

सूचना प्रौद्योगिकी अभ्यास

कक्षा XII — CBSE / राज्य बोर्ड

अध्याय 1

SQL प्रश्न बैंक एवं फलन

सम्पूर्ण संस्करण — हिन्दी माध्यम

30 बहुविकल्पीय प्रश्न	20 रिक्त स्थान	20 अक्षर पूछे जाने वाले प्रश्न
लघु एवं दीर्घ उत्तर	मलिन प्रश्न	शिक्षक नोट्स सहित

यह प्रश्न बैंक वास्तविक शिक्षकीय अनुभव के आधार पर तैयार किया गया है। प्रत्येक प्रश्न CBSE परीक्षा पैटर्न और व्यावहारिक समझ को ध्यान में रखकर बनाया गया है।

खंड A: बहुविकल्पीय प्रश्न (MCQ)

प्रत्येक प्रश्न 1 अंक का है। सबसे उचित उत्तर चुनिए।

⇒ शिक्षक नोट: MCQ त्वरित याद और वैचारिक स्पष्टता की परीक्षा लेते हैं। प्रत्येक विकल्प ध्यान से पढ़ें — SQL फलन प्रश्नों में अक्सर भ्रामक विकल्प होते हैं!

Q1. निम्नलिखित में से कौन सा SQL में Single Row Function है?

- (a) COUNT()
- (b) SUM()
- (c) ROUND()
- (d) AVG()

✓ उत्तर: (c) ROUND()

📌 स्पष्टीकरण: Single Row Functions एक समय में एक पंक्ति पर काम करते हैं। ROUND(), POWER(), MOD() संख्यात्मक single row functions हैं।

Q2. SELECT POWER(3,4) क्या लौटाएगा?

- (a) 12
- (b) 64
- (c) 81
- (d) 34

✓ उत्तर: (c) 81

📌 स्पष्टीकरण: $POWER(x,y) = x$ को y की घात। $POWER(3,4) = 3 \times 3 \times 3 \times 3 = 81$ ।

Q3. SELECT ROUND(456.7896, 2) क्या लौटाएगा?

- (a) 456.78
- (b) 456.79
- (c) 456.80
- (d) 456.00

✓ उत्तर: (b) 456.79

📌 स्पष्टीकरण: ROUND(संख्या, दशमलव_स्थान) नरिदष्ट दशमलव स्थानों तक गोल करता है। 456.7896 को 2 दशमलव स्थानों तक गोल करने पर = 456.79।

Q4. SELECT MOD(17, 5) क्या लौटाता है?

- (a) 3
- (b) 2
- (c) 5
- (d) 1

✓ उत्तर: (b) 2

📌 स्पष्टीकरण: MOD(m,n) m को n से विभाजित करने पर शेषफल देता है। $17 \div 5 = 3$ शेष 2।

Q5. कौन सा function एक string को UPPERCASE में बदलता है?

- (a) LOWER()
- (b) UCASE()
- (c) MID()
- (d) LTRIM()

✓ उत्तर: (b) UCASE()

☒ स्पष्टीकरण: UCASE() या UPPER() किसी string को uppercase अक्षरों में बदलता है।

Q6. SELECT LENGTH('Informatics') क्या लौटाता है?

- (a) 10
- (b) 11
- (c) 12
- (d) 9

✓ उत्तर: (b) 11

☒ स्पष्टीकरण: LENGTH() स्थानों सहित कुल वर्णों की गणना करता है। 'Informatics' में 11 वर्ण हैं।

Q7. SELECT LEFT('Computer Science', 4) क्या देता है?

- (a) 'Comp'
- (b) 'Scie'
- (c) 'Compu'
- (d) 'nce'

✓ उत्तर: (a) 'Comp'

☒ स्पष्टीकरण: LEFT(string, n) सबसे बाईं ओर के n वर्ण निकालता है। 'Computer Science' के पहले 4 वर्ण = 'Comp'।

Q8. SELECT RIGHT('SCIENCE', 3) क्या लौटाता है?

- (a) 'SCI'
- (b) 'NCE'
- (c) 'CIE'
- (d) 'ENE'

✓ उत्तर: (b) 'NCE'

☒ स्पष्टीकरण: RIGHT(string, n) सबसे दाईं ओर के n वर्ण निकालता है। 'SCIENCE' के अंतिम 3 वर्ण = 'NCE'।

Q9. SELECT MID('Informatics', 3, 4) क्या लौटाता है?

- (a) 'Info'
- (b) 'form'
- (c) 'orma'
- (d) 'nfor'

✓ उत्तर: (b) 'form'

☒ स्पष्टीकरण: MID(string, pos, n) position pos से n वर्ण निकालता है। 'Informatics' में pos 3 से: f-o-r-m = 'form'।

Q10. कौन सा function वर्तमान सिस्टम दिनांक और समय लौटाता है?

- (a) DATE()
- (b) NOW()
- (c) MONTH()
- (d) YEAR()

✓ उत्तर: (b) NOW()

☒ स्पष्टीकरण: NOW() वर्तमान सिस्टम दिनांक और समय लौटाता है। DATE() किसी datetime अभिव्यक्ति से केवल date भाग निकालता है।

Q11. SELECT MONTH('2023-08-15') क्या लौटाता है?

- (a) 2023

- (b) 15
- (c) 8
- (d) 08

✓ उत्तर: (c) 8

☒ स्पष्टीकरण: MONTH(date) किसी दिनांक से माह संख्या लौटाता है। अगस्त = 8।

Q12. SELECT MONTHNAME('2023-11-26') क्या लौटाता है?

- (a) 'November'
- (b) '11'
- (c) 'Nov'
- (d) '26'

✓ उत्तर: (a) 'November'

☒ स्पष्टीकरण: MONTHNAME() माह का पूरा नाम लौटाता है, न कि संक्षिप्त रूप या संख्या।

Q13. कौन सा aggregate function NULL मानों को अनदेखा करता है?

- (a) COUNT(*)
- (b) SUM(column)
- (c) दोनों (a) और (b)
- (d) कोई नहीं

✓ उत्तर: (b) SUM(column)

☒ स्पष्टीकरण: COUNT(*) NULLs सहित सभी पंक्तियों को गणिता है। SUM(column), AVG(column), MIN(), MAX() सभी NULL मानों को अनदेखा करते हैं।

Q14. SELECT COUNT(*) FROM EMPLOYEE क्या गणिता है?

