STUDENT ACTIVITY 6.1.1: VOCABULARY STUDY GUIDE

MTA Course: Database Administration Fundamentals
Topic: Master Summary, Part 1
File name: DBAdminFund_SA_6.1.1

Lesson Objective:

6.1: Master review of the Database Administration Fundamentals review lessons.

Part 1: Vocabulary Review for Database Administration Fundamentals

The vocabulary words are listed in order of their appearance in the review lessons. Some words will have more than one definition. The definition is dependent on the words' use within the lesson. For example, the word *table* can mean a group of row and columns or the heart of a database, and both definitions are correct in this course, depending on the context.

1. **Understanding Core Database Concepts**

    1.1. Understand how data is stored in tables.

    This objective may include but is not limited to: understanding what a table is and how it relates to the data that will be stored in the database; columns/fields, rows/records.

    field—a location in a record in which a particular type of data is stored.

    column—a group of items that are listed vertically in a table.

    row—a group of items that are listed horizontally in a table.

    table—a data structure characterized by rows and columns, with data occupying or potentially occupying each cell formed by a row-column intersection.

    record—**a** data structure that is a collection of fields (elements), each with its own name and type. A group of fields across one row.

    variable—the name (label) given to data, or in a database. It is the label for a column.

1.2. Understand relational database concepts.

This objective may include but is not limited to: understanding what a relational database is, the need for relational database management systems (RDBMS), and how relations are established.

relational model—a data model.

Three foundations of a modern database:

   entries—data.
   attributes/fields—characteristics.
   relationships/keys—how data ties together

one-to-one relationship—an association between two tables.

many-to-many relationship—a complex association between two sets of parameters.

parent/child—a relationship between nodes.

key—a unique identifier. Only the primary key is unique.

1.3. Understand data manipulation language (DML).

This objective may include but is not limited to: understanding what DML is and its role in databases.

SQL—Structured Query Language; a database sublanguage used in querying, updating, and managing relational databases. The de facto standard for database products.

DML—data manipulation language; a language that is used to insert data in, update, and query a database. DMLs are often capable of performing mathematical and statistical calculations that facilitate generating reports. DML is also used to delete the data in addition to insert, update, and query.

DDL—data definition language; a language that defines all attributes and properties of a database, especially record layouts, field definitions, key fields, file locations, and storage strategy.

SELECT—a command that chooses data.

INSERT—a command that adds data.

UPDATE—a command that changes data.

DELETE—a command that removes data.

1.4. Understand data definition language (DDL).

This objective may include but is not limited to: understanding how T-SQL can be used to create database objects such as tables and views.

DDL—data definition language; a language that defines all attributes and properties of a database, especially record layouts, field definitions, key fields, file locations, and storage strategy.

DML—data manipulation language; a language that is used to insert data in, update, and query a database. DMLs are often capable of performing mathematical and statistical calculations that facilitate generating reports. DML is also used to delete the data in addition to insert, update, and query.

schema—a description of a database to a database management system (DBMS) in the language provided by the DBMS. A schema defines aspects of the database, such as attributes (fields) and domains and parameters of the attributes.

CREATE—a command that adds items to your database.

ALTER—a command that changes items within your database.

DROP—a command that destroys items within your database.

2. Creating Database Objects

2.1. Choose data types.

This objective may include but is not limited to: understanding what data types are, why they are important, and how they affect storage requirements.

data type—a definition of a set of data that specifies the possible range of values of the set, the operations that can be performed on the values, and the way in which the values are stored in memory.

integer—a whole number.

floating point type—a real number.

character—A–Z or 0–9.

boolean—1 or 0; true or false.

2.2 Understand tables and how to create them.

This objective may include but is not limited to: purpose of tables; creating tables in a database by using proper ANSI SQL syntax.

ANSI SQL syntax—the grammar of the SQL language; the rules governing the structure and content of statements.

> **table**—a data structure usually consisting of a list of entries.

2.3 Create views.

a. This objective may include but is not limited to: understanding when to use views and how to create a view by using T-SQL**.**

> **T-SQL**—T-SQL standards for Transact-SQL. All versions of Transact-SQL are command-driven databases.

> **view**—a virtual table; that is, an object that acts exactly like a table but does not store its own data, instead referencing data that is stored in other tables.

b. *This objective may include but is not limited to:* understanding when to use views and how to create a view by using a graphical designer.

> **view**—a saved query creating a virtual table of the result set of that query.

> **query**—a specific set of instructions for extracting particular data**.**

> **graphical designer**—applications such as Microsoft® Access® that are object-oriented and drag-and-drop driven.

2.4. Create stored procedures and functions.

> *This objective may include but is not limited to:* selecting, inserting, updating, or deleting data.

> **functions**—commands that return a single value written in the command set of SQL.

> **aggregate functions**—commands that return a single value, calculated from values in a column.

> **scalar functions**—commands that return a single value, based on the input value.

> **user-defined functions**—compact, user-written SQL code that can accept parameters and return either a value or a table.

> **stored procedure**—a precompiled group of SQL statements written by users and saved to the database. Stored procedures can accept one or more input parameters and can produce one or more output parameters.

3. Manipulating Data

3.1. Select data.

a. *This objective may include but is not limited to:* utilizing SELECT with simple queries and single conditionals**.**

> **SELECT**—the most-used SQL command for selecting the data from the database.

