IAE ACADEMY INSTITUTE FOR ENGINEERS

GATE CS · DA

IAE Academy free study material · Updated 27 October 2026

Database Management and Warehousing Formula Sheet

INSIDE

- 77 formulas in 8 topics
- When to use each one
- Taken from the free topic notes on iaeacademy.in

Online, one page per topic: **iaeacademy.in/study/databases/formula-sheet**

FREE COUNSELLING

Preparing for GATE CS or DA? Talk to IAE Academy.

Or call **+91 84204 36524**

Every day, 10:30 am to 7:30 pm

Chat on WhatsApp

Free. Not for sale. Pass it on.

The entity-relationship (ER) model

UPPER BOUND ON A RELATIONSHIP SET

$$|R| \leq |E_1| \cdot |E_2| \cdots |E_n|$$

For an n-ary relationship set R over entity sets $E_1, \ldots, E_n$. It holds because no two instances relate the same combination of entities. The bound is reached only when every combination is related.

BOUND FROM AN AT-MOST-ONE SIDE

$$|R| \leq |B| \quad \text{if each } b \in B \text{ appears in at most one tuple of } R$$

In a binary 1:N relationship from A to B (each B related to at most one A), the count is at most $|B|$. With total participation of B it equals $|B|$. In a 1:1 relationship both sides are at-most-one, so $|R| \leq \min(|A|, |B|)$.

LOWER BOUND FROM TOTAL PARTICIPATION

$$|R| \geq |E_i| \quad \text{if } E_i \text{ participates totally in } R$$

Every entity of E_i appears in at least one instance, and each instance contains exactly one entity of E_i.

BINARY M:N WITH TOTAL PARTICIPATION ON BOTH SIDES

$$\max(|A|, |B|) \leq |R| \leq |A| \cdot |B|$$

Both bounds can be reached: the lower one by covering every entity with as few pairs as possible, the upper one by relating every pair.

KEY OF A WEAK ENTITY'S RELATION

$$\mathrm{PK}(W) = \mathrm{PK}(\text{owner}) \cup \{\text{partial key of } W\}$$

The owner's key is copied in as a foreign key, normally declared ON DELETE CASCADE so that deleting an owner removes its weak entities. No separate relation is made for the identifying relationship.

KEY OF AN M:N RELATIONSHIP RELATION

$$\mathrm{PK}(R) = \mathrm{PK}(A) \cup \mathrm{PK}(B)$$

Applies to a binary many-to-many relationship. Each key column is also a foreign key.

MINIMUM NUMBER OF TABLES (DEFAULT COUNT)

$$\#\text{tables} = \#\text{entity sets (strong and weak)} + \#\text{multivalued attributes} + \#\text{M:N and n-ary relationship sets}$$

Counts recursive M:N relationship sets too. This is the count after merging every 1:1 and 1:N relationship (identifying relationships included) into an entity's relation. Each 1:1 relationship with total participation on both sides reduces the count by one more. Assumes no specialisation or aggregation.

The relational model: relational algebra and tuple calculus

SUPERKEY CONDITION

$$K \text{ superkey of } R \iff \forall t_1 \neq t_2 \in R : \; t_1[K] \neq t_2[K]$$

Every superset of a superkey (within the attributes of R) is a superkey, so under set semantics the set of all attributes is always a superkey. Candidate keys are the minimal superkeys.

COUNTING SUPERKEYS

$$\text{one candidate key } K: \; 2^{n-|K|}, \qquad \text{two candidate keys } K_1, K_2: \; 2^{n-|K_1|} + 2^{n-|K_2|} - 2^{n-|K_1 \cup K_2|}$$

n is the number of attributes of R. A superkey is any attribute set that contains at least one candidate key. With two keys, inclusion-exclusion subtracts the sets counted twice: those containing $K_1 \cup K_2$.

SELECTION

$$\sigma_\theta(R) = \{\, t \in R \mid \theta(t) \text{ is true} \,\}$$

Keeps whole tuples satisfying the condition; schema unchanged; $|\sigma_\theta(R)| \le |R|$.

PROJECTION

$$\pi_{A_1,\ldots,A_k}(R) = \{\, t[A_1,\ldots,A_k] \mid t \in R \,\}$$

Keeps the listed columns and removes duplicate rows (set semantics). For non-empty R, $1 \le |\pi_L(R)| \le |R|$, with $|\pi_L(R)| = |R|$ whenever the list L contains a superkey.