- (a) केवल non-NULL पंक्तियाँ
- (b) NULL designation वाली पंक्तियाँ
- (c) NULLs सहित सभी पंक्तियाँ
- (d) केवल distinct पंक्तियाँ

✓ उत्तर: (c) NULLs सहित सभी पंक्तियाँ

☒ स्पष्टीकरण: COUNT(*) NULL मानों की परवाह किए बिना तालिका की प्रत्येक पंक्ति को गणिता है। COUNT(column_name) NULLs को अनदेखा करता है।

Q15. GROUP BY के साथ समूहीकृत रिकॉर्ड को फ़िल्टर करने के लिए कौन सा clause उपयोग होता है?

- (a) WHERE
- (b) ORDER BY
- (c) HAVING
- (d) DISTINCT

✓ उत्तर: (c) HAVING

☒ स्पष्टीकरण: HAVING clause GROUP BY द्वारा बने समूहों को फ़िल्टर करता है। WHERE समूहीकरण से पहले व्यक्तिगत पंक्तियों को फ़िल्टर करता है।

Q16. दो तालिकाओं के बीच UNION operation के लिए क्या आवश्यक है?

- (a) केवल समान column संख्या
- (b) केवल समान data types
- (c) समान columns और compatible data types दोनों
- (d) कोई शर्त नहीं

✓ उत्तर: (c) समान columns और compatible data types दोनों

☒ स्पष्टीकरण: UNION, INTERSECT, MINUS के लिए — दोनों तालिकाओं में attributes की समान संख्या और समान domain होनी चाहिए।

Q17. INTERSECT operation क्या लौटाता है?

- (a) दोनों तालिकाओं की सभी पंक्तियाँ
- (b) केवल पहली तालिका की पंक्तियाँ
- (c) दोनों तालिकाओं में उपस्थित सामान्य पंक्तियाँ
- (d) केवल दूसरी तालिका की पंक्तियाँ

✓ उत्तर: (c) दोनों तालिकाओं में उपस्थित सामान्य पंक्तियाँ

☒ स्पष्टीकरण: INTERSECT केवल वे tuples/rows लौटाता है जो दोनों तालिकाओं में प्रकट होते हैं।

Q18. MINUS operation (Table A MINUS Table B) क्या लौटाता है?

- (a) A और B की सभी पंक्तियाँ
- (b) A में वे पंक्तियाँ जो B में नहीं हैं
- (c) B में वे पंक्तियाँ जो A में नहीं हैं
- (d) A और B की सामान्य पंक्तियाँ

✓ उत्तर: (b) A में वे पंक्तियाँ जो B में नहीं हैं

☒ स्पष्टीकरण: MINUS पहली तालिका से वे पंक्तियाँ लौटाता है जो दूसरी तालिका में नहीं हैं। इसे set difference भी कहते हैं।

Q19. क्रमशः 4 और 5 पंक्तियों वाली दो तालिकाओं का Cartesian Product कतिनी पंक्तियाँ देगा?

- (a) 9 पंक्तियाँ
- (b) 20 पंक्तियाँ
- (c) 1 पंक्ति
- (d) 45 पंक्तियाँ

✓ उत्तर: (b) 20 पंक्तियाँ

☒ स्पष्टीकरण: Cartesian Product = $m \times n$ पंक्तियाँ। $4 \times 5 = 20$ पंक्तियाँ। Degree (columns) = दोनों तालिकाओं के degrees का योग।

Q20. कौन सा JOIN प्रकार redundant common column को हटाता है?

- (a) Simple JOIN
- (b) Cartesian JOIN
- (c) NATURAL JOIN
- (d) CROSS JOIN

✓ उत्तर: (c) NATURAL JOIN

☒ स्पष्टीकरण: NATURAL JOIN स्वतः common columns पर match करता है और result में common column को दोहराता नहीं।

Q21. SELECT INSTR('Informatics', 'ma') क्या लौटाता है?

- (a) 7
- (b) 8
- (c) 6
- (d) 9

✓ उत्तर: (b) 8

☒ स्पष्टीकरण: INSTR(string, substring) पहली घटना की position लौटाता है। 'Informatics' में 'ma' position 8 से शुरू होता है।

Q22. SELECT LTRIM(' Hello') क्या लौटाता है?

- (a) ' Hello'
- (b) 'Hello'
- (c) 'Hello '
- (d) 'Hello'

✓ उत्तर: (b) 'Hello'

☒ स्पष्टीकरण: LTRIM किसी string से अग्रणी (बाईं ओर के) whitespace को हटाता है।

Q23. SELECT ROUND(156.862, -1) का output क्या है?

- (a) 156.9
- (b) 160
- (c) 150
- (d) 157

✓ उत्तर: (b) 160

☒ स्पष्टीकरण: जब decimal_places ऋणात्मक हो, ROUND दशमलव बिंदु के बाईं ओर काम करता है। -1 निकटतम 10 तक गोल करता है। 156.862 → 160।

Q24. MySQL में निम्न में से कौन सा Date function नहीं है?

- (a) DAY()
- (b) DAYNAME()
- (c) DATEDIFF()
- (d) DATESTR()

✓ उत्तर: (d) DATESTR()

☒ स्पष्टीकरण: DATESTR() MySQL में मौजूद नहीं है। DAY(), DAYNAME(), DATEDIFF() वैध MySQL date functions हैं।

Q25. SELECT AVG(Price) FROM INVENTORY क्या करेगा?

- (a) सबसे बड़ी कीमत लौटाएगा
- (b) सभी कीमतों का योग लौटाएगा
- (c) सभी कीमतों का औसत लौटाएगा
- (d) कीमतों की संख्या लौटाएगा

✓ उत्तर: (c) सभी कीमतों का औसत लौटाएगा

☒ स्पष्टीकरण: AVG(column) एक aggregate function है जो column के सभी non-NULL मानों का अंकगणितीय माध्य लौटाता है।

Q26. GROUP BY का उपयोग करते समय कौन सा सही है?

