

रिलेशनल डाटाबेस की अवधारणायें

पाठ्यपुस्तक के प्रश्न

वस्तुनिष्ठ प्रश्न

प्रश्न 1. ये एक SQL खण्ड नहीं है

- (अ) Select
- (ब) From
- (स) Where
- (द) Condition

उत्तर: (द) Condition

प्रश्न 2. SQL का पूर्ण रूप है

- (अ) Structure Question Language
- (ब) Syntax Question Language
- (स) Structure Query Language
- (द) Structure Question Language

उत्तर: (स) Structure Query Language

प्रश्न 3. DDL केलिए है

- (अ) डेटा डेफिनेशन लैंग्वेज
- (स) डेटा डेटा लैंग्वेज
- (ब) डबल डेटा लैंग्वेज
- (द) इनमें से कोई नहीं

उत्तर: (अ) डेटा डेफिनेशन लैंग्वेज

प्रश्न 4. Count() है एक

- (अ) स्ट्रिंग फंक्शन
- (ब) सांख्यिक फंक्शन
- (स) दोनों
- (द) मौजूद नहीं

उत्तर: (ब) सांख्यिक फंक्शन

प्रश्न 5. Like ऑपरेटर के लिए उपयोग किया जाता है, जैसे

- (अ) Concatenating strings
- (ब) Count string character
- (स) String matching
- (द) All

उत्तर: (स) String matching

अतिलघु उत्तरीय प्रश्न

प्रश्न 1. SQL क्या है?

उत्तर- SQL आर डी बी एम एस (RDBMS) की एक स्टैंडर्ड लैंग्वेज है जो किसी रिलेशनल डेटाबेस को ऑर्गेनाइज करके उसमें उपस्थित डाटा को प्रबंधन करने एवम् रिलेशनल डेटाबेस में से डाटा रिट्राइवल (retrieval) के लिए उपयोग में लाई जाती है।

प्रश्न 2. आप SQL from clause द्वारा क्या समझते हो?

उत्तर- FROM clause में हम उन रिलेशन को लिखते हैं जिनको हमें query expression में उपयोग करना है। FROM clause में लिखे रिलेशनस का Cartesian product होता है।

प्रश्न 3. SQL select clause का महत्त्व क्या है?

उत्तर- SELECT clause में हम उन ऐट्रीब्यूट्स को लिखते हैं जो हमें हमारे आउटपुट रिलेशन में चाहिए।

प्रश्न 4. SQL के प्रकार का नाम दें।

उत्तर- SQL के निम्न प्रकार हैं

- डेटा डेफिनेशन लैंग्वेज (Data Definition Language)
- डेटा मनीपुलेशन लैंग्वेज (Data Manipulation Language)
- डेटा कन्ट्रोल लैंग्वेज (Data Control Language)

प्रश्न 5. अद्वितीय और प्राथमिक बाधाओं के बीच क्या अन्तर है?

उत्तर- अद्वितीय बाधा (Unique Constraint) व प्राथमिक बाधा (Primary constraint) में अन्तर Unique Constraint यह सुनिश्चित करता है कि कॉलम के सभी मान अलग-अलग हैं। दोनों अद्वितीय और

प्राथमिक बाधाएँ कॉलम के सेट के लिए विशिष्टता की गारंटी प्रदान करती हैं। एक प्राथमिक बाधा स्वचालित रूप से एक अद्वितीय बाधा है।

एक टेबल पर कई अद्वितीय बाधाएँ लगाई जा सकती हैं परन्तु प्राथमिक बाधा सिर्फ एक table पर एक ही लगाई जाती है।

प्रश्न 6. डेटाबेस instances को परिभाषित कीजिए।

उत्तर- किसी डाटाबेस में समय के किसी भी क्षण डाटा के समूह को डाटाबेस इन्स्टेन्स कहते हैं। उदाहरणार्थ एक student टेबल student डाटाबेस का एक इन्स्टेन्स है।

प्रश्न 7. हम SQL में order by clause का उपयोग कैसे करते हैं?

उत्तर- SQL ORDER BY क्लॉज-SQL के SELECT statement के द्वारा चुनी गई पंक्तियों का क्रम कुछ भी हो सकता है। अगर इन पंक्तियों को हमें किसी क्रम में देखना है तो हमें SQL के ORDER BY क्लॉज का उपयोग उस स्तंभ या स्तंभों के साथ करते हैं। जिनको हम क्रम में देखना चाहते हैं। अर्थात ORDER BY के द्वारा हम किसी स्तंभ की वेल्यूस को बढ़ते (ascending) या घटते (descending) क्रम में देख सकते हैं।

Syntax :

SELECT field1, field2, ... fieldn

FROM T1, T2, .. Tn

ORDER BY field1, field2... fieldn [Asc[Desc]];

प्रश्न 8. SQL में NULL क्या है?

उत्तर- एक फील्ड NULL वेल्यू के साथ का मतलब है कि फील्ड में कोई मान नहीं है। अगर एक field किसी टेबल में वैकल्पिक (optional) है, तो इस Field में एक नया रिकॉर्ड सम्मिलित करना या अपडेट करना संभव है। फिर field को NULL वेल्यू के साथ सहेजा (Save) जाएगा।

प्रश्न 9. आप Aggregate फंक्शन द्वारा क्या समझते हो?

उत्तर- Aggregate फंक्शन-My SQL में Aggregate फंक्शन इनपुट के तौर पर वेल्यूस का संग्रह लेते हैं और आउटपुट में एक वेल्यू देते हैं। My SQL में निम्न 5 प्रकार के Built in Aggregate फंक्शन होते हैं

AVERAGE : Avg ()

MAXIMUM : Max()

MINIMUM . : Min ()

TOTAL : Sumo COUNT_Count ()

प्रश्न 10. हम SQL grant कमाण्ड का उपयोग क्यों करते हैं?

उत्तर- रिलेशनल डाटाबेस में एक यूजर के द्वारा बनाये (create) गये ऑब्जेक्ट्स को दूसरे यूजर्स के पास तब तक नहीं पहुँचा (access) सकते जब तक उनको बनाने वाला यूजर दूसरे यूजर्स को इसकी सहमति नहीं देता। यह सहमति (permission) GRANT कमाण्ड के उपयोग द्वारा दी जा सकती है।

लघु उत्तरीय प्रश्न

प्रश्न 1. SQL की बुनियादी संरचना क्या है?

उत्तर- स्ट्रक्चर क्यूरी लैंग्वेज SQL रिलेशनल ऐलजेबरा एवं रिलेशनल कैलकुलस के कोम्बीनेशन का उपयोग करती है। SQL आर.डी.बी.एम.एस. की एक स्टैन्डर्ड लैंग्वेज है जो किसी रिलेशनल डाटा बेस को ऑर्गेनाइज करने उसमें उपस्थित डाटा को प्रबंधन करने एवं रिलेशनल डाटा बेस में से डाटा रिट्रीवल के लिए उपयोग में लाई जाती है।

डी.बी.एम.एस. के विक्रेताओं जैसे कि ऑरेकल IBM, DB2, Sybase and Ingress अपने डाटाबेस के लिए SQL का एक प्रोग्रामिंग लैंग्वेज के तौर पर उपयोग करते हैं। इसका मूल संस्करण sequel कहलाता था जिसको IBM ने विकसित किया था।

इसके कई संस्करण होते हैं जैसे SQL-86, SQL-89 (extended standard), SQL-92 and SQL-1999 और वर्तमान संस्करण SQL-2003, SQL केवल मात्र एक डाटाबेस query लैंग्वेज नहीं है बल्कि अपने आप में एक स्टैन्डर्ड है। जिसके निम्नलिखित भाग हैं।

1. डेटा डेफिनेशन लैंग्वेज (Data Definition Language)
2. डेटा मनीपुलेशन लैंग्वेज (Data Manipulation Language)
3. डेटा कन्ट्रोल लैंग्वेज (Data Control Language)

ज्यादातर कामर्शियल रिलेशनल डाटा बेस जैसे IBM, Oracle, Microsoft, Sybase आदि SQL का उपयोग करते हैं।

प्रश्न 2. विभिन्न DML कमाण्ड क्या हैं? उनके लिए Syntax's दें।

उत्तर- SQL DML कमाण्ड्स निम्नलिखित हैं।

- (i) INSERT
- (ii) DELETE
- (iii) UPDATE
- (iv) SELECT

(i) INSERT कमाण्ड – CREATE TABLE कमाण्ड के उपयोग द्वारा किसी टेबल को बनाने पर एक खाली टेबल बनती है अर्थात उस टेबल में किसी प्रकार की कोई वेल्यू या रिकार्ड या टप्लस नहीं होते हैं। किसी टेबल में रिकार्ड या डाटा डालने के लिए हम SQL.INSERT INTO कमाण्ड्स का उपयोग करते हैं।

