**QUESTIONS AND ANSWER**

***Ques:-Data Integrity control uses \_\_\_\_\_\_\_***

(A) Upper and lower limits on numeric data.

(B) Passwords to prohibit unauthorised access to files.(yes)

(C) Data dictionary to keep the data

(D) Data dictionary to find last access of data

***What is Granularity? (NET - 2012 - July)***

(A) The size of database (B) The size of data item

(C) The size of record (D) The size of file

**Explanation:-** Granularity refers to the size of any data item. It could be a tuple, relation, database,anything. So the option is B.

***Which of the following is correct? (GATE – 1999)***

(A) B-trees are for storing data on disk and B+ trees are for main memory.

(B) Range queries are faster on B+ trees.(yes)

(C) B-trees are for primary indexes and B+ trees are for secondary indexes.

(D) The height of a B+ tree is independent of the number of records.

***Ques:-Let R = (A, B, C, D, E, F) be a relational schema with the below mentioned dependencies. Which of the following is a key for R?***

***C -> F, E -> A, EC -> D, A -> B***

A)CD B)EC(yes) C)AE D)AC

***Consider the schema R = {S, T, U, V} and the dependencies S -> T, T -> U, U -> V and V -> S. If R = (R1 and R2) be a decomposition such that R1 intersection R2 = pi then the decomposition is***

(A) not in 2NF

(B) in 2NF but not in 3NF

(C) in 3NF but not in 2NF

(D) in both 2NF and 3NF (yes)

***The best normal form of relation scheme R(A, B, C, D) along with the set of functional dependencies F = {AB→C, AB→D, C→A, D→B} is***

**(Dec-2014)**

(A) Boyce-Codd Normal form

(B) Third Normal form

(C) Second Normal form

(D) First Normal form

View/Hide Ans Correct Answer is B

***Identify the minimal key for relational scheme R(A, B, C, D, E) with functional dependencies F = {A→B, B→C, AC→D}***

(A) A(yes) (B) AE (C) BE (D) CE

***Ques:- A relation R={A,B,C,D,E,F,G} is given with following set of functional dependencies: F={AD→E, BE→F, B→C, AF→G} Which of the following is a candidate key?***

(A) A (B) AB (C) ABC (D) ABD (yes)

***Ques:- Select the ‘False’ statement from the following statements about Normal Forms:***

(A) Lossless preserving decomposition into 3NF is always possible

(B) Lossless preserving decomposition into BCNF is always possible(yes)

(C) Any relation with two attributes is in BCNF

(D) BCNF is stronger than 3NF

***Ques:-Which one of the following statements about normal forms is FALSE?***

a) BCNF is stricter than 3NF

b) Lossless, dependency-preserving decomposition into 3NF is always possible

c) Lossless, dependency-preserving decomposition into BCNF is always possible

d) Any relation with two attributes is in BCNF

Answer: c

Explanation: Achieving Lossless and dependency-preserving decomposition property into BCNF is difficult.

----------------------------------------------------------

***Ques:- Which one of the following statements is FALSE?***

a) Any relation with two attributes is in BCNF

b) A relation in which every key has only one attribute is in 2NF

c) A prime attribute can be transitively dependent on a key in a 3 NF relation.

d) A prime attribute can be transitively dependent on a key in a BCNF relation.

View Answer

Answer: d

**Explanation:** A table is in 3NF if and only if, for each of its functional dependencies X -> A, at least one of the following conditions holds

\* X contains A (that is, X -> A is trivial functional dependency), or

\* X is a superkey, or

\* A should be prime attribute.

***A table has fields F1, F2, F3, F4, and F5, with the following functional dependencies:F1->F3; F2->F4; (F1,F2)->F5 in terms of normalization, this table is in***

a) 1NF(yes) b) 2NF c) 3NF d) None of the mentioned

**Explanation:**Since the primary key is not given we have to derive the primary key of the table. Using the closure set of attributes we get the primary key as (F1,F2). From functional dependencies, “F1->F3, F2->F4”, we can see that there is partial functional dependency therefore it is not in 1NF. Hence the table is in 1NF.

