

FREE EDITION · NOTES + 3 SAMPLE MCQS

CUET · COMPUTER SCIENCE · CLASS XI · CODE 308

Database Concepts

CUET unit: Database Concepts

By UniDrill · NCERT-grounded study material

WWW.UNIDRILL.IN

UniDrill

Snapshot

- Manual and file-based data management have limitations; a Database Management System (DBMS) overcomes them and is preferable.
- The core components of a DBMS are: schema, constraints, meta-data, instance, query, data manipulation, and the database engine.
- The Relational Data Model uses formal terminology — relations, attributes, tuples, degree, cardinality — plus the three key properties of a relation.
- Keys in a relational database — Candidate Key, Primary Key, Composite Primary Key, and Foreign Key — are explained with concrete examples from the StudentAttendance database.
- Database concepts underpin all SQL and database design questions; understanding keys, relation properties, and file-system limitations is essential for both conceptual MCQs and applied questions, and CUET tests them heavily.

Detailed Notes

2.1 Core concepts

- **7.1 Introduction:** Manual record-keeping (e.g., school attendance register with 50 students × 26 working days) is tedious and error-prone. Limitations include data entry duplication when students are promoted, inconsistency due to incorrectly written names, data loss if the register is damaged, and erroneous manual calculations. Storing data in computerised files solves several of these problems.
(NCERT §7.1, p. 123-124)
- **7.2 File System:** A file is a container to store data in a computer; contents can be text, program code, CSV, pictures, audio/video, or web pages. Files are accessed through application programs rather than directly. Two example files illustrate this — STUDENT (RollNumber, SName, SDateofBirth, GName, GPhone, GAddress) and ATTENDANCE (AttendanceDate, RollNumber, SName, AttendanceStatus) — maintained separately by office staff and the class teacher respectively.
(NCERT §7.2, p. 124-125)
- **7.2.1 Limitations of a File System:** Six limitations are identified: (A) Difficulty in Access — files provide no retrieval mechanism; application programs must be written and may not anticipate all access patterns. (B) Data Redundancy — same

data duplicated in multiple files (e.g., SName in both STUDENT and ATTENDANCE; guardian name repeated for siblings). (C) Data Inconsistency — redundant copies may diverge when one copy is updated but another is not. (D) Data Isolation — no link or mapping between related files; separate programs must be written for each file. (E) Data Dependence — changing a file's structure or format requires modifying all application programs that access it. (F) Controlled Data Sharing — enforcing role-based access (e.g., guardians can only read attendance, not modify it) is very difficult in a file system. (NCERT §7.2.1, p. 126-127)

- **7.3 Database Management System (DBMS):** A DBMS is software used to create and manage databases. It lets users create, store, manage, update/modify, and retrieve data. Examples of open source and commercial DBMS: MySQL, Oracle, PostgreSQL, SQL Server, Microsoft Access, MongoDB. A DBMS hides physical storage details and provides an abstract view of data. It serves as an interface between the database and end users or application programs. Retrieving data through special commands is called querying; the query language is discussed in Chapter 8. (NCERT §7.3, p. 127-128)
- **7.3.1 File System to DBMS:** Converting the two-file example to a database requires: (a) removing SName from ATTENDANCE since it is already in STUDENT, linking via RollNumber; (b) splitting the STUDENT file into STUDENT and GUARDIAN to eliminate guardian-detail redundancy when siblings share a guardian; (c) introducing a GUID (Guardian ID) column in GUARDIAN as a unique identifier, and keeping GUID in STUDENT. The resulting schema has three tables: STUDENT (RollNumber, SName, SDateofBirth, GUID), GUARDIAN (GUID, GName, GPhone, GAddress), ATTENDANCE (AttendanceDate, RollNumber, AttendanceStatus). A DBMS maintains a single centralised repository usable by multiple users simultaneously. High costs of shifting to a DBMS include purchasing hardware/software, training users, and recurrent backup/recovery costs. (NCERT §7.3.1, p. 128-130)
- **7.3.2 Key Concepts in DBMS:** (A) Database Schema — the design (logical architecture) of a database; it describes table names, columns, data types, constraints, and relationships among tables; also stored as visual/logical architecture. (B) Data Constraint — a restriction on the type of data insertable into a column (e.g., mobile number must be a non-negative 10-digit integer; RollNumber must be NOT NULL and UNIQUE). Constraints ensure accuracy and reliability. (C) Meta-data / Data Dictionary — the schema plus constraints stored by the DBMS in a database catalog; it is data about the data. (D) Database Instance — the snapshot of the database at any given time; the schema can have many instances as data changes. (E) Query — a request to a database for obtaining information, made in a query language (SQL, covered in Chapter 8). (F) Data Manipulation — three operations: Insertion (adding new records), Deletion (removing records), Update (modifying existing records). (G) Database Engine — the underlying component/set

