Primary Key vs. Unique Not Null

The key differences between a column defined as **id NUMBER(2) PRIMARY KEY** and one defined as **id NUMBER(2) UNIQUE NOT NULL** are primarily related to their roles and implications within a database table.

Uniqueness and Nullability
A primary key inherently enforces both uniqueness and non-nullability, meaning that each value in the primary key column must be distinct and no null values are permitted. In contrast, a UNIQUE NOT NULL constraint explicitly combines these two properties, ensuring that all values are unique and no null values are allowed.

Number of Keys per Table
Only one primary key can be defined per table, serving as the principal identifier for each record. However, multiple UNIQUE NOT NULL constraints can be applied to different columns within the same table, providing additional unique identifiers.

Indexing
A clustered index is typically created automatically when a primary key is defined, which can influence the physical storage order of data. Conversely, a UNIQUE NOT NULL constraint usually results in the creation of a non-clustered index, which does not dictate the physical order of data.

Foreign Key Relationships
A primary key is the target of foreign key relationships, allowing other tables to reference its values to establish links between data. While a UNIQUE NOT NULL column can also be referenced by a foreign key, the primary key is the standard and preferred choice for this purpose.

Final Answer
The primary key is a single, non-nullable, and unique identifier for each record in a table, often leading to a clustered index, and is the standard target for foreign key relationships. A UNIQUE NOT NULL constraint also ensures uniqueness and non-nullability but allows for multiple such constraints per table, typically results in a non-clustered index, and can be used as a target for foreign keys, although the primary key is generally preferred.