- (a) GROUP BY के साथ ORDER BY उपयोग नहीं हो सकता
- (b) HAVING, GROUP BY से पहले आता है
- (c) GROUP BY नरिदृष्ट column(s) में समान मानों वाली पंक्तियों को समूहीकृत करता है
- (d) GROUP BY केवल COUNT के साथ उपयोग हो सकता है

✓ उत्तर: (c) GROUP BY नरिदृष्ट column(s) में समान मानों वाली पंक्तियों को समूहीकृत करता है

☒ स्पष्टीकरण: GROUP BY समान मान साझा करने वाली पंक्तियों को समूहीकृत करता है। ORDER BY, GROUP BY के बाद आ सकता है, और HAVING, GROUP BY के बाद आता है।

Q27. R(3 attributes) और S(4 attributes) के Cartesian Product की degree क्या है?

- (a) 12
- (b) 3

(c) 4

(d) 7

✓ उत्तर: (d) 7

☒ स्पष्टीकरण: Cartesian Product की Degree = degrees का योग = 3 + 4 = 7 | Cardinality = cardinalities का गुणनफल ।

Q28. SELECT TRIM(' SQL ') क्या लौटाता है?

(a) ' SQL '

(b) 'SQL '

(c) 'SQL'

(d) 'SQL'

✓ उत्तर: (d) 'SQL'

☒ स्पष्टीकरण: TRIM() किसी string से अग्रणी और अनुगामी दोनों whitespace हटाता है ।

Q29. केवल unique/distinct समूहीकृत मान प्रदर्शित करने के लिए कौन सा SQL clause उपयोग होता है?

(a) WHERE

(b) UNIQUE

(c) HAVING

(d) DISTINCT

✓ उत्तर: (d) DISTINCT

☒ स्पष्टीकरण: DISTINCT result set से duplicate मानों को समाप्त करता है । उदाहरण: SELECT DISTINCT Model FROM INVENTORY ।

Q30. JOIN operation में, table alias क्या है?

(a) तालिका का स्थायी नाम बदलना

(b) केवल उस query के लिए वैध एक संक्षिप्त नाम

(c) हर query में अनिवार्य रूप से आवश्यक

(d) केवल subqueries में उपयोग होता है

✓ उत्तर: (b) केवल उस query के लिए वैध एक संक्षिप्त नाम

☒ स्पष्टीकरण: Table aliases (जैसे DANCE के लिए D, MUSIC के लिए M) अस्थायी हैं और केवल वर्तमान query में वैध हैं । ये अस्पष्टता दूर करने में मदद करते हैं ।

खंड B: रिक्त स्थान भरिए

प्रत्येक रिक्त स्थान 1 अंक का है।

⇒ शिक्षक नोट: रिक्त स्थान में SQL functions की वर्तनी महत्वपूर्ण है। Function का नाम बिल्कुल सही लिखें — जैसे MONTHNAME, Monthname नहीं।

1. वे functions जो एक पंक्ति पर काम करते हैं और प्रत्येक पंक्ति के लिए एक परिणाम देते हैं, उन्हें _____ functions कहते हैं।

✓ उत्तर: Single Row (Scalar) Functions

2. वे functions जो पंक्तियों के एक समूह पर काम करते हैं और एक ही परिणाम देते हैं, उन्हें _____ functions कहते हैं।

✓ उत्तर: Aggregate (Multiple Row) Functions

3. Function _____ एक संख्या को दूसरे से विभाजित करने पर शेषफल लौटाता है।

✓ उत्तर: MOD()

4. SELECT _____ (Price, 2) Price column को 2 दशमलव स्थानों तक गोल करता है।

✓ उत्तर: ROUND

5. String function _____ किसी string में वर्णों की संख्या लौटाता है।

✓ उत्तर: LENGTH()

6. 'Informatics' से position 1 से लंबाई 3 के साथ 'Inf' substring निकालने के लिए, हम _____ (string, 1, 3) उपयोग करते हैं।

✓ उत्तर: MID या SUBSTR

7. Function _____ किसी string से अनुगामी whitespace हटाता है।

✓ उत्तर: RTRIM()

8. SELECT _____ ('hello world') string को 'HELLO WORLD' में बदलेगा।

✓ उत्तर: UCASE() या UPPER()

9. किसी दी गई दिनांक के लिए दिन का नाम लौटाने वाला date function _____ है।

✓ उत्तर: DAYNAME()

10. SELECT YEAR('2017-08-12') मान _____ लौटाएगा।

✓ उत्तर: 2017

11. Aggregate function जो column में सबसे बड़ा मान लौटाता है _____ है।

✓ उत्तर: MAX()

12. Aggregate function जो किसी numeric column का कुल योग परिकल्पित करता है _____ है।

✓ उत्तर: SUM()

13. SQL में _____ clause का उपयोग GROUP BY द्वारा बनाए गए समूहों को फ़िल्टर करने के लिए किया जाता है।

✓ उत्तर: HAVING

14. वह operation जो दो तालिकाओं की पंक्तियों को मिलाता है और सामान्य पंक्तियों को केवल एक बार दिखाता है _____ कहलाता है।

✓ उत्तर: UNION

15. _____ operation केवल वे पंक्तियाँ लौटाता है जो दोनों तालिकाओं में उपस्थित हैं।

✓ उत्तर: INTERSECT

16. _____ operation पहली तालिका से वे पंक्तियाँ लौटाता है जो दूसरी तालिका में नहीं हैं।

✓ उत्तर: MINUS (Set Difference)

17. 5 और 6 cardinalities वाले दो relations के Cartesian Product में _____ पंक्तियाँ होंगी।

✓ उत्तर: 30

18. JOIN operation में, _____ JOIN स्वतः result से repeated common attribute को हटाता है।

✓ उत्तर: NATURAL

19. Yahoo email ID वाले customers के विवरण प्रदर्शित करने के लिए: SELECT * FROM CUSTOMER WHERE Email LIKE _____।

✓ उत्तर: '%yahoo%'

20. Function _____ किसी string में एक substring की position लौटाता है।

✓ उत्तर: INSTR()

खंड C: अक्षर पूछे जाने वाले प्रश्न (FAQs)

ये प्रश्न बोर्ड परीक्षाओं में सामान्यतः पूछे जाते हैं।

⇒ शिक्षक नोट: FAQ उत्तर अपने शब्दों में लिखने चाहिए। परभाषाएँ शब्द-दर-शब्द याद करने से अवधारणा को समझकर स्वाभाविक रूप से समझाना अधिक उपयोगी है।

Q1. Single Row functions और Aggregate functions में क्या अंतर है?

