1. Every value in a column or set of columns (a composite key) must be unique: Unique Constraint
2. For every row entered into the table, there must be a value for that column: Not Null Constraint
3. Constraint ensures that the column contains no null values and uniquely identifies each row of the table: Primary Key
4. Specifies a condition for a column that must be true for each row of data: Check Constraint
5. Identifies that table and column in the parent table: Foreign Key
6. An integrity constraint that requires every value in a column or set of columns to be unique: Unique Constraint
7. Designates a column (child table) that establishes a relationship between a primary key in the same table and a different table (parent table): Foreign Key Constraint
8. References one or more columns and is defined separately from the definitions of the columns in the table: Table-Level Constraint
9. Database rule: Constraint
10. Database rule that references a single column: Column-Level Constraint

1. A constraint is a rule enforced on data in a table to ensure accuracy and consistency
2. Column-Level Constraints can only apply to individual columns and Table-Level Constraints can reference multiple columns, making them more flexible but potentially more complex to manage.
3. It is important to give meaningful names to constraints because meaningful names make it easier to understand the purpose of a constraint, facilitate debugging, and simplify database maintenance.
4. Id: NUMBER, length 10, precision 10, scale 0
   Name: VARCHAR2, length 100
   Date_opened: DATE
   Address: VARCHAR2, length 150
   City: VARCHAR2, length 50
   Zip/Postal Code: VARCHAR2, length 20
   Phone: VARCHAR2, length 20
   Email: VARCHAR2, length 100
   Manager_id: NUMBER, length 10, precision 10, scale 0
   Emergency Contact: VARCHAR2, length 50
5. Nullable columns:
   Columns that can have null values: Phone, Emergency Contact
   Non-nullable columns: Id, Name, Date_opened, Address, City, Email