Integrity Constraint means condition for constraining saving, deleting and modifying to guarantee accuracy, consistency of database.

1. Key Constraints  
   Keys are the entity set that is used to identify an entity within its entity set uniquely  
   example)  
   Possible

|  |  |  |  |
| --- | --- | --- | --- |
| Index(key) | name | job | income |
| 0 | A | solider | 1000 |
| 1 | B | SW Engineer | 2000 |
| 2 | C | CEO | 5000 |

Impossible

|  |  |  |  |
| --- | --- | --- | --- |
| Index(key) | name | job | income |
| 0 | A | solider | 1000 |
| 1 | B | SW Engineer | 2000 |
| 1 | C | CEO | 5000 |

1. Entitiy integrity constraints  
   the entity integrity constraints states that primary key value cant’be null.

Impossible

|  |  |  |  |
| --- | --- | --- | --- |
| Index(key) | name | job | income |
| 0 | A | solider | 1000 |
| NULL | B | SW Engineer | 2000 |
| 2 | C | CEO | 5000 |

1. Domain constraints  
   Domain constraints can be defined as the definition fo a valid set of values for an attribute  
     
   Impossible  
   (Gender, name, job and income can be duplicated)

|  |  |  |  |
| --- | --- | --- | --- |
| Index(key) | name | job | income |
| 0 | A | solider | 1000 |
| 1 | B | SW Engineer | A |
| 2 | C | CEO | 5000 |

1. Referential integrity constraints  
   It can be specified between two tables. In case of referential integrity constraints, if a Foreign key in Table 1 refers to Primary key of Table 2 then every value of the Foreign key in Table 1 mush be null or available in Table 2  
     
   Impossible(in below example Block\_No 22 entry is not allowed because it is not present in 2nd table  
   Table 1

|  |  |  |  |
| --- | --- | --- | --- |
| Index(key) | name | job | Block\_No |
| 0 | A | solider | 20 |
| 1 | B | SW Engineer | 21 |
| 2 | C | CEO | 22 |

Table2

|  |  |
| --- | --- |
| Block\_No | Block Location |
| 20 | Chandigarh |
| 21 | Punjab |
| 25 | Delhi |

ACID is reduced word that means four critical attributes. It means Atomicity, Consistency, Reliability, Isolation and Durability.

1. Lost update  
   lost update occurs when two or more transactions read the same data and then update it based on the value they read, leading to one of the updates overwriting the others. This situation results in the loss of updates, as the final value reflects only the last update, ignoring the others.  
   ex) two bank transactions attempting to update the balance of the same account. If both transactions retrieve the balance, add their respective amounts, and write the balance back, one of the additions could be lost.
2. Dirty read  
   dirty read happens when a transaction reads data that has been written by another transaction that has not yet been committed. Since the second transaction has not been finalized, its changes could still be rolled back, masking the data read by the first transaction potentially invalid or dirty.  
   ex) transaction A modifies a record. Before A commits the changes, Transaction B reads the modified record. If A then rolls back its changes, B has read data that was never committed and is considered dirty.
3. Unrepeatable read  
   unrepeatable read occurs when a transaction reads the same row of data more than once and gets a different value each time. This can happened if another transaction updates the data between the two reads, leading to inconsistencies within the same transaction.  
   ex) Transaction A reads a row. transaction B then updates or deletes that row and commits the change. If transaction A reads the row again, it finds a different value, leading to and unrepeatable read.
4. Lost transaction  
   It would imply a transaction that, for some reason, does not get executed or whose effects are not reflected in the database, possibly due to system crashes, failures in persistence mechanisms, or synchronization issues in distributed systems.  
   ex) you’re purchasing a book from an online store. You have added the book to your cart, entered your payment information, and clicked the confirm purchase button, initiating a transaction. But system error occurred. The result is what could be considered a ‘lost transaction’  
   a. financially : The amount has been deducted from your credit card, indicating a transaction took place.  
   b. Inventory-wise : The book is still marked as available because the inventory wasn’t updated.  
   c. Record-wise : The sales database has no record of the sale, meaning there’s no easy way to correlate the deduction on your card with any specific purchase in the store’s system