----------------------------------

***Ques:-Which of the following is TRUE?***

a) Every relation in 2NF is also in BCNF

b) A relation R is in 3NF if every non-prime attribute of R is fully functionally dependent on every key of R

c) Every relation in BCNF is also in 3NF

d) No relation can be in both BCNF and 3NF

View Answer

Answer: c

**Explanation:** A relational database table is often described as “normalized” if it is in the Third Normal Form because most of the 3NF tables are free of insertion, update, and deletion anomalies.

----------------------------------------------

***Ques:-The relation schema Student\_Performance (name, courseNo, rollNo, grade) has the following FDs:***

name,courseNo->grade

rollNo,courseNo->grade

name->rollNo

rollNo->name

The highest normal form of this relation scheme is

(a) 2NF (b) 3NF (c) BCNF (d)4NF

Ans: option (b)

**Explanation 1:**

With the help of closure set of attributes we can find the candidate keys: (name,courseNo) and (rollNo,courseNo).

**Explanation 2:-**

C.K = CN, CR

The only non-prime attribute is grade(G)

G is fully functional dependent on each candidate key, so 2NF

There is only one non-prime attribute hence no chance of transitive dependency => 3NF

As, name(N) and rollNo(R) are not a superkey hence, not in BCNF

Hence the correct answer is (b). 3NF

**Explanation 3:-**

For simplicity , I will rename attributes as follows: name= N, course= C, grade= G, and rollnumber= R.

FDs are : NC->G, RC->G , N-> R, R->N.

PKs will be NC,RC. Prime attributes are { N,C,R} and non prime { G}

This relation is in 2NF as non prime attributes {G} is fully functionally dependent on PKs.

This relation is in 3NF as well because it is in 2NF and every FD is in either of the form X->Y,where

--------------------------------------------------------------------------------------------

***Ques:-Consider the following functional dependencies in a database***.

Date\_of\_Birth->Age Age->Eligibility

Name->Roll\_number Roll\_number->Name

Course\_number->Course\_name Course\_number->Instructor

(Roll\_number, Course\_number)->Grade

The relation (Roll\_number, Name, Date\_of\_birth, Age) is

(a) in second normal form but not in third normal form

(b) in third normal form but not in BCNF

(c) in BCNF

(d) in none of the above

Ans: option (d)

Explanation:For the given relation only some of the above FDs are applicable. The applicable FDs are given below:

Date\_of\_Birth->Age

Name->Roll\_number

Roll\_number->Name

Finding the closure set of attributes we get the candidate keys:(Roll\_number,Date\_of\_Birth), and (Name,Date\_of\_Birth) .

On selecting any one of the candidate key we can see that the FD Date\_of\_Birth->Age is a partial dependency. Hence the relation is in 1NF.

***Consider a schema R(MNPQ) and functional dependencies M→N, P→Q. Then the decomposition of R into R1(MN) and R2(PQ) is \_\_\_\_\_\_\_ (UGC - 2017 )***

a) Dependency preserving but not lossless join (yes)

b) Dependency preserving and lossless join

c) Lossless join but not dependency preserving

d) Neither dependency preserving nor lossless join

***Explanation1:-*** here clearly the dependencies are preserved but they are not lossless as both R1 and R2 cannot be formed back again as they donot have a common attribute which is a key so answer is 2

***Explanation2:-*** Schema R(MNPQ) is decomposed into R1(MN) M → N is preserved and R2(PQ) P → Q is also preserved, dependency will be preserved and there will be no loss of any dependency.