My SQL Syntax

INSERT INTO

Table_Name (Column_Name 1, Column_Name 2, ...,Column_Name n) VALUES (Value 1, value 2,...,value n);

स्ट्रिंग टाइप्स के डाटा के लिए सभी वेल्यूज को सिंगल या डबल quotes (" ") में लेंगे।

(ii) SQL DELETE कमाण्डस – DELETE कमाण्डस के द्वारा हम एक पूरा टप्लस DELETE करते हैं। इसके द्वारा हम किसी ऐट्रीब्यूट के वेल्यू को DELETE नहीं कर सकते हैं।

Syntax: DELETE FROMT, WHEREP;

यहाँ T एक रिलेशन है जिसमें से टपल्स DELETE करना है। तथा वह predicate (condition) है। जिसके अनुसार टपल्स DELETE होंगे।

(iii) SQL UPDATE कमाण्डस – UPDATE कमाण्ड का उपयोग हम किसी टपल्स के विशिष्ट ऐट्रीब्यूट के वेल्यू को बदलने के लिए करते हैं। अर्थात अगर हम पूरे टपल्स को नहीं बदलना चाहते हैं, केवल इसमें किसी ऐट्रीब्यूट के वेल्यू को ही बदलना चाहते हैं तो हम UPDATE कमाण्ड का उपयोग करते हैं। जिसका Syntax निम्न है।

Syntax:

UPDATE table_name SET first_field = value1, second_field = value2

(iv) SQL SELECT statement – किसी SQL query expression में मुख्य रूप से निम्नलिखित तीन क्लॉजेस (clauses) होते हैं। अर्थात कोई भी SQL query जो हम रिलेशन डाटा बेस के लिए लिखते हैं, उसकी बुनियादी संरचना (basic structure) में उक्त 3 क्लॉज होंगे।

- SELECT क्लॉज में हम उन ऐट्रीब्यूट्स को लिखते जो हमें हमारे आउटपुट रिलेशन में चाहिए।
- FROM क्लॉज में हम उन रिलेशन को लिखते हैं। जिनको हमें query expression में उपयोग करना है। FROM क्लॉज में लिखे रिलेशनस का Cartesian product होता है।
- WHERE क्लॉज में हम predicate लिखते हैं। जो FROM clause के रिलेशनस के ऐट्रीब्यूट्स को लिप्त रखता है। अर्थात जिसकी Boolean वेल्यू (true or false) होती है।

SQL query का निम्न फोर्म (Form) होता है।

SELECT AT1, AT2, AT3, ... ,ATn,

FROM r1, r2, r3,... rn.

WHERE P;

यहाँ AT_i एक ऐट्रीब्यूट को प्रदर्शित करता है। और r_i एक रिलेशन को एक predicate है।

प्रश्न 3. फोरेन key क्या है? हम किसी टेबल में एक फोरेन key कैसे बनाते हैं?

उत्तर- फोरेन की (Foreign key) इन्टीग्रीटी कन्स्ट्रेंट्स-इस कन्स्ट्रेंट्स को समझने के लिए निम्न प्रदर्शित टेबल को उदाहरण के लिए हम लेते हैं। यहाँ दो टेबल student एवं Class के नाम से है एवं किसी क्षण उनमें एन्टर वेल्यूस भी प्रदर्शित हैं।

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

Classes		
Class_name	Class_room	Strength
12 th	F-1	95
10 th	F-2	80
9 th	F-3	70
4 th	F4	110

चित्र 1 : फोरेन की स्टूडेंट रिलेशन टेबल

उक्त टेबल Student में टेबल की प्राइमरी की Roll_no फील्ड है। जबकि Classes टेबल की प्राइमरी की Class_name फील्ड है। यहाँ पर हमने यह माना है कि सभी स्टूडेंट्स की एक ही class विद्यमान है, जैसे कि 12th की एक class, 10th की एक class, 9th की एक class इसी प्रकार अन्य, अर्थात एक ही class के section अलग-अलग नहीं है। इसलिए classes टेबल की प्राइमरी की Class_name है।

Student टेबल अपने ऐट्रीब्यूट (Roll_no, address, age, name, class) के बीच में एक ऐसा ऐट्रीब्यूट भी रखता है जो किसी अन्य टेबल की प्राइमरी key है। उदाहरणस्वरूप Student टेबल में Class ऐट्रीब्यूट टेबल की प्राइमरी Key है। अतः Student टेबल में इस ऐट्रीब्यूट class को हम इस टेबल की primary key कहेंगे। जो कि टेबल classes को रेफर करेगी।

रिलेशन Student टेबल को हम रेफरेन्सिंग कहेंगे जबकि Classes को रेफरेन्सड रिलेशन ऑफ फोरेन key कहेंगे। किसी ऐट्रीब्यूट को primary key होने के लिए उसका टाइप domain वही होना चाहिए जो दूसरे रिलेशन के ऐट्रीब्यूट को है तथा foreign key में ऐट्रीब्यूट्स की संख्या भी दूसरे रिलेशन के ऐट्रीब्यूट के बराबर होनी चाहिए अर्थात वे काम्पैटेबल होने चाहिए।

प्रश्न 4. SQL के group by clause का उपयोग उदाहरण के द्वारा समझाओ।

उत्तर- SQL GROUP BY क्लॉज-MySQL का यह क्लॉज अत्यन्त उपयोगी है। इस क्लॉज के उपयोग द्वारा कई महत्वपूर्ण क्वेरी लिखी जा सकती हैं। GROUP BY क्लास के द्वारा किसी स्तंभ (column) या स्तंभों या ऐट्रीब्यूट्स की वैल्यूज का समूह बनाया जा सकता है। अर्थात क्लाज में दिये गये ऐट्रीब्यूट का उपयोग समूह (GROUP) बनाने के लिए करते हैं। GROUP BY क्लॉज में दिये गये ऐट्रीब्यूट या ऐट्रीब्यूट्स के वैल्यूज जिन टप्लस या पंक्तियाँ के लिए एक समान हैं। वे सभी टप्लस या पंक्तियाँ एक समूह में आयेंगी।

GROUP BY क्लॉज को हम निम्न उदाहरण द्वारा समझ सकते हैं। इसके लिए Student टेबल को लेते हैं। जिसमें किसी क्षण निम्न रिकॉर्ड्स हैं।

Roll_No	Name	Age	Address	Class
1	Ajay	9	Jaipur	4th
2	Vijay	17	Kota	12th
10	Hari	11	Udaipur	7th
17	Shanker	13	Jaipur	8th
21	Om	21	Ajmer	12th
51	Mayank	15	Ajmer	9th
90	Anju	18	Ajmer	11th
53	Suman	12	Ajmer	10th
64	Kamal	10	Kota	4th
500	Komal	16	Udaipur	9th
700	Aryabhata	11	Jaipur	7th
900	Bodhayan	13	Jodhpur	8th

चित्र : Student table

उदाहरण

क्योरी – प्रत्येक Class में पढ़ने वाले छात्रों की संख्या बताइयें।

अगर हम यह क्वेरी निम्न प्रकार से लिखें तो परिणाम गलत प्राप्त होगा।

MySQL>SELECT Count(*) FROM Student

Output->

Count(*)
12

क्योंकि इस Syntax के द्वारा कुल पढ़ने वालों छात्रों की संख्या प्राप्त होगी अर्थात Student टेबल में जितने छात्रों को रिकार्ड उपलब्ध हैं। वही उस स्कूल में पढ़ने वाले छात्र हैं। अतः यह Syntax, Student टेबल में कुल कितने टप्लस का हैं, उनकी संख्या देगा। सही परिणाम के लिए count aggregate फंक्शन के GROUP BY क्लॉज का उपयोग करना पड़ता है। जिसका Syntax निम्नानुसार है।

MySQL>SELECT Class, Count (Roll_No) FROM Student GROUP BY Class;

Student टेबल में समूह बनने के बाद टेबल के बाद Student टेबल कुछ इस प्रकार दिखेगी क्योंकि GROUP BY क्लॉज में Class ऐट्रीब्यूट के द्वारा समूह बनाया जा रहा है। अतः एक समान Class वाली पंक्तियों एक समूह में दिखेगी।

Output->

Class	Roll_No	Age	Name	Address
11th	90	18	Anju	Ajmer
12th	2	17	Vijay	Kota
12th	21	21	Om	Ajmer
10th	53	12	Suman	Ajmer
9th	51	15	Mayank	Ajmer
9th	500	16	Komal	Ajmer
8th	17	13	Shanker	Udaipur
8th	900	13	Bhodhayan	Jaipur
7th	10	11	Hari	Udaipur
7th	700	11	AryaBhatta	Jaipur
4th	1	9	Ajay	Jaipur
4th	64	10	Komal	Kota

Class	Count (Roll_No)
10th	1
11th	1
12th	2
9th	2
8th	2
7th	2
4th	2

जिस column के द्वारा GROUP बनाया जाता है। उस column पर कोई calculation Count, Avg, Max. Min आदि aggregate function के द्वारा की जा सकती है। अतः इस क्यूरी में Count, aggregate फंक्शन को हर समूह के टपल्स जिनकी Class एक समान है उसके लिए लगाया गया है। क्योंकि SELECT क्लॉज में दो ही फील्ड हैं अतः परिणाम निम्नानुसार प्राप्त होगा।

Class	Count (Roll_No)
10th	1
11th	1
12th	2
9th	2
8th	2
7th	2
4th	2

प्रश्न 5. Cartesian join और Natural join में क्या अन्तर है?