CARTESIAN PRODUCT

$$|R \times S| = |R| \cdot |S|, \qquad \deg(R \times S) = \deg R + \deg S$$

Pairs every tuple of R with every tuple of S. Clashing attribute names are qualified ($R.A$, $S.A$) or renamed with ρ.

UNION AND SET DIFFERENCE

$$R \cup S = \{\, t \mid t \in R \vee t \in S \,\}, \qquad R - S = \{\, t \mid t \in R \wedge t \notin S \,\}$$

Defined only for union-compatible relations: the same degree, with the i-th attributes of both having the same domain. By convention the result takes its attribute names from the first operand. Union is commutative and associative; difference is neither.

CARDINALITY BOUNDS FOR SET OPERATIONS

$$\max(|R|,|S|) \le |R \cup S| \le |R| + |S|, \quad 0 \le |R \cap S| \le \min(|R|,|S|), \quad \max(0, |R| - |S|) \le |R - S| \le |R|$$

Set semantics, R and S union-compatible. Also $|R \cup S| = |R| + |S| - |R \cap S|$ and $|R - S| = |R| - |R \cap S|$. The upper bounds of $\cup$ and $-$ are reached when $R \cap S = \varnothing$; their lower bounds when the smaller relation is contained in the larger.

THETA JOIN

$$R \bowtie_\theta S = \sigma_\theta(R \times S)$$

$0 \le |R \bowtie_\theta S| \le |R| \cdot |S|$. An equi-join is a theta join whose condition is only equalities; it keeps both copies of the compared attributes.

NATURAL JOIN

$$R \bowtie S = \pi_{\text{all attributes, common ones once}}\Big(\sigma_{R.C_1 = S.C_1 \wedge \cdots \wedge R.C_m = S.C_m}(R \times S)\Big)$$

$C_1, \ldots, C_m$ are the attributes with the same name in both relations, and the result has degree $\deg R + \deg S - m$. With no common attribute it equals $R \times S$; when the two schemas are identical it equals $R \cap S$. A `NULL` never matches, not even another `NULL`. Natural join is commutative and associative (up to column order).

INTERSECTION

$$R \cap S = R - (R - S)$$

Union-compatible relations; $|R \cap S| \le \min(|R|,|S|)$.

DIVISION

$$R \div S = \pi_X(R) - \pi_X\big((\pi_X(R) \times S) - R\big)$$

$R(X,Y)$ and $S(Y)$, where the attributes Y of S are a proper subset of those of R and X is the rest. The result has schema X (degree $\deg R - \deg S$): the X-values that appear in R paired with every tuple of S. If S is empty, the formula gives $R \div S = \pi_X(R)$.

DIVISION SIZE BOUNDS

$$|R \div S| \le |\pi_X(R)|, \qquad |R \div S| \le \left\lfloor \frac{|R|}{|S|} \right\rfloor \quad (S \neq \varnothing)$$

Each X-value in the result needs its own $|S|$ tuples in R, one for each tuple of S, and these tuples are different for different X-values.

OUTER JOIN COUNTS

$$|R \bowtie^F S| = |R \bowtie S| + u_R + u_S$$

$|R \bowtie S|$ is the inner (natural) join; u_R and u_S are the numbers of tuples of R and of S that match no tuple of the other relation. Left outer adds only u_R, right outer only u_S.

NATURAL JOIN SIZE BOUNDS

$$0 \le |R \bowtie S| \le |R| \cdot |S|$$

If the common attributes form a key of S, each tuple of R matches at most one tuple of S, so $|R \bowtie S| \le |R|$. If they are also a foreign key of R referencing S with no `NULL` in it, the size is exactly $|R|$.

LARGEST VALUE WITHOUT AGGREGATION

$$\pi_A(R) - \pi_{R.A}\big(\sigma_{R.A<T.A}(R \times \rho_T(R))\big)$$

$\rho_T(R)$ is a renamed copy of R. The subtracted part is every A-value that some other value of A exceeds, so what remains is the maximum value of A. Use $>$ in place of $<$ for the minimum. This assumes A contains no `NULL`.

