10002	MANAN	SHRI VINOD AGRAWAL	16/11/1995	V
10003	SURESH	SHRI DL AGRAWAL	25/06/1960	XII
10004	RUBAL	DR. VINAY GOYAL	27/11/1989	XI
10005	DHRUV	SHRI SHIV KMAR	06/11/2000	I
10006	RANJANA	SHRI VINOD AGRAWAL	19/04/1971	Ph.D
10007	RADHIKA	SHRI VINOD AGRAWAL	14/08/2003	PLAY

इसी प्रकार मार्क्स टेबल में भी डाटा इन्सर्ट कर सकते है।

```
INSERT INTO marks
values (10001,90,'Maths');
```

उपरोक्त कमाण्ड से मार्क्स टेबल के ROLLNO, SUBJECT, MARKS फील्डों में क्रमशः 10001,90,'Maths' के मान इन्सर्ट हो जाते हैं ।

इसी प्रकार अन्य रिकॉर्डस भी जोडे जा सकते है। इसी प्रकार रिकॉर्ड इन्सर्ट करते हुये निम्नलिखित टेबल बनायें।

ROLLNO	SUBJECT	MARKS
10001	HINDI	90
10001	ENGLISH	95
10001	MATHS	98
10002	SCIENCE	80
10002	MATHS	99
10002	ENGLISH	90
10002	HINDI	88

3.1.2 SELECT

SELECT कमाण्ड को सामान्यतः MySQL में बहुतायत से प्रयोग किया जाता है। यह किसी भी ऑपरेशनल डाटाबेस से उपयोगकर्ता (users) को वांछित डाटा retrieve करने के लिए उपयोग होता है। हम School database में student table का एक उदाहरण लेते हैं। नीचे वर्णित उदाहरण टेबल

में उपलब्ध समस्त सूचना को दिखाता है। यहाँ asterisk (*) का उपयोंग MySQL में wild card की तरह किया जाता है जिससे कि टेबल में उपलब्ध समस्त सूचना का चयन हो जाता है।

उदाहरण 1

```
.      SELECT *
       FROM student;
```

उपरोक्त कमाण्ड से निम्न परिणाम प्राप्त होगा।

ROLLNO	STUDENTNAME	FNAME	DOB	CLASS
10001	VINOD	SHRI DL AGRAWAL	07/10/1965	MBA
10002	MANAN	SHRI VINOD AGRAWAL	16/11/1995	V
10003	SURESH	SHRI DL AGRAWAL	25/06/1960	XII
10004	RUBAL	DR. VINAY GOYAL	27/11/1989	XI
10005	DHRUV	SHRI SHIV KMAR	06/11/2000	I
10006	RANJANA	SHRI VINOD AGRAWAL	19/04/1971	Ph.D
10007	RADHIKA	SHRI VINOD AGRAWAL	14/08/2003	PLAY

उदाहरण 2

```
SELECT *
       FROM marks;
```

उपरोक्त कमाण्ड से निम्न परिणाम प्राप्त होगा।

ROLLNO	SUBJECT	MARKS
10001	HINDI	90
10001	ENGLISH	95
10001	MATHS	98
10002	SCIENCE	80
10002	MATHS	99
10002	ENGLISH	90
10002	HINDI	88

यदि आप किसी सूचना का केवल कुछ ही हिस्सा चाहते हैं तो आप उसको निम्नलिखित कमाण्ड द्वारा प्राप्त कर सकते हैं। उदाहरण के लिए यदि आप 10005 से कम अथवा बराबर rollno वालों की सूचना student table से चाहते है तो निम्न कमाण्ड द्वारा प्राप्त किया जा सकता है।

SELECT *

FROM Student

WHERE ROLLNO<=10005;

उपरोक्त कमाण्ड से निम्न परिणाम प्राप्त होगा।

ROLLNO	STUDENTNAME	FNAME	DOB	CLASS
10001	VINOD	SHRI DL AGRAWAL	07/10/1965	MBA
10002	MANAN	SHRI VINOD AGRAWAL	16/11/1995	V
10003	SURESH	SHRI DL AGRAWAL	25/06/1960	XII
10004	RUBAL	DR. VINAY GOYAL	27/11/1989	XI
10005	DHRUV	SHRI SHIV KMAR	06/11/2000	I

WHERE clause का उपयोग सूचना को चाहे गये क्राईटेरिया के अनुसार प्राप्त करने के लिए किया जाता है। इस क्लॉज के द्वारा चुने हुए रिकॉर्डो तथा फील्डों को प्राप्त किया जा सकता है। उदाहरण के लिए हमें ऐसे रिकॉर्डों को चुनना है जिनमें मार्क्स फील्ड का मान 90 या इससे अधिक है तो हमें निम्न प्रकार से कमाण्ड लिखना होगा।

SELECT *

FROM marks

WHERE marks >=90;

उपरोक्त कमाण्ड से निम्न परिणाम प्राप्त होगा

ROLLNO	SUBJECT	MARK
10001	HINDI	90
10001	ENGLISH	95
10001	MATHS	98
10002	MATHS	99
10002	ENGLISH	90

इसी प्रकार निम्न उदाहरण में हम student टेबल का वह रिकॉर्ड चुनना चाहते हैं जिसमें STUDENTNAME फील्ड का मान 'MANAN' है।

SELECT * FROM student

WHERE studentname='MANAN';

उपरोक्त कमाण्ड से निम्न परिणाम प्राप्त होगा।

ROLLNO	STUDENTNAME	FNAME	DOB	CLASS
10002	MANAN	SHRI VINOD AGRAWAL	16/11/1995	V

3.1.3 UPDATE

UPDATE command का प्रयोग सामान्यतः किसी पंक्ति के किसी फील्ड के मान को बदलने के लिए किया जाता है। यदि हम प्रत्येक विद्यार्थी को प्रत्येक विषय में 2 अंक बतौर बोनस के देना चाहते हैं तो निम्नलिखित कमाण्ड द्वारा इस कार्य को शीध्र किया जा सकता है।

उदाहरण 1

```
UPDATE marks
SET marks=marks+2;
```

उपरोक्त कमाण्ड से निम्न परिणाम प्राप्त होगा।

ROLLNO	SUBJECT	MARKS
10001	HINDI	92
10001	ENGLISH	97
10001	MATHS	100
10002	SCIENCE	82
10002	MATHS	101
10002	ENGLISH	92
10002	HINDI	100

उपरोक्त परिणाम के फील्ड **MARKS** को देखने पर यह प्रतीत होता है कि प्रत्येक रोल नम्बर के मार्क्स में पूर्व के अंकों में दो अंक जोड़कर परिणाम दिखाया गया।

उदाहरण 2

इसी प्रकार यदि हम पुनः मार्क्स टेबल में दिये गये मार्क्स में से दो अंक कम करना चाहते हैं तथा पूर्व की स्थिति में टेबल को रखना चाहते हैं तो निम्न कमाण्ड द्वारा यह कार्य किया जा सकता है।

```
UPDATE marks
SET marks=marks-2;
```

उपरोक्त कमाण्ड से निम्न परिणाम प्राप्त होगा।

ROLLNO	SUBJECT	MARKS
10001	HINDI	90
10001	ENGLISH	95
10001	MATHS	98
10002	SCIENCE	80
10002	MATHS	99
10002	ENGLISH	90
10002	HINDI	98