उत्तर- Cartesian join-SQL CROSS JOIN एक result set का उत्पादन (produce) करता है जो पहली table में पंक्तियों (rows) की संख्या से, दूसरी table में पंक्तियों (rows) की संख्या से गुणा होती है। अगर CROSS JOIN के साथ WHERE clause का उपयोग नहीं किया जाता तो इस तरह के परिणाम (result) को Cartesian product कहा जाता है यदि WHERE clause को CROSS JOIN के साथ उपयोग किया जाए तो यह INNER JOIN की तरह फंक्शन करता है।

Natural Join-Natural Join वह join ऑपरेशन है जो कि एक implicit join clause उन कॉमन columns पर बनाता है जो उन दो tables में हों, जो आप जोड़ना चाहते हैं। कॉमन columns वे columns होती हैं जिनका नाम दोनों tables में एक (same) ही होता है।

एक Natural join Inner join, Left Outer join या Right outer join हो सकता है। डिफॉल्ट रूप से Inner join होता है।

निबंधात्मक प्रश्न

प्रश्न 1. SQL Joins, टेबिल्स के उपयुक्त उदाहरण लेने के साथ समझाओ।

उत्तर- SQL JOIN-SQL के JOIN key word का उपयोग दो या दो से अधिक टेबिल्स से डाटा की क्यूरी करने के काम के लिए होता है। ऑपरेशन दो रिलेशन को इनपुट के तौर पर लेते हैं और एक रिलेशन आउटपुट के तौर पर देते हैं। SQL में दो रिलेशन को JOIN करने के कई तंत्र (mechanisms) हैं।

जैसे कि-

1. Cartesian product mechanism
2. Inner join
3. Outer join (left, right, full)

ऊपर दिये प्रत्येक join type के लिए एक Join condition भी जुड़ी हुई होती है। अतः एक Join expression इन दोनों (join type और join expression) से मिलकर बनती है जिसे हम FROM क्लॉज में उपयोग करते हैं। JOIN को समझने के लिए हम Student टेबल एवं Class टेबल जिसमें किसी क्षण निम्न रिकॉर्ड को लेते हैं।

Class-Home	Room No	CStrength
12th	F-1	90
11th	F-2	75
10th	F-5	99
9th	S-21	110
8th	S-10	70
7th	F-10	85
6th	F-17	65
5th	F-7	60
4th	F-9	55
3rd	F-8	50
2nd	S-15	35
1st	S-9	60

Class table

Roll_No	Name	Age	Address	Class
1	Ajay	9	Jaipur	4th
2	Vijay	17	Kota	12th
10	Hari	11	Udaipur	7th
17	Shanker	13	Jaipur	8th
21	Om	21	Ajmer	12th
51	Mayank	15	Ajmer	9th
90	Anju	18	Ajmer	11th
53	Suman	12	Ajmer	10th
64	Kamal	10	Kota	4th
500	Komal	16	Udaipur	9th
700	Aryabhatt	11	Jaipur	7th
900	Bodhayan	13	Jodhpur	8th

Student table Student एवं Class टेबल को JOIN करने के लिए एक क्वेरी लिखते हैं।

MySQL>SELECT Roll_No, Class, CStrength FROM Student AS St, Classes AS S, WHERE St.
Class = S. Class_name;

इस क्वेरी में FROM क्लॉज में Student टेबल को RENAME करके St एवं Class को S किया गया है। इन दोनों रिलेशन का Cartesian product होगा जिसमें St टेबल के हर टपल्स का S टेबल के हर टपल्स से JOIN होगा। अतः प्राप्त रिलेशन में कुल टपल्स होंगे।

$$N1 * N2 = 12 * 12 = 144$$

यहाँ N1 St टेबल में टपल्स की संख्या व N2 S टेबल में टपल्स की संख्या है। यहाँ पर रिजल्ट टेबल में टपल्स प्राप्त होंगे वो WHERE क्लॉज की शर्तों को पूर्ण करने वाली ही होंगे।

OUTER JOIN आपरेशन – ये निम्न प्रकार के होते हैं।

1. LEFT OUTER JOIN
2. RIGHT OUTER JOIN
3. FULL OUTER JOIN

Outer Joins के साथ निम्न Join Condition का उपयोग करते हैं।

1. Natural
2. ON (Predicate)
3. Using (A1, A2, An)

Left outer join और ON Join condition – Left Join को समझने के लिए निम्न दो टेबल लेते हैं।

Classes		
Class_name	C Room No	C Strength
12th	F-11	90
9th	S-21	110
7th	F-10	85
4th	F-9	55

Admission		
Class_name	Roll_No	admission_date
12th	79	2000/7/15
9th	89	2010/8/13
6th	69	2012/7/21
5th	49	2013/7/03

Syntax:

Select Classes, Class_Name, Roll_No, CStrength From Classes Left Outer Join Admission on Classes. class_Name = Admission. Class_Name

यहाँ रिलेशन का नाम उसके ऐट्रीब्यूट के साथ लिखते हैं क्योंकि एक समान नाम का ऐट्रीब्यूट एक से ज्यादा रिलेशन में हैं। अतः अस्पष्टता (ambiguity) को दूर करने के लिए ऐसा किया है।

Output		
Class_name	Room No	C Strength
12th	79	90
9th	89	110
7th	Null	85
4th	Null	55

Left Outer Join के रिजल्ट में दोनों रिलेशन के Matching tuples तथा Left वाले रिलेशन (classes) के unmatched tuples उपस्थित होते हैं।

Right Outer Join और on Join Condition

Syntax:

Classes Right Outer Join admission on classes. Class Name=Admission. Class_Name

Output

Class_Name	CRoomNo	CStrength	Class_Name	Roll_No	Admission_date
12th	F-11	90	12th	79	2000/7/15
9th	S-12	110	9th	89	2010/8/13
Null	Null	Null	6th	69	2012/7/21
Null	Null	Null	5th	49	2013/7/03

Right Outer Join आपरेशन Left Outer Join के समान ही है। किन्तु इसमें Join operation के दाँयी (right) ओर वाले रिलेशन के unmatched tuples भी आते हैं। left रिलेशन के एट्रीप्यूटस के लिए वेल्यूम Null रखेंगे।

Full outer join और On condition

Syntax:

Classes full outer join admission on classes. Class_Name = Admission. Class_Name

Class_Name	CRoomNo	CStrength	Class_Name	Roll_No	Admission_date
12th	F-11	90	12th	79	2000/7/15
9th	S-12	110	9th	89	2010/8/13
7th	F-10	85	Null	Null	Null
4th	F-9	55	Null	Null	Null
Null	Null	Null	6th	69	2012/7/21
Null	Null	Null	5th	49	2013/7/03

यहाँ दोनों रिलेशन के unmatched टपल्स भी आयेगें। साथ में दूसरे रिलेशन के unmatched टपल्स के लिए Null आयेगा।

Outer Join और Natural Condition

दो रिलेशन का Natural Join करने पर उन टपल्स की संख्या उन रिलेशन में उपस्थित एक समान (common) एट्रीप्यूटस के द्वारा प्राप्त होते हैं और common एट्रीप्यूटस result रिलेशन में एक बार आते हैं। वह क्रम में सबसे पहले आते हैं।

उदाहरण-Classes Natural right outer join admission

Class_Name	CRoomNo	CStrength	Roll_No	Admission_date
12th	F-11	90	79	2000/7/15
9th	S-12	110	89	2010/8/13
6th	Null	Null	69	2012/7/21
5th	Null	Null	49	2013/7/03

अन्य outer join भी Natural condition शर्त के लिए ऊपर दिये अनुसार हम प्राप्त कर सकते हैं।

Inner join-

उदाहरण

Classes inner Join Admission on Classes. Class_Name=Admission. Class_Name;

Class_Name	CRoomNo	CStrength	Class_Name	Roll_No	Admission_date
12th	F-11	90	12th	79	2000/7/15
9th	S-12	110	9th	89	2010/8/13