TRC EXPRESSION

$$\{\, t \mid P(t) \,\}$$

The result is the set of all tuples t for which formula P is true. t is the only free variable of P; every other tuple variable must be bound by a quantifier.

SQL

LOGICAL EVALUATION ORDER OF A QUERY BLOCK

$$\text{FROM} \to \text{WHERE} \to \text{GROUP BY} \to \text{HAVING} \to \text{SELECT} \to \text{DISTINCT} \to \text{ORDER BY}$$

Rows are filtered by WHERE before grouping; groups are filtered by HAVING after aggregation; ORDER BY runs last and does not change which rows are returned.

BAG COUNTS FOR SET OPERATIONS

$$\begin{aligned} \text{UNION ALL:} &\quad m + n \\ \text{INTERSECT ALL:} &\quad \min(m, n) \\ \text{EXCEPT ALL:} &\quad \max(m - n,\, 0) \end{aligned}$$

Here m and n are the number of copies of one particular tuple in the first and second operand. Without ALL, each of these results has at most one copy of every tuple.

AGGREGATES AND NULL

$$\text{AVG}(c) = \frac{\text{SUM}(c)}{\text{COUNT}(c)}$$

Both SUM and COUNT here skip NULLs, so AVG divides by the number of non-NULL values, not by the number of rows; when c has NULLs, AVG(c) differs from SUM(c)/COUNT(). *If every value of c is NULL (or there are no rows), SUM, AVG, MIN and MAX return NULL and COUNT(c) returns 0, while COUNT()* returns the number of rows (0 only when there are no rows).

ROW COUNT OF AN EQUI-JOIN ON ONE COLUMN

$$|R \bowtie_{R.k=S.k} S| = \sum_{v \neq \text{NULL}} c_R(v)\, c_S(v)$$

Here c_R(v) and c_S(v) are the numbers of rows of R and S with key value v. Sum over non-NULL values v only: a NULL key never satisfies equality, so it matches nothing.

ROW COUNTS OF OUTER JOINS

$$|R \text{ LEFT JOIN } S| = |R \bowtie S| + \#\{\text{unmatched rows of } R\}$$
$$|R \text{ FULL JOIN } S| = |R \bowtie S| + \#\{\text{unmatched rows of } R\} + \#\{\text{unmatched rows of } S\}$$

A row is unmatched if no row of the other table satisfies the ON condition; rows with a NULL join key are always unmatched. Every row of R appears at least once in R LEFT JOIN S, so the left join has at least |R| rows.

IN, ANY AND ALL

$$x \text{ IN } S \equiv x = \text{ANY } S, \qquad x \text{ NOT IN } S \equiv x \neq \text{ALL } S$$

x op ANY S is TRUE if the comparison is TRUE for at least one value in S; x op ALL S is TRUE if it is TRUE for every value in S. On an empty S, ANY is FALSE and ALL is TRUE.

NEGATION IN THREE-VALUED LOGIC

$$\text{NOT } T = F, \quad \text{NOT } F = T, \quad \text{NOT } U = U$$

Because NOT UNKNOWN is still UNKNOWN, a row failing both p and NOT p can exist: WHERE p OR NOT p does not return every row when p can be UNKNOWN.

Integrity constraints, functional dependencies and normal forms

ARMSTRONG'S AXIOMS (SOUND AND COMPLETE)

$$\text{Reflexivity: } Y \subseteq X \Rightarrow X \to Y$$
$$\text{Augmentation: } X \to Y \Rightarrow XZ \to YZ$$
$$\text{Transitivity: } X \to Y,\ Y \to Z \Rightarrow X \to Z$$

Sound: they derive only FDs implied by F. Complete: they derive every FD in the closure F^+.

DERIVED RULES

$$\text{Union: } X \to Y,\ X \to Z \Rightarrow X \to YZ$$
$$\text{Decomposition: } X \to YZ \Rightarrow X \to Y,\ X \to Z$$
$$\text{Pseudotransitivity: } X \to Y,\ WY \to Z \Rightarrow WX \to Z$$

All follow from the three axioms. Decomposition applies to the right side only; $XY \to Z$ does not give $X \to Z$.

