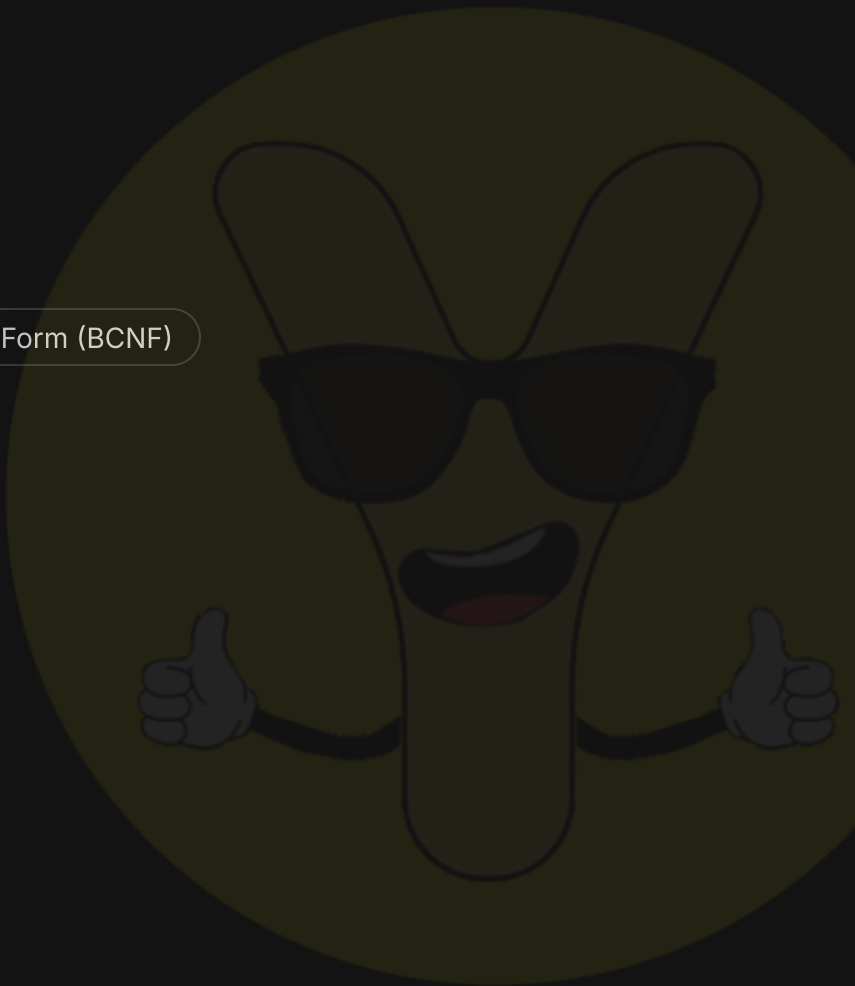
AKTU · 3RD YEAR · SEMESTER 5 · BCS501

Database Management System

Unit 3 — Database Design and Normalization

- Functional Dependencies
- Normal Forms
- First, Second and Third Normal Forms
- Boyce-Codd Normal Form (BCNF)
- Inclusion Dependence
- Lossless Join Decompositions
- Normalization Using FD, MVD, and JDs
- Alternative Approaches to Database Design



DECODED BY

Chirag Jain

Founder, Ycotes · Gold Medalist in Artificial Intelligence, IIT Madras · taught 10,000+ students

Functional Dependencies (FD)

FUNCTIONAL DEPENDENCY

A constraint between two sets of attributes in a relation schema R, denoted as $X \to Y$, stating that for any two tuples t_1, t_2, if $t_1[X] = t_2[X]$, then $t_1[Y] = t_2[Y]$.

- X is called the determinant set, and Y is called the dependent set.
- Trivial FD: $X \to Y$ holds trivially if $Y \subseteq X$.
- Non-trivial FD: $X \to Y$ where $Y \nsubseteq X$.
- Completely non-trivial FD: $X \to Y$ where $X \cap Y = \emptyset$.

