

CUET · COMPUTER SCIENCE · CLASS XII · CODE 308

Structured Query Language (SQL)

CUET unit: Structured Query Language (SQL)

By UniDrill · NCERT-grounded study material

WWW.UNIDRILL.IN

UniDrill

Snapshot

- SQL is the standard query language used by major RDBMS software such as MySQL, Oracle, and SQL Server to create, populate, and query relational databases.
- SQL has two main sublanguages: DDL (Data Definition Language) for schema management and DML (Data Manipulation Language) for data manipulation.
- Core data types (CHAR, VARCHAR, INT, FLOAT, DATE), constraints (NOT NULL, UNIQUE, DEFAULT, PRIMARY KEY, FOREIGN KEY), and their correct assignment to table attributes are extensively tested.
- CUET regularly tests SELECT query clauses (WHERE, ORDER BY, GROUP BY, HAVING, DISTINCT, LIKE), aggregate functions (COUNT, MAX, MIN, AVG, SUM), and single-row functions (Math, String, Date).
- Operations on relations (UNION, INTERSECT, MINUS, Cartesian Product) and multi-table queries using JOIN and NATURAL JOIN are higher-difficulty topics that appear as case-based MCQs.

Detailed Notes

2.1 Core concepts

- **SQL and MySQL:** SQL (Structured Query Language) is the most popular query language used by relational DBMS. MySQL is a free, open-source RDBMS available from dev.mysql.com/downloads. SQL is case-insensitive, and each statement must end with a semicolon (;). (NCERT §9.2, p. 131–132)
- **Installing and using MySQL:** After installation, the `mysql>` prompt signals readiness. Multi-line statements use `->` continuation. Column names like 'salary' and 'SALARY' are treated identically by SQL. (NCERT §9.2.1, p. 132)
- **Data types in MySQL:** The five commonly used data types are CHAR(n) — fixed-length string up to 255 characters; VARCHAR(n) — variable-length string up to 65535 characters; INT — 4-byte integer (unsigned range 0 to 4,294,967,295); FLOAT — decimal numbers, 4 bytes; DATE — stored as 'YYYY-MM-DD', range '1000-01-01' to '9999-12-31'. (NCERT §9.3.1, Table 9.1, p. 133)
- **Constraints:** Constraints restrict data values in an attribute. NOT NULL prevents missing values; UNIQUE ensures all values in a column are distinct; DEFAULT assigns

- a fallback value; PRIMARY KEY uniquely identifies each row; FOREIGN KEY references the primary key of another table. (NCERT §9.3.2, Table 9.2, p. 134)
- **CREATE DATABASE and USE:** CREATE DATABASE databasename; creates a new database. USE databasename; selects it for subsequent operations. SHOW DATABASES; lists all databases; SHOW TABLES; lists tables in the current database. (NCERT §9.4.1, p. 134–135)
 - **CREATE TABLE:** Defines relation schema with attribute names, data types, and constraints. The number of columns defines the degree (N) of the relation. By default every attribute accepts NULL except the primary key. Syntax: CREATE TABLE tablename(attr1 datatype constraint, ...); (NCERT §9.4.2, p. 135)
 - **DESCRIBE / DESC:** DESCRIBE tablename; or DESC tablename; displays the structure (field name, type, nullability, key, default) of a created table. (NCERT §9.4.3, p. 137)
 - **ALTER TABLE:** Used post-creation to (A) add primary key, (B) add foreign key with REFERENCES clause, (C) add UNIQUE constraint, (D) add a new attribute (ADD), (E) modify datatype (MODIFY), (F) change constraint (MODIFY ... NOT NULL), (G) add DEFAULT value (MODIFY ... DEFAULT), (H) remove attribute (DROP column), (I) remove primary key (DROP PRIMARY KEY). (NCERT §9.4.4, p. 137–140)
 - **DROP statement:** DROP TABLE tablename; permanently removes a table. DROP DATABASE databasename; removes the database and all its tables. This action cannot be undone. (NCERT §9.4.5, p. 141)
 - **INSERT INTO (DML):** INSERT INTO tablename VALUES(v1, v2, ...); inserts a full row. To insert values for specific columns only: INSERT INTO tablename(col1, col2) VALUES(v1, v2); . Text and date values must be enclosed in single quotes. When a foreign key column has no value it must be explicitly set to NULL. (NCERT §9.5.1, p. 141–143)
 - **SELECT statement:** Core query syntax: SELECT attr1, attr2 FROM table_name WHERE condition; . SELECT * retrieves all columns. Column alias is specified with AS keyword. DISTINCT eliminates duplicate rows from query output. (NCERT §9.6.1, p. 144–147)
 - **WHERE clause operators:** Relational operators (=, <, <=, >, >=, !=); Logical operators AND, OR, NOT; BETWEEN...AND for range checks (inclusive of boundary values); IN operator checks membership in a value set; NOT IN excludes a set; IS NULL / IS NOT NULL checks for missing values; LIKE for pattern matching with wildcards % (zero or more characters) and _ (exactly one character). (NCERT §9.6.2 D–H, p. 147–153)
 - **ORDER BY clause:** Sorts query output. Default is ascending (ASC). Use DESC keyword for descending order. Multiple columns can be specified. (NCERT §9.6.2 F, p. 151)
 - **NULL handling:** NULL ≠ 0. Any arithmetic with NULL yields NULL (e.g., 5 + NULL = NULL). Use IS NULL or IS NOT NULL, never = NULL. (NCERT §9.6.2 G, p. 151–152)