Single Row functions (Scalar functions भी कहते हैं) एक समय में एक पंक्ति पर काम करते हैं और प्रत्येक पंक्ति एक परिणाम लौटाते हैं। उदाहरण के लिए, ROUND(Price, 2) प्रत्येक पंक्ति के लिए व्यक्तिगत रूप से गोल मान देता है।

Aggregate functions (Multiple Row functions) पंक्तियों के एक समूह पर काम करते हैं और पूरे समूह के लिए एक ही परिणाम देते हैं। उदाहरण के लिए, SUM(Price) सभी कीमतें जोड़कर एक कुल देता है।

मुख्य अंतर: single row → अनेक परिणाम; aggregate → एक परिणाम।

Q2. SQL में कनिहीं तीन Numeric functions का नाम और स्पष्टीकरण दीजिए।

तीन सामान्यतः उपयोग किए जाने वाले Numeric functions:

(i) POWER(X, Y) — X को Y की घात लौटाता है। उदाहरण: POWER(2,5) = 32। (ii) ROUND(N, D) — N को D दशमलव स्थानों तक गोल करता है। यदि D = 0 है तो निकटतम पूर्ण संख्या तक गोल करता है। उदाहरण: ROUND(4.567, 2) = 4.57। (iii) MOD(M, N) — M को N से विभाजित करने पर शेषफल लौटाता है। उदाहरण: MOD(10, 3) = 1।

Q3. GROUP BY clause का उद्देश्य क्या है? उदाहरण दीजिए।

GROUP BY clause का उपयोग समान डेटा को समूहों में व्यवस्थित करने के लिए किया जाता है। यह उन पंक्तियों को समूहीकृत करता है जिनके निर्दिष्ट column(s) में समान मान हों, ताकि aggregate functions प्रत्येक समूह पर अलग-अलग लागू किए जा सकें।

उदाहरण: SELECT CustID, COUNT(*) AS 'कार की संख्या' FROM SALE GROUP BY CustID;

यह query सभी बिक्री रिकॉर्ड को customer ID द्वारा समूहीकृत करती है और गिनती है कि प्रत्येक customer ने कितनी कारें खरीदीं। GROUP BY के बिना, COUNT(*) सभी पंक्तियों को एक साथ गिनता।

Q4. WHERE और HAVING clauses में अंतर स्पष्ट कीजिए।

WHERE clause समूहीकरण से पहले व्यक्तिगत पंक्तियों को फिल्टर करता है। इसे aggregate functions के साथ उपयोग नहीं किया जा सकता।

HAVING clause GROUP BY operation के बाद समूहों को फिल्टर करता है। यह विशेष रूप से aggregate functions के साथ उपयोग किया जाता है।

उदाहरण: SELECT EmpID, COUNT(*) FROM SALE GROUP BY EmpID HAVING COUNT(*) > 1;

यहाँ HAVING केवल उन employees को रखता है जिन्होंने एक से अधिक बिक्री की। WHERE यह नहीं कर सकता था।

Q5. UNION operation क्या है? इसकी क्या शर्तें हैं?

UNION operation दो तालिकाओं की पंक्तियों को मिलाता है और दोनों से सभी unique पंक्तियाँ प्रदर्शित करता है, duplicates को हटाकर। इसे U प्रतीक द्वारा दर्शाया जाता है।

UNION (और अन्य set operations) के लिए शर्तें: 1. दोनों तालिकाओं में attributes की समान संख्या होनी चाहिए। 2. दोनों तालिकाओं में corresponding attributes का समान domain (data type) होना चाहिए।

Syntax: SELECT columns FROM Table1 UNION SELECT columns FROM Table2;

Q6. उदाहरण सहित INTERSECT operation समझाइए।

INTERSECT operation केवल वे पंक्तियाँ लौटाता है जो दोनों तालिकाओं में उपस्थित हैं। यह सामान्य tuples दिखाता है।

उदाहरण: यदि DANCE तालिका में छात्र {Aastha, Mohit, Mahira, Sanjay} हैं और MUSIC तालिका में {Mohak, Mahira, Lavanya, Sanjay, Abhay} हैं, तो: DANCE INTERSECT MUSIC = {Mahira, Sanjay}

ये वे छात्र हैं जो दोनों events में नामांकित हैं। INTERSECT को \cap प्रतीक द्वारा दर्शाया जाता है।

Q7. Cartesian Product क्या है? इसकी degree और cardinality कैसे परिकलित होती है?

Cartesian Product पहले relation के प्रत्येक tuple को दूसरे relation के प्रत्येक tuple के साथ जोड़ता है। यदि Table A में 'm' पंक्तियाँ और Table B में 'n' पंक्तियाँ हैं, तो Cartesian Product में $m \times n$ पंक्तियाँ होती हैं।

परिणाम की Degree = दोनों तालिकाओं के degrees का योग (यानी, कुल combined columns की संख्या)। परिणाम की Cardinality = cardinalities का गुणनफल = $m \times n$ ।

उदाहरण: DANCE (4 पंक्तियाँ, 3 columns) \times MUSIC (5 पंक्तियाँ, 3 columns) = 20 पंक्तियाँ, 6 columns।

Q8. NATURAL JOIN क्या है? यह simple JOIN से कैसे भिन्न है?

NATURAL JOIN दो तालिकाओं को उनके common attribute(s) के आधार पर जोड़ता है और परिणाम से duplicate common column को स्वतः हटाता है।

Simple JOIN (WHERE या ON clause के साथ) common column की दोनों प्रतियाँ रख सकता है, जिसके लिए उन्हें अलग करने हेतु table aliases का उपयोग आवश्यक होता है।

Syntax: SELECT * FROM UNIFORM NATURAL JOIN COST;

NATURAL JOIN का परिणाम condition के साथ JOIN जैसा ही होता है, लेकिन common column केवल एक बार आता है, जिससे output अधिक स्वच्छ होता है।

Q9. उदाहरणों सहित कनिही चार String functions समझाइए।

चार उपयोगी String functions:

(i) UCASE('hello') \rightarrow 'HELLO' — Uppercase में बदलता है। (ii) MID('Informatics', 3, 4) \rightarrow 'form' — Position 3 से 4 वर्ण निकालता है। (iii) LENGTH('Computer') \rightarrow 8 — वर्णों की गणना लौटाता है। (iv) INSTR('Informatics', 'ma') \rightarrow 8 — string में 'ma' की position लौटाता है।