----------------------------------------------------------

***Consider a schema R(A, B, C, D) and functional dependencies A -> B and C -> D. Then the decomposition of R into R1 (A, B) and R2(C, D) is***

***(GATE – 2001 )***

(a) dependency preserving and lossless join

(b) lossless join but not dependency preserving

(c) dependency preserving but not lossless join

(d) not dependency preserving and not lossless join

Ans: option (c)

Explanation:

While decomposing a relational table we must verify the following properties:

i) Dependency Preserving Property: A decomposition is said to be dependency preserving if F+=(F1 ∪ F2 ∪ .. Fn)+, Where F+=total functional dependencies(FDs) on universal relation R, F1 = set of FDs of R1, and F2 = set of FDs of R2.

For the above question R1 preserves A->B and R2 preserves C->D. Since the FDs of universal relation R is preserved by R1 and R2, the decomposition is dependency preserving.

ii) Lossless-Join Property:

The decomposition is a lossless-join decomposition of R if at least one of the following functional dependencies are in F+:-

a) R1 ∩ R2 -> R1

b) R1 ∩ R2 -> R2

It ensures that the attributes involved in the natural join ( ) are a candidate key for at least one of the two relations.In the above question schema R is decomposed into R1 (A, B) and R2(C, D), and R1 ∩ R2 is empty. So, the decomposition is not lossless.

----------------------------------------------------------------------------------------------

***Ques:- The relation EMPDT1 is defined with attributes empcode(unique), name, street, city, state, and pincode. For any pincode,there is only one city and state. Also, for any given street, city and state, there is just one pincode. In normalization terms EMPDT1 is a relation in***

(a) 1NF only

(b) 2NF and hence also in 1NF

(c) 3NF and hence also in 2NF and 1NF

(d) BCNF and hence also in 3NF, 2NF and 1NF

Ans: option (b)

**Explanation:**

empcode is unique, therefore it is the primary key. Since the primary key consists of a single attribute there will be no partial dependency, hence the relation is in 2NF.

From the question we get the FDs as below:

pincode -> city, state

street,city,state ->pincode

From the FDs we can see that there are transitive dependencies, hence the table is not in 3NF.

----------------------------------------------------------------------------

***Ques:- Consider the following relational schemes for a library database:***

Book (Title, Author, Catalog\_no, Publisher, Year, Price)

Collection (Title, Author, Catalog\_no)

With the following functional dependencies:

I. Title Author ->Catalog\_no

II. Catalog\_no -> Title, Author, Publisher Year

III. Publisher Title Year -> Price

Assume {Author, Title} is the key for both schemes. Which of the following statements is true?

(a) Both Book and Collection are in BCNF

(b) Both Book and Collection are in 3NF only

(c) Book is in 2NF and Collection is in 3NF

(d) Both Book and Collection are in 2NF only

Ans: option (c)

Explanation:

The relation Collection is in BCNF: Its given that {Author, Title} is the key and there is only one functional dependency (FD) applicable to the relation Collection {i.e. Title Author –>Catalog\_no}.

As per the definitions of the normal forms (given in the explanation of question no. 8) Book is in 2NF.

------------------------------------------------------------------------------

**Ques:-** Let R(A,B,C,D,E,P,G) be a relational schema in which the following FDs are known to hold:

AB->CD DE->P C->E P->C B->G The relation schema R is

(a) in BCNF (b) in 3NF, but not in BCNF

(c) in 2NF, but not in 3NF (d) not in 2NF

Ans: option (d)

Explanation:

From the closure set of attributes we can see that the key for the relation is AB. The FD B->G is a partial dependency, hence it is not in 2NF.

***Consider the table R with attributes A, B and C. The functional dependencies that hold on R are : A -> B, C -> AB. Which of the following statements is/are True ?***

I. The decomposition of R into R1(C, A) and R2(A, B) is lossless.

II. The decomposition of R into R1(A, B) and R2(B, C) is lossy.

(1) Only I (2) Only II

(3) Both I and II(yes) (4) Neither I nor II

Explanation:

1.R1(C,A) R1(A,B)

R1∩R2 =A which is super key for A

so its lossless join .