- **UPDATE and DELETE (DML):** `UPDATE table SET col=val WHERE condition;` modifies existing records. `DELETE FROM table WHERE condition;` removes records. Omitting WHERE affects all rows — a common and dangerous mistake. (NCERT §9.7.1–9.7.2, p. 154–155)
- **Single Row (Scalar) functions:** Applied row-by-row; usable in SELECT, WHERE, and ORDER BY clauses. Three categories: Math (POWER, ROUND, MOD); String (UCASE/UPPER, LCASE/LOWER, MID/SUBSTRING/SUBSTR, LENGTH, LEFT, RIGHT, INSTR, LTRIM, RTRIM, TRIM); Date/Time (NOW, DATE, MONTH, MONTHNAME, YEAR, DAY, DAYNAME). (NCERT §9.8.1, Tables 9.13–9.15, p. 157–164)
- **Aggregate (Multiple Row) functions:** Operate on groups of rows and return a single value per group. Used only in SELECT clause. Functions: MAX, MIN, AVG, SUM, COUNT(*) (counts all rows), COUNT(column) (ignores NULLs). (NCERT §9.8.2, Table 9.17, p. 165–166)
- **GROUP BY and HAVING:** GROUP BY groups rows with identical values in a specified column; aggregate functions then apply per group. HAVING filters groups based on aggregate conditions — analogous to WHERE but for groups. (NCERT §9.9, p. 167–168)
- **Operations on Relations:** UNION (∪) — combines rows of two compatible tables, eliminating duplicates; INTERSECT (∩) — returns only common rows; MINUS (−) — rows in first table not in second; Cartesian Product (×) — all possible row combinations; degree of result = sum of degrees; cardinality = product of cardinalities. Both tables must have same number of attributes and compatible domains for UNION, INTERSECT, and MINUS. (NCERT §9.10, p. 168–171)
- **Multi-table queries and JOIN:** Cartesian product in FROM clause lists both tables separated by commas. JOIN combines tables on a condition (usually PK = FK). NATURAL JOIN eliminates the redundant common attribute. For N tables, N-1 joins are needed. (NCERT §9.11, p. 172–174)

2.2 Definitions to memorise

Term	Definition	Page
SQL	Structured Query Language; standard language used to access and manipulate data in RDBMS	131
DDL	Data Definition Language; SQL statements for defining/modifying/deleting schema (CREATE, ALTER, DROP)	134
DML	Data Manipulation Language; SQL statements for inserting, retrieving, updating, deleting data (INSERT, SELECT, UPDATE, DELETE)	141
CHAR(n)	Fixed-length character data type; reserves n bytes; pads with spaces if data shorter than n	133
VARCHAR(n)		133