DISTINCT—returns only distinct (unique) values.

WHERE—specifies which rows to retrieve.

ORDER BY—specifies an order in which to return the rows.

AND—a boolean logical operator that displays a record if both the first condition <u>and</u> the second condition are true.

OR—a boolean logical operator that displays a record if either the first condition <u>or</u> the second conditions are true.

truth table—a table showing the relationships of the Boolean Logical Operators.

b. *This objective may include but is not limited to:* utilizing SELECT with complex queries and multiple conditionals and **UNION** and **INTERSECT** multiple tables**.**

UNION—combine two or more SELECT statements with an OR function.

JOIN—used to query data from two or more tables.

subquery—a query that nests inside another query

INTERSECT—combines two or more SELECT statements with an AND function.

3.2. Insert data.

This objective may include but is not limited to: understanding how data is inserted into a database; how to use INSERT statements.

INSERT INTO—this statement is used to insert a new row into a table.

INSERT INTO with SELECT—this statement is used to insert a new row in a table when a sub-SELECT is used instead of the VALUES clause.

3.3. Update data.

This objective may include but is not limited to: understanding how data is updated in a database and how to write the update data to the database by using the appropriate UPDATE statements; update by using a table.

UPDATE—this statement is used to update existing records in a table.

CASE—this statement creates when-then-else functionality; that is, WHEN "this condition is met," THEN "do this."

3.4. Delete data.

This objective may include but is not limited to: deleting data from single or multiple tables; ensuring data and referential integrity by using transactions.

DELETE FROM—used to delete rows in a table.

TRANSACTIONS—a set of two or more statements grouped into a single unit.

ROLLBACK—the process of reversing changes. ROLLBACK will reverse changes until the prior COMMIT point and not all the actions are reversible.

COMMIT—if all statements are correct within a single transaction, all changes are recorded to the database.

4. **Understanding Data Storage**

   4.1. Understand normalization.

   This objective may include but is not limited to: understanding the reasons for normalization, the five most common levels of normalization, how to normalize a database to third normal form.

   normalization—the process of applying a body of techniques to a relational database to minimize the inclusion of duplicate information.

   normal form—structuring (organizing) information to avoid redundancy and inconsistency and to promote efficient maintenance, storage, and updating.

   1NF—First Normal Form (1NF); this sets a few basic rules for a database:

   - Eliminate duplicative columns from the same table.

   - Create separate tables for each group of related data and identify each row with a unique column (the primary key).

   2NF—Second Normal Form (2NF); the equivalent of 1NF Plus, this has the same rules as 1NF, plus the following:

   - Remove subsets of data that apply to multiple rows of a table and place them in separate tables.

   - Create relationships between these new tables and their predecessors through the use of foreign keys.

   3NF—Third Normal Form (3NF); the equivalent of 2NF Plus, this has the same rules as 2NF, plus the following:

   - Remove columns that are not dependent upon the primary key.

4NF—Fourth Normal Form (4NF); the equivalent of 3NF Plus, this has the same rules as 3NF, plus the following:

- It has no multi-valued dependencies. This form is mostly academic and rarely used.

5NF—Fifth Normal Form (5NF); the equivalent of 4NF Plus, this has the same rules as 4NF, plus the following:

- States that no nontrivial join dependencies exist. This form is mostly academic and rarely used.

4.2. Understand primary, foreign, and composite keys.

This objective may include but is not limited to: understanding the reason for keys in a database, choosing appropriate primary keys, selecting appropriate data type for keys, selecting appropriate fields for composite keys, understanding the relationship between foreign and primary keys.

primary key—a unique key that defines the items in the records.

foreign key—a foreign key in one table points to a primary key in another table.

data integrity—a set of constraints or rules that protect the data.

composite key—a key that is made up of two or more columns.

surrogate key—a key that is not generated by the data of the database but a generated unique key used as a primary key. Also, a key that stands in place of the other key and takes care of business.

4.3. Understand indexes.

This objective may include but is not limited to: understanding clustered and non-clustered indexes and their purpose in a database.

index—an index is a list of some object(s).

B-tree—a balance data structure created for faster search times.

non-clustered index—the standard index used in databases. Non-clustered indexes use the classic B-tree format with pointers.

clustered—a special form of index that uses the B-tree data structure but has eliminated the pointer system with data pages.

5. Administering a Database

5.1. Understand database security concepts.

This objective may include but is not limited to: understanding the need to secure a database, what objects can be secured, what objects should be secured, user accounts, and roles.

security plan—identifies which users can do which action(s) to data in the database.

physical security—the safety of the actual location of the database.

access control—ensures and restricts who can connect and what they can do.

user account—the means of access for people to the database.

role—contains permissions to the database. Used to give groups of users the same permissions.

5.2. Understand database backups and restore.

This objective may include but is not limited to: understanding various backup types, such as full and incremental, importance of backups, how to restore a database.

full backup—a type of backup where all files are copied or backed up.

incremental backup—a type of backup where only files that have been changed since the last backup.

differential backup—a type of backup where only files that have been changed since the last full backup.

replicated services—reproduce or replicate the database and server environment to create a clone of the database and/or the servers that the database is located on.