Inner Join और Natural Condition

उदाहरण

Classes Natural Inner-Join Admission

Output->

Class_Name	CRoomNo	CStrength	Roll_No	Admission_date
12th	F-11	90	79	2000/7/15
9th	S-12	110	89	2010/8/13

यहाँ केवल एक ही ऐट्रीब्यूट दोनों रिलेशन में समान है। अतः Join केवल उस ही ऐट्रीब्यूट के द्वारा होगा।

प्रश्न 2. Aggregate फंक्शन क्या है? हम aggregate फंक्शन का उपयोग कैसे करें? प्रत्येक का एक उदाहरण दें।

उत्तर- Aggregate फंक्शन – MySQL में Aggregate फंक्शन इनपुट के तौर पर वेल्यूस का संग्रह लेते हैं और आउटपुट में एक वेल्यू देते हैं। MySQL में निम्न 5 प्रकार के built in Aggregate फंक्शन होते हैं।

AVERAGE: Avg ()

MAXIMUM: Max ()

MINIMUM: Min()

TOTAL: Sum ()

COUNT: Count ()

यहाँ Sum एवं Average फंक्शन की इनपुट वेल्यूस आवश्यक रूप से नम्बर होने चाहिए। जबकि दूसरे ऑपरेटर्स, स्ट्रिंग के ऊपर भी काम कर सकते हैं।

Avg () function फंक्शन – यह फंक्शन टेबल के किसी फील्ड की वेल्यूस का औसत निकालने के उपयोग में आता है।

उदाहरण-

MySQL>SELECT Avg (Salary) FROM Teacher;

Output->

Avg (Salary)
14250.0000

ऊपर दिये उदाहरण में Average फंक्शन Salary फील्ड में वेल्यूस का औसत रिटर्न करता है।

Sum () Function फंक्शन – यह फंक्शन किसी फील्ड की सभी वेल्यूस का Sum देता है।

उदाहरण-

MySQL>SELECT Sum (Salary) FROM Teacher;

OUTPUT->

Sum (Salary)
57000

Max () function फंक्शन – यह फंक्शन वह रिकॉर्ड जो किसी रिकॉर्ड सेट में अधिकतम है को देता है।

उदाहरण-

MySQL>SELECT Max (Salary) FROM Teacher

Output->

Max (Salary)
40000

Min () function फंक्शन – यह फंक्शन निम्नतम वेल्यू वाला रिकार्ड देता है।

उदाहरण-

MySQL>SELECT Min (Salary) FROM Teacher

Output->

Min (Salary)
3000

Count () function फंक्शन – यह फंक्शन टेबल में रिकार्ड की संख्या गणना के लिए काम में आता है।
अर्थात रिकार्ड्स की कुल संख्या पता करने के काम आता है। (counting the number of records)

उदाहरण-

SELECT Count(*) FROM Student

Output->

Count (*)
4

उदाहरण-

SELECT Count(*) FROM Student WHERE Class="9th"

Output->

Count (*)
2

प्रश्न 3. किसी एकल SQL क्वेरी में Where, group by और having clause के उपयोग की व्याख्या करें। एक उपयुक्त उदाहरण दें।

उत्तर:

Class	Roll No	Age	Name	Address
11th	90	18	Anju	Ajmer
12th	2	17	Vijay	Kota
12th	21	21	Om	Ajmer
10th	53	12	Suman	Ajmer
9th	51	15	Mayank	Ajmer
9th	500	16	Komal	Udaipur
8th	17	13	Shanker	Jaipur
8th	900	13	Bhodhayan	Jodhpur
7th	10	11	Hari	Udaipur
7th	700	11	Arya Bhatta	Jaipur
4th	1	9	Ajay	Jaipur
4th	64	10	Komal	Kota

Class	Count (Roll_No)
10th	1
11th	1
12th	2
9th	2
8th	2
7th	2
4th	2

Student table

जिस column के द्वारा GROUP बनाया जाता है। उस column पर कोई calculation Count, Avg, Max, Min आदि aggregate function के द्वारा किया जा सकता है। अतः इस क्वेरी में Count, aggregate फंक्शन को हर समूह के टपल्स जिनकी Class एक समान है के लिए लगाया गया है। क्योंकि SELECT क्लॉज में दो ही फील्ड है। अतः परिणाम निम्नानुसार प्राप्त होगा।

Class	Count (Roll_No)
10th	1
11th	1
12th	2
9th	2
8th	2
7th	2
4th	2

उदाहरण-

क्वेरी – उन Class के नाम बताइये (प्रत्येक Class) जिनमें पढ़ने वाले छात्रों की संख्या 1 से अधिक है।

Syntax :

MySQL> SELECT Class, Count (Roll_No) FROM Student GROUP BY Class HAVING Count (Roll_No)>1;

Output->

Class	Count (Roll_No)
12th	2
9th	2
8th	2
7th	2
4th	2

यहाँ रिजल्ट में प्रत्येक GROUP से वही टपल्स SELECT हुए हैं जिनका Count 1 से अधिक है। SQL में एक किसी टपल्स के लिए कोई शर्त पूर्ण होती है या नहीं यह देखने के लिए WHERE क्लॉज का उपयोग करते हैं। जबकि GROUP BY क्लॉज के द्वारा बनाये गये समूहों में उपस्थित टपल्स के लिए शर्त को देखने (test) के लिए HAVING क्लॉज का उपयोग करते हैं। WHERE क्लॉज एवम् HAVING क्लॉज में यह मुख्य अन्तर है।

SQL में HAVING क्लॉज में दर्शाये गये predicate को लागू GROUP BY क्लॉज के द्वारा GROUP बनाने के बाद करते हैं। इसलिए इसके साथ aggregate फंक्शन भी उपयोग कर सकते हैं।

नोट – अगर किसी क्वेरी में WHERE, HAVING GROUP BY आते हैं तो सबसे पहले WHERE क्लॉज में लागू होगा उसके बाद जिन टपल्स के लिए शर्त पूर्ण होगी वे GROUP BY क्लॉज के द्वारा समूहों में रखेंगे और अन्त में हर समूह के लिए HAVING क्लॉज को लागू करेंगे जिन समूहों के लिए HAVING क्लॉज संतोषजनक नहीं रहता वे समूह परिणाम में से हट जाते हैं।

प्रश्न 4. दिए गए स्कीमा पर विचार करें।

students (Roll_no, Sname, age, phone, address, class)

Classes (Class_name, CRoom No, CStrength)

और निम्न के लिए कोई SQL सिंटैक्स लिखें।

1. कमरा नंबर F-12 में बैठे 5 वीं class के छात्रों का नाम का पता लगाएँ।
2. अजमेर में रहने वाले 10 वीं कक्षा के छात्रों की संख्या का पता लगाएँ।

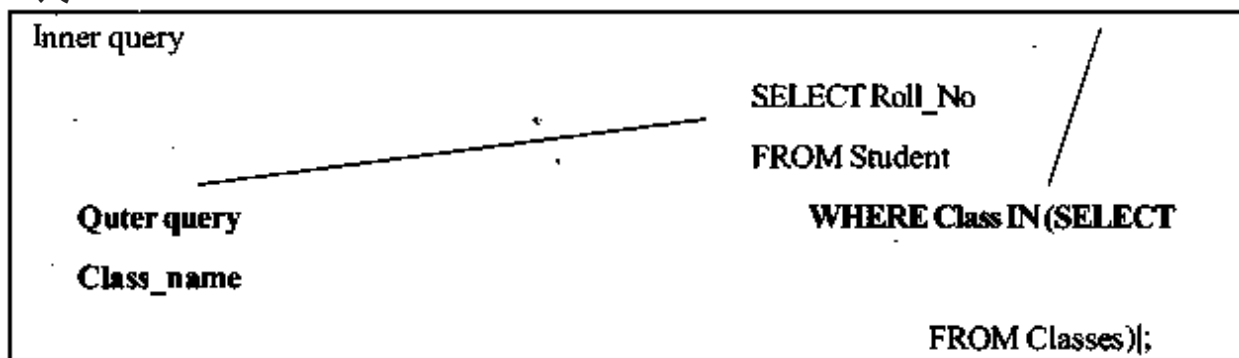
उत्तर: (1) Select Sname, address
From Students, Classes
Where classes. C Room No=F-12 and
Students. class = 5

(2) Select count (S Name)
From Students
Where class = 10 and address = "Ajmer"

प्रश्न 5. उप क्वेरीज (Sub queries) द्वारा आपका क्या मतलब है? उप क्वेरीज क्यों उपयोगी होती है? सेट comparison में उप क्वेरी का उपयोग समझाओ।