Q10. SQL में Date और Time functions क्या है? तीन उदाहरण दीजिए।

Date और Time functions SQL के built-in functions हैं जो तालिका में संग्रहीत date या time डेटा पर operations करते हैं।

(i) NOW() — वर्तमान सिस्टम दिनांक और समय लौटाता है। उदाहरण: 2025-07-11 10:41:17 (ii) MONTHNAME(date) — माह का पूरा नाम लौटाता है। MONTHNAME('2023-11-26') \rightarrow 'November' (iii) DAYNAME(date) — दिन का नाम लौटाता है। DAYNAME('2023-03-24') \rightarrow 'Friday'

ये दिनांकों पर आधारित रिपोर्ट तैयार करने के लिए विशेष रूप से उपयोगी हैं।

Q11. EMPLOYEE तालिका से ₹30,000 से अधिक वेतन वाले employees के नाम प्रदर्शित करने की query लिखिए।

Query: SELECT EmpName, Salary FROM EMPLOYEE WHERE Salary > 30000;

स्पष्टीकरण: WHERE clause पंक्तियों को फ़िल्टर करता है — केवल वे employees जिनके Salary column का मान 30000 से अधिक है, परिणाम में प्रकट होंगे।

Q12. JOIN query में table alias का उपयोग कैसे होता है? उदाहरण दीजिए।

Table alias किसी तालिका को एक संक्षिप्त, अस्थायी नाम देता है जो केवल उस विशेष query में वैध होता है। यह तब अस्पष्टता से बचने में मदद करता है जब दो तालिकाओं में समान नाम के columns हों।

उदाहरण: SELECT * FROM DANCE D, MUSIC M WHERE D.Name = M.Name;

यहाँ 'D' DANCE का alias है और 'M' MUSIC का alias है। D.Name DANCE तालिका के Name column को संदर्भित करता है। Alias केवल इस query में वैध है — तालिका का मूल नाम नहीं बदलता।

Q13. COUNT(*) और COUNT(column_name) का महत्व क्या है?

COUNT(*) तालिका की सभी पंक्तियाँ गनता है, जिसमें किसी भी column में NULL मान वाली पंक्तियाँ भी शामिल हैं।

COUNT(column_name) केवल वे पंक्तियाँ गनता है जहाँ निर्दिष्ट column NOT NULL है। यह NULL मानों को अनदेखा करता है।

उदाहरण: यदि MANAGER तालिका में 4 पंक्तियाँ हैं लेकिन एक पंक्ति में MNAME column NULL है: SELECT COUNT(*) \rightarrow 4 लौटाता है SELECT COUNT(MNAME) \rightarrow 3 लौटाता है

Q14. उदाहरण सहित MINUS (set difference) operation समझाइए।

MINUS operation पहली तालिका से वे सभी पंक्तियाँ लौटाता है जो दूसरी तालिका में नहीं हैं। यह दिखाता है कि पहली तालिका में क्या unique है।

उदाहरण: DANCE MINUS MUSIC = DANCE में वे छात्र जो MUSIC में नहीं हैं। DANCE = {Aastha, Mohit, Mahira, Sanjay}; MUSIC = {Mohak, Mahira, Lavanya, Sanjay, Abhay} DANCE MINUS MUSIC = {Aastha, Mohit}

नोट: MINUS MySQL में supported नहीं है लेकिन Oracle SQL में उपलब्ध है। MySQL में NOT IN या LEFT JOIN उपयोग करते हैं।

Q15. Aggregate functions से आप क्या समझते हैं? पाँचों का नाम और उद्देश्य बताइए।

Aggregate functions पंक्तियों के एक समूह पर काम करते हैं और परिणाम के रूप में एक ही मान लौटाते हैं। इन्हें Multiple Row functions भी कहते हैं।

पाँच Aggregate functions: 1. MAX(col) — column में अधिकतम (सबसे बड़ा) मान लौटाता है। 2. MIN(col) — न्यूनतम (सबसे छोटा) मान लौटाता है। 3. AVG(col) — औसत (माध्य) मान लौटाता है। 4. SUM(col) — सभी मानों का योग लौटाता है। 5. COUNT(col) — non-NULL मानों की गणना लौटाता है।

ये सभी (COUNT(*) को छोड़कर) NULL मानों को अनदेखा करते हैं।

Q16. एक से अधिक बार उपयोग किया गया payment mode कैसे प्रदर्शित करें?

PaymentMode द्वारा group करना और फिर count > 1 वाले groups को filter करना होगा:

Query: SELECT PaymentMode, COUNT(PaymentMode) FROM SALE GROUP BY PaymentMode HAVING COUNT(PaymentMode) > 1;

यह सभी बिक्री को payment mode द्वारा समूहीकृत करती है, प्रत्येक समूह में घटनाओं की गणना करती है, और HAVING केवल उन modes को दिखाने के लिए filter करता है जो एक से अधिक बार उपयोग हुए।

Q17. नमिनलखिति का output क्या है: SELECT LCASE('INFORMATION PRACTICES')?

Output: 'information practices'

LCASE() (या LOWER()) किसी string के सभी uppercase अक्षरों को lowercase में बदलता है। यह spaces, numbers या special characters को प्रभावित नहीं करता।

Q18. SQL में LIKE operator का % और _ wildcards के साथ उपयोग समझाइए।

LIKE operator का उपयोग WHERE clause में किसी column में निर्दिष्ट pattern खोजने के लिए किया जाता है।

% (percent) wildcard किसी भी प्रकार के शून्य या अधिक वर्णों से मेल खाता है। उदाहरण: WHERE Email LIKE '%yahoo%' — कहीं भी 'yahoo' वाले emails ढूँढता है।

_ (underscore) wildcard ठीक एक वर्ण से मेल खाता है। उदाहरण: WHERE Model LIKE 'L_I' — 'LXI', 'LZI' आदि models ढूँढता है।

ये string columns में आंशिक डेटा खोजने के लिए बहुत उपयोगी हैं।

Q19. INVENTORY तालिका में उपलब्ध विभिन्न models की कुल संख्या ज्ञात करने की query लिखिए।

Query: SELECT COUNT(DISTINCT Model) FROM INVENTORY;

स्पष्टीकरण: DISTINCT सुनिश्चित करता है कि प्रत्येक unique model केवल एक बार गिना जाए, भले ही वह कई बार प्रकट हो। COUNT() फिर उन distinct मानों को गणना करता है।

Q20. GROUP BY query में ORDER BY की क्या भूमिका है?

