Assignment 1
Instructor: Dr. ...
Name : Tilak Barde
SR# : 16512141

Ans. to the Q. No. 1

Two properties of data independence provided in a DBMS are

a. Logical Data Independence: The ability to change the logical schema (i.e., add/remove attributes) without requiring changes to application programs.

b. Physical Data Independence: The ability to change the physical storage structures or access methods (i.e., reorganizing indexes, moving files) without affecting the logical schema or application programs.

Data independence is important because they decouple applications from storage details, reducing maintenance costs. In addition, they allow the system to adapt to growth, optimization and technological change without disrupting users or developers.

To think about an example situation, let a "movie cast" database that stores actors records in a ACTORS table/relation. Now, if a new attribute "phone no." (logical change) is added to the table, existing applications that only query "name" and "age" remain unaffected. This is logical data independence. Next, if the database administrator changes how actor data is indexed (i.e., creating an index on "first_name" so that the DBMS can quickly find a specific name without scanning the whole table) or moves it from HDD to SDD (physical change), queries and reports still work the same way. This is called physical data independence.

Ans. to the Q. No. 2 (a)

Primary key for each relation and the assumptions/conditions under each key would be valid are given below in the table.

Relation	Primary key	Assumptions/conditions
BRANCH(branchName, address, city, sales, manager)	branchName	(i) Each branch has a unique name. (ii) branchName cannot be NULL. (iii) branchName is immutable (does not change over time)
CUSTOMER(cID, firstName, lastName, address, city, birthDate, phoneNum, totalAssets)	cID	(i) cID is unique for each customer. (ii) cID can't be NULL. (iii) cID is immutable once assigned.
LOAN(loanNum, branchName, amount, duration, interest)	loanNum	(i) loanNum is unique for each loan. (ii) loanNum cannot be NULL. (iii) loanNum is immutable once assigned. (iv) branchName references an existing BRANCH (referential constraint).
BORROWER(cID, loanNum)	(cID, loanNum)	(i) In case of joint loan, multiple customers can have the same loan number. On the other hand, one customer can have multiple loan, which means multiple loan numbers can associate with the same cID. So, the combination of (cID, loanNum) should be the primary key. (ii) The combination of cID and loanNum is unique. (iii) cID and loanNum can't be NULL. (iv) cID must reference an existing CUSTOMER. (v) loanNum must reference an existing LOAN.
ACCOUNT(actNum, branch, balance, type)	actNum	(i) actNum is unique for each account. (ii) actNum can't be NULL. (iii) actNum is immutable once assigned. (iv) branch references an existing BRANCH.
DEPOSITER(cID, actNum)	(cID, actNum)	(i) The same customer can deposit to multiple accounts and multiple customers can deposit to the same account. So, the combination of (cID, actNum) is chosen as the primary key. (ii) The combination of (cID, actNum) must be unique. (iii) cID and actNum cannot be NULL. (iv) cID must reference an existing CUSTOMER. (v) actNum must reference an existing ACCOUNT.

Ans. to the Q. No. 2(b)

Foreign key relation	Foreign key(s)	References primary key(s)	Primary-to-foreign key references
LOAN	branchName	BRANCH.branchName	Each loan must reference an existing branch.
BOROWER	cID, loanNum	CUSTOMER.cID, LOAN.loanNum	Each borrower must reference an existing customer and an existing loan.
DEPOSITER	cID, actNum	CUSTOMER.cID, ACCOUNT.actNum	Each depositor must reference an existing customer and an existing account.