ATTRIBUTE CLOSURE

$$X^+ = \{A \mid X \to A \text{ is derivable from } F\}$$

Algorithm: start with result = X; repeatedly, for any FD $U \to V$ in F with $U \subseteq$ result, add V; stop when nothing is added.

MEMBERSHIP TEST AND SUPERKEY TEST

$$X \to Y \in F^+ \iff Y \subseteq X^+ \qquad X \text{ is a superkey of } R \iff X^+ = R$$

Use the closure algorithm for both. A candidate key is a minimal superkey: removing any attribute breaks $X^+ = R$.

COUNTING SUPERKEYS

$$\text{One candidate key } K,\ |R| = n: \quad 2^{n-|K|}$$
$$\text{Keys } K_1, K_2: \quad 2^{n-|K_1|} + 2^{n-|K_2|} - 2^{n-|K_1 \cup K_2|}$$

A superkey is any set containing at least one candidate key. For more keys, extend by inclusion and exclusion over unions of keys. Example: keys AB, BC, BD in R(A,B,C,D) give $3 \cdot 4 - 3 \cdot 2 + 1 = 7$.

MAXIMUM NUMBER OF CANDIDATE KEYS

$$\text{at most } \binom{n}{\lfloor n/2 \rfloor} \text{ candidate keys for } n \text{ attributes}$$

No candidate key contains another, and this bound is reached (for example, when every set of $\lfloor n/2 \rfloor$ attributes is a key).

3NF AND BCNF CONDITIONS

$$\text{3NF: } \forall\, X \to A \text{ non-trivial}: \ X \text{ superkey} \ \vee \ A \text{ prime}$$
$$\text{BCNF: } \forall\, X \to A \text{ non-trivial}: \ X \text{ superkey}$$

For the relation R on which F is given, it is enough to test the FDs of F after splitting right sides into single attributes. For a part R_i of a decomposition, test the FDs of F^+ that lie inside R_i (the projection), not just those of F.

LOSSLESS-JOIN TEST FOR A BINARY DECOMPOSITION

$$R_1 \bowtie R_2 = R \iff (R_1 \cap R_2) \to (R_1 - R_2) \ \text{ or } \ (R_1 \cap R_2) \to (R_2 - R_1) \ \text{ in } F^+$$

Conditions: $R_1 \cup R_2 = R$ and F is the only set of constraints. The common attributes must be a superkey of at least one of the two parts. For more than two parts, use the tableau (chase) test.

DEPENDENCY PRESERVATION

$$\Big(\bigcup_i F_i\Big)^+ = F^+ \qquad \text{where } F_i = \{X \to Y \in F^+ \mid XY \subseteq R_i\}$$

Practical test for each $X \to Y$ in F: set $Z = X$, and repeat $Z \leftarrow Z \cup \big((Z \cap R_i)^+ \cap R_i\big)$ for every R_i, with closures computed using F, until Z stops growing. The FD is preserved if $Y \subseteq Z$.

File organisation and indexing

BLOCKING FACTOR (UNSPANNED)

$$bfr = \left\lfloor \frac{B}{R} \right\rfloor$$

Records per block. The floor is essential: with unspanned records the leftover $B - bfr \cdot R$ bytes in each block are unused.

NUMBER OF DATA BLOCKS

$$b = \left\lceil \frac{r}{bfr} \right\rceil$$

Blocks needed for r records; round up because a partly filled last block still occupies a block.

BINARY SEARCH ON A SORTED FILE

$$\text{binary search cost} = \lceil \log_2 b \rceil$$

Applies only to the field the file is sorted on, and counts block accesses, not record comparisons. This is the usual convention in problems; the strict worst case of binary search is $\lfloor \log_2 b \rfloor + 1$, which is one more only when b is a power of 2. Follow the convention the question states.

INDEX BLOCKING FACTOR (FAN-OUT)

$$bfr_i = \left\lfloor \frac{B}{V + P} \right\rfloor$$

Entries per index block, where an entry is a key plus a block pointer (replace P by P_r if the entries hold record pointers).

SINGLE-LEVEL INDEX SIZE

$$b_i^{\text{dense}} = \left\lceil \frac{r}{bfr_i} \right\rceil, \qquad b_i^{\text{sparse}} = \left\lceil \frac{b}{bfr_i} \right\rceil$$

Dense: one entry per record. Sparse primary index: one entry per data block. A clustering index on a field with d distinct values needs $\lceil d/bfr_i \rceil$ blocks.