of programs used by a DBMS to create a database and handle queries. (NCERT §7.3.2, p. 130-132)

- **7.4 Relational Data Model:** A data model describes structure of the database including data definitions, representations, relationships, and constraints. The most commonly used is the Relational Data Model; others include object-oriented, entity-relationship, document, and hierarchical models. In the relational model, tables are called relations; each table can have multiple columns each with a unique column name; each row represents a related set of values. A database modelled on the relational data model is a Relational Database. (NCERT §7.4, p. 132-133)
- **Relational Terminology (§ 7.4, Figure 7.5):** (i) Attribute — characteristic/parameter for which data are stored; corresponds to a column/field. (ii) Tuple — each row of data in a relation; in a table with n columns, a tuple is a relationship between n related values. (iii) Domain — the set of values an attribute can take; usually specified by a data type (e.g., RollNumber has domain of integers; SName has domain of character strings). (iv) Degree — number of attributes in a relation (e.g., GUARDIAN with 4 columns has degree 4). (v) Cardinality — number of tuples in a relation (e.g., GUARDIAN with 5 rows has cardinality 5). (NCERT §7.4, p. 134-135)
- **7.4.1 Three Important Properties of a Relation:** Property 1 (attributes): each attribute has a unique name; sequence of attributes is immaterial. Property 2 (tuples): each tuple is distinct (no two identical tuples); sequence of tuples is immaterial. Property 3 (state): all values in an attribute must be from the same domain; each data value must be atomic (indivisible); no attribute can have multiple values in one tuple; NULL represents unknown or non-applicable values. (NCERT §7.4.1, p. 135)
- **7.5 Keys in a Relational Database:** No two tuples may have the same value for all attributes; at least one attribute must be unique and NOT NULL. (NCERT §7.5, p. 136)
- **7.5.1 Candidate Key:** One or more attributes that can uniquely identify every tuple. All such attributes are candidates for the primary key (e.g., in GUARDIAN both GUID and GPhone always take unique values, so both are candidate keys). (NCERT §7.5.1, p. 136)
- **7.5.2 Primary Key:** The candidate key chosen by the database designer to uniquely identify tuples. Remaining candidate keys become Alternate Keys (e.g., if GUID is chosen as primary key in GUARDIAN, then GPhone is the alternate key). (NCERT §7.5.2, p. 136)
- **7.5.3 Composite Primary Key:** When no single attribute uniquely identifies tuples, two or more attributes together form the primary key (e.g., in ATTENDANCE neither RollNumber alone nor AttendanceDate alone is unique, but {RollNumber, AttendanceDate} together are unique). (NCERT §7.5.3, p. 136-137)

- 7.5.4 Foreign Key:** An attribute whose value is derived from the primary key of another relation; used to represent relationships between two relations. The referencing relation is the Foreign Relation; the relation containing the referenced primary key is the Primary/Master Relation. A foreign key can take NULL if it is not part of the primary key of the foreign table (e.g., GUID in STUDENT references GUID in GUARDIAN; RollNumber in ATTENDANCE references RollNumber in STUDENT). In schema diagrams, foreign keys are shown as directed arcs (arrows) from the referencing attribute to the referenced primary key; primary key attributes are underlined. (NCERT §7.5.4, p. 137)

