**DBMS**

**(IMPORTANT\*)**

**1a).What do you mean by Normalisation?**

* Normalisation is a systematic approach of decomposing(breaking up) tables
* To eliminate data redundancy
* To eliminate Anomalies like Insertion,Deletion,Updation Anamolies.

Insertion Anamoly:

Suppose for a new admission the student has yet not opted for any subjects,then we have to insert NULL there.

Updation Anamoly:

To avoid inconsistency,if a student occurs twice/more in a table,address will have to be updated in all such rows.

Deletion Anamoly:

sIf a student who has only one subject,temporarily drops it,then deletion of row ,deletes entire record.

|  |  |  |  |
| --- | --- | --- | --- |
| S\_id | S\_name | S\_address | Subject\_opted |
| 401 | Adam | Noida | Bio |
| 402 | Alex | Panipat | Maths |
| 403 | Smith | Jammu | Maths |
| 404 | Adam | Noida | Physics |

**\*1b).Define the Normal Forms? \*Decompose tables.**

* First Normal Form:

A relation is in 1NF if

-No duplicate rows

-Each cell is single valued

-Entries in column are of same kind.

* Second Normal Form:

A relation is in 2NF if

-It is in 1NF

-Every non-key attribute is fully dependent on each candidate key.

* Third Normal Form:

A relation is in 3NF if

-It is in 2NF

-Every non-key is non-transitively(directly) dependent on each candidate key.

-NOTE: TRANSITIVE MEANS A->B ,B->C THEN A->C

* BCNF (Boyce-Codd Normal Form) :

A relation is in BCNF if

-A relation is in 3NF

-Every determinant (LHS of FD) is a candidate key

**1c)What is the difference between 3NF and BCNF?**

A 3NF relation is not in BCNF if all holds true:

* There are more than one candidate key
* Candidate keys are composite keys
* Candidate keys are overlapping(have FD on others)

**2a)What is Functional Dependency?**

* A functional dependency is a relationship between two attributes
* Typically between the primary key and other non-key attribute within the table.
* X->Y (Eg:Roll->Name)
* Attribute Y is functionally dependent on attribute X if value of X,uniquely determines value of Y .

**2b)What do you understand by lossless-join decomposition and lossy decomposition?**

A decomposition of relation schema R<S,F> into a relation schema Ri is said to be the following when:

|  |  |
| --- | --- |
| Lossless Join | Lossy Join |
| Natural join of projection of Ri gives the original relation R | After natural join,actual relation is not obtained and extra informations are visible. |

**3a)What is Transaction?**

* Collection of operations into a **single logical unit** of work
* access and update data items.
* Transaction access data using two operations:

-read(X):

transfers data item X from DB to a local buffer of T

-write(X):

transfers data item X from local buffer back to DB of T

Eg:

Ti:

read(A)

A:=A-50;

write(A)

read(B);

B:=B+50;

write(B);

commit;

**\*3b)What do you mean by Transaction property(ACID)?**

* Atomicity:

All or none effect on DB.

Failure cannot lead to partial execution.

* Consistency:

If initially consistent, then T leaves DB consistent.

Assuming isolation/no concurrency.

* Isolation:

Unaware of concurrent transaction.

Or T is finished before other T begins.

* Durability:

Updates not lost once committed.

Even if system failure occurs.

**3c)What do you mean by serializability,conflict,conflict serializability?**

|  |  |  |
| --- | --- | --- |
| Serializability | Conflict | Conflict serializability |
| Concurrent transactions produce an effect equivalent to serial transactions | Instructions Ii and Ij are said to conflict if:   * different T perform on same data item. * at least 1 instruction is write operation. | * The ability in concurrent schedule to produce conflict equivalent(by swapping non conflicting instructions ) * Which also gives the effect of serial schedule * is known as conflict serializability. |

**\*4a)Explain Natural Join Operation and Outer join operation?**

Natural Join Operation:

* Binary operation
* Denoted by |><|
* Definition:

Let the 2 relation be r(R) and s(S).The natural join of r and s is(inside to outside):

-cartesian product

-selection operation

-projection operation

NOTE:IF R AND S HAVE NO COMMON ATTRIBUTE THEN

r |><| s = r X s i.e. Natural join=cartesion product

Theta join:

* Extension to natural join
* Allows to combine a selection and cartesion product into a single operstion.
* Combines the selection and Cartesian product.

Outer Join Operation:

* To avoid loss of information outer join is used.

(In natural join tuples are retrieved on the basis of equality of values from both the relation due to which some information is also lost.)

* It’s 3 forms are:

-Left outer join

-Right outer join

-Full outer join

**\*4b)Explain Outer Joins?**

* Left Outer Join:

1. Takes in all tuples in left relation that did not match with any tuple in the right relation.
2. Fills tuples with NULL values for all the other attributes from the right relation and then adds them to the result.

* Right Outer Join:

1. Takes in all tuples in right relation that did not match with any tuple in the left relation.
2. Fills the tuples with NULL values for all the other attributes from the left relation and then adds them to the result.

* Full Outer Join:

1. Fills in tuples from left relation that did not match any from the right relation
2. Also fills in tuples from right relation that did not match any from the left relation
3. And add them to the result of Natural Join.

**Q).Mention some Relational Algebra Operation?**

**Q).Deadlock and its prevention?**

**Q)Codd’s Rule for relational model?**

**Q)What is DBMS?**