उत्तर- SQL Sub queries-एक Sub queries इस प्रकार की SQL क्वेरी होती है जो किसी अन्य क्वेरी के भीतर नेस्टेड (nested) होती है। इसके अलावा Sub queries खुद भी अन्य Sub queries के भीतर नेस्टेड (nested) हो सकती है। Sub queries को inner query क्वेरी भी कहते हैं तथा क्वेरी के भीतर Sub queries होती है उसे Outer क्वेरी (बाहरी क्वेरी) कहते हैं।

उदाहरण:



Sub queries के द्वारा दी गई एक वैल्यू को कम्पेयर करने के लिए कम्पेरिजन ऑपरेटर (=, >=, <= etc) आदि का उपयोग कर सकते हैं। उदाहरण के लिए Teacher टेबल का उपयोग करते हैं। SELECT Tname, Salary FROM Teacher WHERE Salary=(SELECT Max (Salary) FROM Teacher);

Output->

Tname	Salary
Hariom	40000

उदाहरण:

उन Teacher के नाम बताइये जिनकी Salary सभी Teacher की औसत Salary से कम है।

SELECT Tname, Salary FROM Teacher WHERE Salary < (SELECT Avg (Salary) FROM Teacher);

Output->

Tname	Salary
Radha Krishnan	3000
Rajesh	5000
Lalaji	9000

अन्य महत्वपूर्ण प्रश्न

अतिलघु उत्तरीय प्रश्न

प्रश्न 1. टेबल से आप क्या समझते हैं?

उत्तर- RDBMS में डाटा एक प्रकार के डाटाबेस ऑब्जेक्ट में स्टोर होता है जिसे हम टेबल कहते हैं। दूसरे शब्दों में टेबल सम्बन्धित डाटा एन्ट्रीज का संग्रहण होता है जिसमें पंक्तियाँ एवं स्तम्भ होते हैं।

प्रश्न 2. फिल्ड (Field) क्या होता है?

उत्तर- किसी टेबल का फिल्ड (Field) उसका एक स्तम्भ होता है जो कि उस टेबल में किसी रिकॉर्ड की specific इंकोमेशन को रखता है।

प्रश्न 3. रिकॉर्ड (Record) क्या होता है?

उत्तर- रिकॉर्ड को हम टेबल की पंक्ति भी कहते हैं तथा यह एक टेबल की वह individual entry है जो उस टेबल में होती है।

प्रश्न 4. स्तम्भ से आप क्या समझते हैं?

उत्तर- स्तम्भ किसी एक टेबल की वह वर्टिकल एन्ट्री है जो किसी विशिष्ट फिल्ड से सम्बन्धित सभी सूचनाएँ (information) रखता है।

प्रश्न 5. डोमेन की परिभाषा बताइए।

उत्तर- किसी फ़िल्ड की परमिटेड वैल्यू सेट को उसका डोमेन (Domain) कहते हैं। उदाहरणस्वरूप field name के लिए डोमेन सभी नामों का सेट है।

प्रश्न 6. डाटाबेस स्कीमा से आप क्या समझते हैं?

उत्तर- डाटाबेस स्कीमा (Database Schema)-डाटाबेस स्कीमा किसी डाटाबेस की लॉजिकल डिजाइन है जो कि, शायद ही बदलती है।

प्रश्न 7. डेटा डेफिनेशन लैंग्वेज के विषय में बताइए।

उत्तर- डेटा डेफिनेशन लैंग्वेज (DDL), SQL का एक भाग है जिसकी सहायता से हम डाटाबेस स्कीमा को स्पेसीफाई (specify) कर सकते हैं।

प्रश्न 8. डेटा मैनीपुलेशन लैंग्वेज के विषय में बताइए।

उत्तर- डेटा मैनीपुलेशन लैंग्वेज (DML), SQL का वह भाग है, जिसे हम क्वेरी लैंग्वेज भी कहते हैं। अतः DML एवं लैंग्वेज समानार्थक शब्द हैं। DML का उपयोग डाटा जो कि रिलेशन में स्टोर है को मैनीपुलेशन (इन्सर्ट, डिलीट, अपडेट और रिट्राइवल) करने में करते हैं।

प्रश्न 9. SQL का मूल संस्करण क्या था और इसको किसने विकसित किया था?

उत्तर- SQL का मूल संस्करण Sequel कहलाता था जिसको IBM ने विकसित किया था।

प्रश्न 10. डेटा कंट्रोल लैंग्वेज के विषय में बताइए।

उत्तर- डेटा कंट्रोल लैंग्वेज (DCL), SQL की उपकमाण्ड्स का संग्रह है जो डाटाबेस में डाटा की सुरक्षा तथा डेटा को मैनीपुलेशन के अधिकारों से सम्बन्धित है।

प्रश्न 11. Table create करने का Syntax लिखिए।

उत्तर- Table create करने का Syntax-MySQL में एक टेबल बनाने का generic syntax है CREATE TABLE table-Name (F1 D1, F2 D2,,Fn Dn < Integrity constraints 1, < ICK >); इस Syntax में प्रत्येक टेबल के फ़िल्ड या ऐट्रीब्यूट का नाम है तथा Di प्रत्येक Fi के अन्तर्गत आने वाली वैल्यू का डोमेन टाइप है।

प्रश्न 12. Create Index कमाण्ड क्या है? इसका syntax भी लिखिए।

उत्तर- Create Index कमाण्ड-यह कमाण्ड किसी टेबल पर इन्डेक्स क्रिएट करने के काम आती है। हम इन्डेक्स को नहीं देख सकते पर यह टेबल में डाटा तेजी से सर्च करने में मदद करता है।
इसका syntax निम्न है CREATE INDEX Index_name ON table_name (column_name)

प्रश्न 13. INSERT कमाण्ड के विषय में Syntax सहित बताइए।

उत्तर- किसी टेबल में रिकार्ड या डाटा डालने के लिए हम SQL INSERT INTO कमाण्ड्स का उपयोग करते हैं। इसका Syntax निम्न है INSERT INTO
Table_Name (Column_Name1, Column_Name2,,Column_Namen)
VALUES (Value1, Value2, valuen);

प्रश्न 14. SQL DELETE कमाण्ड क्या है?

उत्तर- DELETE कमाण्ड के द्वारा हम एक पूरा टपल्स DELETE करते हैं। इसके द्वारा हम किसी ऐट्रीब्यूट की वैल्यू को DELETE नहीं कर सकते हैं।

Syntax: DELETE FROM T, WHERE P;

प्रश्न 15. SQL UPDATE कमाण्ड क्या है?

उत्तर- UPDATE कमाण्ड का उपयोग हम किसी टपल्स के विशिष्ट ऐट्रीब्यूट की वैल्यू को बदलने के लिए करते हैं।

Syntax: UPDATE table_name SET first_field = value1,
Second_field = value2, [WHERE clause];

प्रश्न 16. SELECT कमाण्ड के विषय में Syntax सहित बताइए।

उत्तर- SELECT कमाण्ड उन ऐट्रीब्यूट की लिस्ट रखता है जो उस रिलेशन से retrieve करने हैं।

Syntax: SELECT field_names, FROM relation_names;

प्रश्न 17. WHERE clause किन लॉजिकल (Logical) connections का उपयोग करते हैं?

उत्तर- WHERE clause निम्नलिखित लॉजिकल (Logical) connections का उपयोग करते हैं

- AND

- OR
- NOT

प्रश्न 18. SQL के SET ऑपरेटर कौन-कौन से हैं?

उत्तर- SQL के SET ऑपरेटर निम्नलिखित हैं जो कि रिलेशनस् के ऊपर ऑपरेट करते हैं

- UNION
- UNION ALL
- INTER SET
- EXCEPT

प्रश्न 19. UNION SET ऑपरेटर का उपयोग बताइए।

उत्तर- इस ऑपरेटर का उपयोग दो या दो से अधिक SELECT statement के रिजल्ट सेट को संयुक्त (combine) करने के काम आता है। यह ऑपरेटर duplicate पंक्तियों को हटा (remove) कर रिजल्ट देता है।

प्रश्न 20. INTER SET ऑपरेटर का उपयोग बताइए।

उत्तर- इस ऑपरेटर का उपयोग दो या दो से अधिक डाटा सेट के intersection के लिए काम आता है। अर्थात् यदि दोनों सेटों में कोई रिकॉर्ड विद्यमान है तो INTER SET ऑपरेटर के रिजल्ट में हमें वह रिकॉर्ड प्राप्त होगा अन्यथा अगर कोई रिकॉर्ड केवल एक ही डाटा सेट में है तो वह रिजल्ट में नहीं आयेगा।

प्रश्न 21. EXCEPT ऑपरेटर का उपयोग बताइए।

उत्तर- यह दो SELECT statements को संयुक्त करने के उपयोग में आता है और यह इनको संयुक्त इस प्रकार करता है कि पहले SELECT statement की वह पंक्तियों जो दूसरे SELECT statement में नहीं है, को परिणामस्वरूप देता है।