2.R1(A,B) R1(B,C)

R1∩R2=B which is not super key for any relation so its lossyjoin .

both are correct

***Ques:- If a relation with a Schema R is decomposed into two relations R1 and R2 such that (R1 U R2)=R1 then which one of the following is to be satisfied for a lossless joint decomposition (→ indicates functional dependency)***

(A) ( R1 ∩ R2 ) → R1 or R1 ∩ R2 → R2 (yes) (B) R1 ∩ R2 → R1

(C) R1 ∩ R2 → R2 (D) R1 ∩ R2 → R1 and R1 ∩ R2 → R2

***Division operation is ideally suited to handle queries of the type :***

(A) customers who have no account in any of the branches in Delhi.

(B) customers who have an account at all branches in Delhi.(yes)

(C) customers who have an account in atleast one branch in Delhi.

(D) customers who have only joint account in any one branch in Delhi

***The best normal form of relation scheme R(A, B, C, D) along with the set of functional dependencies*** F = {AB → C, AB → D, C → A, D → B} is

(A) Boyce-Codd Normal form (B) Third Normal form(yes)

(C) Second Normal form (D) First Normal form

***Identify the minimal key for relational scheme R(A, B, C, D, E) with functional dependencies*** F = {A → B, B → C, AC → D}

(A) A(yes) (B) AE (C) BE (D) CE

***Which of the following is true ?***

I. Implementation of self-join is possible in SQL with table alias.

II. Outer-join operation is basic operation in relational algebra.

III. Natural join and outer join operations are equivalent.

(A) I and II are correct. (B) II and III are correct.

(C) Only III is correct. (D) Only I is correct.(yes)

***Fact-less fact table in a data warehouse contains***

(A) only measures (B) only dimensions(yes)

(C) keys and measures (D) only surrogate keys

***Which of the following is true ?***

A. A relation in BCNF is always in 3NF(yes)

B. A relation in 3NF is always in BCNF

C. BCNF and 3NF are same

D. A relation in BCNF is not in 3NF

***Manager’s salary details are to be hidden from Employee Table. This Technique is called as***

(A) Conceptual level Datahiding

(B) Physical level Datahiding

(C) External level Datahiding(yes)

(D) Logical level Datahiding

***Link analysis operation in data mining uses \_\_\_\_\_\_\_\_\_\_\_ technique.***

(A) Classification (B) Association discovery(yes)

(C) Visualisation (D) Neural clustering

**Explanation:-**

There are three specializations of link analysis. These are:

Associations discovery,Sequential pattern discovery,Similar time sequence discovery.

***Which statement is false regarding data independence ?***

(A) Hierarchical data model suffers from data independence

(B) Network model suffers from data independence

(C) Relational model suffers only from logical data independence(yes)

(D) Relational model suffers only from physical data independence

**In a Hierarchical database, a hashing function is used to locate the \_\_\_\_\_\_\_.**

(1) Collision (2) Root

(3) Foreign Key (4) Records

**Explination:**

\*)The data is organize in a tree structure where the nodes represent the records and the branches of the tree represent the fields.

\*)Since the data is organized in a tree structure, the parent node has the links to its child nodes.

\*)If we want to search a record, we have to traverse the tree from the root through all its parent nodes to reach the specific record. Thus, searching for a record is very time consuming.

\*)The hashing function is used to locate the root.

***In DBMS, deferred update means :***

(A) All the updates are done first but the entries are made in the log file later

(B) All the log files entries are made first but the actual updates are done later(yes)

(C) Every update is done first followed by a writing on the log file

(D) Changes in the views are deferred till a query asks for a view

***The best normal form of relation scheme R(A,B,C,D) along with the set of functional dependencies F={AB→C,AB→D,C→A,D→B} is***

A. Boyce-Codd Normal form

B. Third Normal form

C. Second Normal form

D. First Normal form

**Explanation 1 :-**

keys are AB,CD,BC,AD

Here A,B,C,D all are prime attributes

So, no partial or transitive dependency are there

Therefore it is in 3NF

but as C and D are not keys for the relation A,B,C,D , therefore it is not in BCNF

**Explanation 2 :-**

C -> A and D-> B should not be partial dependency, because here in C -> A,

A is a Key element and in D -> B, B is key element.