उपरोक्त कमाण्ड से मार्क्स फील्ड का परिणाम उदाहरण 1 में दर्शाये गये अंकों से दो अंक कम है।

3.1.4 DELETE

इस कमाण्ड का उपयोग किसी रिकॉर्ड या रिकॉर्ड के समूह को हटाने के लिये किया जाता है। यदि हम rollno 10002 के रिकॉर्ड को हटाना चाहते है तो DELETE command को WHERE clause के साथ प्रयुक्त करके मार्क्स टेबल से हटाया जा सकता है। इसके लिये निम्न कमाण्ड उपयोग में लेगे–

```
DELETE FROM marks
WHERE rollno=10002;
```

इसके पश्चात SELECT * FROM MARKS कमाण्ड को देने पर निम्न परिणाम प्राप्त होगा।

ROLLNO	SUBJECT	MARKS
10001	HINDI	90
10001	ENGLISH	95
10001	MATHS	98

उपरोक्त परिणाम से यह विदित होता है कि टेबल में पूर्व में उपलब्ध रोल नम्बर 10002 का डाटा खत्म हो गया है। इस प्रकार टेबल से डाटा को हटाया जा सकता है।

3.2 व्यू (View):

व्यू का उपयोग टेबल में से डाटा रिट्रीव करने के लिए किया जाता है। क्वैरी वास्तविक डाटा पर कार्य करती हैं जो कि टेबल में स्टोर रहता है। व्यू एक वर्चुअल टेबल होती है जो कि सलेक्ट स्टेटमेन्ट

द्वारा तैयार होती है। व्यू में पंक्ति एवं कॉलम दोनों ही होते हैं जैसा कि टेबल में होते हैं। व्यू में वही फील्डस होते हैं जो फील्ड वास्तविक टेबल में होते हैं। फील्डों की संख्या Select कथन पर निर्भर करती है। व्यू से डाटा को पढना, जोडना एवम् अपडेट किया जाना संभव है। यह अन्य तरीका है जिसके द्वारा एक या एक से अधिक टेबल्स के डाटा को देखा जा सकता है।

व्यू का उपयोग:

- व्यू का सामान्यतः उपयोग अनाधिकृत यूजर से डाटा को संरक्षित करना है।
- पंक्तियों को टेबल में से फिल्टर करता है।

व्यू के लाभ:

- यूजर को आसानी से परिणाम (आउटपुट) देता है।
- यूजर को आसानी से डेटा देता है।
- डवलपर द्वारा डाटा रिट्रीबल को रोका जा सकता है।
- डवलपर द्वारा आसानी से एप्लीकेशन को मेन्टेन किया जा सकता है।

Syntax

CREATE VIEW view_name AS

SELECT column_name(s)

FROM table_name

WHERE condition;

Drop view: व्यू को हटाने के लिए निम्न कमाण्ड उपयोग करते है:

DROP VIEW <viewname>;

3.3 Index:

हमें बहुत बडे डेटा को बिना समय नष्ट किये एक्सेस करना होता है। Index द्वारा टेबल से डेटा को शीघ्रता से एक्सेस करना संभव है। किसी टेबल में Index पंक्तियो को अतिशीघ्रता तथा निपुणता से एक्सेस करने के लिए क्रियेट किये जाते है। किसी भी टेबल में एक या एक से अधिक कॉलम पर Index क्रियेट करना संभव है तथा प्रत्येक Index को नाम देना आवश्यक है। उपयोगकर्ता (Users) Indexes को नहीं देख सकते हैं, वे केवल अपनी क्वैरीज की गति तेज करने के लिए इनका उपयोग करते हैं।

Index के प्रकार:

1. Hash indexes:

Hash indexes तुल्यता खोज (Equality search) के लिए जिसमें कॉलम के मान भिन्न–भिन्न (Unique) हो के लिए बहुत तेज कार्य करते है। अपेक्षित डेटा को प्राप्त करने के लिए इसमें प्राप्त डिस्क ब्लॉक की संख्या बहुत कम (एक अथवा दो) होती है। परन्तु Hash indexes range queries के लिए निपुणता से कार्य नहीं करते है क्योंकि वे clustering indexes नहीं है। एक रेन्ज के अंतर्गत डेटाबेस

आईटम डिस्क पर पास–पास स्टोर नहीं होते क्योंकि उनके Hash मान बहुत भिन्नता लिये हुए हो सकते है।

2. B+tree indexes:

B+tree indexes तुल्यता खोज (Equality search) तथा रेंज क्वैरीज दोनों के लिए निपुणता से कार्य करती है। यद्यपि ये तुल्यता खोज के लिए Hash indexes जितने तेज गतिशील नहीं है।

3.4 डाटाबेस ऑब्जेक्ट्स (Database Objects) :

Database Objects का संधारण RDBMS द्वारा किया जाता है। मुख्यतः सारे ही Database Objects डाटाबेस में जगह घेरते हैं। कुछ डाटाबेस ऑब्जेक्ट दूसरे डाटाबेस के ऑब्जेक्ट्स को भी रखते है। टेबल्स ही ऐसा डाटाबेस ऑब्जेक्ट है जिसमें यूजर का डाटा रखा जाता है और यूजर टेबल्स को सीधे ही एक्सेस कर सकता है। डाटाबेस ऑब्जेक्ट्स का नाम बनाते समय दिया जाता है और इनका नाम नियमानुसार दिया जाता है। सभी डेटाबेस ऑब्जेक्ट्स एक क्रियेटर तथा ऑनर रखते है

निम्नलिखित डाटाबेस ऑब्जेक्ट्स हैं :–

- clusters
- columns
- constraints
- database
- database links
- indexes
- rollback segments
- savepoints
- tables
- tablespaces
- users
- views

डाटाबेस ऑब्जेक्ट के नाम देने के नियम

- ✓ डाटाबेस ऑब्जेक्ट का नाम 64 करेक्टर से ज्यादा लम्बा नहीं होना चाहिए।
- ✓ कोटेशन मार्क नाम में उपयोग नहीं होता है।
- ✓ नाम की शुरूआत अक्षर से होनी चाहिए।
- ✓ नाम MySQL रिजर्व वर्ड्स में से नहीं होना चाहिए।
- ✓ नाम में अक्षर जैसे कि A-Z,0-9,_,#,$ हो सकते हैं। (#,& का उपयोग नहीं किया जाना चाहिऐ)

✓ उसी नाम का पुनः उपयोग नहीं किया जाना चाहिए यदि पूर्व में उसी ऑनर द्वारा किसी डाटाबेस ऑब्जेक्ट का नाम उपयोग में ले लिया गया है।

अन्य जानकारियां :–

- पूरे एवम् वर्णात्मक नाम का उपयोग करें।
- एक वचनीय (Singular) नाम का उपयोग करें।
- किसी भी एक एन्टिटी को दर्शाने के लिये एक ही नाम का प्रयोग करें।