प्रश्न 22. RAND फंक्शन उदाहरण सहित बताइए।

उत्तर- RAND फंक्शन-MYSQL में 0 व 1 के बीच में कोई भी नम्बर randomly निकालने के लिए हम RAND फंक्शन का उपयोग करते हैं।

उदाहरण- MySQL> SELECT RAND(), RAND();

प्रश्न 23. SQRT फंक्शन उदाहरण सहित बताइए।

उत्तर- SQRT फंक्शन- यह फंक्शन किसी नम्बर का square root निकालने के काम आता है।

उदाहरण- MySQL>SELECT SQRT (64)

प्रश्न 24. MySQL में default यूजर कौन होता है? MySQL में नया यूजर बनाने का Syntax बताइए।

उत्तर- MySQL में default यूजर Root होता है। जिसकी सभी डाटाबेस पर फुल पहुँच (full access) होती है। My SQL में नया यूजर बनाने का Syntax है

MySQL> CREATE USER 'new_user' @ 'localhost'
IDENTIFIED BY 'Password';

लघु उत्तरीय प्रश्न

प्रश्न 1. प्राइमरी की (Primary key) पर संक्षिप्त टिप्पणी लिखिए।

उत्तर- प्राइमरी की-किसी टेबल में एक या अधिक फिल्ड (attribute) का ऐसा set जो कि उस टेबल की किसी भी पंक्ति अथवा टपल्स को uniquely identify करता हो तो इस attributes के सैट को collectively लेने पर यह उस टेबल की Primary key कहलाती है। जो एक प्रकार का constraints भी है। student टेबल की primary key, Roll_no फिल्ड है क्योंकि student टेबल में इस फिल्ड के द्वारा सभी छात्रों को uniquely identify किया जा सकता है। एवं Roll_no फिल्ड की सहायता से किसी छात्र का रिकार्ड टेबल से निकाला जा सकता है। जैसे कि अगर Roll_no फिल्ड की वैल्यू 105 लेने पर जो रिकार्ड टेबल से निकलेगा वह छात्र Kailash का होगा।

Student				
Roll_No	Name	Age	Address	Class
101	Harish	10	Ajmer	5 th
105	Kailash	20	Kota	10 th
109	Manish	18	Ahmadabad	9 th
120	Ronak	14	Udaipur	8 th
135	Shanker	13	Jaipur	7 th

प्रश्न 2. एंटीटी इन्टीग्रीटी कन्स्ट्रेन्ट्स के विषय में बताइए।

उत्तर- एंटीटी इन्टीग्रीटी कन्स्ट्रेन्ट्स-यह सुनिश्चित करता है कि किसी भी टेबल में कोई दो रिकार्ड्स या पंक्तियाँ . या टपल्स डुप्लीकेट नहीं हो सकते हैं। इसके अलावा वह फिल्ड जो प्रत्येक रिकार्ड की उस टेबल में पहचान कर रहा है वह एक यूनिक फिल्ड है तथा इस फिल्ड की वैल्यू कभी भी Null नहीं होगी।

एंटीग्रीडि कन्स्टेन्ट्स प्राइमरी की के द्वारा लगाया जा सकता है। हर एंटीग्रीडि के लिए अगर हम प्राइमरी की को परिभाषित करते हैं तो वह स्वतः ही एंटीग्रीडि की पूर्ति करता है। उदाहरणार्थ

Student table

Roll_No	Name	Address	Age	Class
110	Komal	Jaipur	17	12th
120	Kailash	Udaipur	14	8th
105	Manish	Kota	20	10th
107	Ronak	Chittorah	10	5th

उक्त Student टेबल की प्राइमरी की अगर Roll no फिल्ड्स है तो इस फिल्ड में प्रत्येक छात्र का Roll_no अलग-अलग होगा। साथ ही किसी छात्र के लिए उसकी वैल्यू Null नहीं होगी अर्थात् सभी का अपना अलग Roll no होगा अलग-अलग Roll_no की वजह से इस टेबल में कोई दो पंक्तियों एक समान नहीं होंगी।

प्रश्न 3. रेफरेन्शीयल इन्टीग्रीटी पर संक्षिप्त टिप्पणी लिखिए।

उत्तर- रेफरेन्शीयल इन्टीग्रीटी (Referential integrity)-अगर हम किसी प्रकार से यह सुनिश्चित करना चाहते हैं कि किसी रिलेशन में कुछ ऐट्रीब्यूट के लिए उनकी वैल्यू वही हो जो किसी अन्य रिलेशन में कुछ ऐट्रीब्यूट के लिए है। अर्थात् दोनों टेबल में कुछ ऐट्रीब्यूट वैल्यू एक समान हो तो यह शर्त रेफरेन्शीयल इन्टीग्रीटी कहलाती है। रेफरेन्शीयल इन्टीग्रीटी की सुनिश्चितता फोरेन के द्वारा की जा सकती है।

प्रश्न 4. SQL के प्रमुख लाभ बताइए।

उत्तर- SQL के लाभ-SQL सभी डाटाबेस सिस्टम चाहे कॉमर्शियल (Oracle, IBM, DB2, Sybase) हो या ऑपन सॉर्स (MySQL, Postgres) के लिए लाभदायक है। इसके लाभ निम्नलिखित हैं

- उच्च गति (High speed)-इस प्रकार की. लैंग्वेज है जो बड़े से बड़े ऑर्गेनाइजेशन के डाटाबेस से बहुत ही कुशलता (efficiently) के साथ तथा जल्दी-जल्दी डाटा निकाल सकती है अतः SQL एक उच्च गति की लैंग्वेज है।
- सीखने की सुविधा (Easy to learn)-SQL लैंग्वेज को सीखना बहुत ही आसान है क्योंकि इसमें प्रोग्राम का किसी प्रकार का लम्बा कोड नहीं होता है। अर्थात् इसमें ज्यादा कोडिंग की आवश्यकता नहीं होती है।
- अच्छी तरह से परिभाषित मानक लैंग्वेज (Well defined standard)-SQL एक मानक (standard) भाषा है जिसको स्टैंडराइज ANSI & ISO ने किया है।

प्रश्न 5. डेटा डेफिनेशन लैंग्वेज (DDL) का उपयोग बताइए।

उत्तर- डाटा डेफिनेशन लैंग्वेज (Data Definition Language)-DDL, SQL का एक भाग है। जिसकी सहायता से हम डाटाबेस स्कीमा को स्पेसीफाई कर सकते हैं।

DDL केवल डाटाबेस स्कीमा (schema) को ही specified नहीं करती बल्कि हर रिलेशन के बारे में भी स्पेसीफिकेशन रखती है। कुछ निम्न हैं

- हर रिलेशन के स्कीमा के लिए
- हर ऐट्रीब्यूट की वेल्यूज का डोमेन
- कन्स्ट्रेंट के लिए
- इन्डेक्स के लिए
- किसी स्कीमा के ऑथोराइजेशन एवम् सिक्योरिटी के लिए
- हर रिलेशन के फिजीकल स्टोरेज के लिए

प्रश्न 6. ALTER table कमाण्ड के विषय में बताइये।

उत्तर- ALTER table कमाण्ड्स-इन कमाण्ड्स का उपयोग किसी टेबल में नया स्तंभ जोड़ने (add), delete करने या modify करने में करते हैं। जिनके Syntax निम्न हैं

(1) नया स्तंभ (column) जोड़ने के लिए

ALTER TABLE table_name ADD column_name data type;

उदाहरण – Classes टेबल में एक नया column classes strength जोड़ सकते हैं।

Alter table classes add class_strength int

(2) कोई स्तंभ टेबल को हटाने के लिए

ALTER TABLE table_name DROP COLUMN column_name;

(3) किसी स्तंभी का डेटा टाइप्स बदलने के लिए

ALTER TABLE table_name MODIFY column_name data type;

उदाहरण-

ALTER TABLE Student MODIFY Age Date;

प्रश्न 7. किसी SQL query की बुनियादी संरचना के कितने क्लॉजेस (clauses) होते हैं?

