**ST. XAVIER’S COLLEGE**

**(Affiliated to Tribhuvan University)**

**Maitighar, Kathmandu**

****

**DATABASE MANAGEMENT SYSTEM**

**THEORY ASSIGNMENT#8**

**Submitted by:**

**Shreesha Pokharel**

**013BSCCSIT038**

**Submitted to:**

|  |  |
| --- | --- |
| **Er. Sanjay Kumar Yadav** |  |

**Lecturer**

**Department of Computer Science**

**Date of submission: 27th September, 2015**

1. **Data definition language:**

Data Definition Language (DDL) is a standard for commands that define the different structures in a database. DDL statements create, modify, and remove database objects such as tables, indexes, and users. Common DDL statements are CREATE, ALTER, and DROP.

**1.1 Domain type in SQL:**

CHAR (n): Fixed length character string, with user-specified length n.

VARCHAR (n): Variable length character strings, with user-specified maximum length n.

Null values are allowed in all the domain types. Declaring an attribute to be NOT NULL prohibits null values for that attribute.

**Number Domain Types in SQL**

• INT (also: INTEGER) Integer (a finite subset of the integers that is machine-dependent).

• SMALLINT Small integer (a machine-dependent subset of the integer domain type).

• DECIMAL (p,d) Fixed point number, with user-specified precision of p digits, with n digits to the right of decimal point.

• FLOAT (n) Floating point number, with user-specified precision of at least n digits.

• REAL (also: DOUBLE PRECISION) Floating point and double-precision floating point numbers, with machine-dependent precision.

**Date Domain Types of SQL**

• DATE dates, containing a (4 digit) year, month and date

– E.g. DATE ‘2001-7-27’

• TIMESTAMP date plus time of day – E.g. TIMESTAMP ‘2001-7-27 09:00:30.75’

**1.2 Schema definition in SQL**

The structure of a database system described in a formal language supported by the database management system (DBMS). In a relational database, the schema defines the tables, the fields in each table, and the relationships between fields and tables.

Schemas are generally stored in a data dictionary. Although a schema is defined in text database language, the term is often used to refer to a graphical depiction of the database structure.

1. **Data manipulation language**

A **data manipulation language** (DML) is a family of syntax elements similar to a computer programming **language** used for selecting, inserting, deleting and updating **data** in a database.

* 1. **The select clause**

The SELECT statement is used to select data from a database.

The result is stored in a result table, called the result-set.

SQL SELECT Syntax

SELECT *column\_name*,*column\_name*  
FROM *table\_name*;

and

SELECT \* FROM *table\_name*;

* 1. **The where clause**

The WHERE clause is used to extract only those records that fulfill a specified criterion.

SQL WHERE Syntax

SELECT *column\_name*,*column\_name*

FROM *table\_name*

WHERE *column\_name operator value*;

* 1. **The from clause**

The FROM clause is required in every SELECT statement in which data is being retrieved from tables or views. Use the FROM clause to:

* List the tables and views containing the columns referenced in the select list and in the WHERE clause. The table or view names can be aliased using the AS clause.
* Join types. These are qualified by join conditions specified in the ON clause.
  1. **The rename clause**

You can rename a table or a column temporarily by giving another name known as alias.

The use of table aliases means to rename a table in a particular SQL statement. The renaming is a temporary change and the actual table name does not change in the database.

The column aliases are used to rename a table's columns for the purpose of a particular SQL query.

## Syntax:

The basic syntax of **table** alias is as follows:

SELECT column1, column2....

FROM table\_name AS alias\_name

WHERE [condition];

* 1. **Tuple variable**

Tuple variables can be used in SQL, and are defined in the **from** clause:

**select distinct** *cname, T.loan#*

**from** *borrower* ***as*** *S, loan* ***as*** *T*

**where** *S.loan# = T.loan#*

Note: The keyword **as** is optional here.

These variables can then be used throughout the expression. Think of it as being something like the rename operator.

* 1. **String operations**
  2. **Ordering the display of tuples**
  3. **Duplicate tuples**