SQL/MySQL DDL/DML Syntax

Statement	Syntax
AND / OR	SELECT column_name(s) FROM table_name WHERE condition AND|OR condition
ALTER TABLE (add column)	ALTER TABLE table_name ADD column_name datatype
ALTER TABLE (drop column)	ALTER TABLE table_name DROP COLUMN column_name
AS (alias for column)	SELECT column_name AS column_alias FROM table_name
AS (alias for table)	SELECT column_name FROM table_name AS table_alias
BETWEEN	SELECT column_name(s) FROM table_name WHERE column_name BETWEEN value1 AND value2
CREATE DATABASE	CREATE DATABASE database_name
CREATE INDEX	CREATE INDEX index_name ON table_name (column_name)
CREATE TABLE	CREATE TABLE table_name

	(column_name1 data_type,
	column_name2 data_type,.......)
CREATE UNIQUE INDEX	CREATE UNIQUE INDEX index_name
	ON table_name (column_name)
CREATE VIEW	CREATE VIEW view_name
	ASSELECT column_name(s)
	FROM table_name
	WHERE condition

DELETE FROM   DELETE FROM table_name
(**Note:** Deletes the entire table!!) *or*

DELETE FROM table_name

	WHERE condition
DROP DATABASE	DROP DATABASE database_name
DROP INDEX	DROP INDEX table_name.index_name
DROP TABLE	DROP TABLE table_name
GROUP BY	SELECT column_name1,SUM(column_name2)
	FROM table_name
	GROUP BY column_name1
HAVING	SELECT column_name1, SUM(column_name2)
	FROM table_name
	GROUP BY column_name1
	HAVING SUM(column_name2) condition value
IN	SELECT column_name(s)
	FROM table_name
	WHERE column_nameIN (value1,value2,..)
INSERT INTO	INSERT INTO table_name
	VALUES (value1, value2,....) *or*
	INSERT INTO table_name
	(column_name1, column_name2,...)
	VALUES (value1, value2,....)
LIKE	SELECT column_name(s)
	FROM table_name
	WHERE column_name
	LIKE pattern

ORDER BY	SELECT column_name(s) FROM table_name ORDER BY column_name [ASC|DESC]
SELECT	SELECT column_name(s) FROM table_name
SELECT *	SELECT * FROM table_name
SELECT DISTINCT	SELECT DISTINCT column_name(s) FROM table_name
SELECT INTO (used to create backup copies of tables)	SELECT * INTO new_table_name FROM original_table_name *or* SELECT column_name(s) INTO new_table_name FROM original_table_name
TRUNCATE TABLE(deletes only the data inside the table)	TRUNCATE TABLE table_name
UPDATE	UPDATE table_name SET column_name=new_value [, column_name=new_value] WHERE column_name=some_value
WHERE	SELECT column_name(s) FROM table_name WHERE condition

महत्वपूर्ण बिन्दु

डाटा मेन्यूपुलेशन लैंग्वेज (DML)

- **SELECT** - Database में संग्रहित सूचना का retrieval.
- **UPDATE** - Database में संग्रहित सूचनाओं को modify करना।
- **DELETE** - Database में से सूचनाओं को delete करना।
- **INSERT INTO** - Database में नयी सूचनाओं का insertion.

View- व्यू का उपयोग टेबल में से डाटा रिट्रीव करने के लिए किया जाता है। क्वैरी वास्तविक डाटा पर कार्य करती हैं जो कि टेबल में स्टोर रहता है। व्यू एक वर्चुअल टेबल होती है जो कि सलेक्ट स्टेटमेन्ट द्वारा तैयार होती है। व्यू में पंक्ति एवं कॉलम दोनों ही होते हैं जैसा कि टेबल में होते हैं। व्यू में वही फील्डस

होते हैं जो फील्ड वास्तविक टेबल में होते हैं। फील्डों की संख्या Select कथन पर निर्भर करती है। व्यू से डाटा को पढना, जोडना एवम् अपडेट किया जाना संभव है।

व्यू का उपयोगः

- व्यू का सामान्यतः उपयोग अनाधिकृत यूजर से डाटा को संरक्षित करना है।
- पंक्तियों को टेबल में से फिल्टर करता है।

व्यू के लाभः

- यूजर को आसानी से परिणाम देता है।
- यूजर को आसानी से डाटा देता है।
- डवलपर द्वारा डाटा रिट्रीबल को रोका जा सकता है।
- डवलपर द्वारा आसानी से एप्लीकेशन को मेन्टेन किया जा सकता है।

Index: हमें बहुत बडे डेटा को बिना समय नष्ट किये एक्सेस करना होता है। Index द्वारा टेबल से डेटा को शीघ्रता से एक्सेस करना संभव है। किसी टेबल में Index पंक्तियो को अतिशीघ्रता तथा निपुणता से एक्सेस करने के लिए क्रियेट किये जाते है। किसी भी टेबल में एक या एक से अधिक कॉलम पर Index क्रियेट करना संभव है तथा प्रत्येक Index को नाम देना आवश्यक है। उपयोगकर्ता (Users) Indexes को नहीं देख सकते हैं, वे केवल अपनी क्वैरीज की गति तेज करने के लिए इनका उपयोग करते हैं।

Database Objects का संधारण RDBMS. द्वारा किया जाता है। मुख्यतः सारे ही Database Objects जगह डाटाबेस में घेरते है। कुछ डाटाबेस ऑब्जेक्ट दूसरे डाटाबेस के ऑब्जेक्ट्स को भी रखते है। टेबल्स ही ऐसा डाटाबेस ऑब्जेक्ट है जिसमें यूजर का डाटा रखा जाता है और यूजर टेबल्स को सीधा एक्सेस कर सकता है। डाटाबेस ऑब्जेक्ट्स का नाम बनाते समय दिया जाता है और इनका नाम लिस्ट के नियमानुसार दिया जाता है। प्रत्येक डाटाबेस ऑब्जेक्ट का बनाने वाला और मालिक होता है। निम्नलिखित डाटाबेस ऑब्जेक्ट्स हैं :–

- clusters
- columns
- constraints
- database
- database links
- indexes
- rollback segments
- savepoints
- sequences
- synonyms
- tables
- tablespaces
- users
- views

अभ्यासार्थ प्रश्न

वस्तुनिष्ठ प्रश्न

1 डी.एम.एल. के द्वारा आप क्या कर सकते हैं?

(अ) डाटा को एन्टर एवम् एडिट (ब) रिपोर्ट्स बना सकते है

(स) उपरोक्त दोनों कार्यो के लिए (द) उपरोक्त में से कोई नहीं

अति लघुत्तरात्मक प्रश्न

1 सलेक्ट कमाण्ड का क्या उपयोग है?

2 अपडेट कमाण्ड का क्या उपयोग है?

3 डिलीट कमाण्ड का क्या उपयोग है?

4 टेबल का स्ट्रक्चर देखने हेतु कौनसी कमाण्ड का उपयोग करेगें ?

5 टेबल,डेटाबेस तथा कॉलम के नाम की अधिकतम लम्बाई कितने अक्षरों की है?

लघुत्तरात्मक प्रश्न

1 इन्सर्ट इन टू कमाण्ड का क्या उपयोग है?

2 व्यू क्या होता है?

3 किसी टेबल के समस्त रिकॉर्डस को प्रदर्शित करने के लिए कमाण्ड लिखें?