ORDER BY अंतिम परिणाम को ascending (ASC, default) या descending (DESC) क्रम में sort करता है।

GROUP BY के साथ उपयोग करने पर, ORDER BY HAVING clause के बाद रखा जाता है और समूहीकृत परिणामों को sort करता है।

उदाहरण: SELECT PaymentMode, COUNT(PaymentMode) FROM SALE GROUP BY PaymentMode ORDER BY PaymentMode;

यह payment modes और उनकी गणना दिखाता है, payment mode name के अनुसार alphabetically sorted।

खंड D: लघु उत्तर प्रश्न (2-3 अंक)

संक्षेप में और बटु पर लखिए — जहाँ भी संभव हो उदाहरण दीजिए।

⇒ शकषक नोट: लघु उत्तर प्रश्नों में सटीकता अपेक्षत है। सीधे बटु पर लखें — लंबी भूमकल की आवश्यकता नहीं।

Q1. INVENTORY तालकल में 'Discount' नामक नया column (numeric, 5 digits, 2 decimal places) जोड़ने की SQL query लखिए।

```
ALTER TABLE INVENTORY ADD Discount NUMERIC(5,2);
```

☞ यह मौजूदा तालकल में नया column जोड़ने के लिए ADD clause के साथ ALTER TABLE command का उपयोग करता है।

Q2. INVENTORY तालकल से car name और GST (Price का 12%), 1 decimal place तक गोल करके प्रदर्शत करने की query लखिए।

```
SELECT CarName, ROUND((12/100)*Price, 1) 'GST' FROM INVENTORY;
```

☞ ROUND() calculated GST मान पर लागू होता है ताकल इसे 1 decimal place तक प्रदर्शत कलया जा सके।

Q3. CUSTOMER तालकल से customer names lowercase में और email uppercase में प्रदर्शत करने की query लखिए।

```
SELECT LCASE(CustName), UCASE(Email) FROM CUSTOMER;
```

☞ LCASE() lowercase में बदलता है; UCASE() uppercase में बदलता है। ये string functions हैं।

Q4. नमूनलखत queries का output क्या होगा? (a) SELECT POWER(2,10) (b) SELECT MOD(100,7)

(a) $POWER(2,10) = 2^{10} = 1024$

(b) $MOD(100,7) = 100 \div 7 = 14$ शेष 2 → Output: 2

☞ POWER(base, exp) घातांक की गणना करता है। MOD(m,n) शेषफल लौटाता है।

Q5. EMPLOYEE तालकल से सभी employees के joining के day, month number और year प्रदर्शत करने की query लखिए।

```
SELECT DAY(DOJ), MONTH(DOJ), YEAR(DOJ) FROM EMPLOYEE;
```

☞ DAY(), MONTH(), YEAR() date functions हैं जो date column से संबंधत भाग निकालते हैं।

Q6. उदाहरण सहत समझाइए कल GROUP BY के साथ HAVING और WHERE कैसे भनून है।

WHERE समूहीकरण से पहले filter करता है:

```
SELECT CustID FROM SALE WHERE SalePrice > 500000 GROUP BY CustID;
```

HAVING समूहीकरण के बाद filter करता है:

```
SELECT CustID, COUNT(*) FROM SALE GROUP BY CustID HAVING COUNT(*) > 1;
```

☞ HAVING aggregate functions का उपयोग कर सकता है; WHERE नहीं कर सकता।

Q7. SALE तालकल से सभी ववरण जहाँ payment mode 'Bank Finance' है, प्रदर्शत करने की query लखिए।

```
SELECT * FROM SALE WHERE PaymentMode = 'Bank Finance';
```

☑ PaymentMode पर equality condition के साथ WHERE clause केवल वे पंक्तियाँ filter करता है जहाँ payment Bank Finance के माध्यम से हुई।

Q8. SALE से InvoiceNo और Commission (SalePrice का 12%, 0 decimal places तक गोल) जहाँ Commission > 73000 हो, प्रदर्शित करने की query लिखिए।

```
SELECT InvoiceNo, ROUND(Commission, 0) FROM SALE WHERE Commission > 73000;
```

☑ (यह मानते हुए कि Commission column पहले से SalePrice का 12% है।)

Q9. SELECT SUBSTR('Informatics', 3, 7) का output क्या है?

SUBSTR('Informatics', 3, 7) position 3 से शुरू होकर 7 वर्ण निकालता है।

Positions: I(1) n(2) f(3) o(4) r(5) m(6) a(7) t(8) i(9) c(10) s(11)

Position 3 (f) से, 7 वर्ण: f-o-r-m-a-t-i

Output: 'formati'

☑ MID और SUBSTR दोनों string से substring निकालने का काम करते हैं।

Q10. EMPLOYEE तालिका से प्रत्येक designation में employees की कुल संख्या गनने की query लिखिए।

```
SELECT Designation, COUNT(*) 'Total Employees' FROM EMPLOYEE GROUP BY Designation;
```

☑ GROUP BY Designation समान designation वाली पंक्तियों को समूहीकृत करता है, और COUNT(*) प्रत्येक समूह में employees गनता है।

खंड E: दीर्घ उत्तर प्रश्न (4-5 अंक)

स्पष्ट संरचना अपनाएँ — परभाषा, स्पष्टीकरण, syntax, उदाहरण।

⇒ शिक्षक नोट: दीर्घ उत्तरों में स्पष्ट संरचना होनी चाहिए। जहाँ उपयुक्त हो, diagram या table बनाएँ।

Q1. SQL में Single Row Functions की तीन श्रेणियों को दो-दो उदाहरणों सहित समझाइए।

Single Row functions प्रत्येक पंक्ति पर अलग-अलग लागू होते हैं और प्रत्येक पंक्ति एक परिणाम देते हैं। इन्हें तीन श्रेणियों में वर्गीकृत किया गया है:

(A) Numeric Functions — संख्यात्मक डेटा पर काम करते हैं और संख्यात्मक परिणाम देते हैं।