उत्तर- किसी SQL query expression में मुख्य रूप से निम्नलिखित तीन क्लॉजेस clauses होते हैं।

अर्थात् कोई भी SQL query जो हम रिलेशन डाटाबेस के लिए लिखते हैं। उसकी बुनियादी संरचना (basic structure) में उक्त 3 क्लॉज होंगे।

1. SELECT क्लॉज में हम उन ऐट्रीब्यूट्स को लिखते हैं जो हमें हमारे आउटपुट रिलेशन में चाहिए।

2. FROM क्लॉज में हम उन रिलेशन को लिखते हैं। जिनको हमें query expression में उपयोग करना है। FROM क्लॉज में लिखे रिलेशनस का Cartesian product होता है।
3. WHERE क्लॉज में हम predicate लिखते हैं जो FROM clause के रिलेशनस के ऐट्रीब्यूट्स को लिप्त रखता है। अर्थात् जिसकी Boolean वेल्यू (true or false) होती है।

SQL query का निम्न फॉर्म (form) होता है।

SELECT AT1, AT2, AT3, ..., ATn,

FROM r1, r2, r3, rn,

WHERE P;

यहाँ AT_i एक ऐट्रीब्यूट को प्रदर्शित करता है और r_i एक रिलेशन को P एक predicate है।

प्रश्न 8. REVOKE कमाण्ड पर संक्षिप्त टिप्पणी लिखिए।

उत्तर- REVOKE कमाण्ड-REVOKE कमाण्डस का प्रयोग किसी ऑब्जेक्ट (टेबल) से सहमती वापिस लेने के लिए करते हैं। जिसकी Syntax GRANT के समान ही है।

Syntax: REVOKE [type of permission] ON[database_name]

[table_name] FROM [user name] '@localhost';

प्रश्न 9. Numeric फंक्शन के क्या उपयोग हैं? प्रमुख Numeric फंक्शन के नाम बताइए तथा उदाहरण भी दीजिए।

उत्तर- Numeric फंक्शन-इस फंक्शन का उपयोग गणितीय ऑपरेशन के काम में लेते हैं। कुछ उपयोगी फंक्शन निम्नलिखित हैं।

फंक्शन	विवरण
(i) ABS (V)	यह फंक्शन V की पूर्ण वेल्यू देता है।
(ii) GREATEST(n1, n2,...)	यह फंक्शन दी हुई पैरामीटर लिस्ट में से अधिकतम (greatest) वेल्यू देता है।
(iii) INTERVAL(N, n1,n2, n3, - -)	यह फंक्शन N की वेल्यू को लिस्ट (n1, n2, n3, - - -) से कम्पेयर करता है और अगर $N < n1$ है तो 0 $N < n2$ है तो 1, $N < n3$ है तो 2 और इसी प्रकार आगे नम्बर रिटर्न करता है,
(iv) LEAST (N1,N2....)	यह GREATEST का विपरीत है।

उदाहरण- (1)MySQL> SELECTABS (-6);

Output:

ABS(, 6)
6

उदाहरण-

(2)MySQL>SELECT GREATEST (4, 3, 7, 9, 8, 0, 10, 50, 70, 11)

Output :

GREATEST(4, 3, 7, 9, 8, 0, 10, 50, 70, 11)
70

उदाहरण-

(3)MySQL>SELECT INTERVAL (4, 3, 5, 8, 11, 12, 17, 18)

Output :

INTERVAL(4, 3, 5, 8, 11, 12, 17, 18)
1

निबंधात्मक प्रश्न

प्रश्न 1. डाटा कन्स्ट्रेंट्स (Data Constraints) से आप क्या समझते हैं? एक रिलेशन वाले कन्स्ट्रेंट्स के विषय में बताइए।

उत्तर- डाटा कन्स्ट्रेंट्स (Data constraints)-किसी टेबल के स्तंभों में इस तरह के नियम लागू करना है जो उस टेबल में डाटा की एन्ट्री की सीमा को निर्धारित करता है कि उस टेबल में केवल उसी प्रकार का डाटा एन्टर हो जो उस डाटाबेस की consistency, reliability एवं accuracy को सुनिश्चित कर सके एवं डाटाबेस में किसी प्रकार का बदलाव जब अधिकृत डाटाबेस यूजर्स करें तब भी डाटाबेस में किसी प्रकार का डाटा consistency loss न हो।

डाटा कन्स्ट्रेंट्स कॉलम (स्तंभ) लेवल ओर टेबल लेवल हो सकते हैं। कॉलम लेवल एवं टेबल लेवल कन्स्ट्रेंट्स में मुख्य अन्तर यह है कि कॉलम लेवल कन्स्ट्रेंट्स एक कॉलम में लगाये जाते हैं जबकि टेबल लेवल कन्स्ट्रेंट्स पूर्ण टेबल में लगाये जाते हैं। डाटा कन्स्ट्रेंट्स के निम्न उदाहरण है जैसे कि

- Student की Class null नहीं हो सकती है।
- किन्ही दो छात्रों के Roll_no एक समान नहीं होंगे।
- Student रिलेशन की हर एक Class रिलेशन में एक matching class जरूर रहेगी।

एक रिलेशन वाले कन्स्ट्रेंट्स

निम्नलिखित कन्स्ट्रेंट्स एक रिलेशन वाले कन्स्ट्रेंट्स हैं

1. Not null
2. Unique
3. Check (<predicate>)

(1) Not null कन्स्ट्रेंट्स – यह कन्स्ट्रेंट्स किसी भी टेबल में किसी फ़िल्ड या एट्रीब्यूट की null वेल्यू की एन્ટ्री को प्रतिबंधित करता है अर्थात् यदि किसी फ़िल्ड के लिए अगर यह कन्स्ट्रेंट्स लगा हुआ है। और उस टेबल में कोई ऑपरेशन जो उस टेबल में बदलाव कर अगर वेल्यू null डालने की कोशिश करता है तो वहाँ पर error जनरेट हो जाती है।

उदाहरण के तौर पर हम student टेबल में class एट्रीब्यूट की वेल्यू null नहीं चाहते हैं किसी प्रकार Roll_no एट्रीब्यूट की वेल्यू भी null नहीं होनी चाहिए क्योंकि वह उस टेबल की एक प्राइमरी की है।

(2) Unique कन्स्ट्रेंट्स – यह सुनिश्चित करता है कि कोई से दो टपल्स या पंक्तियाँ किसी रिलेशन में सभी प्राइमरी की एट्रीब्यूट पर बराबर नहीं हो सकती। अर्थात् दोनों टपल्स के लिए हर एट्रीब्यूट की वेल्यू एक समान नहीं होगी। Unique कन्स्ट्रेंट्स केन्डीडेड की (Key) फोर्म करता है जो कि किसी रिलेशन में एक से ज्यादा भी हो सकती है।

(3) Check कन्स्ट्रेंट्स – Check कन्स्ट्रेंट्स को हम डोमेन एवं रिलेशनल दोनों डिक्लेरेशन पर लगा सकते हैं। जब किसी रिलेशन डिक्लेरेशन पर लगा हो तो सभी टपल्स Check क्लॉज द्वारा specified condition को पूरा करेगा अर्थात् Check क्लॉज यह सुनिश्चित करता है किसी स्तंभ की सभी वेल्यूज उस पर लगी शर्त को पूरा करेगा।

प्रश्न 2. SQL के बेसिक डोमेन टाइप्स का वर्णन कीजिए।

अथवा

SQL के प्रमुख बिल्ट इन डोमेन टाइप्स के विषय में विस्तारपूर्वक बताइए।

उत्तर- SQL के बेसिक डोमेन टाइप्स-मुख्य रूप से उपयोग करने के लिए SQL के सभी बिल्ट इन डोमेन टाइप्स निम्न होते हैं।

(1) न्यूमेरिक डेटा टाइप्स – इसमें निम्न डेटा टाइप्स शामिल हैं

- Int या Integer – बड़े साइज के इन्टीजर्स के लिए है। इसके लिए 4 बाइट का स्टोरेज आवश्यक है।
- Small Int – छोटे साइज के इन्टीजर्स के लिए है। इसके लिए आवश्यक स्टोरेज बाइट 2 है।
- Tiny Int – अत्यधिक छोटे साइज के साइज या अनसाइज इन्टीजर्स के लिए।
- Float (M,D) – फ्लोटिंग पॉइंट नम्बर्स के लिए। यह केवल साइज नम्बर्स के लिए है यहाँ पर M उस नम्बर में कुल डिजिट्स है तथा D डेसिबल नम्बर की संख्या है।

- Double (M.D) – यह डेटा टाइप्स डबल प्रिसीजन वाले फ्लोटिंग पॉइन्ट नम्बर के लिए है तथा यह REAL का समानार्थक शब्द है।

(2) स्ट्रिंग टाइप्स (String Types) – MySQL में निम्न स्ट्रिंग डेटा टाइप्स हैं।

- CHAR (C) – यह डेटा टाइप फिक्सड लम्बाई की स्ट्रिंग के लिए है। यहाँ पर C इस स्ट्रिंग की लम्बाई है जो यूजर देता है। अगर इस लम्बाई से कम लम्बाई की कोई स्ट्रिंग अगर स्टोर करते हैं तो बचे हुए में space स्टोर होता है।
- VARCHAR (C) – यह वेरीयेबल लम्बाई की स्ट्रिंग के लिए है यहाँ पर C स्ट्रिंग की maximum लम्बाई है जिसको यूजर स्पेसीफाई करता है।

