WORKSHEET 6 SQL

1. Out of the four options, A. Commit, C. Rollback, and D. Save point are commands used in TCL
2. Out of the four options, A. Create, C. Drop, and D. Alter are DDL
3. B. SELECT NAME FROM SALES;
4. C. Authorizing Access and other control over Database
5. C. String
6. B. COMMIT
7. A. Parenthesis - (...).
8. C. TABLE
9. D. All of the mentioned
10. A. ASC
11. Denormalization in SQL refers to the process of intentionally introducing redundancy into a database schema in order to improve performance by reducing the number of joins required to access the data.
    Normalization is a process of designing a database schema to minimize data redundancy and dependency, which helps to ensure data consistency and reduce the likelihood of data anomalies. However, in some cases, the cost of performing frequent joins to retrieve the required data can be too high in terms of query performance. This is where denormalization can be useful.
    Denormalization involves adding redundant data to the database schema, either by duplicating data in multiple tables or by adding additional columns to existing tables. This can improve query performance by eliminating the need to perform complex joins, but it can also make data updates more complex and increase the risk of data inconsistencies.
    Denormalization should be used judiciously and only after careful analysis of the specific requirements of the application and the database. In some cases, it may be appropriate to use denormalization to improve performance, while in others, it may be better to optimize the database design or use other techniques such as indexing or caching.
12. In SQL, a database cursor is a database object that allows for the sequential processing of a result set one row at a time. A cursor is typically used to iterate over the rows of a result set and perform some operation on each row. For example, a cursor might be used to update or delete rows in a table based on some criteria, or to perform some complex calculations on the data in the result set.
    To use a cursor, you typically need to define it and then open it to associate it with a specific SELECT statement or other query. Once the cursor is open, you can use various SQL commands to move the cursor through the result set one row at a time, such as FETCH NEXT or FETCH PRIOR. You can also perform various operations on each row, such as reading or modifying the data, or performing some other calculation or processing.
    One of the advantages of using a cursor is that it allows you to process a large result set one row at a time, which can be more efficient than retrieving the entire result set at once. However, cursors can also be resource-intensive and may not be the best choice for all situations. It's important to use cursors judiciously and consider alternative approaches when appropriate.

13. In SQL, there are several types of queries that you can use to retrieve or modify data in a database:
    a. SELECT queries: These are used to retrieve data from one or more tables in a database. SELECT queries can use various clauses such as WHERE, GROUP BY, HAVING, and ORDER BY to filter, group, and sort the results.
    b. INSERT queries: These are used to add new rows to a table in a database. The syntax for an INSERT query typically specifies the name of the table and the values to be inserted into each column.
    c. UPDATE queries: These are used to modify existing rows in a table in a database. The syntax for an UPDATE query typically specifies the name of the table, the columns to be updated, and the new values for each column.
    d. DELETE queries: These are used to remove one or more rows from a table in a database. The syntax for a DELETE query typically specifies the name of the table and a WHERE clause that identifies the rows to be deleted.
    e. CREATE queries: These are used to create new tables, views, indexes, or other database objects. The syntax for a CREATE query varies depending on the type of object being created.
    f. ALTER queries: These are used to modify the structure of an existing table, view, index, or other database object. The syntax for an ALTER query varies depending on the type of object being modified.
    g. DROP queries: These are used to delete an existing table, view, index, or other database object. The syntax for a DROP query typically specifies the name of the object to be deleted.
14. In SQL, a constraint is a rule that you define to enforce data integrity in a database. Constraints are used to prevent invalid data from being inserted, updated, or deleted in a table. They also help to maintain consistency and accuracy of the data in the database. There are several types of constraints that can be defined in SQL:
    a. Primary key constraint: This is a constraint that uniquely identifies each row in a table. It prevents duplicate rows and ensures that each row can be uniquely identified by a single field or combination of fields.
    b. Foreign key constraint: This is a constraint that defines a relationship between two tables in a database. It ensures that the values in a field in one table correspond to the values in a field in another table.
    c. Unique constraint: This is a constraint that ensures that the values in a field or combination of fields in a table are unique. It prevents duplicate values from being inserted or updated in the table.
    d. Not null constraint: This is a constraint that ensures that a field in a table cannot contain a null value. It prevents incomplete or missing data from being inserted in the table.
    e. Check constraint: This is a constraint that defines a condition that must be true for a row to be inserted, updated, or deleted in a table. It allows you to define custom rules for data validation.

    Constraints can be defined when a table is created or modified using the CREATE TABLE or ALTER TABLE statements. Constraints can also be dropped or modified using the ALTER TABLE statement.

15. Auto increment is a feature in many relational database management systems (RDBMS) that allows a field to automatically generate a new value each time a new record is inserted into a table. Auto increment is typically used for primary key fields to ensure that each record in the table has a unique identifier.

    In most RDBMS, auto increment is implemented using an integer field that is set to increase by a value of 1 for each new record that is inserted. The initial value of the field is typically set to 1, and the value increases by 1 for each new record.

    For example, if a table has an auto increment primary key field named "id", and the last record inserted had an "id" value of 100, the next record inserted would have an "id" value of 101. This ensures that each record in the table has a unique "id" value.

    Auto increment is a useful feature because it removes the need for the application to generate a unique identifier for each new record that is inserted into the table. This can simplify the development process and ensure data integrity by preventing duplicate primary keys.