

CUET · COMPUTER SCIENCE · CLASS XII · CODE 308

Database Concepts

CUET unit: Database Concepts

By UniDrill · NCERT-grounded study material

WWW.UNIDRILL.IN

UniDrill

Snapshot

- Databases build on foundational concepts — moving from manual record keeping and flat file systems to a full-fledged Database Management System (DBMS).
- File systems fall short (redundancy, inconsistency, isolation, dependence, poor access control); a DBMS addresses each limitation.
- The Relational Data Model is presented as the dominant paradigm, with tables (relations), attributes, tuples, and the five key terminologies (attribute, tuple, domain, degree, cardinality).
- Three important structural properties of a relation distinguish it from a simple table or file — unique attribute names, distinct tuples, and atomic data values.
- Keys (Candidate, Primary, Composite Primary, Foreign) are the most CUET-tested topic because they underpin data integrity and inter-table relationships.

Detailed Notes

2.1 Core concepts

- **§ 8.1 Introduction — Limitations of manual record keeping:** A school example with 50 students and 26 working days illustrates four manual limitations: (1) re-entry of student details each year, (2) inconsistency from incorrect or skipped entries, (3) data loss if the register is damaged, (4) erroneous calculations during consolidation. Computerised storage overcomes these by allowing copy, search, add, modify and delete operations on data files. (NCERT §8.1, p. 111–112)
- **§ 8.2 File System — definition and structure:** A file is a container to store data on a computer's storage device. Contents can be text, program code, CSV values, pictures, audio/video or web pages. Two files serve as running examples — STUDENT (RollNumber, SName, SDateofBirth, GName, GPhone, GAddress) and ATTENDANCE (AttendanceDate, RollNumber, SName, AttendanceStatus). (NCERT §8.2, p. 112–113)
- **§ 8.2.1 Limitations of a File System — six named problems:**
- **(A) Difficulty in Access:** Files provide no built-in retrieval mechanism; custom application programs must be written, and the developer may not anticipate all required access patterns. (NCERT §8.2.1, p. 114)

- **(B) Data Redundancy:** Same data duplicated across files (e.g., SName in both STUDENT and ATTENDANCE; guardian name repeated for siblings). Redundancy wastes storage and can cause inconsistency. (NCERT §8.2.1, p. 114)
- **(C) Data Inconsistency:** When redundant copies are updated in one file but not another, the data becomes inconsistent. For example, a student's name change must be reflected in both files or inconsistency arises. (NCERT §8.2.1, p. 114)
- **(D) Data Isolation:** Related files (STUDENT and ATTENDANCE) have no built-in link. Separate programs must be written for each file, and understanding every file's structure is necessary for cross-file queries. (NCERT §8.2.1, p. 114–115)
- **(E) Data Dependence:** If the storage format or structure of a file changes, all application programs that access it must also be modified — this is data dependency. (NCERT §8.2.1, p. 115)
- **(F) Controlled Data Sharing:** Enforcing differential access (e.g., guardians can only view attendance, not modify it; only the teacher can update it) is very difficult in a file system. (NCERT §8.2.1, p. 115)
- **§ 8.3 Database Management System (DBMS):** A DBMS is software that creates and manages databases. It lets users create a database, store, manage, update/modify and retrieve data. Examples: MySQL, Oracle, PostgreSQL, SQL Server, Microsoft Access, MongoDB. DBMS hides storage details, providing an abstract view; it serves as an interface between the database and end-users or application programs. Retrieving data through special commands is called **querying the database**. (NCERT §8.3, p. 115–116)
- **§ 8.3.1 File System to DBMS — converting the school example:** The STUDENT file is split into two tables: STUDENT (RollNumber, SName, SDateofBirth, GUID) and GUARDIAN (GUID, GName, GPhone, GAddress). SName is removed from ATTENDANCE. A new unique identifier GUID (Guardian ID) is introduced because guardian names can repeat and phone numbers can change. The resulting three-table database is called STUDENTATTENDANCE (Tables 8.4, 8.5, 8.6). The cost of shifting to DBMS includes purchasing hardware/software, training users, and recurrent backup costs. (NCERT §8.3.1, p. 116–118)
- **§ 8.3.2 Key Concepts in DBMS — seven terms:**
 - **(A) Database Schema:** Design/skeleton of the database showing table names, columns, data types, constraints and relationships. Also called the visual or logical architecture. (NCERT §8.3.2, p. 118)
 - **(B) Data Constraint:** Restrictions on the type of data that can be inserted into columns (e.g., NOT NULL, UNIQUE on RollNumber). Constraints ensure accuracy and reliability. (NCERT §8.3.2, p. 119)
 - **(C) Meta-data / Data Dictionary:** The schema and constraints stored by DBMS in a database catalog or dictionary. Meta-data is data about the data. (NCERT §8.3.2, p. 119)