4 MySQL द्वारा उपयोग किये जाने वाले indexes के प्रकारों का वर्णन कीजिए?

निबन्धात्मक प्रश्न

1 टेबल्स में से कुछ डाटा को कैसे रिट्रीव किया जाता है? उदाहरण सहित वर्णन करें।

2 डी.एम.एल. कमाण्ड्स का विस्तृत वर्णन कीजिए।

3 डाटाबेस ऑब्जेक्ट्स के नाम देने के नियमों का वर्णन कीजिए।

4 VIEW को उदाहरण सहित समझाइये?

उत्तरमाला

1 अ

4

माईएसक्यूएल फंक्शन्स
MySQL Functions

4.1 परिचय

Functions, SQL/MySQL की मुख्य विशेषता है और इसे निम्नलिखित कार्यों को करने के लिये प्रयुक्त किया जा सकता है।

- ✓ डाटा पर विभिन्न गणनायें करने के लिए
- ✓ किसी विशिष्ट डाटा सैट अथवा डाटा सैट के समूह को परिवर्तित करने के लिए।
- ✓ किसी विशिष्ट पंक्ति अथवा पंक्तियों के समूह को परिर्वतित करने के लिए।
- ✓ dates and numbers के प्रदर्शन के लिए फारमेटिंग करना।
- ✓ data types के कन्वर्जन के लिए।

Function तथा Operator समान रूप से कार्य करते हैं। दोनों ही एक अथवा अधिक परिणाम दर्शाते हैं और शून्य अथवा अधिक argument लेते हैं।

4.2 संख्यात्मक फंक्शन (Numeric Functions) :

यह केवल numeric input लेते हैं तथा numeric output देते हैं इन functions को select statement तथा अन्य statements के साथ प्रयुक्त कर सकते हैं।

ABS() Function

फंक्शन का उपयोग किसी भी संख्या की निर्पेक्ष मान निकालने के लिए किया जाता है।

जैसे कि SELECT ABS(-30);

इसका परिणाम 30 प्राप्त होगा।

CEIL() Function

इस फंक्शन का उपयोग किसी भी संख्या का न्यूनतम इंटीजर मान लेने के लिये करते है जो कि दी हुई संख्या के बराबर या अधिक होता है। जैसे कि SELECT CEIL(10.6);

इसका परिणाम 11 प्राप्त होगा।

FLOOR() Function

इस फंक्शन का उपयोग किसी भी संख्या का अधिकतम इंटीजर मान लेने के लिये करते है जो कि दी हुई संख्या के बराबर या कम होता है। जैसे कि SELECT FLOOR(10.6);

इसका परिणाम 10 प्राप्त होगा।

SELECT FLOOR(-10.6);

इसका परिणाम –11 प्राप्त होगा।

MOD() Function

इस फंक्शन का उपयोग शेषफल निकालने के लिये होता है जो कि प्रथम संख्या में द्वितीय संख्या के भाग देने के पश्चात शेष रहती है। जैसे कि SELECT MOD(25,10) ;

इसका परिणाम 5 प्राप्त होगा।

POWER(m,n) Function या POW(m,n)

इस फंक्शन का उपयोग संख्या (m)के घात (n) का परिणाम प्राप्त करने के लिए करते है। जैसे कि SELECT POWER(4,2) ;

इसका परिणाम 16 प्राप्त होगा।

ROUND(n[,m]) Function

इस फंक्शन का उपयोग किसी भी संख्या (n) को (m) स्थान तक राउण्ड करने के लिये होता है। इसमें m का इंटीजर होना आवश्यक है। जैसे कि SELECT ROUND(10.645,2) ;

इसका परिणाम 10.64 प्राप्त होगा।

SIGN(n) Function

फंक्शन का उपयोग किसी भी संख्या (n) का मान –1 देता है यदि n का मान शून्य से कम होता है, n का मान 0 देता है यदि n का मान 0 हो और 1 देता है यदि n का मान 0 से अधिक हो। जैसे कि SELECT SIGN(-30) ;

इसका परिणाम –1 प्राप्त होगा।

SQRT(n) Function

फंक्शन का उपयोग किसी भी संख्या (n) के वर्गमूल का मान देता है। जैसे कि

SELECT SQRT(16) ;

इसका परिणाम 4 प्राप्त होगा।

TRUNCATE(n[,m]) Function

फंक्शन का उपयोग किसी भी संख्या (n) को दशमलव के m स्थान तक छोडकर पश्चात की समस्त दशमलव बिन्दुओं को हटा देता है। जैसे कि

SELECT TRUNCATE(10.645,2) ;

इसका परिणाम 10.64 प्राप्त होगा।

SELECT TRUNCATE(10.645,0);

इसका परिणाम 10 प्राप्त होगा।

4.3 स्ट्रिंग फंक्शन (String Functions) :

यह केवल character input लेते है तथा numeric व character output देते हैं इन functions को select statement तथा अन्य statements के साथ प्रयुक्त कर सकते है। इन फंक्शन्स को नीचे वर्णित स्टूडेन्ट टेबल का उदाहरण लेकर समझाया गया है।

ROLLNO	STUDENTNAME	FNAME	DOB	CLASS
10001	VINOD	SHRI DL AGRAWAL	07/10/1965	MBA
10002	MANAN	SHRI VINOD AGRAWAL	16/11/1995	V
10003	SURESH	SHRI DL AGRAWAL	25/06/1960	XII
10004	RUBAL	DR. VINAY GOYAL	27/11/1989	XI
10005	DHRUV	SHRI SHIV KMAR	06/11/2000	I
10006	RANJANA	SHRI VINOD AGRAWAL	19/04/1971	Ph.D
10007	RADHIKA	SHRI VINOD AGRAWAL	14/08/2003	PLAY

4.3.1 Case Conversion function:

LOWER() फलन (Function)

इस फंक्शन द्वारा किसी भी शब्द या पंक्ति के सभी अक्षरों को छोटे अक्षरों में बदला जाता है।

उदाहरणार्थः SELECT LOWER(FNAME) FROM STUDENT WHERE ROLLNO=10005;

उपरोक्त कमाण्ड के द्वारा स्टूडेन्ट टेबल से FNAME पर यह फंक्शन लगाया गया है साथ ही एक शर्त यह लगाई गयी है कि परिणाम केवल उसी का दे जिसका ROLLNO 10005 हो। इसका परिणाम निम्न प्राप्त होगा।

FNAME

shri shiv kumar

UPPER() फलन (Function)

इस फंक्शन द्वारा किसी भी शब्द या पंक्ति के सभी अक्षरों को बड़े अक्षरों में बदला जाता है।

उदाहरणार्थः SELECT UPPER (FNAME) FROM STUDENT WHERE ROLLNO=10003;

उपरोक्त कमाण्ड के द्वारा स्टूडेन्ट टेबल से FNAME पर यह फंक्शन लगाया गया है साथ ही एक शर्त यह लगाई गयी है कि परिणाम केवल उसी का दे जिसका ROLLNO 10003 हो। इसका परिणाम निम्न प्राप्त होगा।

FNAME
SHRI DL AGRAWAL

4.3.2 String Manipulation function

Concat() function

इस फंक्शन का उपयोग विभिन्न फील्डो से प्राप्त परिणामों को जोडने (Combine) के लिए किया जाता है।

CONCAT(str1, str2, str3, ...);

उदाहरणार्थः

SELECT CONCAT('My', 'S', 'QL');
Output:
MySQL

INSTR(STR, SUBSTR) फलन (Function)