SEARCH WITH A SINGLE-LEVEL INDEX

$$\text{accesses} = \lceil \log_2 b_i \rceil + 1$$

Binary search over the b_i index blocks (same convention as for a sorted file), plus one access to fetch the data block. A non-key secondary index with a level of indirection adds one more access for the block of record pointers.

MULTILEVEL INDEX LEVELS AND COST

$$t = \left\lceil \log_{fo} b_1 \right\rceil + 1 = \left\lceil \log_{fo} r_1 \right\rceil, \qquad \text{accesses} = t + 1$$

$fo = bfr_i$ is the fan-out, r_1 the number of first-level entries and $b_1 = \lceil r_1/fo \rceil$ the number of first-level blocks. The first form is valid for $b_1 \geq 1$ (for $b_1 = 1$, $t = 1$); the second for $r_1 \geq 2$.

B+ TREE INTERNAL NODE ORDER

$$p\,P + (p-1)\,V \leq B$$

Largest p satisfying this; internal nodes hold p block pointers and $p - 1$ keys.

B+ TREE LEAF CAPACITY

$$q\,(V + P_r) + P \leq B$$

Largest number of keys q in a leaf, where the extra P is the next-leaf pointer.

B-TREE NODE ORDER

$$p\,P + (p-1)\,(V + P_r) \leq B$$

Each key in a B-tree node carries a record pointer, so one node type serves internal nodes and leaves.

KEYS STORED IN A B+ TREE OF HEIGHT H

$$\max \text{keys (B+)} = q\,p^{\,h-1}, \qquad \min \text{keys (B+)} = \left\lceil \tfrac{q}{2} \right\rceil \cdot 2 \left\lceil \tfrac{p}{2} \right\rceil^{h-2} \ (h \geq 2)$$

Max: every node full, so p^{h-1} leaves of q keys. Min: the root has 2 children and every other internal node $\lceil p/2 \rceil$, giving $2\lceil p/2 \rceil^{h-2}$ leaves, each with $\lceil q/2 \rceil$ keys.

KEYS STORED IN A B-TREE OF HEIGHT H

$$\max \text{keys (B-tree)} = p^{\,h} - 1, \qquad \min \text{keys (B-tree)} = 2 \left\lceil \tfrac{p}{2} \right\rceil^{h-1} - 1$$

Max: all $(p^h - 1)/(p - 1)$ nodes full with $p - 1$ keys each. Min: the root has 1 key (2 children) and every other node $\lceil p/2 \rceil - 1$ keys.

BLOCK ACCESSES FOR AN EQUALITY SEARCH

$$\text{accesses (B+ tree, record via leaf pointer)} = h + 1$$

One block per level down to the leaf, plus one to read the data record. Leaves holding the record itself need only h.

EXTENDIBLE HASHING DIRECTORY

$$\text{directory entries} = 2^{d}, \qquad \text{entries pointing to a bucket} = 2^{\,d-d'}$$

d is the global depth and d' a bucket's local depth.

Data types and data transformation: normalisation, discretisation, sampling and compression

MIN-MAX NORMALISATION TO [NEW_MIN, NEW_MAX]

$$v' = \text{new}_{\min} + \frac{v - \min}{\max - \min}\,(\text{new}_{\max} - \text{new}_{\min})$$

Linear map of the observed range onto the target range; for [0,1] it reduces to (v - min)/(max - min). Preserves the shape of the distribution and the order of values. Undefined when max = min (all values equal).

Z-SCORE NORMALISATION

$$v' = \frac{v - \bar{x}}{\sigma}$$

Result has mean 0 and standard deviation 1 (under the same convention for sigma). Convention: use the population standard deviation (divide by n) unless the question says sample (divide by n - 1). Not bounded to a fixed range; undefined when sigma = 0. A variant divides by the mean absolute deviation s = (1/n) times the sum of |v - mean| instead of sigma; it is less affected by outliers because deviations are not squared.