Armstrong’s Axioms and Attribute Closure

Armstrong's Axioms are a sound and complete set of inference rules used to derive all functional dependencies implied by a set F.

- Reflexivity Rule: If $Y \subseteq X$, then $X \to Y$.
- Augmentation Rule: If $X \to Y$, then $XZ \to YZ$.
- Transitivity Rule: If $X \to Y$ and $Y \to Z$, then $X \to Z$.
- Secondary Rules: Union ($X \to Y, X \to Z \Rightarrow X \to YZ$), Decomposition, and Pseudotransitivity.
- Attribute Closure X^+: The set of all attributes functionally determined by attribute set X under F.

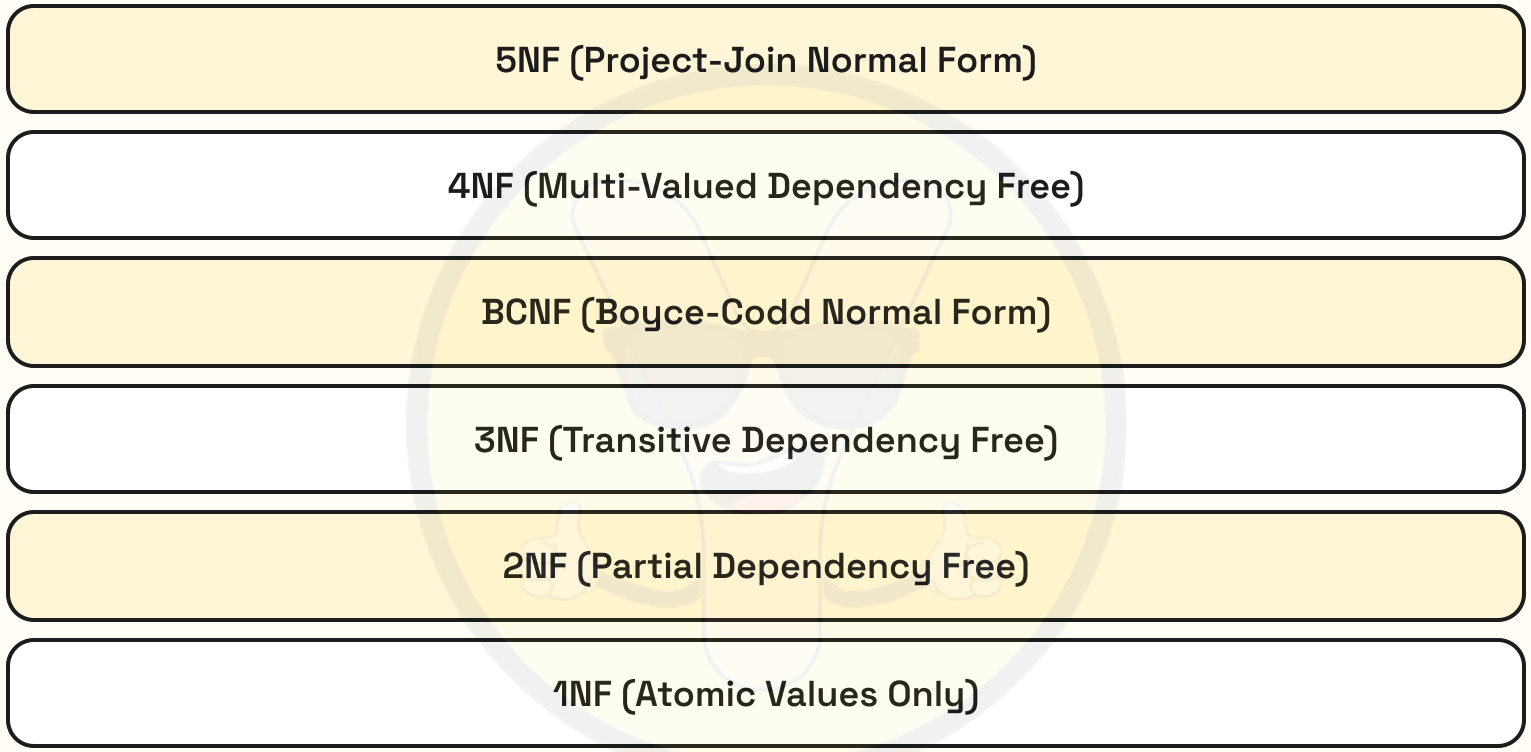
Computing Attribute Closure & Candidate Keys