Partial dependency- { subset of CKs } -> { Non prime attribute key element }.

If all the attributes of a relation are key attributes then relation is automatically in 3NF.

***Cross\_tab displays permit users to view \_\_\_\_\_\_ of multidimensional data at a time.***

(A) One dimension (B) Two dimensions(yes)

(C) Three dimensions (D) Multidimensions

Explanation:- A Crosstab query is a type of select query which allows data to view as both horizontally and vertically so that the data can be more compact and easier to read.

***A Network Schema***

(A) restricts to one to many relationship

(B) permits many to many relationship(yes)

(C) stores Data in a Database

(D) stores Data in a Relation

***A clustering index is defined on the fields which are of type***

***(ugc -2018 Dec, Gate 2008, Isro - 2016 - 60)***

a)non-key and ordering (yes) b)non-key and non-ordering

c)key and ordering d)key and non-ordering

***A clustering index is created when \_\_\_\_\_\_\_. (UGC-NET-2014-JUNE-P2)***

a)Primary key is declared and ordered b)No key ordered

c)Foreign key ordered d)There is no key and no order

**Example:-**

A clustering index = ( non key attribute + ordered )

so C is the answer

***Which of the following statements is TRUE? (NET - 2016 - JULY)***

D1 : The decomposition of the schema R(A, B, C) into R1(A, B) and R2 (A, C) is always lossless.

D2 : The decomposition of the schema R(A, B, C, D, E) having AD → B, C → DE, B → AE and AE → C, into R1 (A, B, D) and R2 (A, C, D, E) is lossless.

(A) Both D1 and D2 (B) Neither D1 nor D2

(C) Only D1 (D) Only D2(yes)

**Explanation:** Only D2 is True because AD is key and present in both the tables.

D1 is not always true because FD’s not given and if we take B->A and C->A then it is lossy decomposition because no common attributes contain key from one of the table

***R(A,B,C,D) is a relation. Which of the following does not have a lossless join, dependency preserving BCNF decomposition? (GATE - 2001)***

Ref https://www.geeksforgeeks.org/gate-gate-cs-2001-question-48

https://www.win.tue.nl/~sidorova/informatica6/some-solutions-chapter7.pdf

(A) A->B, B->CD (B) A->B, B->C, C->D

(C) AB->C, C->AD(yes) (D) A ->BCD

Explanation: Background :

Lossless-Join Decomposition:

Decomposition of R into R1 and R2 is a lossless-join decomposition if at least one of the following functional dependencies are in F+ (Closure of functional dependencies)

R1 ∩ R2 → R1

OR

R1 ∩ R2 → R2

dependency preserving :

Decomposition of R into R1 and R2 is a dependency preserving decomposition if closure of functional dependencies after decomposition is same as closure of of FDs before decomposition.

A simple way is to just check whether we can derive all the original FDs from the FDs present after decomposition.

Question :

We know that for lossless decomposition common attribute should be candidate key in one of the relation.

A) A->B, B->CD

R1(AB) and R2(BCD)

B is the key of second and hence decomposition is lossless.

B) A->B, B->C, C->D

R1(AB) , R2(BC), R3(CD)

B is the key of second and C is the key of third, hence lossless.

C) AB->C, C->AD

R1(ABC), R2(CD)

C is key of second, but C->A violates BCNF condition in ABC as C is not a key. We cannot decompose ABC further as AB->C dependency would be lost.

D) A ->BCD

Already in BCNF.

Therefore, Option C AB->C, C->AD is the answer.