इस फंक्शन से किसी भी शब्द/पंक्ति में चाहे गये अक्षर की प्रथम बार उपस्थिति की स्थिति का पता लगाया जाता है।

उदाहरणार्थः

SYNTAX	EXAMPLE	RESULT
INSTR(STR, SUBSTR)	INSTR('HARDDISK', 'DISK')	5
	INSTR('COMPUTER', 'HARDDISK')	0

LENGTH() फलन (Function)

इस फंक्शन द्वारा किसी भी शब्द/पंक्ति की लम्बाई बताई जाती है।

उदाहरणार्थः

SYNTAX	EXAMPLE	RESULT
LENGTH(char)	LENGTH('alpha')	5

उपरोक्त उदाहरण पंक्ति की कुल लम्बाई बताता है।

LPAD() फलन (Function)

इस फंक्शन द्वारा बायीं तरफ अतिरिक्त अक्षर या अक्षरों के समूह की भराई (insert) की जाती है।

उदाहरणार्थः

SYNTAX	EXAMPLE	RESULT
LPAD(char1,n[,char2])	LPAD('alpha',10,'*')	*****alpha

उपरोक्त उदाहरण में char1 के बायीं तरफ अतिरिक्त अक्षर char2 की भराई की गयी है तथा char1 की कुल लम्बाई n(10) है।

LTRIM() फलन (Function)

यह फंक्शन किसी भी शब्द के पहले के खाली स्थानों को हटा देता है।

उदाहरणार्थः

SYNTAX	EXAMPLE	RESULT
LTRIM(STR)	LTRIM(' HOLIDAY')	'HOLIDAY'

उपरोक्त उदाहरण में char के बायीं तरफ के अक्षर set से तीन स्थान तक मिलते है। जिनको उपरोक्त फंक्शन द्वारा हटाया गया है।

REPLACE() फलन (Function)

यह फंक्शन पंक्ति के सभी अक्षरों को ऑप्शन 2 से बदल देता है जो अक्षर पंक्ति में ऑप्शन 1 से मिलते है।

उदाहरणार्थः

SYNTAX	EXAMPLE	RESULT
REPLACE(line,option1[,option2])	REPLACE ('man & woman','a','e')	men & women

उपरोक्त उदाहरण में line मे स्थित option1 में उपलब्ध अक्षर को option2 के अक्षर से बदला गया है।

RPAD() फलन (Function)

इस फंक्शन द्वारा दायीं तरफ अतिरिक्त अक्षर या अक्षरों के समूह की भराई की जाती है।

उदाहरणार्थः

SYNTAX	EXAMPLE	RESULT
RPAD(char1,n[,char2])	RPAD('alpha',10,'*')	alpha*****

उपरोक्त उदाहरण में char1 के दायीं तरफ अतिरिक्त अक्षर char2 की भराई की गयी है तथा char1 की कुल लम्बाई n(10) है।

RTRIM() फलन (Function)

यह फंक्शन किसी भी शब्द के अन्त में स्थित खाली स्थानों को हटा देता है।

उदाहरणार्थः

SYNTAX	EXAMPLE	RESULT
SELECT RTRIM(STR)	RTRIM('HOLIDAY	') 'HOLIDAY'

SUBSTR() फलन (Function)

इस फंक्शन द्वारा स्ट्रिंग में से सब–स्ट्रिंग निकाली जाती है।

उदाहरणार्थः

SYNTAX	EXAMPLE	RESULT
SUBSTR(char,m[,n])	SUBSTR('RAMSITA',2,3)	AMS

उपरोक्त उदाहरण में char के स्थान 2 (m) से सबस्ट्रिंग की लम्बाई 3 (n) अक्षर की लेकर परिणाम दिया गया है।

उदाहरणार्थः

SELECT SUBSTRING('MATHEMATICS',5,6);

RESULT - 'EMATICS'

4.4 Date Functions :

सभी date functions द्वारा date data type परिणाम दिया जाता है केवल month_between function को छोडकर जो कि अंकीय संख्या में परिणाम देता है।

LAST_DAY() फलन (Function)

इस फंक्शन द्वारा बताये गये माह की अन्तिम तिथि को ज्ञात किया जाता है।

उदाहरणार्थः

SYNTAX	EXAMPLE	RESULT
LAST_DAY(n)	LAST_DAY(SYSDATE)	31-JAN-2006

उपरोक्त उदाहरण में SYSDATE जो कि जनवरी माह सन् 2006 की है द्वारा जनवरी 2006 के माह की अन्तिम तिथि जो कि 31 जनवरी 2006 ज्ञात की गयी है।

SELECT LAST_DAY('2003-05-32')

RESULT NULL

CURDATE() फलन (Function)

यह फंक्शन द्वारा वर्तमान तारीख को 'YYYY-MM-DD' या 'YYYYMMDD' फोरमेट में प्रदर्शित करता है।

उदाहरणार्थः

```
select curdate();
2009-01-03
```

CURTIME() :

यह फंक्शन द्वारा वर्तमान समय को 'HH:MM:SS' or 'HHMMSS' फोरमेट में प्रदर्शित करता है।

उदाहरणार्थः

```
select curtime();
18:38:09
```

DATEDIFF(expression1,expression2) :

expression1 और expression2 दोनो दिनांक या दिनांक समय फोरमेट में होते हैं। यह फंक्शन दोनो दिनांको के अन्तर को दिनों की संख्या के रूप में बताता है। केवल दिनांक वाला भाग ही गणना में काम आता है।

उदाहरणार्थः

select datediff('2007-2-10 17:33:25','2007-1-1');
40

DATE_ADD(datetime, INTERVAL expression datetimetype)

यह फंक्शन दिये गये मान (INTERVAL) को दिनांक (datetime) में जोड़ कर प्रदर्शित करता है।

select date_add('2007-1-13', interval 15 day);
2007-01-28

ADDDATE('2008-1-10',INTERVAL 10 DAYS);
'2008-01-20'
ADDDATE()तथा DATE_ADD functions समानार्थक है।

SUBDATE(*date*,INTERVAL *expr unit*), SUBDATE(*expr*,*days*)

यह फंक्शन दिये गये मान (INTERVAL) को दिनांक (datetime) में से घटा कर प्रदर्शित करता है।

SUBDATE()rFkk DATE_SUB functions lekukFkZd gSA

select date_sub('2007-1-14', interval 10 day);
'2007-01-04'

DAYNAME(date) :

यह फंक्शन दी गई दिनांक के दिन के नाम को प्रदर्शित करता है।

select dayname('2010-04-22');
Thursday

DAYOFMONTH(date) or DAY(date) :

यह फंक्शन दी गई दिनांक के दिन को अंक के रूप में (1 से 31 के बीच) प्रदर्शित करता है। फंक्शन DAY() तथा DAYOFMONTH() समानार्थक है।

select dayofmonth('2007-01-04');
4

DAYOFWEEK(date) :

यह फंक्शन दी गई दिनांक के सप्ताह के दिन को अंक के रूप में (1 से 7 के बीच) प्रदर्शित करता है।

```
select dayofweek('2007-01-03');
4
```

DAYOFYEAR(date) :