Given relation schema $R(A, B, C, D, E)$ and FDs $F = \{A \to B, B \to C, CD \to E\}$. Find the closure $\{A, D\}^+$ and check if $\{A, D\}$ is a candidate key.

1. Initialize closure set $X^{(0)} = \{A, D\}$.
2. Apply $A \to B$: Since $A \subseteq \{A, D\}$, add B to get $\{A, B, D\}$.
3. Apply $B \to C$: Since $B \subseteq \{A, B, D\}$, add C to get $\{A, B, C, D\}$.
4. Apply $CD \to E$: Since $CD \subseteq \{A, B, C, D\}$, add E to get $\{A, B, C, D, E\}$.

$\{A, D\}^+ = \{A, B, C, D, E\}$. Since it contains all attributes of R and no proper subset determines R, $\{A, D\}$ is a Candidate Key.

Hierarchy of Database Normal Forms



Hierarchy of Database Normal Forms where each higher level strictly satisfies all lower levels.

First Normal Form (1NF)

FIRST NORMAL FORM (1NF)

A relation schema R is in First Normal Form (1NF) if and only if all attribute values in every tuple are atomic (indivisible) and there are no repeating groups.

- Eliminates multi-valued attributes and composite attributes.
- Each cell in a table must store single, indivisible data values.
- Requires the relation to have a well-defined primary key.
- Conversion: Expand repeating elements into separate tuple rows.

Second Normal Form (2NF)

SECOND NORMAL FORM (2NF)

A relation schema R is in 2NF if it is in 1NF and no non-prime attribute is functionally dependent on any proper subset of any candidate key (No Partial Dependency).

- Prime Attribute: An attribute that is a member of any candidate key.
- Non-Prime Attribute: An attribute that is not part of any candidate key.
- Partial Dependency: $X \rightarrow A$ where X is a proper subset of a candidate key and A is non-prime.
- Conversion: Decompose relation into sub-relations separating partially dependent attributes.

Third Normal Form (3NF)

THIRD NORMAL FORM (3NF)

A relation schema R is in 3NF if for every non-trivial functional dependency $X \to A$, either X is a super key or A is a prime attribute.

- Eliminates Transitive Dependencies: Non-prime attribute determining another non-prime attribute ($X \to Y \to Z$).
- Prevents data redundancy while guaranteeing lossless join and dependency preservation.
- Allows A to be prime even if X is not a super key.
- Conversion: Move transitively dependent attributes into a new relation.

Boyce-Codd Normal Form (BCNF)

