**Assignment 3: Explain the ACID properties of a transaction in your own words. Write SQL statements to simulate a transaction that includes locking and demonstrate different isolation levels to show concurrency control.**

Atomicity: This property ensures that a transaction is treated as a single unit of work, meaning that either all of its operations are successfully completed and committed, or none of them are. There's no in-between state.

Consistency: This property ensures that a transaction brings the database from one valid state to another. In other words, the integrity constraints and rules defined on the database should not be violated by the transaction.

Isolation: This property ensures that the execution of transactions concurrently does not result in interference between them. Each transaction should operate as if it is the only transaction executing on the database, even though multiple transactions may be executing concurrently.

Durability: This property ensures that once a transaction is committed, its effects persist even in the event of system failure. The changes made by the committed transaction are permanently stored in the database and cannot be lost.

**ASSIGNMENT2**

Create table Employee (EMPNO INT NOT NULL,EMPNAME VARCHAR (10),JOB VARCHAR (50),MRG INT ,hiredate date,sal int,comm int,DEPTNO INT );

INSERT INTO Employee VALUES (7369, 'SMITH','WIPRO',NULL,NULL, 800,NULL, 20);

INSERT INTO Employee VALUES (7499, 'ALLEN',NULL,NULL,NULL, 1600,NULL, 30);

INSERT INTO Employee VALUES (7521, 'WARD',NULL,NULL,NULL, 1250,NULL, 30);

INSERT INTO Employee VALUES (7566, 'JONES',NULL,NULL,NULL, 2975,NULL, 20);

INSERT INTO Employee VALUES (7654, 'MARTIN',NULL,NULL,NULL, 1250,NULL, 30);

INSERT INTO Employee VALUES (7369, 'BLAKE',NULL,NULL,NULL, 2850,NULL, 30);

INSERT INTO Employee VALUES (7369, 'CLARK',NULL,NULL,NULL, 2450,NULL, 10);

INSERT INTO Employee VALUES (7788, 'SCOTT',NULL,NULL,NULL, 3000,NULL, 20);

INSERT INTO Employee VALUES (7839, 'KING',NULL,NULL,NULL, 800,NULL, 10);

INSERT INTO Employee VALUES (7844, 'TURNER',NULL,NULL,NULL, 1500,NULL, 30);

INSERT INTO Employee VALUES (7876, 'ADAMS',NULL,NULL,NULL, 1100,NULL, 20);

INSERT INTO Employee VALUES (7900, 'JAMES',NULL,NULL,NULL, 950,NULL, 30);

INSERT INTO Employee VALUES (7902, 'FORD',NULL,NULL,NULL, 3000,NULL, 20);

INSERT INTO Employee VALUES (7934, 'MILLER',NULL,NULL,NULL, 1300,NULL, 10);

--Retrieve employees with a salary greater than 2000:

--Select employees where the SAL column is greater than 2000.

SELECT EMPNAME,sal FROM Employee WHERE sal>2000;

--Retrieve employees hired after January 1, 1990:

--Select employees where the HIREDATE column is greater than January 1, 1990.

SELECT EMPNAME FROM Employee WHERE hiredate > '1990-01-01' ;

--Retrieve employees with a commission:

--Select employees where the COMM column is not null.

SELECT \* FROM Employee WHERE COMM IS NOT NULL;

--Retrieve employees in department 30:

--Select employees where the DEPTNO column is equal to 30.

SELECT \* FROM Employee WHERE DEPTNO=30;

--Retrieve employees whose manager is 7788:

--Select employees where the MGR column is equal to 7788.

SELECT \* FROM Employee WHERE MRG=7788;

--Retrieve employees with a salary between 2000 and 3000:

--Select employees where the SAL column is between 2000 and 3000.

SELECT \* FROM Employee WHERE SAL BETWEEN 2000 AND 3000;

--Retrieve employees with a job title containing the word 'MANAGER':

--Select employees where the JOB column contains the word 'MANAGER'.

SELECT \* FROM Employee WHERE JOB LIKE '%MANAGER%' ;

--Retrieve employees hired in the year 1981:

--Select employees where the HIREDATE column year is equal to 1981.

SELECT \* FROM Employee WHERE YEAR( hiredate)=1981;

--Retrieve employees with a salary greater than the average salary:

--Select employees where the SAL column is greater than the average salary of all employees.

SELECT \* FROM Employee WHERE SAL >(SELECT AVG(sal) FROM Employee);

SELECT \* FROM Employee