

CUET · COMPUTER SCIENCE · CLASS XII · CODE 308

Querying and SQL Functions

CUET unit: Querying and SQL Functions

By UniDrill · NCERT-grounded study material

WWW.UNIDRILL.IN

UniDrill

Snapshot

- Advanced SQL querying techniques extend Class XI database knowledge to manipulate and retrieve data with precision.
- Single Row (Scalar) functions fall into three categories — Numeric, String, and Date/Time — in contrast to Aggregate (Multiple Row) functions.
- GROUP BY and HAVING clauses are introduced to aggregate and filter grouped data, a favourite testing area for NTA.
- Operations on Relations (Union, Intersection, Minus, Cartesian Product) form the relational algebra backbone tested directly in CUET objective questions.
- The JOIN operations — using WHERE condition, explicit JOIN ... ON, and NATURAL JOIN — are consistently tested through output-prediction style MCQs.

Detailed Notes

2.1 Core concepts

- **Database context — CARSHOWROOM:** A four-table database (INVENTORY, CUSTOMER, SALE, EMPLOYEE) serves as the running example. INVENTORY stores CarId, CarName, Price, Model, YearManufacture, FuelType; CUSTOMER stores CustID, CustName, CustAdd, Phone, Email; SALE stores InvoiceNo, CarID, CustID, SaleDate, PaymentMode, EmpID, SalePrice; EMPLOYEE stores EmpID, EmpName, DOB, DOJ, Designation, Salary. (NCERT §1.1, p. 1–3)
- **SQL Functions overview:** A function performs a particular task and returns zero or more values. Functions in SQL can work on single or multiple records. Based on their application, SQL functions are categorised as Single Row functions (also called Scalar functions) and Aggregate functions (also called Multiple Row functions). (NCERT §1.2, p. 4)
- **Single Row Functions — three categories:** Single row functions are applied on a single value and return a single value. They are divided into three groups: (i) Numeric (Math) functions — POWER(), ROUND(), MOD(); (ii) String functions — UCASE()/UPPER(), LCASE()/LOWER(), MID()/SUBSTRING()/SUBSTR(), LENGTH(), LEFT(), RIGHT(), INSTR(), LTRIM(), RTRIM(), TRIM(); (iii) Date and Time functions — NOW(), DATE(), MONTH(), MONTHNAME(), YEAR(), DAY(), DAYNAME(). (NCERT §1.2.1, Fig. 1.2, p. 4)

- **Numeric functions:** POWER(X,Y) or POW(X,Y) computes X raised to Y; ROUND(N,D) rounds N to D decimal places (if D=0, rounds to nearest integer); MOD(A,B) returns the remainder after dividing A by B. (NCERT §1.2.1 (A), Table 1.5, p. 5)
- **String functions:** UCASE/UPPER converts to uppercase; LCASE/LOWER converts to lowercase; MID(string, pos, n) returns substring of length n starting from pos (if n is omitted, returns from pos to end); LENGTH returns character count; LEFT(string, N) returns N characters from the left; RIGHT(string, N) returns N characters from the right; INSTR(string, substring) returns position of first occurrence of substring (returns 0 if not found); LTRIM removes leading spaces; RTRIM removes trailing spaces; TRIM removes both leading and trailing spaces. (NCERT §1.2.1 (B), Table 1.6, p. 8–9)
- **Date and Time functions:** NOW() returns current system date and time; DATE() extracts the date part from a date/time expression; MONTH(date) returns month in numeric form; MONTHNAME(date) returns month name; YEAR(date) returns the year; DAY(date) returns the day number; DAYNAME(date) returns the name of the day. (NCERT §1.2.1 (C), Table 1.7, p. 11)
- **Aggregate (Multiple Row) functions:** These work on a set of records as a whole and return a single value for each column. Key functions: MAX(column) — largest value; MIN(column) — smallest value; AVG(column) — average value; SUM(column) — sum of values; COUNT(column) — count of non-NULL values; COUNT() — total number of records including NULL. COUNT(column) ignores NULL values whereas COUNT() counts all rows. (NCERT §1.2.2, Table 1.9, p. 12–13)
- **Differences between Single Row and Multiple Row functions:** Single row functions operate on one row at a time, return one result per row, and can be used in SELECT, WHERE, and ORDER BY clauses. Multiple row functions operate on groups of rows, return one result per group, and can be used in the SELECT clause only. (NCERT §1.2.2, Table 1.8, p. 12)
- **GROUP BY clause:** GROUP BY groups rows that contain the same values in a specified column. Aggregate functions (COUNT, MAX, MIN, AVG, SUM) can be applied to the grouped values. The HAVING clause is used to specify conditions on grouped rows; it works with GROUP BY the way WHERE works with individual rows. (NCERT §1.3, p. 14–15)
- **Operations on Relations:** These are binary operations applied on two relations that must have the same number of attributes, with corresponding attributes having the same domain. Three operations: UNION (U) — combines selected rows of two tables, eliminates duplicates; INTERSECT (∩) — returns common tuples from two tables; MINUS (-) — returns tuples in the first table that are not in the second table. (NCERT §1.4, p. 16–18)
- **Cartesian Product (X):** Combines tuples from two relations resulting in all pairs of rows, regardless of common attribute values. The degree of the result = sum of