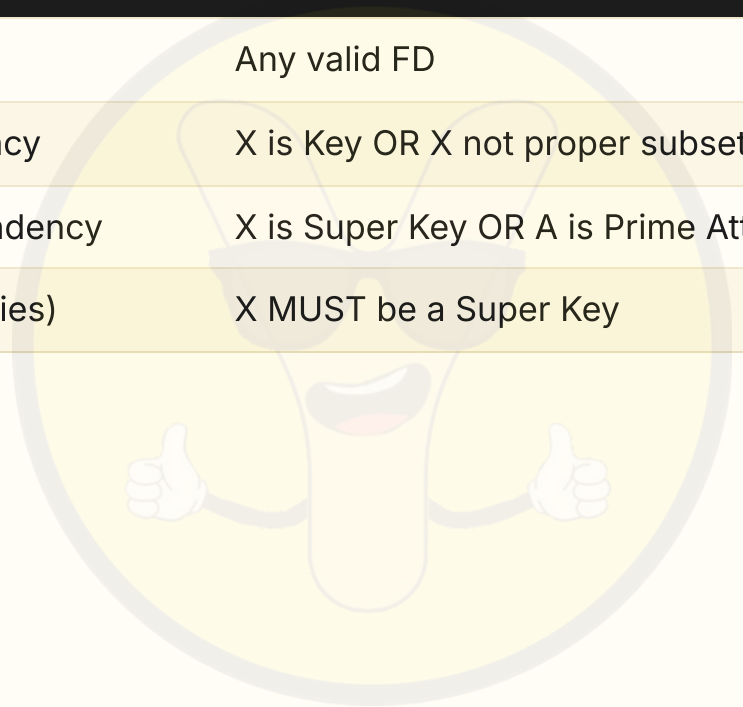
BOYCE-CODD NORMAL FORM (BCNF)

A relation schema R is in BCNF if for every non-trivial functional dependency $X \rightarrow A$, X must be a super key.

- Strict extension of 3NF eliminating anomalies from overlapping candidate keys.
- Does NOT allow A to be prime if X is not a super key.
- Completely eliminates redundancy caused by functional dependencies.
- Trade-off: BCNF decomposition guarantees lossless join, but dependency preservation is not always guaranteed.

Comparison of Relational Normal Forms

Normal Form	Key Requirement	Allowed Rule for X → A	Dependency Preservation
1NF	Atomic attribute domains	Any valid FD	Always preserved
2NF	1NF + No Partial Dependency	X is Key OR X not proper subset	Always preserved
3NF	2NF + No Transitive Dependency	X is Super Key OR A is Prime Attribute	Always preserved
BCNF	Strict 3NF (No Key Anomalies)	X MUST be a Super Key	Not guaranteed



Inclusion Dependency (IND)

INCLUSION DEPENDENCY (IND)

A statement that attribute values in a set of columns of relation R_1 must also exist in a set of columns of relation R_2, written as $R_1[X] \subseteq R_2[Y]$.

- Formal generalization of Referential Integrity and Foreign Key constraints.
- Formally: $\forall t_1 \in R_1, \exists t_2 \in R_2$ such that $t_1[X] = t_2[Y]$.
- Used to assert reference correctness between distinct tables.
- Helps identify inter-relational redundancy during relational database design.

Lossless Join Decomposition

A decomposition of schema R into R_1 and R_2 is lossless if natural join restores the original relation without spurious tuples.

- Formal Identity: $R = \Pi_{R_1}(R) \bowtie \Pi_{R_2}(R)$.
- Necessary & Sufficient Condition: Common attributes $R_1 \cap R_2$ must form a super key for at least one sub-relation.
- Condition Formula: $(R_1 \cap R_2) \rightarrow R_1$ OR $(R_1 \cap R_2) \rightarrow R_2$.
- Spurious Tuples: Extra incorrect tuples generated during natural join of lossy tables.
- Ensures zero loss of information during relational normalization.

Testing Lossless Join Decomposition

Relation $R(A, B, C)$ with FD $F = \{A \rightarrow B\}$ is decomposed into $R_1(A, B)$ and $R_2(A, C)$. Verify if decomposition is lossless.

1. Find common attribute set: $R_1 \cap R_2 = \{A, B\} \cap \{A, C\} = \{A\}$.
2. Determine schema attributes: $R_1 = \{A, B\}$ and $R_2 = \{A, C\}$.
3. Compute closure of common attributes: $\{A\}^+ = \{A, B\}$ under FD $A \rightarrow B$.
4. Check key condition: Since $\{A\}^+$ contains all attributes of R_1, $A \rightarrow R_1$ holds.