Term	Definition	Page
	Variable-length character data type; allocates only as many bytes as needed; max n up to 65535	
PRIMARY KEY	Column (or combination) that uniquely identifies each row in a table; cannot be NULL	134
FOREIGN KEY	Column that references the PRIMARY KEY of another table; enforces referential integrity	134
NOT NULL	Constraint ensuring a column cannot hold NULL values	134
UNIQUE	Constraint ensuring all values in a column are distinct	134
DEFAULT	Constraint specifying a fallback value when none is provided during insert	134
NULL	Represents missing or unknown value; different from 0 or empty string	151
DISTINCT	Keyword in SELECT that eliminates duplicate rows from output	147
BETWEEN... AND	Operator checking if a value falls within an inclusive range	149
IN	Membership operator; returns true if value matches any value in a specified list	150
LIKE	Pattern matching operator; % matches zero or more characters; _ matches exactly one character	152
IS NULL	Operator used to test if a column value is NULL	152
ORDER BY	Clause to sort query results; default ascending; DESC for descending	151
GROUP BY	Clause to group rows with same column value; used with aggregate functions	167
HAVING	Clause to filter groups produced by GROUP BY based on aggregate conditions	167
Aggregate function	Function (MAX, MIN, AVG, SUM, COUNT) that operates on a set of rows and returns one value	165
Single row function	Function applied on each row individually; returns one result per row (e.g., ROUND, UCASE, MONTH)	157
UNION	Set operation combining all distinct rows from two compatible relations	169
INTERSECT	Set operation returning only rows common to both relations	170
MINUS	Set operation returning rows in first relation that are not in the second	170
Cartesian Product	Set operation combining every row of one table with every row of another; result degree = sum of degrees	171

Term	Definition	Page
NATURAL JOIN	JOIN variant that merges tables on common attribute(s) and eliminates the duplicate column	174
ALTER TABLE	DDL command that modifies the structure of an existing table	137
DROP TABLE	DDL command that permanently deletes a table	141
INSERT INTO	DML command that inserts records	141
UPDATE	DML command modifying existing records	154
DELETE	DML command removing rows but keeping structure	154
COUNT(*)	Aggregate counting all rows including NULLs	165
COUNT(col)	Aggregate counting non-NULL values in col	165
AVG	Aggregate returning average of numeric column	165
SUM	Aggregate returning sum of numeric column	165
ROUND	Single-row function rounding numeric values to given decimals	158
UCASE / UPPER	Single-row function converting string to uppercase	159
MID / SUBSTR	Single-row function returning a substring	160
NOW()	Date/time function returning current date and time	162
MONTHNAME	Date function returning name of the month	163
Equi-join	Join based on equality of values in common attributes	173

2.3 Diagrams / processes to remember

- **Figure 9.1 (p. 132):** MySQL 5.7 Command Line Client showing the `mysql>` prompt — confirms MySQL is ready to accept SQL statements after correct password entry.
- **Tables 9.3, 9.4, 9.5 (p. 136):** Data types and constraints for STUDENT (RollNumber INT PK, SName VARCHAR(20) NOT NULL, SDateofBirth DATE NOT NULL, GUID CHAR(12) FK), GUARDIAN (GUID CHAR(12) PK, GName VARCHAR(20) NOT NULL, GPhone CHAR(10) NULL UNIQUE, GAddress VARCHAR(30) NOT NULL), and ATTENDANCE (AttendanceDate DATE PK, RollNumber INT PK FK, AttendanceStatus CHAR(1) NOT NULL) — the running StudentAttendance database used in all examples.
- **Figure 9.2 (p. 156):** Schema diagram of CARSHOWROOM database showing four relations (INVENTORY, CUSTOMER, SALE, EMPLOYEE) and their relationships via foreign keys — used for all SQL function examples in §9.8.
- **Figure 9.3 (p. 158):** Classification tree of Single Row Functions — Numeric (POWER, ROUND, MOD), String (UCASE, LCASE, MID, LENGTH, LEFT, RIGHT, INSERT, LTRIM, RTRIM, TRIM), Date (NOW, DATE, MONTH, MONTHNAME, YEAR, DAY, DAYNAME).

- **Table 9.16 (p. 165):** Comparison table of Single Row vs Multiple Row (Aggregate) functions — key differences: single row works one row at a time; aggregate works on groups; aggregate usable only in SELECT clause.
- **Figures 9.4, 9.5, 9.6 (p. 169–171):** Venn diagrams illustrating UNION, INTERSECT, and MINUS operations on relations DANCE and MUSIC — visual anchor for set-operation MCQs.