उदाहरण:

- POWER(2,8) → 256 — 2 को 8 की घात देता है।
- ROUND(345.678, 1) → 345.7 — 1 दशमलव स्थान तक गोल करता है।

(B) String Functions — Character data पर काम करते हैं और character या numeric परिणाम देते हैं।

उदाहरण:

- UCASE('hello') → 'HELLO' — uppercase में बदलता है।
- LENGTH('Computer') → 8 — वर्ण गणित है।

(C) Date और Time Functions — date/time input स्वीकार करते हैं और date, time या numeric परिणाम देते हैं।

उदाहरण:

- MONTHNAME('2023-04-15') → 'April' — माह का नाम लौटाता है।
- DAY('2023-08-26') → 26 — दिन की संख्या निकालता है।

सारांश: तीनों श्रेणियाँ SQL को वास्तविक जीवन में डेटा संसाधित करने और प्रस्तुत करने में अत्यंत लचीला बनाती हैं।

Q2. Relations पर चार set operations — UNION, INTERSECT, MINUS और Cartesian Product — का उदाहरण सहित वर्णन करें।

(i) UNION (U) — दोनों तालिकाओं की सभी unique पंक्तियों को मिलाता है। Duplicate पंक्तियाँ केवल एक बार प्रकट होती हैं।

शर्त: समान column संख्या, compatible data types।

उदाहरण: DANCE U MUSIC = किसी भी event के सभी छात्र (बिना repetition के)।

(ii) INTERSECT (∩) — केवल वे पंक्तियाँ लौटाता है जो दोनों तालिकाओं में सामान्य हों।

उदाहरण: DANCE ∩ MUSIC = dance और music दोनों में नामांकित छात्र।

(iii) MINUS (−) — पहली तालिका में उपस्थित पंक्तियाँ लौटाता है जो दूसरी में नहीं हैं।

उदाहरण: DANCE − MUSIC = DANCE में पर MUSIC में नहीं वाले छात्र।

(iv) Cartesian Product (×) — पहली तालिका की प्रत्येक पंक्ति को दूसरी की प्रत्येक पंक्ति के साथ जोड़ता है।

Degree of result = degree(R) + degree(S)

Cardinality of result = cardinality(R) × cardinality(S)

उदाहरण: DANCE (4 rows) × MUSIC (5 rows) = 20 rows।

महत्वपूर्ण: UNION, INTERSECT, MINUS के लिए समान structure आवश्यक है। Cartesian Product कनिहीं दो तालिकाओं पर काम करता है।

Q3. SQL में JOIN operation समझाइए। UNIFORM और COST तालिकाओं से JOIN query लिखने के तीन तरीके बताइए।

JOIN किसी common attribute या condition के आधार पर दो तालिकाओं के tuples को जोड़ता है। यह condition के साथ Cartesian Product का वस्तुतः है।

UNIFORM Table: UCode, UName, UColor | COST Table: UCode, Size, Price | Common attribute: UCode

(A) WHERE clause का उपयोग (Implicit JOIN):

```
SELECT * FROM UNIFORM, COST WHERE UNIFORM.UCode = COST.UCode;
```

यहाँ WHERE clause में join condition मैन्युअल रूप से निर्दिष्ट की गई है।

(B) JOIN...ON clause का उपयोग (Explicit JOIN):

```
SELECT * FROM UNIFORM JOIN COST ON UNIFORM.UCode = COST.UCode;
```

ON clause condition निर्दिष्ट करता है। WHERE clause में कोई condition देने की आवश्यकता नहीं।

(C) NATURAL JOIN का उपयोग:

```
SELECT * FROM UNIFORM NATURAL JOIN COST;
```

SQL स्वतः common column (UCode) की पहचान करता है और उस पर join करता है। Result में common column केवल एक बार आता है — duplicate नहीं।

महत्वपूर्ण बिंदु:

- N तालिकाओं के लिए, N-1 JOIN conditions आवश्यक हैं।
- Table aliases (जैसे UNIFORM के लिए U) दोनों तालिकाओं में समान column names की अस्पष्टता दूर करते हैं।
- NATURAL JOIN सबसे स्वच्छ syntax है क्योंकि यह redundancy से बचता है।

Q4. HAVING के साथ GROUP BY clause समझाइए। CARSHOWROOM database के लिए कम से कम तीन SQL queries दीजिए।

GROUP BY clause उन पंक्तियों को समूहीकृत करता है जिनके निर्दिष्ट columns में समान मान हों। Aggregate functions फिर प्रत्येक समूह पर अलग-अलग लागू होते हैं। HAVING clause aggregate conditions के आधार पर समूहों को filter करता है।

Syntax: SELECT column, aggregate_function FROM table GROUP BY column HAVING condition ORDER BY column;

उदाहरण 1 — प्रत्येक customer द्वारा खरीदी गई कारों की गणना:

```
SELECT CustID, COUNT(*) 'Number of Cars' FROM SALE GROUP BY CustID;
```

उदाहरण 2 — एक से अधिक कार खरीदने वाले customers:

```
SELECT CustID, COUNT(*) FROM SALE GROUP BY CustID HAVING COUNT(*) > 1;
```

उदाहरण 3 — Payment mode द्वारा समूहीकृत कुल बिक्री और गणना:

```
SELECT PaymentMode, COUNT(PaymentMode), SUM(SalePrice) FROM SALE GROUP BY PaymentMode ORDER BY PaymentMode;
```

मुख्य अंतर: WHERE aggregate functions के साथ उपयोग नहीं हो सकता; HAVING इसी के लिए बना है।

Query में क्रम: GROUP BY → HAVING → ORDER BY।

Q5. CARSHOWROOM database schema बनाइए। (a) 12% GST जोड़कर FinalPrice, (b) 10 कस्तियों में EMI, (c) प्रत्येक EMI को नकिटतम 1000 तक गोल करने के बाद शेष राशि दिखाने की queries लिखिए।

Database: CARSHOWROOM

Table: INVENTORY (CarID, CarName, Price, Model, YearManufacture, FuelType)

Step 1 — FinalPrice column जोड़ना:

```
ALTER TABLE INVENTORY ADD FinalPrice NUMERIC(10,1);
```

```
UPDATE INVENTORY SET FinalPrice = ROUND(Price + (12/100)*Price, 1);
```