यह फंक्शन दी गई दिनांक को वर्ष के दिन के रूप में (1 से 366 के बीच) प्रदर्शित करता है।

```
select dayofyear('2007-07-09');
190
```

MONTH(date) :

यह फंक्शन दी गई दिनांक के महीने को अंक के रूप में (1 से 12 के बीच) प्रदर्शित करता है।

```
select month('2007-09-09');
9
```

MONTHNAME(date) :

यह फंक्शन दी गई दिनांक के महीने के नाम को प्रदर्शित करता है।

```
select monthname('2007-09-09');
September
```

NOW() :

यह फंक्शन वर्तमान दिनांक व समय को प्रदर्शित करता है। जिसका फोरमेट 'YYYY-MM-DD HH:MM:SS' or YYYYMMDDHHMMSS gSA

```
select now();
2007-01-04  14:56:15
```

STR_TO_DATE(*str*,*format*)

यह फंक्शन str में दी गई पंक्ति को format में दिये गये फोरेमेट के अनुसार दिनांक या समय या दिनांक तथा समय के रूप में प्रदर्शित करता है। यदि str पंक्ति में दिनांक या समय के अनुरूप मान नहीं है तो null मान प्रदर्शित होता है। यह फंक्शन **DATE_FORMAT()**के विपरीत है।

```
SELECT STR_TO_DATE('01,5,2003','%D,%M,%Y');

    '2013-05-01'
```

उपरोक्त उदाहरण के फोरमेट में %D दिन को ,%M महीने को तथा %Y साल को प्रदर्शित करता है।

DATE_FORMAT() function

यह फंक्शन दिनांक तथा समय के डेटा को विभिन्न फोरमेट में प्रदर्शित करता है।

DATE_FORMAT(date,format)

उपरोक्त syntax में date एक मान्य दिनांक है तथा format दिनांक को प्रदर्शित करने का फोरमेट है।

Format	Description
%b	महीने के नाम शब्दों में
%d	महीने का दिन अंको में (00-31)
%m	महीना अंको में (00-12)
%Y	वर्ष चार अंको में

Example: DATE_FORMAT(NOW(),'%m-%d-%Y')

Result : 11-04-2008

Example: DATE_FORMAT(NOW(),'%d %b %y')

Result : 04 Nov 08

4.5 Data type conversion functions

ये फंक्शन दिये गये मान को विभिन्न उपयोगों के लिए एक डेटा टाईप से दूसरे डेटा टाईप में बदलने के काम आते हैं। यह फंक्शन select तथा अन्य कथनों (statements) के साथ उपयोग में आते है।

4.5.1 इम्पलिसिट (Implicit) रूपान्तरण (Conversion):

यह रूपांतरण वे है जो बिना CAST या CONVERT फंक्शन के काम में आते है।

उदाहरणार्थः एक पंक्ति (string) को पूर्ण संख्या में बदलना

SELECT 1+'12';

13

यहां पंक्ति (string) '12' को संख्या(Numeric) में बदल कर पद (expression) का परिणाम 13 भी संख्या में प्राप्त हुआ है।

उदाहरण : संख्या का पंक्ति में रूपातंरण

SELECT CONCAT(1,'ALPHA');

'1 ALPHA'

4.5.2 एक्सप्लिसिट (Explicit) रूपान्तरण

ये रूपातंरण वे है जो CAST या CONVERT फंक्शन को काम में लेकर किये जाते है।

CAST(*expr* AS *type*) फंक्शन

<u>CAST()</u> फंक्शन एक डेटा टाइप के मान को लेकर दूसरे डेटा टाइप के मान में बदलता है।

उदाहरण : संख्या का पंक्ति में रूपातंरण या Casting

```
SELECT 12, CAST(12 as CHAR);

        12, '12'
```

CONVERT(expr,type) फंक्शन

CONVERT एक पद से मान लेकर दूसरे डेटा टाइप में बदल देता है।

```
1)SELECT ('1999-05-15 00:00:00' AS date) as DATE1
Date1
1999-05-15
2)SELECT ('99-05-15 00:00:00' AS date) as DATE2
Date2
1999-05-15
```

4.6. Aggregate फंक्शन :

ये फंक्शन कई records से मान लेकर परिणाम की गणना करने के काम में आते है। ये मुख्य पांच प्रकार के होते हैं।

1. COUNT(*x*) x की भरी हुई मानों की गणना करता हैं।
2. SUM(*x*) *x* के मानों को जोडता है।
3. AVG(*x*) *x* के मानों का औसत निकालता है, null values की उपेक्षा करते हुए
4. MIN(*x*) *x* के न्यूनतम मान को प्रदर्शित करता है।
5. MAX(*x*) *x* के अधिकतम मान को प्रदर्शित करता है।

उदाहरणार्थः

1. SELECT COUNT(*) FROM marks WHERE marks >90;

   5
2. SELECT sum(marks) as 'TOTAL' FROM marks ;

WHERE rollno=10002;

TOTAL

357

3. SELECT AVG(marks) FROM marks;

91.43

4. SELECT MIN(marks) FROM marks;

80

5. SELECT MAX(marks) FROM marks;

99

ग्रुप फंक्शन (Group Functions) :

ग्रुप फंक्शन द्वारा प्राप्त परिणाम एक या एक से अधिक पंक्तियों के समूह पर निर्भर करता है। यह फंक्शन select तथा अन्य कथनों (statements) के साथ उपयोग में आते है।

रिकॉर्डस का समूह बनाना :

Group by Clause -

The GROUP BY clause द्वारा किसी टेबल को एक या अधिक उपसमूह में मिला करके, उस उपसमूह में कोई भी expr समान होता है और उस उपसमूह में कोई मान होता है। GROUP BY clause में एक से अधिक कॉलम हो सकते हैं। उदाहरणार्थ नीचे दिया गया है कि विद्यार्थियों को कक्षा, वर्ग, एवम् विषयानुसार उप समूह में बनाया जा सकता है–

SYNTAX

SELECT * from student GROUP BY CLASS

4.7 सब क्वैरी (Sub Query):

सब क्वैरी SQL/MySQL का विकसित रूप हैं तथा MySQL द्वारा समर्थित है। सब क्वैरी में एक क्वैरी का परिणाम दूसरी क्वैरी का भाग होता हैं। यह वास्तव में एक select statement के अन्दर दूसरा select, insert, update या delete statement होता है।

Subquery महत्तवपूर्ण है क्योंकि -

- Subquery द्वारा यह आसान हो जाता है किसी भी क्वैरी को टुकडों में विभक्त करना एवं टुकडों को पुनः जोडकर एक करना।
- कुछ क्वेरीज को बिना सबक्वेरी का उपयोग किये MySQL में परिभाषित नहीं किया जा सकता है।

उदाहरणार्थ

SELECT * FROM MARKS

WHERE ROLLNO = (SELECT ROLLNO FROM student WHERE SUDENTNAME='MANAN');

उक्त उदाहरण में हमने sudent table में से मनन का रोल नम्बर लिया है तथा प्राप्त परिणाम की तुलना marks table के रोल नम्बर से की है क्योंकि marks table में नाम उपलब्ध नहीं है। इसलिये हमने नाम के आधार पर रोल नम्बर को student table से लिया है तथा रोल नम्बर को marks table मे इनपुट की तरह प्रयुक्त किया हे। प्राप्त परिणाम नीचे दिया गया है।