DECIMAL SCALING

$$v' = \frac{v}{10^{j}}, \quad j = \text{smallest non-negative integer with } \max|v'| < 1; \quad j = \lfloor \log_{10} \max|v| \rfloor + 1 \text{ if } \max|v| \geq 1$$

Moves the decimal point; j is fixed by the value of largest absolute magnitude. Result lies strictly inside (-1, 1). Examples: max|v| = 986 gives j = 3; max|v| = 1000 gives j = 4.

EQUAL-WIDTH BINNING

$$w = \frac{\max - \min}{k}, \qquad \text{bin } i = [\min + (i-1)w,\ \min + iw)$$

Bins have the same width and usually unequal counts. Convention here: left-closed, right-open, with the last bin closed so the maximum belongs to it.

EQUAL-FREQUENCY (EQUAL-DEPTH) BINNING

$$\text{depth} = \frac{n}{k}$$

Sort the data and put n/k consecutive values in each bin; widths differ. If k does not divide n, bin sizes are floor(n/k) or ceil(n/k) and the question must say which bins take the extra values. Tied values may need to share a bin.

ENTROPY-BASED (SUPERVISED) SPLIT

$$\text{Info}_T(D) = \frac{|D_1|}{|D|}\,\text{Entropy}(D_1) + \frac{|D_2|}{|D|}\,\text{Entropy}(D_2), \qquad \text{Entropy}(D) = -\sum_i p_i \log_2 p_i$$

D_1 and D_2 are the tuples with attribute value at most, and above, the candidate split point T; p_i is the fraction of a set in class i. Choose the T that minimises Info_T(D), which is the same as maximising the information gain Entropy(D) - Info_T(D). Apply recursively to each part until a stopping rule is met.

INCLUSION PROBABILITY, SRSWOR

$$P(\text{tuple in sample}) = \frac{n}{N}, \qquad \text{number of samples} = \binom{N}{n}$$

Sample of n distinct tuples from N, all subsets equally likely.

INCLUSION PROBABILITY, SRSWR

$$P(\text{tuple appears at least once}) = 1 - \left(1 - \frac{1}{N}\right)^n$$

n independent draws, each tuple chosen with probability 1/N per draw; N^n ordered outcomes.

EXPECTED NUMBER OF DISTINCT TUPLES, SRSWR

$$E[\text{distinct}] = N\left[1 - \left(1 - \frac{1}{N}\right)^n\right], \qquad n = N,\ N \to \infty:\ \frac{E[\text{distinct}]}{N} \to 1 - \frac{1}{e} \approx 0.632$$

By linearity of expectation over the N tuples. The limit is the bootstrap fact: a with-replacement sample of size N contains about 63.2% of the distinct tuples, and about 36.8% never appear.

PROPORTIONAL ALLOCATION IN STRATIFIED SAMPLING

$$n_h = n \cdot \frac{N_h}{N}$$

Stratum h of size N_h receives a share of the total sample n in proportion to its size; round so that the n_h sum to n.

SYSTEMATIC SAMPLING

$$k = \frac{N}{n}, \qquad \text{sample} = \{r,\ r + k,\ r + 2k,\ \dots\},\ r \text{ uniform in } \{1, \dots, k\}$$

When N = nk exactly, every tuple has inclusion probability 1/k = n/N, but only k distinct samples are possible.

COMPRESSION RATIO

$$\text{ratio} = \frac{\text{original size}}{\text{compressed size}}$$

A ratio above 1 means the data shrank. Space saving = 1 - compressed/original.

FIXED-LENGTH CODE SIZE

$$\text{bits per symbol} = \lceil \log_2 m \rceil$$

For an alphabet of m >= 2 distinct symbols. A Huffman code never needs more bits per symbol on average than this, and needs fewer for some unequal frequency distributions (not all: probabilities 0.3, 0.3, 0.2, 0.2 give Huffman lengths 2, 2, 2, 2).

AVERAGE CODE LENGTH AND THE ENTROPY BOUND

$$\bar{L} = \sum_i p_i\, l_i, \qquad H = -\sum_i p_i \log_2 p_i, \qquad H \le \bar{L}_{\text{Huffman}} < H + 1$$

p_i is the probability (relative frequency) of symbol i and l_i its code length in bits. No lossless symbol-by-symbol prefix code averages below H; Huffman's code is optimal among them and reaches H exactly when every p_i is a power of 1/2.