degrees of both relations. The cardinality = product of cardinalities of both relations. (NCERT §1.4.4, p. 18–19)

- **Using Two Relations in a Query:** When more than one table is used, table names are separated by commas in the FROM clause; the DBMS first applies Cartesian product internally. To avoid ambiguity when an attribute name appears in both tables, table qualifiers (aliases) are used. (NCERT §1.5, §1.5.1, p. 19–20)
- **JOIN on two tables:** JOIN combines tuples from two tables based on specified conditions. It can be written three ways: (a) using WHERE condition on common attribute; (b) explicit JOIN ... ON clause in FROM; (c) NATURAL JOIN — works like JOIN but automatically removes the redundant common attribute from the result. Table aliases are valid only within the current query; once an alias is assigned in FROM, the original table name cannot be used in that query. (NCERT §1.5.2, p. 20–22)

2.2 Definitions to memorise

Term	Definition	Page
Single Row Function	A function applied on a single value that returns a single value; also called Scalar function	4
Aggregate Function	A function that works on a set of records as a whole and returns a single value; also called Multiple Row function	12
POWER(X,Y)	Returns X raised to the power Y	5
ROUND(N,D)	Rounds number N to D decimal places; if D=0, rounds to nearest integer	5
MOD(A,B)	Returns the remainder after dividing A by B	5
MID(string, pos, n)	Returns a substring of length n starting from position pos; also written as SUBSTRING() or SUBSTR()	8
INSTR(string, substr)	Returns the position of the first occurrence of substr in string; returns 0 if not found	8
LTRIM / RTRIM / TRIM	Remove leading / trailing / both leading and trailing white space characters respectively	8–9
NOW()	Returns current system date and time	11
DAYNAME(date)	Returns the name of the day from the given date	11
COUNT(column)	Returns count of non-NULL values in the specified column	13
COUNT(*)	Returns total number of records in a table including NULL values	13
GROUP BY	Clause that groups rows with same values in a specified column for use with aggregate functions	14
HAVING	Clause used to specify conditions on grouped rows (works with GROUP BY)	14

Term	Definition	Page
UNION	Set operation combining rows of two tables, showing duplicate rows only once	16
INTERSECT	Set operation returning only the common tuples from two tables	17
MINUS	Set operation returning tuples in the first table that are not in the second	18
Cartesian Product	Binary operation combining all pairs of rows from two tables; degree = sum of degrees; cardinality = product of cardinalities	18
NATURAL JOIN	Extension of JOIN that removes the redundant common attribute from the result	21
Table Alias	A shorthand name for a table, valid only for the current query; once set in FROM, original name cannot be used	20
UCASE / UPPER	String function returning the uppercase form of a string	8
LCASE / LOWER	String function returning the lowercase form of a string	8
LEFT(s, n)	Returns the leftmost n characters of string s	8
RIGHT(s, n)	Returns the rightmost n characters of string s	8
LENGTH	Returns the number of characters in a string	8
INSTR	Returns the 1-based position of first occurrence of substring (0 if absent)	8
SUM	Aggregate returning the sum of a numeric column	13
MAX, MIN, AVG	Aggregates returning maximum, minimum, and average of a column	12-13
JOIN ... ON	Explicit join syntax specifying the join condition in the FROM clause	20-21
Equi-Join	Join based on equality of values in a common attribute	21
Cardinality	Number of rows in a relation	18
Degree	Number of attributes in a relation	18

2.3 Diagrams / processes to remember

- **Figure 1.1 (p. 2):** Schema diagram of CARSHOWROOM database showing four relations — INVENTORY, CUSTOMER, SALE, EMPLOYEE — and their attributes. SALE acts as the central table linked to all others via foreign keys.
- **Figure 1.2 (p. 4):** Hierarchical classification of Single Row Functions into Numeric (POWER, ROUND, MOD), String (UCASE, LCASE, MID, LENGTH, LEFT, RIGHT, INSTR, LTRIM, RTRIM, TRIM), and Date (NOW, DATE, MONTH, MONTHNAME, YEAR, DAY, DAYNAME).