ROLLNO	SUBJECT	MARKS
10002	SCIENCE	90
10002	MATHS	85
10002	ENGLISH	90
10002	HINDI	85

एकाधिक टेबल्स से डाटा का प्रदर्शन करना/ प्राप्त करना

हमारे पास दो टेबल्स student तथा marks परिशिष्ट में दिये गये डाटानुसार है–

दोनों में ही अद्वितीय key rollno है। निम्न क्वेरी की सहायता से डाटा का डिस्पले दोनों टेबल से किया जा सकता है

SELECT student.rollno, student.studentname, marks.marks, marks.subject FROM student, marks WHERE student.rollno=marks.rollno;

प्राप्त परिणाम नीचे दिया गया है।

rollno	studentname	marks	subject
10001	VINOD	60	HINDI
10001	VINOD	75	ENGLISH
10001	VINOD	55	MATHS
10002	MANAN	90	SCIENCE
10002	MANAN	85	MATHS
10002	MANAN	90	ENGLISH
10002	MANAN	85	HINDI

इस प्रकार एकाधिक टेबल से डाटा का डिस्पले किया जा सकता है।

हम comparison operators का प्रयोग शर्तों में कर सकते हैं।

1. = equal to
2. > greater than
3. < less than
4. >= greater than or equal to
5. <= less than or equal to
6. <> not equal to
7. LIKE string comparison test

comparison operators के उदाहरण ऊपर दिये गये है।

1. SELECT * FROM student WHERE ROLLNO=10004;

ROLLNO	STUDENTNAME	FNAME	DOB	CLASS
10004	RUBAL	DR. VINAY GOYAL	27/11/1989	XI

2. SELECT * FROM student WHERE ROLLNO>10005;

ROLLNO	STUDENTNAME	FNAME	DOB	CLASS
10006	RANJANA	SHRI VINOD AGRAWAL	19/04/1971	Ph.D
10007	RADHIKA	SHRI VINOD AGRAWAL	14/08/2003	PLAY

3. SELECT * FROM student WHERE ROLLNO<10004;

ROLLNO	STUDENTNAME	FNAME	DOB	CLASS
10001	VINOD	SHRI DL AGRAWAL	07/10/1965	MBA
10002	MANAN	SHRI VINOD AGRAWAL	16/11/1995	V
10003	SURESH	SHRI DL AGRAWAL	25/06/1960	XII

4. SELECT * FROM student WHERE ROLLNO<>10004;

ROLLNO	STUDENTNAME	FNAME	DOB	CLASS
10001	VINOD	SHRI DL AGRAWAL	07/10/1965	MBA
10002	MANAN	SHRI VINOD AGRAWAL	16/11/1995	V
10003	SURESH	SHRI DL AGRAWAL	25/06/1960	XII
10005	DHRUV	SHRI SHIV KMAR	06/11/2000	I
10006	RANJANA	SHRI VINOD AGRAWAL	19/04/1971	Ph.D
10007	RADHIKA	SHRI VINOD AGRAWAL	14/08/2003	PLAY

5. SELECT * FROM student WHERE ROLLNO>=10004;

ROLLNO	STUDENTNAME	FNAME	DOB	CLASS
10004	RUBAL	DR. VINAY GOYAL	27/11/1989	XI
10005	DHRUV	SHRI SHIV KMAR	06/11/2000	I
10006	RANJANA	SHRI VINOD AGRAWAL	19/04/1971	Ph.D
10007	RADHIKA	SHRI VINOD AGRAWAL	14/08/2003	PLAY

6. SELECT * FROM student WHERE ROLLNO<=10004;

ROLLNO	STUDENTNAME	FNAME	DOB	CLASS
10001	VINOD	SHRI DL AGRAWAL	07/10/1965	MBA
10002	MANAN	SHRI VINOD AGRAWAL	16/11/1995	V
10003	SURESH	SHRI DL AGRAWAL	25/06/1960	XII
10004	RUBAL	DR. VINAY GOYAL	27/11/1989	XI

7. SELECT * FROM student WHERE FNAME LIKE 'SHRI VI%';

ROLLNO	STUDENTNAME	FNAME	DOB	CLASS
10002	MANAN	SHRI VINOD AGRAWAL	16/11/1995	V
10006	RANJANA	SHRI VINOD AGRAWAL	19/04/1971	Ph.D
10007	RADHIKA	SHRI VINOD AGRAWAL	14/08/2003	PLAY

Mathematical operators

1. + Addition
2. - Subtraction
3. * Multiplication
4. / Division
5. % Modulo

Mathematical operators के उदाहरण दिये गये हैं

1. SELECT * FROM marks where (marks+2) =100;

ROLLNO	SUBJECT	MARKS
10001	MATHS	98

2. SELECT * FROM marks where (marks-2) >90;

ROLLNO	SUBJECT	MARKS
10001	ENGLISH	95
10001	MATHS	98
10002	MATHS	99

3. SELECT * FROM marks where marks*2<=180;

ROLLNO	SUBJECT	MARKS
10001	HINDI	90
10002	SCIENCE	80
10002	ENGLISH	90
10002	HINDI	88

4. SELECT * FROM marks where marks/2>45;

ROLLNO	SUBJECT	MARKS
10001	ENGLISH	95
10001	MATHS	98
10002	MATHS	99

5. SELECT * FROM marks where (marks+2) =100;

ROLLNO	SUBJECT	MARKS
10001	HINDI	90
10001	ENGLISH	95
10001	MATHS	98
10002	SCIENCE	80
10002	MATHS	99
10002	ENGLISH	90
10002	HINDI	88

DISTINCT keyword का प्रयोग करके दोहराव को हटाया जा सकता है।

उदाहरणार्थ

1) SELECT DISTINCT rollno FROM marks;

ROLLNO

10001

10002

USE AND, OR TO SPECIFY MULTIPLE CONDITIONS

उदाहरणार्थ

2) SELECT *

FROM marks

WHERE marks > 90

OR rollno ='10002';

ROLLNO	SUBJECT	MARKS
10001	ENGLISH	95
10001	MATHS	98
10002	SCIENCE	80
10002	MATHS	99
10002	ENGLISH	90
10002	HINDI	88

3) SELECT *
FROM marks
WHERE subject='HINDI'
AND marks < = 90;

ROLLNO	SUBJECT	MARKS
10001	HINDI	90
10002	HINDI	88

4) SELECT *
FROM marks
WHERE marks BETWEEN 95 AND 99;

ROLLNO	SUBJECT	MARKS
10001	ENGLISH	95
10001	MATHS	98
10002	MATHS	99

Sorting of data

5) SELECT *
FROM *student*
ORDER BY *studentname;*

* By default रिकार्डस को बढते हुये क्रम में व्यवस्थित किया जा सकता है।

ROLLNO	**STUDENTNAME**	**FNAME**	**DOB**	**CLASS**
10005	DHRUV	SHRI SHIV KUMAR	2000-07-01	I
10002	MANAN	SHRI VINOD AGARWAL	1995-10-16	V
10007	RADHIKA	SHRI VINOD AGARWAL	2003-08-14	PLAY
10006	RANJANA	SHRI VINOD AGARWAL	1965-07-10	Ph.D
10004	RUBAL	VINAY GOYAL	1989-11-27	XI
10003	SURESH	SHRI DL AGARWAL	1960-06-25	XII
10001	VINOD	SHRI DL AGARWAL	1965-07-10	MBA

