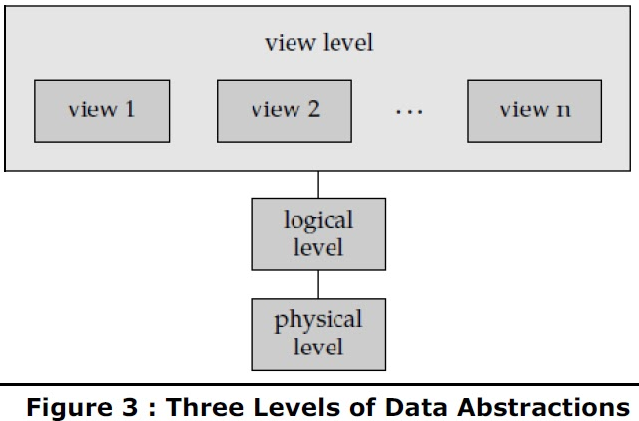
**Three levels of abstraction :-**

**Physical level:** This is the lowest level of data abstraction. It describes how data is actually stored in database.

**Logical level:** It describes what data is stored in database.

**View level:** Highest level of data abstraction. This level describes the user interaction with database system.

****

**Association:-**

Association is a simple relationship between two classes.

**Aggregation:-**

\*) Aggregation is a special type of Association. It is known as “Has-A” relationship.

\*) Aggregation is the process of compiling information on an object,

thereby abstracting a higher level object. So,an entity Person is derived by aggregating

the characteristics name, house\_no., city and social security number(SSN).

\*)A relationship represents a connection between two entity types that are conceptually at the same level. Sometimes you may want to model a ‘has-a’, ‘is-a’ or ‘is-part-of’ relationship, in which one entity represents a larger entity (the ‘whole’) that will consist of smaller entities (the ‘parts’). This special kind of relationship is termed as an aggregation.

**Composition**

Composition is a special type of Aggregation. It is known as “Is-A” relationship.

**Generalization**

Generalization is the process of extracting shared characteristics from two or more classes, and combining them into a generalized superclass. Shared characteristics can be attributes, associations, or methods.

Generalization is a relationship between a Parent and its Derived class. It is nothing but inheritance.

**Specialization**

Specialization means creating new subclasses from an existing class.

**Specialization Hierarchy** - has the constraint that every subclass participates as a subclass in only one class/subclass relationship, i.e. that each subclass has only one parent. This results in a tree structure.

Eg: Tree Topology

**Specialization Lattice** - has the constraint that a subclass can be a subclass of more than one class/subclass relationship.

Eg : DAG topology

**CARDINALITY :-**

In tables, the number of rows (or tuples) is called the cardinality.

**ENTITY INTEGRITY:-**

Entity Integrity is the mechanism the system provides to maintain primary keys. The primary key serves as a unique identifier for rows in the table. Entity Integrity ensures two properties for primary keys: The primary key for a row is unique; it does not match the primary key of any other row in the table.

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

**External Schema** :-

Defines view of the database for particular Users

**Normal Form** :-

* A relation schema R is in **second normal form** (**2NF**) if every non-prime attribute A in R is fully functionally dependent on the primary key
* A relation schema R is in **third normal form** (**3NF**) if it is in 2NF *and* no non-prime attribute A in R is transitively dependent on the primary key.

Non key attributes are non-transitively dependent on the primary key.

* A relation schema R is in **Boyce-Codd Normal Form** (**BCNF**) if whenever an FD X 🡪 A holds in R, then X is a superkey of R

It is not always possible to satisfy all three design goals:

* BCNF.
* Lossless join.
* Dependency preservation.

**LOSSLESS JOIN :-**

For lossless join decomposition, these three conditions must hold true:

1. Union of Attributes of R1 and R2 must be equal to attribute of R. Each attribute of R must be either in R1 or in R2.

Att(R1) U Att(R2) = Att(R)

2. Intersection of Attributes of R1 and R2 must not be NULL.

Att(R1) ∩ Att(R2) ≠ Φ

3. Common attribute must be a key for at least one relation (R1 or R2)

Att(R1) ∩ Att(R2) ->Att(R1) or Att(R1) ∩ Att(R2) -> Att(R2)

**These 3 rules are called ARMSTRONGS AXIOMS**

1. Reflexivity:

If Y is a subset of X, then X->Y

Examples: AB->A, ABC->AB, etc

2. Augmentation:

If X ->Y, then XZ -> YZ

Examples: If A->B, then AC->BC

3 Transitivity:

If X ->Y, and Y -> Z, then X -> Z

4. Union Rule:

ifX->Y, X->Z, then X->YZ

5. Pseudo-Transitivity Rule:

If X->Y, WY->Z, then XW->Z

.Decomposition

If{A → BC} and {A → B}, then {A → C}

.Composition

If X → Y and Z → W then {X, Z} → {Y, W}

**Trivial Functional Dependency:-**

If A holds B {A → B}, where A is a subset of B, then it is called a Trivial Functional Dependency. Trivial always holds Functional Dependency.

**Non-Trivial Functional Dependency:-**

If A holds B {A → B}, where B is not a subset A, then it is called as a Non-Trivial Functional Dependency.

**DBMS**

**Dead Lock Prevention :-**

**Wait-Die Scheme:-**

When a transaction T1 requests data item X held by transaction T2, deadlock prevention protocol decides to allow T1 to wait or to roll-back based on the following conditions;

Condition 1: If timestamp of T1 is smaller than the timestamp of T2, ie, T1 started before T2, then allow T1 to wait for T2 to release lock on X.