- **Table 1.8 (p. 12):** Comparison table of Single Row vs Multiple Row functions — four key differences covering operand scope, result count, clause usage, and examples.
- **Figure 1.3 (p. 16):** Venn diagram illustrating UNION of two sets (Music and Dance). Result shows all rows from both tables without repetition.
- **Figure 1.4 (p. 17):** Venn diagram illustrating INTERSECT — only the overlapping region (common students in DANCE and MUSIC).
- **Figure 1.5 (p. 18):** Venn diagram illustrating MINUS (set difference) — only tuples in MUSIC not found in DANCE.
- **Table 1.15 (p. 19):** Output of DANCE X MUSIC Cartesian Product — 20 rows (4x5), degree 6 (3+3). Students should be able to calculate degree and cardinality without seeing the table.

2.4 Common confusions / NTA trap points

- **COUNT(column) vs COUNT(*):** COUNT(column) ignores NULL values and returns only the count of non-NULL entries; COUNT(*) counts every row including those with NULL. NTA frequently presents a table with one NULL value and asks for the output — the answers differ by exactly 1.
- **HAVING vs WHERE with aggregate functions:** WHERE cannot be used with aggregate functions; HAVING is specifically designed to filter groups after GROUP BY. A query like `WHERE COUNT(*) > 1` is invalid; the correct form is `HAVING COUNT(*) > 1`.
- **Single Row functions in WHERE clause:** Unlike aggregate functions, single row functions CAN be used in WHERE and ORDER BY clauses, not just SELECT. NTA may present a query using UCASE() in a WHERE clause and ask if it is valid — it is.
- **ROUND(N, 0) vs ROUND(N):** Both round to the nearest integer, but ROUND(283, 2) returns 283 (not 283.00) — the function returns a numeric value; the display format depends on context. Students confuse ROUND(N,-1) which rounds to the nearest 10.
- **NATURAL JOIN removes the duplicate column:** Queries (a) using WHERE condition and (b) using JOIN ... ON both show the common column (e.g., UCode) twice; NATURAL JOIN shows it only once. NTA tests this through output-prediction questions.
- **Cartesian Product vs JOIN (NCERT § 1.4.4 vs § 1.5.2, pp. 18-21).** Cartesian product produces all possible row combinations (cardinality = $m \times n$); JOIN applies a condition to filter meaningful combinations.
- **UNION eliminates duplicates (NCERT § 1.4.1, p. 16).** UNION ALL would keep them — but NCERT here covers plain UNION.
- **Set operations need compatible attributes (NCERT § 1.4, p. 16).** Same number of columns with matching domains. Otherwise SQL returns an error.

- **MOD(A,B) returns A mod B (NCERT Table 1.5, p. 5).** Order matters — MOD(B,A) gives a different answer.
- **NATURAL JOIN is automatic on common columns (NCERT § 1.5.2, p. 21).** No need to specify the column; SQL matches by name.
- **Table alias scope is per query (NCERT § 1.5.1, p. 20).** Aliases vanish after the query ends.
- **HAVING needs GROUP BY (NCERT § 1.3, p. 14).** Using HAVING without GROUP BY treats the whole result as one group.

Practice MCQs

Q1. Which of the following SQL functions returns the remainder when 25 is divided by 4?

- A. POWER(25, 4)
- B. ROUND(25, 4)
- C. MOD(25, 4)
- D. MOD(4, 25)

Q2. Consider the following SQL query: `SELECT ROUND(4768.6543, -1);` What will be the output?

- A. 4768.7
- B. 4769.0
- C. 4770.0
- D. 4768.6

Q3. Which of the following statements correctly distinguishes Single Row functions from Aggregate functions?

- A.** Single row functions return one result for the entire table; aggregate functions return one result per row.
- B.** Single row functions can be used only in the SELECT clause; aggregate functions can be used in SELECT, WHERE, and ORDER BY.
- C.** Single row functions operate on one row at a time and return one result per row; aggregate functions operate on a group of rows and return one result per group.
- D.** Single row functions include MAX(), MIN(), and SUM(); aggregate functions include POWER() and ROUND().

 **12 more MCQs + answer key**

Get UniDrill Pro · ₹199/year · unidrill.in/pricing

PYQ Alignment

Advanced SQL is among the most heavily tested areas in CUET Computer Science / Informatics Practices papers; SQL function syntax, output prediction from aggregate queries with GROUP BY/HAVING, and set operation results (UNION, INTERSECT, MINUS) each appear regularly, with JOIN-based output prediction emerging as a newer question type in the 2024–25 pattern. See [PYQ archive for Computer Science](#).