ASC and DESC keywords का प्रयोग ascending अथवा descending क्रम में व्यवस्थित करने के लिये किया जाता है।

उदाहरणार्थ

6) SELECT *
   FROM student
   ORDER BY studentname DESC;

ROLLNO	STUDENTNAME	FNAME	DOB	CLASS
10001	VINOD	SHRI DL AGARWAL	1965-07-10	MBA
10003	SURESH	SHRI DL AGARWAL	1960-06-25	XII
10004	RUBAL	VINAY GOYAL	1989-11-27	XI
10006	RANJANA	SHRI VINOD AGARWAL	1965-07-10	Ph.D
10007	RADHIKA	SHRI VINOD AGARWAL	2003-08-14	PLAY
10002	MANAN	SHRI VINOD AGARWAL	1995-10-16	V
10005	DHRUV	SHRI SHIV KUMAR	2000-07-01	I

GROUP BY प्राप्त परिणाम आप द्वारा दी गई शर्तों के आधार पर प्राप्त होते हैं।

7) SELECT *sum(marks)*
FROM *marks*
GROUP BY *subject;*

1	98
2	90
3	194
4	170
5	98

Use the having clause to apply a search condition to groups

8) SELECT *sum(marks)*
FROM *marks*
GROUP BY *subject*
HAVING sum(*marks) >90*

1	98
2	194
3	170
4	98

Joins का प्रयोग तब किया जाता है जबकि हमें दो अथवा अधिक टेबल्स से डाटा की आवश्यकता होती है

join दो अथवा अधिक टेबल्स को एक टेबल में मिलाने के काम में आता है।

inner join की syntax :

9) SELECT student.rollno,student.studentname,marks.subject,marks.marks
FROM *student* INNER JOIN *marks*
ON *student.rollno = marks. Rollno;*

Rollno	Studentname	Subject	Marks
10001	VINOD	MATHS	90
10001	VINOD	HINDI	95
10001	VINOD	SCIENCE	98
10002	MANAN	HINDI	99
10002	MANAN	MATHS	80

महत्वपूर्ण बिन्दु

फंक्शन SQL/MySQL की अत्यंत सशक्त विशेषता हैं तथा निम्न प्रयोगों के लिए काम में आते है।

- ✓ दिये गये डेटा की गणना के लिए।
- ✓ डेटा अथवा डेटा समूह के अद्यतन (Modify) के लिए।
- ✓ पंक्ति अथवा पंक्ति समूह के output में बदलाव के लिए।
- ✓ दिनांक व संख्याओं को प्रदर्शित करने के लिए Format करने के लिए।
- ✓ Column के डेटा टाइप को बदलने के लिए।

फंक्शन तथा ऑपरेटर एक ही शैली में काम करते हैं। दोनों ही एक या एक से अधिक परिणाम प्रस्तुत करते हैं तथा शून्य व अन्य चरों को स्वीकृत करता है।

सब क्वैरी SQL/MySQL का विकसित रूप हैं। तथा MySQL द्वारा समर्थित है। सब क्वैरी में एक क्वैरी का परिणाम दूसरी क्वैरी का भाग होता हैं। यह वास्तव में एक select statement के अन्दर दूसरा select, insert, update या delete statement होता है।

Subquery की विशेषताएं महत्तवपूर्ण है क्योंकि -

- Subquery द्वारा यह आसान हो जाता है कि एक क्वेरी को टुकडों में करके उसे पुन उन टुकडों को दुबारा एक साथ रख दिया जाता है।
- कुछ क्वेरीज को बिना सबक्वेरी का उपयोग किये SQL में परिभाषित नहीं किया जा सकता है।

डेटाबेस ऑब्जेक्ट्स : डेटाबेस ऑब्जेक्ट्स RDBMS द्वारा संचालित होते है। सामान्यतया सभी डेटाबेस ऑब्जेक्ट्स मैमोरी में संग्रहित रहती हैं।

डेटाबेस ऑब्जेक्ट्स निम्न प्रकार से हैं -

- clusters
- columns
- constraints
- database
- database links
- indexes
- rollback segments
- savepoints
- tables
- tablespaces
- users
- views

संख्यात्मक फंक्शन (Numeric Functions) : यह केवल numeric input लेते हैं तथा numeric output देते हैं इन functions को select statement तथा अन्य statements के साथ प्रयुक्त कर सकते हैं।

स्ट्रिंग फंक्शन (String Functions) : यह केवल character input लेते है तथा numeric व character output देते हैं इन functions को select statement तथा अन्य statements के साथ प्रयुक्त कर सकते है। इन फंक्शन्स को नीचे वर्णित स्टूडेन्ट टेबल का उदाहरण लेकर समझाया गया है।

Date Functions : सभी date functions द्वारा date data type परिणाम दिया जाता है केवल month_between function को छोडकर जो कि अंकीय संख्या में परिणाम देता है।

Data type conversion functions: ये फंक्शन दिये गये मान को विभिन्न उपयोगों के लिए एक डेटा टाईप से दूसरे डेटा टाईप में बदलने के काम आते हैं। यह फंक्शन select तथा अन्य कथनों (statements) के साथ उपयोग में आते है।

इम्पलिसिट (Implicit) रूपान्तरण (Conversion): यह रूपांतरण वे है जो बिना CAST या CONVERT फंक्शन के काम में आते है।

एक्सप्लिसिट (Explicit) रूपान्तरण : ये रूपातंरण वे है जो CAST या CONVERT फंक्शन को काम में लेकर किये जाते है।

अभ्यासार्थ प्रश्न

बहुचयनात्मक प्रश्न :

1 CEIL(10.6) का परिणाम क्या होगा ?

(अं) 10 (ब) 11

(स) 9 (द) 0

2 mod (35,10) का परिणाम क्या होगा?

(अं) 3 (ब) 5

(स) 10 (द) 35

3. sqrt (81) का परिणाम क्या होगा ?

    (अं) 9   (ब) 6

    (स) 8   (द) 1

4. sign (-30) का परिणाम क्या होगा ?

    (अं) -1   (ब) 0

    (स) -30   (द) 30

अति लघुत्तरात्मक प्रश्न :

1. concat() function फंक्शन का उपयोग क्या है?
2. lpad() function फंक्शन का उपयोग क्या है?
3. substr() function फंक्शन का उपयोग क्या है?
4. rtrim() function फंक्शन का उपयोग क्या है?
5. now() function फंक्शन का उपयोग क्या है?
6. cast() function फंक्शन का उपयोग क्या है?
7. str_to_date() function फंक्शन का उपयोग क्या है?

लघुत्तरात्मक प्रश्न :

1. संख्यात्मक फंक्शन (numeric functions) कितने प्रकार के है?
2. स्ट्रिंग फंक्शन कितने प्रकार के है?

निबन्धात्मक प्रश्न :

1. ग्रुप फंक्शन को विस्तार से समझाइये ?
2. डेटा टाइप रूपांतरण फंक्शन को उदाहरण सहित समझाइये ?

उत्तर :

1. ब   2. ब   3. अ   4. अ