***Consider the following four schedules due to three transactions (indicated by the subscript) using read and write on a data item X, denoted by r(X) and w(X) respectively. Which one of them is conflict serializable ? (UGC 2017 NOV)***

S1 : r1 (X); r2 (X); w1 (X); r3 (X); w2 (X)

S2 : r2 (X); r1 (X); w2 (X); r3 (X); w1 (X)

S3 : r3 (X); r2 (X); r1 (X); w2 (X); w1 (X)

S4 : r2 (X); w2 (X); r3 (X); r1 (X); w1 (X)

(1) S1 (2) S2 (3) S3 (4) S4

EXPLANATION:

Ref: <https://www.youtube.com/watch?v=zuXSXCz_C8o&ab_channel=softtecheducation>

https://www.youtube.com/watch?v=rcC5W8fxbS0&ab\_channel=GATENoteBook

***A table has fields Fl, F2, F3, F4, F5 with the following functional dependencies F1 → F3 F2→ F4 (F1 . F2) → F5 In terms of Normalization, this table is in (GATE - 2005)***

(A) 1 NF (yes) (B) 2 NF (C) 3 NF (D) none

***Consider the schema R = {S,T, U,V} and the dependencies S →T, T → U, U → V and V → S. If R = (R1 and R2)be a decomposition such that R1 ∩ R2 ≠ Ф then the decomposition is (GATE 1999)***

(A) Not in 2NF (B) In 2NF but not in 3NF

(C) In 3NF but not in 2NF (D) In both 2NF and 3NF (yes)

***Explanation1 :*** R1∩R2 ≠ Ø means there is common attribute in R1 and R2.

Now if we choose a decomposition positively then we can choose something like R1(S, T, U) and R2(U, V) then we can say that decomposition is lossless because common attribute is U and LHS of every FDs are candidate key, therefore it is in 2NF as well as 3NF. Option (D) is correct.

***Explanation2 :*** There is no partial dependencies here. So it is in 2nd NF.

RHS of every dependencies is key as well as all are prime attributes. so it is in 3NF

***Relation R with an associated set of functional dependencies, F is decomposed into BCNF. The redundancy (arising out of functional dependencies) in the resulting set relations is. (GATE - 2002 )***

(A) Zero (yes)

(B) More than zero but less than that of an equivalent 3NF decomposition

(C) Proportional to the size of F+ (D) Indeterminate

***Explanation1:-*** If a relational schema is in BCNF then all redundancy based on functional dependency has been removed, although other types of redundancy may still exist.

***Explanation2:-*** If a relation schema is in BCNF then all redundancy based on functional dependency has been removed, although other types of redundancy may still exist. A relational schema R is in Boyce–Codd normal form if and only if for every one of its dependencies X → Y, at least one of the following conditions hold:

X → Y is a trivial functional dependency (Y ⊆ X)

X is a super key for schema R

***Relations produced from E-R Model will always be in \_\_\_\_\_ (UGC 2018)***

1 NF (yes) 2 NF 3 NF 4 NF

***Which normal form is considered adequate for normal relational database design?(GATE 1998)***

(A) 2NF (B) 5NF (C) 4NF (D) 3NF

Explanation: 3NF is sufficient because because most of the 3NF tables are free of insertion, update, and deletion anomalies. Moreover, 3NF always ensures functional dependency preserving and lossless. so, option (D) is correct.

***If every non-key attribute is functionally dependent on the primary key then the relation is in \_***

ans 2NF (ugc-2017)

***The relation scheme Student Performance (name, courseNo, rollNo, grade) has the following functional dependencies:***

***name, courseNo → grade***

***rollNo, courseNo → grade***

***name → rollNo***

***rollNo → name***

***The highest normal form of this relation scheme is (GATE 200)***

(A) 2 NF (B) 3 NF (C) BCNF (D) 4NF

Explanation: For easy understanding let’s say attributes (name, courseNo, rollNo, grade) be (A,B,C,D). Then given FDs are as follows:

AB->D, CB->D, A->C, C->A

Here there are two Candidate keys, AB and CB.