The decomposition into $R_1(A, B)$ and $R_2(A, C)$ is LOSSLESS because common attribute A is a super key for R_1.

Dependency Preservation in Decomposition

Decomposition preserves dependencies if all original FDs can be verified using individual sub-relations without taking natural joins.

- Projection of FDs F_i: Subset of FDs in F^+ that contain only attributes present in R_i.
- Preservation Condition: $(F_1 \cup F_2 \cup \cdots \cup F_k)^+ = F^+$.
- Advantage: Enforcement of integrity constraints requires no expensive join operations.
- 3NF Decomposition: Guarantees both Lossless Join AND Dependency Preservation.
- BCNF Trade-off: Guarantees Lossless Join, but may sacrifice Dependency Preservation.

Multi-Valued Dependency (MVD) & 4NF

MULTI-VALUED DEPENDENCY (MVD) AND 4NF

An MVD $X \twoheadrightarrow Y$ states that a given X value uniquely determines a set of Y values independent of remaining attributes. Schema R is in 4NF if for every non-trivial MVD $X \twoheadrightarrow Y$, X is a super key.

- MVD represents independent 1-to-many relationships in a single relation.
- Trivial MVD: $X \twoheadrightarrow Y$ where $Y \subseteq X$ or $X \cup Y = R$.
- Example: Relation $R(\text{Course}, \text{Teacher}, \text{Textbook})$ with independent choices.
- 4NF eliminates redundancy generated by multi-valued dependency facts.
- Every 4NF relation is strictly guaranteed to be in BCNF.

Join Dependency (JD) & 5NF

JOIN DEPENDENCY (JD) AND 5NF

A relation R satisfies Join Dependency $\bowtie (R_1, R_2, \ldots, R_n)$ if R equals the natural join of its projections on $R_1, R_2, \ldots, R_n$. Schema R is in 5NF (PJNF) if every non-trivial JD is implied by candidate keys.

- 5NF is also known as Project-Join Normal Form (PJNF).
- Handles cyclic relationships among 3 or more attributes that cannot be split into pairs.
- Trivial JD: At least one projection R_i equals the complete relation schema R.
- Ultimate normal form with respect to projection and join operations.
- Guarantees complete elimination of structural join redundancies.

Alternative Approaches to Database Design

Relational database design can be approached using top-down decomposition, bottom-up synthesis, or conceptual modeling.

- Top-Down Approach (Decomposition): Starts with a universal relation schema and iteratively splits it to achieve BCNF/4NF.
- Bottom-Up Approach (Synthesis): Starts with atomic attributes and functional dependencies, then synthesizes minimal 3NF relations.
- ER Modeling Approach: High-level conceptual design converted systematically into relational tables.
- Object-Relational Approach: Integrates object-oriented abstractions with normalized relational tables.

Exam Corner

- **Q1** Define Functional Dependency. Explain Armstrong's Axioms and prove the Union rule using primary axioms.
- **Q2** Given schema R(A, B, C, D, E) and FDs {A → BC, CD → E, B → D, E → A}. Find candidate keys and highest normal form of R.
- **Q3** Differentiate between 3NF and BCNF with a suitable relational example.
- **Q4** Define Lossless Join Decomposition. State and prove the condition for lossless join of two relations.
- **Q5** What is Multi-Valued Dependency (MVD)? Explain Fourth Normal Form (4NF) with an example.
- **Q6** Explain Join Dependency (JD) and Fifth Normal Form (5NF/PJNF).
- **Q7** Compare Top-Down (Decomposition) and Bottom-Up (Synthesis) approaches to database design.

Unit 3 done. You are exam-ready.

Revise the Exam corner once more, then open Unit 4 on theycotes.com. Every AKTU subject, unit by unit, free. Notes by Chirag Jain, Ycotes.