2.2 Definitions to memorise

Term	Definition	Page
File	A container to store data in a computer; contents can be text, code, CSV, images, audio/video, etc.	124
Data Redundancy	Same data duplicated in different places (files), leading to excess storage and possible inconsistency	126
Data Inconsistency	Situation where the same data maintained in different places do not match	126–127
Data Isolation	Lack of link/mapping between related files in a file system, requiring separate programs for each file	127
Data Dependence	When changing a file's structure requires modification of all application programs accessing that file	127
DBMS	Software used to create and manage databases; provides create, store, manage, update/modify, and retrieve operations	128
Database Schema	Design/logical architecture of a database; describes table names, columns, data types, constraints, and relationships	130–131
Data Constraint	Restriction on the type of data that can be inserted into a column of a table	131
Meta-data / Data Dictionary	Schema plus constraints stored in a database catalog; data about the data	131
Database Instance	Snapshot of the database (state of data) at a given point in time	131–132
Query	A request to a database for obtaining information in a desired way	132
Database Engine	Underlying component/set of programs used by a DBMS to create the database and handle queries	132
Relation	A table in the relational data model that stores data for different columns	133
Attribute	Characteristic/parameter (column/field) for which data are stored in a relation	134

Term	Definition	Page
Tuple	A row of data in a relation; a relationship between n related values in a table with n columns	134
Domain	Set of values from which an attribute can take a value; usually specified by a data type	134
Degree	Number of attributes (columns) in a relation	135
Cardinality	Number of tuples (rows) in a relation	135
NULL	Special value representing unknown or non-applicable data for an attribute	135
Candidate Key	Attribute(s) that can uniquely identify every tuple in a relation; all are candidates for primary key	136
Primary Key	The candidate key chosen to uniquely identify tuples in a relation	136
Alternate Key	Candidate key(s) not chosen as the primary key	136
Composite Primary Key	Primary key consisting of more than one attribute, used when no single attribute uniquely identifies tuples	136–137
Foreign Key	Attribute whose value is derived from the primary key of another relation; represents a relationship between two relations	137
Primary/Master Relation	The relation in which the referenced primary key is defined (the referenced table in a foreign key relationship)	137
Foreign Relation	The referencing relation that contains the foreign key	137
MySQL	Open-source RDBMS used in NCERT examples	128
MongoDB	NoSQL document-oriented database	128
Data Manipulation	Operations that change the contents of a database: Insert, Delete, Update	132
Insert	DML operation that adds new tuples to a relation	132
Delete	DML operation that removes tuples from a relation	132
Update	DML operation that modifies existing tuples	132
Atomic value	An indivisible data value that cannot be decomposed into meaningful sub-parts	135
Data Model	Description of the structure of a database including data definitions, representations, relationships and constraints	133
Relational Database	A database designed on the relational data model	133
Database Catalog	Storage area where metadata is held by a DBMS	131
Hierarchical Model	Older data model where data is structured as a tree	133
		133

Term	Definition	Page
Object-Oriented Model	Data model representing data as objects with attributes and methods	
Entity-Relationship Model	Conceptual data model representing entities and relationships	133