Now AB->D and CB->D satisfy BCNF as LHS is superkey in both.

But, A->C and C->A, doesn’t satisfy BCNF. Hence we check for 3NF for these 2 FDs.

As C and A on RHS of both the FDs are prime attributes, they satisfy 3NF.

Hence for the whole relation the highest normal form is 3NF.

***For a database relation R(a,b,c,d), where the domains a, b, c, d include only atomic values, only the following functional dependencies and those that can be inferred from them hold: { a → c, b → d } This relation is (Gate - 1997, NET - 2017-IIIP)***

(A) in first normal form but not in second normal form

(B) in second normal form but not in first normal form

(C) in third normal form

(D) None of the above

Explanation:-

2NF -> There should not be any partial dependencies

3NF -> There should not be any transitive dependency and right side functional dependency is a prime attribute

Here candidate key is {ab} Because {ab}+ = {a, b, c, d}

There is a partial dependency so anser is A

Ref: <https://testbook.com/question-answer/for-a-database-relation-ra-b-c-d-where-the-do--5f900e109bb5977ca4414f6b>

**The process of removing details from a given state representation is called \_\_\_.**

**(NET – 2020)**

(1) Extraction (2) Mining (3) Selection (4) Abstraction(yes)

***The data node and name node in HADOOP are(NET - 2020)***

(1) Worker Node and Master Node respectively (yes)

(2) Master Node and Worker Node respectively

(3) Both Worker Nodes

(4) Both Master Nodes

***Consider a Bs-tree with key size 10 bytes, block size 512 bytes, data pointer is of size 8 bytes and block poisnter is 5 bytes. Find the order of B-tree?***

**Explanation:-**

n \* (block pointer) + (n-1) \* (key) + (n-1) \* (record pointer) <= BlockSize

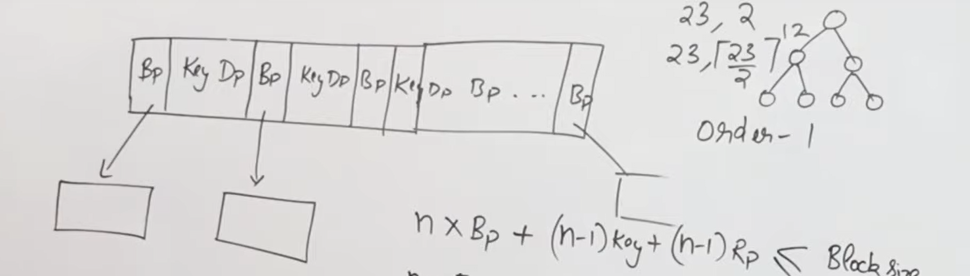
n \* 5 + (n-1)(10+8) <= 512

23n - 18 <= 512

23n <= 530

n <= 23.04

n = 23 ie., maximum children is 23 and minimum children is 2

Ref [https://www.youtube.com/watch?v=yGSHbjA4w0o](https://www.youtube.com/watch?v=yGSHbjA4w0o)

***In a B-Tree, each node represents a disk block. Suppose one block holds 8192 bytes. Each key uses 32 bytes. In a B-tree of order M there are M – 1 keys. Since each branch is on another disk block. We assume a branch is of 4 bytes. The total memory requirement for a non-leaf node is***

(A) 32 M – 32 (B) 36 M – 32 (C) 36 M – 36 (D) 32 M – 36

Explanation:-

Here BlockPointer = 4 Bytes; KeySize = 32 Bytes

BlockSize = 8192 Bytes; RecordPointer is not given. So assume that it is zero

= m \* BlockPointer + (m-1)(key + RecordPointer)

= m \* 4 + (m-1)(32+0)

= 4m + 32m - 32

= 36m - 32 //Hence B is the correct answer

