***Primary Index***

1. It is an ordered file whose records are of fixed length with two fields.
2. Only based on the primary key.
3. The total number of entries in the index is the same as the number of disk blocks in the ordered data file.
4. Primary index is a king of nondense (sparse) index.
5. There may be at most one primary index for a file
6. Needs less storage space.
7. ***Secondary index***
8. It provides a secondary means of accessing a file for which some primary access already exists.
9. May be based on candidate key or secondary key.
10. It has a large number entries due to duplication.
11. Secondary index is a kind of dense index.
12. There may be more than one secondary indexes for the same file.
13. Needs more storage space and longer search time.

***what’s difference between DBMS and RDBMS ?***

DBMS provides a systematic and organized way of storing, managing and retrieving from collection of logically related information. RDBMS also provides what DBMS provides but above that it provides relationship integrity. So in short we can say RDBMS = DBMS + REFERENTIAL INTEGRITY These relations are defined by using “Foreign Keys” in any RDBMS. Many DBMS companies claimed there DBMS product was a RDBMS compliant, but according to industry rules and regulations if the DBMS fulfills the twelve CODD rules it’s truly a RDBMS. Almost all DBMS (SQL SERVER, ORACLE etc) fulfills all the twelve CODD rules and are considered as truly RDBMS.

***The difference between a weak and a strong entity is***

***Strong entity set :***

The relationship between two strong entity set is represented by a diamond symbol

The line connecting strong entity set with the relatiionship is single

Member of a strong entity set is a dominant entity

It has its own primary key and it is represented by a rectangle

***Weak entity set :***

The relationship between one strong entity set is represented by a double diamond sign

The line connecting weak entity set with the relationship is double

Member of a weak entity set is a subordinate entity.

It is represented by a double rectangle

The primary key of a weak entity is found by taking the primary key of the strong entity on which it is existence-dependent, plus the discriminator of the weak entity set.

***Define entity integrity and referential integrity.***

***Entity Integrity***  
Entity integrity ensures each row in a table is a uniquely identifiable entity. For example, the ProductID column of the Products table is a primary key for the table. Entity Integrity ensures two properties for primary keys.The primary key for a row is unique; it does not match the primary key of any other row in the table. The primary key is not null, no component of the primary key may be set to null. The uniqueness property ensures that the primary key of each row uniquely identifies it; there are no duplicates. The second property ensures that the primary key has meaning, has a value; no component of the key is missing.  
  
***Referential Integrity***  
Referential integrity ensures the relationships between tables remain preserved as data is inserted, deleted, and modified. While the Referential Integrity property looks simpler than those for Entity Integrity, the consequences are more complex since both primary and foreign keys are involved.

**Entity Integrity Rule-** If the attribute A of relation R is a prime attribute of R then A cannot accept null values.

**Referential Integrity Rule-** In referential integrity. It is ensured that a value that appears in one relation for a given set of attributes also appears for a certain set of attributes in another relation.

**In SQL,** entity integrity and referential integrity rules are implemented as constraints on the relation called as primary Key constraint and reference Key constraint respectively.

These constraints can be specified with relation at the time of creation of the relations or after the creation of the relations by altering the definition of the relations.

**For example:**

Create Table Dept

              (Deptno Number primary key.

               DName Varchar2(15));

Create Table Emp

               (Empno Number primary Key.

                EName Varchar2(15),

                Job Varchar2(10),

                Deptno Number References Dept(Deptno));

# http://www.enggpedia.com/answers/?qa=blob&qa_blobid=16038232378551665828[?](http://www.enggpedia.com/answers/1411/difference-between-has-a-and-is-a-relationship)

***Difference between HAS-A and IS-A relationship ?***

The relationship between a supertype and its subtypes is also called an **IS-A relationship**.**Entities with an IS-A**relationship should have the same identifier as they represent different aspects of the same thing. **Entities with a HAS-A** relationship represent aspects of different things. They have different identifiers.These relationships do not involve subtypes.

BUILDING with an identifier of buliding Code and subtypes of CLASSROOM, OFFICE and RECREATIONAL have an IS-A relationship.All types of building are identified by building code. All subtypes IS-A BUILDING.

Arelationship between ADVISOR and STUDENT is a HAS-A relationship because a STUDENT HAS-A ADVISOR, ADVISOR is not a type of STUDENT.

*--------------------------------------------------------------------------------*

***Recovery Concepts***  
Recovery from transaction failures means that the db is restored   
to the most recent consistent state just before the time of failure. To do this, the system must keep information about the change that were   
applied to data items by the various transactions. This information is   
typically kept in the system log.

Two main techniques for recovery from non-catastrophic failures.

1. deferred update.
2. immediate update

The ***deferred update*** techniques do not physically update the db on disk until after a transaction reaches its commit point, then the update are recorded in the db. Before reaching commit, all transaction updates are recorded in the local transaction workspace. During commit, the updates are first recorded persistently in the log and then written to the db.

If a transaction fails before reaching its commit point, it will not have changed the db in any way, so UNDO is not needed. It may be necessary to REDO the effect of the operations of committed transaction from the log, because their effect may not yet have been recorded in the db. Hence, differed update is also known as the NO – UNDO/REDO algorithm.

In the ***immediate update***  techniques, the db may be updated by some operations of a transaction before the transaction reaches its commit point. However, these operations are typically recorded in the log on disk by force writing before they are applied to the db, making recovery still possible.

If a transaction fails after reaching its commit point, the effect of its operations on the db must be undone i.e., the transaction must be rolled back. In the general case of immediate update, both undo and redo may be required during recovery. This technique, known as the UNDO/REDO algorithm.

*====================------------------------*