Condition 2: If timestamp of T1 is larger than the timestamp of T2, i.e, T1 started after T2, then roll-back T1. [Also, let T1 starts again with the same timestamp and request X in a random amount of time.]

**Wound-Wait Scheme:-**

When a transaction T1 requests data item X held by transaction T2, deadlock prevention protocol decides to allow T1 to wait or to roll-back based on the following conditions;

Condition 1: If timestamp of T1 is larger than the timestamp of T2, ie, T1 started after T2, then allow T1 to wait for T2 to release lock on X.

Condition 2: If timestamp of T1 is smaller than the timestamp of T2, i.e, T1 started before T2, then roll-back T2. That is, the data item requested by T1 will be preempted from T2 and T2 is rolled-back.

**Dead Lock Avoidance :-**

1. Wait For Graph

This is a simple method available to track if any deadlock situation may arise. For each transaction entering into the system, a node is created. When a transaction Ti requests for a lock on an item, say X, which is held by some other transaction Tj, a directed edge is created from Ti to Tj. If Tj releases item X, the edge between them is dropped and Ti locks the data item.

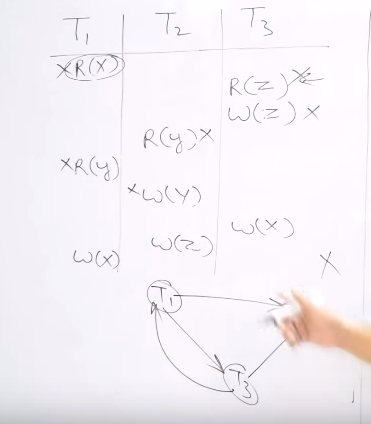
<https://exploredatabase.blogspot.in/search?q=thomas>

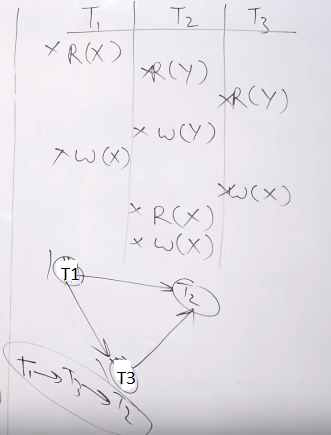
**CONCURRENCY CONTROL**

**CONFLICT SERIALIZABILITY**

Operations are conflict, if they satisfy all three of the following conditions :

1. They belong to different transactions
2. They access the same data item
3. At least one of the operation is a write operation.





Refer Video Tutorial and https://www.geeksforgeeks.org/precedence-graph-testing-conflict-serializability/

**RAID SERVER:-**

http://www.ecs.umass.edu/ece/koren/architecture/Raid/basicRAID.html

-Non-Redundant (RAID Level 0)

-Mirrored (RAID Level 1)

-Memory-Style(RAID Level 2)

-Bit-Interleaved Parity (RAID Level 3)

-Block-Interleaved Parity (RAID Level 4)

-Block-Interleaved Distributed-Parity (RAID Level 5)

-P+Q redundancy (RAID Level 6)

-Striped Mirrors (RAID Level 10)

**JOINS:-**



**DISTRIBUTED DATABASE MANAGEMENT SYSTEM**

**The three dimensions of distribution transparency are −**

**Location transparency:-**

Location transparency ensures that the user can query on any table(s) or fragment(s) of a table as if they were stored locally in the user’s site.

**Fragmentation transparency:-**

Fragmentation transparency enables users to query upon any table as if it were unfragmented. Thus, it hides the fact that the table the user is querying on is actually a fragment or union of some fragments. It also conceals the fact that the fragments are located at diverse sites.

**Replication transparency:-**

Replication transparency ensures that replication of databases are hidden from the users. It enables users to query upon a table as if only a single copy of the table exists.

Whenever a user updates a data item, the update is reflected in all the copies of the table. However, this operation should not be known to the user.

**DATA WARE HOUSING**

In the OLAP world, there are mainly two different types: Multidimensional OLAP (MOLAP) and Relational OLAP (ROLAP). Hybrid OLAP (HOLAP) refers to technologies that combine MOLAP and ROLAP.

**MOLAP**

This is the more traditional way of OLAP analysis. In MOLAP, data is stored in a multidimensional cube. The storage is not in the relational database, but in proprietary formats.

**ROLAP**

This methodology relies on manipulating the data stored in the relational database to give the appearance of traditional OLAP's slicing and dicing functionality. In essence, each action of slicing and dicing is equivalent to adding a "WHERE" clause in the SQL statement.

**HOLAP**

HOLAP technologies attempt to combine the advantages of MOLAP and ROLAP. For summary-type information, HOLAP leverages cube technology for faster performance. When detail information is needed,HOLAP can "drill through" from the cube into the underlying relational data.

**Different views regarding the design of a data warehouse:-**

**The top-down view** - This view allows the selection of relevant information needed for a data warehouse.

**The data source view** - This view presents the information being captured, stored, and managed by the operational system.

**The data warehouse view** - This view includes the fact tables and dimension tables. It represents the information stored inside the data warehouse.

**The business query view** - It is the view of the data from the viewpoint of the end-user.

**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

***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)

December-2014.html

***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

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

b) 2NF

c) 3NF

d) None of the mentioned

Answer: a

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(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***

(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 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