- **(D) Database Instance:** The snapshot/state of the database at any given time after data is loaded; it can change through insertions, updates, or deletions. (NCERT §8.3.2, p. 119)
- **(E) Query:** A request to the database for information retrieval or data manipulation (insertion, deletion, update). Written using a query language. (NCERT §8.3.2, p. 120)
- **(F) Data Manipulation:** Three operations — Insertion (new student joins), Deletion (student leaves school; data removed from STUDENT, GUARDIAN, ATTENDANCE), Update (guardian's phone number changes in GUARDIAN table). (NCERT §8.3.2, p. 120)
- **(G) Database Engine:** The underlying component/set of programs used by a DBMS to create databases and handle queries for retrieval and manipulation. (NCERT §8.3.2, p. 120)
- **§ 8.4 Relational Data Model:** The most commonly used data model. Other models include object-oriented, entity-relationship, document, and hierarchical. In relational model, tables are called **relations** that store data for different columns. Relations are not independent — they are associated through linking attributes (e.g., RollNumber links ATTENDANCE to STUDENT; GUID links STUDENT to GUARDIAN). A database modelled on relational data model is called a **Relational Database**. (NCERT §8.4, p. 120–121)
- **§ 8.4 Relational terminologies (Figure 8.4 — GUARDIAN relation):**
 - **Attribute:** Column of a relation; also called a field. Example: GUID, GName, GPhone, GAddress are attributes of GUARDIAN. (NCERT §8.4, p. 122)
 - **Tuple:** Each row of data in a relation. In a table with n columns, a tuple is a relationship between n values. (NCERT §8.4, p. 122)
 - **Domain:** The set of values an attribute can take in each row; usually specified by a data type. Example: RollNumber's domain is the set of integers. (NCERT §8.4, p. 122)
 - **Degree:** Number of attributes in a relation. GUARDIAN has degree 4. (NCERT §8.4, p. 122)
 - **Cardinality:** Number of tuples (rows/records) in a relation. GUARDIAN has cardinality 5. (NCERT §8.4, p. 123)
- **§ 8.4.1 Three Important Properties of a Relation:**
 - **Property 1 (Attribute rules):** Each attribute has a unique name; sequence of attributes is immaterial. (NCERT §8.4.1, p. 123)
 - **Property 2 (Tuple rules):** Each tuple is distinct (no two tuples identical for all attributes); sequence of tuples is immaterial. (NCERT §8.4.1, p. 123)
 - **Property 3 (State rules):** All values in an attribute must be from the same domain (same data type); each value must be atomic (indivisible); no attribute can have multiple values in one tuple; NULL represents unknown or non-applicable values. (NCERT §8.4.1, p. 123)