2.3 Diagrams / processes to remember

- **Figure 7.1 (p. 129):** Record structure of three files in STUDENTATTENDANCE Database — shows STUDENT (RollNumber, SName, SDateofBirth, GUID), GUARDIAN (GUID, GName, GPhone, GAddress), and ATTENDANCE (AttendanceDate, RollNumber, AttendanceStatus) as three boxes representing the file/table structure before relationships are drawn.
- **Figure 7.3 (p. 131):** StudentAttendance Database Environment — shows Teacher and Office Staff sending queries to the DBMS software layer, which accesses the Student, Guardian, and Attendance tables as well as the Database Catalog. Illustrates the DBMS acting as interface between users and the database.
- **Figure 7.4 / Figure 7.5 (p. 133–134):** Representing StudentAttendance Database using Relational Data Model — shows GUARDIAN, STUDENT, and ATTENDANCE as relation boxes with their attributes listed; Figure 7.5 specifically shows RELATION GUARDIAN populated with 5 tuples, annotated to show Degree = 4, Cardinality = 5, and identifies record/tuple/row and column/attribute.
- **Figure 7.2 / Schema Diagram (p. 137):** StudentAttendance Database with Primary and Foreign Keys — STUDENT(RollNumber underlined, SName, SDateofBirth, GUID) with arrow from GUID to GUARDIAN(GUID underlined, GName, GPhone, GAddress); ATTENDANCE(AttendanceDate + RollNumber underlined together, AttendanceStatus) with arrow from RollNumber to STUDENT. Underlined attributes = primary key components; arrows = foreign key references.

2.4 Common confusions / NTA trap points

- **Degree vs. Cardinality:** Students frequently swap these. Degree = number of attributes (columns); Cardinality = number of tuples (rows). NTA often tests both in the same question using a populated table.
- **Candidate Key vs. Primary Key vs. Alternate Key:** All candidate keys can uniquely identify tuples; the one chosen becomes the Primary Key; the rest become Alternate Keys. NTA distractors often label the alternate key as "secondary key" (which is not an NCERT term) or confuse it with foreign key.
- **Composite Primary Key vs. Foreign Key:** A composite primary key uses multiple attributes of the same table to ensure uniqueness; a foreign key references the primary key of another table. Students confuse them when a foreign key is also part of a composite key in the referencing table.

- **NULL in Foreign Keys:** Foreign keys can take NULL values if the foreign key attribute is not part of the primary key of the foreign (referencing) table. NTA may ask whether a NULL foreign key is valid — it is, under these conditions.
- **Data Inconsistency vs. Data Redundancy (NCERT § 7.2.1, pp. 126-127).** Redundancy is the cause (same data in multiple places); inconsistency is the consequence (copies diverge).
- **Tuples are unordered (NCERT § 7.4.1, p. 135).** NTA distractor: claims tuple order is significant.
- **Primary Key NOT NULL (NCERT § 7.5, p. 136).** Primary key attributes cannot have NULL. Foreign keys (under conditions) can.
- **No two identical tuples (NCERT § 7.4.1, p. 135).** Each tuple must be distinct — the basis of needing a candidate/primary key.
- **DBMS handles concurrent users (NCERT § 7.3, p. 128).** Multiple users can access simultaneously — a major advantage over file system.
- **Foreign Key ≠ Primary Key of same table (NCERT § 7.5.4, p. 137).** FK references the PK of another relation.
- **Schema vs Instance (NCERT § 7.3.2, p. 130-132).** Schema = design (static); Instance = state of data at a moment (dynamic).

Practice MCQs

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

- A. Data Redundancy
- B. Data Inconsistency
- C. Data Isolation
- D. Data Encryption

Q2. In the StudentAttendance database, the ATTENDANCE relation has the following schema: ATTENDANCE(AttendanceDate, RollNumber, AttendanceStatus). Which of the following correctly identifies the primary key of this relation?

- A. AttendanceDate alone
- B. RollNumber alone
- C. {AttendanceDate, RollNumber}
- D. AttendanceStatus

Q3. Consider the following statements about the properties of a relation in the Relational Data Model: **Statement I:** The sequence of tuples in a relation is significant and tuples are considered to be ordered. **Statement II:** Each data value associated with an attribute must be atomic (cannot be further divided into meaningful subparts). 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 one of the most consistently tested areas in CUET Computer Science / Informatics Practices, with questions typically focusing on identifying keys (primary, foreign, composite), distinguishing relational terminology (degree vs. cardinality, tuple vs. attribute), and scenario-based questions asking students to spot violations of relation properties or pick the correct primary key for a given table — all of which map directly to Sections 7.4 and 7.5. See [PYQ archive for Computer Science](#).