(a) GST सहित FinalPrice प्रदर्शित करना:

```
SELECT CarID, CarName, Price, ROUND((12/100)*Price, 1) 'GST', FinalPrice FROM INVENTORY;
```

(b) Monthly EMI (FinalPrice ÷ 10 कस्तियों) परिकलित करना:

```
SELECT CarID, FinalPrice, ROUND(FinalPrice/10, 0) 'EMI' FROM INVENTORY;
```

(c) EMI को नकिटतम 1000 तक गोल करने के बाद शेष राशि:

```
SELECT CarID, FinalPrice, ROUND(FinalPrice/10, -3) 'EMI', MOD(FinalPrice, ROUND(FinalPrice/10, -3) * 10) 'Remaining Amount' FROM INVENTORY;
```

यह elegant query ROUND() का उपयोग negative precision के साथ नकिटतम 1000 तक गोल करने के लिए करती है, और MOD() गोल EMIs द्वारा cover न होने वाली शेष राशि ज्ञात करता है।

खंड F: मिलान प्रश्न

Column A के प्रत्येक item को Column B के सही item से मिलाइए।

शिक्षक नोट: मिलान प्रश्न function names और उनके उद्देश्यों की exact knowledge की परीक्षा लेते हैं। सूची को systematically देखें, अनुमान लगाने की बजाय।

मिलान तालिका 1: SQL Functions और उनका उद्देश्य

Column A (Function)	Column B (उद्देश्य/Output)
1. POWER(2, 5)	A. 'November' लौटाता है
2. ROUND(3.567, 2)	B. 'HELLO' लौटाता है
3. MOD(25, 7)	C. 32 लौटाता है
4. UCASE('hello')	D. 3.57 लौटाता है
5. MONTHNAME('2023-11-10')	E. 4 लौटाता है
6. LENGTH('SQL')	F. 3 लौटाता है
7. LEFT('Science', 3)	G. वर्तमान दिनांक और समय लौटाता है
8. NOW()	H. 'Sci' लौटाता है

उत्तर: 1-C, 2-D, 3-E, 4-B, 5-A, 6-F, 7-H, 8-G

मिलान तालिका 2: SQL Operations और वविरण

Column A (Operation/Clause)	Column B (वविरण)
1. UNION	A. GROUP BY के बाद समूहों को filter करता है
2. INTERSECT	B. पहली तालिका की वे पंक्तियाँ जो दूसरी में नहीं
3. MINUS	C. दोनों तालिकाओं की सभी unique पंक्तियाँ मिलाता है
4. HAVING	D. दोनों ओर के spaces हटाता है
5. NATURAL JOIN	E. केवल दोनों तालिकाओं में सामान्य पंक्तियाँ
6. GROUP BY	F. result में duplicate common column हटाता है
7. TRIM()	G. समान column values वाली पंक्तियों को समूहीकृत करता है
8. DISTINCT	H. result से duplicate पंक्तियाँ हटाता है

उत्तर: 1-C, 2-E, 3-B, 4-A, 5-F, 6-G, 7-D, 8-H

मिलान तालिका 3: Function Category

Column A (Function Name)	Column B (Category)
1. NOW()	A. Aggregate Function

2. AVG()	B. Date Function
3. INSTR()	C. Numeric Function
4. ROUND()	D. String Function
5. DAYNAME()	E. Date Function
6. SUM()	F. Aggregate Function
7. LTRIM()	G. String Function
8. MOD()	H. Numeric Function

✓ उत्तर: 1-B, 2-A, 3-D, 4-C, 5-E, 6-F, 7-G, 8-H

त्वरति संदर्भ: सभी SQL Functions एक नजर में

परीक्षा से पहले इस तालिका को 2-3 बार देखें!

📌 शिक्षक नोट: इस page को revision के दौरान संभालकर रखें। यह तालिका परीक्षा में आत्मवश्वास को बढ़ाएगी!

Function	श्रेणी	Syntax	उदाहरण और Output
POWER	Numeric	POWER(x,y)	POWER(2,8) → 256
ROUND	Numeric	ROUND(n,d)	ROUND(4.567,2) → 4.57
MOD	Numeric	MOD(m,n)	MOD(17,5) → 2
UCASE/UPPER	String	UCASE(str)	UCASE('hi') → 'HI'
LCASE/LOWER	String	LCASE(str)	LCASE('HI') → 'hi'
MID/SUBSTR	String	MID(str,pos,n)	MID('Info',2,3) → 'nfo'
LENGTH	String	LENGTH(str)	LENGTH('SQL') → 3
LEFT	String	LEFT(str,n)	LEFT('Comp',4) → 'Comp'
RIGHT	String	RIGHT(str,n)	RIGHT('Sci',3) → 'Sci'
INSTR	String	INSTR(str,sub)	INSTR('Info','fo') → 3
LTRIM	String	LTRIM(str)	बाईं ओर के spaces हटाता है
RTRIM	String	RTRIM(str)	दाईं ओर के spaces हटाता है
TRIM	String	TRIM(str)	दोनों ओर के spaces हटाता है
NOW	Date/Time	NOW()	वर्तमान दिनांक और समय
DATE	Date/Time	DATE(expr)	Date भाग निकालता है
MONTH	Date/Time	MONTH(date)	MONTH('2023-08-15') → 8
MONTHNAME	Date/Time	MONTHNAME(date)	→ 'August'
YEAR	Date/Time	YEAR(date)	YEAR('2023-08-15') → 2023
DAY	Date/Time	DAY(date)	DAY('2023-08-15') → 15
DAYNAME	Date/Time	DAYNAME(date)	→ 'Tuesday'
MAX	Aggregate	MAX(col)	सबसे बड़ा मान
MIN	Aggregate	MIN(col)	सबसे छोटा मान
AVG	Aggregate	AVG(col)	औसत मान
SUM	Aggregate	SUM(col)	कुल योग
COUNT	Aggregate	COUNT(*)/COUNT(col)	पंक्तिगणना

परीक्षाओं के लिए शुभकामनाएँ! याद रखें — SQL concepts को समझना और नियमित रूप से computer पर queries का अभ्यास करना सबसे अच्छी तैयारी है। आज आप जो भी function सीखते हैं, वह कल परीक्षा में स्वाभाविक रूप से आएगा।