- **§ 8.5 Keys in a Relational Database:**

- **§ 8.5.1 Candidate Key:** One or more attributes that take distinct (unique, not NULL) values and can uniquely identify tuples. Both GUID and GPhone are candidate keys of GUARDIAN. (NCERT §8.5.1, p. 124)
- **§ 8.5.2 Primary Key:** The candidate key chosen by the database designer to uniquely identify tuples. The remaining candidate keys become **Alternate Keys**. Example: if GUID is primary key of GUARDIAN, GPhone is the alternate key. (NCERT §8.5.2, p. 124)
- **§ 8.5.3 Composite Primary Key:** When no single attribute uniquely identifies tuples, two or more attributes together form the primary key. Example: {RollNumber, AttendanceDate} is the composite primary key of ATTENDANCE, since each student has exactly one attendance record per day. (NCERT §8.5.3, p. 124–125)
- **§ 8.5.4 Foreign Key:** An attribute in a referencing (foreign) relation whose values are derived from the primary key of another (referenced/master) relation. Used to represent relationships between two relations. A foreign key can take NULL if it is not part of the primary key of the referencing relation. Example: GUID in STUDENT is a foreign key referencing GUID (primary key) of GUARDIAN; RollNumber in ATTENDANCE references RollNumber in STUDENT. Shown as directed arcs in the schema diagram (Figure 8.5). (NCERT §8.5.4, p. 125)

2.2 Definitions to memorise

Term	Definition	Page
Database	A collection of logically related tables (relations) managed as a single unit	115
DBMS	Software used to create, store, manage, update/modify and retrieve data from databases	115
Database Schema	Design/skeleton of the database showing structure, constraints, and relationships; also called visual/logical architecture	118
Meta-data	Data about the data; the schema and constraints stored in the database catalog/dictionary	119
Database Instance	Snapshot/state of the database at any given time	119
Query	A request to a database for information retrieval or data manipulation, written in a query language	120
Database Engine	Underlying component/programs used by a DBMS to create the database and handle queries	120
Relation	A table in the relational data model; a flat file where each column has a single atomic value per row	120
Attribute	A column of a relation; also called a field	122
Tuple	A row/record in a relation	122

Term	Definition	Page
Domain	Set of values an attribute can take (usually defined by data type)	122
Degree	Number of attributes in a relation	122
Cardinality	Number of tuples in a relation	123
Candidate Key	Attribute(s) with unique, non-NULL values that can uniquely identify tuples	124
Primary Key	The chosen candidate key used for unique identification of tuples in a relation	124
Alternate Key	Candidate key(s) not chosen as the primary key	124
Composite Primary Key	A primary key formed by combining two or more attributes	124
Foreign Key	Attribute in a referencing relation derived from the primary key of a referenced (master) relation	125
Data Redundancy	Same data duplicated in multiple files/tables	114
Data Inconsistency	Mismatch when redundant data is updated in one place but not another	114
NULL	Special value used to represent unknown or non-applicable attribute values	123
Data Constraint	Restriction on the values a column can hold	119
Data Manipulation	Insertion, deletion, update operations on data	120
Atomic value	An indivisible data value that cannot be broken into smaller meaningful parts	123
Relational Database	Database designed on the relational data model	121
Linking attribute	Attribute that establishes the relationship between two relations	121
Master/Primary Relation	The relation whose primary key is referenced by a foreign key elsewhere	125
Referencing Relation	The relation that contains the foreign key	125
File processing	Older approach to data management with redundancy and isolation issues	114
Hierarchical Model	Older data model storing data as tree-like structure	120
Object-Oriented Model	Data model representing data as objects with attributes and methods	120
Database Catalog	Storage area where DBMS keeps meta-data	119

2.3 Diagrams / processes to remember

- **Figure 8.1 (p. 117) — Record structure of three files in STUDENTATTENDANCE database:** Shows the empty schema of STUDENT (RollNumber, SName, SDateofBirth, GUID), GUARDIAN (GUID, GName, GPhone, GAddress), and ATTENDANCE (AttendanceDate, RollNumber, AttendanceStatus). This is the result of converting the file system to DBMS and is the basis for all key discussions.
- **Figure 8.2 (p. 119) — STUDENTATTENDANCE database environment:** Shows Teacher and Office Staff sending queries to DBMS Software, which accesses Student, Guardian, and Attendance tables alongside a Database Catalog, and returns query results. Memorise the flow: User → Query → DBMS → Database + Catalog → Query Result.
- **Figure 8.3 (p. 121) — Relational Data Model with three relations:** Shows GUARDIAN, STUDENT, and ATTENDANCE as boxes with their attributes, and lines indicating relationships via linking attributes (GUID between GUARDIAN and STUDENT; RollNumber between STUDENT and ATTENDANCE).
- **Figure 8.4 (p. 122) — GUARDIAN relation with labelled components:** Illustrates Relation State (all rows), Record/tuple/row, and the fact that GUARDIAN has Degree = 4 (columns) and Cardinality = 5 (rows). Essential for distinguishing degree vs. cardinality.
- **Figure 8.5 (p. 125) — STUDENTATTENDANCE database with primary and foreign keys:** Underlined attributes show primary keys; directed arcs (arrows) show foreign key references. STUDENT.RollNumber (PK) ← ATTENDANCE.RollNumber (FK); GUARDIAN.GUID (PK) ← STUDENT.GUID (FK).