(3) Date और Time डेटा टाइप्स – My SQL में निम्नलिखित Time और Date डेटा टाइप्स हैं

- Date – इस डेटा टाइप्स में कोई Date, YYYY-MM-DD के प्रारूप में स्टोर होती है। तथा वह 1000-01-01 एवं 9999-12-31 के मध्य हो सकती है।
उदाहरणस्वरूप Student टेबल में किसी विद्यार्थी की age अगर 1 July, 1980 हो तो यह 1980-07-01 के प्रारूप में स्टोर होगी।
- DATE TIME – इस डेटा टाइप के द्वारा Date एवं Time के मेल को स्टोर कर सकते हैं। जिसका प्रारूप YYYYMM-DD, HH:MM:SS है। यह Date एवं Time 1000-01-01, 00:00:00 एवं 9999-12-31,23:59:59 के मध्य हो सकता है। उदाहरणस्वरूप मध्याह्न 2:35 दिनांक 1 July, 1980 को 1980-07-01, 14:35:00 इस तरह स्टोर कर सकते हैं।
- Time – यह समय को HH:MM:SS के प्रारूप में स्टोर करता है।
- Year – यह वर्ष को 2 डिजीट या 4 डिजीट के प्रारूप में स्टोर करता है।

प्रश्न 3. SQL ऑपरेटर्स क्या होते हैं? विवरण दें।

अथवा

SQL ऑपरेटर्स के प्रमुख कार्य बताएं।

उत्तर- SQL ऑपरेटर्स का परिचय – SQL ऑपरेटर्स एक प्रकार के रिजर्व (सुरक्षित) शब्द हैं। जो किसी SQL क्वेरी के WHERE क्लॉज में उपयोग किये जाते हैं जिनमें मुख्य ऑपरेशन निम्न हैं

- कम्पेरिजन (Comparison)
- अर्थ टिक (Arithmetic)
- लोजिकल (Logical)
- शर्त को निगेट करने के कार्य में आने वाले ऑपरेटर्स

(i) कम्पेरिजन ऑपरेटर्स – ऑपरेटर्स विवरण निम्नलिखित हैं।

ऑपरेटर्स	विवरण
(=)	यह ऑपरेटर्स दो ऑपरेटर्स की वेल्यू की समानता या असमानता को चेक करता है। समान होने पर शर्त true होगी।
(< > or !=)	यह ऑपरेटर्स दो ऑपरेटर्स की वेल्यू की समानता या असमानता को चेक करता है अगर वेल्यू समान नहीं है तो शर्त true होगी।
(>)	अगर बाँये तरफ के ऑपरेण्ड की वेल्यू दाँये तरफ के ऑपरेण्ड से ज्यादा हो तो शर्त true होगी।
(<)	अगर बाँये तरफ के ऑपरेण्ड की वेल्यू दाँये तरफ के ऑपरेण्ड से कम हो तो शर्त true होगी।
(>=)	अगर बाँये तरफ के ऑपरेण्ड की वेल्यू दाँये तरफ के ऑपरेण्ड से ज्यादा या समान हो तो शर्त true होगी।
(<=)	अगर बाँये तरफ के ऑपरेण्ड की वेल्यू दाँये तरफ के ऑपरेण्ड से कम या समान हो तो शर्त true होगी।

(ii) SQL अर्थमेटिक ऑपरेटर्स – ऑपरेटर्स विवरण निम्नलिखित है।

ऑपरेटर्स	विवरण
(+)	ऑपरेटर्स की दोनों ऑपरेण्ड की वेल्यू को जोड़ता (add) है।
(-)	बाँयी तरफ के ऑपरेण्ड की वेल्यू में से दाँयी तरफ की ऑपरेण्ड की वेल्यू का घटाव (subtract) करता है।
(*)	दोनों तरफ की ऑपरेण्ड की वेल्यू को गुणा करना।
(% or मॉड्यूलस)	बाँयी तरफ के ऑपरेण्ड की वेल्यू को दाँयी तरफ के ऑपरेण्ड की वेल्यू से भाग (divide) देता है। तथा उनका शेष रिटर्न करता है।

(iii) SQL लॉजिकल ऑपरेटर्स-ऑपरेटर्स विवरण निम्नलिखित हैं

ऑपरेटर्स	विवरण
----------	-------

(AND)	WHERE क्लॉज में विभिन्न शर्तों को allow करता है। अर्थात् a AND b दोनों शर्तों a एवं b true होने पर इसका परिणाम true होगा।
(OR)	यह WHERE क्लॉज में विभिन्न शर्तों को संयुक्त करता है। अर्थात् a OR b में कोई भी शर्त a या b के true होने पर इसका परिणाम true होगा।
Not (!)	यह ऑपरेटर ऑपरेन्ड की वेल्यू को इन्वर्स करता है।

(iv) शर्त को निगेट करने वाले ऑपरेटर्स

ऑपरेटर्स	विवरण
BETWEEN	यह ऑपरेटर्स वेल्यूस के मध्य वेल्यूस को सर्च करता है।
ANY	इस ऑपरेटर्स के द्वारा हम वेल्यू को किसी लिस्ट में अन्य वेल्यू से कम्पेयर (compare) करने के काम में लेते हैं।
ALL	इस ऑपरेटर के द्वारा हम एक वेल्यू को किसी अन्य लिस्ट की सभी वेल्यूस से compare करने के काम लेते हैं। अर्थात् ALL ऑपरेटर लिस्ट की सभी वेल्यूस के लिए शर्त सही होने पर ही true return करता है।
LIKE	इस ऑपरेटर का उपयोग किसी string में pattern matching के लिए करते हैं।

प्रश्न 4. SQL के विभिन्न फंक्शन का विवरण दें।

अथवा

SQL के built in फंक्शन के विषय में विस्तारपूर्वक बताइए।

उत्तर- SQL फंक्शन (functions) – SQL कई प्रकार के उपयोगी built in फंक्शन उपलब्ध करवाता है। जिनका विवरण निम्न प्रकार है

(1) Date और Time फंक्शन – इसमें काम आने वाले Date और Time फंक्शन तथा उनका विवरण नीचे दिया गया है

ऑपरेटर्स	विवरण
ADDDATE ()	Date को जोड़ता है।
ADDTIME ()	Time को जोड़ता है।
CURDATE ()	यह वर्तमान Date return करता है।
CURTIME ()	यह वर्तमान Time return करता है।
DATE_SUB ()	यह दो Dates को घटाता है।
NOW ()	यह वर्तमान Date व Time देता है।
STR_TO_DATE ()	यह स्ट्रिंग को Date में परिवर्तित करता है।

(2) स्ट्रिंग (String) फंक्शन – String के उपयोगी स्ट्रिंग फंक्शन निम्नानुसार हैं

ऑपरेटर्स	विवरण
ASCII ()	यह फंक्शन बाँये छोर (left most) के करेक्टर की numeric वेल्यू देगा।
BIN (N)	यह N की बाइनरी वेल्यू का स्ट्रिंग रिप्रजेन्टेशन (representation) देता है।
BIT_LENGTH(str)	यह str स्ट्रिंग की लम्बाई (length) बीट्स में देगा।
CHAR(N)	यह हर आरग्यूमेन्ट N को इन्टीजर मानकर उसका स्ट्रिंग रिप्रजेन्टेशन देगा। यह स्ट्रिंग उन करेक्टर्स से मिलकर बनेगी जो इन्टीजर CHAR (N) में आरग्यूमेन्ट के तौर पर है।
CHAR LENGTH(Sr)	यह Str स्ट्रिंग की लम्बाई करेक्टरस में नापता है।
CONCAT (Str1, Str2, ...)	यह आरग्यूमेन्ट स्ट्रिंग को concatenate करके प्राप्त स्ट्रिंग को return के तौर पर देता है।
FIELD (Str, Str1, Str2,...)	यह Str का इन्डेक्स दी हुई लिस्ट (Str1, Str2, . . .) में से रिटर्न करता है। अगर Str नहीं मिलती तो 0 रिटर्न करता है।
LOAD FILE (file name)	यह फंक्शन फाइल को read करे उसके कन्टेन्ट्स को स्ट्रिंग रूप में देता है। इसके उपयोग के लिए server पर उपस्थित फाइल का पूर्ण path name उल्लेखित करना होता है।
REPLACE (Str, from_Str, to Str)	यह Str स्ट्रिंग को उसमें से from_Str की सारी occurrences को to_Str से replace करके रिटर्न करता है।

उक्त फंक्शन के अलावा की अन्य कई स्ट्रिंग फंक्शन MySQL में है।