Data warehouse modelling: multidimensional schemas, concept hierarchies and measures

CUBOIDS WITH NO HIERARCHIES

$$\text{number of cuboids} = 2^n$$

n dimensions, each with a single level (apart from 'all'). Each dimension is either present or rolled up to all.

CUBOIDS WITH EXACTLY K DIMENSIONS (NO HIERARCHIES)

$$\binom{n}{k}$$

Choose which k of the n dimensions stay; summing over k = 0 to n gives 2^n.

CUBOIDS WITH CONCEPT HIERARCHIES

$$T = \prod_{i=1}^{n} (L_i + 1)$$

L_i is the number of distinct levels of dimension i, excluding the virtual top level 'all'; the +1 accounts for 'all'. A cuboid picks one level (or 'all') per dimension, so this holds for a total order and for a lattice; in a lattice count every level once (day, week, month gives L = 3).

GROUPINGS COMPUTED BY CUBE AND ROLLUP

$$\mathrm{CUBE}(a_1, \ldots, a_n) : 2^n \qquad \mathrm{ROLLUP}(a_1, \ldots, a_n) : n + 1$$

Both counts include the grand total ().

MAXIMUM CELLS IN THE BASE CUBOID

$$\prod_{i=1}^{n} c_i$$

c_i is the number of distinct values of dimension i at its lowest level; this is an upper bound that is reached only if every combination occurs.

CELLS IN THE WHOLE CUBE (NO HIERARCHIES)

$$\prod_{i=1}^{n} (c_i + 1)$$

Each dimension contributes its c_i values or the single value 'all' (written *). Counts every cell of every cuboid, assuming all combinations are non-empty. With hierarchies, replace c_i + 1 by 1 plus the total number of distinct values over all levels of dimension i.

AVERAGE FROM DISTRIBUTIVE PARTS

$$\bar{x} = \frac{\sum_j s_j}{\sum_j n_j}$$

s_j and n_j are the sum and count of partition j. The average of the partition averages differs from this in general; it is guaranteed to equal it only when all n_j are equal.

VARIANCE FROM SUMS (POPULATION)

$$\sigma^2 = \frac{\sum x^2}{N} - \left(\frac{\sum x}{N}\right)^2$$

Needs only count N, sum and sum of squares, each distributive, so variance is algebraic with M = 3. Population variance (divide by N).

Transactions and concurrency control

NUMBER OF POSSIBLE SCHEDULES (INTERLEAVINGS)

$$\frac{(n_1 + n_2 + \cdots + n_k)!}{n_1!\, n_2! \cdots n_k!}$$

For k transactions with n1, ..., nk operations each, keeping each transaction's own order and counting only the order of operations, not their legality. Non-serial schedules = this number minus k!. Example: two transactions with 2 and 3 operations give 5!/(2! 3!) = 10 schedules, of which 2 are serial and 8 non-serial.

NUMBER OF SERIAL SCHEDULES OF K TRANSACTIONS

$$k!$$

Every ordering of whole transactions is one serial schedule.

FROM IAE ACADEMY

Preparing for GATE CS or DA?

IAE Academy teaches GATE CS and DA in Ballygunge, Kolkata, and online. Counselling is free: ask us about batches, fees or your preparation.

FREE COUNSELLING

Chat on WhatsApp

Call +91 84204 36524

Open every day, 10:30 am to 7:30 pm.

Scan to chat on WhatsApp

GATE CS AND DA COURSES

Classroom batches in Ballygunge and online batches. Fees and details are on the course pages.

Scan: courses

See the courses

FREE ON IAEACADEMY.IN

Notes and solved previous-year questions for Database Management and Warehousing, topic by topic.

Scan: the notes

Open the notes

Found this useful? Send it to a friend preparing for GATE.

Send to a friend on WhatsApp

IAE Academy free study material. The formulas come from the free Database Management and Warehousing notes on iaeacademy.in. Prepared and published by IAE Academy, Kolkata. Free to read and free to share, unaltered and with attribution.

IAE ACADEMY INSTITUTE FOR ENGINEERS

GATE coaching in Kolkata since 2012 · EC, EE, IN, CS and IT, DA, BT and CE · in the classroom and online

227A Rash Behari Ave, Ballygunge, Kolkata 700019

+91 84204 36524 · iaeacademy.in · Updated 27 October 2026

Get directions