Explicit VS implicit Casting

Explicit Casts

An explicit cast is a cast that you must specifically invoke, with either the CAST AS keywords or with the cast operator (::). The database server does not automatically invoke an explicit cast to resolve data type conversions. The EXPLICIT keyword is optional; by default, the CREATE CAST statement creates an explicit cast.

Implicit Casts

An implicit cast is a cast that the database server can invoke automatically when it encounters data types that cannot be compared with built-in casts. This type of cast enables the database server to automatically handle conversions between other data types.

Pointer to Function

A pointer to a function points to the address of the executable code of the function. You can use pointers to call functions and to pass functions as arguments to other functions. You cannot perform pointer arithmetic on pointers to functions.
The type of a pointer to a function is based on both the return type and parameter types of the function.
A declaration of a pointer to a function must have the pointer name in parentheses. Function parameters have precedence over pointers in declarations, so parentheses are required to alter the precedence and declare a pointer to a function. Without them, the compiler interprets the declaration as a function that returns a pointer to a specified return type. For example:

```
int *f(int a);       /* function f returning an int*                    */
int (*g)(int a);     /* pointer g to a function returning an int        */
```

Difference Between a Pointer to an Array and Array of Pointers

Parameters	Pointer to an Array	Array of Pointers
Uses and Purposes	A user creates a pointer for storing the address of any given array.	A user creates an array of pointers that basically acts as an array of multiple pointer variables.
Alternative Names	It is alternatively known as an array pointer.	These are alternatively known as pointer arrays.
Allocation	One can allocate these during the run time.	One can allocate these during the compile time.
Initialization at Definition	You cannot initialize a pointer to the definition.	You can easily initialize an array at the definition level.
Nature	It is dynamic in nature.	It is static in nature.
Resizing	One can easily resize the allocated memory of a pointer later at any given time.	Once we declare the size of an array, we cannot resize it any time we want according to our requirements.
Type of Storage	A typical pointer variable is capable of storing only a single variable within.	The size of any given array decides the total number of variables that it can store within.