2.4 Common confusions / NTA trap points

- **CHAR vs VARCHAR:** CHAR pads unused characters with spaces (fixed allocation); VARCHAR allocates only what is needed (variable). Students confuse which one to use for phone numbers (CHAR, since fixed length and no arithmetic) vs names (VARCHAR, variable length). NTA may ask which is more storage-efficient for variable-length data. (NCERT §9.3.1, p. 133)
- **DELETE vs DROP:** DELETE (DML) removes rows from a table based on a WHERE condition — the table structure remains. DROP (DDL) permanently removes the entire table or database. NTA often presents both as options for "removing all records" — DELETE FROM table; without WHERE deletes all rows but keeps the table; DROP TABLE removes the table entirely. (NCERT §9.4.5 and §9.7.2, p. 141 and 155)
- **WHERE vs HAVING:** WHERE filters individual rows before grouping; HAVING filters groups after GROUP BY. HAVING cannot be used without GROUP BY for filtering groups. NTA places HAVING in wrong positions as distractors. (NCERT §9.9, p. 167)
- **COUNT(*) vs COUNT(column):** COUNT(*) counts all rows including those with NULLs; COUNT(column) counts only non-NULL values in that column. NTA gives a table with NULL entries and asks for the count — choosing the wrong form leads to a different answer. (NCERT §9.8.2, Table 9.17, p. 165–166)
- **NULL arithmetic and comparison (NCERT §9.6.2 G, p. 151-152).** Any arithmetic involving NULL gives NULL. You cannot use = NULL; you must use IS NULL.
- **BETWEEN is inclusive (NCERT §9.6.2 D, p. 149).** WHERE x BETWEEN 10 AND 20 includes 10 and 20.
- **ORDER BY default ASC (NCERT §9.6.2 F, p. 151).** DESC must be explicit.
- **GROUP BY with non-aggregated columns (NCERT §9.9, p. 167).** Every non-aggregate column in SELECT must appear in GROUP BY.
- **UNION removes duplicates (NCERT §9.10.1, p. 169).** Use UNION ALL to preserve them — NCERT discusses UNION.
- **Natural Join eliminates duplicate common column (NCERT §9.11.2, p. 174).** Equi-join preserves both copies.
- **DESCRIBE doesn't change data (NCERT §9.4.3, p. 137).** It only displays the schema.

Practice MCQs

Q1. Which of the following data types in MySQL stores a variable-length string with a maximum length of 65535 characters and allocates storage only for the actual length of the entered string?

- A. CHAR(n)
- B. VARCHAR(n)
- C. INT
- D. DATE

Q2. Consider the following SQL constraints listed in Column A and their descriptions in Column B. Match them correctly. | Column A (Constraint) | Column B (Description) | |---|---| | (i) NOT NULL | (P) Column value must be unique across all rows but can be NULL | | (ii) UNIQUE | (Q) Column cannot hold missing or unknown values | | (iii) DEFAULT | (R) A fallback value is used when no value is provided at insert time | | (iv) FOREIGN KEY | (S) Column refers to the primary key of another table |

- A. (i)–Q, (ii)–P, (iii)–R, (iv)–S
- B. (i)–P, (ii)–Q, (iii)–S, (iv)–R
- C. (i)–Q, (ii)–S, (iii)–R, (iv)–P
- D. (i)–R, (ii)–P, (iii)–Q, (iv)–S

Q3. A student writes the following SQL statement to find all employees whose bonus is not recorded in the database: `SELECT * FROM EMPLOYEE WHERE Bonus = NULL;` Which of the following is correct about this statement?

- A. The statement is correct and will return all rows where Bonus is NULL.
- B. The statement will return an error because NULL cannot be used with the = operator.
- C. The statement will return an empty result set because NULL cannot be tested using =; IS NULL must be used instead.
- D. The statement is correct but will also include rows where Bonus is 0.

 **12 more MCQs + answer key**

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



PYQ Alignment

Chapter 9 is one of the highest-weightage chapters for CUET Computer Science, typically contributing 6–10 questions per year. Past papers have focused on SELECT query clauses (especially WHERE conditions with BETWEEN, IN, LIKE, IS NULL), aggregate functions (distinguishing COUNT(*) from COUNT(column)), ALTER TABLE operations, and the effect of missing WHERE in UPDATE/DELETE statements; case-based questions from the CARSHOWROOM or similar database schemas are a recurring pattern. See [PYQ archive for Computer Science](#).

