1. Draw a state diagram .and discuss the typical states that a transaction goes through during execution .

**Database Management System**

Transaction states

Begin Active Partialy Commited Commited -> terminated

Failed -> Aborted -> terminated

State diagram for a transaction in a database or computing syatem typically represents the diffrents stage of transaction undergoes during in execution. The transaction is consistence , atomicity , isolation, durable .

States in transaction lifecycle

1. Active -> This is the first stage of a transaction . when the transaction instructions are being executed .
2. Partialy commited -> the transaction has finished its final operation .but the changes are not saved to the database .
3. Commmited -> After completing all read write operations , if the changes are made permanent on the database then the state wil be change in committed state.
4. Failed-> further execution transaction is stopped and its bought into failure state.
5. Aborted ->I in aborted state the dbms recovery system performance one and two actions
6. 1. Kill the transaction
7. 2 . restart the transaction
8. Terminated-> if there is not any rollback or the transaction comes from committed state then the system is consistency and ready for new transaction and the old transaction is terminated .
9. Logical data independence : It mainly concerned about the structure or the changing data definition.. This is hard to achieve . it is concerned wit the conceptual schema .

Logical data independence is the capacity to the conceptual schema without having change the external schema .

Physical data independence : It mainly concern about how the data is stored into the system.

This easy to achieve . physical deta independence is capacity to change the physical schema without change conceptual schema .

Logical data independence dependence is harder to achieve because logical data indepence changes structure of the definnition of the data base .

1. Difference between database schema and Database state .

|  |  |  |
| --- | --- | --- |
| Subject | Database schema | Database state |
| Definition | Database schema is a skeletal structure in data sets how data is organized . | Database state refers to the content of a data base at any point in time |
| Static / dynamic | Static changes frequently | Dynamic changes with data base operaion |
| Content | Metadata about the data base structure | Actual data store in table |
| Example | Table definition ,constarints | Rows data in the table |

1. What is schedule ?

Schedule is a chronological execution sequence of multiple transaction .

**Integrity constraints** : integraty constraints is a rule of data base to maintain data accuracy , coonsistency, reliability .

4 type of integrity constraints –

1. Domain integrity constraints: value of column should be valid .
2. Entity integrity constraints: I must have been primary key in table . there are never two primary key .
3. Referential integrity constraints: foreign key have been in table .
4. Key integrity constraints: unique key , candidate key .

**Cardinality** : degree of relaionship

4 type of relationship

1. One to one : two entities connected with each other one to one (must have been one primary ) we can reduce the table in one to one relationship .
2. One to many : foreign key is there . because one to many so no unique . many table mai unique part is help to er relationship model .
3. Many ro one : foreign key is there . because many to one so no unique . many table mai unique name .
4. Many to many : here combined key , mean composite key here no primary key . here no reduce in table , because no uniquely attributes . many to many relationship model helps ER model to implement relationship model .

Foreign key : it is a set of attributes that refferences to primary key of same table and another table . RAID - Redundant array of independent disk . this is used for reduce redundant in data .

**Functional Dependency :**

Functional dependency is method which describes the relationship between the attributes .

|  |  |
| --- | --- |
| **Trivial functional dependency** | **Non trivial functional dependency** |
| 1. X-> Y y is a subset of x | 1. X-> Y y is not a subset of x |
| 1. No new information | 1. Reveals relationship |
| 1. Always true, less important | Crucial for data base design |
| 1. Dependent is always subset of determinant. | Dependent is strictly not subset of determinant. |

**Normalization :**

Techniue to remove and reduce reduntancy frm a table .

Ist noraml form : Table should not contain multivalued attributes. Divided two table base table and referencing table .

2nd normal form : there are should not partial dependency , means (lhs shouldbe proper candidate key and rhs should be non prime attributes).

3rd normal form :they should not transitive dependency in table .LHS must be candidate key and sk a or rhs is prim attributes.

**Every relation in BCNF is also in 3nf ; however a relation in 3nf in not necessarily in bcnf .**

**3rd normal form lhs is ck and rhs primary key . always ensures dependency preserving decomposition**

**Bcnf lhs should be ck and sk**

**Bcnf is a stronger normal form than 3rd form.**

**The main difference lies in how they handle certain types of dependencies. BCNF addresses situations where a non-prime attribute might be dependent on part of a candidate key, even if it's not transitively dependent on the entire key. 3NF might not catch these cases, but BCNF does.**

**While BCNF is generally preferred, it's not always necessary or even possible to achieve it without losing some information or introducing complexity. The choice between 3NF and BCNF depends on the specific requirements of the database and the trade-offs between redundancy, complexity, and performance.**