2.4 Common confusions / NTA trap points

- **Degree vs. Cardinality:** Degree = number of columns (attributes); Cardinality = number of rows (tuples). NTA frequently swaps these in distractors. GUARDIAN has Degree 4 and Cardinality 5 — always anchor to this example.
- **Candidate Key vs. Primary Key vs. Alternate Key:** All primary keys are candidate keys, but not all candidate keys are primary keys. The unchosen candidate keys are alternate keys. NTA options often list "alternate key" as if it were independent of candidate keys.
- **Composite Primary Key is NOT two separate primary keys:** {RollNumber, AttendanceDate} together form one composite key in ATTENDANCE — neither attribute alone is a primary key. Students often confuse this with two separate primary keys.
- **Foreign Key can be NULL:** A foreign key can hold NULL if it is not part of the primary key of the referencing relation. NTA uses this as a trap — students incorrectly assume foreign keys must always have a value.

- **Data Inconsistency vs. Data Redundancy (NCERT §8.2.1, p. 114).** Redundancy is the cause; inconsistency is the possible effect when copies differ after update.
- **Schema vs Instance (NCERT §8.3.2, p. 118-119).** Schema is static design; Instance is the current state of data.
- **Primary key cannot be NULL (NCERT §8.5.2, p. 124).** It must be unique AND not null.
- **Sequence of tuples is immaterial (NCERT §8.4.1, p. 123).** A relation is a set, not an ordered list.
- **Insert-Update-Delete are DM operations (NCERT §8.3.2(F), p. 120).** Not DDL — they manipulate data, not structure.
- **MongoDB is a DBMS example (NCERT §8.3, p. 116).** It is a NoSQL document DBMS, listed alongside SQL-based systems.
- **Atomic value rule (NCERT §8.4.1 Property 3, p. 123).** "Mumbai-Pune" as one cell is non-atomic if both cities are separate entities; needs split into two attributes.

Practice MCQs

Q1. Which of the following is NOT listed as a limitation of a file system in the NCERT chapter?

- A. Data Redundancy
- B. Data Inconsistency
- C. Data Normalisation
- D. Data Dependence

Q2. In the STUDENTATTENDANCE database, the attribute GUID (Guardian ID) was introduced to replace GName as a linking field between STUDENT and GUARDIAN tables. What is the primary reason stated in the NCERT chapter for this change?

- A. GName values are too long to store efficiently
- B. Phone numbers can change and GName alone may not be unique across all guardians
- C. The ATTENDANCE table requires an additional foreign key
- D. GUID makes it easier to sort guardian records alphabetically

Q3. Consider the following statements about a relation in the Relational Data Model: **Statement I:** The sequence of tuples in a relation is immaterial. **Statement II:** Each data value associated with an attribute must be atomic (indivisible). Which of the above statements is/are correct?

- A. Only Statement I
- B. Only Statement II
- C. Both Statement I and Statement II
- D. Neither Statement I nor Statement II

 **12 more MCQs + answer key**

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

PYQ Alignment

Database Concepts is consistently tested in CUET Computer Science, with questions focusing on definitions of keys (especially distinguishing candidate, primary, alternate, composite, and foreign keys), properties of relations (degree vs. cardinality, atomic values, tuple distinctness), and limitations of file systems vs. advantages of DBMS. Practical application questions — such as identifying the correct primary key or composite key from a given table schema — are particularly common in recent CUET patterns. See [PYQ archive for Computer Science](#).